



EXTENSÃO DE LIMITES ELIPSOIDAIIS EM PROGRAMAÇÃO QUADRÁTICA INTEIRA

Francisco Ismael Pinillos Nieto

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 2017

EXTENSÃO DE LIMITES ELIPSOIDAIIS EM PROGRAMAÇÃO
QUADRÁTICA INTEIRA

Francisco Ismael Pinillos Nieto

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. Susana Scheimberg de Makler, D.Sc.

Prof. Fernanda Maria Pereira Raupp , D.Sc.

Prof. Rosiane de Freitas Rodrigues, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2017

Pinillos Nieto, Francisco Ismael

Extensão de limites elipsoidais em programação quadrática inteira/Francisco Ismael Pinillos Nieto. – Rio de Janeiro: UFRJ/COPPE, 2017.

XIII, 71 p.: il.; 29, 7cm.

Orientador: Marcia Helena Costa Fampa

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2017.

Referências Bibliográficas: p. 69 – 71.

1. Limite elipsoidal. 2. Elipsoide quase-arredondado.
3. Programação quadrática inteira. I. Fampa, Marcia Helena Costa. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*Aos meus pais pela dedicação,
cuidado e investimento em
minha educação.
À minha amada esposa, Yanina,
pelo incansável apoio,
compreensão e amor.*

Agradecimentos

A Deus.

A meus familiares, por manterem nossos laços afetivos tão vivos.

A minha orientadora Marcia Fampa, pela atenção, apoio e incentivo durante todo o período que estive no Programa de Engenharia de Sistemas e computação da COPPE/UFRJ.

Aos membros da Banca de Tese.

A Capes, pelo apoio financeiro durante o período de realização da tese.

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

EXTENSÃO DE LIMITES ELIPSOIDAIIS EM PROGRAMAÇÃO QUADRÁTICA INTEIRA

Francisco Ismael Pinillos Nieto

Março/2017

Orientador: Marcia Helena Costa Fampa

Programa: Engenharia de Sistemas e Computação

Limites elipsoidais para problemas de programação inteira estritamente convexa foram propostos em [1, 2]. A ideia é subestimar a função objetivo quadrática q do problema por outra função quadrática convexa com o mesmo minimizador contínuo da função q e para a qual um minimizador inteiro pode ser facilmente calculado.

Propomos nesta tese uma maneira diferente de construir o subestimador quadrático para o mesmo problema e então estender a ideia a outros problemas inteiros quadráticos, onde a função objetivo é convexa (não necessariamente estritamente convexa), e onde a função objetivo é não convexa com a introdução de restrições de caixa. A qualidade dos limites propostos é avaliada experimentalmente e comparada com as metodologias relacionadas.

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

EXTENSIONS ON ELLIPSOID BOUNDS FOR QUADRATIC INTEGER PROGRAMMING

Francisco Ismael Pinillos Nieto

March/2017

Advisor: Marcia Helena Costa Fampa

Department: Systems Engineering and Computer Science

Ellipsoid bounds for strictly convex quadratic integer programs have been proposed in [1, 2]. The idea is to underestimate the strictly convex quadratic objective function q of the problem by another convex quadratic function with the same continuous minimizer as q and for which an integer minimizer can be easily computed.

We propose in this thesis a different way of constructing the quadratic underestimator for the same problem and then extend the idea to other quadratic integer problems, where the objective function is convex (not necessarily strictly convex), and where the objective function is nonconvex and box constraints are introduced. The quality of the proposed bounds is evaluated experimentally and compared to the related existing methodologies.

Sumário

Lista de Figuras	x
Lista de Tabelas	xii
Lista de Abreviaturas	xiii
1 Introdução	1
2 Preliminares	4
2.1 Descrição do problema	4
2.2 Notações e definições	5
2.3 Redução de base de um Reticulado	7
2.3.1 Problema dos vetores curtos	10
2.3.2 Redução de base	11
3 Limite elipsoidal para programação quadrática estritamente convexa	15
3.1 Limite elipsoidal	15
3.2 Matrizes de posto 1	22
3.3 Matrizes de posto completo	25
3.4 Algoritmos com matrizes de posto um	33
3.4.1 Proposta de uma nova heurística	34
3.4.2 Teste computacional	36
3.5 Adicionando matrizes de posto um	38
3.6 Testes Computacionais	40
3.6.1 Geração de instâncias	40
3.6.2 Resultados numéricos	40
4 Extensão do limite elipsoidal para a programação quadrática convexa inteira irrestrita	44
4.1 Resultados teóricos	44
4.2 Algoritmos baseados em matrizes de posto um	52

4.3	Testes Computacionais	54
5	Aplicação dos limites elipsoidais na programação quadrática inteira não convexa com restrições de caixa	59
5.1	Estratégia de decomposição	59
5.1.1	Estratégias de decomposição diagonal	61
5.2	Testes Computacionais	62
6	Conclusões e recomendações	67
	Referências Bibliográficas	69

Lista de Figuras

2.1	Os vértices do cubo pertencem ao conjunto $Round(\bar{x}) \subseteq \mathbb{R}^3$	6
2.2	Exemplos de retículos gerado com diferentes bases em \mathbb{R}^2	8
2.3	Pontos do reticulado Λ gerado por M	9
2.4	Pontos do reticulado Λ gerado por \tilde{M}	9
2.5	Pontos do reticulado Λ gerado por M	10
2.6	Reticulado de posto 2 com duas bases diferentes. A segunda é a base reduzida e ortogonal.	12
3.1	Escalação do elipsoide $E(Q, \bar{x})$ pelo fator α até atingir o ponto inteiro x^*	17
3.2	O elipsoide branco $E(Q', \bar{x})$ é escalado até atingir o primeiro ponto inteiro x^* e por sua vez o escalonamento do elipsoide cinza $E(Q, \bar{x})$ produz um limite inferior sobre $q(x^*)$	18
3.3	O ponto $\lfloor \bar{x}_r \rfloor$ é atingido pelo elipsoide $\alpha E(Diag_r(\tau), \bar{x}_r)$	20
3.4	Obtenção de um limite inferior através do elipsoide eixo paralelo, $E(Diag(\tau), \bar{x})$	21
3.5	Maior limite inferior usando o elipsoide eixo paralelo $E(Diag(\tau), \bar{x})$	22
3.6	Elipsoide $E(Q', \bar{x})$ com a matriz Q de posto um.	24
3.7	Exemplos de funções com e sem a propriedade de arredondamento. (Fonte. [3])	26
3.8	Elipsoide $\alpha(\bar{x})$ -quase arredondado com $x^* = (0, 0) \in Round(\bar{x})$	28
3.9	O elipsoide não é $\beta(\bar{x})$ -quase arredondado, $x^* \neq \lfloor \bar{x} \rfloor$	29
3.10	Elipsoide α -quase arredondado, segundo a propriedade de seus autovalores.	30
3.11	Elipsoide $E(Q'_{BHS}, \bar{x})$ gerado tal que $E(Q, \bar{x}) \subseteq E(Q'_{BHS}, \bar{x})$	34
3.12	Vetor v quase ortogonal ao autovetor u_n	35
3.13	Elipsoide $E(Q', \bar{x})$ gerado com o vetor v	36
3.14	$\sqrt{z^* - q(\bar{x})}E(Q, \bar{x}) \subseteq \sqrt{z^* - q'(\bar{x})}E(Q', \bar{x})$	38
3.15	Melhora nos limites inferiores comparados com o mínimo contínuo.	42
3.16	Tempo de processamento da CPU em segundos.	42

4.1	Melhoria nos limites inferiores em relação ao mínimo contínuo.	55
4.2	tempo <i>CPU</i> no cálculo dos limites inferiores (segundos).	56
4.3	Melhoria nos limites inferiores em relação ao mínimo contínuo.	57
4.4	tempo <i>CPU</i> no cálculo dos limites inferiores (segundos).	58
5.1	Melhoria nos limites inferiores comparado ao mínimo contínuo.	64
5.2	Tempo médio de <i>CPU</i> (em segundos).	65

Lista de Tabelas

3.1	Tempo médio de <i>CPU</i> (em segundos).	37
3.2	Média dos <i>gaps</i> (%).	37
3.3	Tempo médio de processamento da <i>CPU</i> em segundos.	41
3.4	Média dos <i>gaps</i> para (<i>PQISC</i>) nas instâncias aleatórias.	41
4.1	Limites inferiores, média dos <i>gaps</i> (%).	54
4.2	Tempo médio de <i>CPU</i> (em segundos).	55
4.3	Tempo médio de <i>CPU</i> (em segundos).	56
4.4	Limites inferiores, média dos <i>gaps</i> (%).	57
5.1	Média dos <i>gaps</i> quando $P \succcurlyeq 0$, ($\epsilon = 0$).	63
5.2	Média dos <i>gaps</i> quando $P \succ 0$, ($\epsilon = 0.01$).	63
5.3	Tempos da <i>CPU</i> quando $P \succcurlyeq 0$, ($\epsilon = 0$).	63
5.4	Tempos da <i>CPU</i> quando $P \succ 0$, ($\epsilon = 0.01$).	63
5.5	Resultados das instâncias <i>Beasley</i> de <i>BiqMacLibrary</i> ($n = 50$).	66

Lista de Abreviaturas

<i>DC</i>	Decomposição em Diferença de Convexos, p. 59
<i>Gul</i>	Algoritmo Guloso, p. 39
<i>Gul</i> ⁺	Algoritmo Guloso semidefinido, p. 53
<i>Oti</i>	Algoritmo Otimizado, p. 39
<i>Oti</i> ⁺	Algoritmo Otimizado semidefinido, p. 53
<i>Rac</i>	Algoritmo Racional, p. 39
<i>Rac</i> ⁺	Algoritmo Racional semidefinido, p. 53
B&B	Branch-and-Bound, p. 1
BHS	Heurística do elipsoide quase arredondado, p. 31
BLC	Algoritmo proposto por Buchheim [1], p. 24
Hv	Heurística do vetor quase ortogonal., p. 34
LLL	Algoritmo de Lenstra, Lenstra e Lovász, p. 10
PIQNC	Programação Inteira Quadrática Não Convexa, p. 59
PQIEC	Programação Quadrática Inteira Estritamente Convexa, p. 4
PQI	Programação Quadrática Inteira, p. 4

Capítulo 1

Introdução

O problema de programação inteira quadrática tem sido foco de intensa pesquisa nos últimos anos, onde a obtenção de limites duais para o seu valor objetivo ótimo é essencial para o sucesso da aplicação de algoritmos baseados em uma abordagem *Branch-and-Bound* (*B&B*).

Quando o problema é convexo, uma abordagem muito comum para obter limites duais é simplesmente relaxar as restrições de integralidade e resolver a relaxação contínua. Uma abordagem alternativa foi proposta em [1] para o caso em que a função objetivo é estritamente convexa, a qual utiliza uma outra função quadrática convexa como subestimadora da função objetivo original, e cujo minimizador inteiro possa ser facilmente obtido.

Mais recentemente, em [2], os autores ampliaram a ideia introduzida em [1], propondo uma outra classe de funções subestimadoras da função objetivo e que tem a propriedade chamada de arredondamento forte, conceito que foi introduzido por Hübner em [4], esta propriedade estabelece que um minimizador inteiro é obtido facilmente pelo arredondamento das componentes do minimizador contínuo para o inteiro mais próximo.

Os autores em [2] propõem heurísticas para procurar funções quadráticas subestimadoras da função objetivo com esta propriedade de arredondamento forte que gerem limites elipsoidais de boa qualidade. A denominação de limites elipsoidais é motivada pela interpretação geométrica dos conjuntos de subnível da função objetivo que são elipsoides.

Os resultados em [2] mostram que os limites elipsoidais nunca são piores do que o limite inferior obtido pelo mínimo contínuo da função objetivo do problema e além disso, podem ser rapidamente calculados. No entanto, aplicam-se apenas para a minimização da função quadrática estritamente convexa.

Inicialmente propomos um novo método para gerar esses limites elipsoidais e em seguida, estendemos a teoria dos limites elipsoidais ao caso mais geral em que a função objetivo é convexa (não necessariamente estritamente convexa). A quali-

dade dos limites propostos é avaliada experimentalmente e em comparação com as metodologias relacionadas existentes.

Por outro lado, problemas de otimização quadrática não convexa também tem sido objeto de estudo nos últimos anos. Por exemplo, Kim e Kojima em [5] propõem uma relaxação mediante Programação Cônica de Segunda Ordem (**PCSO**) a fim de obter melhores limites inferiores para o problema. A ideia básica é acrescentar um número finito de desigualdades quadráticas convexas, que são válidas no cone positivo envolvido na relaxação de um problema de programação semidefinida positiva (**PSD**).

Já Saxena, Bonami e Lee em [6, 7] propõem relaxações convexas para problemas de programação quadrática inteira mista não convexa, chamadas de formulações estendidas e formulações projetadas, utilizando a técnica de programação disjuntiva, metodologia de elevar-projetar e polaridade.

Qualizza, Belloti e Margot em [8] apresentam uma relaxação linear de restrições semidefinidas positivas com o objetivo de capturar a maior parte da força da relaxação semidefinida positiva, com um resolvidor de programação linear. A relaxação linear obtida é uma aproximação externa do cone de matrizes semidefinidas positivas, o seu objetivo porem não é resolver de uma maneira eficiente os problemas do programação semidefinida positiva, mas sim encontrar formas práticas de restrições semidefinidas positivas aproximadas por restrições lineares. De modo geral, esses métodos têm como ponto de partida a relaxação semidefinida positiva dos problemas.

Outro dos objetivos de nossa pesquisa é mostrar como a ideia dos limites elipsoidais podem ser aplicados ao caso em que a função objetivo é não convexa com restrições de caixa. Os resultados numéricos comparam as abordagens propostas neste trabalho às abordagens relacionadas existentes na literatura.

O texto está organizado da seguinte forma:

Capítulo 2 São introduzidos conceitos, definições e propriedades que são fundamentais no desenvolvimento e na obtenção dos resultados deste trabalho. São apresentadas as questões relacionadas ao problema de programação quadrática inteira, assim como o problema do vetor curto, que será utilizado nos diferentes algoritmos propostos neste trabalho.

Capítulo 3 É dedicado ao estudo mais detalhado das ideias apresentadas em [1, 2] para gerar limites inferiores para um problema de programação quadrática inteira irrestrito com a função objetivo estritamente convexa. Também apresentamos uma proposta de algoritmos alternativos utilizando as ideias apresentadas em [1], para calcular melhores limites para o mesmo problema.

Capítulo 4 Foca-se na análise e extensão dos principais resultados teóricos para o

caso em que a função objetivo é convexa (não necessariamente estritamente convexa). Apresentamos também uma análise numérica comparativa dos algoritmos propostos neste capítulo e as propostas dadas em [1, 2].

Capítulo 5 Neste capítulo mostramos como as ideias podem ser aplicadas ao caso onde a função objetivo é não convexa com a adição de restrições de caixa. Resultados numéricos são apresentados para provar a significância das metodologias propostas no presente trabalho.

Capítulo 2

Preliminares

Neste capítulo apresentamos conceitos, definições e propriedades relacionados ao conteúdo da tese. Descrevemos o problema de programação quadrática inteira, assim como o algoritmo para calcular uma base reduzida de um reticulado, algoritmo que é utilizado nos algoritmos propostos neste trabalho. As principais referências utilizadas foram: [1–4, 9–14].

2.1 Descrição do problema

A Programação Quadrática Inteira tem atraído muita atenção recentemente. Embora a pesquisa intensa tenha levado a um enorme progresso na solução prática de programas lineares inteiros nas últimas décadas [9], métodos práticos para o caso não linear ainda são raros [10]. Isso é verdade mesmo em casos especiais, tais como os problemas de Programação Quadrática Inteira Estritamente Convexa (*PQIEC*).

De forma geral, os diferentes problemas de Programação Quadrática Inteira (*PQI*) que serão abordada neste trabalho são apresentados como:

$$\begin{cases} \min & q(x) = x^T Q x + c^T x \\ \text{s.a.} & x \in X \cap \mathbb{Z}^n. \end{cases} \quad (2.1)$$

Ao longo do trabalho, a matriz Q assume diferentes condições, sendo definida positiva ($Q \in S_{++}^n$, $Q \succ 0$) quando a função objetivo é estritamente convexa, ou semidefinida positiva ($Q \in S_+^n$, $Q \succeq 0$) quando a função objetivo é convexa. Já no caso não convexo, a matriz Q é indefinida ($Q \in S^n$).

Além disso, assumimos que $X \subset \mathbb{R}^n$ é um conjunto poliédrico limitado e $c \in \mathbb{R}^n$.

2.2 Notações e definições

Dado o problema quadrático inteiro (2.1), ao tirar as restrições de integralidade nas variáveis obtemos o problema relaxado contínuo, definido da seguinte maneira

$$\begin{cases} \min & q(x) := x^T Q x + c^T x \\ \text{s.a.} & x \in X. \end{cases} \quad (2.2)$$

O conjunto das soluções ótimas do problema quadrático inteiro (2.1) é denotada por X^* e definido como,

$$X^* := \{x^* \in X \cap \mathbb{Z}^n : q(x^*) \leq q(y), \forall y \in X \cap \mathbb{Z}^n\}. \quad (2.3)$$

Além disso, o conjunto das soluções ótimas do problema relaxado (2.2) é denotada por \bar{X} e definida como,

$$\bar{X} := \{\bar{x} \in X : q(\bar{x}) \leq q(y), \forall y \in X\}. \quad (2.4)$$

Os resultados obtidos em [2–4], que são objeto de estudo, serão apresentados nos próximos capítulos, sendo necessária a apresentação das seguintes hipóteses, assim como algumas notações

Hipótese 2.2.1. (i) O conjunto X é convexo.

(ii) Os conjuntos X^* e \bar{X} são não vazios.

(iii) Q é simétrica definida positiva ($Q \succ 0$).

A hipótese ($Q \succ 0$), garante que a função objetivo q é estritamente convexa e portanto tem um único minimizador contínuo. Já no caso convexo, está hipótese é relaxada.

Outro conceito que utilizamos ao longo do trabalho é o de elipsoide, apresentado a seguir:

Notação 2.2.1. (i) Sejam $Q' \succcurlyeq 0$, e $x' \in \mathbb{R}^n$. Define-se o elipsoide com centro em x' por

$$E(Q', x') := \{x \in \mathbb{R}^n : (x - x')^T Q' (x - x') \leq 1\}, \quad (2.5)$$

que é a translação do elipsoide $E(Q) := E(Q, 0)$ por x' .

(ii) Seja $\alpha \geq 0$. O elipsoide $\alpha E(Q', x')$ denota o elipsoide $E(Q', x')$ escalado por

α com respeito ao centro x' , isto é:

$$\begin{aligned}\alpha E(Q', x') &:= x' + \alpha E(Q') = \{x' + \alpha x : x \in E(Q')\} \\ &= \{x \in \mathbb{R}^n : (x - x')^T Q' (x - x') \leq \alpha^2\}.\end{aligned}\quad (2.6)$$

(iii) Qualquer bola euclidiana escalada é um caso especial de elipsoide. Dados $x \in \mathbb{R}^n$ e $\alpha \in \mathbb{R}_+$, define-se

$$B(x, \alpha) := \alpha E(I_n, x)$$

Para os elipsoides escalados e transladados, utilizamos a seguinte relação:

$$\alpha E(Q, x) = x + \alpha E(Q) = E\left(\frac{1}{\alpha^2} Q, x\right).\quad (2.7)$$

Notação 2.2.2. [3] Dado $x \in \mathbb{R}^n$, o conjunto dos pontos arredondados de x é definido da seguinte maneira,

$$\text{Round}(x) := \{y \in \mathbb{Z}^n : y_i \in \{\lfloor x_i \rfloor, \lceil x_i \rceil\}, \forall i \in \{1, \dots, n\}\}\quad (2.8)$$

e o ponto arredondado mais próximo de x é denotado por

$$\lfloor x \rfloor := \{y \in \mathbb{Z}^n : y_i = \lfloor x_i \rfloor, \forall i \in \{1, \dots, n\}\},\quad (2.9)$$

onde $\lfloor a \rfloor := \arg \min_{y \in \mathbb{Z}} |y - a|$, onde desempates são realizados pela regra do arredondamento para cima, ou seja, se $\arg \min_{y \in \mathbb{Z}} |y - a| = \{\lfloor a \rfloor, \lceil a \rceil\}$, então $\lfloor a \rfloor := \lceil a \rceil$.

A Figura 2.1 mostra um exemplo do conjunto $\text{Round}(\bar{x})$, onde $\bar{x} \in \mathbb{R}^3$.

Outro conceito utilizado neste trabalho é o dos conjuntos de subnível da função objetivo do problema (2.1), denotado como segue,

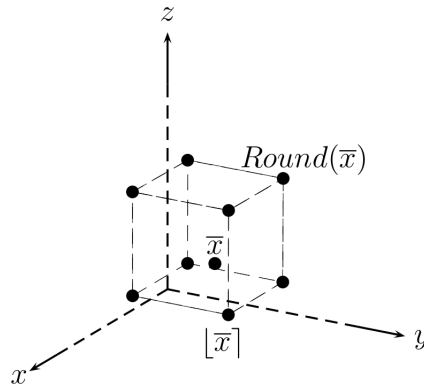


Figura 2.1: Os vértices do cubo pertencem ao conjunto $\text{Round}(\bar{x}) \subseteq \mathbb{R}^3$.

Notação 2.2.3. [3] Seja a função objetivo $q : X \rightarrow \mathbb{R}$, denotamos por $\mathcal{L}_{\leq, q}^X(z)$ ao conjunto de subnível da função q em X correspondente a $z \in \mathbb{R}$, definido da seguinte maneira:

$$\mathcal{L}_{\leq, q}^X(z) := \{x \in X : q(x) \leq z\}. \quad (2.10)$$

Observação 2.2.1. Hamacher e Nickel em [15] mostram a seguinte reformulação do problema (2.1) utilizando os conjuntos de subnível da função objetivo q ,

$$\begin{cases} \min & z \\ \text{s.a.} & \mathcal{L}_{\leq, q}^X(z) \cap \mathbb{Z}^n \neq \emptyset. \end{cases} \quad (2.11)$$

Note que esta formulação significa encontrar o menor subnível z tal que há pelo menos um ponto viável no conjunto de subnível correspondente. Em outras palavras, a solução ótima do problema (2.11) é também solução do problema (2.1). Na presente pesquisa, fazemos um estudo baseado nesta abordagem, utilizando os conjuntos de subnível para encontrar limites inferiores para o problema (2.1).

2.3 Redução de base de um Reticulado

Nesta seção apresentamos algumas definições e propriedades da teoria de reticulados que serão utilizados nos algoritmos propostos no presente trabalho, baseados em [11–14].

Definição 2.3.1. Um reticulado é uma estrutura $\Lambda = (\Lambda, R)$, tal que Λ é parcialmente ordenado segundo a relação R e para cada dois elementos $a, b \in \Lambda$, existe o supremo (menor limite superior) e o ínfimo (maior limite inferior) de $\{a, b\}$.

Nesta seção vamos detalhar o reticulado dos números inteiros, $\Lambda = (H \subseteq \mathbb{Z}^n, \leq)$

Definição 2.3.2. Dado um conjunto de vetores $\{b_1, b_2, \dots, b_m\}$ em \mathbb{R}^n , dizemos que os vetores são linearmente independentes se a equação

$$\alpha_1 b_1 + \alpha_2 b_2 + \dots + \alpha_m b_m = 0 \quad (\alpha_i \in R) \quad (2.12)$$

aceita só como solução trivial $\alpha_1 = \alpha_2 = \dots = \alpha_m = 0$

Definição 2.3.3. Um subespaço de \mathbb{R}^n é um conjunto H que tem as seguintes propriedades:

1. O vetor nulo $\mathbf{0}$ é um elemento de H
2. H é fechado para a adição: para qualquer $u, v \in H$, temos $u + v \in H$

3. H é fechado para a multiplicação por um escalar: para qualquer $u \in H$, e $k \in \mathbb{R}$, o vetor $ku \in H$

Definição 2.3.4. Uma base B do subespaço H de \mathbb{R}^n é um conjunto de vetores linearmente independentes em \mathbb{R}^n que gera H

$$B = \{b_1, b_2, \dots, b_m\}, \text{ onde } b_i \in \mathbb{R}^n \quad (2.13)$$

$$H = \sum_{i=1}^m \mathbb{R}b_i := \left\{ \sum_{i=1}^m \alpha_i b_i; \alpha_i \in \mathbb{R}; b_i \in B \right\} \quad (2.14)$$

De modo geral, um reticulado é um conjunto de pontos discretos que representam combinações lineares inteiras de vetores linearmente independentes, e o conjunto de vetores linearmente independentes que geram um reticulado é chamado de base para o reticulado.

Definição 2.3.5. Uma reticulado Λ é um subgrupo discreto de H gerado pelas combinações dos vetores da mesma base B

$$\Lambda = \sum_{i=1}^m \mathbb{Z}b_i := \left\{ \sum_{i=1}^m z_i b_i; z_i \in \mathbb{Z}; b_i \in B \right\} \quad (2.15)$$

Na Figura 2.2, apresentam-se alguns exemplos de bases em \mathbb{R}^2 de reticulados.

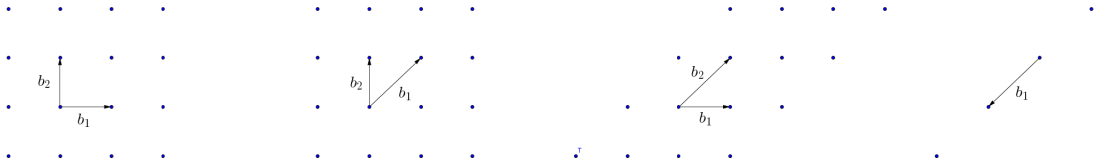


Figura 2.2: Exemplos de retículos gerado com diferentes bases em \mathbb{R}^2 .

Definição 2.3.6. O conjunto

$$\mathcal{R} = \{x \in \mathbb{R}^n : x = \alpha_1 b_1 + \dots + \alpha_m b_m, 0 \leq \alpha_i < 1\},$$

formado pelos vetores da base do reticulado Λ é chamado paralelepípedo fundamental do reticulado.

Definição 2.3.7. A matriz

$$M = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ & & \ddots & \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{pmatrix}$$

é chamada uma matriz geradora ou base do reticulado Λ .

Duas matrizes M e \tilde{M} geram o mesmo reticulado se, e somente se, $M = H\tilde{M}$ onde $H \in \mathbb{Z}^{n \times n}$ e $\det(H) = \pm 1$. A matriz $G = MM^T$ é chamada uma **matriz de Gram** para o reticulado.

Como M contém em suas linhas os vetores da base do reticulado $\{b_i\}_{i=1}^m$, a (i, j) -ésima entrada da matriz G é o produto interno $\langle b_i, b_j \rangle = b_i b_j^T$.

Exemplo 2.3.1. Seja

$$M = \begin{pmatrix} 2 & 0 \\ 2.7 & 0.7 \end{pmatrix},$$

uma matriz geradora para o reticulado Λ , como se mostra na Figura (2.3).

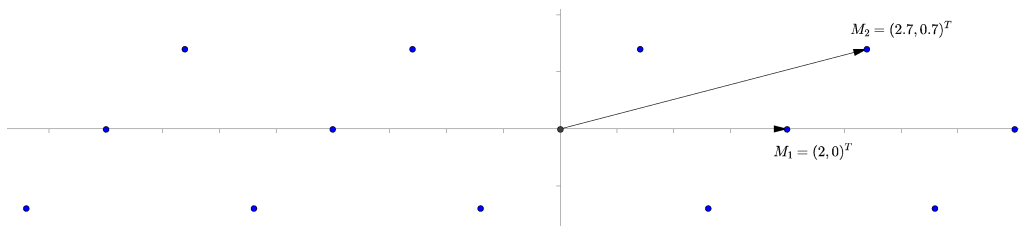


Figura 2.3: Pontos do reticulado Λ gerado por M .

Como o conjunto de pontos do reticulado não determinam de forma única uma base. Por exemplo, a matriz

$$\tilde{M} = \begin{pmatrix} -0.7 & -0.7 \\ 1.3 & -0.7 \end{pmatrix},$$

gera o mesmo reticulado, como se mostra na Figura (2.4)

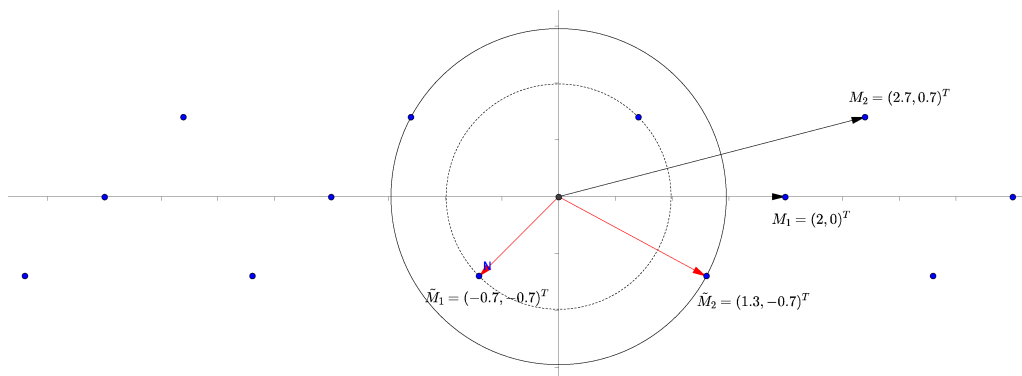


Figura 2.4: Pontos do reticulado Λ gerado por \tilde{M} .

Além disso, temos que, $M = H\tilde{M}$, onde

$$H = \begin{pmatrix} 1 & 2 \\ -1 & -1 \end{pmatrix}$$

Definição 2.3.8. O determinante do reticulado Λ é o volume da bola $B \subseteq \mathbb{R}^m$, onde m é o posto de Λ , dividido pelo número de elementos que pertencem ao reticulado $\Lambda \cap B$ quando o raio de B tende para o infinito.

$$\det(\Lambda) := \lim_{r \rightarrow \infty} \frac{\text{vol}(B(r, m))}{\#\{y \in \Lambda, \|y\| \leq r\}}.$$

Alternativamente, $\det(\Lambda) = \det(G)$, além disso temos que o determinante é um invariante do reticulado, pois não depende da escolha da base, propriedade muito importante.

Teorema 2.3.1. [11] Dado um reticulado Λ com uma base $\{b_1, b_2, \dots, b_m\}$ o determinante do reticulado é igual ao volume do paralelepípedo gerado por b_1, \dots, b_m .

Na Figura 2.5, apresenta-se um exemplo do paralelepípedo gerado pelas colunas da matriz M descrita no exemplo (2.3.1), onde o $\det(\Lambda)$ representa a área do paralelepípedo.

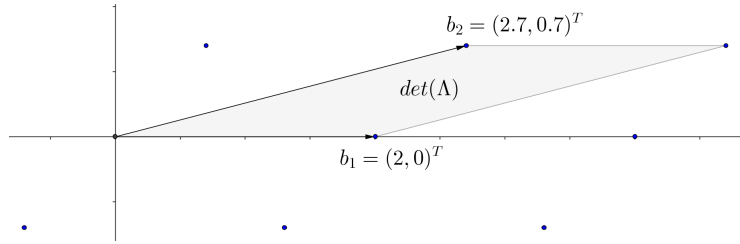


Figura 2.5: Pontos do reticulado Λ gerado por M .

Observação 2.3.1. Um reticulado Λ pode ser gerado por muitas bases diferentes, porém entre todas elas, algumas são especiais. Aquelas cujos elementos são, de certa forma, os menores possíveis no sentido da norma, são chamadas reduzidas.

Os algoritmos utilizados para obter uma base reduzida a partir de uma base qualquer são denominados algoritmos de redução de base. Neste trabalho utilizamos o algoritmo de redução denominado *LLL* (*Lenstra Lenstra Lovász*) de tempo polinomial, proposto por Arjen Lenstra, Hendrik Lenstra e László Lovász em 1982, que resolve o problema dos vetores curtos, descrito a seguir.

2.3.1 Problema dos vetores curtos

Dado um retículo Λ , o problema do vetor mais curto consiste em encontrar um vetor $v \in \Lambda$ com comprimento mínimo no reticulado. Em outras palavras, o algoritmo deve produzir um vetor não nulo v , que soluciona:

$$\xi(\Lambda) = \min_{v \in \Lambda \setminus \{0\}} \|v\|_2,$$

O $\xi(\Lambda)$ representa o comprimento do vetor mais curto.

O principal resultado teórico sobre vetores curtos é o teorema de Minkowski que nos dá um limite superior para o problema do vetor mais curto.

Teorema 2.3.2. *(Teorema de Minkowski's)[13] Dado um reticulado Λ de posto m , se $\xi(\Lambda)$ é a norma do vetor curto, então:*

$$\xi(\Lambda) \leq \frac{2}{\sqrt{\pi}} \frac{m!^{\frac{1}{m}}}{2} \det(\Lambda^{\frac{1}{m}})$$

2.3.2 Redução de base

A ideia da redução de base é transformar uma base B de um reticulado Λ em uma base curta B' tal que Λ permanece o mesmo. Para fazer isso, podemos usar as seguintes operações:

1. Trocando dois vetores da base. Isto é, de $B = \{b_1, \dots, b_i, \dots, b_j, \dots, b_m\}$ segue $B' = \{b_1, \dots, b_j, \dots, b_i, \dots, b_m\}$. Portanto, como a troca altera apenas a ordem dos vetores na base é trivial que Λ não seja afetado.
2. Substituindo b_j por $-b_j$. É trivial que Λ não seja afetado.
3. Adicionando (ou subtraindo) a um vetor b_j uma combinação linear e discreta dos outros vetores da base. O reticulado Λ não é afetado porque se tomarmos um vetor arbitrário v que pertence a Λ podemos expressar o mesmo como uma combinação discreta dos vetores da base $v = \sum_{i=1}^m z_i b_i$ e se substituirmos b_j pela combinação discreta dos outros vetores da base, $b_j \leftarrow b_j + \sum_{i \neq j} y_i b_i$ ainda podemos expressar v como uma combinação discreta dos vetores da nova base

$$v = \sum_{i \neq j} z_i b_i + z_j (b_j - \sum_{i \neq j} y_i b_i).$$

De maneira similar podemos mostrar que se v não pertence ao reticulado Λ , então não podemos expressá-lo com uma combinação discreta da nova base. Segue-se que as 2 bases geram exatamente o mesmo reticulado.

Na Figura 2.6 apresenta-se um exemplo de uma base e sua base reduzida.

Quando o posto de uma base reduzida é dois, temos que b_1 é o vetor mais curto do reticulado Λ e podemos obtê-lo em tempo polinomial. Mas para postos superiores não existe nenhum algoritmo conhecido que encontre a base reduzida em tempo polinomial. O algoritmo de redução de base *LLL* encontra um vetor em tempo polinomial e é muitas vezes suficiente para aplicações sendo o algoritmo que utilizamos em nossas implementações.

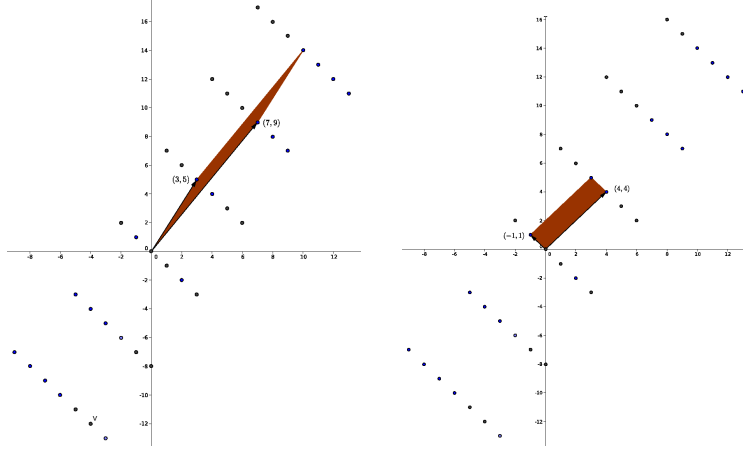


Figura 2.6: Reticulado de posto 2 com duas bases diferentes. A segunda é a base reduzida e ortogonal.

A grosso modo, o algoritmo *LLL* realiza projeções ortogonais sucessivas, se necessário, a fim de obter uma base reduzida ou quase ortogonal. Os autores em [12, 13], propõem o Algoritmo (1) de redução de base de um reticulado, que foi implementado para ser utilizado em nossas propostas.

Segundo o Algoritmo (1), dada uma base $\{b_1, b_2, \dots, b_m\}$ do reticulado Λ , gera-se uma base ortogonal $\{b_1^*, b_2^*, \dots, b_m^*\}$ pelo processo de Gram-Schmidt, e cada vetor $\mu_i \in \mathbb{R}^m$ representa os coeficientes da projeção ortogonal utilizadas no processo de Gram-Schmidt, dada por $\mu_i[j] := (b_i \cdot b_j^*) / (b_j^* \cdot b_j^*)$.

O Algoritmo (1) está baseado nos seguintes teoremas, que estabelecem as condições para a convergência do algoritmo em tempo polinomial.

Teorema 2.3.3. *Dado o reticulado Λ gerado pela base $\{b_1, \dots, b_m\}$ e a base ortogonal $\{b_1^*, \dots, b_m^*\}$ obtida pelo método de Gram-Schmidt, então:*

$$\det(\Lambda) = \prod_{i=1}^m \|b_i^*\|. \quad (2.16)$$

Definição 2.3.9 (Base *c*-reduzida). *Uma base $\{b_1, \dots, b_m\}$ é dita *c*-reduzida, se, e somente, se os vetores da base ortogonal obtida pelo processo de Gram-Schmidt $\{b_1^*, \dots, b_m^*\}$ satisfazem as seguinte desigualdade para $i = 1, \dots, m - 1$*

$$\|b_{i+1}^*\| \geq \frac{\|b_i^*\|^2}{c}. \quad (2.17)$$

Teorema 2.3.4 (Quase-ortogonalidade de uma base *c*-reduzida). *Dado o reticulado Λ , a base *c*-reduzida, $\{b_1, \dots, b_m\}$, com $c \geq \frac{3}{4}$ é quase-ortogonal no sentido que:*

$$\prod_{i=1}^m \|b_i\| \leq c^{\frac{m(m-1)}{4}} \det(\Lambda). \quad (2.18)$$

O seguinte teorema, estabelece que, quando $c \geq \frac{3}{4}$, o Algoritmo (1) obtêm uma base c -reduzida em tempo polinomial.

Teorema 2.3.5. *Para $c \geq \frac{3}{4}$ LLL encontra uma base c -reduzida em tempo polinomial.*

```

1 Input : base  $\{b_1, b_2, \dots, b_m\}$ , e  $c$  tal que  $b_i \in \mathbb{R}^n$  e  $c \geq \frac{3}{4}$ .
   Data: Base ortogonal  $\{b_1^*, b_2^*, \dots, b_m^*\}$ , e vetores dos coeficientes da projeção
           ortogonal  $\{\mu_1, \dots, \mu_m\}$ , tal que  $b_i^* \in \mathbb{R}^n$  e  $\mu_i \in \mathbb{R}^m$ 
2 for  $i \leftarrow 1$  to  $m$  do
3    $\mu_i := 0$ ;
4    $\mu_i[i] := 1$ ;
5    $b_i^* = b_i$ ;
6   for  $j \leftarrow 1$  to  $i-1$  do
7      $\mu_i[j] := (b_i \cdot b_j^*) / (b_j^* \cdot b_j^*)$ ;
8      $b_i^* := b_i^* - \mu_i[j] b_j^*$ ;
9    $reduce(i)$ 
10 while  $i \leftarrow 1$  to  $m$  do
11   if  $\|b_i^*\|^2 \leq c \|b_{i+1}^*\|^2$  then
12      $i := i + 1$ ;
13   else
14     Modifica  $Q$  e  $R$  com o fim de ter a relação  $B = QR$  depois da divisão.
15      $b_{i+1}^* := b_{i+1}^* + \mu_{i+1}[i] b_i^*$ 
16      $\mu_i[i] := (b_i \cdot b_{i+1}^*) / (b_{i+1}^* \cdot b_{i+1}^*)$ 
17      $\mu_i[i+1] := 1$ 
18      $\mu_{i+1}[i] := 1$ 
19      $\mu_{i+1}[i+1] := 0$ 
20      $b_i^* := b_i^* - \mu_i[i] b_{i+1}^*$ 
21      $swap(\mu_i, \mu_{i+1})$ 
22      $swap(b_i^*, b_{i+1}^*)$ 
23      $swap(b_i, b_{i+1})$ 
24     for  $k \leftarrow i+2$  to  $m$  do
25        $\mu_k[i] := (b_k \cdot b_i^*) / (b_i^* \cdot b_i^*)$ ;
26        $\mu_k[i+1] := (b_k \cdot b_{i+1}^*) / (b_{i+1}^* \cdot b_{i+1}^*)$ ;
27     if  $|\mu_{i+1}[i]| > \frac{1}{2}$  then
28        $reduce(i+1)$ ;
29      $i = \max(i-1, 1)$ ;
30 Subroutine :  $reduce$ 
31 Input :  $i \leq m$ 
32 while  $j \leftarrow (i-1) > 0$  do
33    $b_i = b_i - round(\mu_i[j]) b_j$ 
34    $\mu_i = \mu_i - round(\mu_i[j]) \mu_j$ 
35    $j = j - 1$ 
36 return base reduzida  $\{b_1, b_2, \dots, b_m\}$ ;

```

Algorithm 1: Algoritmo de redução de base *LLL*

Capítulo 3

Limite elipsoidal para programação quadrática estritamente convexa

Neste capítulo, consideramos o problema (2.1) no caso estritamente convexo e irrestrito. Isto é, a matriz Q é definida positiva ($Q \succ 0$) e o conjunto $X = \mathbb{R}^n$. Desta forma o problema é representado por:

$$\begin{cases} \min & q(x) = x^T Q x + c^T x \\ \text{s.a.} & x \in \mathbb{Z}^n. \end{cases} \quad (3.1)$$

Na Seção 3.1 é apresentada a metodologia de obtenção de limites elipsoidais proposta em [2], já nas seções 3.2 e 3.3 são discutidos os resultados propostos em [1, 2]. Uma primeira contribuição é apresentada na Seção 3.4, na Seção 3.5 são propostos mais três algoritmos para gerar limites inferiores do problema (3.1). Finalmente os resultados numéricos são apresentados na Seção 3.6.

3.1 Limite elipsoidal

Nesta seção, apresentamos uma revisão da pesquisa desenvolvida que servirá como referência para a discussão dos resultados obtidos no presente capítulo.

Os autores em [1–4] propõem uma metodologia focada nos conjuntos de subnível da função objetivo q para a obtenção de limites inferiores para o problema (3.1) no caso irrestrito.

Os minimizadores contínuos são obtidos com a solução do problema relaxado (2.1), ao fazer $X = \mathbb{R}^n$. Portanto, a relaxação contínua do problema (3.1), é escrita

como:

$$\begin{cases} \min & q(x) = x^T Q x + c^T x \\ \text{s.a.} & x \in \mathbb{R}^n \end{cases} \quad (3.2)$$

O problema (3.1) pode ser resolvido por uma abordagem *branch-and-bound*, tal como menciona Buchheim et al. [1, 2], onde os limites inferiores podem ser obtidos a partir dos minimizadores contínuos. Dado que $Q \succ 0$ temos um mínimo contínuo único \bar{x} de q em \mathbb{R}^n que é facilmente calculado, gerando um limite inferior que é, naturalmente, válido também para qualquer conjunto X .

O limite inferior pode ser melhorado ao levar-se em conta as restrições de integralidade, utilizando-se um reticulado de elipsoides. Sua proposta é determinar um elipsoide auxiliar $E(Q', \bar{x})$ centrado no mínimo contínuo \bar{x} e calcular o valor do parâmetro de escalonamento α tal que o elipsoide escalado $\alpha E(Q', \bar{x})$ contém pelo menos um ponto inteiro em sua fronteira e nenhum ponto inteiro no seu interior.

O cálculo do parâmetro α pode ser feito rapidamente, se o elipsoide $E(Q', \bar{x})$ é escolhido apropriadamente. Assim, encontrar o mínimo inteiro da função q ao longo da fronteira do elipsoide $\alpha E(Q', \bar{x})$ gera um limite inferior maior do que o gerado pelo mínimo contínuo \bar{x} .

A ideia de utilizar aproximações elipsoidais também tem sido utilizada em otimização não convexa: Bienstock [16] considera problemas convexos com restrições não convexas, enquanto na otimização inteira quadrática não convexa com restrições de caixa é utilizada por Buchheim et al. [17]

O lema 3.1.1 é um resultado muito importante, uma vez que a partir dele são construídos os resultados sobre os limites inferiores elipsoidais.

Lema 3.1.1. [1] *Seja Q definida positiva, o mínimo contínuo \bar{x} de $q(x) := x^T Q x + c^T x$ é único e dado por $\bar{x} = -\frac{1}{2} Q^{-1} c$, e $q(\bar{x}) = -\frac{1}{4} c^T Q^{-1} c$. Além disso, para qualquer $x \in \mathbb{R}^n$, $q(x) = q(\bar{x}) + (x - \bar{x})^T Q (x - \bar{x})$.*

Prova. *Utilizando as condições de otimalidade e o fato de Q ser definida positiva, temos que $\nabla q(\bar{x}) = 2Q\bar{x} + c = 0$. Ao isolar \bar{x} , obtemos $\bar{x} = -\frac{1}{2} Q^{-1} c$.*

Além disso, temos

$$\begin{aligned} q(\bar{x}) &= \bar{x}^T Q \bar{x} + c^T \bar{x} \\ &= \bar{x}^T \left(-\frac{1}{2} c\right) + c^T \bar{x} \\ &= \frac{1}{2} c^T \bar{x} \\ &= \frac{1}{2} c^T \left(-\frac{1}{2} Q^{-1} c\right) \\ &= -\frac{1}{4} c^T Q^{-1} c \end{aligned}$$

Por outro lado, seja $x \in \mathbb{R}^n$, temos

$$\begin{aligned}
q(\bar{x}) + (x - \bar{x})^T Q(x - \bar{x}) &= -\frac{1}{4}c^T Q^{-1}c + x^T Qx + c^T x - \frac{1}{2}c^T \bar{x} \\
&= \frac{1}{4}c^T Q^{-1}c + x^T Qx + c^T x + \frac{1}{4}c^T Q^{-1}c \\
&= x^T Qx + c^T x \\
&= q(x)
\end{aligned}$$

Portanto, $q(x) = q(\bar{x}) + (x - \bar{x})^T Q(x - \bar{x})$, $\forall x \in \mathbb{R}^n$ e $\bar{x} \in \bar{X}$, definido em (2.4). ■

Como foi explicado anteriormente, a ideia é escalar o elipsoide $E(Q, \bar{x})$ até atingir o primeiro ponto inteiro na sua fronteira, este fator de escalonamento é denotado como $\mu(Q, \bar{x})$, e definido da seguinte maneira:

$$\mu(Q, \bar{x}) := \sup\{\alpha : \alpha E(Q, \bar{x}) \cap \mathbb{Z}^n = \emptyset\} \quad (3.3)$$

$$= \min\{\alpha : \alpha E(Q, \bar{x}) \cap \mathbb{Z}^n \neq \emptyset\} \quad (3.4)$$

Consequentemente, é possível escrever o fator de escalonamento $\mu(Q, \bar{x})$ da seguinte maneira:

$$\mu(Q, \bar{x}) = \min\{\sqrt{(x - \bar{x})^T Q(x - \bar{x})} : x \in \mathbb{Z}^n\} \quad (3.5)$$

$$= \max\{\alpha : (x - \bar{x})^T Q(x - \bar{x}) \geq \alpha^2, x \in \mathbb{Z}^n\} \quad (3.6)$$

A Figura (3.1) mostra geometricamente o processo de escalonamento do elipsoide $E(Q, \bar{x})$, com $x^* \in \mathbb{Z}^n$.

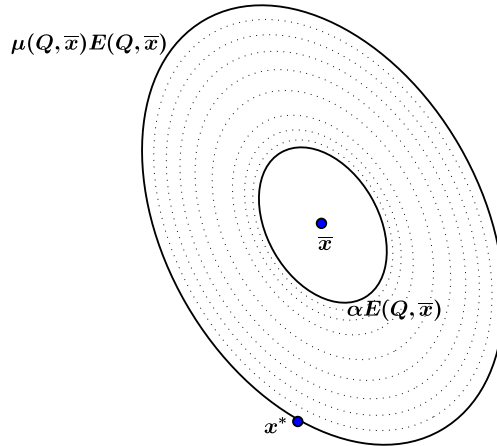


Figura 3.1: Escalação do elipsoide $E(Q, \bar{x})$ pelo fator α até atingir o ponto inteiro x^* .

Assim, dada a função objetivo $q(x) := x^T Qx + c^T x$, e seu minimizador contínuo

\bar{x} , podemos ver que os conjuntos de subnível da função q incluem as fronteiras dos elipsoides da forma $\alpha E(Q, \bar{x})$ como mostra-se na Figura (3.1). Sendo assim, o fato de encontrar o ponto inteiro que minimiza q é equivalente a escalar o elipsoide $E(Q, \bar{x})$ por α , iniciando com $\alpha = 0$ e parando quando a fronteira do elipsoide $\alpha E(Q, \bar{x})$ contem um ponto inteiro, o que é equivalente ao cálculo de $\mu(Q, \bar{x})$.

Porém, encontrar o valor de $\mu(Q, \bar{x})$ é tão difícil quanto resolver o problema irrestrito original (3.1). Por tal motivo, os autores propõem escalar um outro elipsoide $E(Q', \bar{x})$ que contem o elipsoide $E(Q, \bar{x})$ como se mostra na Figura (3.2), de tal maneira que ao escalar simultaneamente o elipsoide $E(Q, \bar{x})$, ele toque a borda do primeiro elipsoide. Este procedimento requer o cálculo de $\mu(Q', \bar{x})$ assim como, o máximo valor de $\alpha \in \mathbb{R}_+$ tal que $\alpha E(Q, \bar{x})$ esteja contido em $E(Q', \bar{x})$.

O valor máximo α é denotado por $\lambda(Q, Q')$, e definido como segue:

$$\lambda(Q, Q') := \max\{\alpha : \alpha E(Q) \subseteq E(Q')\}, \quad (3.7)$$

note que o valor de $\lambda(Q, Q')$ não depende de \bar{x} .

Por outro lado, o problema (3.7) pode se reescrever como:

$$\lambda(Q, Q') = \min \left\{ \frac{1}{\sqrt{x^T Q' x}} : x \in E(Q) \right\} \quad (3.8)$$

$$= \max \left\{ \alpha : Q - \alpha^2 Q' \succcurlyeq 0 \right\} \quad (3.9)$$

as provas destes resultados são dadas em [1].

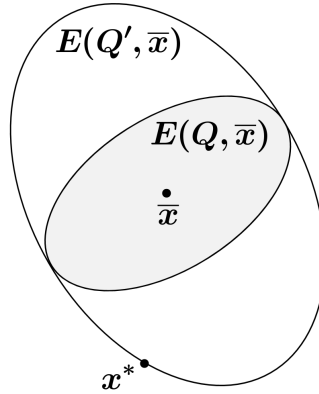


Figura 3.2: O elipsoide branco $E(Q', \bar{x})$ é escalado até atingir o primeiro ponto inteiro x^* e por sua vez o escalonamento do elipsoide cinza $E(Q, \bar{x})$ produz um limite inferior sobre $q(x^*)$.

A seguinte proposição é um dos resultados mais importantes apresentados em [1], pois mostra que é possível obter melhores limites inferiores do que os obtidos pelo minimizador contínuo \bar{x} utilizando-se elipsoides auxiliares. Desta forma, o objetivo

do estudo é a escolha dos elipsoides auxiliares tal que os valores de $\lambda(Q, Q')$ e $\mu(Q', \bar{x})$ sejam mais simples de calcular e sem perder a qualidade dos limites inferiores.

Proposição 3.1.1. [1] Seja $q(x) := x^T Q x + c^T x$ com $Q \succ 0$, seu mínimo contínuo \bar{x} e a matriz $Q' \succcurlyeq 0$. Temos que,

$$\min \{q(x) : x \in \mathbb{Z}^n\} \geq q(\bar{x}) + \lambda^2(Q, Q') \mu^2(Q', \bar{x}) \quad (3.10)$$

Prova. Utilizando a definição (3.9) de $\lambda(Q, Q')$, temos $Q - \lambda^2(Q, Q') Q' \succcurlyeq 0$. Por conseguinte, $\forall x \in \mathbb{R}^n$, $(x - \bar{x})^T (Q - \lambda^2(Q, Q') Q') (x - \bar{x}) \geq 0$, de maneira que,

$$\begin{aligned} (x - \bar{x})^T Q (x - \bar{x}) &\geq (x - \bar{x})^T (\lambda^2(Q, Q') Q') (x - \bar{x}) \\ (x - \bar{x})^T Q (x - \bar{x}) &\geq \lambda^2(Q, Q') (x - \bar{x})^T Q' (x - \bar{x}) \\ (x - \bar{x})^T Q (x - \bar{x}) &\geq \lambda^2(Q, Q') \mu^2(Q', \bar{x}), \forall x \in \mathbb{Z}^n. \end{aligned}$$

Portanto, $q(\bar{x}) + (x - \bar{x})^T Q (x - \bar{x}) \geq q(\bar{x}) + \lambda^2(Q, Q') \mu^2(Q', \bar{x})$, $\forall x \in \mathbb{Z}^n$ ■

Definição 3.1.1. Utilizando o resultado na desigualdade (3.10), define-se o limite inferior dada pela a função quadrática subestimadora da função objetivo q , como:

$$LB := q(\bar{x}) + \lambda^2(Q, Q') \mu^2(Q', \bar{x}) \quad (3.11)$$

tal que

$$\min \{q(x) : x \in \mathbb{Z}^n\} \geq LB \quad (3.12)$$

Notação 3.1.1. [3] Se $Q = \text{Diag}(\tau)$, $\tau \geq 0$, o elipsoide $E(\text{Diag}(\tau), \bar{x})$ é chamado de eixo paralelo. Isto é, suas direções principais são paralelas aos eixos cartesianos.

Lema 3.1.2. Se $Q' = \text{Diag}(\tau)$, para algum $\tau \in \mathbb{R}_+^n$, então o primeiro ponto inteiro atingido ao escalar o elipsoide $E(\text{Diag}(\tau), \bar{x})$ é o ponto $\lfloor \bar{x} \rfloor$, para qualquer $\bar{x} \in \bar{X}$ (Ou um dos primeiros pontos, no caso de haver mais de um).

Prova. Dado que, a matriz $Q' = \text{Diag}(\tau)$ é diagonal, em primeiro lugar vamos a simplificar a expressão $(x - \bar{x})^T Q (x - \bar{x})$ na definição de $\mu(Q, \bar{x})$ em (3.5). Assim obtemos:

$$\begin{aligned} (x - \bar{x})^T \text{Diag}(\tau) (x - \bar{x}) &= (x_r - \bar{x}_r)^T \left(\sum_{i=1}^n \tau_i e_i e_i^T \right) (x_r - \bar{x}_r) \\ &= \sum_{i=1}^n \tau_i \left[e_i^T (x - \bar{x}) \right]^2 \\ &= \sum_{i=1}^n \tau_i (x_i - \bar{x}_i)^2, \end{aligned}$$

logo,

$$\mu(\text{Diag}(\tau), \bar{x}) = \min \left\{ \sqrt{\sum_{i=1}^n \tau_i (x_i - \bar{x}_i)^2} : x \in \mathbb{Z}^n \right\}. \quad (3.13)$$

Pode-se observar diretamente que $x = \lfloor \bar{x} \rfloor$ é o ponto que minimiza a expressão (3.13). Geometricamente significa que ao escalar um elipsoide $E(\text{Diag}(\tau), \bar{x})$ o primeiro ponto inteiro atingido é o ponto $\lfloor \bar{x} \rfloor$ tal como se mostra na Figura 3.3, que também dá uma noção de proximidade do ponto $x = \lfloor \bar{x} \rfloor$ ao ponto \bar{x} .

Portanto,

$$\mu^2(\text{Diag}(\tau), \bar{x}) = \sum_{i=1}^n \tau_i (\lfloor \bar{x}_i \rfloor - \bar{x}_i)^2 \quad (3.14)$$

■

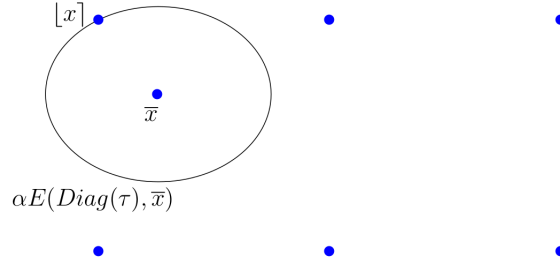


Figura 3.3: O ponto $\lfloor \bar{x} \rfloor$ é atingido pelo elipsoide $\alpha E(\text{Diag}_r(\tau), \bar{x}_r)$.

Exemplo 3.1.1. Dado o problema

$$\begin{aligned} \min \quad & q(x) = 2x_1^2 + 4x_1x_2 + 3x_2^2 + 5x_1 + 2x_2 + 7 \\ \text{s.a.} \quad & x \in \mathbb{Z}^2. \end{aligned}$$

A matriz que define o termo quadrático da função objetivo do problema é dada por:

$$Q = \begin{pmatrix} 2 & 2 \\ 2 & 3 \end{pmatrix}.$$

Além disso temos que a solução contínua é $\bar{x} = (-2.75, 1.5)$ e seu valor mínimo contínuo é $q(\bar{x}) = 1.625$. Também temos que $\lfloor \bar{x} \rfloor = (-3, 2)$ com um valor na função objetivo de $q(\lfloor \bar{x} \rfloor) = 2$, de maneira particular construímos o elipsoide eixo paralelo $E(Q', \bar{x})$, sendo

$$Q' = \text{Diag}(\tau) = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.2 \end{pmatrix}.$$

Logo,

$$\begin{aligned}\mu^2(\text{Diag}(\tau), \bar{x}) &= ([\bar{x}] - \bar{x})^T \begin{pmatrix} 0.5 & 0 \\ 0 & 0.2 \end{pmatrix} ([\bar{x}] - \bar{x}) \\ &= 0.5(-3 + 2.75)^2 + 0.2(2 - 1.5)^2 = 0.0813.\end{aligned}$$

Na figura 3.4 também pode-se observar que ao escalar o elipsoide $E(Q, \bar{x})$ até ele atingir o elipsoide $\mu E(\text{Diag}(\tau), \bar{x})$, tem-se o fator de escalonamento $\alpha^2 = 0.1$, o qual significa que, $Q - 0.1\text{Diag}(\tau) \succcurlyeq 0$.

Segundo a definição de $\lambda(Q, Q')$ o seu valor é $\lambda(Q, Q') = \alpha = \sqrt{0.1}$, e por sua vez chegamos ao fato de $\sqrt{0.1}E(Q, \bar{x}) \subseteq E(\text{Diag}(\tau), \bar{x})$. Sabemos ainda que o valor da solução inteira do problema é $q(x^*) = 2$.

A seguir verificamos o resultado da Proposição 3.1.1,

$$\begin{aligned}\min \{q(x) : x \in \mathbb{Z}^n\} &\geq q(\bar{x}) + \lambda^2(Q, Q')\mu^2(Q', \bar{x}) \\ q(x^*) &\geq 1.625 + 0.00813 = 1.63313\end{aligned}$$

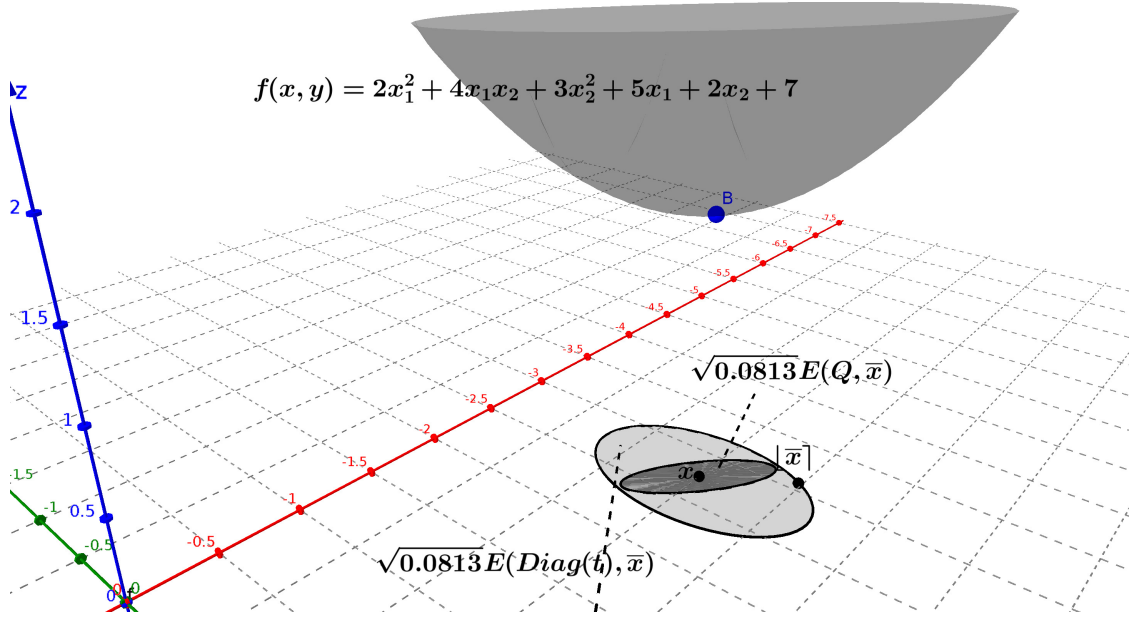


Figura 3.4: Obtenção de um limite inferior através do elipsoide eixo paralelo, $E(\text{Diag}(\tau), \bar{x})$.

No exemplo acima, foi calculado um limite inferior utilizando um elipsoide eixo paralelo como um elipsoide auxiliar. No entanto, a motivação é achar os melhores limites inferiores, e portanto, sempre vamos tentar maximizar o fator $\lambda^2(Q, Q')\mu^2(Q', \bar{x})$ da Proposição 3.1.1. Isto é, achar aquele elipsoide que maximize o limite inferior para o problema quadrático inteiro.

Observação 3.1.1. [1] Dados o mínimo contínuo $\bar{x} \in \bar{X}$ e $Q \succcurlyeq 0$, o máximo

valor de $\lambda^2(Q, \text{Diag}(\tau))\mu^2(\text{Diag}(\tau), \bar{x})$ para $\tau \in \mathbb{R}_+^n$, é obtido ao resolver o seguinte problema de programação semidefinida

$$\begin{cases} \max & \sum_{i=1}^n \tau_i (\lceil \bar{x}_i \rceil - \bar{x}_i)^2 \\ \text{s.a.} & Q \succeq \text{Diag}(\tau). \\ & \tau \geq 0. \end{cases} \quad (3.15)$$

Exemplo 3.1.2. Dado o problema do Exemplo (3.1.1), utilizamos a biblioteca computacional CVX 3.0 [18] para resolver o problema de programação semidefinida (3.15), obtendo como solução

$$\lambda^2(Q, \text{Diag}(\tau))\mu^2(\text{Diag}(\tau), \bar{x}) = 0.25,$$

com

$$\text{Diag}(\tau) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Portanto, $q(x^*) \geq 1.625 + 0.25 = 1.875$.

Na Figura 3.5, mostra-se o elipsoide eixo paralelo que gera o maior limite inferior para $q(x^*)$, que neste caso é uma parábola.

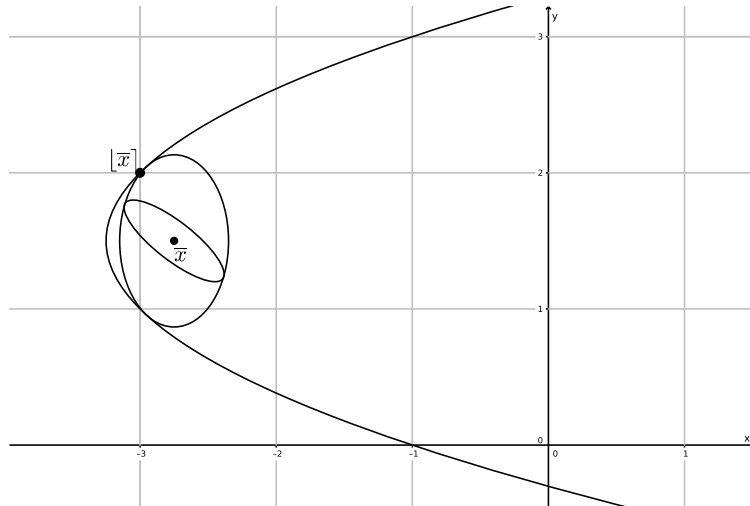


Figura 3.5: Maior limite inferior usando o elipsoide eixo paralelo $E(\text{Diag}(\tau), \bar{x})$.

3.2 Matrizes de posto 1

Em [1], os autores estudaram as matrizes de posto um, para a construção dos subestimadores q' . Dos resultados apresentados, temos que, se Q' é uma matriz semidefinida

positiva de posto um, dada por $Q' = \delta^* vv^T$, com $\delta^* \geq 0$, então

$$\begin{aligned} \min\{q'(x) : x \in \mathbb{Z}^n\} &= q(\bar{x}) + \min\{(x - \bar{x})^T Q'(x - \bar{x}) : x \in \mathbb{Z}^n\} \\ &= q(\bar{x}) + \delta^* \min\{(v^T(x - \bar{x}))^2 : x \in \mathbb{Z}^n\} \\ &= q(\bar{x}) + \delta^* \min\{(v^T x - v^T \bar{x})^2 : x \in \mathbb{Z}^n\}. \end{aligned} \quad (3.16)$$

Consequentemente, se $v \in \mathbb{Z}^n \setminus \{0\}$,

$$\min\{q'(x) : x \in \mathbb{Z}^n\} \geq q(\bar{x}) + \delta^* \left(\lfloor v^T \bar{x} \rfloor - v^T \bar{x} \right)^2. \quad (3.17)$$

Além disso, nota-se que se existe um vetor $y \in \mathbb{Z}^n$ tal que

$$v^T y = \lfloor v^T \bar{x} \rfloor, \quad (3.18)$$

este vetor é um minimizador inteiro de q' , e a desigualdade em (3.17) torna-se uma igualdade. Se uma solução inteira y para $v^T y = \lfloor v^T \bar{x} \rfloor$ pode ser facilmente calculada, então a relaxação inteira do (PQI) ainda pode ser facilmente resolvido. Se, por outro lado, o inteiro y não existe, ainda teremos um limite inferior para o (PQI) dado pela desigualdade (3.17), que nunca é pior do que o limite contínuo $q(\bar{x})$.

Finalmente, de [1] temos a seguinte proposição

Proposição 3.2.1. *Seja $Q = LL^T$, com $L \in \mathbb{R}^{n \times n}$ não singular, e $E(Q) := \{x \in \mathbb{R}^n : x^T Q x \leq 1\}$. Para um dado $v \in \mathbb{Z}^n \setminus \{0\}$,*

$$\delta^* := \max\{\delta : Q - \delta vv^T \succeq 0\} = \frac{1}{\max_{x \in E(Q)} (v^T x)^2} = \frac{1}{\|L^{-1}v\|_2^2}. \quad (3.19)$$

Considerando o resultado da Proposição 3.2.1 os autores em [1] fazem uma busca por vetores curtos no reticulado gerado pelas colunas da matriz (L^{-1}) . Os vetores curtos são calculados pela redução da base dada pelas referidas colunas. Se w é um vetor na base reduzida, um vetor $v \in \mathbb{Z}^n \setminus \{0\}$ é, portanto, dado por $v = Lw$.

Exemplo 3.2.1. *Dado a matriz Q do problema do Exemplo (3.1.1),*

$$Q = \begin{pmatrix} 2 & 2 \\ 2 & 3 \end{pmatrix},$$

sendo a solução continua $\bar{x} = (-2.75, 1.5)$.

Tomando o vetor $v = (-408/985, -1)^T$ e calculando o valor de δ^* segundo a Proposição 3.2.1, obtemos $\delta^* = 2/3$, construímos o elipsoide $E(Q', \bar{x})$, sendo $Q' = \delta^* vv^T$. Na Figura 3.6, mostra-se o elipsoide $E(Q', \bar{x})$ contendo ao elipsoide $E(Q, \bar{x})$.

Finalmente, visando melhores limites, adiciona-se matrizes de posto um da forma $\delta_i^* v_i v_i^T$, com cada vetor v_i obtido de um vetor ω_i da base reduzida. Obtendo a matriz

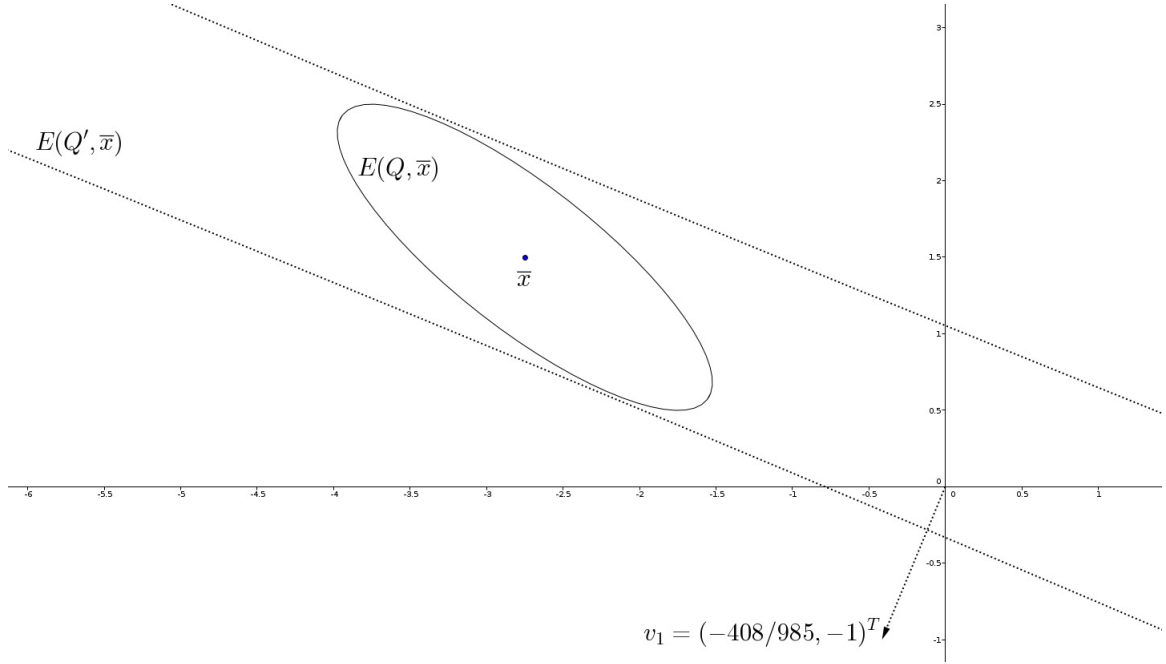


Figura 3.6: Elipsoide $E(Q', \bar{x})$ com a matriz Q de posto um.

$Q' = \sum_{i=1}^n \delta_i^* v_i v_i^T$. No entanto, para que o limite inferior seja garantido, deve-se escalar o elipsoide $E(Q', \bar{x})$ por um escalar β^* , tal que $Q - \beta^* Q' \succcurlyeq 0$. O valor de β^* pode ser obtido como a solução do seguinte problema de programação semidefinida

$$\max\{\beta : Q - \beta Q' \succcurlyeq 0, \beta \geq 0\} \quad (3.20)$$

Sendo os vetores $v_1, \dots, v_n \in \mathbb{Z}^n \setminus \{0\}$ as colunas da matriz de transformação de bases, isto é,

$$V = L^{-1}W, \quad (3.21)$$

sendo L^{-1} a base do reticulado e W a base reduzida obtida pelo Algoritmo (1). Além disso, temos que a matriz de transformação V é inteira e unimodular, isto é, o seu determinante tem valor em $\{-1, 1\}$.

Por outro lado, devido à unimodularidade da matriz de transformação V para a base original dada pelas colunas de L^{-1} à base reduzida W , é fácil ver que para os vetores selecionados v_i , o mínimo inteiro de q' ainda é facilmente calculado por

$$\min\{q'(x) : x \in \mathbb{Z}^n\} = q(\bar{x}) + \beta^* \sum_{i=1}^n \delta_i^* (\lfloor v_i^T \bar{x} \rfloor - v_i^T \bar{x})^2 \quad (3.22)$$

No restante do trabalho, fazemos referência ao algoritmo proposto em [1] para construir a matriz Q' como BCL . As principais etapas do algoritmo são descritas no Algoritmo (2).

Input: $L \in \mathbb{R}^{n \times n}$, onde $\text{posto}(L) = n$, e $Q := LL^T$
Output: Q'

- 1 **for** $i = 1, \dots, n$ **do**
- 2 Selecione v_i , tal que $w_i := L^{-1}v_i$ é o i -ésimo vetor na base reduzida para o reticulado gerado pelas colunas de L^{-1} ;
- 3 $\delta_i := \frac{1}{\|L^{-1}v_i\|_2^2}$;
- 4 $\beta^* := \max \{ \beta : Q - \beta \sum_{i=1}^n \delta_i^* v_i v_i^T \succeq 0 \}$;
- 5 $Q' := \beta^* \sum_{i=1}^n \delta_i^* v_i v_i^T$;

Algorithm 2: *BCL* ([1])

3.3 Matrizes de posto completo

Hübner em [3], apresenta uma classe especial de elipsoides chamados de quase arredondados que utilizam os pontos $\lfloor \bar{x} \rfloor$ no cálculo do fator de escalonamento μ . Mais recentemente, Buchheim et al. [2], propõem uma heurística para gerar este tipo de elipsoides quase arredondados, que detalhamos nesta seção.

Definição 3.3.1. [3] Dizemos que a função q tem a propriedade de arredondamento, se para qualquer minimizador contínuo \bar{x} de q , existe um minimizador inteiro x^* de q , tal que $x^* \in \text{Round}(\bar{x})$

Definição 3.3.2. [3] Dizemos que a função q tem a propriedade de arredondamento forte, se para qualquer minimizador contínuo \bar{x} de q , existe um minimizador inteiro x^* de q , tal que $x^* = \lfloor \bar{x} \rfloor$

Observação 3.3.1. Se $Q = \text{Diag}(\tau)$ então a função q tem a propriedade de arredondamento forte, dado que um minimizador inteiro da função q é $x^* = \lfloor \bar{x} \rfloor$, como foi mostrado no Lema 3.1.2.

A propriedade de arredondamento, também pode se relacionar com os conjuntos de subnível da função objetivo q . Na observação a seguir caracterizamos os conjuntos de subnível da função q e no Lema 3.3.1 estabelecemos a relação deles com a propriedade de arredondamento.

Observação 3.3.2. Os conjuntos de subnível da função q são da forma

$$\mathcal{L}_{\leq}(z) = \sqrt{z - q(\bar{x})} E(Q, \bar{x}), \forall z \geq q(\bar{x}). \quad (3.23)$$

De fato, da Observação 3.1.1 temos que $q(x) = q(\bar{x}) + (x - \bar{x})^T Q (x - \bar{x})$, e utilizando

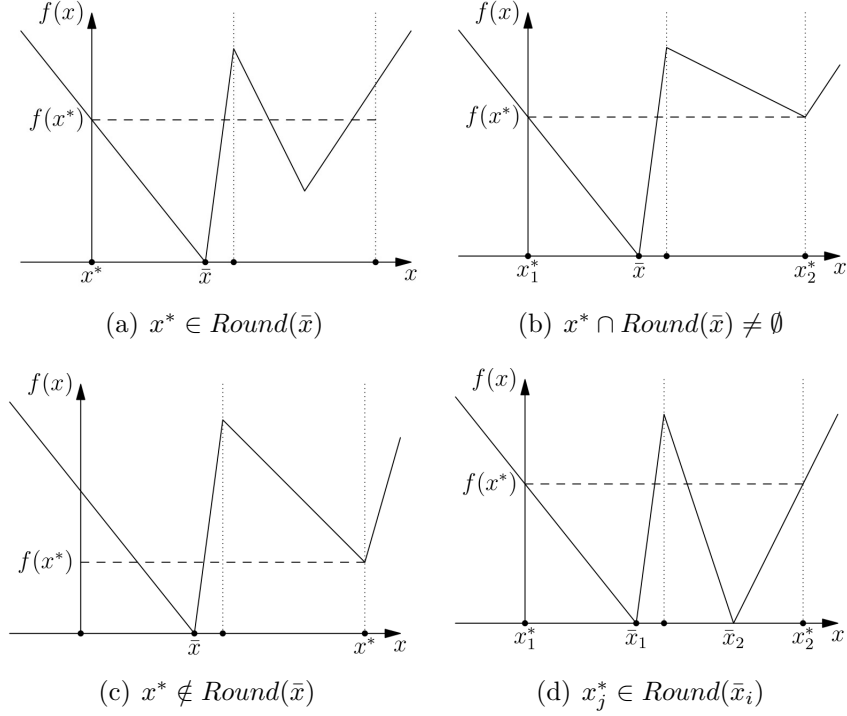


Figura 3.7: Exemplos de funções com e sem a propriedade de arredondamento. (Fonte. [3])

a definição de conjunto de subnível, temos

$$\begin{aligned}
\mathcal{L}_{\leq}(z) &:= \{x \in \mathbb{R}^n : q(x) \leq z\} \\
&= \{x \in \mathbb{R}^n : q(\bar{x}) + (x - \bar{x})^T Q(x - \bar{x}) \leq z\} \\
&= \{x \in \mathbb{R}^n : (x - \bar{x})^T Q(x - \bar{x}) \leq z - q(\bar{x})\} \\
&= \sqrt{z - q(\bar{x})} E(Q, \bar{x}), \forall z \geq q(\bar{x})
\end{aligned}$$

Lema 3.3.1. [4]

- (i) A função q tem a propriedade de arredondamento \iff para qualquer minimizador contínuo \bar{x} e $\forall x \in \mathbb{Z}^n \cap X$ temos que $\mathcal{L}_{\leq}(q(x)) \cap X \cap \text{Round}(\bar{x}) \neq \emptyset$
- (ii) A função q tem a propriedade de arredondamento forte \iff para qualquer minimizador contínuo \bar{x} e $\forall x \in \mathbb{Z}^n \cap X$ temos que $\lfloor \bar{x} \rfloor \in \mathcal{L}_{\leq}(q(x)) \cap X$

Prova. Ver [4]. ■

Portanto, a ideia é obter limites inferiores para o Problema 3.1, subestimando a função objetivo q com uma função quadrática convexa q' que possua a propriedade de arredondamento forte, para qual o mínimo inteiro pode ser obtido simplesmente pelo arredondamento de seu minimizador contínuo.

Para aplicar eficientemente os limites elipsoidais, é essencial que o mínimo inteiro da função subestimadora q' seja facilmente calculado. Nas seguintes seções apresentamos uma visão geral das diferentes classes de matrizes Q' que foram propostas na literatura para compor o subestimador q' .

Guiado pela caracterização geométrica dos conjuntos de subnível elipsoidais da função objetivo q' para as matrizes diagonais Q' , os autores em [2] inicialmente definem conjuntos *quase-arredondados*, como segue.

Definição 3.3.3. *Seja $B(x, \rho) := \{y \in \mathbb{R}^n : (y - x)^T(y - x) \leq \rho^2\}$, uma bola euclidiana de raio $\rho \in \mathbb{R}_+$ e centro $x \in \mathbb{R}^n$. Dado $\alpha \in \mathbb{R}_+$ e $x_0 \in M$, chamamos um conjunto $M \subseteq \mathbb{R}^n$ α -quase-arredondado com respeito a x_0 se existem os raios $\rho_1, \rho_2 \in \mathbb{R}_+$ tal que $B(x_0, \rho_1) \subseteq M \subseteq B(x_0, \rho_2)$ e $\rho_2 - \rho_1 \leq \alpha$.*

Agora, seja $d(x, D) := \inf\{\|x - y\| : y \in D\}$, para qualquer $x \in \mathbb{R}^n$ e $D \subset \mathbb{R}^n$, e

$$\alpha(x) := d(x, \mathbb{Z}^n \setminus \{\lfloor x \rfloor\}) - d(x, \mathbb{Z}^n). \quad (3.24)$$

Notamos que $\alpha(x)$ é a diferença entre duas distâncias euclidianas: de x ao ponto inteiro mais próximo deste, e de x ao segundo ponto inteiro mais próximo deste. O exemplo a seguir mostra de uma maneira direta a aplicação dos conceitos mencionados acima.

Exemplo 3.3.1. *Dado o problema*

$$\begin{aligned} \min \quad & f(x) = 3.7x_1^2 + 22x_1x_2 + 35x_2^2 + x_1 + 2x_2 + 8 \\ \text{s.a.} \quad & x \in \mathbb{Z}^2 \end{aligned}$$

O minimizador contínuo de f é $\bar{x} = (-\frac{13}{17}, \frac{18}{85})$ e de valor mínimo contínuo é $f(\bar{x}) = \frac{1331}{170}$.

Além disso, temos que $\lfloor \bar{x} \rfloor = (-1, 0)$ com $f(\lfloor \bar{x} \rfloor) = 10.7$, o conjunto dos pontos arredondados de \bar{x} , é dado por $\text{Round}(\bar{x}) = \{(0, 0), (0, 1), (-1, 0), (-1, 1)\}$.

- Em primeiro lugar, vamos mostrar que f tem a propriedade de arredondamento.

Ao calcular o valor de $\alpha(\bar{x})$ como se mostra na Figura 3.8 obtemos $\alpha(\bar{x}) = d_2(\bar{x}, \mathbb{Z}^n \setminus \text{Round}(\bar{x})) - d_2(\bar{x}, \mathbb{Z}^n) = 0.91$. Por outro lado, o valor da função f em cada um dos pontos do conjunto $\text{Round}(\bar{x})$ é $\{8, 45, 10.7, 25.7\}$ respectivamente, assim obtemos:

$$\begin{aligned} f(\bar{x}) & \leq z \leq \min\{f(x) : x \in \text{Round}(\bar{x})\} \\ 7.8294 & \leq z \leq 8 \\ 0 & \leq z - f(\bar{x}) \leq 0.1706. \end{aligned}$$

Logo, segundo a Observação 3.3.2 os conjuntos de subnível da função f são da forma $\mathcal{L}_{\leq}(z) = \sqrt{\hat{z}}E(Q, \bar{x})$, com $\hat{z} = z - f(\bar{x}) \in [0, 0.1706]$ serão $\alpha(\bar{x})$ -quase arredondados com $R \leq 1.23$ e $r \leq 0.32$. Assim ao tomar o valor de $\hat{z} = 0.1706$ se obtêm o maior elipsoide que é $\alpha(\bar{x})$ -quase arredondados, dado que obtemos as bolas de raios $\rho_2 = 1.23$ e $\rho_1 = 0.31$ que satisfazem a definição de quase arredondamento. Além disso, na Figura (3.8), pode-se observar que o primeiro ponto inteiro atingido pelo elipsoide escalado $\sqrt{0.1706}E(Q, \bar{x})$ é o ponto $x^* = (0, 0) \in \text{Round}(\bar{x})$. Por tanto a função f tem a propriedade de arredondamento.

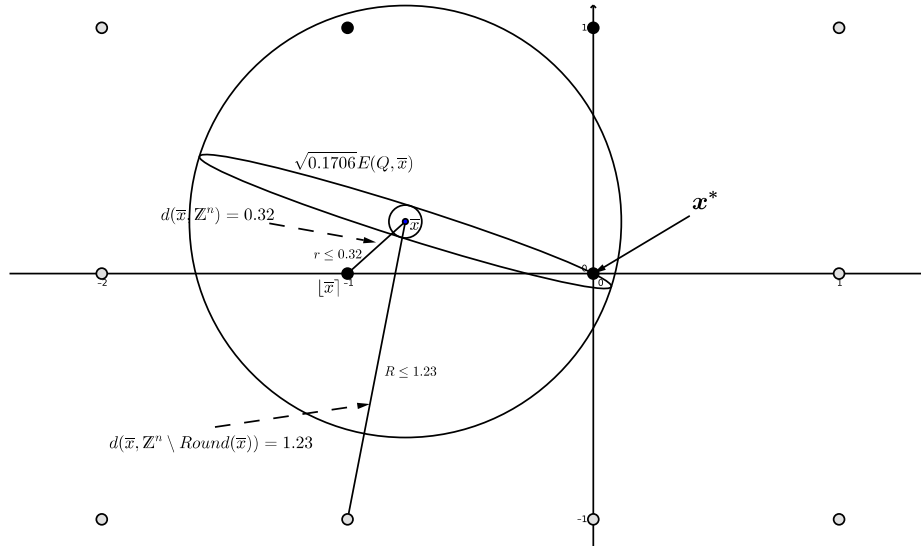


Figura 3.8: Elipsoide $\alpha(\bar{x})$ -quase arredondado com $x^* = (0, 0) \in \text{Round}(\bar{x})$.

- Agora vamos verificar se a função f tem a propriedade de arredondamento forte.

Ao calcular o valor de $\beta(\bar{x})$ como se mostra na Figura 3.9 obtemos $\beta(\bar{x}) = d_2(\bar{x}, \mathbb{Z}^n \setminus [\bar{x}]) - d_2(\bar{x}, \mathbb{Z}^n) = 0.47$ e dado que $R \leq 0.79$ não pode-se construir uma bola que contenha ao elipsoide escalado $\sqrt{0.1706}E(Q, \bar{x})$, dado que deve-se satisfazer a seguinte desigualdade $R - r \leq 0.47$. Portanto o elipsoide $\sqrt{0.1706}E(Q, \bar{x})$ não é $\beta(\bar{x})$ -quase arredondado. Na Figura 3.9, também pode-se observar que o primeiro ponto inteiro atingido ao escalar o elipsoide $E(Q, \bar{x})$ é o ponto $x^* = (0, 0) \neq [\bar{x}]$. Portanto a função f não tem a propriedade de arredondamento forte.

O valor de $\alpha(x)$ no exemplo anterior foi calculado de maneira direta por (3.24). A dificuldade é maior a medida que aumenta a dimensão do problema, por conseguinte o seguinte lema mostra uma maneira mais prática de calcular o valor de $\alpha(x)$.

Nota-se que $\alpha(x)$ é a diferença entre duas distâncias euclidianas: de x ao ponto inteiro mais próximo de x , e de x ao segundo ponto inteiro mais próximo dele.

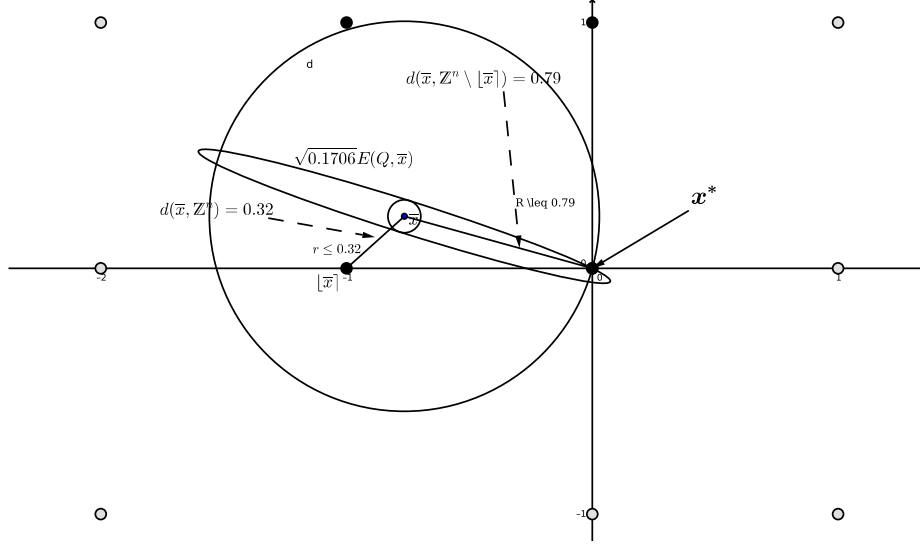


Figura 3.9: O elipsoide não é $\beta(\bar{x})$ -quase arredondado, $x^* \neq \lfloor \bar{x} \rfloor$.

Os elipsoides que tem a propriedade de quase-arredondamento satisfazem o seguinte Lema 3.3.2, que está baseada na análise dos autovalores da matriz Q , sendo λ_{\min} e λ_{\max} o menor e maior autovalor de Q respectivamente.

Lema 3.3.2. [2] *Seja $\alpha \geq 0$, $x \in \mathbb{R}^n$. Então $E(Q, x)$ é α -quase-arredondado com respeito a x se, e somente se,*

$$\frac{1}{\sqrt{\lambda_{\min}}} - \frac{1}{\sqrt{\lambda_{\max}}} \leq \alpha.$$

Prova. *Dado que, o circuncentro e o incentro (centro da esfera que contém o elipsoide e o centro da esfera contida no elipsoide, respectivamente) de um elipsoide coincidem com o centro do elipsoide e o circunraio e inraio são dados pelo raios equatoriais como se mostra na Figura 3.10, temos*

$$B(x, \frac{1}{\sqrt{\lambda_{\max}}}) \subseteq E(Q, x) \subseteq B(x, \frac{1}{\sqrt{\lambda_{\min}}})$$

Reciprocamente, pela definição, temos $B(x, r) \not\subseteq E(Q, x)$ para todo $r > \frac{1}{\sqrt{\lambda_{\max}}}$ e $E(Q, x) \not\subseteq B(x, R)$ para todo $R < \frac{1}{\sqrt{\lambda_{\min}}}$ ■

Teorema 3.3.1. *Seja $q' : \mathbb{R}^n \rightarrow \mathbb{R}$. Assuma que os conjuntos de subnível $\mathcal{L}_{q'}(z)$, definidos como em (2.10), são $\alpha(\bar{x})$ -quase-arredondado, com respeito a \bar{x} , para qualquer minimizador contínuo \bar{x} de q' , e para todo $z \leq q'(\lfloor \bar{x} \rfloor)$. Então q' tem a propriedade de arredondamento forte.*

Prova. *Sejam \bar{x} qualquer minimizador contínuo de q' e $x \in \mathbb{Z}^n$. Devemos mostrar que $q'(\lfloor \bar{x} \rfloor) \leq q'(x)$. Assumimos que, $q'(\lfloor \bar{x} \rfloor) > q'(x)$, então sabemos que $x \neq \lfloor \bar{x} \rfloor$*

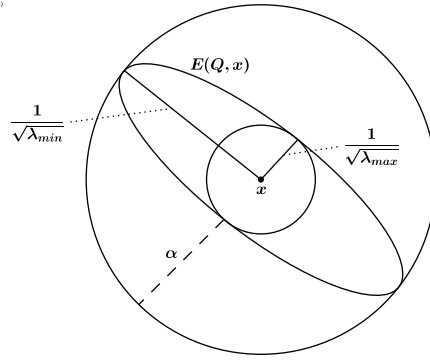


Figura 3.10: Elipsoide α -quase arredondado, segundo a propriedade de seus autovalores.

e existem $r, R \in \mathbb{R}_+$ tal que $B(\bar{x}, r) \subseteq \mathcal{L}_{q'}(z) \subseteq B(\bar{x}, R)$ e $R - r \leq \alpha(\bar{x})$. Uma vez que, $\alpha(\bar{x}) = d_2(\bar{x}, \mathbb{Z}^n \setminus \{\lfloor \bar{x} \rfloor\}) - \|\bar{x} - \lfloor \bar{x} \rfloor\|_2$, temos

$$\|\bar{x} - \lfloor \bar{x} \rfloor\|_2 = d_2(\bar{x}, \mathbb{Z}^n \setminus \{\lfloor \bar{x} \rfloor\}) - \alpha(\bar{x}) \quad (3.25)$$

$$\leq \|\bar{x} - x\|_2 - \alpha(\bar{x}) \quad (3.26)$$

$$\leq R - \alpha(\bar{x}) \leq r \quad (3.27)$$

Isto significa que $\lfloor \bar{x} \rfloor \in B(\bar{x}, r) \subseteq \mathcal{L}_{q'}(z) \subseteq B(\bar{x}, R)$ e portanto $q'(\lfloor \bar{x} \rfloor) \leq q'(x)$, o que é uma contradição. ■

Corolário 3.3.1. Sejam $\lambda'_{\max} := \lambda'_1 \geq \lambda'_2 \geq \dots \geq \lambda'_n =: \lambda'_{\min}$ os autovalores de Q' . Se

$$\lambda'_{\max} \leq \frac{\lambda'_{\min}}{\left(1 - \frac{\alpha(\bar{x})}{\sqrt{q'(\lfloor \bar{x} \rfloor) - q'(\bar{x})}} \sqrt{\lambda'_{\min}}\right)^2}, \quad (3.28)$$

então a função quadrática q' tem a propriedade de arredondamento forte.

Prova. Dada a desigualdade (3.28), por simplicidade fazemos $\beta := \frac{\alpha(\bar{x})}{\sqrt{q'(\lfloor \bar{x} \rfloor) - q'(\bar{x})}}$, assim obtemos

$$\frac{1}{\sqrt{\lambda'_{\max}}} \geq \frac{1 - \beta \sqrt{\lambda'_{\min}}}{\sqrt{\lambda'_{\min}}} \quad (3.29)$$

Logo,

$$\frac{1}{\sqrt{\lambda'_{\min}}} - \frac{1}{\sqrt{\lambda'_{\max}}} \leq \frac{1}{\sqrt{\lambda'_{\min}}} - \frac{1 - \beta \sqrt{\lambda'_{\min}}}{\sqrt{\lambda'_{\min}}} = \beta \quad (3.30)$$

Portanto, o elipsoide $E(Q')$ é β quase arredondado, além disso segundo o Teorema (3.3.1), temos que a função q' tem a propriedade de arredondamento forte. ■

O cálculo da matriz Q' satisfazendo a desigualdade (3.28), que leva ao melhor limite inferior para o problema quadrático inteiro, não pode ser feito de forma tão

eficiente como para o caso onde Q' é diagonal. Por tal motivo, os autores em [2] propõem uma heurística que constrói uma matriz $Q' = \sum_{i=1}^n \lambda'_i \mu_i \mu_i^T$, com seus autovetores iguais aos autovetores da matriz Q , e cada autovalor λ'_i é definido como

$$\lambda'_i = \min \left\{ \lambda_i, \frac{\lambda_{\min}}{\left(1 - \frac{\alpha(\bar{x})}{\sqrt{q(\lfloor \bar{x} \rfloor) - q(\bar{x})}} \sqrt{\lambda_{\min}}\right)^2} \right\}. \quad (3.31)$$

Assim o elipsoide $E(Q', \bar{x})$ é α -quase arredondado segundo Corolário 3.3.1, com $\alpha := \frac{\alpha(\bar{x})}{\sqrt{q(\lfloor \bar{x} \rfloor) - q(\bar{x})}}$.

Por outro lado, como $q'_Q(\lfloor \bar{x} \rfloor) \leq q(\lfloor \bar{x} \rfloor)$ e $q'_Q(\bar{x}) = q(\bar{x})$, temos que $E(Q', \bar{x})$ é também α' -quase arredondado, para $\alpha' = \frac{\alpha(\bar{x})}{\sqrt{q'_Q(\lfloor \bar{x} \rfloor) - q'_Q(\bar{x})}}$, sendo $\alpha' \geq \alpha$. Concluindo, que os conjuntos de subnível da função subestimadora $q'_Q(\bar{x}) := x^T Q' x + c^T x$ são quase arredondados para todo nível $z \leq q'_Q(\lfloor \bar{x} \rfloor)$ e portanto q'_Q tem a propriedade de arredondamento forte segundo o Teorema 3.3.1.

De fato, a heurística proposta, gera um limite inferior. Isto é, dado que $\lfloor \bar{x} \rfloor$ é uma solução ótima para $\min\{q'_Q(x) : x \in \mathbb{Z}^n\}$, então

$$\min\{q(x) : x \in \mathbb{Z}^n\} \geq q(\bar{x}) + (\bar{x} - \lfloor \bar{x} \rfloor)^T Q' (\bar{x} - \lfloor \bar{x} \rfloor).$$

Os principais passos da heurística, a qual vamos nos referir como *BHS*, são descritas no Algoritmo (3).

Input: $\bar{x}, q(\cdot), U, \Sigma := \text{Diag}(\lambda_1, \dots, \lambda_n)$, onde $Q := U \Sigma U^T$
Output: Q'

- 1 $\lambda_{\min} := \lambda_n$;
- 2 $\lfloor \bar{x} \rfloor := \text{round}(\bar{x})$;
- 3 $\alpha(\bar{x}) := d(\bar{x}, \mathbb{Z}^n \setminus \{\lfloor \bar{x} \rfloor\}) - d(\bar{x}, \mathbb{Z}^n)$;
- 4 **for** $i = 1, \dots, n$ **do**
- 5 $\lambda'_i := \min \left\{ \lambda_i, \frac{\lambda_{\min}}{\left(1 - \frac{\alpha(\bar{x})}{\sqrt{q(\lfloor \bar{x} \rfloor) - q(\bar{x})}} \sqrt{\lambda_{\min}}\right)^2} \right\}$;
- 6 $\Sigma' := \text{Diag}(\lambda'_1, \dots, \lambda'_n)$;
- 7 $Q' := U \Sigma' U^T$;

Algorithm 3: Heurística do elipsoide quase arredondado *BHS* ([2])

Em [2], os autores mostram através de resultados numéricos que esta heurística é o método mais promissor apresentado no artigo a ser utilizado dentro de uma estrutura de *branch-and-bound*, principalmente devido à computação muito rápida dos limites. Entretanto, eles não comparam seus resultados com o método apresentado em [2], que usa o algoritmo de redução de bases para calcular os vetores v_i em (3.20).

Por outro lado, os autovalores λ'_i definidos segundo a equação (3.31), satisfazem a desigualdade (3.28) do Corolário 3.3.1, levando a uma subestimador q' com a propriedade de arredondamento forte. Resultados em [2] mostram que esta heurística leva ao cálculo de limites mais rápido do que os limites dados pela solução do problema semidefinido (3.1.1). Resultados promissores são apresentados com a sua utilização em uma metodologia *branch-and-bound*.

Uma intuição sobre os limites elipsoidais é mais uma vez dada pelos conjuntos de subnível de q e q' , com ambos elipsoides com centro em \bar{x} . O elipsoide correspondente a q' para o nível z fixado como $q'(\lfloor \bar{x} \rfloor)$ deve conter o elipsoide correspondente a q para o mesmo nível (como $Q - Q' \succeq 0$), deve conter $\lfloor \bar{x} \rfloor$ na sua fronteira, e não deve ter ponto inteiro no seu interior (quando q' tem a propriedade de arredondamento forte).

Para gerar bons limites, nós gostaríamos de ter os dois elipsoides tão semelhantes quanto possível. Portanto, definir os autovetores próprios de Q' iguais aos autovetores de Q parece uma ideia interessante. No entanto, o limite superior sobre os autovalores de Q' , dado por

$$\frac{\lambda_{\min}}{\left(1 - \frac{\alpha(\bar{x})}{\sqrt{q(\lfloor \bar{x} \rfloor) - q(\bar{x})}} \sqrt{\lambda_{\min}}\right)^2}, \quad (3.32)$$

pode mudar muito a forma do elipsoide correspondente, quando comparado com Q .

Por outro lado, quando os autovetores de Q' e Q coincidem, os autovalores de Q' que levam ao melhor limite elipsoidal possível e que satisfazem (3.28) são a solução de

$$\left\{ \begin{array}{l} \max \sum_{i=1}^n \lambda'_i y_i^2 \\ \text{s.a. } \lambda - \lambda' \geq 0. \\ \\ \lambda'_{\max} \leq \frac{\lambda'_{\min}}{\left(1 - \frac{\alpha(\bar{x})}{\sqrt{q'(\lfloor \bar{x} \rfloor) - q'(\bar{x})}} \sqrt{\lambda'_{\min}}\right)^2} \\ \\ \lambda' \geq 0 \end{array} \right. \quad (3.33)$$

onde $y := U(\lfloor \bar{x} \rfloor - \bar{x})$, sendo U a matriz dos autovetores de Q , e $\lambda_i, \forall i = 1, \dots, n$, os autovalores de Q . Como (3.28) torna o problema (3.33) difícil, a heurística substitui este limite por (3.31). Para o problema (3.33), vemos claramente que, o limite inferior de (PQI) dado pela heurística, enfraquece quando a expressão em (3.32) aproxima-se de λ_{\min} , especialmente se $\lambda_{n-1} \gg \lambda_{\min}$, nesse caso, cada valor próprio de Q' , exceto λ_{\min} , torna-se muito menor do que o autovalor correspondente de Q . Uma situação extrema ocorre quando \bar{x} é equidistante dos dois inteiros mais próximos a ele, neste caso $\alpha(\bar{x})$ torna-se igual a zero. Interpretando geometricamente

esta situação, temos o elipsoide correspondente a Q' transformado em uma bola, e, em geral, muito diferente do elipsoide correspondente Q , levando a limites inferiores fracos.

No exemplo a seguir, mostramos a obtenção de um limite inferior segundo a heurística do elipsoide quase arredondado denotada por BHS

Exemplo 3.3.2. *Dado o problema*

$$\begin{aligned} \min \quad & q(x) = 3.7x_1^2 + 22x_1x_2 + 35x_2^2 + x_1 + 2x_2 + 8 \\ \text{s.a.} \quad & x \in \mathbb{Z}^2 \end{aligned}$$

Como foi visto no exemplo (3.3.1), f não tem a propriedade de arredondamento forte, isto é, o elipsoide $E(Q, \bar{x})$ não é α -quase arredondado. Por outro lado, temos que a solução contínua é $\bar{x} = (-\frac{13}{17}, \frac{18}{85})$ e seu valor mínimo contínuo é $q(\bar{x}) = \frac{1331}{170}$. Além disso, temos que $\lfloor \bar{x} \rfloor = (-1, 0)$.

Calculamos o valor de $\alpha(\bar{x})$ utilizando a Definição 3.3.3, obtendo $\alpha(\bar{x}) = 0.4769$. Logo calculamos os novos autovalores segundo a proposta de Buchheim em [2], obtendo $\lambda'_1 = \frac{909}{4115}$ e $\lambda'_2 = \frac{967}{2635}$.

$$Q = \begin{pmatrix} 3.7 & 11 \\ 11 & 35 \end{pmatrix} = \begin{pmatrix} -\frac{553}{580} & \frac{329}{1091} \\ \frac{329}{1091} & \frac{553}{580} \end{pmatrix} \begin{pmatrix} \frac{909}{4115} & 0 \\ 0 & \frac{11967}{311} \end{pmatrix} \begin{pmatrix} -\frac{553}{580} & \frac{329}{1091} \\ \frac{329}{1091} & \frac{553}{580} \end{pmatrix}^T$$

$$Q'_{BHS} = \begin{pmatrix} \frac{459}{1960} & \frac{167}{3976} \\ \frac{167}{3976} & \frac{1731}{4894} \end{pmatrix} = \begin{pmatrix} -\frac{553}{580} & \frac{329}{1091} \\ \frac{329}{1091} & \frac{553}{580} \end{pmatrix} \begin{pmatrix} \frac{909}{4115} & 0 \\ 0 & \frac{967}{2635} \end{pmatrix} \begin{pmatrix} -\frac{553}{580} & \frac{329}{1091} \\ \frac{329}{1091} & \frac{553}{580} \end{pmatrix}^T$$

A Figura 3.11 mostra o elipsoide $E(Q'_{BHS}, \bar{x})$ que é $\alpha(\bar{x})$ -quase arredondado, de modo que:

$$\min_{x \in \mathbb{Z}^n} q_{Q'_{BHS}}(x) = q(\bar{x}) + (\lfloor \bar{x} \rfloor - \bar{x})^T Q'_{BHS} (\lfloor \bar{x} \rfloor - \bar{x}) = \frac{1331}{170} + 0.033 = 7.8624.$$

Finalmente, obtemos um limite inferior para o Problema 3.1

$$\min_{x \in \mathbb{Z}^n} q(x) \geq 7.8624$$

3.4 Algoritmos com matrizes de posto um

Nesta seção, apresentamos uma primeira e simples contribuição nossa para o cálculo dos limites elipsoidais. Ela consiste de uma nova construção da matriz $Q' = \delta^* vv^T$, onde $\delta^* \in \mathbb{R}_+$ e $v \in \mathbb{Z}^n \setminus \{0\}$, para compor a função subestimadora q' .

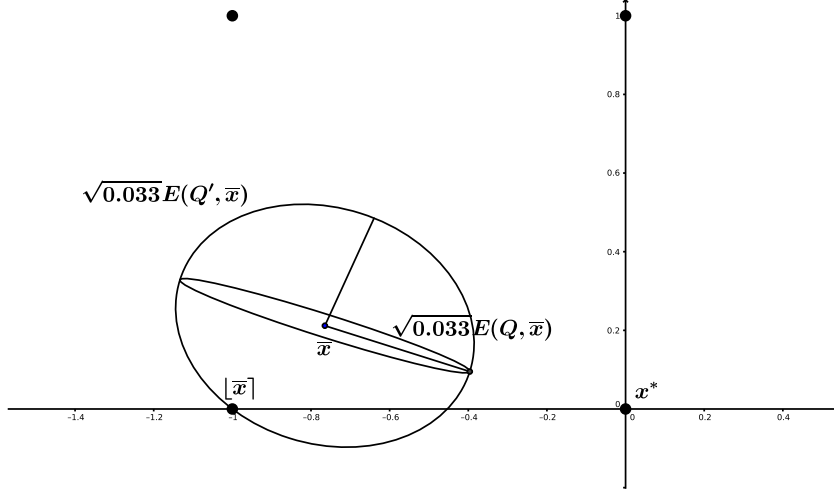


Figura 3.11: Elipsoide $E(Q'_{BHS}, \bar{x})$ gerado tal que $E(Q, \bar{x}) \subseteq E(Q'_{BHS}, \bar{x})$.

Nosso objetivo é ter limites inferiores para o problema (PQI) que podem ser calculados de uma maneira simples e rápida, com a qualidade dos limites obtidos por outras metodologias apresentadas na literatura.

3.4.1 Proposta de uma nova heurística

Para a construção da matriz Q' , tal que $E(Q) \subseteq E(Q')$, consideramos os resultados da Proposição 3.2.1. Conforme enfatizado em [1], o valor de $\max_{x \in E(Q)} v^T x$ é a largura do elipsoide $E(Q)$ ao longo do vetor v . De modo a obter limites fortes, este vetor v deve ser selecionado de tal modo que a largura do elipsoide $E(Q)$ na sua direção seja tão pequena quanto possível.

Considerando-se, então, que a direção ao longo da qual o elipsoide $E(Q)$ é mais longo é a direção dada por u_n (o autovetor de Q associado ao seu menor autovalor), por tal motivo procuramos por um vetor inteiro v , com uma projeção nula ou, pelo menos, com uma pequena projeção na direção dada por u_n , tal como se mostra na Figura 3.12

Finalmente, aplicando o resultado na Proposição 3.2.1 e considerando a decomposição espectral de Q dadas por $Q = U \Sigma U^T$, o parâmetro δ^* que maximiza o limite inferior assegurando que q' subestime q , é calculado por

$$\delta^* := \frac{1}{\|v^T (\Sigma^{\frac{1}{2}} U^T)^{-1}\|_2^2} = \frac{1}{\|v^T (U \Sigma^{-\frac{1}{2}})\|_2^2} = \frac{1}{\sum_{j=1}^n \frac{1}{\lambda_j^2} (v^T u_j)^2}. \quad (3.34)$$

O Algoritmo 1 mostra os passos de nosso procedimento, utilizado no cálculo da matriz de posto um Q' .

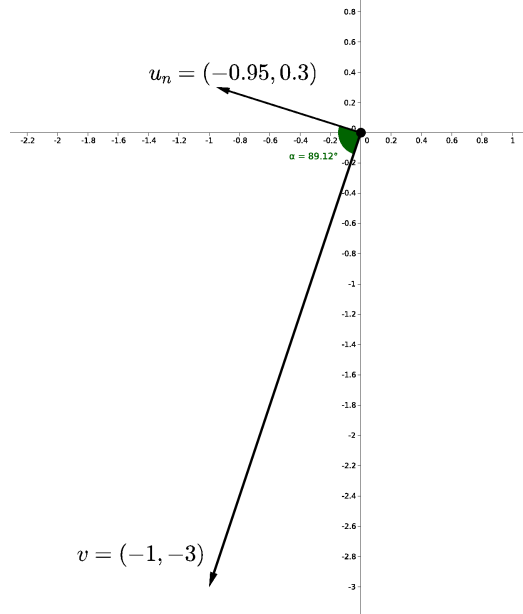


Figura 3.12: Vetor v quase ortogonal ao autovetor u_n .

Data: $Q \succ 0$,
 $\{\lambda_1, \dots, \lambda_n\}$ autovalores
 $\{u_1, \dots, u_n\}$ autovetores

- 1 $(\alpha, \beta) := \arg \min_{\{(i,j): i,j \in \{1, \dots, n\}, i \neq j\}} \left\{ \left| \frac{u_n(i)}{u_n(j)} \right| - \frac{u_n(i)}{u_n(j)} \right\}$;
- 2 $v(\alpha) := -1$;
- 3 $v(\beta) := \left\lfloor \frac{u_n(\alpha)}{u_n(\beta)} \right\rfloor$;
- 4 $v(k) := 0, \forall k \in \{1, \dots, n\}, k \neq \alpha, k \neq \beta$;
- 5 $\delta^* := \frac{1}{\sum_{j=1}^n \frac{1}{\lambda_j^2} (v^T u_j)^2}$;
- 6 **return** $Q' := \delta^* v v^T$;

Algorithm 4: Heurística que gera a matriz Q' de posto um, denotada por Hv

Nota-se que, se $\frac{u_n(\alpha)}{u_n(\beta)} \in \mathbb{Z}$, então

$$v^T u_n = -u_n(\alpha) + \left\lfloor \frac{u_n(\alpha)}{u_n(\beta)} \right\rfloor u_n(\beta) = 0.$$

Da igualdade acima, podemos dizer que o vetor v é ortogonal a u_n . Caso contrário, nós selecionamos os índices α e β com o objetivo de ter o ângulo entre v e u_n tão próximo quanto possível a 90° .

Como $v \in \mathbb{Z}^n$, um limite inferior de (PQI) , é dado por $q(\bar{x}) + \delta^* (\lfloor v^T \bar{x} \rfloor - v^T \bar{x})^2$ ver (3.22). Finalmente, para nosso vetor selecionado v , é muito simples verificar que tal limite inferior é de fato o mínimo inteiro de q' .

A seguir, apresentamos um exemplo para a obtenção de um limite inferior, utilizando a nossa heurística, comparando com o limite inferior obtido pela heurística do elipsoide quase arredondado apresentado no Exemplo 3.3.2

Exemplo 3.4.1. *Dado o problema*

$$\begin{aligned} \min \quad & q(x) = 3.7x_1^2 + 22x_1x_2 + 35x_2^2 + x_1 + 2x_2 + 8 \\ \text{s.a.} \quad & x \in \mathbb{Z}^2. \end{aligned}$$

Como foi visto no exemplo 3.3.2, o limite inferior obtido pela heurística BHC proposta por Buchheim em [2], foi

$$\min_{x \in \mathbb{Z}^n} q_{Q'_{BHS}}(x) = f(\bar{x}) + ([\bar{x}] - \bar{x})^T Q'_{BHS} ([\bar{x}] - \bar{x}) = \frac{1331}{170} + 0.033 = 7.8624.$$

Por outro lado, utilizando a nossa proposta, obtemos o vetor $v = (0, -1)^T$, tal que $\angle(v, u_n) = 89.12^\circ$, com $\delta^* = 3.6957$ e o elipsoide $E(Q', \bar{x})$ como se mostra na Figura 3.13, obtendo o seguinte limite inferior

$$\min_{x \in \mathbb{Z}^n} q_{Q'}(x) = q(\bar{x}) + \delta^* ([v^T \bar{x}] - v^T \bar{x})^2 = \frac{1331}{170} + 0.0619 = 7.8913.$$

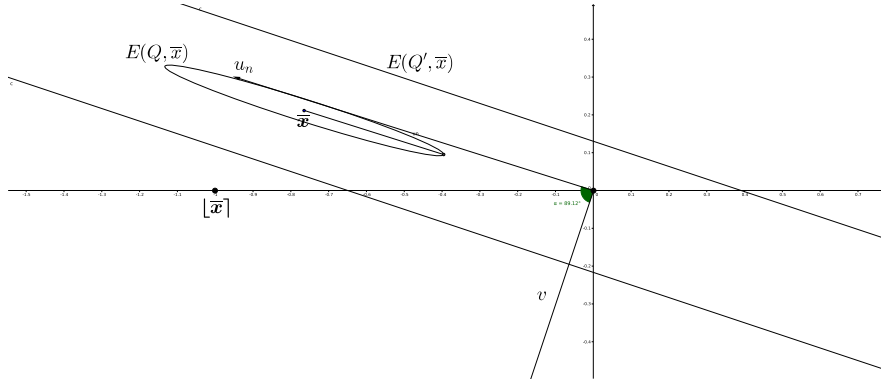


Figura 3.13: Elipsoide $E(Q', \bar{x})$ gerado com o vetor v .

3.4.2 Teste computacional

Com a finalidade de testar a proposta em nossos experimentos, utilizamos um grupo de testes, considerando 10 instâncias para cada número de variáveis $n \in \{10, 20, \dots, 50\}$. As instâncias contem matrizes de posto completo L , tal que $Q = LL^T$, e as entradas na matriz L e no vetor c são inteiros escolhidos aleatoriamente e uniformemente no intervalo $[-5, 5]$.

Implementamos os diferentes métodos que calculam os limites inferiores para o problema estritamente convexo (PQIEC) discutidas nas seções previas, e compara-

mos eles numericamente. Dois métodos são aplicados para construir a matriz Q' , o método proposto em [1] e discutido na Seção 3.3 (denotado por BHS), e o algoritmo proposto na Seção 3.4 (denotado por Hv).

Assim, nesta subseção fazemos uma comparação entre os limites inferiores obtidos pela heurística BHS e a nossa proposta Hv .

Na Tabela 3.1 mostramos a média do tempo de processamento da CPU em segundos, utilizado em cada uma das heurísticas BHS e Hv .

Tabela 3.1: Tempo médio de CPU (em segundos).

n	Hv	BHS
10	0.003	0.001
20	0.002	0.000
30	0.001	0.004
40	0.004	0.004
50	0.003	0.003

Por outro lado, para medir a eficiência na obtenção dos limites inferiores, utilizamos as seguintes expressões no cálculo dos $gaps$ obtidos pelas heurísticas,

$$gap_{Hv} = \frac{LB_{Hv} - f(\bar{x})}{|f(\bar{x})|} \times 100\%; \quad (3.35)$$

$$gap_{BHS} = \frac{LB_{BHS} - f(\bar{x})}{|f(\bar{x})|} \times 100\% \quad (3.36)$$

Na Tabela 3.2 podemos ver que apesar de ter um bom desempenho quanto ao tempo de processamento, nossa heurística na média gera limites inferiores menores aos da heurística dos elipsoides quase arredondados.

Tabela 3.2: Média dos $gaps$ (%).

n	Hv	BHS
10	2.3541	1.3254
20	0.0983	1.0002
30	0.0042	0.1968
40	0.0057	0.1982
50	0.0179	0.4987

Na próxima seção veremos outras alternativas mais sofisticadas, mas com base nas mesmas ideias que Hv , para melhorar os resultados obtidos por esta heurística.

3.5 Adicionando matrizes de posto um

Em nossos experimentos numéricos com matrizes Q' calculadas segundo a equação (3.20), notamos que o parâmetro β^* era muitas vezes muito pequeno, levando a limites fracos. Isso equivale a dizer que tivemos que ampliar muito, ou aumentar a escala do elipsoide $\sqrt{z^* - q'(\bar{x})}E(\sum_{i=1}^n \delta_i^* v_i v_i^T, \bar{x})$ por um fator muito grande, de tal modo que ele contivesse o elipsoide $\sqrt{z^* - q(\bar{x})}E(Q, \bar{x})$, como se mostra na Figura 3.14

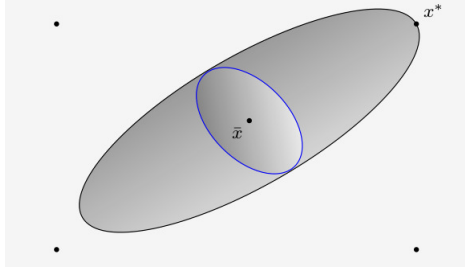


Figura 3.14: $\sqrt{z^* - q(\bar{x})}E(Q, \bar{x}) \subseteq \sqrt{z^* - q'(\bar{x})}E(Q', \bar{x})$.

Na tentativa de melhorar os limites, propomos então nesta Seção três alternativas para gerar as matrizes Q' , adicionando matrizes de posto um. As matrizes geradas são também da forma $Q' = \sum_{i=1}^s \delta_i v_i v_i^T$, onde $\delta_i \in \mathbb{R}_+$, $v_i \in \mathbb{Z}^n$, para $i = 1, \dots, s$, com $s \leq n$, e $Q - Q' \succeq 0$, mas com o valor de s possivelmente menor que n . Como em [1] os limites inferiores para $(PQISC)$ são dados por

$$\min\{q(x) : x \in \mathbb{Z}^n\} \geq q(\bar{x}) + \sum_{i=1}^s \delta_i \left(\lfloor v_i^T \bar{x} \rfloor - v_i^T \bar{x} \right)^2 \quad (3.37)$$

O que difere entre os três algoritmos propostos é como selecionar os vetores v_i , e, uma vez selecionados, como calcular os valores de δ_i , objetivando gerar os maiores limites inferiores possíveis.

Em nossos primeiros dois algoritmos, descritos no Algoritmo 5 e no Algoritmo 6, selecionamos os vetores v_i como sugerido em [1], usando um algoritmo de redução de base para calcular uma base reduzida para o reticulado gerado pelas colunas da matriz L^{-1} , onde $Q = LL^T$. Para esta escolha de vetores v_i , propomos dois métodos para calcular os valores de δ_i e adicionando as matrizes de posto um.

O Algoritmo 5 é um método guloso que consiste em uma pequena modificação na ideia apresentada em [1], onde matrizes de posto um são adicionadas, mas evitando o fator de escalonamento β^* . As matrizes de posto um são iterativamente adicionadas a Q' , e cada uma é escolhida para aumentar ao máximo o limite inferior, até que nenhuma outra possa ser adicionada sem violar a desigualdade $Q - Q' \succeq 0$. Vamos nos referir a este algoritmo como (Gul)

O Algoritmo 6 calcula o melhor limite possível para os vetores selecionados v_i resolvendo o problema de programação semidefinida seguinte

$$\left\{ \begin{array}{l} \max \quad \sum_{i=1}^n \delta_i (\lfloor v_i^T \bar{x} \rfloor - v_i^T \bar{x}_i)^2 \\ \text{s.a.} \quad Q - \sum_{i=1}^r \delta_i v_i v_i^T \succcurlyeq 0. \\ \delta_i \geq 0, \end{array} \right. \quad (3.38)$$

mas também é a estratégia mais cara que propomos para gerar a matriz Q' . Vemos isso como uma forma de medir o quão longe estamos desses limites otimizados ao usar o algoritmo (*Gul*). Nós nos referiremos a este segundo algoritmo como (*Oti*). Finalmente, o último método descrito no Algoritmo 7 é muito simples e pode ser executado mais rapidamente do que os outros métodos, mas só é aplicável se a matriz L , tal que $Q = LL^T$ é racional. Neste caso, escolhemos o vetor inteiro v_i paralelo à i -ésima coluna de L , isto é, $v_i = \alpha_i L_i \in \mathbb{Z}^n$, para o menor número real positivo α_i . Para os vetores selecionados v_i neste algoritmo, não garantimos que um inteiro y satisfazendo (3.18) existe, no entanto, ressaltamos que a desigualdade (3.17) ainda é verdadeira, o que assegura que um limite inferior para o problema estritamente convexo (*PQIEC*) não é menor que o limite contínuo $q(\bar{x})$.

Uma vantagem desta estratégia é o cálculo trivial do valor máximo de δ_i , tal que $Q - \sum_{i=1}^n \delta_i v_i v_i^T \succcurlyeq 0$, que é dado por $1/\alpha_i^2$. No restante da tese, faremos referência a este algoritmo como (*Rac*).

<p>Input: $L \in \mathbb{R}^{n \times n}$, onde $\text{posto}(L) = n$, e $Q := LL^T$ Output: Q'</p> <ol style="list-style-type: none"> 1 $Q' := \emptyset$; 2 for $i = 1, \dots, n$ do 3 Seleciona v_i, tal que $w_i := L^{-1}v_i$ é o i-ésimo vetor na base reduzida do reticulado gerado pelas colunas de L^{-1}; 4 $\delta_i := \frac{1}{\ L^{-1}v_i\ _2^2}$; 5 Seja $\{j(1), \dots, j(n)\}$ tal que <div style="text-align: center; margin: 10px 0;"> $\delta_{j(1)} \left(\lfloor v_{j(1)}^T \bar{x} \rfloor - v_{j(1)}^T \bar{x} \right)^2 \geq \dots \geq \delta_{j(n)} \left(\lfloor v_{j(n)}^T \bar{x} \rfloor - v_{j(n)}^T \bar{x} \right)^2$ </div> 6 for $i = 1, \dots, n$ do 7 if $Q' + \delta_{j(i)} v_{j(i)} v_{j(i)}^T \succeq 0$ then 8 $Q' := Q' + \delta_{j(i)} v_{j(i)} v_{j(i)}^T$; 9 else 10 Parar;

Algorithm 5: Guloso (*Gul*)

Input: $\bar{x}, L \in \mathbb{R}^{n \times n}$, onde $\text{posto}(L) = n$, e $Q := LL^T$

Output: Q'

- 1 **for** $i = 1, \dots, n$ **do**
- 2 Selecione v_i , tal que $w_i := L^{-1}v_i$ é o i -ésimo vetor na base reduzida do reticulado gerado pelas colunas de L^{-1} ;
- 3 $\delta := \operatorname{argmax} \left\{ \sum_{i=1}^n \delta_i (|v_i^T \bar{x}| - v_i^T \bar{x})^2 : Q - \sum_{i=1}^n \delta_i v_i v_i^T \succeq 0, \delta \geq 0 \right\}$;
- 4 $Q' := Q + \sum_{i=1}^n \delta_i v_i v_i^T$;

Algorithm 6: Otimizado (*Oti*)

Input: $L \in \mathbb{Q}^{n \times n}$, onde $\text{posto}(L) = n$, e $Q := LL^T$

Output: Q'

- 1 Seja $L_{.i}$ a i -ésima coluna de L , para $i = 1 \dots, n$;
- 2 $v_i := \alpha_i L_{.i} \in \mathbb{Z}^n$;
- 3 $Q' := \sum_{i=1}^n \frac{1}{\alpha_i^2} v_i v_i^T$;

Algorithm 7: Para uma matriz racional L (*Rac*)

3.6 Testes Computacionais

3.6.1 Geração de instâncias

Em nossos experimentos, utilizamos um grupo de testes, considerando 10 instâncias para cada número de variáveis $n \in \{10, 20, \dots, 50\}$. As instâncias contêm matrizes de posto completo L , tal que $Q = LL^T$, e as entradas na matriz L e no vetor c são inteiros escolhidos aleatoriamente e uniformemente no intervalo $[-5, 5]$.

Implementamos os diferentes métodos que calculam os limites inferiores para o problema estritamente convexo (*PQIEC*) discutidas nas seções previas, e comparamos eles numericamente. Cinco métodos são aplicados para construir a matriz Q' , o método proposto em [1] e discutido na Seção 3.2 (denotado por *BCL*), a heurística proposta em [2] e discutida na Seção 3.3 (denotado por *BHS*), e os três algoritmos propostos na Seção 3.5 (denotadas por *Gul*, *Oti* e *Rac*).

Nós implementamos os diferentes métodos em C^{++} , e realizamos todos os experimentos computacionais em um computador com as seguintes características, Intel i3 com processador de 1.7 GHz e 4 MB cachê, 4 GB de memória RAM, com o sistema operacional Ubuntu Linux 14.04 LTS. Resolvemos os problemas de programação semidefinida com a biblioteca computacional *CVX* [18] para otimização convexa, e realizamos os cálculos de autovalores e autovetores das matrizes com as bibliotecas *LAPACK* [19] e *BLAS* [20].

3.6.2 Resultados numéricos

Nesta subseção, apresentamos os resultados computacionais comparando os limites inferiores, obtidos pelas diferentes metodologias descritas nas seções anteriores.

Na Tabela 3.3 apresenta-se o tempo médio de processamento da *CPU* em segundos, utilizado por cada um dos métodos. Por outro lado, Na Tabela 3.4 mostra-se a média dos *gaps*, segundo a Equação (3.39), onde *LB* é o limite inferior calculado por cada um dos métodos.

<i>n</i>	<i>BHS</i>	<i>BCL</i>	<i>Gul</i>	<i>Oti</i>	<i>Rac</i>
10	0.0000	0.0047	0.0094	0.1732	0.0000
20	0.0000	0.0234	0.0250	0.2434	0.0016
30	0.0000	0.0484	0.0515	0.3572	0.0031
40	0.0062	0.3682	0.4477	2.3400	0.0312
50	0.0000	0.5039	0.6380	3.1840	0.0499

Tabela 3.3: Tempo médio de processamento da *CPU* em segundos.

<i>n</i>	<i>BHS</i>	<i>BCL</i>	<i>Gul</i>	<i>Oti</i>	<i>Rac</i>
10	59.5103	44.5370	54.1777	72.8959	68.4977
20	37.1043	41.3834	42.4413	69.9523	53.8916
30	19.9918	30.3653	44.7137	69.9552	45.4651
40	10.4554	25.8381	35.2372	65.0045	52.4135
50	7.6414	16.1155	30.0671	55.8879	40.2828

Tabela 3.4: Média dos *gaps* para (*PQISC*) nas instâncias aleatórias.

Para uma melhor interpretação dos resultados, na Figura 3.15 mostramos a média relativa de melhora do limite inferior obtido em cada método, quando comparamos o limite dado pelo mínimo contínuo. Mas precisamente, apresentamos

$$gap_{LB} := \frac{LB - q(\bar{x})}{|q(\bar{x})|} * 100\%, \quad (3.39)$$

onde *LB* é o limite inferior resultante do método escolhido. Na Figura 3.16, também mostramos a média do tempo de processamento da *CPU* em segundos no cálculo dos limites inferiores.

Podemos ver nas Figuras 3.15 e 3.16 que o tempo processamento da *CPU* para executar o método heurístico *BHS* é muito pequeno, mas os limites calculados por esta estratégia para as nossas instancias aleatórias são muito mais fracos do que os outros para $n \geq 20$.

Também notamos que os resultados obtidos pelo algoritmo *BCL* proposto em [1] em comparação com o algoritmo *BHS*, apresenta um pequeno incremento no tempo de processamento da *CPU*, no entanto apresenta uma considerável melhoria nos limites. Isto nós leva a dizer que a utilização de vetores curtos do reticulado gerado pelas colunas da matriz L^{-1} é uma ideia interessante, dado que levou a bons resultados quando cada matriz de posto um $Q' = \delta_i^* v_i v_i^T$ é considerada.

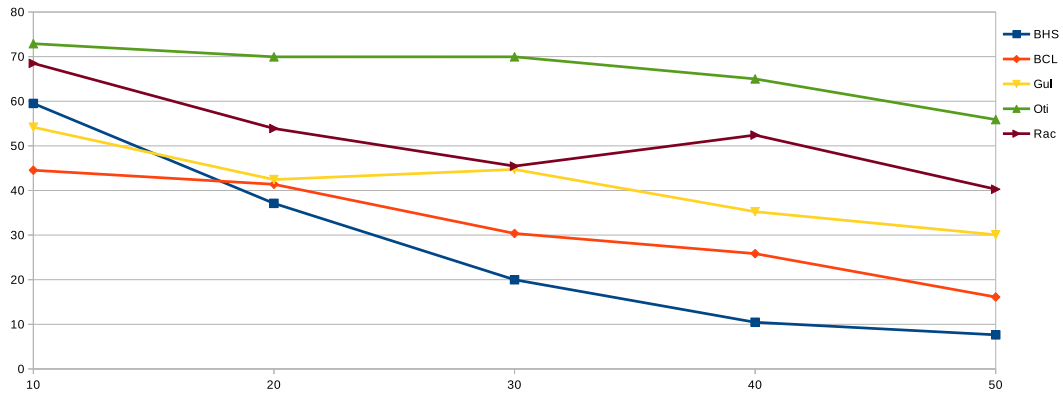


Figura 3.15: Melhora nos limites inferiores comparados com o mínimo contínuo.

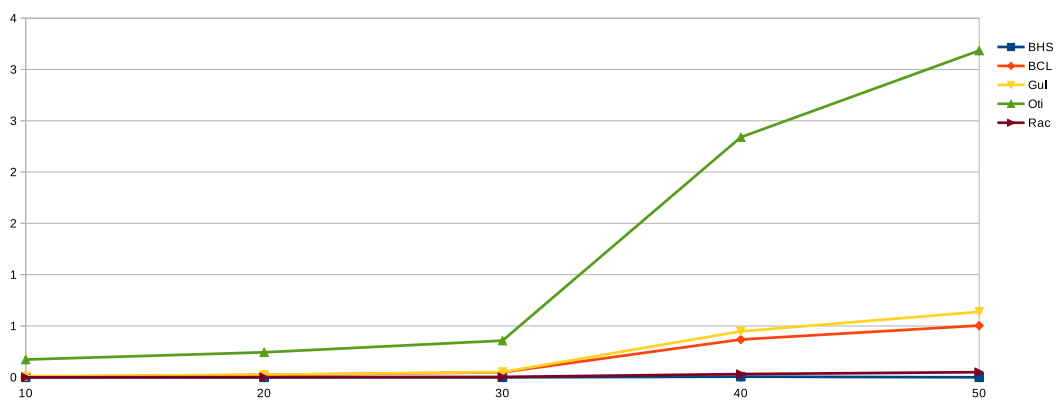


Figura 3.16: Tempo de processamento da *CPU* em segundos.

No entanto, como mencionado na Secção 3.5, observamos em vários casos que após a adição das n matrizes de posto um, a matriz resultante de posto completo era muitas vezes consideravelmente reduzida para cumprir a condição de $Q - Q' \succcurlyeq 0$ e, novamente, temos que o conjunto de nível elipsoidal correspondente à matriz Q' é muito diferente do conjunto de nível elipsoidal correspondente à matriz Q , levando a limites inferiores fracos.

A heurística gulosa *Gul* que propomos para tentar superar o problema do mau escalonamento que ocorre no método *BCL* parece ser uma boa alternativa. Os tempos de processamento da *CPU* permanecem competitivos e os limites calculados fizeram um grande passo na direção dos limites otimizados dados pelo algoritmo mais caro *Oti*, onde um problema de programação semidefinida é resolvido. Certamente, os grandes tempos para resolvê-los tornam proibitivo usar esses limites dentro de um algoritmo que utiliza uma metodologia de *Branch-and-Bound*.

Finalmente, vemos que no caso particular de nossas instâncias, onde as entradas em L são inteiras tomando v_i como as colunas de L leva a bons limites e baratos. É interessante notar que, neste caso, temos realmente $Q' = Q$, e os limites são dados por uma abordagem diferente, pois não consideramos a solução inteira da relaxação, mas em vez disso, um limite inferior melhor para o problema do que o limite dado pelo mínimo contínuo de sua função objetivo.

Capítulo 4

Extensão do limite elipsoidal para a programação quadrática convexa inteira irrestrita

Nossa principal contribuição neste capítulo é a extensão dos conceitos dados em [1, 3, 16], para o caso em que a matriz Q do Problema (2.1) é semidefinida positiva (PQIC). Em primeiro lugar, na Seção 4.1 fazemos a análise teórica que envolve a extensão dos limites elipsoidais para o caso convexo. Em seguida na Seção 4.2 apresentamos a adaptação dos algoritmos baseados em matrizes de posto um para serem utilizados no caso convexo. Finalmente na Seção 4.3 apresentamos os resultados computacionais.

4.1 Resultados teóricos

Nesta seção vamos detalhar as diferenças advindas do fato da matriz Q ser semidefinida positiva no Problema (2.1), assim como as complicações no cálculo de sua solução ótima. Uma das principais diferenças é o fato de o minimizador contínuo \bar{x} da função objetivo q não ser mais único.

Admitimos que a matriz Q é semidefinida positiva, e que $\text{posto}(Q) = r < n$, onde o valor de r é a quantidade de autovalores positivos da matriz Q . Também assumimos que $c \in \text{Im}(Q)$, para garantir que o problema tem solução.

Começamos caracterizando o conjunto dos minimizadores contínuos do Problema (2.1), utilizando as condições de otimalidade.

Lema 4.1.1. *O conjunto dos minimizadores de q no problema (2.1) é caracterizado da seguinte maneira,*

$$\bar{X} := \left\{ \bar{x} \in \mathbb{R}^n : \bar{x} = -\frac{1}{2}U_r(\Sigma_r)^{-1}U_r^T c + U_k t, \quad t \in \mathbb{R}^k \right\}$$

onde $r = \dim(\text{Im}(Q))$, $k = \dim(\text{Ker}(Q)) = n - r$, $Q = U\Sigma U^T$, $\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r)$, $U = [U_r \ U_k]$, onde $U_r = [u_1, \dots, u_r]$ são as r primeiras colunas de U e $U_k = [u_{r+1}, \dots, u_n]$ são as k últimas colunas de U .

Prova. Seja $Q = U\Sigma U^T$ a decomposição em valores singulares de Q , onde U é unitária e $\Sigma = \text{Diag}(\sigma_1, \dots, \sigma_n)$ é diagonal cujos elementos são os valores singulares de Q . Sem perda de generalidade, podemos assumir que $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0$, isto pelo fato de Q ter posto r .

Consideremos que pelas condições de otimalidade, os minimizadores \bar{x} de q satisfazem o sistema $Q\bar{x} = -\frac{1}{2}c$. Substituindo então Q por sua decomposição singular no sistema, obtemos

$$\begin{aligned} U\Sigma U^T \bar{x} &= -\frac{1}{2}c \\ \Sigma U^T \bar{x} &= -\frac{1}{2}U^T c \\ \Sigma y &= b \end{aligned}$$

onde $y = U^T \bar{x}$ e $b = -\frac{1}{2}U^T c$. Com a finalidade de simplificar as expressões, denotamos $\Sigma_r = \text{Diag}(\sigma_1, \dots, \sigma_r)$ e $y_r = (y_1, \dots, y_r)^T$, $y_k = (y_{r+1}, \dots, y_n)^T$, $b_r = (b_1, \dots, b_r)^T$, $b_k = (b_{r+1}, \dots, b_n)^T$. Observamos que como c pertence ao espaço imagem de Q , temos $b_k = 0$. Ao resolver o sistema $\Sigma y = b$ temos então

$$\begin{aligned} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_r \\ y_k \end{bmatrix} &= \begin{bmatrix} b_r \\ 0 \end{bmatrix} \\ \begin{bmatrix} y_r \\ y_k \end{bmatrix} &= \begin{bmatrix} (\Sigma_r)^{-1} b_r \\ t \end{bmatrix}, \quad t \in \mathbb{R}^k. \end{aligned}$$

Por outro lado, é possível escrever de forma alternativa a matriz U da seguinte maneira, $U = [U_r \ U_k]$, onde $U_r = [u_1, \dots, u_r]$ contem as r primeiras colunas de U e $U_k = [u_{r+1}, \dots, u_n]$ contem as k últimas colunas de U . Então, considerando que $b = \begin{bmatrix} b_r \\ 0 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} U_r^T c \\ 0 \end{bmatrix}$, ao substituir o valor de $b_r = -\frac{1}{2}U_r^T c$ na expressão de y temos

$$y = \begin{bmatrix} -\frac{1}{2}(\Sigma_r)^{-1}U_r^T c \\ t \end{bmatrix}, \quad t \in \mathbb{R}^k$$

Finalmente, substituindo o valor de y em $\bar{x} = Uy$, obtemos:

$$\begin{aligned} \bar{x} &= [U_r \ U_k] \begin{bmatrix} -\frac{1}{2}(\Sigma_r)^{-1}U_r^T c \\ t \end{bmatrix}, \quad t \in \mathbb{R}^k \\ \bar{x} &= -\frac{1}{2}U_r(\Sigma_r)^{-1}U_r^T c + U_k t, \quad t \in \mathbb{R}^k \end{aligned} \tag{4.1}$$

Logo, pelo fato de Q ter posto r o ponto \bar{x} ser escrito como $\bar{x} = \bar{x}_r + \bar{x}_k \in \bar{X}$, onde $\bar{x}_r = -\frac{1}{2}U_r(\Sigma_r)^{-1}U_r^T c$ é único e $\bar{x}_k = U_k t \in \text{Ker}(Q)$ com $t \in \mathbb{R}^k$. ■

Com a caracterização de \bar{x} descrita acima, generalizamos o Lema 3.1.1, para o caso em que Q é semidefinida positiva, como mostramos no seguinte lema

Lema 4.1.2. *Os minimizadores contínuos de $q(x) := x^T Q x + c^T x$, quando a matriz Q é semidefinida positiva, são dados por, $\bar{x} = \bar{x}_r + \bar{x}_k$, onde $\bar{x}_r = -\frac{1}{2}U_r(\Sigma_r)^{-1}U_r^T c$ e $\bar{x}_k = U_k t$, ($t \in \mathbb{R}^k$). O valor mínimo de q é dado por $q(\bar{x}) = c + \frac{1}{2}c^T \bar{x}_r$. Além disso, para qualquer $x \in \mathbb{R}^n$, $q(x) = q(\bar{x}) + (x - \bar{x})^T Q(x - \bar{x})$.*

Prova. *Seja $\bar{x} \in \bar{X}$ dado na Observação 4.1.1. Considerando que $Q\bar{x} = -\frac{1}{2}c$ e substituindo a expressão de \bar{x} em $q(x)$, obtemos*

$$\begin{aligned}
q(\bar{x}) &= \bar{x}^T Q \bar{x} + c^T \bar{x} + c \\
&= \bar{x}^T \left(-\frac{1}{2}c\right) + c^T \bar{x} + c \\
&= \frac{1}{2}c^T \bar{x} + c \\
&= \frac{1}{2}c^T \left(-\frac{1}{2}U_r(\Sigma_r)^{-1}U_r^T c + U_k t\right) + c \\
&= -\frac{1}{4}c^T U_r(\Sigma_r)^{-1}U_r^T c + \frac{1}{2}c^T U_k t + c \\
&= -\frac{1}{4}c^T U_r(\Sigma_r)^{-1}U_r^T c + c \\
&= c + \frac{1}{2}c^T \bar{x}_r
\end{aligned}$$

Note que, $c^T U_k t = 0$, porque $c \in \text{Im}(Q)$. Por outro lado, levando em conta que $Q\bar{x} = -\frac{1}{2}c$ e que $c^T \bar{x}_k = 0$, para todo $x \in \mathbb{R}^n$, temos

$$\begin{aligned}
q(\bar{x}) + (x - \bar{x})^T Q(x - \bar{x}) &= c + \frac{1}{2}c^T \bar{x} + x^T Q x - 2x^T Q \bar{x} + \bar{x}^T Q \bar{x} \\
&= c + \frac{1}{2}c^T \bar{x}_r + x^T Q x + c^T x - \frac{1}{2}c^T \bar{x}_r \\
&= x^T Q x + c^T x + c \\
&= q(x)
\end{aligned}$$

Portanto, $q(x) = q(\bar{x}) + (x - \bar{x})^T Q(x - \bar{x}) \forall x \in \mathbb{R}^n$ e $\bar{x} \in \bar{X}$. ■

Exemplo 4.1.1. *Seja o problema*

$$\begin{aligned}
\min \quad & q(x) = x_1^2 + 2x_1x_2 + x_2^2 + x_3^2 - x_1 - x_2 + x_3 \\
\text{s.a.} \quad & x \in \mathbb{Z}^3
\end{aligned}$$

Tomando os dados do problema acima, temos $c = (-1, -1, 1)^T$ e $Q = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

A decomposição em valores singulares da matriz Q é dada por

$$Q = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \overbrace{\begin{bmatrix} \sqrt{2}/2 & 0 & -\sqrt{2}/2 \\ \sqrt{2}/2 & 0 & \sqrt{2}/2 \\ 0 & 1 & 0 \end{bmatrix}}^U \overbrace{\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}^\Sigma \overbrace{\begin{bmatrix} \sqrt{2}/2 & 0 & -\sqrt{2}/2 \\ \sqrt{2}/2 & 0 & \sqrt{2}/2 \\ 0 & 1 & 0 \end{bmatrix}}^{U^T},$$

logo, utilizando então os resultados do Lema (4.1.2), concluímos que os minimizadores contínuos de q são dados por,

$$\bar{x} = \begin{bmatrix} 0.25 \\ 0.25 \\ -0.5 \end{bmatrix} + \begin{bmatrix} -\sqrt{2}/2 \\ \sqrt{2}/2 \\ 0 \end{bmatrix} t,$$

com $t \in \mathbb{R}^n$, e o valor do mínimo contínuo da função é $q(\bar{x}) = -0.5$.

Conforme vimos no Lema 4.1.2, para qualquer $x \in \mathbb{R}^n$,

$$q(x) = q(\bar{x}) + (x - \bar{x})^T Q (x - \bar{x}),$$

de modo que,

$$\min_{x \in \mathbb{Z}^n} q(x) = q(\bar{x}) + \min_{x \in \mathbb{Z}^n} (x - \bar{x})^T Q (x - \bar{x}) \quad (4.2)$$

$$= q(\bar{x}) + \mu^2(Q, \bar{x}); \quad (4.3)$$

conforme a definição de $\mu(Q, \bar{x})$ em (3.3).

Na Observação 4.1.1 mostramos um resultado muito interessante, o qual afirma que o valor do fator de escalonamento $\mu^2(Q, \bar{x})$ do elipsoide $E(Q, \bar{x})$ é totalmente independente do minimizador contínuo de q .

Observação 4.1.1. Dada a matriz Q , e considerando que $\bar{x} = \bar{x}_r + \bar{x}_k$ e que $c^T \bar{x}_k = 0$ onde $\bar{x}_r = -\frac{1}{2} U_r (\Sigma_r)^{-1} U_r^T c$ e $\bar{x}_k = U_k t$ determinamos a seguir $(x - \bar{x})^T Q (x - \bar{x})$, $\forall x \in \mathbb{Z}^n$. Assim

$$\begin{aligned} (x - \bar{x})^T Q (x - \bar{x}) &= x^T Q x + c^T x - \frac{1}{2} c^T \bar{x} \\ &= x^T Q x + c^T x - \frac{1}{2} c^T \bar{x}_r \\ &= x^T Q x + c^T x + \frac{1}{4} c^T U_r (\Sigma_r)^{-1} U_r^T c. \end{aligned}$$

Portanto ao substituir a expressão acima na equação (3.6) obtemos

$$\mu^2(Q, \bar{x}) = \max\{\alpha : x^T Q x + c^T x + \frac{1}{4} c^T U_r (\Sigma_r)^{-1} U_r^T c \geq \alpha^2, x \in \mathbb{Z}^n\}. \quad (4.4)$$

Pode-se ver que o valor de $\mu^2(Q, \bar{x})$ proposto em [1] para o caso onde Q é definida positiva também é independente da eleição do $\bar{x} \in \bar{X}$. No entanto, resolver o problema inteiro irrestrito do lado direito da equação (4.4) é tão difícil quanto resolver o problema irrestrito original (3.2). Se tivéssemos a capacidade de encontrar o ponto $x^* \in \arg \min_{x \in \mathbb{Z}^n} (x - \bar{x})^T Q (x - \bar{x})$; então teríamos que

$$\min_{x \in \mathbb{Z}^n} q(x) = q(\bar{x}) + (x^* - \bar{x})^T Q (x^* - \bar{x}). \quad (4.5)$$

Um caso mais simples se obtêm quando temos que a matriz Q é diagonal. A seguir, fazemos uma análise no caso hipotético da matriz Q ser diagonal. Na observação seguinte relacionamos os conceitos propostos em [1–3] e como pode ser calculada uma solução no caso de Q ser diagonal.

Observação 4.1.2. *Seja $Q = \text{Diag}(\tau)$, $\tau \geq 0$, então $x^* = \lfloor \bar{x}_r \rfloor$ é uma solução inteira do problema (4.5), onde $\bar{x}_r \in \text{Im}(Q)$.*

Assumindo que a matriz Q tem posto r , construímos a matriz diagonal Σ_r com os elementos não nulos da matriz Q . Logo a solução continua é dada por

$$\bar{x} = \begin{bmatrix} \bar{x}_r \\ \bar{x}_k \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}(\Sigma_r)^{-1}c_r \\ t \end{bmatrix}, \quad t \in \mathbb{R}^k \ (k = n - r);$$

ao substituir, temos

$$\begin{aligned} (x - \bar{x})^T \text{Diag}(\tau)(x - \bar{x}) &= (x_r - \bar{x}_r)^T \left(\sum_{i=1}^r \tau_i e_i e_i^T \right) (x_r - \bar{x}_r) \\ &= \sum_{i=1}^r \tau_i [e_i^T (x - \bar{x})]^2 \\ &= \sum_{i=1}^r \tau_i (x_i - \bar{x}_i)^2 \end{aligned}$$

e dado que o valor de $\mu(\text{Diag}(\tau), \bar{x})$ é o fator de escalonamento α tal que o elipsoide $\alpha E(\text{Diag}(\tau), \bar{x})$ contém algum ponto inteiro na sua fronteira, assim obtemos

$$\mu(\text{Diag}(\tau), \bar{x}) = \min \left\{ \sqrt{\sum_{i=1}^r \tau_i (x_i - \bar{x}_i)^2} : x_r \in \mathbb{Z}^r \right\}, \quad (4.6)$$

então, podemos ver que o ponto $x_r = \lfloor \bar{x}_r \rfloor \in \mathbb{Z}^r$ é o ponto que minimiza a expressão (4.6). Geometricamente quer dizer que ao escalar o elipsoide $E(Q, \bar{x})$ sobre o espaço

gerado pelos autovetores associados aos autovalores maiores do que zero, neste caso o espaço projetado $\text{Im}(\text{Diag}(\tau))$ é isomorfo ao espaço euclidiano \mathbb{R}^r . Assim o primeiro ponto inteiro atingido é o ponto $\lfloor \bar{x}_r \rfloor \in \mathbb{Z}^r$.

Portanto,

$$\mu^2(\text{Diag}(\tau), \bar{x}) = \sum_{i=1}^r \tau_i (\lfloor \bar{x}_i \rfloor - \bar{x}_i)^2,$$

logo, dado que o valor de $\mu^2(\text{Diag}(\tau), \bar{x})$ é independente de \bar{x}_k , temos

$$f(x^*) = f(\bar{x}) + \sum_{i=1}^r \tau_i (\lfloor \bar{x}_i \rfloor - \bar{x}_i)^2. \quad (4.7)$$

Podemos reparar que, de uma maneira intuitiva se sugeriu projetar o elipsoide diagonal sobre o espaço gerado pelos autovetores associados a os elementos da diagonal não nulos, que a su vez é isomorfo ao espaço \mathbb{R}^r .

Exemplo 4.1.2. *Dado o problema*

$$\begin{aligned} \min \quad & f(x) = 2x_1^2 + x_3^2 - x_1 + x_3 + 2 \\ \text{s.a.} \quad & x \in \mathbb{Z}^3; \end{aligned}$$

temos que a decomposição da matriz Q é

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}^U \overbrace{\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}^\Sigma \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}^{U^T}$$

além disso temos $c = [-1, 0, 1]^T$. Ao fazer os cálculos obtemos como solução os pontos de mínimos globais caracterizados pelo valor do parâmetro $t \in \mathbb{R}$,

$$\bar{x} = \begin{bmatrix} 0.25 \\ 0 \\ -0.5 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} t,$$

com $f(\bar{x}) = 1.625$. Logo ao arredondar o ponto \bar{x} , obtemos $\lfloor \bar{x} \rfloor = (0, 0, 0)^T$, e $f(x^*) = 1.625 + 0.375 = 2$.

Dado que o caso diagonal é muito particular, a seguir analisamos o fato de projetar o elipsoide degenerado sobre o espaço gerado pelos autovetores associados aos autovalores não nulos da matriz Q .

Assim, dado o elipsoide

$$E(Q, \bar{x}) := \{x \in \mathbb{R}^n : (x - \bar{x})^T Q (x - \bar{x}) \leq 1\};$$

vamos projetar o elipsoide sobre o espaço gerado pelos autovetores associados aos autovalores maiores do que zero. Isto é, utilizamos a matriz U_r , assim definimos o elipsoide projetado

$$E(Q_r, \bar{x}_r) := \{x_r \in \text{Im}(Q) : (x_r - \bar{x}_r)^T Q_r (x_r - \bar{x}_r) \leq 1\}; \quad (4.8)$$

onde $\bar{x}_r \in \text{Im}(Q)$ e $Q_r = U_r \Sigma_r U_r^T$, é a decomposição em valores singulares reduzida da matriz Q . Por outro lado, o fato de $x_r \in \text{Im}(Q)$ implica que existe um $y \in \mathbb{R}^r$ tal que $x_r = U_r y$, desta forma podemos escrever o elipsoide (4.8) como:

$$\begin{aligned} (x - \bar{x})^T Q (x - \bar{x}) &= (x_r - \bar{x}_r)^T Q_r (x_r - \bar{x}_r) \\ &= (U_r y - U_r \bar{y})^T Q_r (U_r y - U_r \bar{y}) \\ &= (y - \bar{y})^T U_r^T Q_r U_r (y - \bar{y}) \\ &= (y - \bar{y})^T U_r^T (U_r \Sigma_r U_r^T) U_r (y - \bar{y}) \\ &= (y - \bar{y})^T \Sigma_r (y - \bar{y}). \end{aligned}$$

Portanto,

$$E(Q_r, \bar{x}_r) \equiv E(\Sigma_r, \bar{y}) := \{y \in \mathbb{R}^r : (y - \bar{y})^T \Sigma_r (y - \bar{y}) \leq 1\} \quad (4.9)$$

Observação 4.1.3. Se $Q = \text{Diag}(\tau)$, então $\mu^2(Q, \bar{x}) = \mu^2(\Sigma_r, \bar{y})$.

Com efeito, dado que o elipsoide $E(\Sigma_r, \bar{y})$ é eixo paralelo com respeito a seu novo sistema de referência, temos

$$\begin{aligned} \mu^2(\Sigma_r, \bar{y}) &= ([\bar{y}] - \bar{y})^T \Sigma_r ([\bar{y}] - \bar{y}) \\ &= \left([U_r^T \bar{x}_r] - U_r^T \bar{x}_r \right)^T \Sigma_r \left([U_r^T \bar{x}_r] - U_r^T \bar{x}_r \right) \\ &= \left(U_r^T [\bar{x}_r] - U_r^T \bar{x}_r \right)^T \Sigma_r \left(U_r^T [\bar{x}_r] - U_r^T \bar{x}_r \right) \\ &= ([\bar{x}_r] - \bar{x}_r)^T U_r^T \Sigma_r U_r^T ([\bar{x}_r] - \bar{x}_r) \\ &= ([\bar{x}_r] - \bar{x}_r)^T Q_r ([\bar{x}_r] - \bar{x}_r) \\ &= \mu^2(Q_r, \bar{x}_r) \\ &= \mu^2(Q, \bar{x}) \end{aligned}$$

Ao assumir que Q é diagonal, então a matriz U_r é construída pelos vetores canônicos, desta forma $U_r^T \bar{x}_r = \bar{x}_r$ implica $[U_r^T \bar{x}_r] = U_r^T [\bar{x}_r]$ a qual é substituída na linha três das igualdades acima.

A questão que fica pendente é o que acontece quando Q não for diagonal. Será que a igualdade (4.5) ainda é válida, ou de outra forma, sob que condições é válida a igualdade.

Dado que no caso não diagonal resolver o problema de maneira direta é complicado demais, procuramos por elipsoides auxiliares como foi feito no capítulo anterior. Porém agora o elipsoide $E(Q)$ é degenerado, assim os conceitos de quase arredondamento não são aplicáveis neste caso.

A seguir propomos uma abordagem para calcular um limite inferior para o Problema (2.1) baseada na projeção do elipsoide $E(Q, \bar{x})$ sobre o subespaço gerado pelos autovetores da matriz Q .

Considerando o Lema 4.1.2 definimos a função subestimadora

$$q'(x) := q(\bar{x}) + (x - \bar{x})^T Q'(x - \bar{x}),$$

e procuramos por funções convexas quadráticas q' que subestime a função q . A principal dificuldade na extensão dos limites elipsoidais para funções quadráticas convexas, onde $\text{rank}(Q) < n$, pode ser feita pela seguinte generalização da Proposição 3.2.1.

Proposição 4.1.1. *Seja $Q = U\Sigma U^T$ a decomposição em valores singulares da matriz Q . Seja $r := \dim(\text{Im}(Q))$, $k := \dim(\text{ker}(Q)) = n - r$, $\Sigma_r := \text{Diag}(\lambda_1, \dots, \lambda_r)$, e $U := (U_r \ U_k)$, onde $U_r := (u_1, \dots, u_r)$ e $U_k := (u_{r+1}, \dots, u_n)$. Seja $v \in \mathbb{R}^n$ e $v = v^r + v^k$, onde $v^r \in \text{Im}(Q)$ e $v^k \in \text{ker}(Q)$. Seja δ^* o maior $\delta \in \mathbb{R}_+$ tal que $Q - \delta v v^T \succeq 0$.*

Se $v^k \neq 0$,

$$\delta^* = 0,$$

senão,

$$\delta^* = \frac{1}{\|v^T \Lambda^+\|_2^2} > 0,$$

onde Λ^+ é uma pseudoinversa pela direita de $\Sigma_r^{\frac{1}{2}} U_r^T$, dada por $U_r \Sigma_r^{-\frac{1}{2}}$.

Prova. *Seja $Q' := \delta^* v v^T$, e considere a condição de inclusão dada por $Q - Q' \succeq 0$, devemos ter $E(Q) \subseteq E(Q')$, ou equivalentemente $\delta^* x^T v v^T x \leq 1$, para todo $x \in \mathbb{R}^n$ tal que $x^T Q x \leq 1$. Desse modo,*

$$\delta^* = \min \left\{ \frac{1}{(v^T x)^2} : x^T Q x \leq 1 \right\} \quad (4.10)$$

$$= \frac{1}{\max \{(v^T x)^2 : x^T Q x = 1\}}. \quad (4.11)$$

Considerando $x = x_r + x_k$, onde $x_r \in \text{Im}(Q)$ e $x_k \in \text{ker}(Q)$, temos

$$x^T Q x = x_r^T U_r \Sigma_r U_r^T x_r = \|\Sigma_r^{1/2} U_r^T x_r\|_2^2,$$

e,

$$\delta^* := \frac{1}{\max \left\{ (v_r^T x_r + v_k^T x_k)^2 : \|\Sigma_r^{1/2} U_r^T x_r\|_2^2 = 1 \right\}}. \quad (4.12)$$

Por outro lado, seja $y := \Sigma_r^{1/2} U_r^T x_r$, e dado que $U_r U_r^T$ é a matriz de projeção sobre o espaço imagem de Q . Assim como, $x_r \in \text{Im}(Q)$, temos

$$x_r = U_r U_r^T x_r = U_r \Sigma_r^{-1/2} y.$$

Consequentemente,

$$\delta^* := \frac{1}{\max \left\{ (v_r^T U_r \Sigma_r^{-1/2} y + v_k^T x_k)^2 : \|y\|_2^2 = 1 \right\}}. \quad (4.13)$$

Portanto, se $v_k \neq 0$, o problema de maximização em (4.13) é ilimitado e $\delta^* = 0$. Caso contrário, a solução do problema de maximização é atingida em

$$y^T = \frac{v_r^T (U_r \Sigma_r^{-1/2})}{\|v_r^T U_r \Sigma_r^{-1/2}\|_2^2},$$

obtendo o valor de

$$\delta^* = \frac{1}{\|v_r^T U_r \Sigma_r^{-1/2}\|_2^2}. \quad (4.14)$$

■

Considerando a Proposição 4.1.1, nosso objetivo é gerar vetores inteiros v no espaço imagem de Q , e para o qual $v^T \Lambda$ são vetores curtos, dado que a matriz de transformação V leva um vetor do reticulado a outro vetor curto segundo a Equação (3.21). Para criar estes vetores, nós utilizamos o algoritmo de redução de base de um reticulado, e também a matriz de projeção sobre o espaço imagem de Q , como descrevemos na seguinte seção.

4.2 Algoritmos baseados em matrizes de posto um

Nesta seção, adaptamos para o caso convexo, os algoritmos que geram a matriz Q' pela adição de matrizes de posto um, que foram propostas na Seção 3.4. As matrizes geradas são da forma $Q' = \sum_{i=1}^s \delta_i v_i v_i^T$, onde $\delta_i \in \mathbb{R}_+$, $v_i \in \text{Im}(Q) \cap \mathbb{Z}^n$, para $i = 1, \dots, s$, $s \leq n$, e $Q - Q' \succcurlyeq 0$.

Como no caso estritamente convexo, os limites inferiores são dados por

$$q(x) \geq q'(x) \geq q(\bar{x}) + \sum_{i=1}^s \delta_i \left(\lfloor v_i^T \bar{x} \rfloor - v_i^T \bar{x} \right)^2 \quad (4.15)$$

A adaptação dos três algoritmos apresentados para o caso estritamente convexo são descritos nos seguintes itens, e referidos nos testes computacionais como Gul^+ , Oti^+ e Rac^+ .

1. Nos algoritmos Gul^+ , Oti^+ , selecionamos r colunas linearmente independentes da matriz Λ^{+T} que compõem a matriz não singular $\Theta \in \mathbb{R}^{r \times r}$, sendo Λ^+ é a pseudoinversa da matriz $\Sigma^{\frac{1}{2}} U^T$, segundo a Proposição 4.1.1.

A seguir, selecionamos os vetores ω_i tal que $z_i := \Theta \omega_i$, para $i = 1, \dots, r$, que compõe a base reduzida para o reticulado gerado pelas colunas da matriz Λ^{+T} . Obtemos então os vetores $v_i \in Im(Q) \cap \mathbb{Z}^n$ através da projeção dos vetores ω_i sobre o espaço imagem de Q , isto é, definimos $v_i = \alpha L \omega_i$, onde $\alpha L \in \mathbb{Z}^{n \times r}$

- (a) Como foi feito nos algoritmos Gul e Oti , definimos δ_i , para cada $i = 1 \dots r$, como o máximo valor para o qual $Q - \delta_i v_i v_i^T \succcurlyeq 0$, isto é, $\delta_i := \frac{1}{\|v_i^T \Lambda^+\|_2^2}$. Então, utilizamos um algoritmo guloso para construir a matriz $Q' = \sum_{i=1}^s \delta_i v_i v_i^T$.

As matrizes de posto um na forma $\delta_i v_i v_i^T$ são iterativamente adicionadas à matriz Q' , e cada uma é escolhida para incrementar ao máximo o limite inferior, podendo ser adicionadas aquelas que violam a restrição de inclusão $Q - Q' \succcurlyeq 0$.

- (b) Como foi feito em Oti , neste algoritmo Oti^+ , os valores de δ_i , para $i = 1, \dots, r$, são dados pela solução do problema de programação semidefinida

$$\max \left\{ \sum_{i=1}^r \delta_i \left(\lfloor v_i^T \bar{x} \rfloor - v_i^T \bar{x} \right)^2 : Q - \sum_{i=1}^r \delta_i v_i v_i^T \succcurlyeq 0, \delta_i \geq 0 \right\} \quad (4.16)$$

2. Finalmente, como foi feito no Rac , em Rac^+ , selecionamos os vetores $v_i \in \mathbb{Z}^n$ como $\alpha_i L_i$. Neste caso, $s = r$, e o máximo valor para cada δ_i , tal que $Q - \sum_{i=1}^r \delta_i v_i v_i^T \succcurlyeq 0$, é trivial e dado por $\delta_i = \frac{1}{\alpha_i^2}$

Vale ressaltar que tomamos uma base do espaço imagem de Q dado pelas colunas de $U_r \Sigma_r^{\frac{1}{2}}$, e procuramos pelo vetor $v = U_r \Sigma_r^{\frac{1}{2}} w$, com $w \in \mathbb{R}^k$, e $v \in \mathbb{Z}^n$. Neste caso, $w = \Sigma_r^{-\frac{1}{2}} U_r^T v = \Lambda^{+T} v$, e portanto, de modo a obter valores grandes para δ^* calculamos os vetores curtos w no reticulado gerado pelas colunas de Λ^{+T} . Desta maneira aplicamos o algoritmo de redução de base sobre a matriz Λ^{+T} .

4.3 Testes Computacionais

Nesta seção comparamos numericamente os três algoritmos que calculam os limites inferiores para o caso em que a matriz da função objetivo é semidefinida positiva, descritos na última seção (Gul^+ , Oti^+ e Rac^+), também apresentamos uma adaptação do algoritmo BCL , descrita na Seção 3.2, ao problema convexo. Nesta adaptação os vetores v_i são calculados exatamente como descrito nos algoritmos Gul^+ e Oti^+ , da última seção, o que significa que apenas r vetores são obtidos pelo algoritmo de redução de base, e em seguida são projetados sobre o espaço imagem da matriz Q .

Não foi possível adaptar o algoritmo BHS ao problema convexo ($PQIC$). A dificuldade em generalizar a classe de funções quadráticas subestimadoras com a propriedade de arredondamento forte para o caso onde o posto da matriz Q é menor que n , vem da degeneração dos conjuntos de subnível elipsoidais das funções subestimadoras quadráticas q' de q . Dado que, $Q - Q' \succcurlyeq 0$, temos que o posto da matriz Q' também é menor que n . Essa degeneração torna impossível para os conjuntos de subnível de q' satisfazer a caracterização de quase arredondado na Definição 3.3.3. Note também que para cumprir (3.28) no Corolário 3.3.1 devemos ter $\lambda_{\min} > 0$.

O ambiente de teste para os problemas convexos é semelhante ao descrito na Seção 3.4.2 para o caso estritamente convexo. Nestes testes o vetor c é gerado aleatoriamente e projetado sobre o espaço imagem de Q , a fim de garantir a existência do mínimo contínuo \bar{x} . Além disso, a matriz fator $L \in \mathbb{R}^{n \times r}$ é de posto $r < n$.

A Tabela 4.1 mostra o valor das médias dos $gaps$, segundo a Equação (3.39), obtidas por cada um dos métodos. Cabe ressaltar que quanto maior o gap , melhor é o limite inferior obtido pelo método. Já na Tabela 4.2 apresentamos o valor das

Tabela 4.1: Limites inferiores, média dos $gaps$ (%).

n	BCL	Gul^+	Oti^+	Rac^+
10	19.1904	38.0017	38.1181	69.7029
20	17.3467	51.2494	51.4197	88.4769
30	14.2389	44.0476	44.6794	97.2279
40	17.6325	55.539	56.4508	99.7759
50	12.5665	49.0349	50.0018	95.8057

médias de tempo de processamento da CPU (em segundos) gasto no cálculo dos limites inferiores para cada um dos métodos propostos, a primeira coluna da tabela corresponde ao valor da dimensão n das instâncias. O posto da matriz Q é dado por $r = 0.5n$.

De forma gráfica, na Figura 4.1 mostramos as curvas dos valores da Tabela 4.1, apresentamos a melhoria média relativa nos limites inferiores dada por cada método

Tabela 4.2: Tempo médio de *CPU* (em segundos).

n	BCL	Gul^+	Oti^+	Rac^+
10	0	0	0.1591	0
20	0.0125	0.025	0.3401	0.0062
30	0.0172	0.0952	0.3728	0
40	0.142	0.1045	1.2324	0.0016
50	0.2465	0.22	1.4524	0.0078

para o problema convexo, quando é comparado ao limite dado pelo mínimo contínuo, (medido segundo a equação (3.39)). Por outro lado, na Figura 4.2, mostramos a média do tempo de CPU (em segundos) para calcular os limites, segundo os valores da Tabela 4.2.

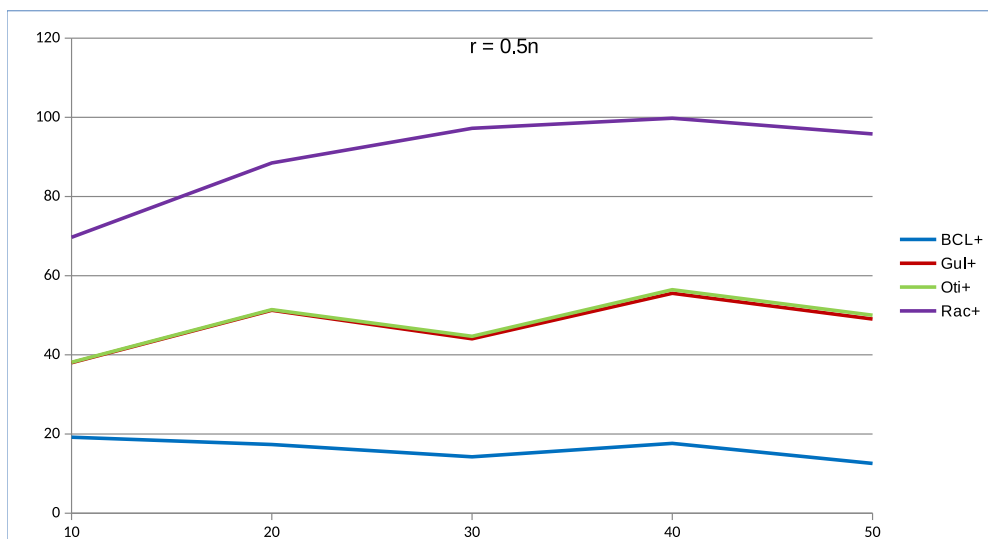


Figura 4.1: Melhoria nos limites inferiores em relação ao mínimo contínuo.

Podemos observar nas Figuras 4.1 e 4.2 que a adaptação da BCL não é competitiva com a estratégia gulosa de adicionar as matrizes de posto um, utilizadas no algoritmo Gul^+ .

Os tempos computacionais para ambos os algoritmos são muito semelhantes, mas os limites calculados pelo algoritmo Gul^+ são muito melhores para essas instâncias. De fato, o algoritmo Gul^+ tem uma melhor performance nestes testes, já que seus limites estão agora muito próximos dos calculados pela solução do problema de programação semidefinida no algoritmo Oti^+ . É novamente interessante observar que a integralidade das entradas de L pode ser bem explorada pelo algoritmo Rac^+ , que gera neste caso bons limites e muito mais rápido.

Adicionalmente, também geramos 10 testes com $n = 50$ para cada valor de r no conjunto $\{0.1n, 0.2n, \dots, 0.9n\}$ e relatamos novamente a média dos resultados. Já de

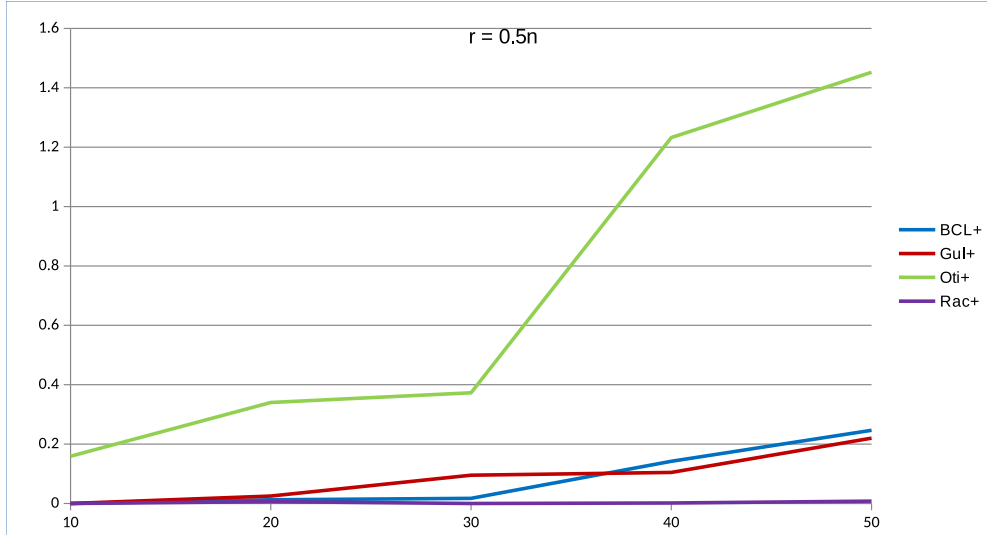


Figura 4.2: tempo *CPU* no cálculo dos limites inferiores (segundos).

forma mais detalhada nas Tabelas 4.3 e 4.4, mostramos os valores correspondentes aos *gaps* e tempos de processamento da *CPU*, onde a primeira coluna de cada uma das tabelas corresponde ao valor de μ , que representa a proporção do posto da matriz Q com respeito a sua dimensão, por exemplo $\mu = 0.4$ significa que o posto da matriz Q é $r = 0.4n$.

Tabela 4.3: Tempo médio de *CPU* (em segundos).

μ	<i>BCL</i>	<i>Gul</i> ⁺	<i>Oti</i> ⁺	<i>Rac</i> ⁺
0.1	0.0062	0.0047	0.9001	0
0.2	0.0562	0.0484	1.1388	0.0125
0.3	0.1248	0.1186	1.2199	0
0.4	0.1794	0.103	1.1825	0.0125
0.5	0.3276	0.1326	1.5491	0.0062
0.6	0.4976	0.3604	2.2667	0.0062
0.7	0.4306	0.5054	2.6005	0.025
0.8	0.624	0.6084	2.8969	0
0.9	0.6958	0.7223	3.1918	0.025

Finalmente, nas Figuras 4.3 e 4.4, analisamos como o posto da matriz Q interfere nos resultados obtidos. Observamos que são obtidos limites relativamente melhores para postos pequenos. Quando r aumenta, os conjuntos de níveis elipsoidais correspondentes às matrizes Q e Q' tendem a ficar mais diferentes uns dos outros. Por exemplo, analisando o Algoritmo *BCL*, podemos esperar que β^* fique menor quando r aumenta, de forma a garantir $Q - Q' \succcurlyeq 0$. Além disso, como esperado, os limites são mais rapidamente calculados para valores menores de r . A partir da descrição

Tabela 4.4: Limites inferiores, média dos *gaps*(%).

μ	<i>BCL</i>	<i>Gul</i> ⁺	<i>Oti</i> ⁺	<i>Rac</i> ⁺
0.1	45.9082	75.863	78.3157	100
0.2	25.0596	60.571	61.4581	98.0594
0.3	16.7025	51.4716	52.6614	97.8081
0.4	18.2531	61.1017	62.0397	100
0.5	12.5665	49.0349	50.0018	95.8057
0.6	13.847	52.3258	52.7083	92.5183
0.7	12.436	51.1074	51.4697	92.0015
0.8	6.275	29.0412	29.3765	69.7755
0.9	2.8379	15.5852	15.9514	48.3106

dos algoritmos, é claro que o esforço computacional de todos os algoritmos é proporcional ao valor do posto r da matriz, principalmente para os algoritmos *Gul*⁺ e *Oti*⁺.

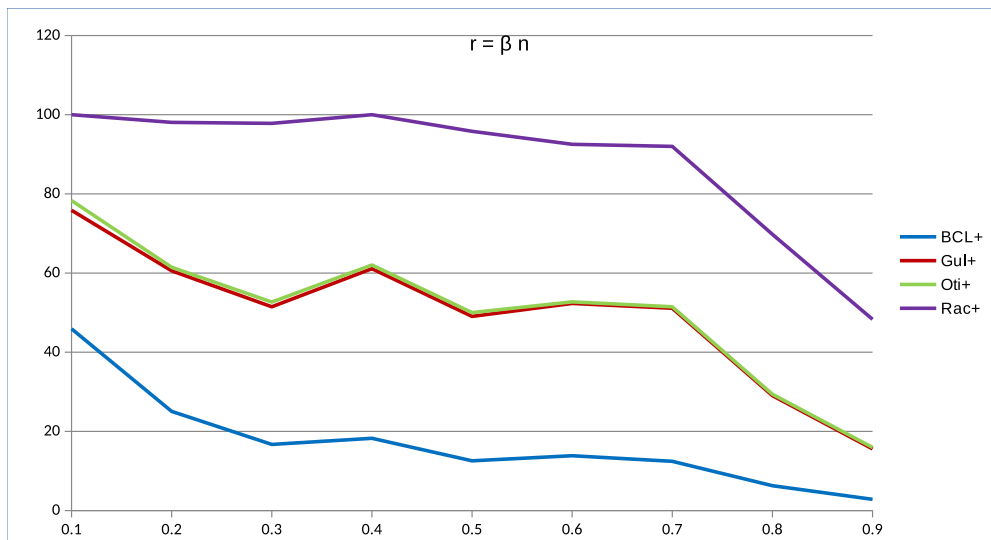


Figura 4.3: Melhoria nos limites inferiores em relação ao mínimo contínuo.

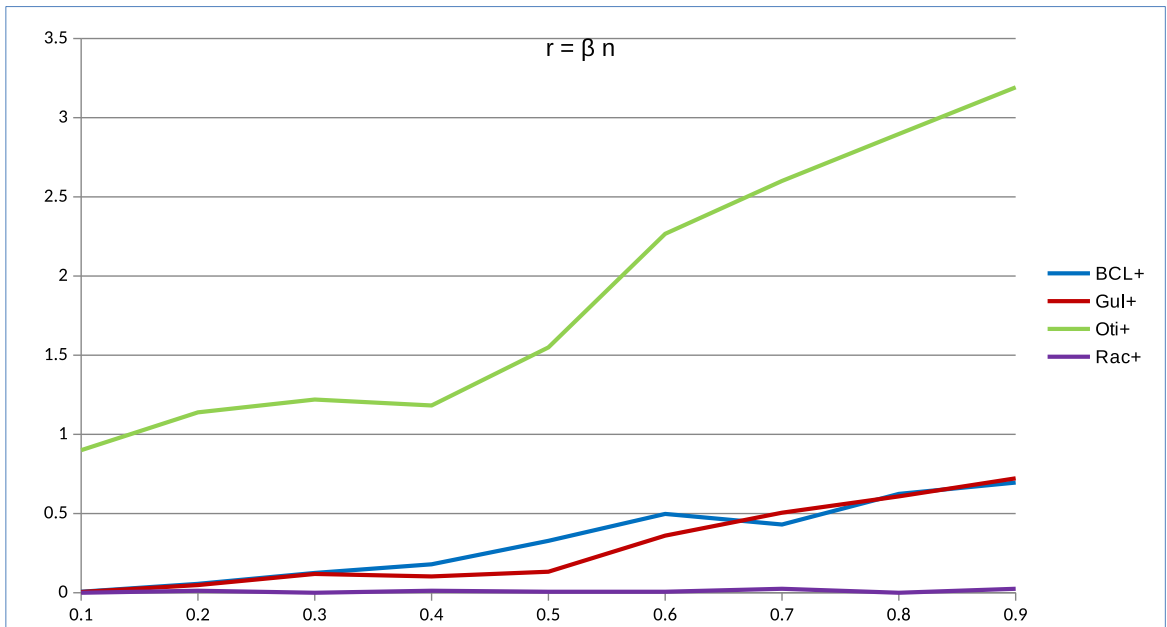


Figura 4.4: tempo *CPU* no cálculo dos limites inferiores (segundos).

Capítulo 5

Aplicação dos limites elipsoidais na programação quadrática inteira não convexa com restrições de caixa

Nosso objetivo final neste trabalho é aplicar as ideias anteriores para calcular limites inferiores para problemas de programação inteira quadrática não-convexas com restrições de caixa, (PIQNC):

$$\begin{cases} \min & q(x) = x^T Q x + c^T x \\ \text{s.a.} & l \leq x \leq u \\ & x \in Z^n \end{cases} \quad (5.1)$$

onde $Q \in \mathbb{R}^{n \times n}$ é simétrica e indefinida, $c \in \mathbb{R}^n$, $l, u \in \mathbb{Z}^n$, e $l < u$. Sejam $\lambda_1 \geq \dots \geq \lambda_r \geq 0 > \lambda_{r+1} \geq \dots \geq \lambda_n$ os autovalores de Q .

5.1 Estratégia de decomposição

A fim de construir uma relaxação para o problema (5.1), inicialmente reescrevemos a matriz Q como $Q + D - D$, onde D é diagonal e semidefinida positiva e $Q + D$ é semidefinida positiva.

A ideia é aplicar uma decomposição DC à matriz Q , expressando $q(x)$ como a diferença de duas funções convexas, ou de modo equivalente, como a soma da função quadrática convexa $q_1(x) = x^T(Q + D)x + c^T x$ e a função quadrática côncava separável $q_2(x) = -x^T D x$. O uso de decomposições DC no cálculo dos limites para problemas quadráticos não convexos têm sido amplamente utilizados na literatura [21, 22] e estratégias de decomposição da função quadrática q foram recentemente

empírico e teoricamente analisados em [23].

Ao fazer a decomposição da matriz Q da função objetivo do problema (5.1), como $Q = Q + D - D$, onde $P := Q + D$ é semidefinida positiva e $D := \text{Diag}(d_1, \dots, d_n)$ é diagonal e definida positiva, reformulamos o problema como

$$\begin{cases} \min & q(x) = x^T P x + c^T x - \sum_{i=1}^n d_i x_i^2 \\ \text{s.a.} & l \leq x \leq u, \\ & x \in \mathbb{Z}^n \end{cases} \quad (5.2)$$

Utilizando a aproximação secante para subestimar cada função côncava univariada $-x_i^2$ no intervalo $[l_i, u_i]$, temos que

$$- \left((x_i - l_i) \frac{u_i^2 - l_i^2}{u_i - l_i} + l_i^2 \right) \leq -x_i^2 \quad (5.3)$$

Logo,

$$\begin{aligned} q(x) &\geq x^T P x + c^T x - \sum_{i=1}^n d_i \left((x_i - l_i)(u_i + l_i) + l_i^2 \right) \\ &= x^T P x + c^T x - \sum_{i=1}^n d_i (u_i + l_i) x_i + \sum_{i=1}^n d_i l_i (u_i + l_i) - \sum_{i=1}^n d_i l_i^2 \\ &= x^T P x + c^T x - \sum_{i=1}^n d_i (u_i + l_i) x_i + \sum_{i=1}^n d_i l_i u_i \\ &= x^T P x + \left(c - \sum_{i=1}^n d_i (u_i + l_i) \right)^T x + l^T D u \\ &= x^T P x + (c - D(u + l))^T x + l^T D u \\ &= x^T P x + \tilde{c}^T x + k \end{aligned}$$

onde $\tilde{c} = c - D(u + l)$ e $k = l^T D u$.

Finalmente substituindo as funções côncavas por estas aproximações secantes, obtemos a seguinte relaxação quadrática inteira convexa de (5.1)

$$\begin{cases} \min & p(x) = x^T P x + \tilde{c}^T x + k \\ \text{s.a.} & l \leq x \leq u, \\ & x \in \mathbb{Z}^n, \end{cases} \quad (5.4)$$

Note que no caso em que todas as variáveis são binárias, temos $k = 0$ e $\tilde{c}_j = c_j - d_j$ para todo $j = 1, \dots, n$.

Em pesquisas em que decomposições DC de funções quadráticas não conve-

xas, juntamente com a aproximação secante da parte côncava, são utilizadas para calcular limites para o problema (5.1), a forma habitual para se obter o limite inferior é relaxar ainda mais as restrições de integralidade nas variáveis em (5.4) e resolver o problema quadrático convexo contínuo derivado (ver, por exemplo, [23] e referências).

Propomos neste capítulo utilizar os limites inferiores elipsoidais como uma alternativa para calcular limites para o problema (PIQNC), bem como, através da aplicação das ideias apresentadas nos capítulos anteriores, limitar o problema de programação quadrática inteira convexa (5.4).

Vamos denotar agora o minimizador contínuo da função objetivo p do problema (5.4) por \bar{x} , e seja p' o subestimador quadrático convexo de p , construídos tal como nos capítulos 3 e 4, que corresponde ao desenvolvimento no caso estritamente convexo e convexo, respectivamente.

Temos

$$p'(x) := p(\bar{x}) + (x - \bar{x})P'(x - \bar{x}),$$

onde $P' = \sum_{i=1}^s \delta_i^* v_i v_i'$, e $v_i \in \mathbb{Z}^n \setminus \{0\}$, para todo $i = 1, \dots, s$.

Como enfatizado em [2], os limites propostos nos capítulos 3 e 4, podem ser melhorados se as restrições de caixa forem consideradas, porque, para todo $x \in \mathbb{R}^n$,

$$p'(x) = p(\bar{x}) + \sum_{i=1}^s \delta_i^* (v_i^T x - v_i^T \bar{x})^2, \quad (5.5)$$

e, para todo $x \in \mathbb{Z}^n$, tal que $l \leq x \leq u$,

$$(v_i^T x - v_i^T \bar{x})^2 \geq \left(\max \left\{ \left| v_i^T \bar{x} \right| - v_i^T \bar{x}, (v_i^{\min} - v_i^T \bar{x}), (v_i^T \bar{x} - v_i^{\max}) \right\} \right)^2,$$

onde

$$v_i^{\min} = \min \{ v_i^T x, x \in [l, u] \} = \sum_{j=1}^n \min \{ v_{i_j} l_j, v_{i_j} u_j \},$$

e

$$v_i^{\max} = \max \{ v_i^T x, x \in [l, u] \} = \sum_{j=1}^n \max \{ v_{i_j} l_j, v_{i_j} u_j \}.$$

Um ponto muito importante é que, para aplicar estas ideias para obter limites inferiores para o problema (5.4), é requerido que a função objetivo do problema tenha um minimizador contínuo.

5.1.1 Estratégias de decomposição diagonal

Nesta subseção, consideramos apenas decomposições do tipo $Q = P - D$, onde D é diagonal. A razão para esta escolha é evitar a inclusão de restrições lineares na

relaxação (5.2), ao substituir funções côncavas multivariadas por subestimadores secantes lineares na função objetivo do problema (5.1). A fim de decompor a matriz $Q = P - D$ e obter uma matriz semidefinida positiva P , aplicamos as duas estratégias seguintes:

1. Seleccionamos a matriz diagonal D que faz P definida positiva. Por isso definimos

$$D := (-\lambda_n + \epsilon)I, \quad (5.6)$$

com $\epsilon > 0$. Neste caso $P := Q + D$ é definida positiva, com os autovalores trasladados com respeito a Q , isto é, o menor autovalor de P é igual a ϵ e seu maior autovalor é igual a $\lambda_1 - \lambda_n + \epsilon$.

2. Fixando $D := -\lambda_n I$, neste caso $P := Q + D$ é semidefinida positiva, também com os autovalores trasladados com respeito a Q , mas com autovalores iguais a zero. Para garantir que a função objetivo tenha minimizadores contínuos, substituímos \tilde{c} por $\tilde{c}^r := Proj_P(\tilde{c})$, que é a projeção de \tilde{c} sobre o espaço imagem da matriz P , isto é, calculamos a solução de $\min_x \|\tilde{c} - Px\|_2$, dada por $x = P^+ \tilde{c}$, onde P^+ é uma pseudoinversa da matriz P . Assim, $\tilde{c}^r := PP^+ \tilde{c}$, e definindo $\tilde{c}^k := \tilde{c} - \tilde{c}^r$, finalmente substituímos k por k' , onde

$$k' := k + \sum_{j=1}^n \min\{\tilde{c}_j^k l_j, \tilde{c}_j^k u_j\} \quad (5.7)$$

Que assegura que a solução de (5.4) é ainda um limite inferior para (NCQIP).

5.2 Testes Computacionais

Nesta seção analisamos os resultados obtidos pelas estratégias descritas na última seção para obter limites para os problemas de programação quadrática inteira não convexa com restrições de caixa. Em nossos experimentos consideramos problemas quadráticos binários, com restrições de caixa $[0, 1]^n$ em todos os testes.

Na Tabela 5.1 mostramos para cada valor n que representa a dimensão da matriz Q , a média das melhorias relativas dos limites obtidos pelos três algoritmos, BCL^+ , Gul^+ e Oti^+ , neste caso a matriz P é semidefinida positiva ($\epsilon = 0$) e por BCL , Gul e Oti , no caso em que P é definida positiva ($\epsilon = 0.01$). O valor de gap_{LB} é calculado segundo a Equação (5.8).

Nas Tabelas 5.3 e 5.4 apresentamos os tempos de processamento da CPU para cada um dos métodos e para cada caso da matriz P . Isto é, $P \succcurlyeq 0$ e $P \succ 0$, respectivamente.

Tabela 5.1: Média dos *gaps* quando $P \succcurlyeq 0$, ($\epsilon = 0$).

n	Gul^+	Oti^+	BCL
10	46.3789	49.1364	28.0483
20	29.2032	31.3444	11.4922
30	25.9098	27.438	7.3531
40	19.0797	19.8583	4.1205
50	20.935	22.0585	3.7071

Tabela 5.2: Média dos *gaps* quando $P \succ 0$, ($\epsilon = 0.01$).

n	Gul^+	Oti^+	BCL
10	23.0013	30.138	9.0638
20	37.9904	45.0617	9.3185
30	33.4329	40.695	5.0024
40	43.3604	58.9793	11.0253
50	36.2471	49.8087	9.9167

Tabela 5.3: Tempos da *CPU* quando $P \succcurlyeq 0$, ($\epsilon = 0$).

n	Gul^+	Oti^+	BCL
10	0.0047	0.1778	0.0047
20	0.0328	0.2746	0.025
30	0.0624	0.3962	0.064
40	0.39	2.5896	0.4867
50	0.5023	3.7159	0.5663

Tabela 5.4: Tempos da *CPU* quando $P \succ 0$, ($\epsilon = 0.01$).

n	Gul^+	Oti^+	BCL
10	0.0031	0.2012	0.0062
20	0.0328	0.3058	0.0359
30	0.1123	0.4696	0.1092
40	0.6396	3.1965	0.6739
50	0.6833	3.8766	0.7753

Definimos $\epsilon = 0.01$ nestes testes porque valores menores para este parâmetro geraram erros numéricos em nosso resultados. Dez testes aleatórios para cada valor de $n \in \{10, 20, \dots, 50\}$ foram considerados, para os quais definimos a matriz $Q - \lambda_n I := LL^T$, onde $L \in \mathbb{Q}^{n \times n-1}$. As instâncias permitiram comparar as duas estratégias, correspondendo a $\epsilon > 0$ e $\epsilon = 0$.

Na Figura 5.1 mostramos de forma gráfica os valores obtidos na Tabela 5.1.

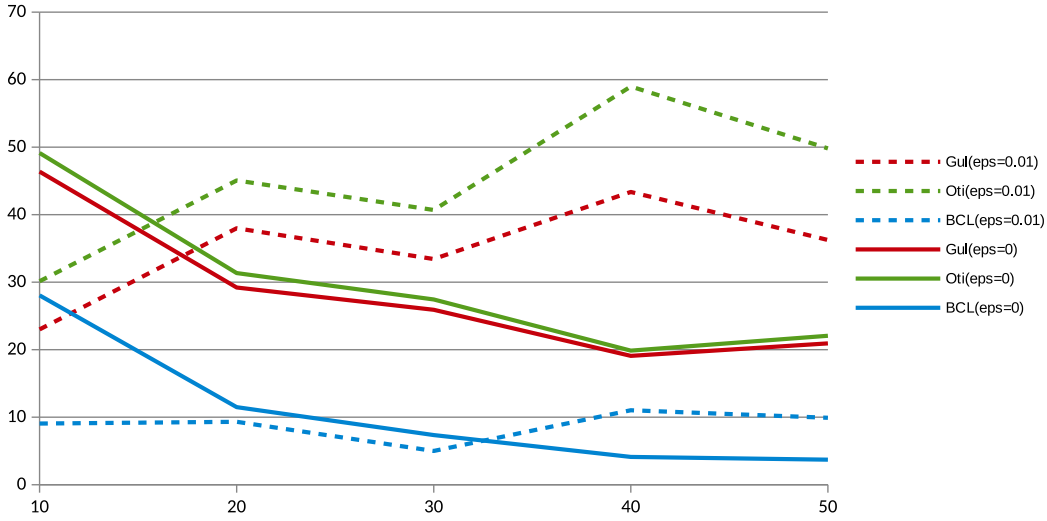


Figura 5.1: Melhoria nos limites inferiores comparado ao mínimo contínuo.

Observamos que para obter uma comparação justa entre as duas estratégias, as melhorias nos limites inferiores mostradas na Figura 5.1 são sempre comparadas com o mínimo contínuo $p(\bar{x})$ da função objetivo $p(x) := x^T P x + \tilde{c}^T x$, considerando a matriz semidefinida positiva P , correspondente a $\epsilon = 0$. Mais especificamente, nós relatamos na Figura 5.1,

$$gap_{LB} := \frac{LB - LB^-}{|LB^-|} \times 100\%, \quad (5.8)$$

onde LB é o limite inferior obtido por cada algoritmo e para cada ϵ , e $LB^- := k' + p(\bar{x})$, onde k' é definida em (5.7).

Na Figura 5.2 apresenta-se a media do tempo da *CPU* em segundos gasto no cálculo dos limites inferiores.

Quando comparamos os resultados obtidos para $\epsilon = 0$ e $\epsilon = 0.01$, vemos que as melhorias dos limites com respeito aos mínimos são na media mais significativas do que no caso estritamente convexo ($\epsilon = 0.01$). Isto pode ser explicado pelo fato que quando projetamos os vetores ω_i , tal que os vetores $r_i := \Theta \omega_i$ são curtos, sobre o espaço imagem de P , os vetores resultantes v_i correspondentes aos vetores modificados $r_i := \Theta v_i$ podem não ser tão curtos quanto r_i , levando a limites mais fracos.

No entanto, consideráveis melhorias nos limites dado pelos mínimos contínuos são ainda obtidos no caso convexo. O pequeno aumento nos tempos computacionais para o caso estritamente convexo, pode ser explicado pelo incremento do posto da matriz P , de $n - 1$ para n . Conforme discutido na Seção 4.3 do Capítulo 4, o esforço computacional de todos os algoritmos é proporcional ao posto da matriz.

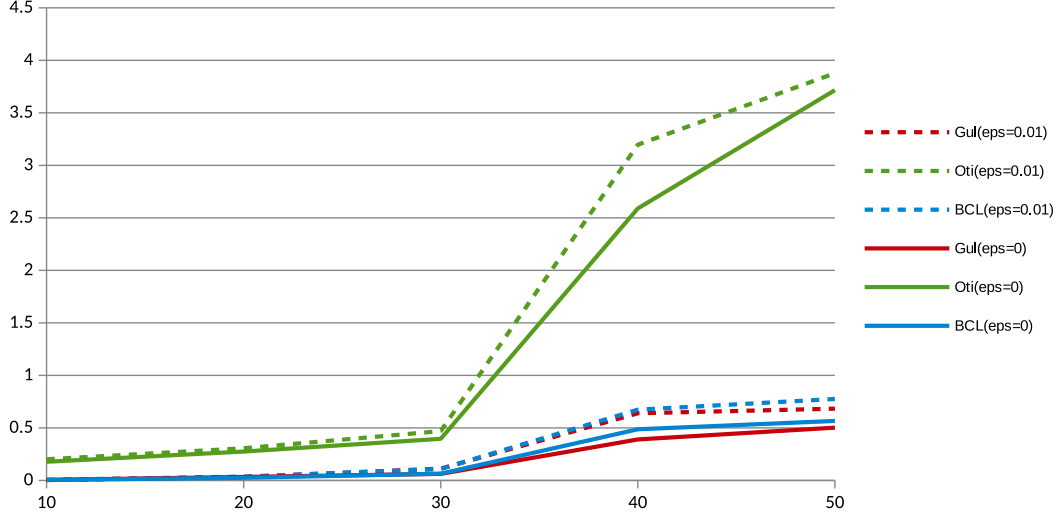


Figura 5.2: Tempo médio de *CPU* (em segundos).

Finalmente, considerando tanto a qualidade dos limites como o tempo computacional, vemos que o algoritmo *Gul* combinado com P definida positiva é a melhor estratégia.

Em nossos últimos experimentos, aplicamos essa estratégia para calcular limites para 10 instâncias de *Benchmark* do problema quadrático binário com $n = 50$, tomadas da biblioteca de instâncias *Big Mac Library* [24] correspondente as instâncias *Beasley*. Mostramos na Tabela 5.5 o nome do problema, o valor da solução ótima (z^*), o decréscimo relativo no *gap* dual comparando o limite calculado pelo algoritmo *Gul* para o limite contínuo $p(\bar{x})$, que é especificamente definido como

$$gap := \frac{z^* - LB}{z^* - p(\bar{x})} \times 100\% \quad (5.9)$$

e o tempo computacional em segundos para calcular os limites. Notamos que neste experimento final, utilizamos $\epsilon = 0.0001$, que não gerou erros numéricos e deu melhores resultados.

Vemos a partir dos resultados da Tabela 5.5 que nossa estratégia levou a melhorias nos limites contínuos dados por $p(\bar{x})$. Os tempos computacionais seguem o mesmo padrão observado para os outros dois problemas discutidos neste trabalho, ou seja, para o caso estritamente convexo e para o caso convexo.

Tabela 5.5: Resultados das instâncias *Beasley* de *BiqMacLibrary*($n = 50$).

<i>Problema</i>	z^*	<i>gap</i>	<i>CPU</i> (segundos)
bqp50-1	-2098	0.0395	0.5928
bqp50-2	-3702	0.0374	0.7020
bqp50-3	-4626	0.0004	0.4992
bqp50-4	-3544	0.0205	0.6396
bqp50-5	-4012	0.0594	0.6240
bqp50-6	-3693	0.0238	0.6240
bqp50-7	-4520	0.0178	0.6396
bqp50-8	-4216	0.0129	0.9828
bqp50-9	-3780	0.0200	0.4680
bqp50-10	-3507	0.0064	0.3900

Capítulo 6

Conclusões e recomendações

Neste trabalho, inicialmente fizemos uma rápida revisão dos limites elipsoidais para o problema de programação quadrática inteira estritamente convexo *PQIEC* proposta por Buchheim, Caprara, and Lodi in [1], e por Buchheim, Hübner, e Schöbel em [2].

A abordagem interessante utilizada pelos autores para gerar limites inferiores para *PQIEC* consiste em construir relaxações inteiras quadráticas para o problema, onde a função objetivo $q(x) = x^T Q x + c^T x$ do problema original é subestimada por outra função quadrática com o mesmo minimizador contínuo de q , e para o qual um mínimo inteiro pode ser calculado de forma fácil e barata.

Depois de analisar os resultados numéricos para obter uma visão das metodologias, propomos modificações no algoritmo proposto em [2] que geram melhorias nos limites em nossas experiências numéricas sem aumento significativo nos tempos computacionais. Os resultados apresentados na literatura têm uma forte base na interpretação geométrica dos conjuntos de subnível elipsoidal das funções quadráticas. Ainda com base nessa interpretação intuitiva, discutimos por que a estratégia desenvolvida em [1] levou aos piores limites em nossas experiências. Também mostramos como a matriz $Q = LL^T$ com um fator racional L pode ser explorada para gerar limites ainda melhores e mais baratos.

Estudamos a extensão das ideias para o problema quadrático inteiro convexo *PQIC*, onde $Q = LL^T$ e $L \in \mathbb{Q}^{n \times r}$, com $r < n$, e também para o problema quadrático não-convexo inteiro com restrições de caixa *PQINC*.

Estendendo a Proposição 3.2.1 apresentada em [1] para o caso onde Q é de posto incompleto, podemos tanto generalizar as ideias para *PQIC*, como também entender as dificuldades impostas sobre a racionalidade de L e a impossibilidade de estender a teoria desenvolvida em [1] em relação à propriedade de arredondamento forte da função quadrática subestimadora. No entanto, mostramos que no caso de L ser racional, a metodologia estendida é novamente capaz de gerar limites muito melhores do que os limites contínuos alternativos. O estudo mostra vantagens da metodologia quando Q tem um posto pequeno.

Além disso, para expandir as ideias para a programação quadrática inteira não convexa *PQINC*, primeiro relaxamos o problema original usando uma decomposição DC (diferença de funções convexas) da matriz Q bastante utilizada na literatura, e subestimamos a parte côncava resultante da função objetivo por sua função linear secante.

Observamos que esta primeira relaxação é de fato uma reformulação convexa do problema, quando todas as variáveis são binárias, portanto o problema é relaxado somente em uma segunda etapa do procedimento, onde os limites elipsoidais são aplicados para gerar limites inferiores para a relaxação quadrática convexa obtida no primeiro passo.

Mostramos como, dependendo da estratégia utilizada para decompor Q , podemos chegar ao segundo passo do procedimento com uma relaxação semelhante ao *PQIC* ou ao *PQIEC*. Gerando algumas instâncias específicas, fomos capazes de analisar a vantagem de ter uma relaxação quadrática estritamente convexa. Usando a metodologia, nós finalmente obtemos limites para problemas quadráticos binários da biblioteca *BiqMac*, e mostramos melhorias significativas nos limites contínuos para essas instâncias.

Concluimos com a nossa pesquisa que relaxações inteiras para os problemas de programação quadrática inteira podem levar a resultados que são teoricamente e numericamente competitivos. Apontamos como direções para o trabalho futuro, a busca de metodologias estendidas para o problema *PQIC* que não dependam da racionalidade do fator de matriz L , permitindo a aplicação das estratégias para instâncias mais gerais do problema.

Referências Bibliográficas

- [1] BUCHHEIM, C., CAPRARA, A., LODI, A. “An effective branch-and-bound algorithm for convex quadratic integer programming”, *Math. Program.*, v. 135, pp. 369–395, 2012.
- [2] BUCHHEIM, C., HÜBNER, R., SCHÖBEL, A. “Ellipsoid bounds for convex quadratic integer programming”, *SIAM J. Optim.*, v. 25, n. 2, pp. 741–769, 2015.
- [3] HÜBNER, R. *A level set approach to integer nonlinear optimization*. Doctoral thesis, Georg-August University School of Science, Göttingen, 2013.
- [4] HÜBNER, R., SCHÖBEL, A. “When is rounding allowed in integer nonlinear optimization?” *European Journal of Operational Research*, v. 237, pp. 404–410, 2014.
- [5] KIM, S., KOJIMA, M. “Second order cone programming relaxation of nonconvex quadratic optimization problems”, *Optim. Methods Softw.*, v. 15, pp. 201–204, 2000.
- [6] SAXENA, A., BONAMI, P., LEE, J. “Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations”, *Math. Program.*, v. 124, pp. 383–411, 2010.
- [7] SAXENA, A., BONAMI, P., LEE, J. “Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations”, *Math. Program.*, v. 130, pp. 359–413, 2011.
- [8] QUALIZZA, A., BELOTTI, P., MARGOT, F. “Linear Programming Relaxations of Quadratically Constrained Quadratic Programs”. In: Lee, J., Leyffer, S. (Eds.), *Mixed Integer Nonlinear Programming*, v. 154, *The IMA Volumes in Mathematics and its Applications*, Springer New York, pp. 407–426, 2012.
- [9] JÜNGER, M., LIEBLING, T., NADDEF, D., et al. *50 Years of integer programming 1958-2008*. Berlin, Springer, 2010.

- [10] HEMMECKE, R., KÖPPE, M., LEE, J., et al. “Nonlinear integer programming”. In: *50 Years of integer programming*, cap. 15, pp. 561–618, Springer, 2010.
- [11] CONWAY, J., J., N., SLOANE, A. *Sphere Packings, Lattices and Groups*. Third ed. New York, Springer-Verlag, 1999.
- [12] LENSTRA, A. K., LENSTRA, H. W., J., LOVÁSZ, L. “Factoring polynomials with rational coefficients”, *Mathematische Annalen*, v. 261, n. 4, pp. 515–534, 1982.
- [13] LENSTRA, A. K., LENSTRA, H. W., J., LOVÁSZ, L. “Algorithmic in number theory: lattices, number fields, curves and cryptography Math”, *Sci. Res. Inst. Publ.*, pp. 127–181, 2008.
- [14] NGUYEN, P. Q., VALLÉE, B. *The LLL Algorithm: Survey and Applications*. 1 ed. Berlin Heidelberg, Springer-Verlag, 2010.
- [15] HAMACHER, H. W., NICKEL, S. “Restricted planar location problems and applications”, *Naval Research Logistics (NRL)*, v. 42, n. 6, pp. 967–992, 1995.
- [16] BIENSTOCK, D. “Eigenvalue techniques for convex objective, nonconvex optimization problems”. In: Eisenbrand, F., Shepherd, B. (Eds.), *IPCO 2010*, pp. 29–42, 2010.
- [17] BUCHHEIM, C., SANTIS, M. D., PALAGI, L., et al. “An exact algorithm for nonconvex quadratic integer minimization using ellipsoidal relaxations”, *SIAM J. Optim.*, v. 23, n. 3, pp. 1867–1889, 2013.
- [18] GRANT, M., BOYD, S. “CVX: Matlab Software for Disciplined Convex Programming, version 2.1”. <http://cvxr.com/cvx>, mar. 2014.
- [19] ANDERSON, E., BAI, Z., BISCHOF, C., et al. *LAPACK Users’ Guide*. Third ed. Philadelphia, PA, Society for Industrial and Applied Mathematics, 1999. ISBN: 0-89871-447-8 (paperback).
- [20] BLAS. “Basic Linear Algebra Subprograms”. 2009. Disponível em: www.netlib.org/blas.
- [21] LE, T. H. A., PHAM, D. T. “The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems”, *Annals of Operations Research*, v. 133, pp. 23–46, 2005.

- [22] HORST, R., N.V., T. “DC programming: overview”, *Journal of Optimization Theory and Applications*, v. 103, pp. 1–43, 1999.
- [23] FAMPA, M., LEE, J., MELO, W. “On global optimization with indefinite quadratics”, *Isaac Newton Institute Preprint*, , n. NI13066, 2013.
- [24] WIEGELE, A. “Biq Mac Library - A collection of Max-Cut and quadratic 0-1 programming instances of medium size”. <http://biqmac.uni-klu.ac.at/biqmaclib.html>, 2007.