



## RELAXAÇÕES CONVEXAS E DESIGUALDADES VÁLIDAS PARA O PROBLEMA DA MOCHILA QUADRÁTICO BINÁRIO

Daniela Cristina Lubke de Mello

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Marcia Helena Costa Fampa

Rio de Janeiro  
Março de 2019

RELAXAÇÕES CONVEXAS E DESIGUALDADES VÁLIDAS PARA O  
PROBLEMA DA MOCHILA QUADRÁTICO BINÁRIO

Daniela Cristina Lubke de Mello

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ  
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)  
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS  
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM  
CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof. Marcia Helena Costa Fampa, D.Sc.

---

Prof. Nelson Maculan Filho, D.Sc.

---

Prof. Luidi Gelabert Simonetti, D.Sc.

---

Prof. Fernanda Maria Pereira Raupp, D.Sc.

---

Prof. Carlile Campos Lavor, D.Sc.

---

Prof. Wendel Alexandre Xavier de Melo, D.Sc.

---

Prof. Philippe Yves Paul Michelon, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2019

Mello, Daniela Cristina Lubke de

Relaxações convexas e desigualdades válidas para o problema da mochila quadrático binário/Daniela Cristina Lubke de Mello. – Rio de Janeiro: UFRJ/COPPE, 2019.

XII, 76 p.: il.; 29, 7cm.

Orientador: Marcia Helena Costa Fampa

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2019.

Referências Bibliográficas: p. 58 – 62.

1. Problema quadrático da mochila. 2. Programação quadrática binária. 3. Relaxação de programação quadrática convexa. 4. Otimização paramétrica. 5. Desigualdades válidas. 6. Problema de separação. I. Fampa, Marcia Helena Costa. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*Para: Cristina Lubke, Edna Gomes  
Lubke (em memória) e Thiago  
Ferreira de Mello.*

# Agradecimentos

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), 142143/2015-4, pelo suporte financeiro.

À minha orientadora Marcia Fampa por todos os ensinamentos passados durante meu doutorado.

Aos professores Maculan, Luidi, Fernanda, Carlile, Wendel e Michelin por aceitarem prontamente avaliar este trabalho.

Aos professores do PESC: Adilson Xavier, Laura Bahiense e Abílio Lucena.

Um agradecimentos especial aos amigos mais mamatas de toda a COPPE: Evandro Macedo, Rebeca Mota, Renan Spencer, Hugo Barbalho, Renan Vicente, Ana Flávia, Israel Mendonça, Brunno Goldstein, Pedro e Dani. Os cafés, almoços e lanches com vocês serviram para tornar a jornada mais leve.

Aos companheiros do laboratório de otimização: Helder, Marilis, Matheus, Jesus, Sávio, Nancy, Orlando, Marco Costa e Hildebrando. Agradeço à PSR pela bolsa fornecida para administrar o LabOtim.

À Fátima Marques e Carolina Vieira, por todo apoio e incentivo. Aos demais funcionários do PESC por sempre estarem dispostos a ajudar.

Aos familiares agradeço por todo apoio e carinho: Fernando, Rosane, Angela Becker, Família Maiani, Walter, Eliana e Camila. Agradeço em especial ao meu esposo Thiago que sempre esteve ao meu lado e a minha mãe Cristina por sempre incentivar meus passos.

Agradeço ao professor Henry Wolkowicz por ter me orientado em Waterloo. Ao amigo Fei Wang por todas as discussões e contribuições. Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), 88881.131629/2016-01, pelo suporte financeiro durante o meu período sanduíche. Finalmente agradeço aos amigos que fiz no Canadá: Bruna, Paulo, Marcela, André, Poliana, Stefan, Lazima, Jimit, Linchuan e Mali.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

## RELAXAÇÕES CONVEXAS E DESIGUALDADES VÁLIDAS PARA O PROBLEMA DA MOCHILA QUADRÁTICO BINÁRIO

Daniela Cristina Lubke de Mello

Março/2019

Orientador: Marcia Helena Costa Fampa

Programa: Engenharia de Sistemas e Computação

Consideramos a relaxação de programação quadrática paramétrica convexa, *convex quadratic programming*, **CQP**, para o problema da mochila quadrático, *quadratic knapsack problem*, **QKP**. Nesta relaxação mantêm-se informações quadráticas parciais da função objetivo original do **QKP**, perturbando a função objetivo para obter-se um termo quadrático côncavo. A parte não côncava, gerada pela perturbação, é linearizada utilizando um procedimento padrão no qual o problema é reescrito no espaço matricial. Apresentamos um método de pontos interiores primal-dual para otimizar a perturbação da função quadrática, em busca de limites superiores mais apertados para o **QKP**. Provamos que o mesmo procedimento que perturba a função objetivo, quando aplicado no contexto de relaxações de programação semidefinida, *semidefinite programming*, **SDP**, para o **QKP**, não pode melhorar o limite superior dado pela relaxação linear **SDP** correspondente. O resultado também pode ser aplicado para os problemas mais gerais de programação quadrática inteira. Finalmente, propomos novas desigualdades válidas para as variáveis matriciais derivadas das desigualdades de cobertura e das desigualdades de mochila, para o **QKP**, e apresentamos problemas de separação para gerar cortes para a solução corrente da relaxação **CQP**. Nossos melhores limites são obtidos ao alternarmos entre a otimização da relaxação quadrática paramétrica e a adição de planos de cortes gerados pelas desigualdades propostas.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

## CONVEX RELAXATION AND VALID INEQUALITIES OF THE BINARY QUADRATIC KNAPSACK PROBLEM

Daniela Cristina Lubke de Mello

March/2019

Advisor: Marcia Helena Costa Fampa

Department: Systems Engineering and Computer Science

We consider a parametric convex quadratic programming, **CQP**, relaxation for the quadratic knapsack problem, **QKP**. This relaxation maintains partial quadratic information from the original **QKP** by perturbing the objective function to obtain a concave quadratic term. The nonconcave part generated by the perturbation is then linearized by a standard approach that lifts the problem to the matrix space. We present a primal-dual interior point method to optimize the perturbation of the quadratic function, in a search for the tightest upper bound for the **QKP**. We prove that the same perturbation approach, when applied in the context of semidefinite programming, **SDP**, relaxations of the **QKP**, cannot improve the upper bound given by the corresponding linear **SDP** relaxation. The result also applies to more general integer quadratic problems. Finally, we propose new valid inequalities on the lifted matrix variable, derived from cover and knapsack inequalities for the **QKP**, and present the separation problems to generate cuts for the current solution of the **CQP** relaxation. Our best bounds are obtained from alternating between optimizing the parametric quadratic relaxation over the perturbation and adding cutting planes generated by the valid inequalities proposed.

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Problema da mochila quadrática 0-1 <b>QKP</b>	1
1.2 Sequência da Apresentação	2
<b>2 Revisão Bibliográfica</b>	<b>3</b>
2.1 Limitantes Superiores	3
2.1.1 Planos superiores	3
2.1.2 Relaxação Lagrangeana	4
2.1.3 Linearização do termo quadrático	4
2.1.4 Relaxação Lagrangeana e Reformulação	5
2.1.5 Decomposição Lagrangeana	5
2.1.6 Programação Semidefinida	6
2.2 Polítopo da Mochila e Desigualdades de Cobertura	10
2.2.1 Polítopo da Mochila 0-1	10
2.2.2 Desigualdades de Cobertura	11
2.2.3 Desigualdades de Cobertura Estendida	11
2.2.4 Procedimento de <i>lifting</i> para Desigualdades de Cobertura	11
<b>3 Método Proposto</b>	<b>13</b>
3.1 Relaxação convexa quadrática paramétrica	14
3.1.1 Perturbação da função objetivo	14
3.1.2 Otimização do problema paramétrico sobre o parâmetro $Q_p$	14
3.2 A relaxação quadrática paramétrica SDP	19
3.3 Desigualdades válidas	20
3.3.1 Adicionando cortes a relaxação	20
3.3.2 Novas desigualdades válidas no espaço <i>lifted</i>	21
3.3.3 Relação de dominância entre as novas desigualdades válidas	28



3.4	Limites inferiores para a solução da relaxação <b>QKP</b> <i>lifted</i> . . . . .	31
<b>4</b>	<b>Outra abordagem pesquisada</b>	<b>32</b>
4.1	Novos Limites Superiores . . . . .	32
4.2	O Algoritmo de plano de cortes . . . . .	34
4.3	Programação Disjuntiva . . . . .	34
4.3.1	Corte Secante . . . . .	35
4.3.2	Cortes Disjuntivos . . . . .	37
<b>5</b>	<b>Resultados Computacionais</b>	<b>41</b>
5.1	Resultados obtidos com plano de cortes e programação disjuntiva . . . . .	41
5.2	Resultados utilizando a relaxação quadrática paramétrica convexa <b>CQP</b> .	47
5.2.1	Estrutura algorítmica . . . . .	47
5.2.2	A influência dos cortes . . . . .	49
5.2.3	Desigualdades válidas agregadas ao método de pontos interiores .	51
<b>6</b>	<b>Conclusões e trabalhos futuros</b>	<b>56</b>
	<b>Referências Bibliográficas</b>	<b>58</b>
<b>A</b>	<b>Detalhamento dos resultados</b>	<b>63</b>
A.1	Tabelas comparando plano de cortes e relaxações da literatura . . . . .	63
A.2	Tabelas comparando cortes propostos e o <b>IPM</b> . . . . .	67

# Lista de Figuras

4.1	Corte secante em uma função quadrática. . . . .	37
4.2	Disjunção de cortes secantes. . . . .	38
4.3	Corte disjuntivo em uma função quadrática. . . . .	40
5.1	Limites encontrados durante a execução dos CPAs com densidade 25 e 50. . . . .	45
5.2	Limites encontrados durante a execução dos CPAs com densidade 75 e 100. . . . .	46
5.3	Média dos gaps de otimalidade apresentados na Tabela 5.2. . . . .	50
5.4	Média dos gaps de otimalidade apresentados na Tabela 5.3. . . . .	53
5.5	Média dos gaps para as tabelas com $n = 30$ itens, 5.4 . . . . .	54
6.1	Resumo da pesquisa desenvolvida e trabalhos futuros. . . . .	57

# Lista de Tabelas

5.1	Gaps obtidos pelas diferentes relaxações do <b>QKP</b> . . . . .	43
5.2	Impacto dos cortes adicionados ao L $\tilde{P}R$ (10 instâncias, $n = 10$ ). . . . .	50
5.3	Média dos resultados para 10 instâncias ( $n = 10$ ). . . . .	52
5.4	Média dos resultados para 10 instâncias ( $n = 30$ ). . . . .	53
5.5	Média dos resultados para 5 instâncias ( $n = 50$ ). . . . .	54
A.1	Gaps obtidos para $n = 50$ itens e $d = 25$ pelas diferentes relaxações do <b>QKP</b> . . . . .	63
A.2	Gaps obtidos para $n = 50$ itens e $d = 50$ pelas diferentes relaxações do <b>QKP</b> . . . . .	64
A.3	Gaps obtidos para $n = 50$ itens e $d = 75$ pelas diferentes relaxações do <b>QKP</b> . . . . .	64
A.4	Gaps obtidos para $n = 50$ itens e $d = 100$ pelas diferentes relaxações do <b>QKP</b> . . . . .	65
A.5	Gaps obtidos para $n = 100$ itens e $d = 25$ pelas diferentes relaxações do <b>QKP</b> . . . . .	65
A.6	Gaps obtidos para $n = 100$ itens e $d = 50$ pelas diferentes relaxações do <b>QKP</b> . . . . .	66
A.7	Gaps obtidos para $n = 100$ itens e $d = 75$ pelas diferentes relaxações do <b>QKP</b> . . . . .	66
A.8	Gaps obtidos para $n = 100$ itens e $d = 100$ pelas diferentes relaxações do <b>QKP</b> . . . . .	67
A.9	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^a$ ,QPR, Tabela 5.3. . . . .	67
A.10	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^b$ ,QPR, Tabela 5.3. . . . .	68
A.11	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^a$ ,I, Tabela 5.3. . . . .	68
A.12	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^b$ ,I, Tabela 5.3. . . . .	69
A.13	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^a$ ,BFGS, Tabela 5.3. . . . .	69
A.14	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^b$ ,BFGS, Tabela 5.3. . . . .	70
A.15	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^a$ ,I,Cuts, Tabela 5.3. . . . .	70
A.16	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^b$ ,I,Cuts, Tabela 5.3. . . . .	71
A.17	Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^a$ ,BFGS,Cuts, Tabela 5.3. . . . .	71

A.18 Resultados para 10 instâncias ( $n = 10$ ) usando $Q_p^b$ ,BFGS,Cuts, Tabela 5.3.	72
A.19 Resultados para 10 instâncias ( $n = 30$ ) usando $Q_p^a$ ,QPR, Tabela 5.4. . . .	72
A.20 Resultados para 10 instâncias ( $n = 30$ ) usando $Q_p^b$ ,QPR, Tabela 5.4. . . .	73
A.21 Resultados para 10 instâncias ( $n = 30$ ) usando $Q_p^a$ ,BFGS, Tabela 5.4. . . .	73
A.22 Resultados para 10 instâncias ( $n = 30$ ) usando $Q_p^b$ ,BFGS, Tabela 5.4. . . .	74
A.23 Resultados para 10 instâncias ( $n = 30$ ) usando $Q_p^a$ ,BFGS,Cuts, Tabela 5.4.	74
A.24 Resultados para 10 instâncias ( $n = 30$ ) usando $Q_p^b$ ,BFGS,Cuts, Tabela 5.4.	75
A.25 Resultados para 5 instâncias ( $n = 50$ ) usando $Q_p^a$ ,QPR, Tabela 5.5. . . . .	75
A.26 Resultados para 5 instâncias ( $n = 50$ ) usando $Q_p^b$ ,QPR, Tabela 5.5. . . . .	75
A.27 Resultados para 5 instâncias ( $n = 50$ ) usando $Q_p^a$ ,BFGS, Tabela 5.5. . . .	76
A.28 Resultados para 5 instâncias ( $n = 50$ ) usando $Q_p^b$ ,BFGS, Tabela 5.5. . . .	76
A.29 Resultados para 5 instâncias ( $n = 50$ ) usando $Q_p^a$ ,BFGS,Cuts, Tabela 5.5. .	76
A.30 Resultados para 5 instâncias ( $n = 50$ ) usando $Q_p^b$ ,BFGS,Cuts, Tabela 5.5. .	76

# Capítulo 1

## Introdução

Este trabalho tem como objetivo encontrar limites superiores para o problema da mochila quadrática 0-1. Este problema é conhecido na literatura como *Quadratic Knapsack Problem QKP*.

O problema da mochila quadrática é uma generalização do problema da mochila 0-1, *Knapsack Problem KP*. No problema linear, dada uma mochila com capacidade máxima  $c$  e um conjunto de  $n$  itens, de forma que cada item  $i$  possui um peso  $w_i$  e um benefício  $q_i$ , todos os valores  $w_i$ ,  $q_i$  e  $c$  são inteiros. A questão central do problema é selecionar os itens de forma a maximizar os benefícios sem que a capacidade seja excedida. O **KP** pode ser resolvido em tempo pseudo-polinomial usando programação dinâmica.

O problema da mochila é classificado na literatura como *NP-Hard* através da redução pelo problema de clique [1].

### 1.1 Problema da mochila quadrática 0-1 QKP

Seja  $c$  a capacidade da mochila e  $N = \{1, \dots, n\}$  um conjunto de itens candidatos a entrar na mochila. Cada item  $j$  possui um peso positivo  $w_j$ . Seja  $Q = \{q_{ij}\}$  uma matriz  $n \times n$  inteira com todos os elementos não negativos, onde cada  $q_{jj}$  é o benefício alcançado se o item  $j$  for selecionado e  $2q_{ij}$  corresponde ao benefício alcançado se  $i$  e  $j$  são selecionados. O problema consiste em selecionar um subconjunto  $N'$  de  $N$  que forneça o máximo benefício sem que a capacidade da mochila seja excedida. O problema pode ser modelado como:

$$\begin{aligned} \text{(QKP)} \quad & \text{maximize} \quad \sum_{i \in N} \sum_{j \in N} q_{ij} x_i x_j \\ & \text{sujeito a:} \quad \sum_{j \in N} w_j x_j \leq c, \\ & \quad \quad \quad x_j \in \{0, 1\}, \quad j \in N. \end{aligned}$$

O problema da mochila quadrática aparece em uma ampla variedade de campos como biologia, logística, finanças, telecomunicações e teoria de grafos, e tem recebido muita

atenção nas últimas décadas. Uma aplicação do **QKP** na área de telecomunicações é descrita por Witzgall [2]. O problema consiste em selecionar um número de locais para estações de satélite de modo que o tráfego global entre estas estações seja maximizado e um orçamento máximo seja respeitado. Aplicações similares surgem ao considerar-se a localização de aeroportos, estações ferroviárias ou terminais de movimentação de carga [3].

Propomos nesta tese a construção de uma relaxação de programação quadrática paramétrica convexa, *parametric convex quadratic programming*, **CQP**, para encontrar limites superiores para o problema da mochila quadrática. Nesta relaxação mantêm-se informações quadráticas parciais da função objetivo original do **QKP**, perturbando a função objetivo para obter-se um termo quadrático côncavo. Finalmente, a parte não côncava, gerada pela perturbação, é linearizada utilizando um procedimento padrão que redefine o problema para o espaço matricial.

Para otimizar a perturbação da função quadrática apresentaremos um método de pontos interiores primal-dual, *primal-dual interior point method* **IPM**, que a cada iteração busca melhorar o limite superior encontrado para a relaxação do **QKP**. Apresentamos, também, novas desigualdades válidas, que são derivadas das desigualdades de cobertura. Estas desigualdades podem ser aplicadas a outras relaxações de problemas quadráticos binários.

## 1.2 Sequência da Apresentação

Este trabalho está organizado em 6 capítulos. O Capítulo 2 apresenta a revisão bibliográfica sobre as relaxações do problema da mochila quadrática 0-1 e as definições para as conhecidas desigualdades de cobertura. O Capítulo 3 apresenta a metodologia proposta para obtenção de limites superiores, as definições e as propriedades das novas desigualdades que foram inspiradas nas desigualdades de cobertura. No Capítulo 4 apresentamos uma outra estrutura para encontrar limites superiores para o **QKP** utilizando desigualdades válidas e programação disjuntiva. O Capítulo 5 apresenta os experimentos computacionais desenvolvidos e seus respectivos resultados. Finalmente, o Capítulo 6 apresenta as conclusões e sugestões para trabalhos futuros.

# Capítulo 2

## Revisão Bibliográfica

O problema da mochila quadrática 0-1, originalmente apresentado por Gallo et al. [1], é um problema de complexidade não polinomial, conforme provado em [1] através da redução pelo problema da clique que é *NP-Hard*.

O **QKP**, apesar de apresentar uma formulação simples, é difícil de ser resolvido, por este fato, nas últimas décadas diversos trabalhos têm proposto algoritmos *branch-and-bound* para o **QKP**. A principal diferença entre eles é o método utilizado para obter limites superiores para os subproblemas [4–9].

Neste capítulo, apresentaremos algumas das principais relaxações propostas para o **QKP**. Recomendamos as referências [4–8, 10], para uma leitura mais completa sobre as diferentes metodologias utilizadas para a obtenção de limites superiores para o problema.

### 2.1 Limitantes Superiores

Nesta seção, apresentamos algumas técnicas utilizadas para encontrar limites superiores para o **QKP**.

#### 2.1.1 Planos superiores

Os primeiros limites para o **QKP** foram apresentados por Gallo et al. [1]. O problema original é relaxado utilizando planos superiores.

Seja  $g$  uma função linear tal que:

$$g(x) \geq \sum_{i \in N} \sum_{j \in N} q_{ij} x_i x_j \text{ para qualquer } x \text{ viável.}$$

Os planos superiores são da seguinte forma:

$$\sum_{j \in N} \pi_j x_j$$

Assim o **QKP** pode ser relaxado como:

$$\begin{aligned} & \text{maximize} && g(x) = \sum_{j \in N} \pi_j x_j \\ & \text{sujeito a:} && \sum_{j \in N} w_j x_j \leq c, \\ & && x_j \in \{0, 1\}, \quad j \in N. \end{aligned}$$

É importante observar que com esta relaxação linear, o **QKP** tornou-se um problema da mochila linear **KP**. Em [1] os autores apresentam quatro diferentes formas para determinar  $\pi_j$ .

### 2.1.2 Relaxação Lagrangeana

Chaillou, Hansen e Mahieu [4] utilizaram uma relaxação lagrangeana do **QKP** e obtiveram o seguinte problema lagrangeano:

$$\begin{aligned} & \text{maximize} && \sum_{i \in N} \sum_{j \in N} q_{ij} x_i x_j - \lambda \left( \sum_{j \in N} w_j x_j - c \right) \\ & \text{sujeito a:} && x_j \in \{0, 1\}, \quad j \in N. \end{aligned}$$

O problema relaxado pode ser reformulado como um problema de programação quadrática 0-1, *0-1 quadratic programming problem QP*, ao definirmos  $\tilde{q}_{ij} = q_{ij}$  se  $i \neq j$  e  $\tilde{q}_{ij} = q_{ij} - \lambda w_j$  se  $i = j$ .

Então:

$$\begin{aligned} & \text{maximize} && \sum_{i \in N} \sum_{j \in N} \tilde{q}_{ij} x_i x_j + \lambda c \\ & \text{sujeito a:} && x_j \in \{0, 1\}, \quad j \in N. \end{aligned}$$

Picard e Ratliff [11] mostraram que este problema pode ser resolvido em tempo polinomial se a matriz  $\{\tilde{q}_{ij}\}$  não tiver elementos negativos na diagonal principal.

### 2.1.3 Linearização do termo quadrático

Para encontrar limites superiores para o **QKP** Billionnet e Calmels [5] linearizaram o termo quadrático e resolveram o problema linear relaxado.

Utilizaram para isso as variáveis  $y_{ij}$  para substituir o produto  $x_i x_j$ ,  $i < j$ . Temos que  $y_{ij} = 1$  se e somente se  $x_i = 1$  e  $x_j = 1$ .

Podemos formular essa equivalência pelas restrições:



$$y_{ij} \leq x_i, \quad y_{ij} \leq x_j, \quad x_i + x_j \leq 1 + y_{ij}$$

O problema reformulado através da linearização do termo quadrático é dado por:

$$\begin{aligned} &\text{maximize} && \sum_{i,j \in N, i < j} 2q_{ij}y_{ij} + \sum_{j \in N} q_{jj}x_j \\ &\text{sujeito a:} && \sum_{j \in N} w_j x_j \leq c. \\ &&& y_{ij} \leq x_i, \quad i, j \in N, i < j \\ &&& y_{ij} \leq x_j, \quad i, j \in N, i < j \\ &&& x_i + x_j \leq 1 + y_{ij} \quad i, j \in N, i < j \\ &&& x_j, \quad y_{ij} \in \{0, 1\} \quad i, j \in N, i < j \end{aligned}$$

### 2.1.4 Relaxação Lagrangeana e Reformulação

Caprara, Pisinger, Toth [6] também utilizaram o conceito de planos superiores para encontrar limites superiores para o **QKP**. Para isto utilizaram a seguinte formulação para o problema:

$$\begin{aligned} &\text{maximize} && \sum_{j \in N} \sum_{i \in N \setminus \{j\}} q_{ij}y_{ij} + \sum_{j \in N} q_{jj}x_j \\ &\text{sujeito a:} && \sum_{j \in N} w_j x_j \leq c. \\ &&& 0 \leq y_{ij} \leq x_j \leq 1, \quad i, j \in N, j \neq i \\ &&& y_{ij} = y_{ji}, \quad i, j \in N, i < j \\ &&& \sum_{i \in N} w_i y_{ij} \leq (c - w_j)x_j, \quad j \in N \\ &&& x_j, \quad y_{ij} \in \{0, 1\} \quad i, j \in N, i \neq j \end{aligned}$$

Caprara et al. [6] reformulam o problema e utilizam o melhor limite superior encontrado no nó raiz ao longo do algoritmo de *branch-and-bound*. Este limite domina todos os limites de Gallo, Hammer e Simeone [1].

### 2.1.5 Decomposição Lagrangeana

A decomposição lagrangeana é um tipo particular de relaxação lagrangeana, obtida ao duplicar-se as variáveis do **QKP**. Michelon e Veilleux [12] utilizaram decomposição lagrangeana mostrando que os limitantes são melhores que os encontrados ao utilizar-se relaxação lagrangeana. Para isto utilizaram variáveis auxiliares que estão ligadas as variáveis originais do problema. Ao relaxar as restrições de igualdade, é possível dividir o problema em uma série de subproblemas. Podemos escrever o problema de decomposição Lagrangeana da seguinte forma:

$$\begin{aligned}
& \text{maximize} && \sum_{i \in N} \sum_{j \in N} q_{ij} x_i x_j \\
& \text{sujeito a:} && \sum_{j \in N} w_j y_j \leq c, \\
& && x_j = y_j \quad j \in N, \\
& && x_j, y_j \in \{0, 1\}, \quad j \in N.
\end{aligned}$$

Associando multiplicadores de Lagrange a restrição  $x_j = y_j$  temos:

$$\begin{aligned}
& \text{maximize} && \sum_{i \in N} \sum_{j \in N} q_{ij} x_i x_j - \sum_{j \in N} \lambda_j (x_j - y_j) \\
& \text{sujeito a:} && \sum_{j \in N} w_j y_j \leq c, \\
& && x_j, y_j \in \{0, 1\}, \quad j \in N.
\end{aligned}$$

Este problema é decomposto em dois subproblemas:

- Problema quadrático ( $QP$ )

$$\begin{aligned}
(\text{QP}) \quad & \text{maximize} && \sum_{i \in N} \sum_{j \in N} \tilde{q}_{ij} x_i x_j \\
& \text{sujeito a:} && x_j \in \{0, 1\}, \quad j \in N.
\end{aligned}$$

- Problema da mochila ( $KP$ )

$$\begin{aligned}
(\text{KP}) \quad & \text{maximize} && \sum_{i \in N} \lambda_j y_j \\
& \text{sujeito a:} && \sum_{j \in N} w_j y_j \leq c, \\
& && y_j \in \{0, 1\}, \quad j \in N.
\end{aligned}$$

Os limites superiores e inferiores obtidos pela decomposição lagrangeana são mais apertados, isto permite fixar mais variáveis e reduzir o número de nós durante o algoritmo de *Branch-and-Bound*. Assim Michelon e Veilleux [12] mostram que para o **QKP** a decomposição lagrangeana é mais eficiente que a relaxação lagrangeana.

## 2.1.6 Programação Semidefinida

Helmberg, Rendl e Weismantel [8, 10] propuseram limites superiores para o **QKP** baseados em programação semidefinida. Estes limites são válidos para uma versão mais geral do problema em que a matriz  $Q$  pode conter elementos negativos.

Antes de apresentar a reformulação do problema utilizando programação semidefinida introduzimos algumas notações e definições com o objetivo de lembrar alguns conceitos de álgebra linear e auxiliar na leitura deste texto.

Dadas duas matrizes reais  $W_{n \times n}$  e  $Y_{n \times n}$ , o produto interno entre  $W$  e  $Y$  é definido como:

$$\langle W, Y \rangle = \text{traço}(W^T Y) = \sum_{i=1}^n \sum_{j=1}^n W_{ij} Y_{ij}.$$

Neste texto denotamos por  $\text{diag}(W)$  o vetor que contém os elementos da diagonal da matriz  $W$ .

**Definição 2.1.1.** *Seja  $A_{n \times n}$  uma matriz simétrica. Se  $x^T A x \geq 0 \ \forall x$  então dizemos que  $A$  é semidefinida positiva, em símbolos  $A \succeq 0$ .*

Seja  $A \in \mathbb{S}^n$ , então  $\text{svec}(A)$  é um vetor definido como :

$$\text{svec}(A) := (a_{11}, \dots, a_{n1}, a_{22}, \dots, a_{n2}, \dots, a_{nn})^T \in \mathbb{R}^{n(n+1)/2}.$$

O operador  $\text{sMat}$  é o inverso de  $\text{svec}$ , assim  $\text{sMat}(\text{svec}(A)) = A$ .

Denotamos por  $\lambda_{\min}(A)$ , o menor autovalor de  $A$  e  $\lambda_i(A)$  é o  $i^{\text{th}}$  maior autovalor de  $A$ .

**Definição 2.1.2.** *Seja  $M$  uma matriz em blocos definida como:*

$$M = \begin{pmatrix} P & L \\ R & S \end{pmatrix}. \quad (2.1)$$

*O complemento de Schur de  $M$  é  $S - RP^{-1}L$ . A matriz  $M$  é semidefinida positiva se e somente se  $S - RP^{-1}L \succeq 0$ .*

A função objetivo do problema quadrático da mochila pode ser reformulada como:

$$\sum_{i \in N} \sum_{j \in N} q_{ij} x_i x_j = x^T Q x = \langle Q, X \rangle,$$

onde  $X := x x^T$ , ou seja,  $X$  é a matriz com elementos  $X_{ij} := x_i x_j$ . Desta forma o problema **QKP** pode ser reescrito:

$$\begin{aligned} & \text{maximize} && \langle Q, X \rangle \\ & \text{sujeito a:} && \sum_{j \in N} w_j x_j \leq c, \\ & && X = x x^T \\ & && x_j \in \{0, 1\}, \quad j \in N. \end{aligned} \quad (2.2)$$

É importante observar que na reformulação 2.2 do **QKP** existem dois tipos de não convexidade:  $x_j \in \{0, 1\}$  e  $X = x x^T$ . Desta forma, para encontrar limites superiores para o **QKP** é importante relaxar as duas restrições. A restrição de integralidade (a primeira) pode ser relaxada como  $x_j \in [0, 1]$ .

Para relaxar a restrição  $X = x x^T$  existem duas opções propostas na literatura:

- Primeira opção:

Substituir  $X = xx^T$  por  $X - xx^T \succeq 0$ , ou de forma equivalente utilizando o complemento de Schur.

$$X - xx^T \succeq 0 \iff \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0. \quad (2.3)$$

- Segunda opção:

Substituir  $X = xx^T$  por inequações lineares utilizando a Técnica de Reformulação Linear, *Reformulation Linearization Technique* (RLT). Para isto devemos ainda substituir cada termo não linear,  $x_i x_j$ , por uma nova variável  $X_{ij}$ .

Considerando então que para cada par  $x_i$  e  $x_j$  onde  $i, j \in \{1, \dots, n\}$ . Temos

<p>Para <math>x_i</math></p> $0 \leq x_i \leq 1$ $1 - x_i \geq 0 \leftarrow (\times x_j)$ $x_j - \underbrace{x_i x_j}_{X_{ij}} \geq 0$ $x_j - X_{ij} \geq 0$ <p style="text-align: center;">↓</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto;"><math>X_{ij} \leq x_j</math></div>	<p>Para <math>x_j</math></p> $0 \leq x_j \leq 1$ $1 - x_j \geq 0 \leftarrow (\times x_i)$ $x_i - \underbrace{x_j x_i}_{X_{ij}} \geq 0$ $x_i - X_{ij} \geq 0$ <p style="text-align: center;">↓</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto;"><math>X_{ij} \leq x_i</math></div>
--	--

Temos também:

$(1 - x_i)(1 - x_j) \geq 0$ $1 - x_j - x_i + \underbrace{x_i x_j}_{X_{ij}} \geq 0$ $1 - x_j - x_i + X_{ij} \geq 0 \leftarrow (\times -1)$ $-1 + x_j + x_i - X_{ij} \leq 0$ $-1 + x_j + x_i \leq X_{ij}$ <p style="text-align: center;">↓</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto;"><math>x_j + x_i \leq 1 + X_{ij}</math></div>	$x_i \geq 0, x_j \geq 0 \Rightarrow x_i x_j \geq 0$ <p style="text-align: center;">↓</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 0 auto;"><math>X_{ij} \geq 0</math></div>
---	---

Assim deduzimos as desigualdades RLT:

$$\begin{aligned} X_{ij} &\leq x_i, \\ X_{ij} &\leq x_j, \\ x_i + x_j &\leq 1 + X_{ij}, \\ X_{ij} &\geq 0. \end{aligned} \quad (2.4)$$

As desigualdades RLT foram utilizadas por Billionnet e Calmels [5] e por Caprara,

Pisinger e Toth [6]. Elas são conhecidas na literatura com desigualdades de McCormick [13].

Para cada variável  $x_i$   $i \in \{1, \dots, n\}$  consideramos também o limite da restrição  $x_i \geq 0$  e a restrição de capacidade

$$\sum_{j \in N} w_j x_j \leq c,$$

Multiplicando ambos os lados por  $x_i$ :

$$\begin{aligned} \sum_{i \in N} w_j x_j &\leq c && \leftarrow (\times x_i) \\ \sum_{i \in N} w_j \underbrace{x_j x_i} &\leq c x_i && (2.5) \\ \sum_{i \in N} w_j X_{ij} &\leq c x_i \end{aligned}$$

Considerando que todas as variáveis do problema são binárias, temos

$$X_{ii} := x_i x_i = x_i \forall i \in \{1, \dots, n\}$$

que pode ser reescrito como:

$$\text{diag}(X) = x.$$

Consequentemente temos que

$$X_{ii} \leq 1. \quad (2.6)$$

Aplicando o complemento de Schur a matriz

$$\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0 \iff X - x x^T \succeq 0. \quad (2.7)$$

Finalmente, substituindo  $x = \text{diag}(X)$  na equação 2.7

$$\begin{pmatrix} 1 & \text{diag}(X)^T \\ \text{diag}(X) & X \end{pmatrix} \succeq 0 \iff X - \text{diag}(X) \text{diag}(X)^T \succeq 0. \quad (2.8)$$

e na restrição de capacidade substituímos  $x_i = X_{ii}$ .

$$\sum_{j \in N} w_j X_{ij} \leq c x_i \iff \sum_{j \in N} w_j X_{ij} \leq c X_{ii} \iff \sum_{j \in N} w_j X_{ij} - c X_{ii} \leq 0.$$

Assim obtêm-se a relaxação proposta por Helmberg, Rendl e Weismantel:

$$\begin{aligned}
(HRW) \quad & \text{maximize} \quad \langle Q, X \rangle \\
& \text{sujeito a:} \quad \sum_{j \in N} w_j X_{ij} - X_{ii} c \leq 0, \quad i \in N, \\
& \quad \quad \quad X - \text{diag}(X) \text{diag}(X)^T \succeq 0,
\end{aligned} \tag{2.9}$$

Observe que nesta formulação a restrição  $X - \text{diag}(X) \text{diag}(X)^T \succeq 0$  garante que  $0 \leq x_i \leq 1$ .

Esta formulação proposta por [8] e [10] tem como base a formulação 2.2 e as relaxações apresentadas acima com exceção das restrições RLT que não foram utilizadas. Em [14] a partir de resultados computacionais mostrou-se que esta relaxação *SDP* apresentou os melhores limites superiores para o **QKP** nas instâncias abordadas.

## 2.2 Politopo da Mochila e Desigualdades de Cobertura

Nesta seção apresentamos o conceito do politopo da mochila, *knapsack polytope* **KPol**, e as desigualdades de cobertura, *cover inequalities* **CI**. As desigualdades de cobertura foram originalmente apresentadas em [15, 16]; mais detalhes podem ser encontrados em [17]. Estas desigualdades podem ser utilizadas em problemas de otimização com variáveis binárias.

O politopo da mochila é o fecho convexo dos pontos viáveis do problema da mochila linear **KP**:

$$\mathbf{KF} := \{x \in \{0, 1\}^n : w^T x \leq c\}.$$

### 2.2.1 Politopo da Mochila 0-1

Definimos o politopo da mochila 0-1 como:

$$\mathbf{KPol} := \text{conv}(\mathbf{KF}) = \text{conv}\{x \in \{0, 1\}^n : w^T x \leq c\}.$$

A dimensão do politopo da mochila 0-1 é

$$\dim(\mathbf{KPol}) = n,$$

e o **KPol** é um sistema independente, isto é,

$$x \in \mathbf{KPol}, y \in \{0, 1\}^n, y \leq x \implies y \in \mathbf{KPol}.$$

## 2.2.2 Desigualdades de Cobertura

O subconjunto  $C \subseteq N$  é uma cobertura se satisfaz

$$\sum_{j \in C} w_j > c \quad (2.10)$$

As desigualdades de cobertura **CI** são definidas como

$$\sum_{j \in C} x_j \leq |C| - 1. \quad (2.11)$$

A **CI** é chamada mínima se nenhum subconjunto próprio de  $C$  também for uma cobertura.

## 2.2.3 Desigualdades de Cobertura Estendida

Seja  $w^* := \max_{j \in C} w_j$ , definimos a extensão de  $C$  como

$$E(C) := C \cup \{j \in N \setminus C : w_j \geq w^*\}.$$

Assim a desigualdades de cobertura estendida, *extended cover inequality ECI*, é

$$\sum_{j \in E(C)} x_j \leq |C| - 1.$$

## 2.2.4 Procedimento de *lifting* para Desigualdades de Cobertura

No procedimento de *lifting* para desigualdades de cobertura, *lift cover inequalities LCI*, dada uma cobertura mínima  $C$ , existe pelo menos uma faceta definida por **LCI** da forma:

$$\sum_{j \in C} x_j + \sum_{j \in N \setminus C} \alpha_j x_j \leq |C| - 1, \quad (2.12)$$

onde  $\alpha_j \geq 0$  para todo  $j \in N \setminus C$ , além disso cada **LCI** domina **ECI**.

As desigualdades de cobertura foram intensamente discutidas em [15–20]. Detalhes sobre a complexidade computacional dos **LCI** podem ser encontrados em [21, 22]. O Algoritmo 1, originalmente apresentado em [23], é um procedimento para encontrar **LCI** de uma dada cobertura mínima  $C$ .

---

**Algoritmo 1:** Procedimento para encontrar LCI

---

- 1 Ordene os elementos  $w_i$  de forma crescente onde  $i \in N \setminus C$ , defina  $\{i_1, i_2, \dots, i_r\}$
- 2 Para: t=1 to r

$$\begin{aligned} \zeta_t = \max & \quad \sum_{j=1}^{t-1} \alpha_{i_j} x_{i_j} + \sum_{i \in C} x_i \\ \text{sujeito a:} & \quad \sum_{j=1}^{t-1} w_{i_j} x_{i_j} + \sum_{i \in C} w_i x_i \leq c - w_{i_t} \\ & \quad x \in \{0, 1\}^{|C|+t-1}. \end{aligned} \quad (2.13)$$

Faça  $\alpha_{i_t} = |C| - 1 - \zeta_t$

Fim

---



# Capítulo 3

## Método Proposto

O problema da mochila quadrática 0-1, introduzido por Gallo et al. [1], consiste em selecionar um subconjunto de itens de forma a maximizar o benefício sem que a capacidade da mochila seja excedida. Desta forma o problema é formulado como:

$$\begin{aligned} (\mathbf{QKP}) \quad & \text{maximize} \quad \sum_{i \in N} \sum_{j \in N} q_{ij} x_i x_j \\ & \text{sujeito a:} \quad \sum_{j \in N} w_j x_j \leq c, \\ & \quad \quad \quad x_j \in \{0, 1\}, \quad j \in N \end{aligned} \tag{3.1}$$

onde,

- $N = \{1, \dots, n\}$  - conjunto de itens;
- $w_j$  - peso do item  $j$ ;
- $Q = \{q_{ij}\}$  - matriz  $n \times n$  com benefício dos itens;
- $q_{jj}$  - benefício alcançado se o item  $j$  é selecionado;
- $2q_{ij}$  - benefício alcançado se os itens  $i$  e  $j$  são selecionados;
- $x_j$  - variável binária. Indica se o item  $j$  é selecionado para entrar na mochila;
- $c$  - capacidade da mochila  $c \in \mathbb{N}^+$ .

Neste capítulo apresentaremos uma relaxação para o **QKP** utilizando programação quadrática paramétrica convexa, **CQP**. Nesta relaxação a função objetivo do **QKP** é perturbada, e obtem-se um termo quadrático côncavo mantendo as informações quadráticas da função objetivo. A parte não côncava gerada pela perturbação é linearizada.

Para otimizar a perturbação da função quadrática apresentaremos um método de pontos interiores primal-dual, **IPM**, que a cada iteração busca melhorar o limite superior encontrado para a relaxação do **QKP**. Apresentamos também novas desigualdades válidas derivadas das desigualdades de cobertura. Estas desigualdades podem ser aplicadas a outras relaxações de problemas quadráticos binários.

### 3.1 Relaxação convexa quadrática paramétrica

Para construir a relaxação convexa do **QKP**, considere a seguinte reformulação do problema no espaço de matrizes simétricas definido por  $X := xx^T$ .

$$\begin{aligned}
 p_{\text{QKP}^{\text{lifted}}}^* &:= \text{maximize} && \text{trace}(QX) \\
 \text{(QKP}^{\text{lifted}}) &&& \text{sujeito a: } w^T x \leq c \\
 &&& X = xx^T \\
 &&& x \in \{0, 1\}^n.
 \end{aligned} \tag{3.2}$$

Consideramos a relaxação linear inicial do **QKP**,

$$\begin{aligned}
 \text{(LPR)} &&& \text{maximize} && \text{trace}(QX) \\
 &&& \text{sujeito a: } && (x, X) \in \mathcal{P},
 \end{aligned} \tag{3.3}$$

onde  $\mathcal{P} \subset [0, 1]^n \times \mathbb{S}^n$  é um poliedro tal que

$$\{(x, X) : w^T x \leq c, X = xx^T, x \in \{0, 1\}^n\} \subset \mathcal{P}.$$

#### 3.1.1 Perturbação da função objetivo

Nesta seção propomos uma relaxação convexa quadrática utilizando o conjunto viável definido em (3.3), mas mantendo uma versão perturbada côncava da função objetivo quadrática do **QKP**, e linearizando apenas a parte restante não côncava derivada da perturbação. Especificamente escolhemos  $Q_p \in \mathbb{S}^n$  tal que

$$Q - Q_p \preceq 0, \tag{3.4}$$

e seja

$$\begin{aligned}
 x^T Q x &= x^T (Q - Q_p) x + x^T Q_p x = x^T (Q - Q_p) x + \text{trace}(Q_p x x^T) \\
 &= x^T (Q - Q_p) x + \text{trace}(Q_p X).
 \end{aligned}$$

Definimos desta forma a relaxação convexa quadrática paramétrica do **QKP**:

$$\begin{aligned}
 \text{(CQP}_{Q_p}) &&& p_{\text{CQP}}^*(Q_p) := \text{maximize} && x^T (Q - Q_p) x + \text{trace}(Q_p X) \\
 &&& \text{sujeito a: } && (x, X) \in \mathcal{P},
 \end{aligned} \tag{3.5}$$

#### 3.1.2 Otimização do problema paramétrico sobre o parâmetro $Q_p$

O limite superior  $p_{\text{CQP}}^*(Q_p)$  no problema convexo quadrático (3.5) depende da perturbação viável  $Q_p$  da matriz Hessiana  $Q$ . Para melhorar o limite superior conside-

ramos o problema paramétrico

$$param_{\text{QKP}}^* := \min_{Q_n(Q_p)=2(Q-Q_p)\preceq 0} p_{\text{CQP}}^*(Q_p). \quad (3.6)$$

Resolvemos este problema com o algoritmo primal-dual de pontos interiores, **IPM**, e descrevemos nesta seção como encontrar a direção de busca a cada iteração do algoritmo. Iniciamos com a minimização da função barreira logarítmica. Usamos a *função barreira*,  $B_\mu(Q_p, Z)$  com *parâmetro*  $\mu > 0$ , para obter o problema

$$\begin{aligned} \text{minimize} \quad & B_\mu(Q_p, Z) := p^*(Q_p) - \mu \log \det Z \\ \text{sujeito a:} \quad & Q - Q_p + Z = 0 \quad (:\Lambda) \\ & Z \succ 0, \end{aligned} \quad (3.7)$$

onde  $\Lambda \in \mathbb{S}^n$  denota a matriz de variáveis duais. A função Lagrangeana é

$$L_\mu(Q_p, Z, \Lambda) := p_{\text{CQP}}^*(Q_p) - \mu \log \det Z + \text{trace}((Q - Q_p + Z)\Lambda).$$

Note que a função objetivo para  $p_{\text{CQP}}^*(Q_p)$  é linear em  $Q_p$ , isto é, esta função é o máximo da função linear sobre o ponto viável  $x, X$ . Além disto é uma função convexa. Pela análise de sensibilidade padrão, por exemplo [24, Corolário 3.4.2],[25], [26, Teorema 1], sabemos que se a solução ótima  $x, X$  for única, então o gradiente é obtido pela diferenciação do Lagrangeano. Como  $Q_p$  aparece apenas da função objetivo em (3.5) e

$$\frac{1}{2}x^T Q_n(Q_p)x + \text{trace}(Q_p X) = \frac{1}{2}x^T (2(Q - Q_p))x + \text{trace}(Q_p X) = x^T Q x + \text{trace}(Q_p(-xx^T + X)),$$

tomamos a derivada direcional em  $Q_p$  na direção  $\Delta Q_p$

$$D(p_{\text{CQP}}^*(Q_p); \Delta Q_p) = \max_{\text{óptimal } x, X} \text{trace}((X - xx^T)\Delta Q_p).$$

No caso de unicidade da solução ótima  $x = x(Q_p), X = X(Q_p)$ , temos o gradiente

$$\nabla p_{\text{CQP}}^*(Q_p) = X - xx^T. \quad (3.8)$$

O gradiente da função barreira, é então:

$$\nabla B_\mu(Q_p) = (X - xx^T) - \mu Z^{-1}.$$

As condições de otimalidade para (3.7) são obtidas pela diferenciação do Lagrangeano

$L_\mu$  em relação a  $Q_p, \Lambda, Z$ , respectivamente,

$$\begin{aligned} \frac{\partial}{\partial Q_p} : \quad \nabla p_{\text{CQP}}^*(Q_p) - \Lambda &= 0, \\ \frac{\partial}{\partial \Lambda} : \quad Q - Q_p + Z &= 0, \\ \frac{\partial}{\partial Z} : \quad -\mu Z^{-1} + \Lambda &= 0, \quad (\text{ou}) \quad Z\Lambda - \mu I = 0. \end{aligned} \quad (3.9)$$

Isto resulta no seguinte sistema não linear

$$G_\mu(Q_p, \Lambda, Z) = \begin{pmatrix} \nabla p_{\text{CQP}}^*(Q_p) - \Lambda \\ Q - Q_p + Z \\ Z\Lambda - \mu I \end{pmatrix} = 0, \quad Z, \Lambda \succ 0. \quad (3.10)$$

Para o cálculo da Hessiana  $p_{\text{CQP}}^*$ , utilizamos a aproximação BFGS, como se fosse duas vezes diferenciável e atualizamos a cada iteração, (mais detalhes em [27]). Denotamos a aproximação de  $\nabla_{\text{BFGS}}^2 p^*(Q_p)$  por  $B$ , e iniciamos o algoritmo com a aproximação  $B_0 = I$ .

É importante observar que se  $Q_p^k, Q_p^{k+1}$  são duas iterações sucessivas com gradiente  $\nabla p_{\text{CQP}}^*(Q_p^k), \nabla p_{\text{CQP}}^*(Q_p^{k+1})$ , respectivamente, e com aproximação corrente da Hessiana dada por  $B_k \in \mathbb{S}^{n(n+1)/2}$ , então fixamos

$$Y_k := \nabla p_{\text{CQP}}^*(Q_p^{k+1}) - \nabla p_{\text{CQP}}^*(Q_p^k), \quad S_k := Q_p^{k+1} - Q_p^k,$$

e,

$$v := \langle Y_k, S_k \rangle, \quad \omega := \langle \text{svec}(S_k), B_k \text{svec}(S_k) \rangle.$$

Note que a condição de curvatura  $v > 0$  deve ser verificada. Finalmente, calculamos a atualização da Hessiana

$$B_{k+1} := B_k + \frac{1}{v} (\text{svec}(Y_k) \text{svec}(Y_k)^T) - \frac{1}{\omega} (B_k \text{svec}(S_k) \text{svec}(S_k)^T B_k).$$

A equação sobredeterminada para a direção de busca é

$$G'_\mu(Q_p, \Lambda, Z) \begin{pmatrix} \Delta Q_p \\ \Delta \Lambda \\ \Delta Z \end{pmatrix} = -G_\mu(Q_p, \Lambda, Z), \quad (3.11)$$

onde

$$G'_\mu(Q_p, \Lambda, Z) = \begin{pmatrix} \nabla p^*(Q_p) - \Lambda \\ Q - Q_p + Z \\ Z\Lambda - \mu I \end{pmatrix} =: \begin{pmatrix} R_d \\ R_p \\ R_c \end{pmatrix}. \quad (3.12)$$

Se  $B$  é a estimativa corrente da Hessiana, então o sistema pode ser reescrito como:

$$\begin{cases} \text{sMat}(B \text{svec}(\Delta Q_p)) - \Delta\Lambda = -R_d, \\ -\Delta Q_p + \Delta Z = -R_p, \\ Z\Delta\Lambda + \Delta Z\Lambda = -R_c. \end{cases}$$

Podemos substituir  $\Delta\Lambda$  e  $\Delta Z$  na terceira equação do sistema. Note que, como o sistema é sobredeterminado, esta substituição muda a solução de mínimos quadrados. Ainda assim, a eliminação nos fornece um sistema simplificado e, portanto, aplicamos isto ao algoritmo, usando as duas equações abaixo para eliminar a substituição reversa

$$\Delta\Lambda = \text{sMat}(B \text{svec}(\Delta Q_p)) + R_d, \quad \Delta Z = -R_p + \Delta Q_p. \quad (3.13)$$

Desta forma, temos que resolver uma equação única e o sistema pode ser escrito como:

$$Z \text{sMat}(B \text{svec}(\Delta Q_p)) + (\Delta Q_p)\Lambda = -R_c - ZR_d + R_p\Lambda.$$

Enfatizamos que, para calcular a direção de busca a cada iteração do algoritmo, **IPM**, proposto, é necessário atualizar os resíduos definidos na equação (3.12) e portanto é necessário encontrar a solução ótima  $x = x(Q_p)$ ,  $X = X(Q_p)$  da relaxação convexa quadrática **CQP** <sub>$Q_p$</sub> , para a perturbação corrente  $Q_p$ . A cada iteração do **IPM** o problema **CQP** <sub>$Q_p$</sub>  é resolvido para uma nova perturbação  $Q_p$ .

Além disso note que a cada iteração do **IPM**, temos:  $Z \succ 0$  e  $Q - Q_p \prec 0$ . Então o problema **CQP** <sub>$Q_p$</sub>  maximiza uma função côncava quadrática, sujeito a restrições lineares e, portanto, tem uma solução ótima única (veja ex.: [28]). O resultado garante que o gradiente em (3.8) foi bem definido.

O Algoritmo 2, apresenta em detalhes uma iteração do **IPM**. O algoritmo é parte da estrutura completa usada para gerar os limites para o **QKP** a ser descrito no Capítulo 5.

---

**Algoritmo 2:** Atualização da perturbação  $Q_p$ 

---

Entrada:  $k, Q_p^k, Z^k, \Lambda^k, x(Q_p^k), X(Q_p^k), \nabla p_{\text{CQP}}^*(Q_p^k), B_k, \mu^k, \tau_\alpha := 0.95,$   
 $\tau_\mu := 0.9.$

Calcule os resíduos:

$$\begin{pmatrix} R_d \\ R_p \\ R_c \end{pmatrix} := \begin{pmatrix} \nabla p_{\text{CQP}}^*(Q_p^k) - \Lambda^k \\ Q - Q_p^k + Z^k \\ Z^k \Lambda^k - \mu^k I \end{pmatrix}.$$

Resolva o sistema linear para  $\Delta Q_p$ :

$$Z^k \text{sMat}(B_k \text{svec}(\Delta Q_p)) + (\Delta Q_p) \Lambda^k = -R_c - Z^k R_d + R_p \Lambda^k.$$

Fixar:

$$\begin{aligned} \Delta \Lambda &:= \text{sMat}(B_k \text{svec}(\Delta Q_p)) + R_d, \\ \Delta Z &:= -R_p + \Delta Q_p. \end{aligned}$$

Atualize  $Q_p, Z$  e  $\Lambda$ :

$$\begin{aligned} Q_p^{k+1} &:= Q_p^k + \hat{\alpha}_p \Delta Q_p, \\ Z^{k+1} &:= Z_p^k + \hat{\alpha}_p \Delta Z, \\ \Lambda^{k+1} &:= \Lambda^k + \hat{\alpha}_d \Delta \Lambda, \end{aligned}$$

onde

$$\begin{aligned} \hat{\alpha}_p &:= \tau_\alpha \times \min\{1, \text{argmax}_{\alpha_p} \{Z_p^k + \alpha_p \Delta Z \succeq 0\}\}, \\ \hat{\alpha}_d &:= \tau_\alpha \times \min\{1, \text{argmax}_{\alpha_d} \{\Lambda^k + \alpha_d \Delta \Lambda \succeq 0\}\}. \end{aligned}$$

Obtenha a solução ótima  $x(Q_p^{k+1}), X(Q_p^{k+1})$  da relaxação **CQP**  $Q_p$ , onde  $Q_p := Q_p^{k+1}$ .

Atualize o gradiente de  $p_{\text{CQP}}^*$ :

$$\nabla p_{\text{CQP}}^*(Q_p^{k+1}) := X(Q_p^{k+1}) - x(Q_p^{k+1})x(Q_p^{k+1})^T.$$

Atualize a aproximação da Hessiana de  $p_{\text{CQP}}^*$ :

$$\begin{aligned} Y_k &:= \nabla p_{\text{CQP}}^*(Q_p^{k+1}) - \nabla p_{\text{CQP}}^*(Q_p^k), \\ S_k &:= Q_p^{k+1} - Q_p^k, \\ v &:= \langle Y_k, S_k \rangle, \\ \omega &:= \langle \text{svec}(S_k), B_k \text{svec}(S_k) \rangle, \\ B_{k+1} &:= B_k + \frac{1}{v} (\text{svec}(Y_k) \text{svec}(Y_k)^T) - \frac{1}{\omega} (B_k \text{svec}(S_k) \text{svec}(S_k)^T B_k). \end{aligned}$$

Atualize  $\mu$ :

$$\mu^{k+1} := \tau_\mu \frac{\text{trace}(Z^{k+1} \Lambda^{k+1})}{n}.$$

Saída:  $Q_p^{k+1}, Z^{k+1}, \Lambda^{k+1}, x(Q_p^{k+1}), X(Q_p^{k+1}), \nabla p_{\text{CQP}}^*(Q_p^{k+1}), B_{k+1}, \mu^{k+1}.$

## 3.2 A relaxação quadrática paramétrica SDP

Para obter limitantes mais apertados uma abordagem que poderia parecer promissora seria a adição de restrições **SDP**,  $X - xx^T \succeq 0$ , à relaxação quadrática paramétrica e desta forma resolver uma relaxação **SDP** convexa quadrática usando o **IPM**. Entretanto, mostraremos nesta seção que a relaxação **SDP** convexa quadrática não gera limites melhores que a relaxação **SDP** linear obtida ao considerar  $Q_p$  igual a  $Q$ .

Considere a reformulação **QKP** *lifted* em (3.2), do problema da mochila quadrática 0-1, e sua relaxação **SDP** dada por

$$\begin{aligned} (\text{LSDP}) \quad p_{\text{LSDP}}^* &:= \sup \text{trace}(QX) \\ &\text{s.a. } (x, X) \in \mathcal{F} \\ &\quad X - xx^T \succeq 0, \end{aligned} \quad (3.14)$$

onde  $\mathcal{F}$  é qualquer relaxação do conjunto viável do **QKP** *lifted*.

Considere a relaxação paramétrica **SDP** do **QKP** *lifted* dada por:

$$\begin{aligned} (\text{QSDP}_{Q_p}) \quad p_{\text{QSDP}_{Q_p}}^* &:= \sup x^T(Q - Q_p)x + \text{trace}(Q_p X) \\ &\text{s.a. } (x, X) \in \mathcal{F} \\ &\quad X - xx^T \succeq 0, \end{aligned} \quad (3.15)$$

onde  $Q - Q_p \preceq 0$ .

**Teorema 3.2.1.** *Seja  $\mathcal{F}$  qualquer subconjunto de  $\mathbb{R}^n \times \mathbb{S}^n$ . Para qualquer escolha da matriz  $Q_p$  que satisfaça  $Q - Q_p \preceq 0$ , temos*

$$p_{\text{QSDP}_{Q_p}}^* \geq p_{\text{LSDP}}^*. \quad (3.16)$$

Além disso,  $\inf\{p_{\text{QSDP}_{Q_p}}^* : Q - Q_p \preceq 0\} = p_{\text{LSDP}}^*$ .

*Demonstração.* Seja  $(\tilde{x}, \tilde{X})$  uma solução viável para **LSDP**. Temos

$$p_{\text{QSDP}_{Q_p}}^* \geq \tilde{x}^T(Q - Q_p)\tilde{x} + \text{trace}(Q_p \tilde{X}) \quad (3.17)$$

$$\begin{aligned} &= \text{trace}((Q - Q_p)(\tilde{x}\tilde{x}^T - \tilde{X})) + \text{trace}((Q - Q_p)\tilde{X}) \\ &\quad + \text{trace}(Q_p \tilde{X}) \end{aligned} \quad (3.18)$$

$$= \text{trace}((Q - Q_p)(\tilde{x}\tilde{x}^T - \tilde{X})) + \text{trace}(Q\tilde{X}) \quad (3.19)$$

$$\geq \text{trace}(Q\tilde{X}). \quad (3.20)$$

A desigualdade (3.17) é válida pois  $(\tilde{x}, \tilde{X})$  também é uma solução viável para **QSDP** $_{Q_p}$ . A desigualdade (3.20) é válida porque  $Q - Q_p$  e  $\tilde{x}\tilde{x}^T - \tilde{X}$  são semidefinidas negativas. Dado que  $p_{\text{QSDP}_{Q_p}}^*$  é um limite superior para o valor da função objetivo do

**LSDP** em qualquer solução viável, podemos concluir que  $p_{\text{QSDP}_{Q_p}}^* \geq p_{\text{LSDP}}^*$ . Notadamente  $Q_p = Q$  e satisfaz  $Q - Q_p = 0 \preceq 0$  e **LSDP** é o mesmo que **QSDP** para esta escolha de  $Q_p$ . Assim  $\inf\{p_{\text{QSDP}_{Q_p}}^* : Q - Q_p \preceq 0\} = p_{\text{LSDP}}^*$ .  $\square$

Note que, pelo Teorema 3.2.1 não é necessário que a relaxação de  $\mathcal{F}$  seja convexa nem que haja qualquer relação com a região viável do **QKP**. Além disso, em particular, para alguma escolha de  $Q_p$ , podemos ter  $p_{\text{QSDP}_{Q_p}}^* = +\infty$  com  $p_{\text{LSDP}}^* = +\infty$  ou não.

### 3.3 Desigualdades válidas

Nesta seção apresentamos novas desigualdades válidas que fortalecem a relaxação do **QKP** no espaço definido por  $X := xx^T$ . Denotaremos por **CRel** qualquer relaxação convexa do **QKP** neste espaço, onde a equação  $X = xx^T$  é relaxada de alguma forma utilizando restrições convexas, i.e, qualquer relaxação convexa de **QKP** *lifted*.

Inicialmente, se a desigualdade

$$\tau^T x \leq \beta \tag{3.21}$$

é válida para o **QKP**, onde  $\tau \in \mathbb{Z}_+^n$  e  $\beta \in \mathbb{Z}_+$ , então, como  $x$  é não negativo e  $X := xx^T$ , a desigualdade

$$(x \ X) \begin{pmatrix} -\beta \\ \tau \end{pmatrix} \leq 0 \tag{3.22}$$

é válida para **QKP** *lifted*. Neste caso dizemos que (3.22) é uma desigualdade válida para **QKP** *lifted* derivada de (3.21) que, por sua vez, é uma desigualdade válida para o **QKP**.

#### 3.3.1 Adicionando cortes a relaxação

Dada a solução  $(\bar{x}, \bar{X})$  de **CRel**, nosso objetivo inicial é encontrar a desigualdade, válida para **QKP** *lifted*, derivada da desigualdade de cobertura, **CI**, que seja a mais violada por  $(\bar{x}, \bar{X})$ . Uma **CI** é formulada como  $\alpha^T x \leq e^T \alpha - 1$ , onde  $\alpha \in \{0, 1\}^n$  e  $e$  denota o vetor de uns. Então procuramos pela **CI** que maximiza a máxima violação entre a desigualdade  $\bar{Y} \text{cut}(\alpha) \leq 0$ , onde  $\bar{Y} := [\bar{x} \ \bar{X}]$  e

$$\text{cut}(\alpha) = \begin{pmatrix} -e^T \alpha + 1 \\ \alpha \end{pmatrix}.$$

Para obter **CI** resolvemos o seguinte problema da mochila linear, **KP**,

$$v^* := \max_{\alpha} \{e^T \bar{Y} \text{cut}(\alpha) : w^T \alpha \geq c + 1, \alpha \in \{0, 1\}^n\}. \tag{3.23}$$



Seja  $\alpha^*$  a solução de (3.23). Se  $v^* > 0$ , então pelo menos uma das  $n$  desigualdades válidas no conjunto das desigualdades de cobertura, denotado por **SCI**, é violada por  $(\bar{x}, \bar{X})$ .

$$(x \ X) \begin{pmatrix} -e^T \alpha^* + 1 \\ \alpha^* \end{pmatrix} \leq 0. \quad (3.24)$$

Com base no Teorema 3.3.1, para fortalecer o corte (3.24), podemos aplicar o Algoritmo 1 ao **CI** obtido e desta forma obter a desigualdade **LCI**, e finalmente adicionar a desigualdade válida (3.22) derivada de **LCI**, ao **CRel**.

**Teorema 3.3.1.** *A desigualdade válida (3.22) para **QKP** lifted, que é derivada de uma **LCI** válida, domina todas as desigualdades derivadas de **CI** que podem ser reescritas como **LCI**.*

*Demonstração.* Considere a **LCI** (2.12) derivada de **CI** (2.11) para o **QKP**. As desigualdades de cobertura (3.22) derivadas de **CI** e de **LCI** são, respectivamente,

$$\sum_{j \in C} X_{ij} \leq (|C| - 1)x_i, \quad \forall i \in N,$$

e

$$\sum_{j \in C} X_{ij} + \sum_{j \in N \setminus C} \alpha_j X_{ij} \leq (|C| - 1)x_i, \quad \forall i \in N,$$

onde  $\alpha_j \geq 0, \forall j \in N \setminus C$ . Como todo  $X_{ij}$  é não negativo, a segunda desigualdade domina a primeira para todo  $i \in N$ . □

### 3.3.2 Novas desigualdades válidas no espaço lifted

Como discutido anteriormente, após encontrar alguma desigualdade válida para o **QKP** da forma (3.21), podemos adicionar a restrição (3.22) à relaxação **CRel** para encontrar limites melhores. Observe agora que, além de (3.22) podemos também gerar desigualdades válidas fortes no espaço lifted tomando vantagem de  $X := xx^T$  e de  $x$  ser binário. Abaixo explicamos como esta ideia pode ser aplicada para desigualdades de cobertura.

Seja

$$\sum_{j \in C} x_j \leq \beta. \quad (3.25)$$

onde  $C \subset N$  e  $\beta < |C|$ , são desigualdades válidas para **KPol**.

Note que (3.25) pode ser ao mesmo tempo uma desigualdade de cobertura, **CI**, e uma desigualdade de cobertura estendida, **ECI**, ou uma desigualdade **LCI**, onde  $\alpha_j \in \{0, 1\}, \forall j \in N \setminus C$  em (2.12). Além disso dada uma desigualdade **LCI** qualquer, onde

$\alpha_j \in \mathbb{Z}_+$ , para todo  $j \in N \setminus C$ , uma desigualdade válida do tipo (3.25) pode ser construída substituindo cada  $\alpha_j$  com  $\min\{\alpha_j, 1\}$  na **LCI**.

**Definição 3.3.1** (*Desigualdade de cobertura no espaço lifted **CILS**, cover inequality in the lifted space*). Sejam  $C \subset N$  e  $\beta < |C|$  conforme definidos na desigualdade (3.25), e considere  $\beta > 1$ . Definimos

$$\sum_{i,j \in C, i < j} X_{ij} \leq \binom{\beta}{2}. \quad (3.26)$$

como a **CILS** derivado de (3.25).

**Teorema 3.3.2.** *Se a desigualdade (3.25) é válida para **QKP**, então a **CILS** (3.26) é uma desigualdade válida para **QKP** lifted.*

*Demonstração.* Considere (3.25), podemos concluir que no máximo  $\binom{\beta}{2}$  produtos de variáveis  $x_i x_j$ , onde  $i, j \in C$  podem ser iguais a 1. Assim, como  $X_{ij} := x_i x_j$ , o teorema está provado.  $\square$

Na literatura o problema no qual dado um ponto fracionário  $\bar{x}$  deve-se encontrar desigualdades violadas por  $\bar{x}$  ou mostrar que não existe nenhuma desigualdade violada por este ponto é conhecido como Problema de Separação, *separation problem*. Nesta seção apresentaremos problemas de separação que serão utilizados para encontrar desigualdades válidas para o **QKP**.

Dada uma solução  $(\bar{x}, \bar{X})$  de **CRel**, o seguinte problema de programação inteira mista, *mixed-integer linear program (MILP)*, é o problema de separação, que busca por **CILS** violadas pelo ponto  $\bar{X}$ .

$$\begin{aligned} z^* := \text{maximize}_{\alpha, \psi, K} \quad & \text{trace}(\bar{X}K) - \psi(\psi - 1), & (\text{MILP}_1) \\ \text{sujeito a:} \quad & w'\alpha \geq c + 1, \\ & \psi = e'\alpha - 1, \\ & K_{i,i} = 0, & i = 1, \dots, n, \\ & K_{i,j} \leq \alpha_i, & i, j = 1, \dots, n, \quad i < j, \\ & K_{i,j} \leq \alpha_j, & i, j = 1, \dots, n, \quad i < j, \\ & K_{i,j} \geq 0, & i, j = 1, \dots, n, \quad i < j, \\ & K_{i,j} \geq \alpha_i + \alpha_j - 1, & i, j = 1, \dots, n, \quad i < j, \\ & \alpha \in \{0, 1\}^n, \quad \psi \in \mathbb{R}, \quad K \in \mathbb{S}^n. \end{aligned}$$

Se  $\alpha^*, \psi^*, K^*$  são a solução do **MILP**<sub>1</sub>, com  $z^* > 0$ , o **CILS** dado por

$$\text{trace}(K^* X) \leq \psi^*(\psi^* - 1)$$

é violado por  $\bar{X}$ . O vetor binário  $\alpha^*$  define o **CI** do qual o corte é derivado. O **CI** é dado

especificamente por

$$\alpha^{*T}x \leq e^T\alpha^* - 1$$

e  $\psi^*(\psi^* - 1)$  determina o lado direito do **CILS**.

**Teorema 3.3.3.** *A desigualdade válida **CILS** para **QKP** lifted, que é derivada de um **LCI** válido da forma (3.25), domina qualquer **CILS** derivado de uma **CI** que pode ser lifted para um **LCI**.*

*Demonstração.* Como  $X$  é não negativo, é fácil verificar que se  $X$  satisfaz um **CILS** derivado de um **LCI**,  $X$  também satisfaz qualquer **CILS** derivado de um **CI** que pode ser lifted para um **LCI**.  $\square$

Qualquer solução viável do **MILP**<sub>1</sub>, tal que,  $\text{trace}(\bar{X}K) > \psi(\psi - 1)$ , gera uma desigualdade válida para o **QKP** lifted que é violada por  $\bar{X}$ . Assim não é necessário encontrar a solução ótima do **MILP**<sub>1</sub> para gerar um corte. Além disso, para gerar cortes distintos, podemos resolver **MILP**<sub>1</sub> várias vezes (não necessariamente até a otimalidade), cada vez adicionando ao problema o seguinte corte, que tem como objetivo, evitar cortes gerados anteriormente:

$$\sum_{i \in N} \bar{\alpha}_i(1 - \alpha_i) \geq 1, \quad (3.27)$$

onde  $\bar{\alpha}$  é o valor da variável  $\alpha$  na solução do **MILP**<sub>1</sub>, quando gerava o corte anterior.

Note que, se  $\alpha^*, \psi^*, K^*$  resolvem o problema **MILP**<sub>1</sub>, então  $\alpha^{*T}x \leq e^T\alpha^* - 1$  é um **CI** válido para o **QKP**, entretanto pode não ser uma cobertura mínima. Visando gerar cortes válidos fortes, baseado no Teorema 3.3.3, poderíamos adicionar a função objetivo do **MILP**<sub>1</sub>, o termo  $-\delta e^T\alpha$ , para algum  $\delta > 0$ . Desta forma a função objetivo favoreceria coberturas mínimas, que podem ser reescritas como **LCI** utilizando o Algoritmo 1, e finalmente, podem gerar o **CILS**. Enfatizamos que o **CILS** derivado do **CI** é violado por um dado  $\bar{X}$ , então claramente, o **CILS** derivado do **LCI** será também violado por  $\bar{X}$ .

Além de definir uma desigualdade de cobertura no espaço lifted considerando todos os possíveis pares de índices em  $C$ , podemos definir um conjunto de desigualdades de cobertura no espaço lifted, considerando que o lado esquerdo de cada desigualdade definida não devesse ser maior que  $\beta/2$ . A ideia será melhor apresentada abaixo.

**Definição 3.3.2** (Conjunto de desigualdades de cobertura no espaço lifted, *Set of cover inequalities in the lifted space, SCILS*). Seja  $C \subset N$  e  $\beta < |C|$  como na desigualdade (3.25). Seja

- $C_s := \{(i_1, j_1), \dots, (i_p, j_p)\}$  uma partição de  $C$ , se  $|C|$  é par.
- $C_s := \{(i_1, j_1), \dots, (i_p, j_p)\}$  uma partição de  $C \setminus \{i_0\}$  para cada  $i_0 \in C$ , se  $|C|$  é ímpar e  $\beta$  é ímpar.

- $C_s := \{(i_0, i_0), (i_1, j_1), \dots, (i_p, j_p)\}$ , onde  $\{(i_1, j_1), \dots, (i_p, j_p)\}$  é uma partição de  $C \setminus \{i_0\}$  para cada  $i_0 \in C$ , se  $|C|$  é ímpar e  $\beta$  é par.

Em todos os casos  $i_k < j_k$  para todo  $k = 1, \dots, p$ .

As desigualdades em **SCILS** correspondentes ao (3.25) são dadas por

$$\sum_{(i,j) \in C_s} X_{ij} \leq \left\lfloor \frac{\beta}{2} \right\rfloor, \quad (3.28)$$

para todas as partições de  $C_s$  definidas acima.

**Teorema 3.3.4.** *Se a desigualdade (3.25) é válida para o **QKP**, então as desigualdades em **SCILS** (3.28) são válidas para o **QKP** lifted.*

*Demonstração.* A prova da validade de **SCILS** é baseada na relação  $X_{ij} = x_i x_j$ . Note que, a variável binária  $x_i$  indica se um item  $i$  é selecionado, ou não, para a solução, a variável  $X_{ij}$  indica se um par de itens  $i$  e  $j$ , são selecionados para estarem juntos na solução.

1. Se  $|C|$  é par,  $C_s$  é uma partição de  $C$  em exatamente  $|C|/2$  subconjuntos com dois elementos cada, e assim, se pelo menos  $\beta$  elementos de  $C$  podem ser selecionados na solução, claramente pelo menos  $\lfloor \frac{\beta}{2} \rfloor$  subconjuntos de  $C_s$  podem ser selecionados.
2. Se  $|C|$  e  $\beta$  são ímpares,  $C_s$  é uma partição de  $C \setminus \{i_0\}$  em exatos  $|C - 1|/2$  subconjuntos com dois elementos cada, onde  $i_0$  pode ser qualquer elemento de  $C$ . Neste caso, se pelo menos  $\beta$  elementos de  $C$  podem ser selecionados para a solução, claramente, pelo menos  $\frac{\beta-1}{2}$  ( $= \lfloor \frac{\beta}{2} \rfloor$ ) subconjuntos de  $C_s$  também podem ser selecionados.
3. Se  $|C|$  é ímpar e  $\beta$  é par,  $C_s$  é a união de  $\{(i_0, i_0)\}$  com a partição de  $C \setminus \{i_0\}$  em exatos  $|C - 1|/2$  subconjuntos com dois elementos cada, onde  $i_0$  pode ser qualquer elemento de  $C$ . Neste caso, se pelo menos  $\beta$  elementos de  $C$  podem ser selecionados na solução, claramente pelo menos  $\frac{\beta}{2}$  ( $= \lfloor \frac{\beta}{2} \rfloor$ ) subconjuntos de  $C_s$  podem ser também selecionados.

□

Dada uma solução  $(\bar{x}, \bar{X})$  de **CRel**, apresentamos o problema de separação **MILP**<sub>2</sub>, que busca pela desigualdade de **SCILS** mais violada por  $\bar{X}$ .

Seja a matriz  $A \in \{0, 1\}^{n \times \frac{n(n+1)}{2}}$ . Nas primeiras  $n$  colunas de  $A$  temos a matriz identidade  $n \times n$ . Nas  $n(n-1)/2$  colunas restantes de  $A$  existem exatos dois elementos iguais a 1 em cada coluna. Todas as colunas de  $A$  são distintas. Por exemplo, para  $n = 4$  temos:

$$A := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

As colunas de  $A$  representam todos os subconjuntos de itens em  $N$  com um ou dois elementos. Seja

$$\begin{aligned} z^* := & \text{maximize}_{\alpha, v, K, y} \text{trace}(\bar{X}K) - 2v, & (\text{MILP}_2) \\ \text{sujeito a:} & w'\alpha \geq c + 1, \\ & K_{i,i} = 2y_i, & i = 1, \dots, n, \\ & \sum_{i=1}^n y_i \leq 1, \\ & K_{i,j} = \sum_{t=n+1}^{n(n+1)/2} A_{i,t}A_{j,t}y_t, & i, j = 1, \dots, n, i < j, \\ & v \geq (e'\alpha - 1)/2 - 0.5, \\ & v \leq (e'\alpha - 1)/2, \\ & y_t \leq 1 - A_{i,t} + \alpha_i, & i = 1, \dots, n, t = 1, \dots, \frac{n(n+1)}{2}, \\ & \alpha \leq Ay \leq \alpha + \left(\frac{n(n+1)}{2}\right)(1 - \alpha), \\ & \alpha \in \{0, 1\}^n, y \in \{0, 1\}^{\frac{n(n+1)}{2}}, \\ & v \in \mathbb{Z}, K \in \mathbb{S}^n. \end{aligned}$$

Se  $\alpha^*, v^*, K^*, y^*$  são a solução do  $\text{MILP}_2$ , com  $z^* > 0$ , então uma desigualdade particular **SCILS** é dada por

$$\text{trace}(K^*X) \leq 2v^* \quad (3.29)$$

e é violada por  $\bar{X}$ . O vetor binário  $\alpha^*$  define o **CI** do qual o corte é derivado. Como o **CI** é dado por  $\alpha^*x \leq e'\alpha^* - 1$ , podemos concluir que o corte gerado pertence ao caso (1) ou ao caso (3) da Definição 3.3.2. Este fato é considerado na formulação do  $\text{MILP}_2$ . O vetor  $y^*$  define uma partição  $C_s$  como a apresentada no caso (3), se  $\sum_{i=1}^n y_i = 1$ , e no caso (1), caso contrário. Finalmente observe que o número 2 do lado direito do corte (3.29) é devido ao fato de  $K^*$  ser uma matriz simétrica.

Podemos repetir as observações feitas para o  $\text{MILP}_1$ . Qualquer solução viável do  $\text{MILP}_2$ , tal que,  $\text{trace}(\bar{X}K) > 2v$  gera uma desigualdade válida para **CRel**, que é violada por  $\bar{X}$ . Portanto não é necessário resolver o  $\text{MILP}_2$  até a otimalidade para gerar o corte. Desta forma para gerar cortes distintos, podemos resolver o  $\text{MILP}_2$  várias vezes (sem necessariamente garantir a otimalidade), cada vez adicionando ao  $\text{MILP}_2$ , o

seguinte corte para evitar cortes gerados anteriormente

$$\sum_{i=1}^{\frac{n(n+1)}{2}} \bar{y}_i(1 - y_i) \geq 1, \quad (3.30)$$

onde  $\bar{y}$  é o valor da variável  $y$  na solução do **MILP**<sub>2</sub>, ao gerar o corte anterior.

O corte  $\alpha^* x \leq e' \alpha^* - 1$  pode não ser uma cobertura mínima. Para gerar cortes válidos mais fortes, novamente podemos adicionar a função objetivo do **MILP**<sub>2</sub>, o termo  $-\delta e' \alpha$ , para algum  $\delta > 0$ . Desta forma a função objetivo priorizará coberturas mínimas, que podem ser reescritas como **LCI** utilizando o Algoritmo 1. Neste caso entretanto, após calcular o **LCI**, é necessário resolver o **MILP**<sub>2</sub> novamente fixando  $\alpha$  em valores que representem o **LCI** e fixando  $v$  de modo que o lado direito da desigualdade seja igual ao lado direito do **LCI**. Todos os componentes de  $y$  que forem iguais a 1 na solução anterior do **MILP**<sub>2</sub> devem ser fixados em 1. A nova solução do **MILP**<sub>2</sub> deve indicar outro subconjunto de  $N$  que será adicionado a  $C_s$ . Um último detalhe deve ser levado em consideração, se a cobertura  $C$  correspondente ao **LCI** é tal que  $|C|$  é ímpar e o lado direito do **LCI** também é ímpar, então o corte gerado pertence ao caso (2) da Definição 3.3.2, desta forma, o **MILP**<sub>2</sub> deve ser modificado conforme a definição. Especificamente, a segunda e terceira restrições do **MILP**<sub>2</sub>, devem ser substituídas por

$$\begin{aligned} K_{i,i} &= 0, & i &= 1, \dots, n, \\ \sum_{i=1}^n y_i &= 1. \end{aligned}$$

*Observação 3.3.1.* Seja  $\gamma := |C|$ . Então, o número de desigualdades **SCILS** é

$$\frac{\gamma!}{2^{\binom{\gamma}{2}} (\frac{\gamma!}{2})},$$

se  $\gamma$  é par, ou

$$\gamma \times \frac{(\gamma - 1)!}{2^{\binom{\gamma-1}{2}} (\frac{\gamma-1!}{2})},$$

se  $\gamma$  é ímpar. Na prática, se  $\gamma$  não for pequeno, o número de desigualdades **SCILS** pode ser muito grande, o que torna inviável a geração de todas as desigualdades. Portanto geramos apenas um número fixo de desigualdades **SCILS** que estão descritas no Capítulo 5.

Finalmente, estendemos as ideias apresentadas até aqui para o caso mais geral de desigualdades da mochila. Note que a seguinte discussão aplica-se ao **LCI** geral, onde  $\alpha_j \in \mathbb{Z}_+, \forall j \in N \setminus C$ . Seja

$$\sum_{j \in N} \alpha_j x_j \leq \beta. \quad (3.31)$$

uma desigualdade de mochila válida para o **KPol**, com  $\alpha_j, \beta \in \mathbb{Z}_+, \beta \geq \alpha_j, \forall j \in N$ .

**Definição 3.3.3** (*Conjunto de desigualdades da mochila no espaço lifted, Set of knapsack inequalities in the lifted space, SKILS*). Seja  $\alpha_j$  o coeficiente de  $x_j$  em (3.31) e seja  $\{C_1, \dots, C_q\}$  uma partição de  $N$ , tal que  $\alpha_u = \alpha_v$ , se  $u, v \in C_k$  para algum  $k$ , e  $\alpha_u \neq \alpha_v$ , caso contrário. A desigualdade de mochila (3.31) pode ser escrita como

$$\sum_{k=1}^q \left( \tilde{\alpha}_k \sum_{j \in C_k} x_j \right) \leq \beta. \quad (3.32)$$

Para  $k = 1, \dots, q$ , seja  $C_{l_k} := \{(i_{k_1}, j_{k_1}), \dots, (i_{k_{p_k}}, j_{k_{p_k}})\}$ , onde  $i < j$  para todo  $(i, j) \in C_{l_k}$ , e

- $C_{l_k}$  é uma partição de  $C_k$ , se  $|C_k|$  é par.
- $C_{l_k}$  é uma partição de  $C_k \setminus \{i_{k_0}\}$ , onde  $i_{k_0} \in C_k$ , se  $|C_k|$  é ímpar.

As desigualdades **SKILS** correspondentes a (3.31) são dadas por

$$\sum_{k=1}^q \left( \tilde{\alpha}_k X_{i_{k_0} i_{k_0}} + 2\tilde{\alpha}_k \sum_{(i,j) \in C_{l_k}} X_{ij} \right) \leq \beta, \quad (3.33)$$

para todas as partições  $C_{l_k}$ ,  $k = 1, \dots, q$ , definidas anteriormente, e para  $i_{k_0} \in C_k \setminus C_{l_k}$ . (Se  $|C_k|$  é par,  $C_k \setminus C_{l_k} = \emptyset$ , e o termo na variável  $X_{i_{k_0} i_{k_0}}$  não existir.)

*Observação 3.3.2.* Considere  $\{C_1, \dots, C_q\}$  como determinado pela Definição 3.3.3. Para  $k = 1, \dots, q$ , seja  $\gamma_k := |C_k|$  e

$$NC_{l_k} := \frac{\gamma_k!}{2^{\binom{\gamma_k}{2}} (\frac{\gamma_k}{2}!)},$$

se  $\gamma_k$  é par, ou

$$NC_{l_k} := \gamma_k \times \frac{(\gamma_k - 1)!}{2^{\binom{\gamma_k - 1}{2}} (\frac{\gamma_k - 1}{2}!)},$$

se  $\gamma_k$  é ímpar.

Então o número de desigualdades **SKILS** é

$$\prod_{k=1}^q NC_{l_k}.$$

*Observação 3.3.3.* Se  $\gamma_k := |C_k|$  é par para todo  $k$ , ou se  $\tilde{\alpha}_k$  é ímpar para todo  $k$ , tal que  $\gamma_k$  é ímpar, então  $\beta$ , o lado direito da desigualdade (3.33), pode ser substituído por  $2 \times \lfloor \frac{\beta}{2} \rfloor$ , que fortalece a desigualdade, caso  $\beta$  seja ímpar.

### Corolário 3.3.1.

Se a desigualdade (3.31) é válida para **QKP**, então a desigualdade (3.33) **SKILS** é válida para o **QKP** *lifted*, independente da aplicação, ou não, da modificação sugerida na Observação 3.3.3.

*Demonstração.* Novamente o resultado é verificado usando o mesmo argumento utilizado na prova do Teorema 3.3.4, isto é, considerando que  $X_{ij} = 1$ , se e somente se  $x_i = x_j = 1$ .  $\square$

### 3.3.3 Relação de dominância entre as novas desigualdades válidas

Iniciamos esta seção investigando se **SCILS** domina **CILS** ou vice versa.

**Teorema 3.3.5.** *Seja  $C$  uma cobertura em (3.25) e considere  $\gamma := |C|$  par.*

1. *Se  $\beta = \gamma - 1$ , então a soma de todas as desigualdades **SCILS** são equivalentes a **CILS**. Portanto, neste caso o conjunto de desigualdades **SCILS** domina **CILS**.*
2. *Se  $\beta < \gamma - 1$ , então não existe relação de dominância entre **SCILS** e **CILS**.*

*Demonstração.* Seja  $sum(\mathbf{SCILS})$  a desigualdade obtida pela adição de todas as desigualdades **SCILS**, e seja  $rhs(sum(\mathbf{SCILS}))$  o lado direito deste somatório. Temos que,  $rhs(sum(\mathbf{SCILS}))$  é igual ao número e desigualdades **SCILS** multiplicado pelo lado direito de cada desigualdade, ou seja:

$$rhs(sum(\mathbf{SCILS})) = \frac{\gamma!}{2^{\binom{\gamma}{2}} (\frac{\gamma}{2}!)} \times \left\lfloor \frac{\beta}{2} \right\rfloor.$$

O coeficiente de cada variável  $X_{ij}$  em  $sum(\mathbf{SCILS})$  ( $coef_{ij}$ ) é dado pelo número de desigualdades no conjunto **SCILS**, no qual  $X_{ij}$  aparece, isto é:

$$coef_{ij} = \frac{(\gamma - 2)!}{2^{\binom{\gamma-2}{2}} (\frac{\gamma-2}{2}!)}.$$

Dividindo  $rhs(sum(\mathbf{SCILS}))$  por  $coef_{ij}$ , obtêm-se

$$rhs(sum(\mathbf{SCILS}))/coef_{ij} = (\gamma - 1) \times \left\lfloor \frac{\beta}{2} \right\rfloor. \quad (3.34)$$

Por outro lado, o lado direito de **CILS** é:

$$rhs(\mathbf{CILS}) = \binom{\beta}{2} = \frac{\beta(\beta - 1)}{2}. \quad (3.35)$$

1. Substituindo  $\beta$  por  $\gamma - 1$ , e  $\left\lfloor \frac{\beta}{2} \right\rfloor$  por  $\frac{\beta-1}{2}$  (desde que  $\beta$  seja ímpar), encontra-se o resultado.



2. Considere, por exemplo,  $C = \{1, 2, 3, 4, 5, 6\}$  e  $\beta = 3$  ( $\beta < \gamma - 1$  ímpar). Neste caso o **CILS** é:

$$\begin{aligned} X_{12} + X_{13} + X_{14} + X_{15} + X_{16} + X_{23} + X_{24} + X_{25} \\ + X_{26} + X_{34} + X_{35} + X_{36} + X_{45} + X_{46} + X_{56} \leq 3. \end{aligned}$$

E uma desigualdade **SCILS** particular é:

$$X_{12} + X_{34} + X_{56} \leq 1. \quad (3.36)$$

A solução  $X_{1j} = 1$ , para  $j = 2, \dots, 6$ , e as demais variáveis iguais a zero, satisfaz todas as desigualdades **SCILS**, porque apenas uma das variáveis positivas aparece em cada desigualdade no conjunto. Entretanto, a solução não satisfaz o **CILS**. Por outro lado, a solução  $X_{12} = X_{34} = X_{56} = 1$  e todas as outras variáveis iguais a zero, satisfaz o **CILS**, mas não satisfaz (3.36).

Agora considere  $C = \{1, 2, 3, 4, 5, 6\}$  e  $\beta = 4$  ( $\beta < \gamma - 1$  par). Neste caso, o **CILS** é definido por:

$$\begin{aligned} X_{12} + X_{13} + X_{14} + X_{15} + X_{16} + X_{23} + X_{24} + X_{25} + X_{26} \\ + X_{34} + X_{35} + X_{36} + X_{45} + X_{46} + X_{56} \leq 6. \end{aligned}$$

E uma desigualdade **SCILS** particular é:

$$X_{12} + X_{34} + X_{56} \leq 2. \quad (3.37)$$

A solução  $X_{1j} = 1$ , para  $j = 2, \dots, 6$ ,  $X_{2j} = 1$ , para  $j = 3, \dots, 6$ , e todas as outras variáveis iguais a zero satisfaz todas as desigualdades **SCILS**, porque no máximo duas variáveis positivas aparecem em cada desigualdade no conjunto. Entretanto, a solução não satisfaz **CILS**. Por outro lado, a solução  $X_{12} = X_{34} = X_{56} = 1$ , e todas as outras variáveis iguais a zero, satisfaz o **CILS**, mas não satisfaz (3.37).

□

**Teorema 3.3.6.** *Seja  $C$  uma cobertura em (3.25) e considere  $\gamma := |C|$  ímpar. Então não existe dominância entre **SCILS** e **CILS**.*

*Demonstração.* Considere, por exemplo,  $C = \{1, 2, 3, 4, 5\}$  e  $\beta = 3$  ( $\beta$  ímpar). Neste caso o **CILS** é:

$$X_{12} + X_{13} + X_{14} + X_{15} + X_{23} + X_{24} + X_{25} + X_{34} + X_{35} + X_{45} \leq 3.$$

E uma desigualdade **SCILS** particular é:

$$X_{23} + X_{45} \leq 1. \quad (3.38)$$

A solução  $X_{1j} = 1$ , para  $j = 1, \dots, 5$ , e todas as outras variáveis iguais a zero, satisfaz a todas as desigualdades **SCILS**, porque apenas uma das variáveis positivas aparece em cada desigualdade neste conjunto. Entretanto, a solução não satisfaz **CILS**. Por outro lado, a solução  $X_{23} = X_{45} = 1$  e todas as outras variáveis iguais a zero, satisfaz **CILS**, mas não satisfaz (3.38).

Considere  $C = \{1, 2, 3, 4, 5\}$  e  $\beta = 4$  ( $\beta$  par). Neste caso o **CILS** é:

$$X_{12} + X_{13} + X_{14} + X_{15} + X_{23} + X_{24} + X_{25} + X_{34} + X_{35} + X_{45} \leq 6.$$

E uma desigualdade **SCILS** particular é:

$$X_{11} + X_{23} + X_{45} \leq 2. \quad (3.39)$$

A solução  $X_{1j} = 1$ , para  $j = 1, \dots, 5$ ,  $X_{2j} = 1$ , para  $j = 2, \dots, 5$ , e todas as outras variáveis iguais a zero, satisfaz todas as desigualdades **SCILS**, porque no máximo duas variáveis positivas aparecem em cada desigualdade do conjunto. Entretanto, a solução não satisfaz ao **CILS**. Por outro lado a solução  $X_{11} = X_{23} = X_{45} = 1$ , e todas as outras variáveis iguais a zero, satisfaz **CILS**, mas não satisfaz (3.39). □

Agora investigamos se **SCILS** é apenas um caso particular de **SKILS**, quando  $\alpha_j \in \{0, 1\}$ , para todo  $j \in N$  em (3.31).

**Teorema 3.3.7.** *Caso a modificação sugerida pela Observação 3.3.3 seja aplicada, então se  $|C|$  é par em (3.25), **SCILS** é apenas um caso particular de **SKILS**. Porém se  $|C|$  é ímpar a desigualdade em **SCILS** é mais forte.*

*Demonstração.* Se  $|C|$  é par, o resultado é facilmente verificado. Se  $|C|$  é ímpar, as desigualdades **SCILS** são:

$$2 \sum_{(i,j) \in C_s} X_{ij} \leq \beta - 1,$$

se  $\beta$  é ímpar e

$$2X_{i_0 i_0} + 2 \sum_{(i,j) \in C_s} X_{ij} \leq \beta,$$

se  $\beta$  é par, e as desigualdades **SKILS** são:

$$X_{i_0 i_0} + 2 \sum_{(i,j) \in C_s} X_{ij} \leq \beta,$$

para todo  $\beta$ . Em todos os casos  $C_s$  é uma partição de  $C \setminus \{i_0\}$ , onde  $i_0 \in C$ .

Tanto para  $\beta$  par ou ímpar fica claro que **SCILS** é mais forte que **SKILS**.

□

### 3.4 Limites inferiores para a solução da relaxação **QKP** *lifted*

Para avaliar a qualidade do limite superior encontrado com a relaxação **CRel**, comparamos este limite com o limite inferior do **QKP**, dado pelas soluções viáveis construídas de forma heurística.

Seja  $(\bar{x}, \bar{X})$  uma solução da relaxação **CRel**. Aplicamos inicialmente a análise de componente principal, *principal component analysis* (PCA) [29] para construir uma aproximação para a solução do **QKP** e então aplicamos um procedimento de arredondamento para obter uma solução viável para o problema. PCA seleciona o maior autovalor e o autovetor correspondente da matriz  $\bar{X}$ , denotados por  $\bar{\lambda}$  e  $\bar{v}$ , respectivamente. Então  $\bar{\lambda}\bar{v}\bar{v}^T$  é uma aproximação rank-1 de  $\bar{X}$ . Consideramos  $\bar{x} = \bar{\lambda}^{\frac{1}{2}}\bar{v}$  uma aproximação da solução  $x$  em **QKP**. Finalmente arredondamos  $\bar{x}$  para uma solução binária que satisfaz a restrição de capacidade, usando a técnica descrita no Algoritmo 3.

---

**Algoritmo 3:** Procedimento heurístico

---

- 1 Entrada: A solução  $\bar{X}$  da relaxação **CRel**, o vetor de pesos  $w$  e a capacidade da mochila  $c$ .
- 2 Sejam  $\bar{\lambda}$  e  $\bar{v}$ , respectivamente, o maior autovalor e o autovetor correspondentes de  $\bar{X}$ .
- 3 Faça  $\bar{x} = \bar{\lambda}^{\frac{1}{2}}\bar{v}$ .
- 4 Arredonde  $\bar{x}$  para  $\hat{x} \in \{0, 1\}^n$ .
- 5 Enquanto  $w^T\hat{x} > c$ 
  - Faça  $i = \arg \min_{j \in N} \{\bar{x}_j | \bar{x}_j > 0\}$ .
  - Faça  $\bar{x}_i = 0, \hat{x}_i = 0$ .

Final do Enquanto

Saída: uma solução viável  $\hat{x}$  de **QKP**.

---

# Capítulo 4

## Outra abordagem pesquisada

Neste Capítulo propomos a utilização de um outro algoritmo de plano de cortes que, de forma iterativa, fortalece uma relaxação linear inicial do problema da mochila quadrática 0-1, com o objetivo de encontrar limites de boa qualidade sem a necessidade de resolver problemas de programação semidefinida. Para fortalecer a relaxação linear utilizada no algoritmo de plano de cortes adicionamos a relaxação inicial do problema desigualdades RLT válidas, que são bem conhecidas e referenciadas na literatura, e cortes de programação semidefinida.

Considere o problema da mochila quadrática 0-1 definido em 3.1. Conforme descrito no Capítulo 2 o **QKP** pode ser reescrito no espaço matricial e relaxado conforme proposto por Helmberg, Rendl e Weismantel [8], [10]:

$$\begin{aligned} (HRW) \quad & \text{maximize} \quad \langle Q, X \rangle \\ & \text{sujeito a:} \quad \sum_{j \in N} w_j X_{ij} - X_{ii}c \leq 0, \quad i \in N, \\ & \quad \quad \quad X - \text{diag}(X)\text{diag}(X)^T \succeq 0, \end{aligned} \quad (4.1)$$

Observe que nesta formulação a restrição  $X - \text{diag}(X)\text{diag}(X)^T \succeq 0$  garante que  $0 \leq x_i \leq 1$ .

Esta formulação tem como base a formulação 2.2 e as relaxações apresentadas anteriormente na seção 2.1.6 desta tese. Em [14] a partir de resultados computacionais mostrou-se que esta relaxação *SDP* apresentou os melhores limites superiores para o **QKP** nas instâncias abordadas.

### 4.1 Novos Limites Superiores

Nesta seção explicamos como aplicar um algoritmo de plano de cortes, (*cutting plane algorithm* - CPA), para obter limites superiores para o **QKP**.

A cada iteração, o algoritmo de plano de cortes resolve uma relaxação linear do problema. Esta relaxação é obtida com a adição dos cortes *SDP*. A ideia central é construir

iterativamente, uma aproximação externa ao conjunto viável do problema 2.2, resolvendo uma sequência de problemas lineares. A cada iteração é adicionado um corte à formulação linear e assim eliminamos a solução da relaxação anterior do conjunto viável. Consequentemente o conjunto viável torna-se mais bem aproximado.

O objetivo deste algoritmo de plano de cortes é encontrar limitantes quase tão bons quanto os encontrados pela relaxação *SDP*, porém resolvendo apenas problemas lineares.

O modelo linear apresentado abaixo ( $\tilde{LP}$ ) é uma relaxação de *HRW* que é, por um lado, mais fraca, pois a restrição  $X - \text{diag}(X)\text{diag}(X)^T \succeq 0$  é substituída por  $X = X^T$  e, por outro lado, reforçada pois adicionamos nele a restrição de capacidade (2.5), as restrições RLT (2.4) e a restrição de limite (2.6).

$$\begin{aligned}
(\tilde{LP}) \quad & \text{maximize} \quad \langle Q, X \rangle \\
& \text{sujeito a:} \quad \sum_{j \in N} w_j X_{jj} \leq c, \\
& \quad \sum_{j \in N} w_j X_{ij} - X_{ii}c \leq 0, \quad i \in N, \\
& \quad X = X^T, \\
& \quad X_{ij} \leq X_{ii}, \quad i, j \in N, i < j, \\
& \quad X_{ij} \leq X_{jj}, \quad i, j \in N, i < j, \\
& \quad X_{ij} \geq 0, \quad i, j \in N, i < j, \\
& \quad 0 \leq X_{jj} \leq 1, \quad j \in N,
\end{aligned} \tag{4.2}$$

É importante observar que neste modelo ( $\tilde{LP}$ ) não é necessário incluir a restrição RLT ( $X_{ii} + X_{jj} \leq 1 + X_{ij}$ ) pois  $q_{ij} \geq 0$  na função objetivo.

Billionnet e Calmels [5] propuseram em seu trabalho uma relaxação linear para o **QKP**:

$$\begin{aligned}
(BC) \quad & \text{maximize} \quad \sum_{i,j \in N, i < j} 2q_{ij}y_{ij} + \sum_{j \in N} q_{jj}x_j \\
& \text{sujeito a:} \quad \sum_{j \in N} w_j x_j \leq c, \\
& \quad y_{ij} \leq x_i, \quad i, j \in N, i < j, \\
& \quad y_{ij} \leq x_j, \quad i, j \in N, i < j, \\
& \quad x_i + x_j \leq 1 + y_{ij}, \quad i, j \in N, i < j, \\
& \quad y_{ij} \geq 0, \quad i, j \in N, i < j, \\
& \quad 0 \leq x_j \leq 1, \quad j \in N.
\end{aligned} \tag{4.3}$$

Em [14] a relaxação (*BC*) é apresentada como a relaxação mais fraca e computacionalmente mais barata, para obter limites superiores para o **QKP**. Em nosso algoritmo de plano de cortes temos como objetivo gerar limites mais apertados que os encontrados utilizando a formulação proposta por Billionnet e Calmels [5] (*BC*) e computacionalmente mais baratos que os encontrados pela formulação proposta por Helmberg, Rendl, e Weismantel [8], [10] (*HRW*).

## 4.2 O Algoritmo de plano de cortes

Seja  $Y$  a seguinte matriz simétrica  $(n+1) \times (n+1)$

$$Y := \begin{pmatrix} 1 & \text{diag}(X)^T \\ \text{diag}(X) & X \end{pmatrix}. \quad (4.4)$$

O procedimento que utilizaremos no algoritmo de plano de cortes para fortalecer a relaxação 4.2 ( $\tilde{LP}$ ) é baseado nas seguintes equivalências:

$$Y \succeq 0 \iff X - \text{diag}(X)\text{diag}(X)^T \succeq 0,$$

$$Y \succeq 0 \iff v^T Y v \geq 0, \forall v \in \mathbb{R}^{n+1},$$

Os cortes  $SDP$ ,  $\bar{v}^T Y \bar{v} \geq 0$ , são adicionados a relaxação do **QKP** iterativamente. Os vetores  $\bar{v}$  são selecionados de forma criteriosa conforme descrito em [30], trabalho no qual são considerados problemas de programação não convexa.

Seja a decomposição espectral de  $Y$ :

$$Y = \sum_{k=1}^{n+1} \lambda_k v_k v_k^T,$$

onde  $\lambda_k$  são o autovalores e  $v_k$  os autovetores de  $Y$ . Se  $Y \succeq 0$ , então  $\lambda_k \geq 0$  para todo  $k = 1, \dots, n+1$ , caso contrário existe  $\bar{k}$  tal que  $\lambda_{\bar{k}} < 0$ .

Como  $v_{\bar{k}}^T Y v_{\bar{k}} = \lambda_{\bar{k}}$ , a desigualdade  $v_{\bar{k}}^T Y v_{\bar{k}} \geq 0$ , é satisfeita por todas as matrizes semidefinidas positivas  $(n+1) \times (n+1)$ , e é violada por  $Y$ .

O algoritmo proposto neste trabalho utiliza as ideias acima descritas para separar de forma iterativa os cortes  $SDP$  e então adiciona-los à formulação inicial ( $\tilde{LP}$ ) de forma a fortalecer os limites calculados.

Se em nosso algoritmo fosse imposta a condição de parada apenas quando a matriz  $\tilde{Y}$  se tornasse semidefinida positiva, ou seja, quando  $\lambda_k \geq 0$  para todo  $k = 1, \dots, n+1$ , nossos limites não seriam piores que os encontrados por ( $HRW$ ). Contudo o esforço computacional necessário para satisfazer este critério de parada poderia ser muito grande e não compensar.

O principal foco dos resultados computacionais apresentados neste trabalho, utilizando o CPA, é analisar o compromisso entre a qualidade do limitante obtido pelo algoritmo de plano de cortes e o esforço computacional necessário.

## 4.3 Programação Disjuntiva

A programação disjuntiva foi originalmente apresentada por BALAS [31]. Para utilizá-la é necessário uma relaxação linear do problema e uma disjunção que satisfaça todas as soluções viáveis do problema.

---

**Algoritmo 4:** Algoritmo de Plano de Cortes - CPA

---

1 Enquanto Critério de Parada

Seja  $\tilde{X}$  a solução ótima de  $(\tilde{LP})$ , 4.2.

Seja  $\tilde{Y} := \begin{pmatrix} 1 & \text{diag}(\tilde{X})^T \\ \text{diag}(\tilde{X}) & \tilde{X} \end{pmatrix}$ .

Sejam  $\lambda_k$  e  $v_k$  para  $k = 1, \dots, n+1$  respectivamente, os autovalores e os autovetores de  $\tilde{Y}$ , tal que  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n+1}$ .

Seja  $k := 1$ .

Enquanto  $\lambda_k < \lambda_{MAX}$  e  $k \leq K_{MAX}$

Adicione a restrição  $v_k^T Y v_k \geq 0$  ao  $(\tilde{LP})$ , onde  $Y$  é definido em 4.4.  $k := k + 1$ ;

Final do Enquanto

Final do Enquanto

Saída: O valor da solução ótima de  $(\tilde{LP})$ .

---

Saxena et al. [32] pesquisaram a aplicação das técnicas de programação disjuntiva ao problema de programação inteira mista com restrições quadráticas, *mixed integer quadratically constrained programming* (*MIQCP*). Este problema, como o **QKP**, apresenta dois tipos de não convexidade, a saber, variáveis inteiras e restrições quadráticas não convexas. Em [32] utiliza-se técnicas de programação disjuntiva e métodos *lift-and-project* para obter relaxações fortes para o *MIQCP*.

### 4.3.1 Corte Secante

A restrição não convexa  $X = xx^T$  gera dois tipos de desigualdades válidas:

Desigualdade válida tipo I

$$X = xx^T$$

$$X - xx^T \succeq 0$$

$$\underbrace{X \succeq xx^T}$$

Já tratadas pelo CPA

na seção anterior ( $\lambda_k < 0$ )

Desigualdade válida tipo II

$$X = xx^T$$

$$xx^T - X \succeq 0$$

$$-X \succeq -xx^T$$

$$X \preceq xx^T$$

$$X - xx^T \preceq 0.$$

O primeiro tipo foi tratado em nosso algoritmo CPA, para  $\lambda_k < 0$ . O segundo tipo de desigualdade,  $X - xx^T \preceq 0$ , será utilizado para derivar disjunções a partir dos autovalores positivos,  $\lambda_k > 0$ . Estas disjunções são de dois tipos:

- Disjunção diretamente derivada dos autovetores da matriz  $X - xx^T$  associados aos autovalores positivos  $\lambda_k > 0$ ;
- Disjunção obtida através da combinação de vários autovetores.

Aplicando as ideias apresentadas em [32] nosso objetivo é desenvolver uma relaxação convexa para a restrição não-convexa

$$X - xx^T \preceq 0. \quad (4.5)$$

Podemos modelar, de forma equivalente, a restrição (4.5) por um número infinito de desigualdades

$$(\zeta^T x)^2 \geq \langle (\zeta \zeta^T), X \rangle, \quad (4.6)$$

para todo  $\zeta \in \mathbb{R}^n$ . A desigualdade (4.6) é válida mas não pode ser diretamente adicionada ao nosso algoritmo, pois esta desigualdade é não convexa, sua adição ao modelo causaria a perda de convexidade do mesmo. Assim para adicioná-la devemos primeiro convexificá-la conforme feito em [32].

Para convexificar (4.6) devemos substituir a função  $-(\zeta^T x)^2$  pela expressão da reta secante dentro do intervalo  $[\eta_L(\zeta), \eta_U(\zeta)]$ . Desta forma a relaxação convexa de (4.6) é dada por:

$$(\zeta^T x)(\eta_L(\zeta) + \eta_U(\zeta)) - \eta_L(\zeta)\eta_U(\zeta) \geq \langle (\zeta \zeta^T), X \rangle. \quad (4.7)$$

Seja  $P$  uma relaxação poliédrica do conjunto viável do problema  $(\tilde{L}P)$ , (4.2), os limites do intervalo  $[\eta_L(\zeta), \eta_U(\zeta)]$  são calculados da seguinte forma:

$$\eta_L(\zeta) := \min\{\zeta^T x : (x, X) \in P\} \text{ ou seja,}$$

$$\begin{aligned} \eta_L(\zeta) := & \text{ minimize } \zeta^T x \\ & \text{sujeito a: } \sum_{j \in N} w_j X_{jj} \leq c, \\ & \sum_{j \in N} w_j X_{ij} - X_{ii}c \leq 0, \quad i \in N, \\ & X = X^T, \\ & X_{ij} \leq X_{ii}, \quad i, j \in N, i < j, \\ & X_{ij} \leq X_{jj}, \quad i, j \in N, i < j, \\ & X_{ij} \geq 0, \quad i, j \in N, i < j, \\ & 0 \leq X_{jj} \leq 1, \quad j \in N, \end{aligned}$$

e

$$\eta_U(\zeta) := \max\{\zeta^T x : (x, X) \in P\} \text{ ou seja,}$$



$$\begin{aligned}
\eta_U(\varsigma) := & \text{maximize } \varsigma^T x \\
\text{sujeito a: } & \sum_{j \in N} w_j X_{jj} \leq c, \\
& \sum_{j \in N} w_j X_{ij} - X_{ii} c \leq 0, \quad i \in N, \\
& X = X^T, \\
& X_{ij} \leq X_{ii}, \quad i, j \in N, i < j, \\
& X_{ij} \leq X_{jj}, \quad i, j \in N, i < j, \\
& X_{ij} \geq 0, \quad i, j \in N, i < j, \\
& 0 \leq X_{jj} \leq 1, \quad j \in N,
\end{aligned}$$

Tomando como base as ideias apresentadas em [32] suponha que ao resolver a relaxação ( $\tilde{LP}$ ) obtenha-se  $\bar{X}$  e  $\bar{x}$  tais que  $\bar{X} \neq \bar{x}\bar{x}^T$ . Suponha ainda que a matriz  $\bar{X} - \bar{x}\bar{x}^T$  tem, pelo menos, um autovalor positivo ( $\lambda_p > 0$ ). Desta forma podemos escolher o vetor  $\varsigma$  como o autovetor correspondente a qualquer autovalor positivo. A restrição  $(\varsigma^T x)^2 \geq \langle (\varsigma\varsigma^T), X \rangle$  será violada pela solução da relaxação.

A Figura 4.1 ilustra os limites  $[\eta_L(\varsigma), \eta_U(\varsigma)]$ , a região viável e o corte secante em uma função quadrática.

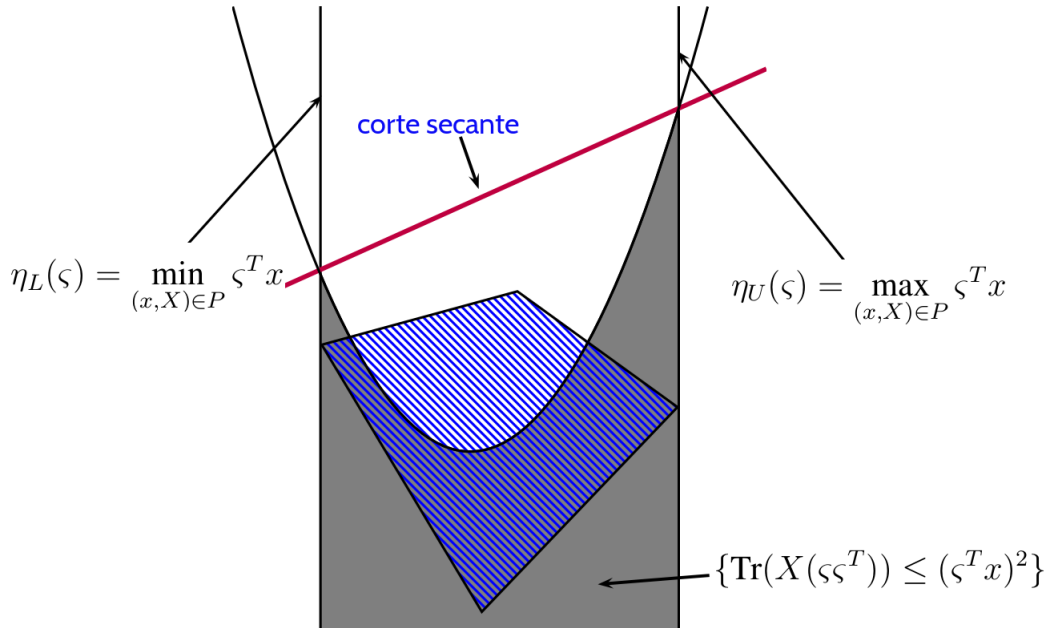


Figura 4.1: Corte secante em uma função quadrática.

### 4.3.2 Cortes Disjuntivos

Os cortes secante são planos de corte válidos para o **QKP**. Mas podemos melhorar estes cortes. Para isto devemos escolher um ponto  $\theta$ , que pode ser o ponto médio por exemplo, tal que  $\theta \in [\eta_L(\varsigma), \eta_U(\varsigma)]$ , e assim temos a disjunção:

$$D_1 := \left\{ (x, X) \in P : \begin{array}{l} \eta_L(\varsigma) \leq \varsigma^T x \leq \theta, \\ (\varsigma^T x)(\eta_L(\varsigma) + \theta) - \theta\eta_L(\varsigma) \geq \langle (\varsigma\varsigma^T), X \rangle \end{array} \right\} \bigvee \quad (4.8)$$

$$D_2 := \left\{ (x, X) \in P : \begin{array}{l} \theta \leq \varsigma^T x \leq \eta_U(\varsigma), \\ (\varsigma^T x)(\eta_U(\varsigma) + \theta) - \theta\eta_U(\varsigma) \geq \langle (\varsigma\varsigma^T), X \rangle \end{array} \right\}.$$

A Figura 4.2 ilustra as duas retas secantes entre  $[\eta_L(\varsigma), \theta]$  dada pela equação  $(\varsigma^T x)(\eta_L(\varsigma) + \theta) - \theta\eta_L(\varsigma) \geq \langle (\varsigma\varsigma^T), X \rangle$  e entre  $[\theta, \eta_U(\varsigma)]$  dada por  $(\varsigma^T x)(\eta_U(\varsigma) + \theta) - \theta\eta_U(\varsigma) \geq \langle (\varsigma\varsigma^T), X \rangle$ .

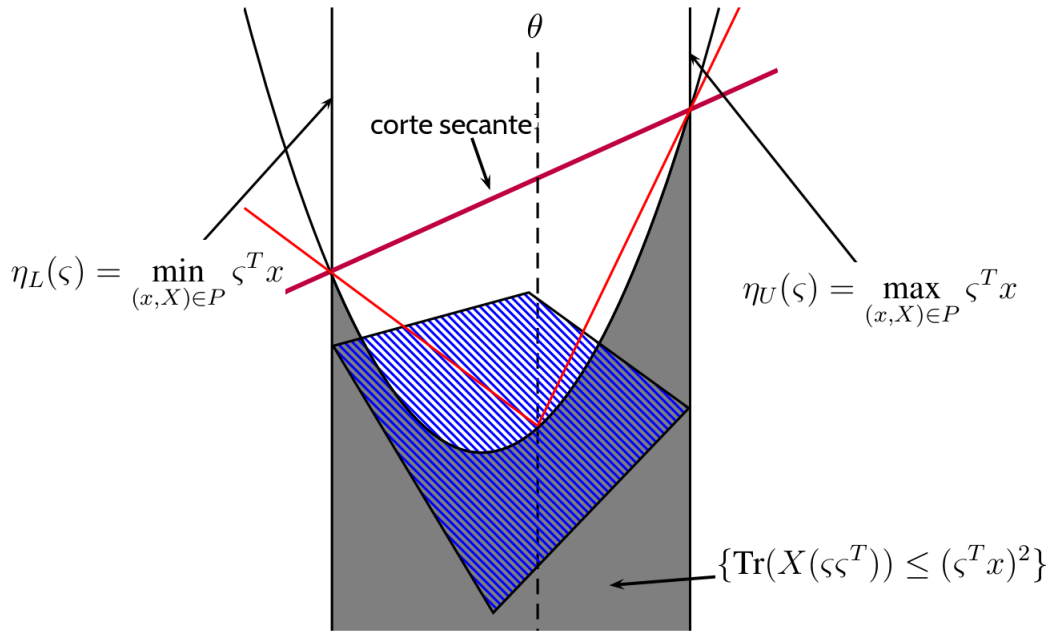


Figura 4.2: Disjunção de cortes secantes.

O vetor  $\varsigma$  é escolhido como o autovetor associado a um autovalor positivo de  $\bar{X} - \bar{x}\bar{x}^T$ , ou seja,  $\lambda > 0 \iff (\varsigma^T x)^2 < \langle (\varsigma\varsigma^T), X \rangle$ . Isto é, todos os autovalores positivos de  $\bar{X} - \bar{x}\bar{x}^T$  produzem uma desigualdade que é violado por  $(\bar{x}, \bar{X})$ .

Conforme explicamos anteriormente não podemos incluir diretamente as desigualdades violadas em nosso algoritmo de plano de cortes pois isto destruiria a convexidade da relaxação 4.2. Portanto precisamos utilizar as técnicas de programação disjuntiva apresentadas em [32].

Dada a disjunção  $D_1 \bigvee D_2$  e um ponto  $\hat{x} \in P$ . Nosso objetivo é separar  $\hat{x}$  de

$$Q := \text{cl conv}(D_1 \cup D_2),$$

ou seja, da envoltória convexa de  $D_1 \cup D_2$ . Desta forma devemos encontrar uma desigualdade válida, tal que:

- Se  $x \in [(P \cap D_1) \cup (P \cap D_2)] \implies x$  satisfaz a desigualdade válida;
- Se  $x \notin [(P \cap D_1) \cup (P \cap D_2)] \implies$  a desigualdade válida deve cortar  $x$ .

Para simplificar a notação vamos representar o conjunto  $P$  apenas pela restrição genérica  $Ax \geq b$ . O mesmo será utilizado para as disjunções  $D_1$  e  $D_2$  que denotaremos, respectivamente, como  $D_1x \geq d_1$  e  $D_2x \geq d_2$ .

Buscamos a desigualdade válida  $\rho x \geq \kappa$  que seja violada por  $\bar{x}$ . Para encontrá-la devemos encontrar  $\rho$  e  $\kappa$  tais que  $\rho \bar{x} < \kappa$ . Notemos que para um dado  $\rho$ , o menor valor que  $\rho x$  assume em cada uma das subregiões da região viável do problema, pode ser obtido pela resolução dos seguintes problemas de programação linear:

$$\begin{aligned} (P \cap D_1) \quad & \text{minimize} \quad \rho^T x \\ & \text{sujeito a:} \quad Ax \geq b \quad \leftarrow (\times u_1) \\ & \quad \quad \quad D_1x \geq d_1 \quad \leftarrow (\times v_1) \end{aligned} \quad (4.9)$$

$$\begin{aligned} (P \cap D_2) \quad & \text{minimize} \quad \rho^T x \\ & \text{sujeito a:} \quad Ax \geq b \quad \leftarrow (\times u_2) \\ & \quad \quad \quad D_2x \geq d_2 \quad \leftarrow (\times v_2) \end{aligned} \quad (4.10)$$

Nos problemas (4.9) e (4.10) as variáveis  $u_1, v_1$  e  $u_2, v_2$  são variáveis duais associadas aos seus respectivos conjuntos de restrições. Note que ao resolver os problemas (4.9) e (4.10) encontramos  $z_{P \cap D_1}^*$  e  $z_{P \cap D_2}^*$ , respectivamente, assim  $\kappa$  deve ser menor ou igual a  $z_{P \cap D_1}^*$  e  $z_{P \cap D_2}^*$ .

Para o problema  $(P \cap D_1) : \kappa \leq \rho^T x \iff \kappa \leq \min \rho^T x \quad \forall x \in P \cap D_1$ , utilizando dualidade de programação linear temos:

$$\begin{aligned} \kappa \leq \quad & \text{maximize} \quad b^T u_1 + d_1^T v_1 \\ & \text{sujeito a:} \quad A^T u_1 + D_1^T v_1 = \rho^T \\ & \quad \quad \quad u_1, v_1 \geq 0 \end{aligned} \quad (4.11)$$

Para o problema  $(P \cap D_2) : \kappa \leq \rho^T x \iff \kappa \leq \min \rho^T x \quad \forall x \in P \cap D_2$ , novamente por dualidade de programação linear temos:

$$\begin{aligned} \kappa \leq \quad & \text{maximize} \quad b^T u_2 + d_2^T v_2 \\ & \text{sujeito a:} \quad A^T u_2 + D_2^T v_2 = \rho^T \\ & \quad \quad \quad u_2, v_2 \geq 0 \end{aligned} \quad (4.12)$$

Para cortar  $\bar{x}$  a equação  $\rho^T \bar{x} - \kappa$  deve ser negativa, ou seja,  $\rho^T \bar{x} - \kappa < 0$ . Desta forma procuramos  $\rho$  e  $\kappa$  que minimizem  $\rho^T \bar{x} - \kappa$ .

- Se o mínimo for negativo  $\implies$  encontramos a desigualdade que corta  $\bar{x}$ ;
- Se o mínimo for positivo  $\implies$  não existe uma desigualdade desta forma violada por  $\bar{x}$ .

Finalmente dado  $\bar{x}$  devemos encontrar  $\rho$  e  $\kappa$  tais que :

$$\begin{aligned}
 (\text{CGLP}) \quad & \text{minimize} \quad \rho^T \bar{x} - \kappa \\
 \text{sujeito a:} \quad & A^T u_1 + D_1^T v_1 = \rho \\
 & A^T u_2 + D_2^T v_2 = \rho \\
 & b^T u_1 + d_1^T v_1 \geq \kappa \\
 & b^T u_2 + d_2^T v_2 \geq \kappa \\
 & u_1, v_1, u_2, v_2 \geq 0 \\
 & \|(u_1, v_1, u_2, v_2)\|_1 = 1.
 \end{aligned} \tag{4.13}$$

A última restrição é introduzida no problema é uma restrição de normalização, utilizada para limitar o valor da função objetivo. A desigualdade linear violada,  $\rho \bar{x} \geq \kappa$ , é encontrada quando o mínimo do (CGLP) é negativo. A Figura 4.3 ilustra o corte disjuntivo na função quadrática. Observe que o corte disjuntivo é mais eficiente que o corte secante.

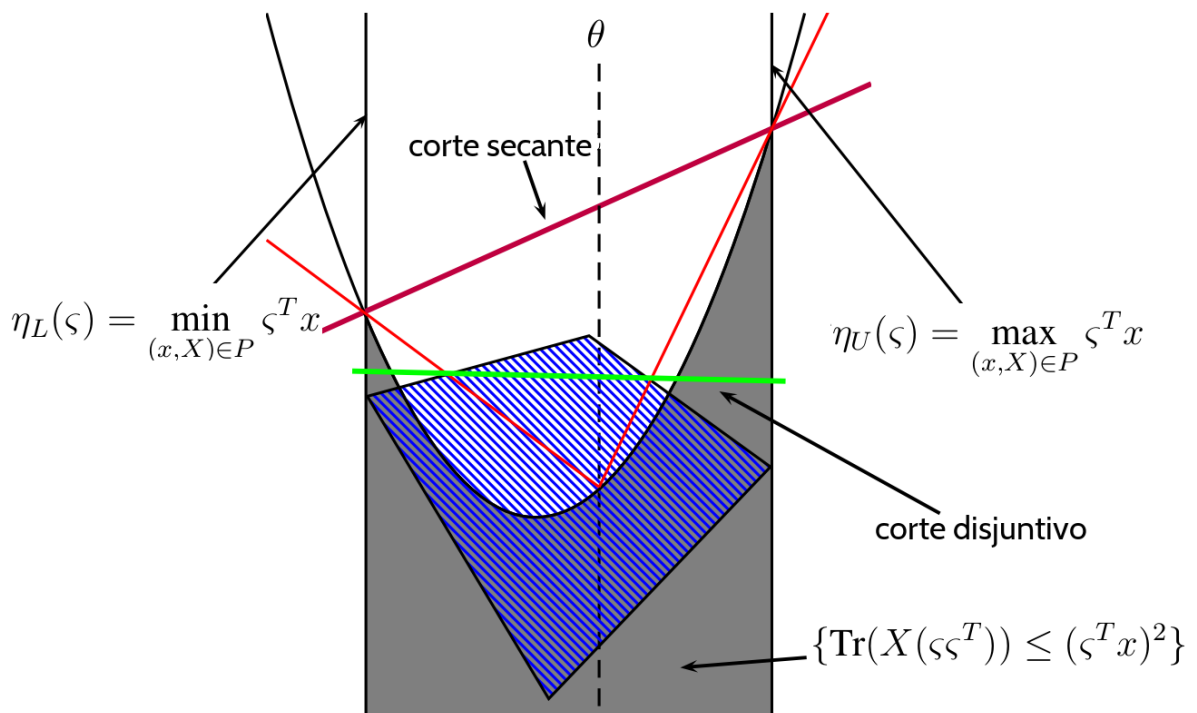


Figura 4.3: Corte disjuntivo em uma função quadrática.

As Figuras 4.1, 4.2 e 4.3 foram retiradas do seminário apresentado pelo Prof. J. Lee no MINO/COST SPRING SCHOOL ON OPTIMIZATION 2014.

# Capítulo 5

## Resultados Computacionais

Neste capítulo apresentamos os resultados computacionais dos métodos discutidos ao longo desta tese. Iniciamos com a apresentação dos resultados obtidos para a relaxação do **QKP** e a aplicação do algoritmo de plano de cortes apresentado no Capítulo 4. Finalmente apresentamos os resultados principais desta tese nos quais combinamos o **IPM** com os novos cortes **CILS** e **SCILS**.

### 5.1 Resultados obtidos com plano de cortes e programação disjuntiva

Para estes testes nosso código foi implementado em Matlab R2014a usando o toolbox CVX 2.1 [33] e o solver MOSEK 7.1 [34]. Os testes foram executados em um computador com processador Intel(R) Core i7, memória de 4GB e sistema operacional Linux Ubuntu 14.04.

Em nossos testes computacionais as instâncias utilizadas foram geradas de forma aleatória. Estas foram utilizadas também na tese de doutorado de Jesus Cunha [35]. O procedimento para gerar as instâncias foi baseado nos trabalhos de [1, 4–6, 12]. Denotamos as instâncias por  $I_{n,d,i}$ :

- $n$  - número de variáveis;
- $d$  - densidade da matriz  $Q$ , porcentagem de benefícios positivos (ou seja, não nulos)  $q_{ij}$ ,  $i \geq j$ ,  $i, j \in N$ , selecionados aleatoriamente no intervalo  $[1, 100]$ ;
- $i$  - índice da instância.

A capacidade da mochila é selecionada aleatoriamente no intervalo  $[50, \sum_{j=1}^n w_j]$  e o peso  $w_j$  é selecionado aleatoriamente no intervalo  $[1, 50]$ , para cada  $j \in N$ .

O objetivo dos testes computacionais é comparar os limites superiores encontrados pelo CPA com os limites obtidos pelas relaxações ( $HRW$ ), 4.1, e ( $BC$ ), 4.3. Utilizamos

para isto diferentes versões de nosso algoritmo de plano de cortes 4. A diferença entre as versões é o número máximo de cortes  $SDP$  que são adicionados à relaxação a cada iteração, denotado como  $K_{MAX}$ .

Desta forma comparamos os limites superiores obtidos pelas cinco relaxações a seguir:

- $LP$  - relaxação proposta por Billionnet e Calmels ( $BC$ ), 4.3;
- $SDP$  - relaxação obtida de ( $HRW$ ) reforçada por (2.4) e 2.5:

$$\begin{aligned}
 (SDP) \quad & \text{maximize} \quad \langle Q, X \rangle \\
 & \text{sujeito a:} \quad \sum_{j \in N} w_j X_{ij} - X_{ii}c \leq 0, \quad i \in N, \\
 & \quad X - \text{diag}(X)\text{diag}(X)^T \succeq 0, \\
 & \quad \sum_{j \in N} w_j x_j \leq c, \\
 & \quad X_{ij} \leq x_i, \\
 & \quad X_{ij} \leq x_j, \\
 & \quad x_i + x_j \leq 1 + X_{ij}, \\
 & \quad X_{ij} \geq 0.
 \end{aligned} \tag{5.1}$$

- $CPA_1$ , Algoritmo 4 com  $K_{MAX} = 1$ , utilizando a relaxação 4.2;
- $CPA_5$ , Algoritmo 4 com  $K_{MAX} = 5$ , utilizando a relaxação 4.2;
- $CPA_{10}$ , Algoritmo 4 com  $K_{MAX} = 10$ , utilizando a relaxação 4.2.

O critério de parada utilizado nestes testes foi escolhido com o objetivo de permitir uma boa análise da convergência do algoritmo de planos de cortes.

Os testes foram executados com limite de tempo de 360 segundos ou até que a matriz de variáveis  $Y$  tornar-se semidefinida positiva. Para evitar a parada prematura por falta de memória, verificamos a cada 5 iterações os cortes  $SDP$  que estão inativos e os eliminamos do modelo.

A Tabela 5.1 apresenta o resumo dos resultados de nossos experimentos comparando as relaxações propostas na literatura com nosso algoritmo de plano de cortes 4. A primeira coluna especifica a instância utilizada. As outras colunas mostram o Gap relativo entre o limite superior ( $UB_i$ ) encontrado com a  $i$ -ésima relaxação, e o valor da solução ótima do problema  $z^*$ . O Gap é calculado da seguinte forma:

$$\text{Gap} = \frac{UB - z^*}{z^*} \times 100$$

Note que o limite de tempo de 360 segundos é maior do que o tempo necessário para resolver  $SDP$ , sempre que conseguimos resolver o  $SDP$ , isto é, quando  $n \leq 100$ . O

<b>Instância</b>	<b>LP</b>	<b>SDP</b>	<b>CPA<sub>1</sub></b>	<b>CPA<sub>5</sub></b>	<b>CPA<sub>10</sub></b>
I <sub>100,25,1</sub>	0.25	0.16	0.25	0.25	0.25
I <sub>100,50,1</sub>	1.11	0.04	0.37	0.16	0.17
I <sub>100,75,1</sub>	6.01	0.49	0.50	0.49	0.49
I <sub>100,100,1</sub>	3.46	0	0	0	0
I <sub>100,25,2</sub>	5.77	0.76	1.42	1.21	1.31
I <sub>100,50,2</sub>	2.82	0.43	0.48	0.48	0.47
I <sub>100,75,2</sub>	1.67	0.20	0.23	0.22	0.23
I <sub>100,100,2</sub>	2.51	0.46	0.46	0.46	0.46
I <sub>100,25,4</sub>	1.05	0.12	1.05	0.55	0.57
I <sub>100,50,4</sub>	3.96	0.19	0.76	0.70	0.65
I <sub>100,75,4</sub>	2.55	0.10	0.20	0.13	0.14
I <sub>100,100,4</sub>	4.32	0.13	0.13	0.13	0.13
I <sub>200,25,1</sub>	0.16	-	0.16	0.16	0.16
I <sub>200,50,1</sub>	0.16	-	0.16	0.16	0.16
I <sub>200,75,1</sub>	16.83	-	0.51	0.48	0.48
I <sub>200,100,1</sub>	0.06	-	0.03	0.03	0.03
Média	3.29	0.26	0.42	0.35	0.36

Tabela 5.1: Gaps obtidos pelas diferentes relaxações do **QKP**.

tempo máximo necessário para resolver *SDP* quando  $n = 100$  é de 340 segundos. No entanto, *SDP* não resolve qualquer instância com  $n = 200$ , devido à falta de memória ou problemas numéricos, o que confirma a bem conhecida dificuldade na resolução das relaxações *SDP* quando o número de variáveis aumenta.

O tempo necessário para resolver *LP* em todas as instâncias apresentadas é muito pequeno, até 1.34 segundos, no entanto, os limites encontrados por esta relaxação às vezes são muito fracos.

Na Tabela 5.1, é possível observar que CPA<sub>1</sub> encontra limites melhores que *LP* para 12 das 16 instâncias, e os outros dois CPAs encontram limites melhores que *LP* para 13 dos 16 casos. O CPA<sub>1</sub> obtém o mesmo limite que *SDP* em 3 instâncias, enquanto os outros dois CPAs obtêm o mesmo limite que *SDP* em 4 instâncias.

Outro ponto que vale a pena mencionar é que em 2 dos 4 casos nos quais o *SDP* falha, o CPAs obtêm limites melhores que os limites encontrados pelo *LP*.

O resultado mais expressivo apresentado na Tabela 5.1 está na instância I<sub>200,75,1</sub>, o gap obtido pelo *LP* é 16,83%, enquanto os gaps obtidos pelos CPAs são muito bons, 0,51% e 0,48%.

Para a única instância na qual o *SDP* obtém a solução ótima, ou seja, gap igual a 0%, os três CPAs também chegam ao mesmo resultado. Para esta mesma instância (I<sub>100,100,1</sub>), o *LP* encontra um gap de 3,46%.

Os gaps médios apresentados na última linha da Tabela 5.1 exemplificam a proximi-

dade dos limites encontrados pela relaxação  $SDP$  e pelos CPAs em 360 segundos.

Modificamos nosso algoritmo de plano de cortes a fim de utilizar, além dos cortes associados aos autovalores negativos, também os cortes disjuntivos. Porém os resultados obtidos até o momento melhoram, os resultados apresentados na Tabela 5.1, apenas na quarta casa decimal após a vírgula, ou seja, são muito modestos. Possivelmente isto ocorreu pois utilizamos apenas um ponto de corte,  $\theta = \frac{\eta_L(\varsigma) + \eta_U(\varsigma)}{2}$ , para dividir o intervalo  $[\eta_L(\varsigma), \eta_U(\varsigma)]$ .

Para ilustrar o comportamento do algoritmo proposto, incluímos no Apêndice A oito tabelas, A.1 até A.8, nas quais comparamos o CPA<sub>5</sub> as relaxações da literatura. Nas tabelas além de apresentarmos o Gap adicionamos também uma coluna para apresentar o tempo computacional, em segundos, gasto por cada método,  $t(s)$ . Observamos que das 80 instâncias testadas,  $SDP$  e o algoritmo de plano de cortes CPA<sub>5</sub> encontram resultados melhores ou iguais a relaxação  $LP$  em todas as instâncias. Destacamos os casos nos quais a relaxação  $SDP$  e o CPA<sub>5</sub> encontram resultados melhores que  $LP$ , instâncias:  $I_{50,100,1}$ ,  $I_{50,100,3}$ ,  $I_{50,100,4}$  e  $I_{50,100,8}$  apresentados na Tabela A.4.

Os dois resultados mais expressivos encontrados pelo plano de cortes são apresentados na tabela A.4,  $I_{50,100,3}$ , na qual o  $LP$  encontra gap igual a 17,75%, enquanto  $SDP$  e CPA<sub>5</sub> encontram gap igual a 0,38% e na Tabela A.7,  $I_{100,75,6}$  onde  $LP$  encontra gap igual a 28,98% enquanto  $SDP$  e CPA<sub>5</sub> encontram o ótimo. O algoritmo de plano de cortes e a relaxação  $SDP$  também encontram o valor ótimo nas instâncias  $I_{100,100,1}$  e  $I_{100,100,3}$ , apresentadas na Tabela A.8.

As Figura 5.1 e Figura 5.2 mostram o comportamento dos três algoritmos de plano de corte durante as iterações, de quatro instâncias. Os eixos horizontal e vertical nos gráficos correspondem, respectivamente, ao número de iterações e ao gap relativo. As duas linhas horizontais sobre os gráficos indicam os limites encontrados pelas duas relaxações ( $LP$ ) e ( $SDP$ ).

Na Figura 5.1 e na Figura 5.2 é possível observar que a medida em que aumenta o número de cortes adicionados as relaxações  $LP$  a cada iteração do algoritmo CPA, os limites calculados melhoram em menos iterações. Isto indica que os cortes adicionados são realmente eficazes.

Durante a análise das figuras fica claro que a convergência do CPA<sub>1</sub> é mais lenta que a dos outros dois CPAs. Após analisarmos as figuras e a Tabela 5.1 decidimos por excluir o CPA<sub>1</sub> de nossas pesquisas futuras e investiremos apenas no CPA<sub>5</sub> e no CPA<sub>10</sub> pois são os que obtêm melhores resultados. E além disso, observamos que é comum que estes dois últimos CPAs obtenham basicamente os mesmos limites em 360 segundos, por isso é importante analisarmos agora o tempo gasto em cada iteração destes algoritmos para identificar melhor o número de cortes a serem adicionados ao problema.

As Figuras 5.1 e 5.2 mostram que os limites obtidos com os CPAs estão sempre entre os encontrados pelo  $LP$  e pelo  $SDP$ . Além disso, a cada iteração, os limites dos CPAs



se aproximam dos limites *SDP* e tornam-se mais distante dos limites *LP*.

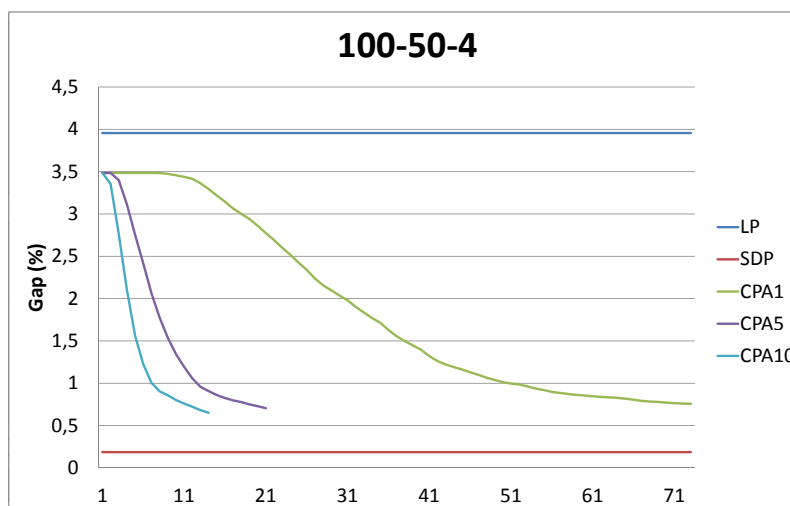
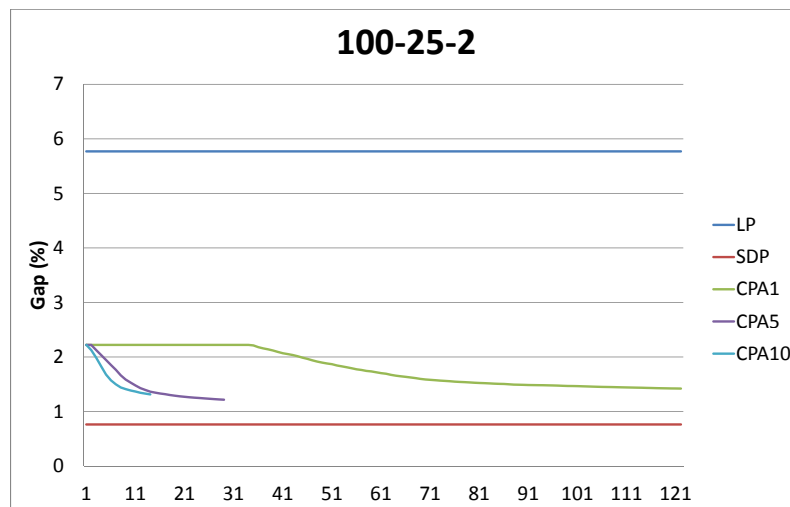


Figura 5.1: Limites encontrados durante a execução dos CPAs com densidade 25 e 50.

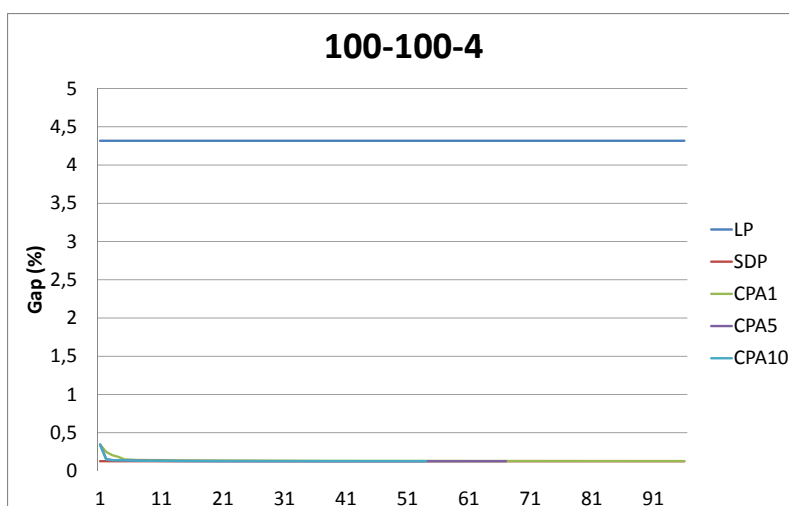
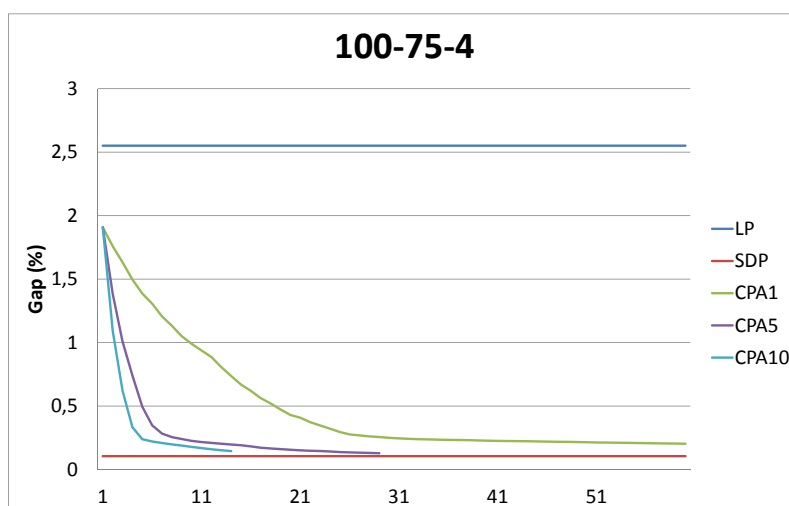


Figura 5.2: Limites encontrados durante a execução dos CPAs com densidade 75 e 100.

Nestes testes preliminares, nós permitimos que o CPA fosse executado por um tempo maior que  $SDP$ . Nosso objetivo era analisar o comportamento de convergência do algoritmo. Observamos também que o  $CPA_5$  e o  $CPA_{10}$ , várias vezes foram interrompidos por falta de memória. Para todos os outros casos o critério de parada era o limite do tempo de 360 segundos. Em nenhuma execução dos CPAs, obtivemos uma matriz  $Y$  semidefinida positiva.

## 5.2 Resultados utilizando a relaxação quadrática paramétrica convexa CQP

Nesta seção apresentamos os resultados principais desta tese. Nosso objetivo é agregar o algoritmo de pontos interiores, **IPM**, e os cortes para obter limites mais apertados para o **QKP**.

Para estes testes nosso código foi implementado em Matlab R2015a usando o toolbox CVX 2.1 [33]. Os testes foram executados em um computador com processador AMD FX- 6300, memória de 16GB e sistema operacional Linux Ubuntu 16.04. Para o método primal-dual de pontos interiores, **IPM**, utilizamos o Mosek [34] versão 8.0 para resolver a relaxação **CQP** $_{Q_p}$ , e para resolver os problemas de separação **MILP** $_1$  e **MILP** $_2$  utilizamos o Gurobi [36], versão 8.

Os testes computacionais têm como objetivo:

- verificar a eficácia do **IPM** descrito na Seção 3.1.2 em reduzir o limite superior enquanto otimiza a perturbação da matriz  $Q_p$ ,
- verificar o impacto das desigualdades válidas **SCI**, **CILS** e **SCILS**, quando adicionadas iterativamente para cortar a solução corrente da relaxação do **QKP**,
- calcular o limite superior e o limite inferior obtidos com o procedimento descrito no Algoritmo 5, e comparar estes, com a solução ótima das instâncias.

### 5.2.1 Estrutura algorítmica

Nesta seção descrevemos o algoritmo que utilizamos para encontrar os limites superiores para o **QKP**. O Algoritmo 5 apresenta a estrutura utilizada durante nossos testes computacionais. A cada iteração a matriz  $Q_p$ , da relaxação paramétrica, é atualizada e os inteiros  $m$  e  $\nu$  controlam, respectivamente, de quantas em quantas iterações os cortes serão adicionados a relaxação e o número máximo de cortes adicionais.

Os dados de entrada, para a primeira iteração do **IPM**, descritos no Algoritmo 2, onde  $k = 0$ , são:  $B_0 = I$ ,  $\mu^0 = 1$ . Partimos de uma matriz  $Q_p^0$ , tal que  $Q - Q_p^0$  é definida negativa. Resolvendo o problema **CQP** $_{Q_p}$ , com  $Q_p := Q_p^0$ , encontramos a solução ótima  $x(Q_p^0)$ ,  $X(Q_p^0)$ , e definimos  $\nabla p_{\text{CQP}}^*(Q_p^0) := X(Q_p^0) - x(Q_p^0)x(Q_p^0)^T$ . O fato de  $Z^0$  e  $\Lambda^0$  serem definidas positivas é assegurado pois

$$\Lambda^0 := \nabla p_{\text{CQP}}^*(Q_p^0) + (2|\lambda_{\min}(\nabla p_{\text{CQP}}^*(Q_p^0))| + .1)I$$

Nas Tabelas apresentamos o método aplicado para calcular o limite superior na primeira coluna. Nas outras colunas apresentamos para cada método os resultados para

---

**Algoritmo 5:** Estrutura algoritmica

---

Entrada:  $Q \in \mathbb{S}^n$ ,  $max.n_{cuts}$ ,  $\epsilon$ ,  $m$  e  $\nu$ .

$k := 0$ ,  $B_0 := I$ ,  $\mu^0 := 1$ .

Sejam  $\lambda_i(Q)$ ,  $v_i$  o  $i^{th}$  maior autovalor e o autovetor correspondentes de  $Q$ .

$Q_n := \sum_{i=1}^n (-|\lambda_i(Q)| - 1)v_i v_i'$ .

$Q_p^0 := Q - Q_n$ .

Resolva a relaxação **CQP** $_{Q_p}$  (em (3.5)), com  $Q_p := Q_p^0$ , e obtenha  $x(Q_p^0)$ ,  $X(Q_p^0)$ .

$\nabla p_{\text{CQP}}^*(Q_p^0) := X(Q_p^0) - x(Q_p^0)x(Q_p^0)^T$ .

$Z^0 := Q_p^0 - Q$ .

$\Lambda^0 := \nabla p_{\text{CQP}}^*(Q_p^0) + (2|\lambda_{\min}(\nabla p_{\text{CQP}}^*(Q_p^0))| + .1)I$ .

**Enquanto** (o critério de parada não é violado)

Execute o Algoritmo 2, onde  $Q_p^{k+1}$  é obtido e a relaxação **CQP** $_{Q_p}$  com  $Q_p := Q_p^{k+1}$  é resolvida. Seja  $(x(Q_p^{k+1}), X(Q_p^{k+1}))$  a solução ótima.

$upper.bound^{k+1} := p_{\text{CQP}}^*(Q_p^{k+1})$ .

Execute o Algoritmo 3, onde  $\hat{x}$  é obtido.

$lower.bound^{k+1} := \hat{x}^T Q \hat{x}$ .

**Se**  $k \bmod m == 0$

Resolva o problema da mochila **KP** (3.23) e obtenha o corte **SCI** (3.24).

Adicione  $\max\{n, max.n_{cuts}\}$  cortes **SCI** que mais violem  $(x(Q_p^{k+1}), X(Q_p^{k+1}))$ , ao problema **CQP** $_{Q_p}$ .

$n_{cuts} := 0$ .

**Enquanto** ( $n_{cuts} < max.n_{cuts}$  e **MILP** $_1$  é viável)

Resolva **MILP** $_1$  e adicione **CI** e **CILS**

encontrados ao **CQP** $_{Q_p}$ .

Adicione o corte (3.27) ao **MILP** $_1$ .

$n_{cuts} := n_{cuts} + 1$ .

**Fim Enquanto**

$n_{cuts} := 0$ .

**Enquanto** ( $n_{cuts} < max.n_{cuts}$  e **MILP** $_2$  é viável)

Resolva o **MILP** $_2$  e adicione **CI** e **SCILS**

encontrados ao **CQP** $_{Q_p}$ .

Adicione o corte (3.30) ao **MILP** $_2$ .

$n_{cuts} := n_{cuts} + 1$ .

**Fim Enquanto**

**Fim Se**

$k := k + 1$ .

**Fim Enquanto**

Saída: Limite Superior  $upper.bound^k$ , Limite Inferior  $lower.bound^k$ , e solução viável  $\hat{x}$  para o **QKP**.

---

- Gap de otimalidade  
(OptGap (%)):= ((upper bound - opt)/opt) × 100, onde opt é o valor da solução ótima;
- Tempo médio em segundos (Tempo (s));
- Gap de dualidade  
(DuGap) := (upper bound - lower bound)/(|upper bound| + (|lower bound|)), onde o limite inferior, *lower bound*, é calculado como descrito na Seção 3.4;
- Número de iterações (Iter);
- Número de cortes adicionados a relaxação (Cortes);
- Tempo médio em segundos gasto para obter os cortes **CILS** e **SCILS** (Tempo<sub>MILP</sub> (s)).

### 5.2.2 A influência dos cortes

Nesta seção apresentamos os resultados obtidos ao adicionarmos os cortes a relaxação linear do problema. Nosso objetivo é reduzir o limite superior e para isso a cada iteração as novas desigualdades, apresentadas nesta tese, são adicionadas a relaxação do **QKP** com o intuito de cortar a solução corrente e desta forma reduzir o limite superior.

Nas Tabelas 5.2–5.3 apresentamos resultados para 10 instâncias com  $n = 10$  itens. Os resultados na Tabela 5.2 tem como objetivo mostrar o impacto dos cortes propostos.

Para obter os resultados apresentados na Tabela 5.2 permitimos que, no máximo, 15 cortes de cada tipo, **SCI**, **CILS** e **SCILS**, sejam adicionados a seguinte relaxação linear:

$$\begin{aligned}
 & \text{maximize} && \text{trace}(QX) \\
 & \text{sujeito a:} && \sum_{j=1}^n w_j x_j \leq c, \\
 (\tilde{\text{LPR}}) & && 0 \leq X_{ij} \leq 1, \quad \forall i, j \in N \\
 & && 0 \leq x_i \leq 1, \quad \forall i \in N \\
 & && X \in \mathbb{S}^n.
 \end{aligned} \tag{5.2}$$

Na primeira linha da Tabela 5.2, os resultados correspondem a solução da relaxação linear **LPR** sem que nenhum corte seja adicionado. Em **SCI**<sub>1</sub>, a cada iteração adicionamos ao problema **LPR**, apenas o corte mais violado dos  $n$  cortes em **SCI**, e em **SCI** adicionamos todos os  $n$  cortes. Em **CILS** e **SCILS**, resolvemos os problemas de separação, **MILP**<sub>1</sub> e **MILP**<sub>2</sub>, respectivamente, para encontrar o corte mais violado de cada tipo. A última linha da tabela (ALL) corresponde ao resultado obtido quando adicionamos todos os  $n$  cortes em **SCI**, e um corte de cada tipo **CILS** e **SCILS**.

Nestes testes iniciais executamos no máximo 50 iterações e o algoritmo pára quando nenhum corte distinto é encontrado para ser adicionado a relaxação.

Conforme discutido anteriormente, na seção 3.3, para encontrar os cortes **CILS** e **SCILS** os problemas de separação não precisam ser resolvidos até a otimalidade, desta forma utilizamos um limite de tempo de 3 segundos para resolver os problemas de separação. Entretanto é importante ressaltar que nestes testes, onde  $n = 10$  itens, este limite de tempo foi suficiente para resolver todos os problemas até a otimalidade.

Método	OptGap (%)	Tempo (s)	DuGap	Iter	Cortes	Tempo <sub>MILP</sub> (s)
L $\tilde{P}R$	38.082	0.35	0.620	1.0		
SCI <sub>1</sub>	36.703	32.38	0.343	1.1	28.4	
SCI	10.036	39.98	0.058	3.0	364.1	
CILS	19.719	9.00	0.293	2.7	82.2	6.91
SCILS	9.121	266.81	0.250	50.0	794.3	198.12
ALL	3.315	315.82	0.016	28.3	646.6	264.91

Tabela 5.2: Impacto dos cortes adicionados ao L $\tilde{P}R$  (10 instâncias,  $n = 10$ ).

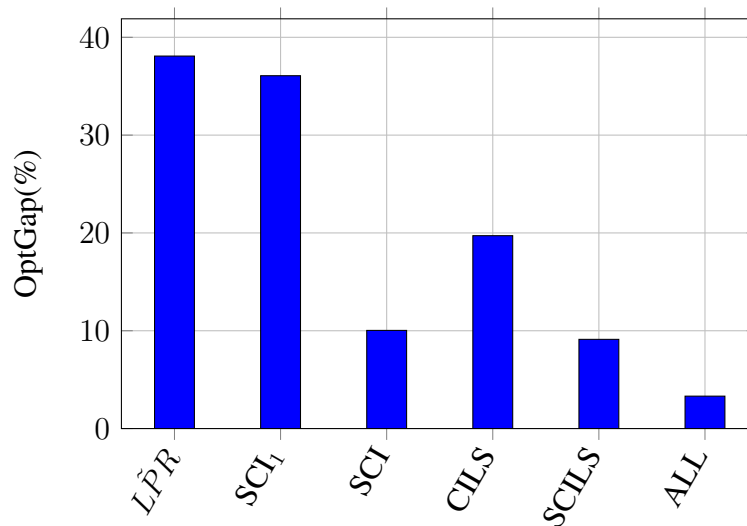


Figura 5.3: Média dos gaps de otimalidade apresentados na Tabela 5.2.

A Figura 5.3 apresenta os gaps de otimalidade da Tabela 5.2. Observa-se um comportamento inverso entre a qualidade dos cortes e o tempo computacional necessário para encontra-los. Considerando um corte único de cada tipo, o **SCILS** é o corte mais forte (OptGap = 9.121%), porém o tempo computacional necessário para encontrá-lo é maior que os tempos necessários para o cálculo de **CILS** e **SCI**. Entretanto o tempo pode ser reduzido se o problema de separação for resolvido de forma heurística. Note que usar todos os cortes juntos possibilita encontrar melhores limites superiores que usar cada tipo em separado (OptGap = 3.315%).

### 5.2.3 Desigualdades válidas agregadas ao método de pontos interiores

Nesta seção, apresentamos os testes principais desta tese que tem como objetivo agregar os novos cortes apresentados na seção 3.3 ao **IPM**, que a cada iteração atualiza a decomposição inicial da matriz  $Q$  e encontrando melhores limites superiores para o **QKP**.

Para melhorar os resultados, consideramos na relaxação quadrática paramétrica a desigualdade válida obtida ao multiplicar a restrição de capacidade por cada variável não negativa  $x_i$  e as desigualdades válidas derivadas de  $x_i \in \{0, 1\}$ . Iniciamos o algoritmo com a relaxação abaixo.

$$\begin{aligned}
 & \text{maximize} && x^T(Q - Q_p^0)x + \text{trace}(Q_p^0 X) \\
 & \text{sujeito a:} && \sum_{j=1}^n w_j x_j \leq c, \\
 & && \sum_{j=1}^n w_j X_{ij} \leq c X_{ii}, && \forall i \in N \\
 (\text{QPR}) & && X_{ii} = x_i, && \forall i \in N && (5.3) \\
 & && 0 \leq X_{ij} \leq 1, && \forall i, j \in N \\
 & && 0 \leq x_i \leq 1, && \forall i \in N \\
 & && X \in \mathbb{S}^n.
 \end{aligned}$$

A fim de avaliar a influência da decomposição inicial da matriz  $Q$  sobre o comportamento do **IPM**, consideramos duas decomposições iniciais diferentes para  $Q$ . Nos dois casos, calculamos a decomposição dos autovalores de  $Q$ , onde  $Q = \sum_{i=1}^n \lambda_i v_i v_i'$ .

- Na primeira decomposição, consideramos  $Q_n := \sum_{i=1}^n (-|\lambda_i| - 1)v_i v_i'$ , e  $Q_p^0 := Q - Q_n/2$ . Nas tabelas e figuras abaixo, nos referimos a esta matriz inicial  $Q_p^0$  como  $Q_p^a$ .
- Na segunda decomposição, consideramos  $Q_n := \sum_{i=1}^n (\min\{\lambda_i, -10^{-6}\})v_i v_i'$ , e  $Q_p^0 := Q - Q_n/2$ . Nos referimos a esta matriz inicial como  $Q_p^b$ .

Implementamos o **IPM** atualizando a matriz hessiana  $B$  usando o procedimento BFGS descrito na seção 3.1.2 e também consideramos uma aproximação simples na qual  $B = I$  em todas as iterações. Na Tabela 5.3 apresentamos os resultados obtidos utilizando estes dois procedimentos e as duas decomposições iniciais da matriz  $Q$  descritas anteriormente.

Nas duas primeiras linhas da tabela, os resultados são obtidos pela solução da relaxação quadrática inicial, com a composição de  $Q$  e sem nenhum corte. Nas próximas quatro linhas da tabela, os resultados são obtidos com a aplicação do **IPM**, e sem que nenhum corte seja adicionado a relaxação. Nas últimas quatro linhas, o resultado é obtido, com a inclusão dos cortes a relaxação. Os cortes são adicionados a cada  $m = 10$  iterações do **IPM** e o número de cortes adicionados a cada iteração é  $n$  **SCI**, 5 **CILS** e 5 **SCILS**.

Note que, quando resolvemos cada problema de separação **MILP**, além do cortes **CILS** ou dos **SCILS**, nós também encontramos as desigualdades de cobertura, **CI**. Verifi-

camos se estas desigualdades já foram adicionadas ao problema e, caso contrário, também adicionamos estas ao problema.

Nestes testes permitimos que o algoritmo execute no máximo 150 iterações e o limite de tempo foi fixado em 900 segundos. Caso o gap de dualidade, DuGap, seja suficientemente pequeno o algoritmo pára.

Método	OptGap (%)	Tempo (s)	DuGap	Iter	Cortes	Tempo <sub>MILP</sub> (s)
$Q_p^a$ ,QPR	21.640	0.79	0.138	1.0		
$Q_p^b$ ,QPR	12.276	0.83	0.076	1.0		
$Q_p^a$ ,I	7.242	26.80	0.042	150.0		
$Q_p^b$ ,I	7.633	25.81	0.055	150.0		
$Q_p^a$ ,BFGS	7.091	27.41	0.041	150.0		
$Q_p^b$ ,BFGS	7.094	25.94	0.041	150.0		
$Q_p^a$ ,I,Cuts	0.863	87.09	0.009	144.5	104	42.13
$Q_p^b$ ,I,Cuts	1.516	77.60	0.012	132.8	97.6	38.37
$Q_p^a$ ,BFGS,Cuts	0.639	46.13	0.008	77.7	275.4	23.57
$Q_p^b$ ,BFGS,Cuts	0.640	56.32	0.008	98.7	80.7	29.24

Tabela 5.3: Média dos resultados para 10 instâncias ( $n = 10$ ).

Na Tabela 5.3 os melhores limites são obtidos quando usamos o **IPM** atualizando a matriz hessiana usando BFGS, e adicionando os cortes. Com relação ao ponto inicial,  $Q_p^a$  encontra melhores limites no geral, mas o tempo gasto é ligeiramente maior que ao utilizamos  $Q_p^b$ . Outros detalhes dos resultados obtidos para cada instância podem ser encontrados no Apêndice A nas Tabelas A.9 até A.18. A Figura 5.4 ilustra os gaps de otimalidade da Tabela 5.3

Destacamos que ao adicionarmos os cortes ao **IPM** foi possível obter o valor ótimo em algumas das instâncias testadas. Sem utilizamos a atualização da matriz  $B$  e adicionando os cortes para  $Q_p^a$ ,I,Cuts o algoritmo encontrou o ótimo em 6 das 10 instâncias, Tabela A.15. E para  $Q_p^b$ ,I,Cuts o ótimo é encontrado em 4 das 10 instâncias apresentadas na Tabela A.16. Nas Tabelas A.17 e A.18 aliamos os cortes à atualização da matriz  $B$  e tanto para a aproximação  $Q_p^a$  e  $Q_p^b$  o ótimo é encontrado em 8 das 10 instâncias testadas.

Para os testes apresentados na Tabela 5.4 ( $n = 30$ ) permitimos que o algoritmo execute no máximo 250 iterações e o limite de tempo foi fixado em 900 segundos. Novamente, caso o gap de dualidade, DuGap, seja suficientemente pequeno o algoritmo pára.

Na Tabela 5.4 observamos que o algoritmo combinado com adição dos cortes apresenta resultados próximos do ótimo na média. É importante destacar que, com o aumento dos itens da mochila, é possível concluir que a parte mais cara do algoritmo é a busca pelos cortes, ou seja, o tempo gasto nos problemas de separação **MILP**<sub>1</sub> e **MILP**<sub>2</sub>. A Figura 5.5 ilustra as médias dos gaps de otimalidade apresentados na Tabela 5.4.



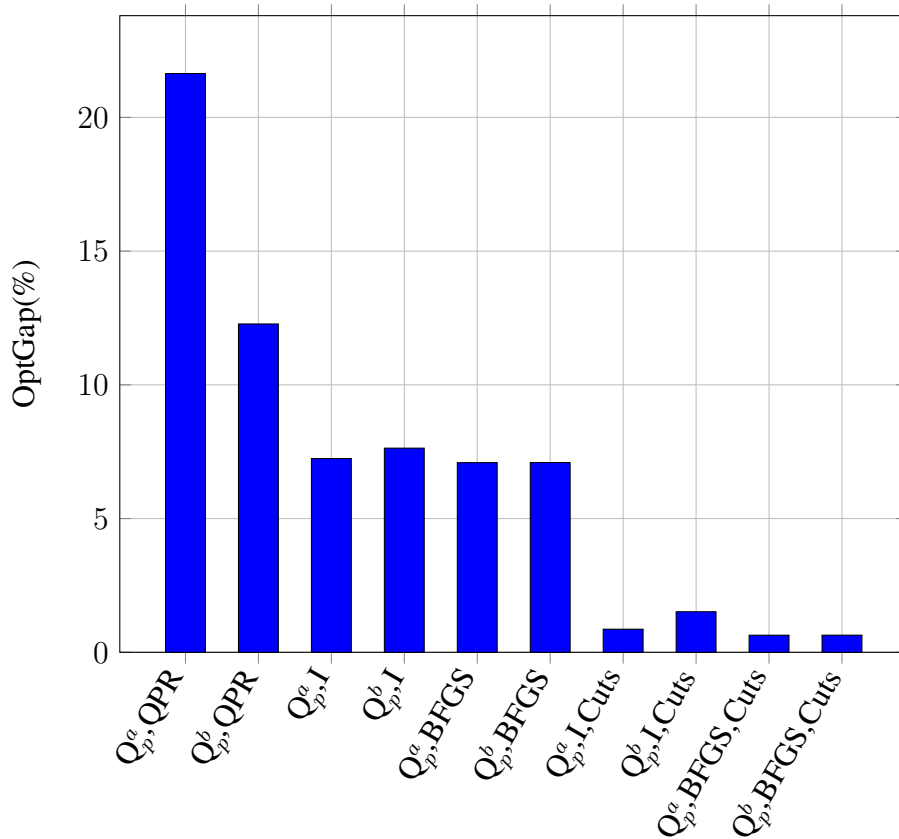


Figura 5.4: Média dos gaps de otimalidade apresentados na Tabela 5.3.

No Apêndice A nas Tabelas A.19 até A.24 apresentamos em detalhes os resultados obtidos para as instâncias com  $n = 30$  itens. Destacamos que para as duas escolhas de  $Q$  sem a adição dos cortes o ótimo é encontrado em 2 das 10 instâncias testadas, Tabelas A.21 e A.22. E com a adição dos cortes é possível encontrar a solução ótima em 6 das 10 instâncias, Tabelas A.23 e A.24.

Método	OptGap (%)	Tempo (s)	DuGap	Iter	Cortes	Tempo <sub>MILP</sub> (s)
$Q_p^a, QPR$	22.840	0.14	0.845	1		
$Q_p^b, QPR$	10.618	0.07	0.779	1		
$Q_p^a, BFGS$	3.064	114.82	0.029	214.6		
$Q_p^b, BFGS$	3.064	110.74	0.029	209.4		
$Q_p^a, BFGS, Cuts$	0.278	493.96	0.005	147.2	2492.50	344.47
$Q_p^b, BFGS, Cuts$	0.300	529.63	0.005	160.6	2757.3	372.16

Tabela 5.4: Média dos resultados para 10 instâncias ( $n = 30$ ).

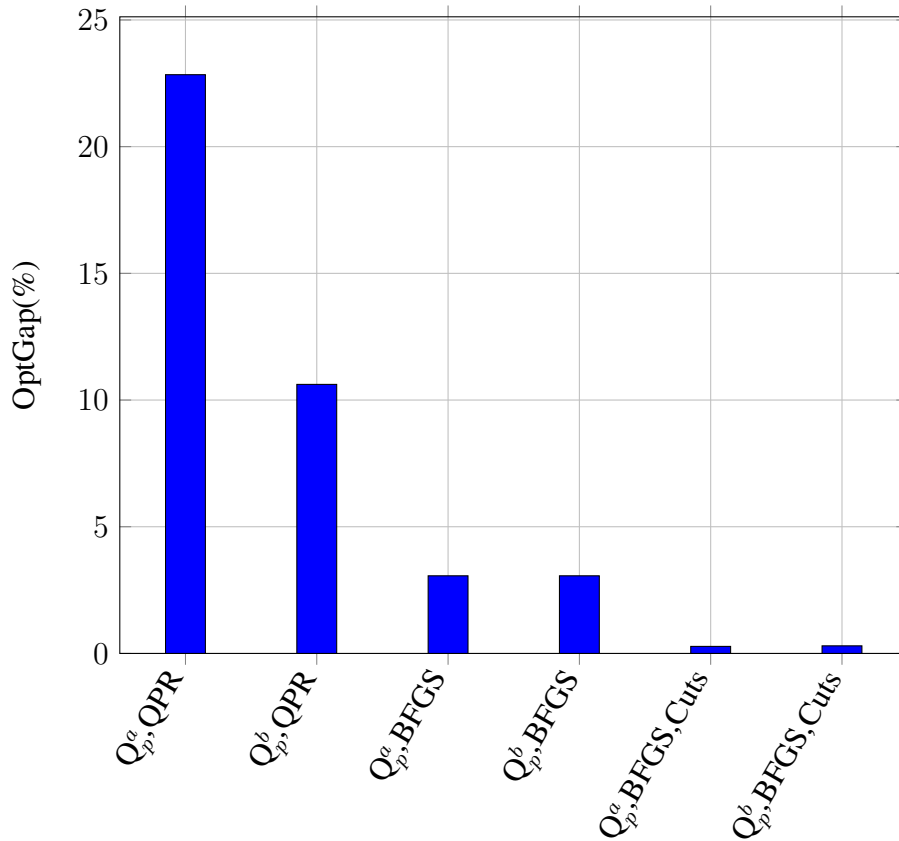


Figura 5.5: Média dos gaps para as tabelas com  $n = 30$  itens, 5.4

Na Tabela 5.5 apresentamos a evolução dos resultados ao ampliarmos o número de itens na mochila,  $n$ . Para estes testes, utilizamos no máximo 500 iterações e o tempo máximo de 2700 segundos, e caso o DuGap seja suficientemente pequeno o algoritmo pára.

Método	OptGap (%)	Tempo (s)	DuGap	Iter	Cortes	Tempo <sub>MILP</sub> (s)
$Q_p^a, \text{QPR}$	21.136	0.87	0.113	1.0		
$Q_p^b, \text{QPR}$	9.460	0.86	0.056	1.0		
$Q_p^a, \text{BFGS}$	1.345	2732.70	0.015	424.0		
$Q_p^b, \text{BFGS}$	1.345	2713.26	0.015	430.0		
$Q_p^a, \text{BFGS, Cuts}$	0.078	2216.81	0.001	168.6	241.2	858.51
$Q_p^b, \text{BFGS, Cuts}$	0.076	1882.68	0.001	145.0	199.4	731.55

Tabela 5.5: Média dos resultados para 5 instâncias ( $n = 50$ ).

Novamente o **IPM**, com ou sem a adição dos cortes, reduz o limite superior inicial dado pela solução da relaxação quadrática. A influência da decomposição inicial da matriz  $Q$  nos limites obtidos pelo **IPM** não é relevante, mas a convergência continua mais rápida com a decomposição inicial  $Q_p^b$ . É interessante observar que, apesar de a solução dos

problemas de separação, **MILP**, serem computacionalmente caras, o tempo gasto nelas é compensado pela rápida convergência do algoritmo e por encontrarem melhores limites.

No Apêndice A nas Tabelas A.25 até A.30 apresentamos em detalhes os resultados obtidos para as instâncias com  $n = 50$  itens. Nas tabelas A.29 e A.30 apresentamos os resultados obtidos ao adicionarmos os cortes ao **IPM** com a atualização da matriz  $B$ . Observamos que para as 5 instâncias testadas o algoritmo encontra o ótimo em 3 instâncias independente da decomposição inicial escolhida para a matriz  $Q$ .

É importante destacar que nas 5 instâncias testadas, para  $n = 50$  itens, o algoritmo de pontos interiores e os cortes, Tabelas A.27 - A.30, encontram melhores resultados que os gaps obtidos pelo CPAs e pela relaxação  $SDP$ , Tabela A.4. Entretanto, o tempo computacional gasto durante a busca pelos cortes **CILS** e **SCILS** é muito grande.

# Capítulo 6

## Conclusões e trabalhos futuros

Nesta tese apresentamos um algoritmo de plano de cortes, que reduz o limite superior do problema da mochila quadrática **QKP**. A relaxação inicial do problema é dada por um problema quadrático convexo paramétrico, onde a Hessiana  $Q$  da função objetivo do **QKP** é perturbada pela matriz  $Q_p$ , tal que  $Q - Q_p \preceq 0$ . Buscando o melhor limite possível, o termo côncavo  $x^T(Q - Q_p)x$  é mantido na função objetivo da relaxação e a parte restante, dada por  $x^T Q_p x$  é linearizada através do procedimento que considera o problema no espaço de matrizes simétricas definidas por  $X := xx^T$ .

Apresentamos um algoritmo de pontos interiores **IPM** que atualiza a perturbação  $Q_p$  a cada iteração do algoritmo de plano de cortes, visando reduzir o limite superior dado pela relaxação. Também apresentamos uma nova classe de cortes que são definidos na variável  $X$  e derivados das desigualdades de cobertura e das restrições binárias.

Em nossos testes mostramos que o **IPM** e os cortes são eficazes em melhorar o limite superior para o **QKP**. Enfatizamos que este procedimento também pode ser aplicado para os problemas quadráticos indefinidos mais gerais.

Finalmente, mostramos que, se a restrição semidefinida positiva

$$X - xx^T \succeq 0 \tag{6.1}$$

for introduzida na relaxação do **QKP**, ou em qualquer outro problema quadrático indefinido (maximizar a função objetivo), então a decomposição da função objetivo que leva a uma relaxação quadrática convexa, na qual a parte côncava perturbada na função objetivo é mantida e a parte restante é linearizada, não é eficaz. Neste caso, o melhor limite é sempre alcançado quando toda a função objetivo é linearizada, isto é, quando a perturbação  $Q_p$  é igual a  $Q$ . Esta observação também se relaciona à conhecida diferença de convexas, *difference of convex*, DC, decomposição de quadráticos indefinidos que tem sido utilizada na literatura para gerar limites para problemas quadráticos indefinidos. Mais uma vez, caso a restrição 6.1 seja adicionada a relaxação, a decomposição DC não é mais efetiva e a relaxação linear **SDP** alternativa encontra os melhores limites possíveis.

Neste trabalho também propusemos a aplicação de um algoritmo de plano de corte que resolve de forma iterativa relaxações de programação linear fortes do problema, adicionando desigualdades válidas bem conhecidas na literatura como desigualdades RLT e cortes *SDP*. Na literatura ideias semelhantes têm sido propostas para problemas quadráticos não convexos. Em nossa pesquisa, nós adaptamos as ideias para melhor atender ao **QKP**, e comparamos, através de resultados computacionais, diferentes versões do algoritmo de plano de corte. Então concluímos que a metodologia proposta oferece resultados promissores para o problema da mochila quadrática.

Nossos resultados iniciais mostram que os limites calculados pelo CPA para algumas instâncias são muito melhores que os encontrados pela relaxação linear simples *LP*, e podem ser calculados com mais eficiência que os limites obtidos utilizando da relaxação *SDP* mais forte, quando o número de variáveis aumenta.

Nosso próximo objetivo em nossa pesquisa é definir um critério de parada para o CPAs, de tal forma que eles possam obter em média um limitante melhor que *LP*, mas em um tempo mais curto que o *SDP*. Ressaltamos que o tempo de processamento do algoritmo CPA é ainda muito grande, se comparado com outras relaxações de programação linear. No entanto, nosso trabalho futuro consiste em aplicar algumas técnicas bem conhecidas para reduzi-lo, como, por exemplo, considerar as desigualdades RLT a serem adicionados a relaxação original, com um algoritmo de separação, assim como é feito para os cortes *SDP*. E um procedimento para eliminar da relaxação as restrições RLTs inativas na solução ótima, a cada iteração do CPA.

Outro ponto que precisa ser explorado em nossas pesquisas futuras é o potencial dos cortes disjuntivos aliados ao **IPM**. Em nossos resultados iniciais, aplicando os cortes disjuntivos, dividimos o intervalo de interesse ao meio. Os autores em [32] mostram que é possível subdividir o intervalo em  $k$  intervalos.

Esquemáticamente, a pesquisa desenvolvida se apresenta como:

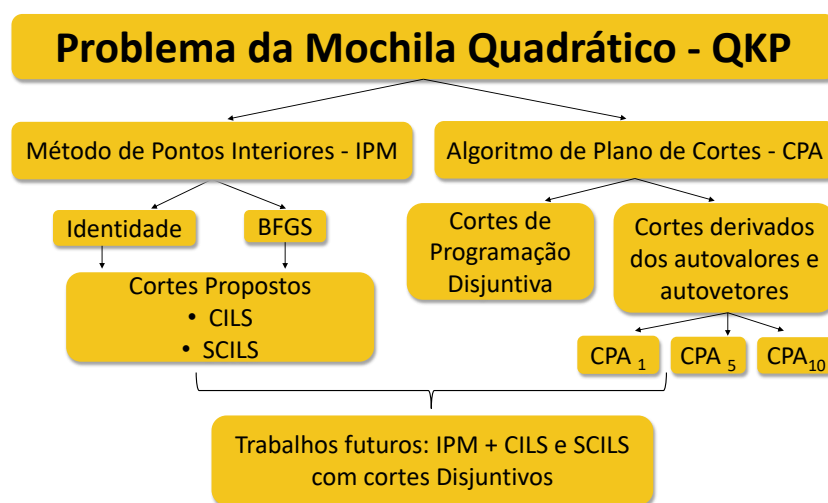


Figura 6.1: Resumo da pesquisa desenvolvida e trabalhos futuros.

# Referências Bibliográficas

- [1] GALLO, G., HAMMER, P. L., SIMEONE, B. “Quadratic knapsack problems”. In: Padberg, M. W. (Ed.), *Combinatorial Optimization*, pp. 132–149, Berlin, Heidelberg, Springer Berlin Heidelberg, 1980. ISBN: 978-3-642-00802-3. doi: 10.1007/BFb0120892. Disponível em: <<https://doi.org/10.1007/BFb0120892>>.
- [2] WITZGALL, C. “Mathematical methods of site selection for Electronic Message Systems (EMS)”, *NASA STI/Recon Technical Report N*, v. 76, jun. 1975.
- [3] RHYS, J. M. W. “A Selection Problem of Shared Fixed Costs and Network Flows”, *Management Science*, v. 17, n. 3, pp. 200–207, 1970. ISSN: 00251909, 15265501. Disponível em: <<http://www.jstor.org/stable/2629089>>.
- [4] CHAILLOU, P., HANSEN, P., MAHIEU, Y. “Best network flow bounds for the quadratic knapsack problem”. In: Simeone, B. (Ed.), *Combinatorial Optimization: Lectures given at the 3rd Session of the Centro Internazionale Matematico Estivo (C.I.M.E.) held at Como, Italy, August 25–September 2, 1986*, pp. 225–235, Berlin, Heidelberg, Springer Berlin Heidelberg, 1989. ISBN: 978-3-540-46810-3. doi: 10.1007/BFb0083467. Disponível em: <<https://doi.org/10.1007/BFb0083467>>.
- [5] BILLIONNET, A., CALMELS, F. “Linear programming for the 0-1 quadratic knapsack problem”, *European Journal of Operational Research*, v. 92, n. 2, pp. 310–325, 1996. ISSN: 0377-2217. doi: [http://dx.doi.org/10.1016/0377-2217\(94\)00229-0](http://dx.doi.org/10.1016/0377-2217(94)00229-0). Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221794002290>>.
- [6] CAPRARA, A., PISINGER, D., TOTH, P. “Exact Solution of the Quadratic Knapsack Problem”, *INFORMS Journal on Computing*, v. 11, n. 2, pp. 125–137, 1999. doi: 10.1287/ijoc.11.2.125. Disponível em: <<http://dx.doi.org/10.1287/ijoc.11.2.125>>.

- [7] BILLIONNET, A., FAYE, A., SOUTIF, É. “A new upper bound for the 0-1 quadratic knapsack problem”, *European Journal of Operational Research*, v. 112, n. 3, pp. 664–672, 1999. doi: 10.1016/S0377-2217(97)00414-1. Disponível em: <[http://dx.doi.org/10.1016/S0377-2217\(97\)00414-1](http://dx.doi.org/10.1016/S0377-2217(97)00414-1)>.
- [8] HELMBERG, C., RENDL, F., WEISMANTEL, R. “Quadratic knapsack relaxations using cutting planes and semidefinite programming”. In: Cunningham, W., McCormick, S., Queyranne, M. (Eds.), *Integer Programming and Combinatorial Optimization*, v. 1084, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 175–189, 1996. ISBN: 978-3-540-61310-7. doi: 10.1007/3-540-61310-2\_14. Disponível em: <[http://dx.doi.org/10.1007/3-540-61310-2\\_14](http://dx.doi.org/10.1007/3-540-61310-2_14)>.
- [9] C. HELMBERG, F. RENDL AND R. WEISMANTEL. “A semidefinite programming approach to the quadratic knapsack problem”, *J. Comb. Optim.*, v. 4, n. 2, pp. 197 – 215, 2000.
- [10] HELMBERG, C., RENDL, F., WEISMANTEL, R. “A Semidefinite Programming Approach to the Quadratic Knapsack Problem”, *Journal of Combinatorial Optimization*, v. 4, n. 2, pp. 197–215, 2000. ISSN: 1382-6905. doi: 10.1023/A:1009898604624.
- [11] PICARD, J. C., RATLIFF, H. D. “Minimum cuts and related problems”, *Networks*, v. 5, n. 4, pp. 357–370, 1975. ISSN: 1097-0037. doi: 10.1002/net.3230050405. Disponível em: <<http://dx.doi.org/10.1002/net.3230050405>>.
- [12] MICHELON, P., VEILLEUX, L. “Lagrangean methods for the 0-1 Quadratic Knapsack Problem”, *European Journal of Operational Research*, v. 92, n. 2, pp. 326 – 341, 1996. ISSN: 0377-2217. doi: [http://dx.doi.org/10.1016/0377-2217\(94\)00286-X](http://dx.doi.org/10.1016/0377-2217(94)00286-X). Disponível em: <<http://www.sciencedirect.com/science/article/pii/037722179400286X>>.
- [13] MCCORMICK, G. P. “Computability of global solutions to factorable nonconvex programs: part I Convex underestimating problems”, *Mathematical Programming*, v. 10, pp. 147–175, 1976.
- [14] PISINGER, D. “The quadratic knapsack problem-a survey”, *Discrete Applied Mathematics*, v. 155, pp. 623–648, 2007. ISSN: 0166218X. doi: 10.1016/j.dam.2006.08.007.

- [15] BALAS, E. “Facets of the knapsack polytope”, *Mathematical Programming*, v. 8, n. 1, pp. 146–164, Dec 1975. ISSN: 1436-4646. doi: 10.1007/BF01580440. Disponível em: <<https://doi.org/10.1007/BF01580440>>.
- [16] WOLSEY, L. “Faces for a linear inequality in 0–1 variables”, *Mathematical Programming*, v. 8, n. 1, pp. 165–178, Dec 1975. ISSN: 1436-4646. doi: 10.1007/BF01580441. Disponível em: <<https://doi.org/10.1007/BF01580441>>.
- [17] NEMHAUSER, G., WOLSEY, L. *Integer and Combinatorial Optimization*. New York, NY, John Wiley & Sons, 1988.
- [18] HAMMER, P., JOHNSON, E., PELED, U. N. “Facet of regular 0–1 polytopes”, *Mathematical Programming*, v. 8, n. 1, pp. 179–206, Dec 1975. ISSN: 1436-4646. doi: 10.1007/BF01580442. Disponível em: <<https://doi.org/10.1007/BF01580442>>.
- [19] BALAS, E., ZEMEL, E. “Facets of the Knapsack Polytope from Minimal Covers”, *SIAM Journal on Applied Mathematics*, v. 34, n. 1, pp. 119–148, 1978. ISSN: 00361399. Disponível em: <<http://www.jstor.org/stable/2100862>>.
- [20] ATAMTÜRK, A. “Cover and Pack Inequalities for (Mixed) Integer Programming”, *Annals of Operations Research*, v. 139, n. 1, pp. 21–38, Oct 2005. ISSN: 1572-9338. doi: 10.1007/s10479-005-3442-1. Disponível em: <<https://doi.org/10.1007/s10479-005-3442-1>>.
- [21] ZEMEL, E. “Easily Computable Facets of the Knapsack Polytope”, *Mathematics of Operations Research*, v. 14, n. 4, pp. 760–764, 1989. doi: 10.1287/moor.14.4.760. Disponível em: <<https://doi.org/10.1287/moor.14.4.760>>.
- [22] GU, Z., NEMHAUSER, G., SAVELSBERGH, M. “Lifted Cover Inequalities for 0-1 Integer Programs: Computation”, *INFORMS Journal on Computing*, v. 10, n. 4, pp. 427–437, 1998. doi: 10.1287/ijoc.10.4.427. Disponível em: <<https://doi.org/10.1287/ijoc.10.4.427>>.
- [23] WOLSEY, L. *Integer programming*. New York, NY, USA, Wiley-Interscience, 1998.
- [24] FIACCO, A. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, v. 165, *Mathematics in Science and Engineering*. Academic Press, 1983.



- [25] HOGAN, W. “The continuity of the perturbation function of a convex program”, *Operat. Research*, v. 21, pp. 351–352, 1973.
- [26] DANSKIN, J. “The Theory of Max-Min, with Applications”, *SIAM Journal on Applied Mathematics*, v. 14, n. 4, pp. 641–664, 1966. doi: 10.1137/0114053. Disponível em: <<https://doi.org/10.1137/0114053>>.
- [27] LEWIS, A., OVERTON, M. “Nonsmooth optimization via quasi-Newton methods”, *Mathematical Programming*, v. 141, n. 1, pp. 135–163, Oct 2013. ISSN: 1436-4646. doi: 10.1007/s10107-012-0514-2. Disponível em: <<https://doi.org/10.1007/s10107-012-0514-2>>.
- [28] TURLACH, B. A., WRIGHT, S. J. “Quadratic programming”, *Wiley Interdisciplinary Reviews: Computational Statistics*, v. 7, n. 2, pp. 153–159. doi: 10.1002/wics.1344. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.1344>>.
- [29] JOLLIFFE, I. *Principal component analysis*. 2nd ed. Springer, Berlin, 2010.
- [30] SHERALI, H. D., FRATICELLI, B. M. P. “Enhancing RLT relaxations via a new class of semidefinite cuts”, *Journal of Global Optimization*, v. 22, n. 1, pp. 233–261, Jan 2002. ISSN: 1573-2916. doi: 10.1023/A:1013819515732. Disponível em: <<https://doi.org/10.1023/A:1013819515732>>.
- [31] BALAS, E. “Disjunctive programming: Properties of the convex hull of feasible points”, *Discrete Applied Mathematics*, v. 89, n. 1–3, pp. 3 – 44, 1998. ISSN: 0166-218X. doi: [http://dx.doi.org/10.1016/S0166-218X\(98\)00136-X](http://dx.doi.org/10.1016/S0166-218X(98)00136-X). Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0166218X9800136X>>.
- [32] SAXENA, A., BONAMI, P., LEE, J. “Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations”, *Mathematical Programming*, v. 124, n. 1, pp. 383–411, 2010. ISSN: 1436-4646. doi: 10.1007/s10107-010-0371-9. Disponível em: <<http://dx.doi.org/10.1007/s10107-010-0371-9>>.
- [33] GRANT, M., BOYD, S. “CVX: Matlab Software for Disciplined Convex Programming, version 2.1”. <http://cvxr.com/cvx>, mar. 2014.
- [34] ANDERSEN, E., ANDERSEN, K. “The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm”. In: et al., H. F. (Ed.), *High Performance Optimization; Appl. Opt.*, v. 33, Kluwer, 1999.

- [35] CUNHA, J. O. *Algoritmos para o problema da mochila quadrática 0-1*. Tese de D.Sc., Universidade Federal do Rio de Janeiro - COPPE, Rio de Janeiro, RJ, Brasil, 2014.
- [36] GUROBI OPTIMIZATION, L. “Gurobi Optimizer Reference Manual”. 2018. Disponível em: <<http://www.gurobi.com>>.

# Apêndice A

## Detalhamento dos resultados

Neste capítulo apresentamos em detalhes os resultados que complementam as Tabelas 5.1, 5.3, 5.4, 5.5, apresentadas no Capítulo 5 desta tese.

### A.1 Tabelas comparando plano de cortes e relaxações da literatura

Instância	LP		SDP		CPA <sub>5</sub>	
	gap	t(s)	gap	t(s)	gap	t(s)
I <sub>50,25,1</sub>	0,92	0,19	0,85	2,91	0,92	11,84
I <sub>50,25,2</sub>	4,45	0,18	1,42	4,19	2,70	26,71
I <sub>50,25,3</sub>	0,56	0,17	0,54	3,63	0,56	14,07
I <sub>50,25,4</sub>	0,05	0,19	0,05	4,01	0,05	14,6
I <sub>50,25,5</sub>	2,66	0,18	1,17	3,95	2,59	12,31
I <sub>50,25,6</sub>	0,68	0,18	0,67	4,49	0,68	15,15
I <sub>50,25,7</sub>	10,87	0,13	1,03	4,13	2,74	64,05
I <sub>50,25,8</sub>	1,28	0,15	1,14	3,13	1,28	12,73
I <sub>50,25,9</sub>	0,54	0,12	0,45	4,24	0,54	14,45
I <sub>50,25,10</sub>	1,02	0,14	0,97	4,27	1,02	11,86
<b>Média</b>	2,3	0,16	0,83	3,9	1,31	19,78

Tabela A.1: Gaps obtidos para  $n = 50$  itens e  $d = 25$  pelas diferentes relaxações do QKP.

Instância	LP		SDP		CPA <sub>5</sub>	
	gap	t(s)	gap	t(s)	gap	t(s)
I <sub>50,50,1</sub>	4,86	0,62	0,75	5,41	0,97	11,11
I <sub>50,50,2</sub>	0,67	0,14	0,66	4,72	0,67	12,23
I <sub>50,50,3</sub>	19,62	0,13	0,28	4,84	0,68	19,96
I <sub>50,50,4</sub>	2,91	0,12	0,72	5,35	0,93	23,35
I <sub>50,50,5</sub>	2,72	0,12	1,07	5,85	1,31	22,87
I <sub>50,50,6</sub>	0,41	0,13	0,38	4,86	0,41	11,73
I <sub>50,50,7</sub>	0,48	0,14	0,25	4,96	0,45	10,79
I <sub>50,50,8</sub>	0,58	0,13	0,24	7,25	0,39	13,51
I <sub>50,50,9</sub>	2,52	0,15	1,38	4,73	1,58	13,92
I <sub>50,50,10</sub>	0,88	0,14	0,68	4,66	0,77	8,45
<b>Média</b>	<b>3,57</b>	<b>0,18</b>	<b>0,64</b>	<b>5,26</b>	<b>0,82</b>	<b>14,79</b>

Tabela A.2: Gaps obtidos para  $n = 50$  itens e  $d = 50$  pelas diferentes relaxações do QKP.

Instância	LP		SDP		CPA <sub>5</sub>	
	gap	t(s)	gap	t(s)	gap	t(s)
I <sub>50,75,1</sub>	1,96	0,17	0,52	4,69	0,54	28,73
I <sub>50,75,2</sub>	6,55	0,14	0,25	4,79	0,33	27,17
I <sub>50,75,3</sub>	0,26	0,15	0,26	3,86	0,26	15,45
I <sub>50,75,4</sub>	9,38	0,2	6,10	4	6,20	14,93
I <sub>50,75,5</sub>	0,71	0,15	0,67	4,61	0,71	14,45
I <sub>50,75,6</sub>	1,59	0,14	0,18	4,55	0,25	15,62
I <sub>50,75,7</sub>	0,34	0,16	0,33	5,34	0,34	15,51
I <sub>50,75,8</sub>	2,58	0,17	0,27	4,35	0,33	12,99
I <sub>50,75,9</sub>	1,94	0,16	1,87	4,4	1,89	18,85
I <sub>50,75,10</sub>	5,56	0,18	0,56	4,87	0,60	22,88
<b>Média</b>	<b>3,09</b>	<b>0,16</b>	<b>1,10</b>	<b>4,55</b>	<b>1,15</b>	<b>18,66</b>

Tabela A.3: Gaps obtidos para  $n = 50$  itens e  $d = 75$  pelas diferentes relaxações do QKP.

Instância	LP		SDP		CPA <sub>5</sub>	
	gap	t(s)	gap	t(s)	gap	t(s)
I <sub>50,100,1</sub>	1,87	0,16	1,49	4,74	1,49	13,83
I <sub>50,100,2</sub>	1,26	0,27	0,81	5,1	0,82	14,17
I <sub>50,100,3</sub>	17,75	0,29	0,38	3,53	0,38	12,09
I <sub>50,100,4</sub>	6,62	0,28	1,59	3,5	1,59	12,69
I <sub>50,100,5</sub>	1,56	0,33	1,51	4,28	1,52	13,03
I <sub>50,100,6</sub>	3,97	0,33	3,33	4,28	3,34	13,60
I <sub>50,100,7</sub>	9,34	0,31	0,44	4,32	0,46	13,60
I <sub>50,100,8</sub>	2,24	0,36	1,53	3,78	1,53	9,94
I <sub>50,100,9</sub>	1,87	0,28	0,06	4,76	0,07	14,12
I <sub>50,100,10</sub>	4,84	0,33	1,36	4,86	1,38	14,81
<b>Média</b>	5,13	0,294	1,25	4,315	1,26	13,19

Tabela A.4: Gaps obtidos para  $n = 50$  itens e  $d = 100$  pelas diferentes relaxações do QKP.

Instância	LP		SDP		CPA <sub>5</sub>	
	gap	t(s)	gap	t(s)	gap	t(s)
I <sub>100,25,1</sub>	0,25	0,39	0,16	212,48	0,25	48,99
I <sub>100,25,2</sub>	5,77	1,34	0,76	209,06	1,21	33,6
I <sub>100,25,3</sub>	0,08	0,37	0,08	231,70	0,08	46,06
I <sub>100,25,4</sub>	1,05	0,32	0,12	202,52	0,55	33,26
I <sub>100,25,5</sub>	0,09	0,38	0,05	241,04	0,09	47,69
I <sub>100,25,6</sub>	0,25	0,35	0,22	264,68	0,25	51,51
I <sub>100,25,7</sub>	0,45	0,35	0,38	340,60	0,45	41,23
I <sub>100,25,8</sub>	0,18	0,36	0,17	218,19	0,18	26,09
I <sub>100,25,9</sub>	2,13	0,39	0,26	197,43	2,13	26,99
I <sub>100,25,10</sub>	24,02	0,37	1,05	193,74	2,01	104,5
<b>Média</b>	3,43	0,462	0,33	231,14	0,78	45,99

Tabela A.5: Gaps obtidos para  $n = 100$  itens e  $d = 25$  pelas diferentes relaxações do QKP.

Instância	LP		SDP		CPA <sub>5</sub>	
	gap	t(s)	gap	t(s)	gap	t(s)
I <sub>100,50,1</sub>	1,11	0,96	0,04	312,21	0,16	29,34
I <sub>100,50,2</sub>	2,82	0,46	0,43	162,67	0,48	55,32
I <sub>100,50,3</sub>	1,81	0,49	0,31	199,68	0,44	73,53
I <sub>100,50,4</sub>	3,96	0,65	0,19	180,11	0,70	126,78
I <sub>100,50,5</sub>	1,84	0,5	0,25	165,44	0,45	59,45
I <sub>100,50,6</sub>	0,16	0,41	0,15	172,46	0,16	39,41
I <sub>100,50,7</sub>	3,75	0,38	0,08	223,02	0,73	215,59
I <sub>100,50,8</sub>	1,28	0,61	0,06	211,42	0,65	55,43
I <sub>100,50,9</sub>	11,18	0,58	0,44	194,68	0,91	62,73
I <sub>100,50,10</sub>	1,22	0,65	0,16	143,93	0,50	46,74
<b>Média</b>	2,91	0,57	0,21	196,56	0,53	76,43

Tabela A.6: Gaps obtidos para  $n = 100$  itens e  $d = 50$  pelas diferentes relaxações do QKP.

Instância	LP		SDP		CPA <sub>5</sub>	
	gap	t(s)	gap	t(s)	gap	t(s)
I <sub>100,75,1</sub>	6,01	0,93	0,49	198,15	0,49	40,73
I <sub>100,75,2</sub>	1,67	0,75	0,20	154,56	0,22	87,52
I <sub>100,75,3</sub>	1,55	0,88	0,01	203,74	0,10	139,12
I <sub>100,75,4</sub>	2,55	0,61	0,10	203,55	0,13	133,94
I <sub>100,75,5</sub>	1,03	0,71	0,08	207,58	0,15	117,1
I <sub>100,75,6</sub>	28,98	0,76	<b>0,00</b>	122,38	<b>0,00</b>	13,89
I <sub>100,75,7</sub>	0,34	0,78	0,18	180,34	0,20	60,99
I <sub>100,75,8</sub>	9,16	1,05	0,53	179,63	0,69	81,31
I <sub>100,75,9</sub>	0,29	0,76	0,05	185,42	0,08	54,52
I <sub>100,75,10</sub>	0,32	0,75	0,22	216,17	0,26	53,86
<b>Média</b>	5,20	0,80	0,19	185,15	0,24	78,30

Tabela A.7: Gaps obtidos para  $n = 100$  itens e  $d = 75$  pelas diferentes relaxações do QKP.

Instância	LP		SDP		CPA <sub>5</sub>	
	gap	t(s)	gap	t(s)	gap	t(s)
I <sub>100,100,1</sub>	3,46	1,04	<b>0,00</b>	111,99	<b>0,00</b>	109,24
I <sub>100,100,2</sub>	2,51	0,54	0,46	207,51	0,46	54,82
I <sub>100,100,3</sub>	0,73	0,63	<b>0,00</b>	126,66	<b>0,00</b>	49,65
I <sub>100,100,4</sub>	4,32	0,79	0,13	223,56	0,13	37,62
I <sub>100,100,5</sub>	0,99	0,78	0,29	236,71	0,30	48,06
I <sub>100,100,6</sub>	0,31	1,07	0,09	218,48	0,09	40,08
I <sub>100,100,7</sub>	0,82	0,64	0,17	214,83	0,17	57,67
I <sub>100,100,8</sub>	6,52	0,58	0,04	140,65	0,05	30,47
I <sub>100,100,9</sub>	0,9	0,84	0,10	202,01	0,10	53,27
I <sub>100,100,10</sub>	1,19	0,63	0,83	167,57	0,85	28,46
<b>Média</b>	2,18	0,75	0,21	185	0,22	50,93

Tabela A.8: Gaps obtidos para  $n = 100$  itens e  $d = 100$  pelas diferentes relaxações do QKP.

## A.2 Tabelas comparando cortes propostos e o IPM

Instância	Optgap (%)	Tempo (s)	DuGap	Iter
I <sub>10,100,1</sub>	14.225	0.83	0.155	1
I <sub>10,100,2</sub>	20.887	0.79	0.099	1
I <sub>10,100,3</sub>	28.223	0.79	0.124	1
I <sub>10,100,4</sub>	19.646	0.79	0.089	1
I <sub>10,100,5</sub>	17.168	0.78	0.210	1
I <sub>10,100,6</sub>	18.074	0.78	0.083	1
I <sub>10,100,7</sub>	12.910	0.80	0.104	1
I <sub>10,100,8</sub>	21.493	0.78	0.117	1
I <sub>10,100,9</sub>	23.219	0.78	0.152	1
I <sub>10,100,10</sub>	40.555	0.78	0.250	1
<b>Média</b>	21.640	0.79	0.138	1.00

Tabela A.9: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^a$ , QPR, Tabela 5.3.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>10,100,1</sub>	6.882	1.23	0.040	1
I <sub>10,100,2</sub>	14.291	0.80	0.071	1
I <sub>10,100,3</sub>	15.306	0.80	0.071	1
I <sub>10,100,4</sub>	9.692	0.79	0.046	1
I <sub>10,100,5</sub>	8.954	0.79	0.050	1
I <sub>10,100,6</sub>	8.245	0.80	0.048	1
I <sub>10,100,7</sub>	6.330	0.79	0.041	1
I <sub>10,100,8</sub>	12.150	0.78	0.077	1
I <sub>10,100,9</sub>	17.512	0.79	0.103	1
I <sub>10,100,10</sub>	23.395	0.79	0.217	1
<b>Média</b>	<b>12.276</b>	<b>0.83</b>	<b>0.076</b>	<b>1.00</b>

Tabela A.10: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^b$ , QPR, Tabela 5.3.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>10,100,1</sub>	5.198	26.78	0.032	150
I <sub>10,100,2</sub>	10.879	26.85	0.056	150
I <sub>10,100,3</sub>	5.412	26.88	0.026	150
I <sub>10,100,4</sub>	5.134	26.63	0.034	150
I <sub>10,100,5</sub>	6.739	26.82	0.040	150
I <sub>10,100,6</sub>	6.003	26.90	0.029	150
I <sub>10,100,7</sub>	3.922	26.79	0.019	150
I <sub>10,100,8</sub>	6.496	26.79	0.051	150
I <sub>10,100,9</sub>	13.740	26.77	0.087	150
I <sub>10,100,10</sub>	8.892	26.82	0.049	150
<b>Média</b>	<b>7.242</b>	<b>26.80</b>	<b>0.042</b>	<b>150.00</b>

Tabela A.11: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^a$ , I, Tabela 5.3.



<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>10,100,1</sub>	5.292	25.57	0.033	150
I <sub>10,100,2</sub>	10.788	25.63	0.056	150
I <sub>10,100,3</sub>	5.587	25.72	0.027	150
I <sub>10,100,4</sub>	5.357	25.85	0.035	150
I <sub>10,100,5</sub>	6.883	25.72	0.041	150
I <sub>10,100,6</sub>	5.965	25.76	0.029	150
I <sub>10,100,7</sub>	4.029	25.76	0.020	150
I <sub>10,100,8</sub>	8.068	26.13	0.059	150
I <sub>10,100,9</sub>	14.200	26.09	0.089	150
I <sub>10,100,10</sub>	10.162	25.90	0.162	150
<b>Média</b>	<b>7.633</b>	<b>25.81</b>	<b>0.055</b>	<b>150.0</b>

Tabela A.12: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^b$ , Tabela 5.3.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>10,100,1</sub>	5.140	27.99	0.032	150
I <sub>10,100,2</sub>	10.683	27.54	0.055	150
I <sub>10,100,3</sub>	5.377	27.42	0.026	150
I <sub>10,100,4</sub>	4.692	27.45	0.023	150
I <sub>10,100,5</sub>	6.578	27.44	0.039	150
I <sub>10,100,6</sub>	5.871	27.41	0.029	150
I <sub>10,100,7</sub>	3.764	27.34	0.018	150
I <sub>10,100,8</sub>	6.341	27.30	0.051	150
I <sub>10,100,9</sub>	13.667	27.23	0.086	150
I <sub>10,100,10</sub>	8.792	27.00	0.049	150
<b>Média</b>	<b>7.091</b>	<b>27.41</b>	<b>0.041</b>	<b>150.0</b>

Tabela A.13: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^a$ , BFGS, Tabela 5.3.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>10,100,1</sub>	5.140	25.89	0.032	150
I <sub>10,100,2</sub>	10.683	26.18	0.055	150
I <sub>10,100,3</sub>	5.358	26.17	0.026	150
I <sub>10,100,4</sub>	4.692	26.21	0.023	150
I <sub>10,100,5</sub>	6.583	25.77	0.039	150
I <sub>10,100,6</sub>	5.873	25.98	0.029	150
I <sub>10,100,7</sub>	3.782	25.95	0.019	150
I <sub>10,100,8</sub>	6.307	25.79	0.050	150
I <sub>10,100,9</sub>	13.731	25.76	0.087	150
I <sub>10,100,10</sub>	8.793	25.69	0.049	150
<b>Média</b>	<b>7.094</b>	<b>25.94</b>	<b>0.041</b>	<b>150.0</b>

Tabela A.14: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^b$ ,BFGS, Tabela 5.3.

<b>Instância</b>	<b>OptGap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>	<b>Cortes</b>	<b>Tempo<sub>MILP</sub> (s)</b>
I <sub>10,100,1</sub>	<b>0.000</b>	86.14	<b>0.000</b>	150	90	40.19
I <sub>10,100,2</sub>	0.237	81.40	0.002	150	73	38.35
I <sub>10,100,3</sub>	2.267	99.15	0.011	150	143	51.24
I <sub>10,100,4</sub>	<b>0.000</b>	89.06	<b>0.000</b>	150	97	42.52
I <sub>10,100,5</sub>	<b>0.000</b>	89.33	<b>0.000</b>	150	108	41.75
I <sub>10,100,6</sub>	<b>0.000</b>	85.63	<b>0.000</b>	147	93	40.16
I <sub>10,100,7</sub>	<b>0.000</b>	78.88	<b>0.000</b>	138	82	37.65
I <sub>10,100,8</sub>	0.430	100.94	0.002	150	131	49.99
I <sub>10,100,9</sub>	<b>0.000</b>	64.22	<b>0.000</b>	110	103	30.53
I <sub>10,100,10</sub>	5.699	96.11	0.079	150	120	48.93
<b>Média</b>	<b>0.863</b>	<b>87.09</b>	<b>0.009</b>	<b>144.50</b>	<b>104.00</b>	<b>42.13</b>

Tabela A.15: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^a$ ,I,Cuts, Tabela 5.3.

<b>Instância</b>	<b>OptGap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>	<b>Cortes</b>	<b>Tempo<sub>MILP</sub> (s)</b>
I <sub>10,100,1</sub>	<b>0.000</b>	83.53	<b>0.000</b>	142	94	40.84
I <sub>10,100,2</sub>	0.088	81.30	0.000	150	73	38.26
I <sub>10,100,3</sub>	2.995	98.87	0.015	150	145	51.43
I <sub>10,100,4</sub>	<b>0.000</b>	78.77	<b>0.000</b>	134	93	39.03
I <sub>10,100,5</sub>	0.001	83.47	0.000	150	108	39.14
I <sub>10,100,6</sub>	<b>0.000</b>	77.30	<b>0.000</b>	141	90	38.25
I <sub>10,100,7</sub>	<b>0.000</b>	6.50	<b>0.000</b>	11	17	3.57
I <sub>10,100,8</sub>	3.273	90.65	0.016	150	125	46.37
I <sub>10,100,9</sub>	0.036	85.67	0.000	150	117	40.26
I <sub>10,100,10</sub>	8.766	89.90	0.093	150	114	46.59
<b>Média</b>	<b>1.516</b>	<b>77.60</b>	<b>0.012</b>	<b>132.80</b>	<b>97.60</b>	<b>38.37</b>

Tabela A.16: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^b, I, Cuts$ , Tabela 5.3.

<b>Instância</b>	<b>OptGap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>	<b>Cortes</b>	<b>Tempo<sub>MILP</sub> (s)</b>
I <sub>10,100,1</sub>	<b>0.000</b>	38.29	<b>0.000</b>	71	150	20.72
I <sub>10,100,2</sub>	<b>0.000</b>	38.06	<b>0.000</b>	80	47	18.95
I <sub>10,100,3</sub>	1.423	97.70	0.007	150	723	50.07
I <sub>10,100,4</sub>	<b>0.000</b>	6.98	<b>0.000</b>	13	111	3.61
I <sub>10,100,5</sub>	<b>0.000</b>	45.60	<b>0.000</b>	82	209	24.07
I <sub>10,100,6</sub>	<b>0.000</b>	7.75	<b>0.000</b>	17	79	3.58
I <sub>10,100,7</sub>	<b>0.000</b>	22.56	<b>0.000</b>	44	147	12.15
I <sub>10,100,8</sub>	<b>0.000</b>	95.27	<b>0.000</b>	149	559	46.94
I <sub>10,100,9</sub>	<b>0.000</b>	12.47	<b>0.000</b>	21	223	7.33
I <sub>10,100,10</sub>	4.967	96.60	0.075	150	506	48.33
<b>Média</b>	<b>0.639</b>	<b>46.13</b>	<b>0.008</b>	<b>77.70</b>	<b>275.40</b>	<b>23.57</b>

Tabela A.17: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^a, BFGS, Cuts$ , Tabela 5.3.

<b>Instância</b>	<b>OptGap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>	<b>Cortes</b>	<b>Tempo<sub>MILP</sub> (s)</b>
I <sub>10,100,1</sub>	<b>0.000</b>	17.40	<b>0.000</b>	32	34	10.00
I <sub>10,100,2</sub>	<b>0.000</b>	41.87	<b>0.000</b>	87	43	21.22
I <sub>10,100,3</sub>	1.431	96.14	0.007	150	145	49.85
I <sub>10,100,4</sub>	<b>0.000</b>	35.95	<b>0.000</b>	69	58	18.97
I <sub>10,100,5</sub>	<b>0.000</b>	51.42	<b>0.000</b>	98	66	26.08
I <sub>10,100,6</sub>	<b>0.000</b>	48.63	<b>0.000</b>	92	64	25.40
I <sub>10,100,7</sub>	<b>0.000</b>	48.33	<b>0.000</b>	91	59	25.42
I <sub>10,100,8</sub>	<b>0.000</b>	72.21	<b>0.000</b>	117	123	38.17
I <sub>10,100,9</sub>	<b>0.000</b>	57.67	<b>0.000</b>	101	90	29.51
I <sub>10,100,10</sub>	4.966	93.59	0.075	150	125	47.77
<b>Média</b>	<b>0.640</b>	<b>56.32</b>	<b>0.008</b>	<b>98.70</b>	<b>80.70</b>	<b>29.24</b>

Tabela A.18: Resultados para 10 instâncias ( $n = 10$ ) usando  $Q_p^b$ , BFGS, Cuts, Tabela 5.3.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>30,100,1</sub>	22.569	0.14	0.853	1
I <sub>30,100,2</sub>	19.710	0.14	0.836	1
I <sub>30,100,3</sub>	33.098	0.25	0.825	1
I <sub>30,100,4</sub>	25.095	0.12	0.824	1
I <sub>30,100,5</sub>	20.667	0.11	0.826	1
I <sub>30,100,6</sub>	22.416	0.15	0.850	1
I <sub>30,100,7</sub>	20.677	0.11	0.854	1
I <sub>30,100,8</sub>	25.450	0.14	0.864	1
I <sub>30,100,9</sub>	23.813	0.17	0.871	1
I <sub>30,100,10</sub>	14.901	0.07	0.850	1
<b>Média</b>	<b>22.840</b>	<b>0.14</b>	<b>0.845</b>	<b>1.00</b>

Tabela A.19: Resultados para 10 instâncias ( $n = 30$ ) usando  $Q_p^a$ , QPR, Tabela 5.4.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>30,100,1</sub>	11.447	0.08	0.183	1
I <sub>30,100,2</sub>	8.586	0.08	0.858	1
I <sub>30,100,3</sub>	16.585	0.16	0.847	1
I <sub>30,100,4</sub>	11.992	0.06	0.853	1
I <sub>30,100,5</sub>	9.638	0.06	0.855	1
I <sub>30,100,6</sub>	8.995	0.04	0.862	1
I <sub>30,100,7</sub>	9.602	0.06	0.827	1
I <sub>30,100,8</sub>	12.409	0.07	0.829	1
I <sub>30,100,9</sub>	10.366	0.08	0.836	1
I <sub>30,100,10</sub>	6.561	0.04	0.836	1
<b>Média</b>	<b>10.618</b>	<b>0.07</b>	<b>0.779</b>	<b>1.00</b>

Tabela A.20: Resultados para 10 instâncias ( $n = 30$ ) usando  $Q_p^b$ , QPR, Tabela 5.4.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>30,100,1</sub>	3.673	132.23	0.042	250
I <sub>30,100,2</sub>	<b>0.000</b>	71.679	<b>0.000</b>	143
I <sub>30,100,3</sub>	3.125	132.52	0.045	250
I <sub>30,100,4</sub>	4.537	135.59	0.022	250
I <sub>30,100,5</sub>	5.571	134.46	0.037	250
I <sub>30,100,6</sub>	<b>0.000</b>	2.49	<b>0.000</b>	3
I <sub>30,100,7</sub>	4.080	133.53	0.036	250
I <sub>30,100,8</sub>	5.979	135.59	0.045	250
I <sub>30,100,9</sub>	1.221	130.97	0.043	250
I <sub>30,100,10</sub>	2.454	139.17	0.021	250
<b>Média</b>	<b>3.064</b>	<b>114.82</b>	<b>0.029</b>	<b>214.60</b>

Tabela A.21: Resultados para 10 instâncias ( $n = 30$ ) usando  $Q_p^a$ , BFGS, Tabela 5.4.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>30,100,1</sub>	3.673	129.38	0.042	250
I <sub>30,100,2</sub>	<b>0.000</b>	48.65	<b>0.000</b>	93
I <sub>30,100,3</sub>	3.097	132.18	0.045	250
I <sub>30,100,4</sub>	4.556	131.59	0.022	250
I <sub>30,100,5</sub>	5.574	131.90	0.037	250
I <sub>30,100,6</sub>	<b>0.000</b>	1.42	<b>0.000</b>	1
I <sub>30,100,7</sub>	4.081	136.12	0.036	250
I <sub>30,100,8</sub>	5.984	130.20	0.045	250
I <sub>30,100,9</sub>	1.221	133.14	0.043	250
I <sub>30,100,10</sub>	2.454	132.85	0.021	250
<b>Média</b>	<b>3.064</b>	<b>110.74</b>	<b>0.029</b>	<b>209.40</b>

Tabela A.22: Resultados para 10 instâncias ( $n = 30$ ) usando  $Q_p^b$ ,BFGS, Tabela 5.4.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>	<b>Cortes</b>	<b>Tempo<sub>MILP</sub> (s)</b>
I <sub>30,100,1</sub>	<b>0.000</b>	129.93	<b>0.000</b>	49	842	95.7836
I <sub>30,100,2</sub>	<b>0.000</b>	410.10	<b>0.000</b>	118	2371	312.6191
I <sub>30,100,3</sub>	1.600	708.66	0.038	250	2243	472.01
I <sub>30,100,4</sub>	0.481	842.92	0.002	250	4683	537.07
I <sub>30,100,5</sub>	<b>0.000</b>	271.30	<b>0.000</b>	84	1596	205.65
I <sub>30,100,6</sub>	<b>0.000</b>	3.01	<b>0.000</b>	3	0	0.00
I <sub>30,100,7</sub>	0.390	928.81	0.005	250	4529	643.50
I <sub>30,100,8</sub>	0.304	899.19	0.002	250	4078	616.90
I <sub>30,100,9</sub>	<b>0.000</b>	406.13	<b>0.000</b>	121	2494	303.96
I <sub>30,100,10</sub>	<b>0.000</b>	339.55	<b>0.000</b>	97	2089	257.20
<b>Média</b>	<b>0.278</b>	<b>493.96</b>	<b>0.005</b>	<b>147.20</b>	<b>2492.50</b>	<b>344.47</b>

Tabela A.23: Resultados para 10 instâncias ( $n = 30$ ) usando  $Q_p^a$ ,BFGS,Cuts, Tabela 5.4.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>	<b>Cortes</b>	<b>Tempo<sub>MILP</sub> (s)</b>
I <sub>30,100,1</sub>	<b>0.000</b>	390.44	<b>0.000</b>	128	1895	291.67
I <sub>30,100,2</sub>	<b>0.000</b>	219.68	<b>0.000</b>	63	1155	174.96
I <sub>30,100,3</sub>	1.607	683.71	0.038	250	2328	452.70
I <sub>30,100,4</sub>	0.649	802.57	0.003	250	4712	526.31
I <sub>30,100,5</sub>	<b>0.000</b>	415.29	<b>0.000</b>	134	2501	300.95
I <sub>30,100,6</sub>	<b>0.000</b>	1.45	<b>0.000</b>	1	0	0.00
I <sub>30,100,7</sub>	0.412	928.55	0.005	250	4466	646.35
I <sub>30,100,8</sub>	0.327	872.49	0.002	250	4717	609.01
I <sub>30,100,9</sub>	<b>0.000</b>	289.23	<b>0.000</b>	92	1684	220.39
I <sub>30,100,10</sub>	<b>0.000</b>	692.87	<b>0.000</b>	188	4115	499.28
<b>Média</b>	<b>0.300</b>	<b>529.63</b>	<b>0.005</b>	<b>160.60</b>	<b>2757.3</b>	<b>372.16</b>

Tabela A.24: Resultados para 10 instâncias ( $n = 30$ ) usando  $Q_p^b$ , BFGS, Cuts, Tabela 5.4.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>50,100,6</sub>	22.615	0.87	0.113	1
I <sub>50,100,7</sub>	24.991	0.87	0.146	1
I <sub>50,100,8</sub>	20.758	0.86	0.115	1
I <sub>50,100,9</sub>	15.056	0.89	0.080	1
I <sub>50,100,10</sub>	22.258	0.87	0.112	1
<b>Média</b>	<b>21.136</b>	<b>0.87</b>	<b>0.113</b>	<b>1.00</b>

Tabela A.25: Resultados para 5 instâncias ( $n = 50$ ) usando  $Q_p^a$ , QPR, Tabela 5.5.

<b>Instância</b>	<b>Optgap (%)</b>	<b>Tempo (s)</b>	<b>DuGap</b>	<b>Iter</b>
I <sub>50,100,6</sub>	11.021	0.86	0.063	1
I <sub>50,100,7</sub>	11.046	0.87	0.064	1
I <sub>50,100,8</sub>	9.100	0.87	0.056	1
I <sub>50,100,9</sub>	6.708	0.86	0.043	1
I <sub>50,100,10</sub>	9.424	0.86	0.055	1
<b>Média</b>	<b>9.460</b>	<b>0.86</b>	<b>0.056</b>	<b>1.00</b>

Tabela A.26: Resultados para 5 instâncias ( $n = 50$ ) usando  $Q_p^b$ , QPR, Tabela 5.5.

Instância	Optgap (%)	Tempo (s)	DuGap	Iter
I <sub>50,100,6</sub>	3.335	2711.39	0.024	420
I <sub>50,100,7</sub>	0.437	2762.92	0.009	430
I <sub>50,100,8</sub>	1.526	2703.64	0.020	420
I <sub>50,100,9</sub>	0.063	2764.89	0.005	430
I <sub>50,100,10</sub>	1.365	2720.67	0.015	420
Média	1.345	2732.70	0.015	424.0

Tabela A.27: Resultados para 5 instâncias ( $n = 50$ ) usando  $Q_p^a$ ,BFGS, Tabela 5.5.

Instância	Optgap (%)	Tempo (s)	DuGap	Iter
I <sub>50,100,6</sub>	3.335	2710.46	0.024	430
I <sub>50,100,7</sub>	0.437	2700.75	0.009	430
I <sub>50,100,8</sub>	1.526	2727.02	0.020	430
I <sub>50,100,9</sub>	0.063	2707.82	0.005	430
I <sub>50,100,10</sub>	1.365	2720.24	0.015	430
Média	1.345	2713.26	0.015	430.0

Tabela A.28: Resultados para 5 instâncias ( $n = 50$ ) usando  $Q_p^b$ ,BFGS, Tabela 5.5.

Instância	Optgap (%)	Tempo (s)	DuGap	Iter	Cortes	Tempo <sub>MILP</sub> (s)
I <sub>50,100,6</sub>	0.195	2811.35	0.001	210	281	1049.42
I <sub>50,100,7</sub>	<b>0.000</b>	2265.98	<b>0.000</b>	172	287	901.29
I <sub>50,100,8</sub>	<b>0.000</b>	1529.21	<b>0.000</b>	121	179	627.09
I <sub>50,100,9</sub>	<b>0.000</b>	1652.68	<b>0.000</b>	130	165	653.44
I <sub>50,100,10</sub>	0.195	2824.85	0.002	210	294	1061.35
Média	0.078	2216.81	0.001	168.6	241.2	858.51

Tabela A.29: Resultados para 5 instâncias ( $n = 50$ ) usando  $Q_p^a$ ,BFGS,Cuts, Tabela 5.5.

Instância	Optgap (%)	Tempo (s)	DuGap	Iter	Cortes	Tempo <sub>MILP</sub> (s)
I <sub>50,100,6</sub>	0.186	2727.93	0.001	210	269	1038.60
I <sub>50,100,7</sub>	<b>0.000</b>	1737.55	<b>0.000</b>	136	205	696.75
I <sub>50,100,8</sub>	<b>0.000</b>	1028.75	<b>0.000</b>	82	127	421.08
I <sub>50,100,9</sub>	<b>0.000</b>	1087.04	<b>0.000</b>	87	111	432.68
I <sub>50,100,10</sub>	0.196	2832.12	0.002	210	285	1068.69
Média	0.076	1882.68	0.001	145.0	199.4	731.55

Tabela A.30: Resultados para 5 instâncias ( $n = 50$ ) usando  $Q_p^b$ ,BFGS,Cuts, Tabela 5.5.