



LUIZ RAFAEL DOS SANTOS

ESCOLHA OTIMIZADA DE PARÂMETROS EM  
MÉTODOS DE PONTOS INTERIORES PARA  
PROGRAMAÇÃO LINEAR

CAMPINAS  
2014





UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística  
e Computação Científica

LUIZ RAFAEL DOS SANTOS

ESCOLHA OTIMIZADA DE PARÂMETROS EM  
MÉTODOS DE PONTOS INTERIORES PARA  
PROGRAMAÇÃO LINEAR

Tese apresentada ao Instituto de Matemática,  
Estatística e Computação Científica da Univer-  
sidade Estadual de Campinas como parte dos  
requisitos exigidos para a obtenção do título de  
Doutor em matemática aplicada.

**Orientador: Aurelio Ribeiro Leite de Oliveira**

**Coorientadores: Fernando Rocha Villas Bôas e Clóvis Perin Filho**

Este exemplar corresponde à versão final da Tese defen-  
dida pelo aluno Luiz Rafael dos Santos e orientada pelo  
Prof. Dr. Aurelio Ribeiro Leite de Oliveira.

**Assinatura do Orientador**



**Assinatura dos Coorientadores**





CAMPINAS  
2014

**Ficha catalográfica**  
Universidade Estadual de Campinas  
Biblioteca do Instituto de Matemática, Estatística e Computação Científica  
Maria Fabiana Bezerra Muller – CRB 8/6162

Santos, Luiz Rafael dos, 1981–  
Sa59e Escolha otimizada de parâmetros em métodos de pontos interiores para programação linear / Luiz Rafael dos Santos – Campinas, SP : [s.n.], 2014.

Orientador: Aurelio Ribeiro Leite de Oliveira.  
Coorientadores: Fernando Rocha Villas Bôas e Clóvis Perin Filho.  
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Programação linear. 2. Métodos de pontos interiores. 3. Análise de algoritmos. I. Oliveira, Aurelio Ribeiro Leite de, 1962–. II. Villas Bôas, Fernando Rocha. III. Perin Filho, Clóvis. IV. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. V. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Optimized choice of parameters in interior point methods for linear programming

**Palavras-chave em inglês:**

Linear programming

Interior-point methods

Algorithm analysis

**Área de concentração:** Matemática Aplicada

**Titulação:** Doutor em Matemática Aplicada

**Banca examinadora:**

Aurelio Ribeiro Leite de Oliveira [Orientador]

Francisco de Assis Magalhães Gomes Neto

Marta Ines Velazco Fontova

Pedro Augusto Munari Junior

Luis Felipe Cesar da Rocha Bueno

**Data de defesa:** 31-07-2014

**Programa de Pós-Graduação:** Matemática Aplicada

**Tese de Doutorado defendida em 31 de julho de 2014 e aprovada**

**Pela Banca Examinadora composta pelos Profs. Drs.**

*Aurelio R L de Oliveira*

---

**Prof(a). Dr(a). AURELIO RIBEIRO LEITE DE OLIVEIRA**

*Francisco de Assis Magalhães Gomes Neto*

---

**Prof(a). Dr(a). FRANCISCO DE ASSIS MAGALHÃES GOMES NETO**

*Marta Ines Velazco Fontova*

---

**Prof(a). Dr(a). MARTA INES VELAZCO FONTOVA**

*Pedro Augusto Munari Junior*

---

**Prof(a). Dr(a). PEDRO AUGUSTO MUNARI JUNIOR**

*Luis Felipe Cesar da Rocha Bueno*

---

**Prof(a). Dr(a). LUIS FELIPE CESAR DA ROCHA BUENO**



## Abstract

In this work we propose a predictor-corrector interior point method for linear programming in a primal-dual context, where the next iterate is chosen by the minimization of a polynomial merit function of three variables: the first one is the step length, the second one defines the central path and the last one models the weight that a corrector direction must have. The merit function minimization is performed by restricting it to constraints defined by a neighborhood of the central path that allows wide steps. In this framework, we combine different directions, such as the predictor, the corrector and the centering directions, with the aim of producing a better direction. The proposed method generalizes most of predictor-corrector interior point methods, depending on the choice of the variables described above. Convergence analysis of the method is carried out, considering an initial point that has a good practical performance, which results in Q-linear convergence of the iterates with polynomial complexity. Numerical experiments are made, using the Netlib test set, which show that this approach is competitive when compared to well established solvers, such as PCx.

**Keywords:** Linear Programming, Interior-point methods, Algorithm analysis.

## Resumo

Neste trabalho, propomos um método de pontos interiores do tipo preditor-corretor para programação linear em um contexto primal-dual, em que o próximo iterado é escolhido através de um subproblema de minimização de uma função de mérito polinomial a três variáveis: a primeira variável é o tamanho de passo, a segunda define a trajetória central e a última modela o peso que uma direção corretora deve ter. A minimização da função de mérito é feita sujeitando-a a restrições definidas por uma vizinhança da trajetória central que permite passos largos. Dessa maneira, combinamos diferentes direções, tais como preditora, corretora e de centralização, com o objetivo de obter uma direção melhor. O método proposto generaliza grande parte dos métodos de pontos interiores preditores-corretores, a depender da escolha do valor das variáveis acima descritas. É feita, então, uma análise de convergência do método proposto, considerando um ponto inicial que tem bom desempenho na prática, e que resulta em convergência linear dos iterados em complexidade polinomial. São feitos experimentos numéricos, utilizando o conjunto de testes Netlib, que mostram que essa abordagem é competitiva, quando comparada a implementações de pontos interiores bem estabelecidas como o PCx.

**Palavras-chave:** Programação linear, Métodos de pontos interiores, Análise de algoritmos.



# Sumário

|   |           |
|---|-----------|
| <b>Introdução</b>   | <b>1</b>  |
| <b>1 Programação linear</b>   | <b>5</b>  |
| 1.1 Método simplex . . . . .  | 8         |
| 1.2 Método do elipsoide . . . . .   | 9         |
| 1.3 Métodos de pontos interiores . . . . .                                  | 10        |
| 1.4 Formulação com variáveis canalizadas e extensões . . . . .              | 12        |
| <b>2 Métodos de Pontos Interiores Primais-Duais para Programação Linear</b> | <b>17</b> |
| 2.1 Um método de pontos interiores . . . . .                                | 18        |
| 2.1.1 Método afim-escala . . . . .  | 21        |
| 2.1.2 O método de barreira logarítmica . . . . .                            | 24        |
| 2.2 Métodos seguidores de caminho . . . . .                                 | 29        |
| 2.2.1 Modelo geral para métodos seguidores de caminho . . . . .             | 29        |
| 2.2.2 Vizinhanças da trajetória central . . . . .                           | 30        |
| 2.2.3 Ponto inicial ineficaz . . . . .                                      | 35        |
| 2.3 Métodos de Pontos Interiores na Prática . . . . .                       | 37        |
| 2.3.1 Método Preditor-Corretor de Mehrotra . . . . .                        | 37        |
| 2.3.2 Método das múltiplas correções de centralização . . . . .             | 43        |
| 2.3.3 Ponto Inicial . . . . .   | 45        |
| 2.3.4 Critério de Parada . . . . .  | 48        |
| <b>3 Método de Escolha Otimizada de Parâmetros</b>                          | <b>51</b> |
| 3.1 Direções de busca . . . . .   | 55        |
| 3.1.1 Direção Afim-escala . . . . .   | 55        |
| 3.1.2 A direção ideal . . . . .   | 57        |
| 3.1.3 Combinando direções . . . . .   | 59        |
| 3.2 O próximo resíduo . . . . .   | 62        |

|          |   |            |
|----------|---|------------|
| 3.3      | Um polinômio como função de mérito . . . . .                                | 67         |
| 3.4      | Vizinhança como restrições polinomiais . . . . .                            | 72         |
| <b>4</b> | <b>Convergência polinomial do Método de Escolha Otimizada de Parâmetros</b> | <b>79</b>  |
| 4.1      | Resultados técnicos . . . . .   | 82         |
| 4.2      | Prova de complexidade e convergência . . . . .                              | 96         |
| <b>5</b> | <b>Implementação</b>  | <b>109</b> |
| 5.1      | Ponto Inicial . . . . .   | 110        |
| 5.2      | Critério de Parada . . . . .  | 112        |
| 5.3      | Solução do subproblema de otimização . . . . .                              | 114        |
| <b>6</b> | <b>Resultados numéricos</b>   | <b>119</b> |
|          | <b>Considerações finais</b>   | <b>129</b> |
|          | <b>Referências</b>  | <b>133</b> |

*Para Carolina.*



# Agradecimentos

Agradeço primeiramente a Deus, o qual foi sempre suporte na jornada de muitos anos em busca da realização de meus sonhos.

Agradeço também aos meus familiares, pelo apoio e incentivo para que eu pudesse escolher uma profissão com a qual me sentisse feliz e realizado.

Aos meus três orientadores e coautores desta tese, sem os quais este trabalho não existiria. Ao Aurelio, pela calma, paciência e apoio durante todo trajeto do doutorado. Ao Fernando, pelas inúmeras vezes em que usou toda sua habilidade para me ensinar não só matemática, como tantas outras coisas não menos importantes. Ao Clóvis, pela humildade com que soube compartilhar todo seu conhecimento.

Ao professor Jacek Gondzio e à University of Edinburgh, por me receberem tão bem na Escócia.

Aos professores Chico, Marta, Felipe e Pedro, por fazerem parte da banca bem como à professora Márcia, por participar do exame de qualificação e darem todos, contribuições e dicas valorosas para a finalização deste trabalho.

Aos professores do IMECC Mário Martínez, Nino, Rodney, Joni, Márcia, Moretti, Chico e Carlile, além dos meus orientadores Aurelio, Fernando e Clóvis. Vocês foram fundamentais na minha formação como matemático, professor e pesquisador, mas sobretudo como pessoa. Tenho a alegria de considerá-los meus amigos.

Também a todos os técnicos-administrativos do IMECC, que sempre me ajudaram prontamente, muitas vezes para além de suas responsabilidades.

Não posso deixar de agradecer aos colegas e amigos que fiz nesses anos de Unicamp. Todos vocês fazem parte da minha história e não serão esquecidos.

Agradeço à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo suporte financeiro fundamental para a realização deste trabalho.

Finalmente agradeço à minha amada Carolina. Seu amor, caráter, companheirismo e incentivo, sua firmeza, amizade, lealdade e determinação me fazem uma pessoa melhor. Esta tese é dedicada a você, amor.



# Lista de Símbolos

$\mathbb{N}$  Conjunto dos números naturais.

$\mathbb{R}$  Conjunto dos números reais.

$m, n, p, q, i, j, \ell$  Variáveis ou constantes naturais.

$\mathbb{R}^n$  Espaço euclidiano de números reais de dimensão  $n$ .

$\infty$  Infinito.

**Letras romanas minúsculas** Vetores reais. Se  $v \in \mathbb{R}^n$ , então  $v_i$  é a  $i$ -ésima componente de  $v$ .

**Letras romanas maiúsculas** Matrizes reais. Se  $A \in \mathbb{R}^{m \times n}$ ,  $a_{ij}$  é a componente de  $A$  que está na  $i$ -ésima linha e na  $j$ -ésima coluna.

**Letras gregas** Escalar ou constante real.

$|\cdot|$  Valor absoluto ou módulo de um número real.

$\|\cdot\|$  Norma-2 de um vetor real.

$\|\cdot\|_2$  Norma-2 de um vetor real.

$\|\cdot\|_1$  Norma-1 de um vetor real.

$\|\cdot\|_\infty$  Norma- $\infty$  de um vetor real.

$k$  Indica índice de iteração de um vetor ou escalar. Para  $v \in \mathbb{R}^n$ ,  $v^k$  representa vetor na iteração  $k$  e para  $\alpha \in \mathbb{R}$ ,  $\alpha_k$  representa o escalar  $\alpha$  na iteração  $k$ .

$(\alpha, \beta)$  Intervalo aberto de  $\alpha$  até  $\beta$ .

$[\alpha, \beta]$  Intervalo fechado de  $\alpha$  até  $\beta$ .

$(u, v)$  Quando  $u, v \in \mathbb{R}^n$ , representa o vetor de  $\mathbb{R}^{2n}$  com as  $n$  primeiras componentes iguais às  $n$  componentes de  $u$  e as  $n$  últimas componentes iguais às  $n$  componentes de  $v$ .

$I_n$  Matriz identidade de ordem  $n$ .

$A^T$  Matriz transposta de  $A$ .

$\text{diag}(A)$  Diagonal da matriz  $A$ .

$\text{diag}(x)$  Matriz diagonal com vetor  $x$  na diagonal principal. Utilizamos  $X = \text{diag}(x)$  para indicar a matriz diagonal  $X$  cuja diagonal principal seja  $x$ .

$uv$  Produto de Hadamard [34, p. 455] entre  $u$  e  $v$  dado por  $uv = \text{diag}(u)v$ , isto é,  $uv$  é o vetor em  $\mathbb{R}^n$  tal que cada componente  $(uv)_i = u_i v_i$ , para  $i = 1, \dots, n$ .

$e$  Vetor com todas as componentes iguais a um, de dimensão apropriada.

$t$  Representa sempre uma variável local, podendo ser escalar ou vetor, real ou natural, de acordo com o contexto.

$\mathcal{O}(n)$  Se  $\alpha = \mathcal{O}(n)$  então existe constante  $\beta$  tal que  $\alpha \leq \beta n$ .

$a \leftarrow b$  Assinala para a variável  $a$  o valor da variável  $b$ .

# Introdução

## Objetivos e estrutura deste trabalho

Desde que os Métodos de Pontos Interiores (MPI) Primais-Duais ficaram famosos nos anos 90 e que o Método Preditor-Corretor de Mehrotra [49] surgiu em 1992, muito se tem feito na tentativa de estender e melhorar tal método. Algumas das principais questões atuais em MPI, incluem o modo como se pode combinar diferentes direções, seja direção corretora, seja preditora, seja ainda alguma outra direção de ordem superior, tal que uma melhor direção seja gerada em cada passo. Como veremos a seguir, uma direção pode ser entendida como a solução do sistema de Newton para as condições KKT do problema de Programação Linear (PL). Nesse sentido, as direções utilizadas a cada iteração são encontradas através da solução de um sistema linear com a mesma matriz de coeficientes – constante em cada iteração –, porém com vetor do lado direito alterado.

Gondzio [25], por exemplo, combina a ideia inicial preditora-corretora de Mehrotra, porém permitindo múltiplas correções em um mesmo iterando, tentando aumentar o tamanho do passo que pode ser dado, impondo que os iterados estejam o mais perto possível da trajetória central. Colombo e Gondzio [9] desenvolvem essa ideia, escolhendo as direções preditora e corretora de forma diferente e permitindo, conforme o caso, quantas correções se deseje.

Um outro caminho é percorrido por Jarre e Wechs [38] que resolvem, em cada iteração, um subproblema que tenta melhorar os resíduos. Tal subproblema é em si um problema de PL de

dimensão bem menor em relação ao problema original e cujas variáveis são o peso que cada componente da direção que utiliza deve ter. A solução desse subproblema é feita através do método Simplex. Para tais autores, o subproblema que resolvem pode ser entendido como uma forma de buscar o próximo iterando no subespaço gerado pelas direções que utilizam em seu método.

Nesse mesmo sentido, Mehrotra e Li [50], também obtém a direção de busca combinando direções corretoras e predictoras através de um pequeno subproblema de PL. Aqui, as múltiplas direções corretoras são encontradas fazendo uso de informações geradas a partir de um subespaço de Krylov apropriado.

Além disso, conforme Hung e Ye [35], métodos de pontos interiores que são seguidores de caminho – veja Seção 2.2 – geram uma sequência de pontos dentro de uma certa vizinhança da trajetória central, a qual previne os iterados de prematuramente chegarem muito perto da fronteira da região de factibilidade. Para tanto, é necessário impor condições pré-definidas que façam com que o próximo ponto esteja dentro de uma dessas vizinhanças, garantido o bom desempenho do método.

Nesta tese, mostraremos algumas alternativas de respostas para essas e outras questões correlatas. Assim o objetivo principal desse trabalho é *propor e implementar* um Método de Pontos Interiores com pontos inactíves para PL, do tipo seguidor de caminho, fazendo uso de polinômios reais nas variáveis  $(\alpha, \mu, \sigma)$ , em que  $\alpha$  é o tamanho do passo,  $\mu$  é o parâmetro que define a trajetória central, e  $\sigma$  modela o peso que uma direção corretora deve ter; trata-se portanto de um método predictor-corretor.

Neste sentido, esses parâmetros são vistos como variáveis, e sua escolha é feita de forma adiada, através da solução de um subproblema que minimiza uma função de mérito *predictiva* que é um polinômio nessas três variáveis e que está sujeita a restrições que impõem que o próximo iterando esteja dentro de uma vizinhança da trajetória central. A função de mérito é predictiva, no sentido que prediz os resíduos lineares e complementares do próximo iterado. Esse é o que

chamamos de Método de Escolha Otimizada de Parâmetros (MEOP).

Uma abordagem similar foi proposta primeiramente por Villas-Bôas e Perin [73], em um contexto auto-dual. Já num contexto primal-dual Villas-Bôas et al. [71, 72], também fazem uso de uma função de mérito polinomial, porém definem uma trajetória central mais geral, que envolve também os resíduos lineares na sua definição. Em particular, a abordagem desses autores será utilizada para resolver o subproblema que surgir em nosso método.

Além disso, temos o objetivo de *demonstrar* resultados de convergência do MEOP, utilizando as ferramentas de análise numérica para Métodos Preditores-Corretores do tipo Mehrotra ineficazes, apresentadas principalmente por Zhang [82] e trabalhos que são sequências desse e das ideias resumidas em Wright [77, cap. 7]. Dentre essas ferramentas, há a necessidade de escolher um ponto inicial adequado para que a convergência do método seja garantida.

Em geral, conforme alerta Wright [77, p. 112], vários autores utilizam pontos iniciais que dependem da norma de uma solução ótima, valor que é desconhecido o que torna a implementação do algoritmo impossível, pelo menos da forma como o algoritmo é descrito. Para contornar tal problema, estabeleceremos uma Condição que o ponto inicial deve satisfazer a fim de que seja possível demonstrar a convergência e a complexidade do algoritmo. Tal condição deve levar em conta o tamanho dos dados do problema a ser resolvido e a distância entre o ponto inicial e uma solução ótima. O ponto inicial será escolhido conforme a heurística de Mehrotra [49] e mostraremos que esse ponto satisfaz a Condição proposta, pelo menos para todo conjunto de testes que utilizamos.

Em relação a experimentos computacionais, o objetivo é implementar o método proposto e testá-lo. Para os testes, utilizaremos o conjunto de testes NETLIB<sup>1</sup> [19, 21]. Também objetivamos comparar nosso método com o PCx [11], uma implementação do método preditor-corretor de Mehrotra.

Com o fim de alcançar esses objetivos, esta monografia foi organizada com a seguinte estrutura:

---

<sup>1</sup>Disponível em [www.netlib.org/lp/data](http://www.netlib.org/lp/data).

- nos Capítulos 1 e 2 apresentamos o problema de programação linear, bem como o estado da arte no que diz respeito aos métodos de pontos interiores;
- no Capítulo 3 apresentamos o desenvolvimento de um polinômio real de três variáveis e de grau total 6, como função de mérito que sirva não só como medida de factibilidade e otimalidade de um ponto, mas que permita escolher, de forma otimizada, os parâmetros que determinam uma melhor direção em cada iteração de um método de pontos interiores primal-dual do tipo seguidor de caminhos;
- no Capítulo 4 fizemos a análise de convergência e de complexidade do método proposto;
- no Capítulo 5 relatamos alguns detalhes de nossa implementação, no que diz respeito ao ponto inicial, ao critério de parada que utilizamos em nosso algoritmo. Além disso, fazemos uma breve explanação de como se resolve o subproblema de otimização que surge em cada iteração, com base em [71, 72, 73];
- No Capítulo 6 descrevemos o conjunto de testes utilizados e relatamos os resultados numéricos.

De modo a concluir esta tese, no último capítulo fizemos considerações finais e encaminhamentos para trabalhos futuros.

# Capítulo 1

## Programação linear

“Linear programming can be viewed as part of a great revolutionary development which has given mankind the ability to state general goals and to lay out a path of detailed decisions to take in order to ‘best’ achieve its goals when faced with practical situations of great complexity.” [15]

---

*(George B. Dantzig)*

Neste capítulo será feita uma breve introdução aos problemas de PL, bem como a aspectos preliminares de alguns métodos desenvolvidos para resolvê-los.

Programação Linear é um tópico que tem sido amplamente estudado no escopo da otimização e que ganhou relevância a partir da década de 1940 com os trabalhos de Dantzig, Kantorovich, Koopmans e von Neumann. A criação do método simplex por Dantzig acelerou o interesse pelo tema, já que se encontrou um meio de resolver tal problema. O texto de Dantzig que serve de epígrafe a este capítulo serve ao mesmo tempo como um resumo histórico do desenvolvimento da PL e a define de maneira genérica.

A partir da necessidade de tratar matematicamente problemas econômicos e de planejamento

militar nos anos que se seguiram à Segunda Guerra Mundial, ao mesmo tempo em que os computadores passaram a poder realizar operações matemáticas com maior velocidade, houve um crescimento espantoso da utilização da PL e, por extensão, da otimização em geral. Para um resumo histórico há boas fontes [14, 64].

Um problema de otimização pode ser descrito em termos de variáveis de decisão, conjuntos de restrições e uma função objetivo. Resolver um problema de otimização significa encontrar a “melhor maneira” na qual as variáveis satisfaçam as restrições ao mesmo tempo em que se otimiza – minimiza ou maximiza – a função objetivo.

Um problema de *Programação Linear* é um problema de otimização no qual as restrições e a função objetivo são lineares. A PL<sup>1</sup> surge diretamente em várias aplicações reais, por exemplo, em problemas de economia, logística, planejamento e controle da produção, entre outros. Ela aparece ainda em aproximações de problemas mais complexos ou na solução relaxada de problemas de programação inteira ou mista.

Matematicamente, podemos escrever qualquer PL na seguinte *forma padrão*:

$$\begin{array}{ll} \min_x & c^T x \\ \text{sujeito a} & \begin{cases} Ax = b \\ x \geq 0 \end{cases}, \end{array} \quad (1.1)$$

em que  $A \in \mathbb{R}^{m \times n}$  e  $c$ ,  $x$  e  $b$  têm dimensões compatíveis. A região  $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b \text{ e } x \geq 0\}$  é conhecida como região factível do problema e é um poliedro com pelo menos um vértice ou ponto extremo.

Como uma função linear é convexa, problemas de PL têm a particularidade dada pelo seguinte teorema [6, cap. 3]:

**Teorema 1.1** (Teorema Fundamental da Programação Linear). *Em um problema de Programação*

---

<sup>1</sup>Usaremos os termos “problema de PL” e “PL” como sinônimos.

Linear com região factível  $\mathcal{P}$ , ou o valor ótimo da função objetivo é ilimitado ou então este valor será atingido em um ponto extremo de  $\mathcal{P}$ .

Um conjunto de restrições lineares define um *poliedro* que constitui então a *região factível*. De acordo com o Teorema 1.1, uma forma de encontrar a solução de um problema de PL seria percorrer todos os vértices da região factível comparando os valores da função objetivo e selecionando o melhor dentre eles.

Essa estratégia, no entanto, é pouco eficiente, haja vista o poliedro ser resultante de um sistema linear de  $m$  restrições e  $n$  variáveis ( $m < n$ ) podendo ter um número de vértices totalizando

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} \geq \left(\frac{n}{m}\right)^m,$$

quando o problema está na forma padrão. Por conta disso, um método do tipo exaustão seria pouco eficiente. Por outro lado, o fato de o número de vértices ser limitado garante terminação finita para qualquer método que funcione desta forma. Entretanto, esse número é exponencial, já que

$$\left(\frac{n}{m}\right)^m \geq 2^m \text{ para } n \geq m.$$

Encontrar um *vértice ótimo* de maneira eficiente é o segredo dos bons métodos para resolver problemas de PL.

Associado a todo problema PL, escrito na forma padrão, existe um outro PL, chamado *problema dual* dado por

$$\begin{array}{ll} \max & b^T y \\ \text{sujeito a} & \left\{ \begin{array}{l} A^T y \leq c \\ y \text{ livre} \end{array} \right. \end{array}$$

em que  $A$ ,  $b$  e  $c$  são os mesmos do problema original, que, neste caso, é chamado *problema primal*. A variável  $y \in \mathbb{R}^m$  é chamada variável dual. A fim de que tenhamos restrições de igualdade na

definição do problema dual, introduzimos variáveis de folga duais representadas por  $z \in \mathbb{R}^n$ , tais que  $z \geq 0$ , obtendo

$$\begin{array}{ll} \max & b^T y \\ \text{sujeito a} & \begin{cases} A^T y + z = c \\ z \geq 0, \quad y \text{ livre} \end{cases} \end{array} \quad (1.2)$$

Os problemas primal e dual são comumente chamados par primal-dual e as relações existentes entre estes permitiram a criação ou refinamento de diversos métodos para a solução de PL [6]. No Capítulo 2 faremos uma explanação mais detalhada sobre esses dois problemas, já que estudaremos métodos primais-duais.

## 1.1 Método simplex

O Método Simplex foi apresentado por Dantzig [13] em 1947 e foi desenvolvido ao mesmo tempo em que houve a percepção do poder da PL como ferramenta auxiliar na tomada de decisões. Seu pioneirismo deve-se à busca da solução do PL caminhando pelos vértices de maneira a utilizar o valor da função objetivo para determinar qual o próximo vértice adjacente deve ser visitado, fazendo assim uma aplicação do Teorema 1.1.

Mais que isso, dado um método de escolha do próximo vértice, o conjunto de vértices possíveis decresce em cada iteração, ainda no caso não degenerado. Degenerescência (primal) ocorre quando um vértice em  $\mathbb{R}^n$  é definido por  $p > n$  restrições, quando temos restrições na forma  $Ax \leq b$ , e um passo de tamanho zero pode ser produzido pelo método. Desta maneira, o método simplex não sairia do lugar, logo nenhuma melhora na função objetivo seria alcançada.

O método simplex se mostrou, e ainda se mostra, robusto e eficiente na solução de grande quantidade de problemas, e por isso mesmo foi considerado *o método preferencial* para se resolver PL. Entretanto, dado o número exponencial de vértices possíveis, havia sempre o receio de que seria necessário esforço exponencial, isto é, que todos os vértices da região viável fossem

visitados para que se encontrasse a solução ótima. De fato, Klee e Minty [43] foram os primeiros a apresentar uma classe de exemplos patológicos para a qual esse comportamento do método simplex apareceu. No exemplo criado por estes autores, com  $n$  variáveis e  $2n$  restrições, o método simplex de Dantzig precisou passar por todos os  $2^n - 1$  pontos extremos da região viável antes de encontrar a solução ótima.

Apesar disso, nenhum caso com número exponencial de iterações foi encontrado na vida real, e geralmente apenas uma pequena porcentagem de vértices é consultada até que se chegue a uma solução ótima. Mais que isso, em geral, o método simplex mostra comportamento polinomial, sendo linear em relação a  $m$  e sublinear em relação a  $n$  [20, pg. 94]. Um resumo sobre a eficiência do método simplex na prática pode ser encontrada no artigo de Shamir [65].

## 1.2 Método do elipsoide

Enquanto na prática o método Simplex continuava como o método mais robusto e eficiente, a busca por um método que tivesse *complexidade polinomial* no pior caso foi satisfeita por Khachiyan [41] através de seu *Método do Elipsoide*. Antes de continuarmos, vale a pena fazermos uma observação sobre o sentido que daremos a complexidade.

*Observação 1.2.* Neste trabalho, quando mencionarmos *complexidade*, o faremos no sentido de um limitante para o número de iterações apenas, descartando aqui o esforço gasto para calcular cada iteração. Para isso, utilizaremos a seguinte notação: se  $\alpha$  tem ordem de complexidade  $n$ , indicaremos  $\alpha = \mathcal{O}(n)$  significando que existe constante  $\beta$  tal que  $\alpha \leq \beta n$ .

O método elipsoide, ao contrário do método simplex, não se baseia em caminhar pelos vértices da região factível, mas em diminuir o volume de um elipsoide que contenha um potencial ponto de solução através de um conjunto de desigualdades lineares estritas.

Inquestionavelmente, o método elipsoide representou um grande avanço teórico, significando que a Programação Linear faz parte dos problemas polinomiais, isto é, que pode ser resolvido

por um algoritmo de complexidade polinomial, já que se provou que este método tem  $\mathcal{O}(n^2(1/\varepsilon))$  iterações, em que  $\varepsilon$  é a precisão desejada para o algoritmo.

Todavia, do ponto de vista prático, o método elipsoide não conseguiu competir com o método simplex [7], pois sua convergência era muito lenta, na presença de erros de arredondamento perdia robustez e em cada iteração a quantidade de memória necessária para armazenamento era muito grande, tornando a iteração cara. Consequentemente, embora desafortunadamente o método simplex tenha complexidade exponencial na análise do *pior caso*, experimentos numéricos indicaram que na prática este era absolutamente superior ao método elipsoide.

## 1.3 Métodos de pontos interiores

Em problemas práticos, o método Simplex reinou absoluto na solução de problemas de PL até meados da década de 1980 como único método viável para resolver tal classe de problemas. Em 1984, Karmarkar [39] inicia uma nova abordagem que ficou conhecida como Métodos de Pontos Interiores (MPI) . Por outro lado, em 1967, Dikin [18] já havia publicado um trabalho, no qual um método de pontos interiores é apresentado. O método de Dikin ficou conhecido como método afim-escala.

A ideia principal dos MPI difere-se fundamentalmente da que inspira o método simplex. No método simplex as soluções encontradas em cada iteração estão na fronteira da região factível, já que o método visita os vértices do poliedro que define tal região. Por outro lado, nos MPI estas soluções em cada iteração estão no interior de uma região.

Isto é feito criando-se uma família parametrizada de soluções que convergem assintoticamente para uma solução exata, isto é, temos um método homotópico. Com efeito, um MPI basicamente utiliza um método do tipo Newton nas condições de otimalidade de Primeira Ordem, ou Condições KKT, do Problema de Programação Linear. Há, nos diferentes métodos, variação do lado direito que se usa na solução da linearização das condições KKT que o Método de

Newton utiliza, podendo gerar, nesse caso, direções de Newton diferentes. Daí o nome *Métodos de Pontos Interiores*.

Em resumo, uma abordagem não-linear é utilizada para resolver um problema linear, escapando então da dificuldade que a dimensão do problema apresenta ao lidar com as características combinatoriais do PL.

Além disso, como exposto acima, no método simplex a quantidade de iterações pode crescer exponencialmente, de acordo com o tamanho do problema, o que não acontece MPI. Do ponto de vista da complexidade computacional, esse método também é polinomial, com  $\mathcal{O}(n(1/\varepsilon))$  iterações [39].

Ao contrário do método elipsoide, o método de Karmarkar tem um desempenho muito melhor e compete com o simplex, sendo consideravelmente melhor que este em problemas de grande porte. Uma variante do método de Karmarkar foi implementada por Adler et al. [1] e, desde então, a compreensão do funcionamento desses métodos aumentou consideravelmente, ao mesmo tempo em que variantes têm sido propostas, muitas delas se apresentando como alternativas computacionais viáveis ao método simplex.

Atualmente, o melhor algoritmo de pontos interiores conhecido para PL, do ponto de vista teórico, encontra uma solução com  $\varepsilon$ -precisão em  $\mathcal{O}(\sqrt{n} \ln(1/\varepsilon))$  iterações [61]. De acordo com a teoria geral [57, Capítulo 4], o termo  $\sqrt{n}$  é o melhor que se pode esperar para um MPI usando uma função barreira como a logarítmica. Na prática, no entanto, as implementações de MPI têm desempenho muito melhor que isso, e, em algumas dessas, o número de iterações é quase constante, independente da dimensão do problema [9].

Em linhas gerais, há classes de problemas que são melhor resolvidos pelo método simplex e outras para os quais os métodos de pontos interiores são mais adequados. Tamanho e estrutura de esparsidade, entre outros, são fatores preponderantes para a escolha do método apropriado. Entretanto, pode-se dizer que com o aumento da dimensão do problema, os MPI ficam mais atraentes e efetivos. Em alguns problemas altamente esparsos e com estrutura particular, é

possível uma revisão do método simplex que o torna muito eficaz [32]. Para problemas de fluxo em rede, nos quais uma especialização do método simplex permite explorar a estrutura do problema de maneira muito eficiente [55], também é comprovada a eficácia do simplex. Vale a pena notar no entanto que, para um problema de fluxo de rede com cerca de milhões de variáveis, Resende e Veiga [62] mostraram que um MPI foi mais eficiente.

## 1.4 Formulação com variáveis canalizadas e extensões

Embora a formulação padrão do problema primal de PL dada em (1.1) considere apenas que devemos ter  $x$  não-negativo, em problemas práticos, nem sempre isso acontece.

Podemos ter, por exemplo, alguma componente de  $x$  – ou todo o vetor – irrestrita ou livre. Para resolver esse problema, considere que  $x_i$  seja livre, para algum  $i$ . As variáveis  $x_i^+, x_i^- \geq 0$  são criadas, tais que  $x_i = x_i^+ - x_i^-$ . Nesse caso, uma variável livre é trocada por duas variáveis não-negativas nas restrições do problema.

Pode haver também algum  $x_i$  com limitante inferior não nulo, digamos,  $\ell \in \mathbb{R}$ . Isto significa que se está exigindo agora que  $x_i \geq \ell$ . Uma forma de contornar esse problema é criar a variável  $\tilde{x}_i = x_i - \ell$  e substituir  $x_i$  por  $\tilde{x}_i + \ell$  nas restrições do problema. Temos, nesse caso,  $\tilde{x}_i \geq 0$ .

Um caso não menos comum e bastante importante é o da variável  $x$  ser canalizada, isto é, possuir algum limitante superior. Queremos dizer com isso que para algum  $i$ , existe  $u_i > 0$  tal que  $0 \leq x_i \leq u_i$ . Para esse caso, uma saída possível é a seguinte.

Suponha que existam  $p \leq n$  componentes de  $x$  que sejam canalizadas. Definimos então a matriz de posto completo  $E \in \mathbb{R}^{p \times n}$ , em que cada coluna e cada linha tem exatamente um elemento não nulo e igual a 1. Aqui,  $E$  representa a transformação linear  $E : \mathbb{R}^n \rightarrow \mathbb{R}^p$  que mapeia  $x$  em sua parte canalizada  $Ex$ . Neste caso teremos  $Ex \leq u$ , em que  $u \in \mathbb{R}^p$  é o vetor das canalizações do problema.

Com isso, o PL é escrito como

$$\begin{array}{ll} \min_x & c^T x \\ \text{sujeito a} & \begin{cases} Ax = b \\ Ex \leq u \\ x \geq 0 \end{cases} \end{array} .$$

Vamos fazer com que a desigualdade desapareça das restrições do problema acima, definindo variáveis de folga primais, dadas pelo vetor  $s \in \mathbb{R}^p$ , tais que  $s \geq 0$  e  $Ex + s = u$ . Assim, obtemos o *problema primal canalizado*

$$\begin{array}{ll} \min_x & c^T x \\ \text{sujeito a} & \begin{cases} Ax = b \\ Ex + s = u \\ (x, s) \geq 0 \end{cases} \end{array} . \tag{1.3}$$

A esse problema, associamos o *problema dual canalizado*

$$\begin{array}{ll} \max_{(y,v)} & b^T y + u^T v \\ \text{sujeito a} & \begin{cases} A^T y + z + E^T v = c \\ v + w = 0 \\ (w, z) \geq 0, \quad y, v \text{ livres} \end{cases} \end{array} . \tag{1.4}$$

É possível escrever tais problemas em uma forma matricial como

$$\begin{array}{l} \min_x \quad c^T x \\ \text{sujeito a} \quad \begin{cases} \begin{bmatrix} A & 0 \\ E & I_p \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} b \\ u \end{bmatrix} \\ (x, s) \geq 0 \end{cases} \end{array} \quad \text{e} \quad \begin{array}{l} \max_{y,v} \quad b^T y + u^T v \\ \text{sujeito a} \quad \begin{cases} \begin{bmatrix} A^T & E^T \\ 0 & I_p \end{bmatrix} \begin{bmatrix} y \\ v \end{bmatrix} + \begin{bmatrix} z \\ w \end{bmatrix} = \begin{bmatrix} c \\ 0 \end{bmatrix} \\ (w, z) \geq 0, \quad y, v \text{ livre} \end{cases} \end{array} .$$

Sejam  $\tilde{A} = \begin{bmatrix} A & 0 \\ E & I_p \end{bmatrix}$ ,  $\tilde{b} = \begin{bmatrix} b \\ u \end{bmatrix}$ ,  $\tilde{c} = \begin{bmatrix} c \\ 0 \end{bmatrix}$ ,  $\tilde{x} = \begin{bmatrix} x \\ s \end{bmatrix}$ ,  $\tilde{y} = \begin{bmatrix} y \\ v \end{bmatrix}$  e  $\tilde{z} = \begin{bmatrix} z \\ w \end{bmatrix}$ . Essas definições transformam (1.3) e (1.4) em

$$\begin{array}{l} \min_x \quad \tilde{c}^T \tilde{x} \\ \text{sujeito a} \quad \begin{cases} \tilde{A}\tilde{x} = \tilde{b} \\ \tilde{x} \geq 0 \end{cases} \end{array} \quad \text{e} \quad \begin{array}{l} \max \quad \tilde{b}^T \tilde{y} \\ \text{sujeito a} \quad \begin{cases} \tilde{A}^T \tilde{y} + \tilde{z} = \tilde{c} \\ \tilde{z} \geq 0 \quad \tilde{y} \text{ livre} \end{cases} \end{array} . \quad (1.5)$$

Observe que essa notação transforma o par primal-dual canalizado na forma do par primal-dual padrão. Assim, temos uma vantagem que utilizaremos em todo este trabalho: agora é possível tratar apenas do par primal dual na forma padrão sem canalizações, e a notação acima faz com que todos os resultados obtidos também serão estendidos para os problemas canalizados. Do ponto de vista dos experimentos numéricos, essa abordagem também não causará problemas já que, em nossa implementação, a estrutura de dados é construída pra lidar com as canalizações sem que a ordem de complexidade das operações em álgebra linear seja afetada.

Com isso, a partir de agora, consideraremos apenas PL na forma dada em (1.1) e (1.2) e faremos comentários relativos a canalizações quando oportunos, em particular quando tratarmos da implementação.

Conforme dissemos, a presente tese trata de MPI para problemas de PL. No entretanto, tais

métodos também podem ser estendidos com o objetivo de resolver problemas não-lineares de certos tipos, em geral quando a função objetivo é não linear, porém convexa, mas as restrições são lineares e há a exigência da não negatividade das variáveis.

Em particular, problemas de Programação Quadrática (QP) convexa são usualmente resolvidos com MPI, principalmente quando são de grande porte e é possível especializar o método [26]. Com efeito, um QP convexo pode ser escrito como

$$\begin{array}{ll} \min_x & x^T Q x + c^T x \\ \text{sujeito a} & \begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{array} \quad ,$$

em que  $Q \in \mathbb{R}^{n \times n}$  é simétrica semi-definida positiva,  $A \in \mathbb{R}^{m \times n}$  e  $c$ ,  $x$  e  $b$  têm dimensão compatível. Com exceção da matriz  $Q$ , temos a mesma estrutura de um PL.

Neste capítulo, fizemos uma breve introdução à Programação Linear (PL). No próximo capítulo mencionaremos resultados importantes de PL bem como faremos uma revisão dos Métodos de Pontos Interiores para PL.



## Capítulo 2

### Métodos de Pontos Interiores

### Primais-Duais para Programação Linear

“In the simplex method, the current solution is modified by introducing a nonzero coefficient for one of the columns in the constraint matrix. Our method allows the current solution to be modified by introducing several columns at once” [39]

---

*(Narendra Karmarkar)*

Este capítulo tem o objetivo de apresentar os Métodos de Pontos Interiores primais-duais seguidores de caminho. A teoria central que versa sobre MPI está bem estabelecida e há ótimos textos sobre o assunto [6, 20, 67, 77]. Recentemente, Gondzio [27] fez um resumo histórico, no qual resgata as principais contribuições no tema. Estas referências servem como texto introdutório e foi com base nelas e nos trabalhos de Colombo [8] e Villas-Bôas [70], que este capítulo foi escrito.

## 2.1 Um método de pontos interiores

O problema de PL na forma padrão,

$$\begin{array}{ll} \min_x & c^T x \\ \text{sujeito a} & \begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{array} \quad (2.1)$$

é chamado de *problema primal*.

Associado a todo PL existe um outro PL chamado *problema dual*, o qual consiste nos mesmos dados arranjados de maneira diferente. O dual de (2.1) é

$$\begin{array}{ll} \max_{(y,z)} & b^T y \\ \text{sujeito a} & \begin{cases} A^T y + z = c \\ z \geq 0, y \text{ livre} \end{cases} \end{array} \quad (2.2)$$

em que  $A \in \mathbb{R}^{m \times n}$  é uma matriz de posto completo,  $c, x, z \in \mathbb{R}^n$ ,  $y, b \in \mathbb{R}^m$  e  $m < n$ . Chamamos o vetor  $y$  de *variáveis duais*, enquanto que  $z$  é o vetor das *folgas duais*. Note que, como  $A$  tem posto completo, existe uma relação unívoca entre  $z$  e  $y$ .

Sejam

$$\mathcal{P} = \{x : Ax = b, x \geq 0\} \quad \text{e} \quad \mathcal{D} = \{(y, z) : A^T y + z = c, z \geq 0\} \quad (2.3)$$

os conjuntos de pontos factíveis primais e duais respectivamente. Usando essa notação, o par primal-dual de (2.1-2.2) pode ser reescrito como

$$\min_x c^T x \text{ s. a. } x \in \mathcal{P} \quad \text{e} \quad \max_{(y,z)} b^T y \text{ s. a. } (y, z) \in \mathcal{D}. \quad (2.4)$$

Denominamos  $\mathcal{F} = \mathcal{P} \times \mathcal{D}$  o conjuntos dos pontos primais-duais factíveis. Para fins de notação,

podemos dizer que se  $w = (x, y, z)$  é primal-dual factível, então  $w \in \mathcal{F}$ . Definimos também os conjuntos

$$\mathcal{Q} = \{w \in \mathbb{R}^{2n+m} : (x, z) \geq 0\}$$

e

$$\mathcal{Q}^+ = \{w \in \mathbb{R}^{2n+m} : (x, z) > 0\}$$

respectivamente como os conjuntos dos pontos de  $\mathbb{R}^{2n+m}$  não-negativos e positivos nas variáveis  $x$  e  $z$ .

Com isso,

$$\mathcal{F}^+ = \mathcal{F} \cap \mathcal{Q}^+ \tag{2.5}$$

é o conjunto dos pontos primais-duais interiores. Como o próprio nome sugere, em Métodos de Pontos Interiores, dedica-se especialmente a estudar iterados que estejam contidos em  $\mathcal{F}^+$ , no caso de métodos factíveis, ou iniciando com pontos em  $\mathcal{Q}^+$ , no caso de métodos infactíveis. Ainda utilizamos as notações  $\mathcal{P}^+$  e  $\mathcal{D}^+$  para indicar os conjuntos primal e dual estritamente factíveis, respectivamente.

Neste trabalho, o seguinte pressuposto é assumido como verdadeiro:

**Pressuposto 2.1.** *O conjunto  $\mathcal{F}^+$  é não-vazio, isto é, existe  $(x, y, z) \in \mathcal{F}$ , tal que  $(x, z) \in \mathcal{Q}^+$ .*

Com efeito, o Pressuposto 2.1 tem como consequência o fato de existir solução para (2.1) e para (2.2) [30, Teorema 3.1]. Mais que isso, a partir desse pressuposto, conhecido como *pressuposto do ponto interior*, concluímos que o conjunto ótimo primal-dual é limitado [31, Lema 2.2]. Caso esse pressuposto não seja satisfeito, considera-se permitir que o método aceite iterados infactíveis ou introduz-se perturbações que aumentem o conjunto  $\mathcal{F}$ .

Apresentamos agora alguns resultados bem conhecidos sobre a relação entre o par primal-dual.

**Lema 2.2** (Dualidade Fraca). *Seja  $(x, y, z) \in \mathcal{F}$ . Então  $c^T x \geq b^T y$ .*

*Demonstração.* Como  $x \in \mathcal{P}$  e  $(y, z) \in \mathcal{D}$  então vale

$$c^T x - b^T y = c^T x - x^T A^T y = x^T (c - A^T y) = x^T z \geq 0. \quad \blacksquare$$

O significado da Dualidade Fraca é que o valor objetivo primal serve de limitante superior para o valor objetivo dual e vice-versa. A diferença  $c^T x - b^T y$  é chamada *gap de dualidade*. Quando, em um ponto  $(x, y, z)$  factível, o gap de dualidade é zero, isto é, quando tanto o valor objetivo primal quanto dual alcançam seus limites, então este ponto é solução ótima primal-dual. Este resultado é formalizado no lema a seguir.

**Lema 2.3** (Dualidade Forte). *Um ponto  $x \in \mathcal{P}$  é ótimo se e somente se existe um par  $(y, z) \in \mathcal{D}$  tal que  $c^T x = b^T y$ .*

Uma condição necessária e suficiente para que o problema (2.1) tenha uma solução factível é que  $\mathcal{P} \neq \emptyset$ . Além disso, se  $\mathcal{P} \neq \emptyset$  e  $\mathcal{D} \neq \emptyset$  então ambos (2.1) e (2.2) admitem solução ótima  $(x^*, y^*, z^*)$  e pelo Lema 2.3 o valor da função objetivo coincide neste ponto. Por outro lado, se um dos conjuntos  $\mathcal{P}$  ou  $\mathcal{D}$  for vazio, então o outro conjunto será ilimitado ou vazio. Nestes casos, os problemas (2.1-2.2) não terão solução.

Podemos agora expressar condições de otimalidade para (2.1) e (2.2). Tais condições ajudam a reconhecer quando estes problemas possuem solução ótima e, por isso, podem nos ajudar a desenvolver algoritmos ou métodos para encontrar tais soluções. As condições de Karush-Kuhn-Tucker (KKT) expressam uma condição de otimalidade de primeira ordem para um PL e podem ser escritas como

$$\left\{ \begin{array}{l} Ax = b, \\ A^T y + z = c, \\ XZe = 0, \\ (x, z) \geq 0. \end{array} \right. \quad \begin{array}{l} (2.6a) \\ (2.6b) \\ (2.6c) \\ (2.6d) \end{array}$$

em que  $X = \text{diag}(x)$ ,  $Z = \text{diag}(z)$  e  $e = (1, \dots, 1)^T$ .

As duas primeiras equações de (2.6) são conhecidas como *factibilidade ou viabilidade primal e dual* – com exceção feita à não-negatividade –, respectivamente. A equação (2.6c) significa que  $x_i z_i = 0$ , para todo  $i = 1, 2, \dots, n$ , e é chamada de *condição de complementaridade*. A última equação é chamada de *não-negatividade*. Note ainda que, se  $(x, y, z) \in \mathcal{F}$ , então ele satisfaz (2.6a), (2.6b) e (2.6d). Isso significa que uma solução ótima é caracterizada pela factibilidade primal e dual e pela complementaridade. Para pontos não ótimos, mas factíveis, a complementaridade pode nos dar uma medida da *distância* destes pontos para a otimalidade:

$$x^T z = c^T x - b^T y. \quad (2.7)$$

O valor  $x^T z$  é chamado *gap de complementaridade*. Quando este converge para zero, então temos uma solução ótima. Além disso, a igualdade entre o gap de complementaridade e o gap de dualidade mostrado em (2.7) só vale quando o ponto é factível.

As equações (2.6) dão uma condição necessária e suficiente para que  $w^* = (x^*, y^*, z^*)$  seja solução de (2.1-2.2), isto é,  $x^*$  seja solução de (2.1) e  $(y^*, z^*)$  seja solução de (2.2). Consequentemente, um corolário de KKT para PL é o Lema 2.3.

### 2.1.1 Método afim-escala

Os métodos de pontos interiores primais-duais para PL encontram  $w^*$  resolvendo as equações (2.6). Note que a única equação não-linear de (2.6) é  $XZe = 0$  e por conta dela é que alguma variante do método de Newton é aplicada, modificando a direção de busca e o tamanho de passo de modo que (2.6d) seja satisfeita *estritamente* em toda iteração.

Podemos reescrever (2.6) utilizando uma aplicação  $F : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$  da seguinte maneira<sup>1</sup>:

---

<sup>1</sup>A partir de agora, utilizaremos a notação  $uv$  para representar o produto componente a componente de dois vetores  $u$  e  $v$  quaisquer, da mesma forma que fizeram Kitahara e Tsuchiya [42], Potra [60] e Salahi, Peng e Terlaky [63], entre outros. Este é o produto de Haddamard e significa que  $UVe = uv$ .

$$F(w) = \begin{bmatrix} Ax - b \\ A^T y + z - c \\ xz \end{bmatrix} = 0, \quad (2.8a)$$

$$(x, z) \geq 0. \quad (2.8b)$$

Todos os Métodos de Pontos Interiores primais-duais geram iterados  $w^k = (x^k, y^k, z^k)$  que satisfazem (2.8b) estritamente, isto é,  $(x^k, z^k) > 0$ . Esta propriedade dá origem ao termo *ponto interior*.

Ao respeitar estes limites, não são aceitos iterados tais que  $F(w^k) = 0$ . Alguns métodos primais-duais exigem que  $w^k \in \mathcal{F}^+$ , ou seja, que o ponto seja estritamente factível. No entanto, atualmente os métodos mais competitivos trabalham com pontos infactíveis, exigindo somente que o ponto seja estritamente interior (veja Seção 2.2.3)

O método de Newton aplicado a (2.8a) dá uma aproximação linear em torno do ponto atual, obtendo uma direção de busca  $\Delta w = (\Delta x, \Delta y, \Delta z)$  resolvendo o seguinte sistema linear

$$\nabla F(w) \Delta w = -F(w),$$

em que

$$\nabla F = \begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix}$$

é a matriz Jacobiana de  $F$ .

No caso de problemas de QP como dado em (1.4), a diferença estaria justamente na matriz

Jacobiana, que seria dada por

$$\nabla F = \begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I \\ Z & 0 & X \end{bmatrix}.$$

A direção de Newton é encontrado ao resolver

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - z \\ -xz \end{bmatrix} = \begin{bmatrix} r_P \\ r_D \\ r_C \end{bmatrix}. \quad (2.9)$$

Se o ponto atual é estritamente factível, então os resíduos  $r_P$  e  $r_D$  são nulos.

Um passo na direção  $\Delta w = (\Delta x, \Delta y, \Delta z)$  é chamado *afim-escala*. Normalmente não é possível caminhar nesta direção com passo completo, isto é,  $w + \Delta w$  geralmente viola o limite (2.8b). Para evitar essa dificuldade, pode ser feita uma busca linear na direção de Newton tal que o novo iterado será

$$w + \alpha \Delta w,$$

em que  $\alpha \in (0, 1]$  é o parâmetro da busca linear. Mesmo assim, geralmente podemos apenas realizar passos pequenos nesta direção ( $\alpha \ll 1$ ) antes de violar a condição  $(x, z) > 0$ . Consequentemente, passos puramente afim-escala não nos permitem um bom progresso em direção à solução.

De modo a poder contornar os problemas do método afim-escala, os métodos seguidores de caminho sugerem algumas estratégias diferentes do procedimento padrão de Newton:

- (i) Desviar a direção de busca de modo a fazê-la apontar mais para o interior do ortante não-negativo  $(x, z) \geq 0$ , de tal forma que seja possível os iterados moverem-se suficientemente longe da fronteira de tal ortante, antes que algum componente de  $(x, z)$  se torne negativo.
- (ii) Manter os componentes de  $(x, z)$  longe do limite do ortante não negativo. Direções de

busca calculadas por meio de pontos próximos a este limite tendem a ser calculadas com erros de arredondamento relativamente grandes e pouco progresso pode ser feito por elas.

### 2.1.2 O método de barreira logarítmica

Uma maneira de fazer as duas proposições acima valerem é usar uma função barreira logarítmica no PL. Dado o PL na forma padrão (2.1), é possível definir um *problema de barreira* correspondente

$$\begin{aligned} \min_x \quad & c^T x - \tau \sum_{i=1}^n \ln x \\ \text{s.a.} \quad & x \in \mathcal{P}^+ \end{aligned}, \tag{P_\tau}$$

em que  $\tau > 0$  é um escalar, normalmente pequeno, que serve de parâmetro para uma família de problemas  $(P_\tau)$ , e que é chamado de *parâmetro de barreira*.

A presença da barreira logarítmica na função objetivo de  $(P_\tau)$  força o iterado a ficar no interior do ortante não-negativo, já que há uma penalização muito pesada quando os pontos estão perto do limite. Por outro lado, a influência da função barreira pode ser controlada através do parâmetro  $\tau$ . O peso na barreira regula a distância do iterado para o limite, ou seja, quando  $\tau \rightarrow 0$ , o problema  $(P_\tau)$  cada vez mais se parece com o problema (2.1). Esta estratégia só é possível se  $\mathcal{P}^+ \neq \emptyset$ . Além disso, se  $\mathcal{P}$  for limitado, então tanto (2.1) quanto  $(P_\tau)$  admitem solução ótima.

Como a função objetivo de  $(P_\tau)$  é estritamente convexa, o minimizador desta função, se existir, pode ser completamente caracterizado pelas condições KKT:

$$\begin{cases} Ax = b, \\ \tau X^{-1}e + A^T y = c, \\ (x, z) > 0. \end{cases}$$

Fazendo  $z = \tau X^{-1}e$ , ou equivalentemente  $xz = \tau e$ , obtemos a formulação padrão primal-dual das chamadas *condições KKT perturbadas*:

$$\left\{ \begin{array}{l} Ax = b, \\ A^T y + z = c, \\ xz = \tau e, \\ (x, z) > 0. \end{array} \right. \begin{array}{l} (2.10a) \\ (2.10b) \\ (2.10c) \\ (2.10d) \end{array}$$

Se as condições KKT perturbadas admitem solução para algum  $\hat{\tau} > 0$ , então admitem solução para qualquer  $\tau > 0$ . O sistema (2.10) se aproxima mais e mais de (2.6) quando  $\tau \rightarrow 0$  e determina uma única curva suave, contínua e parametrizada pela variável  $\tau$ , definida por  $\mathcal{C} = \{(x(\tau), y(\tau), z(\tau)) : \tau > 0\}$ .

Para cada  $\tau$  o ponto da curva é completamente caracterizado como sendo a solução única do sistema (2.10). Além disso, quando  $\tau \rightarrow 0$ ,  $\mathcal{C}$  converge para uma solução ótima primal-dual do PL. Essa curva é chamada em MPI de *trajetória central*. Esta trajetória nos guia para uma solução em que os pares  $x_i z_i$  são estritamente positivos e decrescem a zero na mesma taxa de decrescimento de  $\tau$ . Estudos sobre a trajetória central podem ser encontrados em Bayer e Lagarias [4, 5], Megiddo [48] e Sonnevend [66], entre outros.

Usando o fato de que para algum  $\tau > 0$  o ponto  $(x(\tau), y(\tau), z(\tau))$  é primal-dual factível, podemos definir o gap de dualidade  $g(\tau)$  para  $(P_\tau)$ , de forma semelhante à Equação (2.7) como função do parâmetro de barreira:

$$g(\tau) = c^T x(\tau) - b^T y(\tau) = x(\tau)^T T(\tau), \quad (2.11)$$

isto é, para cada valor de  $\tau$ , o gap de dualidade corresponde ao gap de complementaridade. Logo reduzir um significa reduzir o outro. Além disso, como  $xz - \tau e = 0$  por (2.10c), então

$x_i z_i = \tau, i = 1, \dots, n$  e logo

$$g(\tau) = x(\tau)^T z(\tau) = \sum_{i=1}^n x_i(\tau) z_i(\tau) = n\tau. \quad (2.12)$$

Isso significa que quando  $\tau \rightarrow 0$  temos  $g(\tau) \rightarrow 0$ . As equações (2.11) e (2.12) em conjunto com o fato de que  $c^T x(\tau) \geq c^T x^* = b^T y^* \geq b^T y(\tau)$ , implicam que

$$c^T x(\tau) \xrightarrow{\tau \rightarrow 0} c^T x^* \quad e \quad b^T y(\tau) \xrightarrow{\tau \rightarrow 0} b^T y^*.$$

Assim, as funções objetivos do problema perturbado convergem para aquelas obtidas por uma solução ótima  $(x^*, y^*, z^*)$  do problema original. Mais que isso, vale o seguinte resultado [48]:

**Teorema 2.4.** *Se um problema de PL for primal e dual factível e  $A$  for de posto completo, então*

$$x(\tau) \rightarrow x^* \quad e \quad (y(\tau), z(\tau)) \rightarrow (y^*, z^*),$$

*sempre que  $\tau \rightarrow 0$ .*

Este teorema significa que, sob algumas condições, a trajetória central converge para a solução ótima dos problemas (2.1) e (2.2). Em consequência disto, a trajetória central pode ser um bom guia para encontrar o conjunto ótimo primal-dual. Métodos que se baseiam na trajetória central são chamados Métodos de Pontos Interiores *seguidores de caminho*.

A solução que se encontra por seguir a trajetória central é caracterizada pela *complementaridade estrita*. Isso é descrito pelo seguinte resultado.

**Teorema 2.5** (Complementaridade estrita). *Se (2.1) e (2.2) forem factíveis, então existe um ponto  $x^* \in \mathcal{P}$  e um par  $(y^*, z^*) \in \mathcal{D}$  tais que*

$$(x^*)^T z^* = 0 \quad e \quad x_i^* + z_i^* > 0, \quad \text{para } i = 1, \dots, n.$$

Uma solução  $(x^*, z^*)$  que satisfaça o teorema acima é chamada estritamente complementar. Nesses termos, definimos o conceito de partição ótima. Seguindo as ideias de Jansen [36], definimos o suporte do vetor  $v \in \mathbb{R}^n$  como

$$\text{supp}(v) = \{i : v_i > 0, i = 1, \dots, n\}.$$

e a partição do conjunto de índices  $\{1, \dots, n\}$  através da definição dos conjuntos

$$\mathcal{B} = \text{supp}(x^*) \quad \text{e} \quad \mathcal{M} = \text{supp}(z^*).$$

Do Teorema 2.5 segue que essa é uma partição bem definida, no sentido de que as soluções que possuem a propriedade da complementaridade estrita satisfazem tanto  $\mathcal{B} \cap \mathcal{M} = \emptyset$  quanto  $\mathcal{B} \cup \mathcal{M} = 1, \dots, n$ . A noção de complementaridade estrita e de partição ótima são ideias recorrentes em análises de MPI.

Um fato que vale a pena chamar atenção é que, nos casos em que o PL tenha múltiplas soluções, um MPI converge para uma vizinhança do centro analítico da face ótima ao invés de em um vértice, como no método simplex. Isso significa que através do conceito de partição ótima, interpretamos essa situação como sendo a determinação de todo o conjunto de soluções ótimas. Em contraste a isso, a escolha do vértice ótimo obtido pelo método simplex é arbitrária e depende de alguns fatores como regras de pivoteamento.

Não raramente, ter uma solução básica que identifica um vértice é equivalente a encontrar uma solução *exata*. Entretanto, podemos discutir o que significa solução *exata*. Em vários casos, não é necessária uma precisão adicional de uma solução em um vértice, ao invés da solução que se encontra no centro analítico da face ótima. Do ponto de vista da programação inteira, temos uma exceção a esse caso, já que as soluções inteiras estão nos vértices do envoltório convexo de pontos inteiros factíveis. A questão é que quase nunca temos o envoltório disponível [53]. A diferença entre ter ou não uma base ótima ou uma partição ótima é de importante consequência

para o uso de soluções na análise de sensibilidade [36, 80].

Vavasis e Ye [69] discutiram propriedades da trajetória central, demonstrando que a mesma é caracterizada por  $\mathcal{O}(n^2)$  curvas de alto grau e segmentos nos quais a trajetória é relativamente reta. Mais que isso, em uma vizinhança suficientemente pequena do ótimo a trajetória central se torna uma linha reta e com isto, nesta região, o método apresenta a boa propriedade da convergência quadrática que os métodos do tipo Newton possuem [48].

Considere o limite de  $(P_\tau)$  quando  $\tau \rightarrow \infty$ . Este é o ponto do qual a trajetória central *inicia*. Tomar esse limite corresponde a encontrar o ponto  $\check{x}$  que minimiza a função de barreira, isto é,

$$\check{x} = \arg \min_{x \in \mathcal{P}^+} \left\{ - \sum_{i=1}^n \ln x_i \right\}.$$

O ponto  $\check{x}$  é o *centro analítico* do politopo factível, e foi primeiramente estudado por Sonnevend [66]. Dada a convexidade estrita da função de barreira, o conceito de centro analítico está bem definido. Como o centro analítico minimiza o problema de barreira, ele é o ponto que se encontra mais distante da fronteira do politopo.

Entretanto, existe um problema em definir a trajetória central em termos de centro analítico: a trajetória central é afetada pela presença de restrições redundantes. Isso acontece porque tem-se aqui um conceito exclusivamente analítico, o qual não explora considerações geométricas. Para vencer essa desvantagem, outros tipos de centro – centro de gravidade, centro do elipsoide de volume máximo que pode ser inscrito em  $\mathcal{P}$ , centro volumétrico – podem ser definidos, mas usualmente são muito custosos de se calcular [29].

Tal desvantagem da trajetória central foi apresentada como potencialmente geradora de consequências extremas por Deza et al. [17], que conseguiram replicar o comportamento do método simplex no cubo de Klee-Minty num contexto de pontos interiores. Isso foi conseguido por, patologicamente, se adicionar um número exponencial de restrições redundantes e paralelas às faces do cubo, tal que a trajetória central torna-se altamente destorcida e está presente em

vizinhanças arbitrariamente pequenas de todos os vértices do cubo.

Mencionamos aqui o fato de que técnicas de pré-processamento são usualmente implementadas em códigos maduros de modo a remover tanto quanto possível essas restrições redundantes. Embora sejam implementadas com heurísticas de sucesso, elas geralmente não são ótimas.

## 2.2 Métodos seguidores de caminho

### 2.2.1 Modelo geral para métodos seguidores de caminho

A maioria dos métodos primais-duais seguidores de caminho realizam passos de Newton em direção a pontos de  $\mathcal{C}$  para os quais  $\tau > 0$ , ao invés de passos puramente Newton para  $F$ . Como vimos, o sistema (2.6) é resolvido por fazer os pares complementares se alinharem – veja Equação (2.10c) – ao mesmo tempo em que  $(x, z) > 0$ . Como estes passos são desviados em direção ao ortante positivo definido por  $(x, z) > 0$ , usualmente é possível tomar passos maiores que um passo de Newton para  $F$  antes de violar a restrição de positividade. Para descrever esta busca enviesada, introduzimos um *parâmetro de centralização*  $\eta \in [0, 1]$ , na equação (2.10c), que faz com que  $\tau$  decresça monotonicamente em cada iteração.

As equações para encontrar a direção de busca tornam-se

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_P \\ r_D \\ r_C + \eta\tau e \end{bmatrix} = \begin{bmatrix} r_P \\ r_D \\ r_\tau \end{bmatrix}, \quad (2.13)$$

em que  $\Delta w = (\Delta x, \Delta y, \Delta z)$  é um passo de Newton apontando para  $(x(\eta\tau), y(\eta\tau), z(\eta\tau)) \in \mathcal{C}$  e o par  $x_i z_i$  é igual à  $\eta\tau$ . Mesmo nesse caso, a direção de busca encontrada  $\Delta w$  nem sempre pode ser utilizada e o novo iterado é calculado novamente usando um teste da razão, isto é, escolhendo  $\alpha_k$ , tal que

$$w^{k+1} = w^k + \alpha_k \Delta w \quad (2.14)$$

e  $(x^{k+1}, z^{k+1}) > 0$ .

### O parâmetro $\eta$

Por um lado, se  $\eta = 1$  então (2.13) define a *direção de centragem* na qual o passo de Newton é dado em direção à trajetória central. Embora esta direção se afaste dos limites do ortante não-negativo, ela pouco ou nada reduz o valor de  $\tau$ . Entretanto, ao se mover mais perto de  $\mathcal{C}$ , estes passos formam uma boa base para um progresso substancial na próxima iteração. Isto porque, como a próxima iteração está mais perto da trajetória central, será possível dar um passo relativamente grande sem que se deixe o ortante não-negativo. Por outro lado, se  $\eta = 0$  temos a direção afim-escala pura como em (2.9). A maioria dos métodos usa valores intermediários de  $\eta$  no intervalo aberto  $(0, 1)$ , de modo a tentar alcançar de maneira eficiente o duplo objetivo de reduzir  $\tau$  e melhorar a centralidade. Esta escolha é dependente do método.

Vamos definir agora uma maneira de escolher  $\tau$  utilizando a equação (2.12). Considere dado um ponto estritamente factível, isto é,  $w^0 \in \mathcal{F}^+$ . Então o valor do parâmetro de barreira será dado por

$$\tau_k = \frac{(x^k)^T z^k}{n}. \quad (2.15)$$

Olhando por este ângulo,  $\tau_k$  representa a média dos valores dos produtos  $x_i^k z_i^k$ . Além disso, com o avanço das iterações, o parâmetro de centralização  $\eta$  faz com que o problema perturbado que estamos resolvendo (2.13) se aproxime cada vez mais do problema original (2.8).

Com estes conceitos e ideias em mão, podemos definir um modelo geral para métodos primais-duais, dado no Pseudo-Código 2.1.

### 2.2.2 Vizinhanças da trajetória central

Um método seguidor de caminho segue a trajetória central  $\mathcal{C}$  no interior da região factível em direção a uma solução ótima. No entanto, manter um iterado exatamente sobre  $\mathcal{C}$  é um objetivo muito difícil, senão impossível. Isto porque encontrar um ponto que resolve a condição

---

**Algoritmo 2.1** Modelo geral para um método primal-dual seguidor de caminho.

---

**Dado:**  $w^0 \in \mathcal{F}^+$

$k \leftarrow 0$

**Repita**

Resolva (2.13) para algum  $\eta \in (0, 1)$ .

Encontre  $\alpha_k$ , tamanho de passo factível máximo na direção  $\Delta w^k$ .

Atualize o iterado conforme (2.14).

$k \leftarrow k + 1$

**Até** O critério de parada ser satisfeito.

---

de complementaridade perturbada (2.10c) para um  $\tau$  específico é um problema tão difícil quanto resolver o próprio PL.

Assim, já que computacionalmente é pouco eficiente caminhar ao longo de uma direção não-linear, um método seguidor de caminho faz com que os iterados fiquem em torno de  $\mathcal{C}$ . Isto significa restringir os iterados a uma *vizinhança* da trajetória central e, durante o progresso das iterações, segui-la até uma solução do PL. Podemos definir vários tipos de vizinhanças, no entanto sempre excluimos os pontos  $(x, z)$  que estão muito perto do limite do ortante não-negativo. Com isto direções de busca calculadas de qualquer ponto desta vizinhança progredem, mesmo que minimamente, em direção ao conjunto solução e ao mesmo tempo são obtidas de maneira mais fácil.

As duas vizinhanças de  $\mathcal{C}$  mais comuns são  $\mathcal{N}_2(\theta)$ , que tem como base a norma-2 e é definida por

$$\mathcal{N}_2(\theta) = \{w \in \mathcal{F}^+ : \|xz - \tau e\|_2 \leq \theta\tau\},$$

em que  $\theta \in (0, 1)$ , e a vizinhança  $\mathcal{N}_{-\infty}(\gamma)$ , baseada na norma- $\infty$ , dada por

$$\mathcal{N}_{-\infty}(\gamma) = \{w \in \mathcal{F}^+ : x_i z_i \geq \gamma\tau, \text{ para todo } i = 1, \dots, n\}, \quad (2.16)$$

para algum  $\gamma \in (0, 1)^2$ .

A vizinhança  $\mathcal{N}_2(\theta)$  é mais restrita já que certos pontos de  $\mathcal{F}^+$  não pertencem à  $\mathcal{N}_2(\theta)$ , não

---

<sup>2</sup>Tipicamente são usados os seguintes valores para os parâmetros:  $\theta = 0,5$  e  $\gamma = 10^{-3}$  [77, pg. 9].

importando quão perto fazamos  $\theta$  se aproximar de seu limitante superior 1. Por outro lado, se um ponto pertence à  $\mathcal{N}_{-\infty}(\gamma)$ , então cada par  $x_i z_i$  deve ser pelo menos um múltiplo  $\gamma$  de seus valores médios  $\tau$ . Com efeito, se escolhermos  $\gamma$  perto de zero, então quase toda região  $\mathcal{F}$  estará contida em  $\mathcal{N}_{-\infty}(\gamma)$ .

Mantendo todos os iterados dentro de uma dessas duas vizinhanças, métodos seguidores de caminho reduzem os produtos  $x_i z_i$  a zero mais ou menos na mesma taxa.

MPI seguidores de caminho são métodos de homotopia continuada [54] similares aos métodos de homotopia para equações não-lineares gerais. Estes definem uma trajetória que deve ser seguida para encontrar a solução. Tradicionalmente, métodos de homotopia mantêm os iterados em uma vizinhança *tubular* da trajetória, realizando mudanças incrementais no parâmetro e perseguindo a trajetória homotópica em busca da solução. Para MPI primais-duais, estas vizinhanças são cônicas, ao invés de tubulares, e tendem a ser amplas e relaxadas para valores grandes da medida de dualidade  $\tau$ , porém se tornam mais estreitas quando  $\tau \rightarrow 0$ , por conta da positividade exigida de  $(x, z)$ .

Direções de busca que se baseiam na vizinhança  $\mathcal{N}_2(\theta)$  podem ser utilizadas com  $\alpha = 1$ , e o parâmetro de barreira decresce pouco em cada iteração, dando lugar aos chamados *métodos de passo-curto*. Esta propriedade impõe e ao mesmo tempo produz os melhores resultados de complexidade existentes para PL: algoritmos de passo curto tem iterações de ordem  $\mathcal{O}(\sqrt{n} \ln(1/\varepsilon))$ . Os métodos de passo curto são os mais simples dos MPI pois fixam  $\alpha_k \equiv 1$  e  $\eta_k \equiv \eta$  dependente de  $\theta$  e foram introduzidos por Kojima, Mizuno e Yoshise [45] e Monteiro e Adler [51]. Como a redução do parâmetro de barreira é muito pequena, estes tipos de métodos são pouco usados na prática.

Métodos baseados na vizinhança  $\mathcal{N}_{-\infty}(\gamma)$  permitem iterados que seguem a trajetória central de maneira mais relaxada pois se obtém mais liberdade para manobrar e é possível se aproximar da fronteira da região factível. De fato, para  $\gamma$  pequeno quase todo o conjunto  $\mathcal{F}^+$  está contido em  $\mathcal{N}_{-\infty}(\gamma)$ .

Por outro lado, direções calculadas a partir de pontos desta vizinhança têm propriedades mais fracas e o teste da razão é necessário para assegurar a positividade de  $(x, z)$ . Portanto, métodos que levam em conta a vizinhança larga são menos conservadores que os métodos de passo-curto e podem decrescer o parâmetro de barreira mais rapidamente. Recentemente, Potra [60] provou  $\mathcal{O}(\sqrt{n}L)$  de iterações e convergência quadrática de um método que faz uso da vizinhança larga. Estas são os chamados de *métodos de passo-longo* e implementações eficientes de MPI baseiam-se em alguma variação de tais métodos.

### Vizinhança Simétrica

A vizinhança  $\mathcal{N}_{-\infty}(\gamma)$ , como vimos, produz uma boa base para métodos práticos, pois permite o parâmetro de barreira reduzir rapidamente. Mesmo assim, permite que os iterados produzam produtos  $x_i z_i$  muito diferentes, já que não impõe um limite superior para a complementaridade.

Essa dificuldade, quando ultrapassada, permite resultados muito melhores na prática, já que os pares  $x_i z_i$ , quando mal escalonados, influenciam o mau comportamento do método de Newton.

A razão

$$\varrho(xz) = \frac{\min_i \{x_i z_i\}}{\max_i \{x_i z_i\}} \quad (2.17)$$

dá uma medida de proporção entre o maior e o menor par de complementaridade. Claramente,  $\varrho(xz) \in (0, 1]$  e, quando o iterado é perfeitamente centrado, este valor é igual a 1. Uma análise da medida (2.17) pode ser encontrada em Jansen, Roos e Terlaky [37].

Gondzio [25] e posteriormente Colombo [8] e Colombo e Gondzio [9] definiram uma vizinhança que tenta fazer com que em cada iteração tenha-se uma certa *centralidade* dos iterados. Por centralidade, os autores entendem a baixa dispersão dos pares  $x_i z_i$ . Quando a discrepância nos valores dos pares complementares é muito grande, e portanto, pouca centralidade é alcançada, as direções de busca não são adequadas, não só com valores pequenos mas também com valores grandes de  $x_i z_i$ . Com efeito, Colombo [8, pp. 26] propõe que a noção de dispersão dos produtos complementares não é adequadamente representada em um ambiente computacional por

nenhuma das vizinhanças  $\mathcal{N}_{-\infty}$ .

Com o propósito de corrigir estes problemas foi proposta a *vizinhança simétrica*  $\mathcal{N}_s$ , na qual os pares complementares não só satisfazem um limite inferior mas também um limite superior, sendo é definida como

$$\mathcal{N}_s(\gamma) = \left\{ w \in \mathcal{F}^+ : \gamma\tau \leq x_i z_i \leq \frac{1}{\gamma}\tau, i = 1, \dots, n \right\}, \quad (2.18)$$

em que  $\tau = x^T z/n$  e  $\gamma \in (0, 1)$ .

A vizinhança simétrica (2.18) pode ser considerada uma modificação de  $\mathcal{N}_{-\infty}$ . No entanto, ela não deixa que os produtos complementares se tornem muito grandes em relação à média. Além disso,  $\mathcal{N}_s$  promove uma diminuição dos pares complementares que são muito grandes, permitindo uma melhor centralidade.

Colombo [8] determinou o valor de  $n$  para o qual a vizinhança simétrica impõe um limite superior mais estreito:

$$n > \frac{1 + \gamma}{\gamma}.$$

Para  $\gamma = 0,1$ , o limite é estreito sempre que  $n > 11$ .

Além disso, a complexidade do pior caso para um método de passo-longo factível que utiliza a vizinhança simétrica  $\mathcal{N}_s$  é de ordem  $\mathcal{O}(n \ln(1/\varepsilon))$  de iterações [9]. Isso significa que os limites superiores que diferenciam a vizinhança simétrica de  $\mathcal{N}_{-\infty}$  não produzem perdas teóricas, ao mesmo tempo em que contribuem com a possibilidade de deixar os iterados mais centrados.

De fato, Zhang e Tapia [83], ao provarem a convergência superlinear de um método primal-dual, utilizam uma vizinhança com parâmetros simétricos para melhorar os limites da convergência. No entanto, à época imaginavam que um limite superior, como o dado por  $\gamma\tau$ , não teria significância prática, previsão que não se concretizou [8, 9, 25].

### 2.2.3 Ponto inicial infactível

Note que, até agora, estamos assumindo que o método inicia um ponto estritamente factível  $w^0 \in \mathcal{F}^+$ . Neste caso,  $r_P = r_D = 0$ , no lado direito de (2.13), e assim a direção de busca encontrada através desta equação garante que

$$A\Delta x = 0 \quad \text{e} \quad A^T \Delta y + \Delta z = 0. \quad (2.19)$$

Consequentemente a factibilidade de todos os iterados é garantida, pois para  $\alpha = 1$  valem as seguintes igualdades:

$$\begin{aligned} A(x + \Delta x) &= Ax + A\Delta x = b, \\ A^T(y + \Delta y) + (z + \Delta z) &= (A^T y + z) + (A^T \Delta y + \Delta z) = c, \text{ e} \\ \Delta x^T \Delta z &= -\Delta x^T (A^T \Delta y) = -(A\Delta x)^T \Delta y = 0. \end{aligned}$$

Métodos que assumem esta hipótese são os chamados *MPI factíveis*.

Nem sempre, porém, obter um ponto inicial factível é uma tarefa trivial. É possível, por exemplo, encontrar um ponto inicial factível reformulando o problema ou utilizando um método do tipo *Big M*, mas estas reformulações podem causar distorções, instabilidades numéricas ou aumento de colunas densas. Além disso, a região factível pode ter o interior vazio, o que torna inválida a teoria apresentada até aqui. Uma abordagem totalmente diferente é baseada na formulação auto-dual [79], mas, nesse caso, para tornar o problema sempre factível, o tamanho do problema é aumentado, o que de certa forma aumenta o tempo computacional por exigir duas retrosubstituições<sup>3</sup> – ou *backsolves* – adicionais no cálculo de próximo iterado, devido à introdução de colunas extra.

---

<sup>3</sup>Utilizaremos aqui o termo retrosubstituição pra indicar ambas a substituição progressiva e a retrosubstituição.

## Métodos Infactíveis

Uma alternativa que permite que essas dificuldades sejam contornadas é utilizar *MPI infactíveis*. Métodos implementados que têm importância prática fazem uso de pontos infactíveis [11, 23, 25]. Em geral, métodos infactíveis exigem apenas que o ponto inicial seja tal que,  $w^0 \in \mathcal{Q}^+$ , isto é,  $(x^0, z^0)$  esteja no ortante positivo. Nesse caso  $r_P^0$  e  $r_D^0$  são não-nulos.

A direção de busca ainda é feita segundo (2.13), e portanto um passo de Newton em direção ao ponto  $(x(\eta\tau), y(\eta\tau), z(\eta\tau)) \in \mathcal{C}$  é encontrado. É possível provar que se fosse viável dar um passo completo ( $\alpha = 1$ ), a infactibilidade seria eliminada. No entanto, na prática, o limite da sequência de iterados é o ponto ótimo e somente quando o ponto atual está em uma vizinhança apropriada deste ótimo é que se dá um passo completo.

Kojima, Megiddo e Mizuno [44] estabeleceram resultados de convergência para um MPI infactível, bem como uma regra para o tamanho do passo que garante a convergência global do método. Isto é obtido ao utilizarmos uma vizinhança similar à  $\mathcal{N}_{-\infty}(\gamma)$  (veja equação (2.16)), dada por

$$\mathcal{N}_{-\infty}(\gamma, \beta) = \left\{ (x, y, z) \in \mathcal{Q}^+ : \frac{\|(r_P, r_D)\|}{\|(r_P^0, r_D^0)\|} \leq \beta \frac{\tau}{\tau_0}, x_i z_i \geq \gamma \tau, \forall i = 1, \dots, n \right\}, \quad (2.20)$$

em que  $\gamma \in (0, 1)$  e  $\beta \geq 1$  são parâmetros,  $\tau_0$  é dado por (2.15) e  $r_P^0, r_D^0$  são os resíduos iniciais primais e duais respectivamente.

Embora  $\mathcal{N}_{-\infty}(\gamma, \beta)$  não exija factibilidade em cada iteração, a primeira de suas condições dá uma prioridade para a factibilidade em relação à complementaridade. Em resumo, não se quer que o ponto satisfaça a restrição de complementaridade sem que a factibilidade seja satisfeita. Com efeito, se  $(x(\alpha), y(\alpha), z(\alpha)) = (x, y, z) + \alpha \Delta(x, y, z)$ , então

$$r_P(\alpha) = (1 - \alpha)r_P \text{ e } r_D(\alpha) = (1 - \alpha)r_D,$$

mostrando que a infactibilidade se reduz linearmente com  $\alpha$ .

Assim, se  $\alpha = 1$ , então os resíduos  $r_P$  e  $r_D$  tornam-se nulos, a factibilidade é restaurada e subsequentemente o método infactível torna-se igual a um método factível, já que todos os próximos iterados estarão em  $\mathcal{F}^+$ . Kojima, Megiddo e Mizuno [44] provaram a convergência global de um método com passos infactíveis, enquanto Zhang [82] provou a complexidade de  $\mathcal{O}(n^2 \ln(1/\varepsilon))$  para um método desse tipo. Para uma explanação completa de métodos infactíveis veja Wright [77, cap. 6].

## 2.3 Métodos de Pontos Interiores na Prática

### 2.3.1 Método Preditor-Corretor de Mehrotra

Do ponto de vista prático, a maioria dos códigos de implementação de MPI é uma modificação em alguma variação de um método do tipo preditor-corretor devido a Mehrotra [49]. O Método Preditor Corretor de Mehrotra (MPC-M) se baseia no modelo geral para métodos seguidores de caminho, dado no Pseudo-código 2.1, mas altera a direção de busca puramente newtoniana com correções que são computacionalmente muito baratas, porém que auxiliam a obtenção de uma melhor direção. Além disso, uma escolha adaptativa do parâmetro de centragem  $\eta$  é feita em cada iteração. Atualmente, nos códigos mais utilizados, tanto os acadêmicos como BPMPD, HOPDM, OOPS, OOQP, PCx, bem como os comerciais tais quais Cplex, Mosek e Xpress, algumas heurísticas e ideias de otimização também foram incorporadas, de modo que todos são variantes do MPC-M.

Este método usa aproximações de ordem maior da trajetória central  $\mathcal{C}$  – abordagem começada por Megiddo [48] e mais tarde desenvolvida por Monteiro, Adler e Resende [52] – e pontos infactíveis. Segundo Wright [77, pg 194], a contribuição principal de Mehrotra foi combinar estas ideias existentes da maneira certa e adicionar heurísticas engenhosas para escolher o parâmetro de centragem (adaptativamente), o tamanho do passo e o ponto inicial.

Até aqui, o tamanho de passo foi sempre considerado igual tanto para o problema primal, quanto para o problema dual. No entanto, a quase totalidade das implementações dos métodos de pontos interiores usa passos diferentes para as variáveis primais e duais. Esta ideia tem sido usada em quase todos os desenvolvimentos práticos de métodos primais-duais e contribuiu para reduzir em até 10% no número de iterações, tendo como base de testes o conjunto NETLIB [77, pg. 195].

Como se quer que  $(x, z)$  permaneça no ortante positivo, uma busca linear é empregada a fim de encontrar os tamanhos de passo primal  $\alpha_P$  e dual  $\alpha_D$  tais que  $x + \alpha_P \Delta x > 0$  e  $z + \alpha_D \Delta z > 0$ . Para isso, podemos fazer um teste da razão, isto é,

$$\alpha_P = \alpha_0 \min \left\{ -\frac{x_i}{\Delta x_i} : \Delta x_i < 0 \right\}, \quad \alpha_D = \alpha_0 \min \left\{ -\frac{z_i}{\Delta z_i} : \Delta z_i < 0 \right\} \quad (2.21)$$

em que  $\alpha_0$  é um fator que garante positividade estrita e em geral vale 0,9995. Além disso, embora algum tipo de vizinhança seja sempre utilizada, muitas vezes é permitido que o ponto esteja fora da mesma e por isso, com a perda da propriedade da convergência global em favor da eficiência computacional.

O método de Mehrotra gera a sequência de iterados infactíveis  $(x, y, z)$  tais que  $(x, z) > 0$ , e a direção de busca consiste em três componentes:

- (i) Uma direção afim-escala *preditora* que nada mais é que uma direção puramente newtoniana;
- (ii) Um termo de centralização, cujo tamanho é controlado pela escolha de forma adaptativa do parâmetro  $\eta$ ;
- (iii) Uma direção *corretora* que tenta compensar de alguma forma a não-linearidade perdida na direção de Newton.

Para compreendermos melhor o que essas ideias significam, note que, no sistema de Newton

(2.13), o lado direito pode ser separado da seguinte forma:

$$\begin{bmatrix} b - Ax \\ c - A^T y - z \\ -xz + \eta\tau \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - z \\ -xz \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \eta\tau \end{bmatrix}.$$

Assim, a direção preditora  $(\Delta x^{\text{af}}, \Delta y^{\text{af}}, \Delta z^{\text{af}})$  é obtida resolvendo-se o sistema de Newton com o lado direito dado por

$$\begin{bmatrix} b - Ax \\ c - A^T y - z \\ -xz \end{bmatrix} = \begin{bmatrix} r_P \\ r_D \\ r_C \end{bmatrix} \quad (2.22)$$

isto é,  $(\Delta x^{\text{af}}, \Delta y^{\text{af}}, \Delta z^{\text{af}})$  corresponde a direção afim-escala dada em (2.9). Como visto, essa direção otimiza fortemente em direção aos pontos para os quais a complementaridade é nula. Entretanto, como o alvo é  $XZe = 0$ , essas direções podem ser excessivamente influenciadas por pontos que têm complementaridade pequena, mas que não são ótimos.

### Direção com correção de segunda ordem

O MPC-M explora uma direção corretora de centralidade de modo a tentar remediar pontos que estão mal centrados. Esta direção caminha para perto da trajetória central e logo reduz o espalhamento dos produtos complementares em relação à média, embora não busque otimalidade.

Uma ferramenta importante para isso foi a avaliação dinâmica do parâmetro de centralização  $\eta$ . Para começar, queremos medir a eficácia da direção afim-escala, e por isso definimos  $\tau_{\text{af}}$  como sendo a média dos valores hipotéticos para os pares complementares resultantes de um passo na direção afim-escala, dada por

$$\tau_{\text{af}} = \frac{g_{\text{af}}}{n} = \frac{(x + \alpha_P^{\text{af}} \Delta x^{\text{af}})^T (z + \alpha_D^{\text{af}} \Delta z^{\text{af}})}{n}, \quad (2.23)$$

em que  $g_{\text{af}}$  é o gap de complementaridade previsto para a direção afim-escala. Note que  $\tau_{\text{af}}/\tau \in (0, 1)$ . Se  $\tau_{\text{af}} \ll \tau$  então a direção afim-escala é boa e permite uma significativa redução do gap de complementaridade. Por outro lado, se a razão  $\tau_{\text{af}}/\tau$  é próxima de 1, então pouco progresso está sendo feito na direção  $(\Delta x^{\text{af}}, \Delta y^{\text{af}}, \Delta z^{\text{af}})$  e um parâmetro  $\eta$  perto de 1 é recomendado. Essa escolha faz com que ocorra uma forte centralização, isto é, move-se para mais perto de  $\mathcal{C}$ . Logo, na próxima iteração, o método está melhor posicionado para reduzir o gap de complementaridade.

Mehrotra [49] utiliza a seguinte heurística na escolha do parâmetro de centralização:

$$\eta = \left( \frac{\tau_{\text{af}}}{\tau} \right)^3. \quad (2.24)$$

Essa escolha sugere um parâmetro de barreira dado por

$$\tau = \left( \frac{g_{\text{af}}}{x^T z} \right)^2 \tau_{\text{af}} = \left( \frac{g_{\text{af}}}{x^T z} \right)^3 \frac{x^T z}{n}. \quad (2.25)$$

Assim, se o passo preditor progride bem,  $\eta$  é escolhido pequeno e há pouca centralização. Caso contrário,  $\eta$  é escolhido próximo a  $\frac{x^T z}{n}$  e busca-se uma forte centralização. Mais geralmente, o parâmetro de centralização pode ser escolhido como

$$\eta = \left( \frac{\tau_{\text{af}}}{\tau} \right)^p = \left( \frac{g_{\text{af}}}{x^T z} \right)^p.$$

Além disso, Mehrotra [49] estudou o efeito de diferentes valores de  $p = 1, 2, 3, 4$  em um subconjunto de problemas da NETLIB e concluiu que não havia muita diferença para  $p$  entre 2 e 4.

Para calcular a direção centralizadora, bastaria resolver o sistema de Newton com lado direito dado por  $(0, 0, \eta\tau e)$ . No entanto, será mais eficiente calcular esse termo em conjunto com a direção corretora.

A direção afim-escala corresponde à aproximação linear da trajetória que liga o ponto atual

ao conjunto ótimo. Esta linearização produz um erro que pode ser determinado. De fato, assumindo que daremos um passo completo na direção afim-escala, temos que o vetor dos pares complementares é dado por

$$\hat{x}\hat{z} = (x + \Delta x^{\text{af}})(z + \Delta z^{\text{af}}) = xz + x\Delta z^{\text{af}} + z\Delta x^{\text{af}} + \Delta x^{\text{af}}\Delta z^{\text{af}} = \Delta x^{\text{af}}\Delta z^{\text{af}}$$

já que, ao usarmos o lado direito (2.22) para resolver o sistema de Newton, temos que  $x\Delta z^{\text{af}} + z\Delta x^{\text{af}} = -xz$ . Isso significa que, quando um passo completo é dado, os produtos  $\hat{x}_i\hat{z}_i$  se transformam em  $\Delta x_i^{\text{af}}\Delta z_i^{\text{af}}$ , ao invés de se anularem como era esperado. Este é exatamente o erro da aproximação linear e um passo corretor  $(\Delta x^c, \Delta y^c, \Delta z^c)$  é dado para tentar compensá-lo.

Podemos ainda ver que, idealmente, o próximo ponto deve satisfazer

$$(x + \Delta x)(z + \Delta z) = \eta\tau,$$

o que é equivalente a resolver o sistema não linear

$$z\Delta x + x\Delta z = -xz + \eta\tau - \Delta x\Delta z.$$

Comparando esta equação com (2.13), vemos que na direção afim-escala falta exatamente o termo  $\Delta x\Delta z$ . Para corrigir isso, o sistema de Newton é resolvido com o seguinte lado direito

$$\begin{bmatrix} 0 \\ 0 \\ -\Delta x^{\text{af}}\Delta z^{\text{af}} + \eta\tau \end{bmatrix}, \quad (2.26)$$

encontrando a direção  $(\Delta x^c, \Delta y^c, \Delta z^c)$ . Note que tal direção combina a centralização e o termo de segunda ordem que é o erro da linearização.

Uma vez que as direções preditora e corretora foram calculadas, elas são simplesmente adicio-

nadas de maneira a ter a direção desejada:

$$(\Delta w, \Delta y, \Delta z) = (\Delta x^{\text{af}}, \Delta y^{\text{af}}, \Delta z^{\text{af}}) + (\Delta x^c, \Delta y^c, \Delta z^c), \quad (2.27)$$

e o próximo iterado será dado por

$$(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + (\alpha_P \Delta x, \alpha_D \Delta y, \alpha_D \Delta z), \quad (2.28)$$

em que  $\alpha_P$  e  $\alpha_D$  são os tamanhos de passo primal e dual dado por (2.21).

Resumimos o método preditor-corretor de Mehrotra no Pseudo-Código 2.2.

---

**Algoritmo 2.2** Método preditor-corretor de Mehrotra.

---

**Dado:**  $(x^0, y^0, z^0)$  inicial tal que  $(x^0, z^0) > 0$

**Repita**

Resolva o sistema de Newton com o lado direito dado por (2.22).

Encontre tamanho de passo  $\alpha_P^{\text{af}}$  e  $\alpha_D^{\text{af}}$  para a direção afim-escala

Encontre  $\tau$ , dado por (2.25) e calcule  $\eta$  de acordo com (2.24).

Resolva o sistema de Newton com o lado direito dado por (2.26).

Encontre tamanhos de passos  $\alpha_P$  e  $\alpha_D$  tais que o próximo iterando continue sendo interior.

Atualize o iterado conforme (2.28).

$k \leftarrow k + 1$

**Até** O critério de parada ser satisfeito.

---

A principal vantagem do MPC-M é que, na prática, ele produz tamanho de passos maiores antes de violar as restrições de não-negatividade. Isso normalmente significa que se está economizando iterações: de fato Mehrotra relata que temos economia da ordem de 35% a 50% quando comparada com outras estratégias [49]. Nesse caso, o custo é apenas de uma retrossubstituição a mais, já que a matriz do sistema de Newton é sempre a mesma. Assim, se o custo da fatoração é elevado, economia nos custos computacionais também é alcançada por essa técnica. De fato, o método de Mehrotra é vantajoso em relação a todas as implementações para programação linear que usam métodos diretos para calcular as direções de Newton [8, pg. 40].

### 2.3.2 Método das múltiplas correções de centralização

O Método Preditor Corretor de Mehrotra é baseado na hipótese de que um passo completo na direção corretora ocorre, o que raramente é possível. Mais que isso, tentar corrigir a complementaridade fazendo todos os seus produtos iguais ao valor  $\tau$  é um objetivo muito forte e algumas vezes pouco eficaz. É possível minorar estes problemas através de uma extensão do MPC-M que utiliza mais correções lineares de uma forma inteligente.

O *Método de Múltiplas Correções* deve-se a Gondzio [25] e é uma tentativa de melhorar a centralização do iterado atual. Essa abordagem busca produzir produtos  $x_i z_i$  mais homogêneos e forçar um aumento do tamanho do passo corrigindo o parâmetro de centralização de Mehrotra.

Implicitamente Gondzio [25] utiliza a Vizinhança Simétrica dada em (2.18), de modo a ter produtos perto o suficiente da trajetória central. Nesta descrição, o passo é dado na direção

$$\Delta w = \Delta w^m + \Delta w^g,$$

em que  $\Delta w^m$  é a direção de Mehrotra (2.27), à qual uma ou mais correções  $\Delta w^g$  podem ser aplicadas. Outras escolhas da direção preditora  $\Delta w^m$  podem ser feitas [9].

Dada a direção preditora-corretora  $\Delta w^m$ , seja

$$\tilde{\alpha}_P = \min(\alpha_P + \delta, 1) \quad \text{e} \quad \tilde{\alpha}_D = \min(\alpha_D + \delta, 1)$$

para algum  $\delta \in (0, 1)$  fixo, entendido como uma quantidade desejável de aumento do passo. O ponto de tentativa

$$\tilde{x} = x + \tilde{\alpha}_P \Delta x^m, \quad \tilde{z} = z + \tilde{\alpha}_D \Delta z^m$$

é calculado, bem como os respectivos produtos complementares  $\tilde{t} = \tilde{x}\tilde{z} \in \mathbb{R}^n$ . Note que, por aumentar o tamanho do passo, o ponto de tentativa é infactível (exterior), mas em compensação é utilizado apenas para buscar corrigir a centralização, logo esse fato não causa preocupação.

Obviamente os produtos  $\tilde{t}$  dificilmente serão iguais a  $\tau$  como desejaríamos. Alguns componentes são bem menores que  $\tau$  – inclusive podendo ser negativos – e outros muito maiores. Impondo que  $\tilde{t} \in \mathcal{N}_s(\gamma)$ , colocamos limites superiores e inferiores para cada par complementar, também verificamos as componentes que não satisfazem a vizinhança simétrica.

De fato, o que faremos é mover produtos pequenos ( $\tilde{t}_i \leq \gamma\tau$ ) para o limite inferior  $\gamma\tau$  e produtos grandes ( $\tilde{t}_i \geq \gamma^{-1}\tau$ ) para o limite superior  $\gamma^{-1}\tau$ . Os produtos que estiverem na vizinhança simétrica já satisfazem estes limites e portanto estão razoavelmente pouco dispersos em relação à média. Na prática estamos movendo os iterados para dentro de  $\mathcal{N}_s(\gamma)$ .

Nesse caso, o termo  $\Delta w^g$  será encontrado resolvendo o sistema de Newton com o lado direito dado por

$$r = \begin{bmatrix} 0 \\ 0 \\ \xi \end{bmatrix} \quad (2.29)$$

em que a meta  $\xi$  é definida como

$$\xi_i = \begin{cases} \gamma\tau - \tilde{t}_i, & \text{se } \tilde{t}_i \leq \gamma\tau, \\ \frac{1}{\gamma}\tau - \tilde{t}_i, & \text{se } \tilde{t}_i \geq \gamma^{-1}\tau, \\ 0, & \text{caso contrário.} \end{cases}$$

Ressalte-se que o ponto que estamos buscando não pertence à trajetória central mas à vizinhança simétrica e que nem todos os produtos complementares sofrem correção. Assim, a meta dada por (2.29) é realmente alcançada.

Dada uma nova direção, poderemos repetir o processo de atualizá-la tantas vezes quantas forem convenientes. Basta fazermos  $\Delta^m \leftarrow \Delta$  e procurar novamente a direção  $\Delta^g$ . No entanto, o número máximo de correções de centralização permitida depende do problema. Tal número é obtido em [25] de modo heurístico. Note que a direção corretora obtida é aceita contanto que o

tamanho do passo tenha aumentado uma fração desejada.

Em códigos como o PCx [11], o número máximo de correções é calculado considerando-se o esforço relativo entre o número de operações necessárias para calcular a decomposição da matriz resultante do sistema de Newton e o número de operações necessárias para fazer a retrossubstituição. Além disso estima-se a redução do número de iterações. Em [25] o custo da fatoração não foi considerado.

As experiências computacionais apresentadas por Gondzio, principalmente utilizando seu *solver* HOPDM [24, 25], mostram que a estratégia das múltiplas correções realmente é efetiva, pois os passos primais e duais calculados para a direção composta são maiores que aqueles correspondentes à direção preditora. Isso resulta em uma redução do número de iterações. O custo de cada correção neste caso é o mesmo, ao contrário de outros métodos cujo custo aumenta a cada correção. Praticamente todos os códigos modernos utilizam essa estratégia [77, Apêndice B]. Uma generalização desta pode ser encontrada em [9], no qual a direção de busca é dada por  $\Delta w = \Delta w^m + \omega \Delta w^g$  em que usa-se o parâmetro de peso  $\omega \in (0, 1]$ . Este peso é encontrado através de uma busca linear tal que a solução  $\hat{w}$  maximiza o tamanho de passo  $\alpha$ .

### 2.3.3 Ponto Inicial

Como dissemos, códigos práticos que resolvem PL através de MPI utilizam pontos inactíveis, exigindo apenas que  $(x, z) > 0$ . Nada obstante, encontrar tal ponto seja tarefa mais fácil que encontrar um ponto inicial factível, obviamente devemos ter uma escolha feita de forma a acelerar a convergência do método.

A escolha de um iterado inicial com boas propriedades é crítica e é uma tarefa difícil tanto para métodos factíveis quanto inactíveis. O ponto deve possuir boas propriedades nas seguintes características: centralidade do ponto e magnitude das inactibilidades.

Mehrotra [49] introduziu uma heurística para encontrar um ponto inicial que satisfaz essas

hipóteses. Nesta, os seguintes problemas de Quadrados Mínimos são resolvidos

$$\begin{aligned} \min x^T x \quad \text{s. a.} \quad Ax = b \\ \min_{(y,z)} z^T z \quad \text{s. a.} \quad A^T y + z = c \end{aligned}$$

que tentam encontrar um ponto  $(\tilde{x}, \tilde{y}, \tilde{z})$  que satisfaça as restrições primais e duais. As soluções para estes problemas são dada por

$$\tilde{x} = A^T(AA^T)^{-1}b, \quad \tilde{y} = (AA^T)^{-1}Ac \quad \text{e} \quad \tilde{z} = c - A^T\tilde{y}.$$

Tal ponto é então transladado para o ortante positivo e o ponto inicial é dado por

$$(x^0, y^0, z^0) = (\tilde{x} + \vartheta_x e, \tilde{y}, \tilde{z} + \vartheta_z e), \tag{2.30}$$

em que  $\vartheta_x$  e  $\vartheta_z$  são escalares positivos tais que  $(x^0, z^0) > 0$ .

Essa estratégia de Mehrotra tem algumas questões a serem consideradas. Ela depende do escalonamento da matriz, é afetada pela presença de restrições redundantes que porventura tenham escapado do pré-processamento, e não garante que o ponto inicial seja bem centrado. Mesmo assim, é uma estratégia comum em códigos de pontos interiores por ser considerada uma heurística muito boa para determinar o ponto inicial, além de ter custo computacional equivalente ao de uma iteração de um MPI.

Gertz e Wright [23], em seu *solver* OOQP, propõem uma heurística que se baseia nos mesmos princípios – que o ponto inicial seja ao mesmo tempo “bem centrado” e “não muito infactível” – agora para usar um MPI no contexto de Programação Quadrática. Descreveremos a seguir essa heurística, no contexto de PL.

Primeiramente, a *norma dos dados* é calculada, a qual é definida como a raiz quadrada da magnitude do maior elemento dos dados do problema, isto é, o maior elemento em valor absoluto

entre a matriz  $A$  e os vetores  $c$  e  $b$ .

Após isso, atribui-se a todas as componentes de  $x$  e  $z$  essa norma dos dados e às componentes de  $y$  o valor zero. A partir desse ponto, o sistema de Newton (2.9) é resolvido e um passo completo na direção afim-escala é realizado, tal que o ponto  $(\tilde{x}, \tilde{y}, \tilde{z})$  é encontrado. Certamente esse procedimento resulta em um ponto infactível. Essa heurística procede agora, de forma similar à heurística de Mehrotra, isto é,  $(\tilde{x}, \tilde{y}, \tilde{z})$  é transladado para o ortante positivo.

Para tal, o valor máximo da violação dos limites das variáveis é calculado, digamos

$$\vartheta = \max_{i \in I_V} \{-x_i, -z_i, 0\},$$

em que  $I_V$  é o conjunto de índices de  $x$  e  $z$  que violaram a não-negatividade, e o ponto com o valor

$$\tilde{\vartheta} = 100 + 2\vartheta,$$

é transladado tal que o ponto inicial seja interior o *suficiente*. Com efeito, o ponto inicial será

$$(x^0, y^0, z^0) = (\tilde{x} + \tilde{\vartheta}e, \tilde{y}, \tilde{z} + \tilde{\vartheta}e).$$

Note que tal ponto, assim como o ponto de Mehrotra, também tem custo computacional de uma iteração de MPI.

As estratégias expostas acima podem ser compreendidas como uma tentativa de encontrar um ponto inicial que esteja perto o suficiente da trajetória central, mas com produtos complementares  $x_i z_i$  maiores que uma fração adequada de  $\tau$ , e que a razão

$$\frac{\|(r_P, r_D)\|}{\tau}$$

não seja muito grande.

Aliás, aqui a *métrica* dada pela vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$  é utilizada, como foi definida em (2.20).

Com isso, essas heurísticas têm como objetivo impedir que o tamanho do passo  $\alpha$  seja muito pequeno, ao mesmo tempo que tendem a limitar o valor de  $\|(r_P^k, r_D^k)\|/\tau_k$ . Além disso, ao utilizarmos  $\mathcal{N}_{-\infty}(\gamma, \beta)$ , desejamos que a infactibilidade seja levada a zero na mesma taxa que a *gap* de dualidade. Para uma análise de várias estratégias de pontos iniciais em MPI veja os trabalhos de Andersen et al. [2], D’Apuzzo, Simone e Serafino [16], Gertz, Nocedal e Sartenaer [22] e Vanderbei e Shanno [68].

### 2.3.4 Critério de Parada

Os MPI, ao contrário do Simplex, por exemplo, encontram uma solução apenas assintoticamente. Por conta do parâmetro de barreira  $\tau$ , um MPI jamais encontra uma solução exata do problema de PL. Com isso, é necessário um critério de parada que possa decidir quando o iterado atual está próximo o suficiente do conjunto solução.

Códigos práticos de pontos-interiores geralmente geram uma sequência de pontos  $\{(x^k, y^k, z^k)\}$  para os quais  $x^k$  e  $z^k$  são ambos estritamente positivos. Nenhum desses iterados pode ser uma solução de fato do PL já que para estes,  $x_i^k z_i^k > 0$ , para  $k = 0, 1, 2, \dots$ , o que viola a condição de complementaridade (2.6d). Com efeito, os iterados convergem para um conjunto de soluções sem de fato alcançá-lo, e para efeitos práticos um iterado avançado –  $(x^k, y^k, z^k)$  para  $k$  suficientemente grande – serve muito bem como uma solução aproximada. Para outras aplicações e para propósitos teóricos, uma solução ótima  $(x^*, y^*, z^*) \in \mathcal{F}$  pode ser necessária. Para tanto, nesses códigos, há uma fase de *terminação finita*, a qual toma um iterado interior suficientemente avançado para um ponto em  $\mathcal{F}$ , encontrando, se for o caso, a base ótima. É necessário também saber quão avançado o iterado precisa ser, tal que o número de iterações de ponto-interior não seja muito grande. Veja o texto de Wright [77, Cap. 7] para uma explanação completa sobre terminação finita em MPI.

A maioria das implementações de pontos interiores, por estar trabalhando com precisão finita, garante apenas um certo grau de exatidão ou tolerância, tanto para a factibilidade quanto para

a otimalidade. Grande parte dos códigos utiliza os seguintes critérios [11, 28]:

$$\frac{\|Ax - b\|}{1 + \|x\|_\infty} \leq \text{tol}_L, \quad (2.31a)$$

$$\frac{\|A^T y + z - c\|}{1 + \|z\|_\infty} \leq \text{tol}_L, \quad (2.31b)$$

$$\frac{|c^T x - b^T y|}{1 + |b^T y|} \leq \text{tol}_G. \quad (2.31c)$$

Os valores de  $\text{tol}_L$  e  $\text{tol}_G$  são em geral potências de 10, e dependem da precisão desejada. Na literatura,  $\text{tol}_L = \text{tol}_G = 10^{-8}$  é a escolha preferida.

Os critérios (2.31a) e (2.31b) exigem que os resíduos primais e duais sejam menores ou iguais a uma tolerância. Assim estaríamos, ainda que assintoticamente, com pontos factíveis. O critério (2.31c) faz uso do Lema 2.3 (Dualidade Forte para PL) e é indicador de otimalidade. Os denominadores em cada critério são construídos de modo a permitirem avaliar relativamente tanto viabilidades primal e dual quanto otimalidade.

Um outro conjunto de critérios muito parecido com (2.31) é

$$\frac{\|Ax - b\|}{1 + \|b\|} \leq \text{tol}, \quad (2.32a)$$

$$\frac{\|A^T y + z - c\|}{1 + \|c\|} \leq \text{tol}, \quad (2.32b)$$

$$\frac{x^T z}{1 + |c^T x|} \leq \text{tol}, \quad (2.32c)$$

em que  $\text{tol} = 10^{-8}$ . Note que agora, em contraste com (2.31), os denominadores de (2.32a) e (2.32b) são fixos, o que de certa maneira diminui a quantidade de operações, já que não é mais necessário encontrar a norma – ainda que norma- $\infty$  – dos vetores  $x$  e  $z$  em cada iteração.

O valor de  $x^T z$ , é usado em (2.32c) ao invés do *gap* dual como em (2.31c), pois além de, para pontos factíveis, *gap* dual e *gap* de complementaridade serem iguais – vide Equação (2.7) –, é

razoável que se  $x^T z$  convergir para zero, então o *gap* dual também convirja para zero.

Todas estas ideias representam o estado da arte no que diz respeito a MPI para problemas de PL. Servirão como estrutura e base para os desenvolvimentos que faremos neste trabalho.

## Capítulo 3

# Método de Escolha Otimizada de Parâmetros

“Every really new idea looks crazy at first.”

---

(Alfred North Whitehead)

Este capítulo tem o objetivo de apresentar o desenvolvimento de uma função de mérito polinomial, utilizada em cada passo do método de pontos interiores proposto, que combina três direções – a direção afim-escala, a direção de centralização e a direção de correção. Além disso, escrevemos a vizinhança  $\mathcal{N}_{-\infty}$ , ajustada para pontos infactíveis, em termos das mesmas variáveis do polinômio. Isso porque a escolha do peso de cada uma das direções, na direção final, é feita através da otimização global da função de mérito sujeita a um conjunto de restrições que garantem que o próximo ponto continuará na vizinhança. Os pesos das direções, são em geral parâmetros escolhidos *a priori*, nos diversos MPI. Em nosso caso, a escolha será determinada em cada iteração do algoritmo, de maneira otimizada. Esses procedimentos reunidos produzem o que chamamos de Método de Escolha Otimizada de Parâmetros (MEOP). Além disso, como veremos a seguir, nosso método é uma generalização de alguns dos MPI encontrados na literatura.

O Algoritmo 3.1 resume o MEOP proposto neste capítulo. A escolha do ponto inicial e do

critério de parada, serão descritas no Capítulo 5, enquanto os resultados numéricos obtidos estão no Capítulo 6.

---

**Algoritmo 3.1** Método de Escolha Otimizada de Parâmetros.

---

- 1: **Função** RESOLVELP( $A, b, c$ )
  - 2:      $k \leftarrow 0$
  - 3:      $(x^0, y^0, z^0) \leftarrow \text{PONTOINICIAL}(A, b, c)$ . ▷ Assegure que  $(x^0, z^0) > 0$ .
  - 4:     **Repita**
  - 5:         Resolva (3.6) e encontre  $((\Delta x^{\text{af}})^k, (\Delta y^{\text{af}})^k, (\Delta z^{\text{af}})^k)$ .
  - 6:         Encontre  $((\Delta x^\mu)^k, (\Delta y^\mu)^k, (\Delta z^\mu)^k)$  e  $((\Delta x^\sigma)^k, (\Delta y^\sigma)^k, (\Delta z^\sigma)^k)$  utilizando (3.18).
  - 7:         Calcule os coeficientes de  $\hat{\varphi}(\alpha, \mu, \sigma)$  usando (3.36)
  - 8:         Calcule os coeficientes de  $g_L(\alpha, \mu, \sigma)$  e  $g_C^i(\alpha, \mu, \sigma)$  utilizando (3.45).
  - 9:         Encontre  $(\alpha_k, \mu_k, \sigma_k)$  resolvendo o subproblema de otimização global (3.46).
  - 10:        Garanta que  $\alpha_k \in (0, 1)$  tal que  $(x^{k+1}, z^{k+1}) > 0$  usando (3.29) e faça
 
$$\begin{aligned} x^{k+1} &= x^k + \alpha_k((\Delta x^{\text{af}})^k) + \mu_k(\Delta x^\mu)^k + \sigma_k(\Delta x^\sigma)^k \\ y^{k+1} &= y^k + \alpha_k((\Delta y^{\text{af}})^k) + \mu_k(\Delta y^\mu)^k + \sigma_k(\Delta y^\sigma)^k. \\ z^{k+1} &= z^k + \alpha_k((\Delta z^{\text{af}})^k) + \mu_k(\Delta z^\mu)^k + \sigma_k(\Delta z^\sigma)^k \end{aligned}$$
  - 11:         $k \leftarrow k + 1$
  - 12:     **Até** O critério de parada ser satisfeito.
  - 13: **Fim Função**
- 

A explicação de cada passo do Algoritmo 3.1 será feita a seguir – a menos dos passos 7 e 12, através da explicitação do MEOP. Para dar conta disso e, portanto, dos objetivos propostos no presente capítulo, são necessárias algumas transformações, as quais explanamos a seguir.

Considere novamente o par primal-dual (2.1-2.2) na sua forma padrão, i.e.,

$$\begin{aligned} \min_x \quad & c^T x \\ \text{sujeito a} \quad & \begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{aligned} \tag{P}$$

e

$$\begin{array}{ll} \max_{(y,z)} & b^T y \\ \text{sujeito a} & \begin{cases} A^T y + z = c \\ z \geq 0, y \text{ livre} \end{cases} \end{array} \quad (D)$$

Como visto no Capítulo 2, as condições KKT para este problema são<sup>1</sup>

$$\begin{cases} Ax = b, & (3.1a) \\ A^T y + z = c, & (3.1b) \\ xz = 0, & (3.1c) \\ (x, z) \geq 0. & (3.1d) \end{cases}$$

Com isso, dado qualquer  $w = (x, y, z)$ , os vetores dos resíduos de (3.1),  $r_P, r_D$  e  $r_C$ , podem ser definidos como

$$r_P = Ax - b, \quad (3.2a)$$

$$r_D = A^T y + z - c, \quad (3.2b)$$

$$r_C = xz. \quad (3.2c)$$

Seja  $(x^0, y^0, z^0)$  um ponto inicial que pertença a  $\mathcal{Q}^+$ , isto é, um ponto inicial com as componentes de  $x$  e de  $z$  positivas. Então

$$r_P^0 = Ax^0 - b,$$

$$r_D^0 = A^T y^0 + z^0 - c,$$

$$r_C^0 = x^0 z^0 > 0.$$

---

<sup>1</sup>Novamente, durante todo este capítulo, estamos utilizando a notação do produto de Haddamard e assim,  $uv$  indica o produto componente a componente de  $u$  e  $v$ .

No método proposto, vamos garantir que os resíduos primais e duais iniciais sejam não-negativos – veja Observação 3.8 na página 68. Escolher um  $(x^0, y^0, z^0)$  que gere resíduos iniciais desse tipo pode ser uma tarefa difícil. Assim, a estratégia adotada é definir  $H_P$  e  $H_D$ , matrizes diagonais, tais que cada entrada de suas diagonais seja formada segundo as seguintes regras:

$$(H_P)_i = \begin{cases} 1, & \text{se } (r_P^0)_i \geq 0 \\ -1, & \text{se } (r_P^0)_i < 0 \end{cases}, \quad (3.3a)$$

para  $i = 1, \dots, m$ , e

$$(H_D)_j = \begin{cases} 1, & \text{se } (r_D^0)_j \geq 0 \\ -1, & \text{se } (r_D^0)_j < 0 \end{cases}, \quad (3.3b)$$

para  $j = 1, \dots, n$ .

Mais que isso, como  $H_D$  e  $H_P$  são matrizes de posto completo, o conjunto solução de (3.1) e do sistema

$$\begin{cases} H_P(Ax - b) = 0, & (3.4a) \end{cases}$$

$$\begin{cases} H_D(A^T y + z - c) = 0, & (3.4b) \end{cases}$$

$$\begin{cases} xz = 0, & (3.4c) \end{cases}$$

$$\begin{cases} (x, z) \geq 0, & (3.4d) \end{cases}$$

é o mesmo. De fato, note que apenas estamos multiplicando cada linha do sistema KKT original por um escalar. Além disso, garantimos que

$$H_P(r_P^0) \geq 0 \text{ e } H_D(r_D^0) \geq 0.$$

Tal era o objetivo principal dessas transformações.

A otimalidade do sistema (3.1) – bem como de (3.4) – é alcançada quando todos os resíduos forem nulos. De fato, para fins de implementação, diremos que a otimalidade foi alcançada sempre que a norma de cada um desses resíduos for menor ou igual a uma tolerância pré-estabelecida. Assim propomos um método para resolver o sistema KKT escalado que resolva aproximadamente, em cada iteração, para  $(x, y, z) \in \mathcal{Q}^+$  e algum  $\mu > 0$ , o sistema

$$\left\{ \begin{array}{l} H_P(Ax - b) = 0, \\ H_D(A^T y + z - c) = 0, \\ xz = \mu e, \\ (x, z) > 0. \end{array} \right. \quad \begin{array}{l} (3.5a) \\ (3.5b) \\ (3.5c) \\ (3.5d) \end{array}$$

*Observação 3.1 (Notação).* De acordo com o contexto, considerando que  $k$  é a iteração atual, podemos nos abster de indicar a iteração de um vetor ou variável qualquer  $t$ , isto é,  $t^k = t$ . Neste caso, utilizamos do símbolo circunflexo  $\hat{\phantom{t}}$  sobre um vetor ou variável  $t$  qualquer em questão para indicar seu valor na iteração  $k + 1$ , isto é,  $\hat{t} = t^{k+1}$ .

## 3.1 Direções de busca

### 3.1.1 Direção Afim-escala

Como consequência da Seção 2.1.1, a direção afim-escala  $(\Delta w^{\text{af}})^k$  que resolve aproximadamente o sistema (3.1) é encontrada através na solução do sistema não linear

$$\left\{ \begin{array}{l} A(\Delta x^{\text{af}})^k + r_P^k = 0, \\ A^T(\Delta y^{\text{af}})^k + (\Delta z^{\text{af}})^k + r_D^k = 0, \\ z^k(\Delta x^{\text{af}})^k + x^k(\Delta z^{\text{af}})^k + r_C^k = 0. \end{array} \right. \quad \begin{array}{l} (3.6a) \\ (3.6b) \\ (3.6c) \end{array}$$

Deixando de utilizar o índice da iteração, para simplificar, podemos encontrar, por meio da equação (3.6c), o valor de  $\Delta z^{\text{af}}$  como

$$\Delta z^{\text{af}} = -X^{-1}(z\Delta x^{\text{af}} + r_C). \quad (3.7)$$

Se esta equação for usada e, além disso,  $\Delta z^{\text{af}}$  for substituído na equação (3.6b), obtemos o sistema

$$\begin{cases} A\Delta x^{\text{af}} + r_P = 0 \\ A^T \Delta y^{\text{af}} - X^{-1}z\Delta x^{\text{af}} - X^{-1}r_C + r_D = 0 \end{cases}.$$

Este sistema é chamado *Sistema Aumentado* e tem sua forma matricial dada por

$$\begin{bmatrix} -D^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x^{\text{af}} \\ \Delta y^{\text{af}} \end{bmatrix} = \begin{bmatrix} -r_D + X^{-1}r_C \\ -r_P \end{bmatrix},$$

em que  $D = XZ^{-1}$  é não-singular pois  $(x, z) > 0$ . Mais que isso,  $\Delta x^{\text{af}}$  pode ser encontrado como

$$\begin{aligned} \Delta x^{\text{af}} &= -D(-r_D + X^{-1}r_C - A^T \Delta y^{\text{af}}) \\ &= D(r_D - X^{-1}r_C + A^T \Delta y^{\text{af}}) \\ &= D(A^T \Delta y^{\text{af}} - t), \end{aligned} \quad (3.8)$$

em que

$$\begin{aligned} t &= X^{-1}r_C - r_D \\ &= (X^{-1})xz - A^T y - z + c \\ &= z - A^T y - z + c \\ &= -(A^T y - c). \end{aligned}$$

Usando a fórmula para  $\Delta x^{\text{af}}$ , a equação (3.6a) torna-se

$$AD(A^T \Delta y^{\text{af}} - t) + r_P = 0$$

e logo

$$\Delta y^{\text{af}} = B(ADt - r_P), \quad (3.9)$$

em que

$$B = (ADA^T)^{-1} \quad (3.10)$$

e portanto  $B^{-1}$  é sempre simétrica definida positiva, já que  $A$  tem posto completo. Consequentemente, para encontrar  $\Delta w^{\text{af}}$  usamos uma fatoração de Cholesky de  $B^{-1}$  e uma retrossubstituição.

### 3.1.2 A direção ideal

O método proposto aqui seguirá a estrutura dos métodos preditores-corretores (vide Seção 2.2). Suponha que, dados o ponto  $w = (x, y, z)$  e um parâmetro  $\mu > 0$ , seja possível encontrar  $\hat{w}$ , solução do sistema

$$\begin{cases} A\hat{x} - b = 0 \\ A^T \hat{y} + \hat{z} - c = 0 \\ \hat{x}\hat{z} = \mu e \end{cases}, \quad (3.11)$$

através de um único passo *ideal*  $\Delta w = (\Delta x, \Delta y, \Delta z)$ , tal que

$$\hat{w} = w + \Delta w.$$

A estratégia adotada neste trabalho para encontrar esse passo ideal é escrever  $\Delta w = \Delta w^{\text{af}} + \Delta w^c$ , em que  $\Delta w^{\text{af}}$  é a direção afim-escala e  $\Delta w^c$  é a direção corretora *ideal*.

Procedendo com as substituições, na parte linear de (3.11), temos

$$A(x + \Delta x) - b = A(x + \Delta x^{\text{af}} + \Delta x^{\text{c}}) - b = \underbrace{(Ax - b) + A\Delta x^{\text{af}} + A\Delta x^{\text{c}}}_{=0 \text{ por (3.6a)}} = A\Delta x^{\text{c}}$$

e

$$\begin{aligned} A^T(y + \Delta y) + (z + \Delta z) - c &= A^T(y + \Delta y^{\text{af}} + \Delta y^{\text{c}}) + (z + \Delta z^{\text{af}} + \Delta z^{\text{c}}) - c \\ &= \underbrace{A^T y + z - c + A^T \Delta y^{\text{af}} + \Delta z^{\text{af}}}_{=0 \text{ por (3.6b)}} + A^T \Delta y^{\text{c}} + \Delta z^{\text{c}} \\ &= A^T \Delta y^{\text{c}} + \Delta z^{\text{c}}. \end{aligned}$$

Por outro lado, para a parte da complementaridade de (3.11), temos que

$$\begin{aligned} \hat{x}\hat{z} &= (x + \Delta x)(z + \Delta z) = xz + x\Delta z + z\Delta x + \Delta x\Delta z \\ &= \underbrace{xz + x\Delta z^{\text{af}} + z\Delta x^{\text{af}}}_{=0 \text{ por (3.6c)}} + z\Delta x^{\text{c}} + x\Delta z^{\text{c}} + \Delta x\Delta z \\ &= x\Delta z^{\text{c}} + z\Delta x^{\text{c}} + \Delta x\Delta z. \end{aligned}$$

Usando tais simplificações, obtemos o seguinte sistema não linear

$$\begin{cases} A\Delta x^{\text{c}} = 0 \\ A^T \Delta y^{\text{c}} + \Delta z^{\text{c}} = 0 \\ x\Delta z^{\text{c}} + z\Delta x^{\text{c}} + \Delta x\Delta z = \mu e \end{cases} . \quad (3.12)$$

O vetor  $\Delta x\Delta z$  corresponde a uma correção de segunda ordem, nos moldes dos trabalhos de Gondzio [25] e Mehrotra [49].

A proposta deste trabalho é utilizar uma forma de generalização das correções de ordem

superior usadas por esses autores. Essa generalização será feita supondo que para algum escalar  $\sigma$  positivo e limitado a aproximação

$$\Delta x \Delta z \approx \sigma \Delta x^{\text{af}} \Delta z^{\text{af}} \quad (3.13)$$

seja aceitável. Estamos aqui fazendo um abuso de notação, ao utilizar o símbolo  $\approx$ . O que queremos dizer é que, estamos usando como hipótese que existe  $\sigma$  tal que  $\|\Delta x \Delta z - \sigma \Delta x^{\text{af}} \Delta z^{\text{af}}\|$  seja suficientemente pequena.

Em particular, note que se  $\sigma = 1$  e  $\mu = \tau_{\text{af}}^3 / \tau$  – em que  $\tau_{\text{af}}$  é dado na equação (2.23) e  $\tau = x^T z / n$  – temos o método de Mehrotra [49]. Por outro lado, para Gondzio [25],  $\mu$  é escolhido como no método de Mehrotra e  $\Delta X \Delta z$  é várias vezes aproximado por direções que projetem, componente a componente, a complementaridade na vizinhança  $N_s(\gamma)$ . Caso  $\mu = 0$  e  $\sigma = 1$ , e o ponto inicial for factível, temos o método de Monteiro, Adler e Resende [52].

De fato, a escolha do valor  $\mu$  e a utilização de correções de ordem superior definem os diferentes tipos de MPI que têm sido utilizados atualmente [77]. Isso significa que de certa forma, nosso método generaliza alguns dos mais conhecidos MPI existentes.

### 3.1.3 Combinando direções

Alguns autores combinam direções de correção utilizando pesos para essas direções. Colombo e Gondzio [9] generalizam e estendem o trabalho de Gondzio [25], fazendo com que a combinação das múltiplas correções tenha um peso, que é escolhido fazendo uma busca linear. Já Jarre e Wechs [38] propõem um subproblema linear que é resolvido a cada iteração e cuja solução determina os pesos que as direções de correções de ordem superior terão na direção final.

Villas-Bôas e Perin [73], também no contexto primal-dual, estudam algumas vantagens de adiar a escolha do parâmetro de barreira e do tamanho do passo. Estes autores mostram que o próximo iterado pode ser expresso por uma função quadrática do parâmetro de barreira, bem

como constatam que tal parametrização é útil para garantir tanto a não negatividade do próximo iterado quanto a proximidade destes da trajetória central.

Uma das principais contribuições do nosso trabalho, através da extensão das ideias de Villas-Bôas e Perin [73], é a escolha do valor de  $\mu$  – que aqui faz as vezes de parâmetro que determina a trajetória central – e de  $\sigma$  – que vai pesar a contribuição da correção – sendo feita de maneira adiada. Por isso tais parâmetros, bem como o tamanho do passo  $\alpha$ , estão sendo tratados como *variáveis*. Para a escolha desses parâmetros, faremos uso de uma função de mérito, a qual será construída a seguir.

Primeiramente, note que a aproximação dada em (3.13) transforma o sistema não linear (3.12) no sistema linear

$$\begin{cases} A\Delta x^c = 0 \\ A^T \Delta y^c + \Delta z^c = 0 \\ x\Delta z^c + z\Delta x^c = \mu e - \sigma \Delta x^{\text{af}} \Delta z^{\text{af}} \end{cases} . \quad (3.14)$$

Expressando tal sistema na forma matricial, temos

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^c \\ \Delta y^c \\ \Delta z^c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e - \sigma \Delta x^{\text{af}} \Delta z^{\text{af}} \end{bmatrix} . \quad (3.15)$$

As matrizes do lado esquerdo das equações em (3.6) e (3.15) são as mesmas. Com isso, podemos resolver (3.15) utilizando a mesma fatoração de Cholesky de  $B^{-1}$  – veja a Equação (3.10).

De fato, o lado direito de (3.15) pode ser reescrito como

$$\mu \begin{bmatrix} 0 \\ 0 \\ e \end{bmatrix} + \sigma \begin{bmatrix} 0 \\ 0 \\ -\Delta x^{\text{af}} \Delta z^{\text{af}} \end{bmatrix} . \quad (3.16)$$

Portanto, para  $\mu, \sigma$  quaisquer – ainda não escolhidos –, é possível encontrar a direção corretora  $\Delta w^c$  como

$$\Delta w^c = \mu \Delta w^\mu + \sigma \Delta w^\sigma, \quad (3.17)$$

resolvendo então dois sistemas lineares similares ao utilizado na direção afim (vide Seção 3.1.1).

Com efeito, encontramos  $\Delta w^\mu$  e  $\Delta w^\sigma$  usando a mesma abordagem da referida seção, i.e., resolvendo os sistemas  $\nabla F(w) \Delta w^\mu = (0, 0, e)$  e  $\nabla F(w) \Delta w^\sigma = (0, 0, -\Delta x^{\text{af}} \Delta z^{\text{af}})$ .

Explicitamente, temos

$$\begin{aligned} \Delta y^\mu &= B(ADt^\mu) & \Delta y^\sigma &= B(ADt^\sigma) \\ \Delta x^\mu &= D(A^T \Delta y^\mu - t^\mu) & \Delta x^\sigma &= D(A^T \Delta y^\sigma - t^\sigma), & (3.18) \\ \Delta z^\mu &= -X^{-1}(z \Delta x^\mu - e) & \Delta z^\sigma &= -X^{-1}(z \Delta x^\sigma + \Delta x^{\text{af}} \Delta z^{\text{af}}) \end{aligned}$$

em que

$$t^\mu = -X^{-1}e \quad \text{e} \quad t^\sigma = (X^{-1}) \Delta x^{\text{af}} \Delta z^{\text{af}}.$$

Desta forma, o maior esforço computacional por iteração do método proposto consiste em uma fatoração de Cholesky da matriz  $B^{-1}$ , seguido de três retrosubstituições.

Definimos o próximo ponto, para cada uma das variáveis, como sendo

$$\hat{w} = w + \alpha(\Delta w^{\text{af}} + \Delta w^c),$$

ou ainda, expandindo para cada variável bem como separando as direções, na forma

$$\hat{x} = x + \alpha(\Delta x^{\text{af}} + \mu \Delta x^\mu + \sigma \Delta x^\sigma), \quad (3.19a)$$

$$\hat{y} = y + \alpha(\Delta y^{\text{af}} + \mu \Delta y^\mu + \sigma \Delta y^\sigma), \quad (3.19b)$$

$$\hat{z} = z + \alpha(\Delta z^{\text{af}} + \mu \Delta z^\mu + \sigma \Delta z^\sigma). \quad (3.19c)$$

Até o presente momento, a tripla  $(\alpha, \mu, \sigma)$ , em que  $\alpha$  é o tamanho do passo, ainda não foi escolhida. Para de fato realizar o passo indicado acima, a abordagem proposta neste trabalho é tratar algebricamente  $(\alpha, \mu, \sigma)$  como uma tripla de variáveis reais, utilizar no máximo três retrosubstituições para escolhê-las e finalmente usar a combinação linear das direções  $\Delta w^{\text{af}}$ ,  $\Delta w^{\mu}$  e  $\Delta w^{\sigma}$ , com  $\alpha, \mu$  e  $\sigma$  determinando as constantes – ou pesos – de tal combinação.

A função de mérito, que será apresentada em sequência, será formulada a partir dos resíduos do sistema KKT escalado (3.4). Para tanto, na próxima Seção será mostrado que é possível prever, a depender de uma escolha de  $(\alpha, \mu, \sigma)$ , o próximo resíduo deste sistema.

## 3.2 O próximo resíduo

**Definição 3.2.** Definimos  $\rho$ , *vetor de resíduos do sistema KKT escalado* (3.4) para um ponto  $(x, y, z)$ , como

$$\rho(x, y, z) = \begin{cases} \rho_P(x, y, z) = H_P(Ax - b) \\ \rho_D(x, y, z) = H_D(A^T y + z - c) \\ \rho_C(x, y, z) = xz \end{cases} \quad . \quad (3.20)$$

Além disso, seja  $\rho_L = (\rho_P, \rho_D)^T \in \mathbb{R}^{m+n}$  o *vetor dos resíduos da parte linear* do sistema KKT escalado. Definimos o *vetor dos resíduos na iteração k* como  $\rho^k$ . Por construção  $\rho^0 > 0$  para  $(x^0, y^0, z^0)$ .

Usando a Definição 3.2, dado um ponto  $(x, y, z)$ , podemos prever o resíduo  $\hat{\rho}$ , para o próximo ponto  $(\hat{x}, \hat{y}, \hat{z})$ , a depender de uma escolha de  $(\alpha, \mu, \sigma)$ , através do Teorema 3.3.

**Teorema 3.3.** *O próximo resíduo para o sistema KKT escalado (3.4) é escrito, em termos da*

tripla  $(\alpha, \mu, \sigma)$ , como

$$\hat{\rho}(\alpha, \mu, \sigma) = \begin{bmatrix} (1 - \alpha)\rho_L \\ (1 - \alpha)\rho_C + \alpha\mu e + \alpha(\alpha - \sigma)L_{0,0} + \alpha^2\Lambda(\mu, \sigma) \end{bmatrix} \quad (3.21)$$

em que

$$\Lambda(\mu, \sigma) = \mu^2 L_{2,0} + \mu L_{1,0} + \mu\sigma L_{1,1} + \sigma^2 L_{0,2} + \sigma L_{0,1}, \quad (3.22)$$

e

$$\begin{aligned} L_{0,0} &= \Delta x^{af} \Delta z^{af}, & L_{1,1} &= \Delta x^\mu \Delta z^\sigma + \Delta x^\sigma \Delta z^\mu, \\ L_{1,0} &= \Delta x^{af} \Delta z^\mu + \Delta z^{af} \Delta x^\mu, & L_{0,1} &= \Delta x^{af} \Delta z^\sigma + \Delta z^{af} \Delta x^\sigma, \\ L_{2,0} &= \Delta x^\mu \Delta z^\mu, & L_{0,2} &= \Delta x^\sigma \Delta z^\sigma. \end{aligned} \quad (3.23)$$

O Teorema 3.3 é consequência direta da aplicação dos Lemas 3.4 e 3.5, que consideram a parte linear e não linear de  $\hat{\rho}$  de forma separada. Esses lemas e suas demonstrações são encontrados na sequência.

**Lema 3.4.** *O resíduo da parte linear de (3.4) para o próximo iterado é escrito como*

$$\hat{\rho}_L(\alpha, \mu, \sigma) = (1 - \alpha)\rho_L.$$

*Demonstração.* Utilizando (3.19) na equação (3.4a), temos

$$\begin{aligned} \hat{\rho}_P &= H_P(A\hat{x} - b) = H_P(A(x + \alpha(\Delta x^{af} + \Delta x^c)) - b) \\ &= H_P(A(x + \alpha\Delta x^{af}) - b) + \alpha H_P A \Delta x^c. \end{aligned}$$

Como, de (3.6),  $A\Delta x^{\text{af}} = b - Ax$ , vale a seguinte igualdade:

$$\begin{aligned} H_P(A(x + \alpha\Delta x^{\text{af}}) - b) &= H_P((1 - \alpha)A\Delta x^{\text{af}}) \\ &= (1 - \alpha) \underbrace{H_P(Ax - b)}_{\rho_P} \\ &= (1 - \alpha)\rho_P. \end{aligned}$$

Por outro lado, pela equação (3.12),  $A\Delta x^c = 0$ . Logo  $\hat{\rho}_P = (1 - \alpha)\rho_P$ .

A prova para parte dual da factibilidade é similar. Com efeito, de (3.4b) e (3.19), segue

$$\begin{aligned} \hat{\rho}_D &= H_D(A^T\hat{y} + \hat{z} - c) \\ &= H_D \left[ A^T \left( y + \alpha(\Delta y^{\text{af}} + \Delta y^c) \right) + \left( z + \alpha(\Delta z^{\text{af}} + \Delta z^c) \right) - c \right] \\ &= H_D \left[ A^T y + z - c + \alpha(A^T \Delta y^{\text{af}} + \Delta z^{\text{af}}) \right] + \underbrace{H_D \left( A^T \Delta y^c + \Delta z^c \right)}_{=0 \text{ por (3.12)}}. \end{aligned} \quad (3.24)$$

Como, de (3.6b),  $A^T \Delta y^{\text{af}} + \Delta z^{\text{af}} = c - A^T y - z$ , a equação (3.24) torna-se

$$H_D \left[ A^T y + z - c + \alpha(A^T \Delta y^{\text{af}} + \Delta z^{\text{af}}) \right] = (1 - \alpha)H_D(A^T y + z - c) = (1 - \alpha)\rho_D.$$

Portanto,  $\hat{\rho}_D = (1 - \alpha)\rho_D$ . ■

**Lema 3.5.** *O resíduo da parte da complementaridade de (3.4), para o próximo iterado é escrito como*

$$\hat{\rho}_C = (1 - \alpha)\rho_C + \alpha\mu e + \alpha(\alpha - \sigma)L_{0,0} + \alpha^2\Lambda(\mu, \sigma). \quad (3.25)$$

em que  $\Lambda(\mu, \sigma)$  é definida por (3.22) os vetores  $L_{i,j}$ ,  $i, j \in \{0, 1, 2\}$  são definidos por (3.23).

*Demonstração.* Para encontrar o resíduo da parte complementar para o próximo iterado, consi-

dere que podemos escrever  $\hat{\rho}_C = \hat{x}\hat{z}$ . Então, de (3.19),

$$\begin{aligned}\hat{x}\hat{z} &= \left(x + \alpha(\Delta x^{\text{af}} + \Delta x^c)\right) \left(z + \alpha(\Delta z^{\text{af}} + \Delta z^c)\right) \\ &= \alpha^2(\Delta x^{\text{af}} + \Delta x^c)(\Delta z^{\text{af}} + \Delta z^c) + \alpha \left[(x\Delta z^{\text{af}} + z\Delta x^{\text{af}}) + (x\Delta z^c + z\Delta x^c)\right] + xz. \quad (3.26)\end{aligned}$$

Note que  $(x\Delta z^{\text{af}} + z\Delta x^{\text{af}}) = -xz$  por (3.6). Além disso usando a equação (3.17), claramente  $x\Delta z^\mu + z\Delta x^\mu = e$  e  $x\Delta z^\sigma + z\Delta x^\sigma = -\Delta x^{\text{af}}\Delta z^{\text{af}}$ . Então, de (3.15),  $(x\Delta z^c + z\Delta x^c) = \mu e - \sigma\Delta x^{\text{af}}\Delta z^{\text{af}}$ .

Assim, a equação (3.26) torna-se

$$\begin{aligned}\hat{x}\hat{z} &= \alpha^2(\Delta x^{\text{af}} + \Delta x^c)(\Delta z^{\text{af}} + \Delta z^c) + \alpha(-xz + \mu e - \sigma\Delta x^{\text{af}}\Delta z^{\text{af}}) + xz \\ &= t_1 + t_2,\end{aligned}$$

em que

$$t_1 = (1 - \alpha)xz + \alpha\mu e + \alpha(\alpha - \sigma)\Delta x^{\text{af}}\Delta z^{\text{af}} \quad (3.27a)$$

e

$$t_2 = \alpha^2 \left(\Delta x^{\text{af}}\Delta z^c + \Delta z^{\text{af}}\Delta x^c + \Delta x^c\Delta z^c\right) \quad (3.27b)$$

são variáveis locais.

Como  $\Delta w^c = \mu\Delta w^\mu + \sigma\Delta w^\sigma$ , então

$$\begin{aligned}\Delta x^{\text{af}}\Delta z^c &= \mu\Delta x^{\text{af}}\Delta z^\mu + \sigma\Delta x^{\text{af}}\Delta z^\sigma, \\ \Delta z^{\text{af}}\Delta x^c &= \mu\Delta z^{\text{af}}\Delta x^\mu + \sigma\Delta z^{\text{af}}\Delta x^\sigma,\end{aligned}$$

e

$$\begin{aligned}\Delta x^c \Delta z^c &= (\mu \Delta x^\mu + \sigma \Delta x^\sigma)(\mu \Delta z^\mu + \sigma \Delta z^\sigma) \\ &= \mu^2 \Delta x^\mu \Delta z^\mu + \mu \sigma (\Delta x^\mu \Delta z^\sigma + \Delta x^\sigma \Delta z^\mu) + \sigma^2 \Delta x^\sigma \Delta z^\sigma.\end{aligned}$$

Consequentemente  $t_2$ , dado em (3.27b), pode ser reescrito como

$$\begin{aligned}t_2 &= \alpha^2 \left( \mu^2 \Delta x^\mu \Delta z^\mu + \mu \sigma (\Delta x^\mu \Delta z^\sigma + \Delta x^\sigma \Delta z^\mu) + \right. \\ &\quad \left. + \mu (\Delta x^{\text{af}} \Delta z^\mu + \Delta z^{\text{af}} \Delta x^\mu) + \sigma (\Delta x^{\text{af}} \Delta z^\sigma + \Delta z^{\text{af}} \Delta x^\sigma) + \sigma^2 \Delta x^\sigma \Delta z^\sigma \right). \quad (3.28)\end{aligned}$$

Basta agora usar (3.27a) e (3.28) para obter  $\hat{x}\hat{z}$ .

Definindo os vetores  $\Lambda(\mu, \sigma)$  como em (3.22) e  $L_{i,j}$  como em (3.23) e substituindo onde for necessário, finalmente encontramos a equação (3.25), levando em conta que  $\rho_C = xz$ . ■

O vetor  $\hat{\rho}(\alpha, \mu, \sigma) \in \mathbb{R}^q$ , em que  $q = m + 2n$  é precisamente o número de linhas de (3.4). Além disso, uma consequência do Teorema 3.3 é que todos os resíduos permanecem não-negativos, se nosso ponto inicial é interior. Formalmente temos o seguinte Corolário.

**Corolário 3.6.** *Em cada iteração  $k$ , para  $\alpha \in (0, 1]$  e  $(\mu, \sigma) > 0$ ,  $\rho^k(\alpha, \mu, \sigma) \geq 0$ .*

*Demonstração.* Para a parte linear de (3.4), pelo Lema 3.4,  $\hat{\rho}_L(\alpha, \mu, \sigma) = (1 - \alpha)\rho_L$ . Como garantimos que,  $\rho^0 \geq 0$ ,  $0 \leq \alpha \leq 1$ , por indução, o corolário se verifica.

A parte da complementaridade, por construção, é positiva, já que  $(x, y, z) \in \mathcal{Q}^+$ . Além disso, faremos a cada iteração o teste da razão, isto é, escolheremos  $\alpha_k$ , tal que

$$\alpha_k < \tilde{\alpha}_k = \frac{-1}{\min \left\{ (X^k)^{-1} \Delta x^k, (Z^k)^{-1} \Delta z^k, -1 \right\}}. \quad (3.29)$$

Logo,  $\rho_C^k > 0$  para todo  $k$  e o corolário é válido. ■

### 3.3 Um polinômio como função de mérito

Em geral, funções de mérito para MPI servem para dar uma medida de quão próximo se está da solução do problema, embora nem sempre elas sejam usadas como critério de parada do algoritmo.

Zhang [82], por exemplo, escolhe o tamanho de passo de seu algoritmo tal que a complementaridade e a infactibilidade primal-dual sejam reduzidas. Para isso, este autor define sua função de mérito como a soma do *gap* de complementaridade e da norma-2 do resíduo primal-dual, i.e.,

$$\phi(x, y, z) = \|(r_P, r_D)\| + x^T z$$

e escolhe  $\alpha$  tal que essa função seja minimizada. Nada obstante, essa função é utilizada apenas de maneira teórica, para demonstrar a convergência do método.

Do ponto de vista prático, o PCx [11] utiliza uma função de mérito para detectar se o problema é infactível ou se a solução é desconhecida ou sub-ótima. Nesse caso, a função é dada por

$$\phi(x, y, z) = \frac{\|r_P\|}{\max\{1, \|b\|\}} + \frac{\|r_D\|}{\max\{1, \|c\|\}} + \frac{c^T x - b^T y}{\max\{1, \|b\|, \|c\|\}}.$$

Note que essa função de mérito tem relação com o próprio critério de parada da implementação, como visto em (2.32).

Neste trabalho, será definida uma função de mérito que não só sirva como medida de complementaridade e factibilidade, mas que também sirva de guia para a escolha do próximo ponto. De maneira similar aos autores acima citados, serão usadas medidas da infactibilidade e do *gap* de complementaridade.

**Definição 3.7** (Função de Mérito). Definimos a *função de mérito* de um ponto  $(x, y, z)$  como

$$\varphi(x, y, z) = \frac{1}{m+n} \|\rho_L\|_1 + \frac{x^T z}{n}, \quad (3.30)$$

em que  $\rho_L = (\rho_P, \rho_D)$  é dado por (3.20) no ponto  $(x, y, z)$ .

*Observação 3.8.* A Definição 3.7 é uma das possíveis maneiras de medir quão perto da solução está um ponto  $(x, y, z)$ . Em particular, esta escolha possui algumas propriedades que serão exploradas para escolhermos a tripla  $(\alpha, \mu, \sigma)$ , a saber:

- (i) A utilização das matrizes  $H_P$  e  $H_D$  e o Corolário 3.6 garantem que dado  $(x, y, z)$  calculado pelo método e que esteja em  $\mathcal{Q}^+$ , o sinal de cada linha de  $\rho_L$  permanece igual em todas as iterações – de fato, determinamos que tais linhas sejam não-negativas. Neste caso, temos certeza de que para cada linha  $i$  de  $\rho_L$ ,  $\rho_L(x, y, z)_i \geq 0$ . Portanto, podemos descartar o módulo na utilização da norma-1 de  $\rho_L$ , isto é,

$$\frac{1}{m+n} \|\rho_L\|_1 = \frac{1}{m+n} \sum_{\ell=1}^{m+n} (\rho_L)_\ell.$$

Tal escolha permite transformar a função de mérito em um polinômio nas variáveis  $(\alpha, \mu, \sigma)$ , conforme veremos a seguir;

- (ii) Podemos rescrever o *gap* de complementaridade como

$$x^T z = \sum_{i=1}^n x_i z_i = \sum_{i=1}^n (\rho_C)_i;$$

- (iii) A média do *gap* de complementaridade,  $x^T z/n$ , é usada em geral como parte determinante do valor para o parâmetro de barreira (vide Seção 2.1.2). Em uma solução ótima, este valor é nulo;

- (iv) Também devido ao Corolário 3.6, dado  $(x, y, z) \in \mathcal{Q}^+$  gerado pelo método, vale

$$\varphi(x, y, z) \geq 0;$$

- (v) Finalmente, se  $(x^*, y^*, z^*)$  é solução de (3.4), então  $\varphi(x^*, y^*, z^*) = 0$ .

Por conta dos itens (i) e (ii) da Observação 3.8, podemos rescrever (3.30) como

$$\varphi(x, y, z) = \frac{1}{m+n} \sum_{\ell=1}^{m+n} (\rho_L)_\ell + \frac{1}{n} \sum_{i=1}^n (\rho_C)_i. \quad (3.31)$$

Para fins de notação, definimos o *operador média* para qualquer vetor  $v \in \mathbb{R}^p$  como

$$\bar{v} = \frac{1}{p} \sum_{i=1}^p v_i.$$

Este operador encontra a média aritmética das componentes do vetor em questão.

Com esta notação, podemos representar a função de mérito (3.31) como

$$\varphi(x, y, z) = \overline{\rho_L} + \overline{\rho_C}. \quad (3.32)$$

Ao usarmos a Definição 3.7 em conjunto com o Teorema 3.3, poderemos encontrar o valor da função de mérito para o próximo ponto  $(\hat{x}, \hat{y}, \hat{z})$ , denotando-a como  $\hat{\varphi}$ , a depender de uma escolha de  $(\alpha, \mu, \sigma)$ . Com efeito, o Teorema 3.9 a seguir nos dá a expressão algébrica de  $\hat{\varphi}$ .

**Teorema 3.9.** *A função de mérito para um ponto  $(\hat{x}, \hat{y}, \hat{z})$  da próxima iteração do Algoritmo 3.1 pode ser escrita como o polinômio  $\hat{\varphi} : \mathbb{R}^3 \rightarrow \mathbb{R}$  a depender das variáveis  $(\alpha, \mu, \sigma)$ , com a seguinte expressão:*

$$\hat{\varphi}(\alpha, \mu, \sigma) = (1 - \alpha)(\overline{\rho_L} + \overline{\rho_C}) + \alpha\mu + \alpha(\alpha - \sigma)\overline{L_{0,0}} + \alpha^2\overline{\Lambda(\mu, \sigma)}, \quad (3.33)$$

em que

$$\overline{\Lambda(\mu, \sigma)} = \mu^2\overline{L_{2,0}} + \mu\overline{L_{1,0}} + \mu\sigma\overline{L_{1,1}} + \sigma^2\overline{L_{0,2}} + \sigma\overline{L_{0,1}}. \quad (3.34)$$

*Demonstração.* Por conta da Equação (3.21), sabemos a expressão do próximo resíduo, a depen-

der de uma escolha de  $(\alpha, \mu, \sigma)$ . Neste caso, a função de mérito preditiva será

$$\hat{\varphi}(\alpha, \mu, \sigma) = \overline{\hat{\rho}_L}(\alpha, \mu, \sigma) + \overline{\hat{\rho}_C}(\alpha, \mu, \sigma). \quad (3.35)$$

Aplicando diretamente o Teorema 3.3 na Equação (3.35) e utilizando a definição do operador média de vetor, temos que

$$\begin{aligned} \hat{\varphi}(\alpha, \mu, \sigma) &= \frac{1}{m+n} \sum_{\ell=1}^{m+n} (\hat{\rho}_L)_\ell + \frac{1}{n} \sum_{j=1}^n (\hat{\rho}_C)_j \\ &= \frac{1}{m+n} \sum_{\ell=1}^{m+n} [(1-\alpha)(\rho_L)_\ell] + \\ &\quad + \frac{1}{n} \sum_{j=1}^n [(1-\alpha)(\rho_C)_j + \alpha\mu + \alpha(\alpha-\sigma)(L_{0,0})_j + \alpha^2\Lambda(\mu, \sigma)_j] \\ &= (1-\alpha)(\overline{\rho_L} + \overline{\rho_C}) + \alpha\mu + \alpha(\alpha-\sigma)\overline{L_{0,0}} + \alpha^2\overline{\Lambda(\mu, \sigma)}, \end{aligned}$$

o que finaliza a demonstração ■

Consideramos que a função de mérito para o próximo ponto,  $\hat{\varphi}(\alpha, \mu, \sigma)$ , bem caracteriza todas propriedades da Observação 3.8. Com isso, alcançamos o objetivo de encontrar uma função de mérito que tenha boas propriedades matemáticas – um polinômio de grau máximo 2 em cada uma das variáveis  $(\alpha, \mu, \sigma)$  – e que ao mesmo tempo dê uma medida adequada da qualidade solução.

*Observação 3.10.* É fácil ver que

$$\overline{L_{2,0}} = \frac{1}{n} \sum_{i=1}^n \Delta x_i^\mu \Delta z_i^\mu = \frac{(\Delta x^\mu)^T (\Delta z^\mu)}{n} = 0$$

e que

$$\overline{L_{0,2}} = \frac{1}{n} \sum_{i=1}^n \Delta x_i^\sigma \Delta z_i^\sigma = \frac{(\Delta x^\sigma)^T (\Delta z^\sigma)}{n} = 0.$$

Com efeito, pelas equações (3.15) e (3.16), tanto  $\Delta x^\mu$  e  $\Delta z^\mu$  quanto  $\Delta x^\sigma$  e  $\Delta z^\sigma$  são vetores

ortogonais.

Definindo os coeficientes

$$\left\{ \begin{array}{ll} a_{0,0,0} = (\overline{\rho_L} + \overline{\rho_C}) & a_{2,1,1} = \overline{L_{1,1}} \\ a_{1,0,0} = -(\overline{\rho_L} + \overline{\rho_C}) & a_{2,1,0} = \overline{L_{1,0}} \\ a_{1,1,0} = 1 & a_{2,2,0} = \overline{L_{2,0}} = 0 \\ a_{1,0,1} = -\overline{L_{0,0}} & a_{2,0,1} = \overline{L_{0,1}} \\ a_{2,0,0} = \overline{L_{0,0}} & a_{2,0,2} = \overline{L_{0,2}} = 0 \end{array} \right. \quad (3.36)$$

e adotando a convenção de que se o coeficiente  $a_{\ell,p,s}$  não está definido acima, então será nulo, podemos escrever  $\hat{\varphi}$  como

$$\hat{\varphi}(\alpha, \mu, \sigma) = \sum_{\ell=0}^2 \sum_{p=0}^2 \sum_{s=0}^2 a_{\ell,p,s} \alpha^\ell \mu^p \sigma^s,$$

ou explicitamente

$$\begin{aligned} \hat{\varphi}(\alpha, \mu, \sigma) &= a_{0,0,0} + a_{1,0,0}\alpha + a_{1,1,0}\alpha\mu + a_{1,0,1}\alpha\sigma \\ &\quad + a_{2,0,0}\alpha^2 + a_{2,1,0}\alpha^2\mu + a_{2,0,1}\alpha^2\sigma \\ &\quad + a_{2,1,1}\alpha^2\mu\sigma. \end{aligned}$$

Esses 8 coeficientes de três índices não-nulos são os únicos coeficientes que são calculados em nossa implementação, além das direções.

O método que está sendo proposto nesta tese encontra, em cada iteração, o minimizador global  $(\alpha^*, \mu^*, \sigma^*)$  para o polinômio  $\hat{\varphi}(\alpha, \mu, \sigma)$ . Como  $\hat{\varphi}$  é capaz de prever a média aritmética do próximo resíduo, esta escolha ótima permite um passo em uma direção que minimizará o resíduo  $\rho$ , pelo menos em média. Com isso, teremos um problema de otimização global a resolver em cada iteração.

### 3.4 Vizinhança como restrições polinomiais

Na Seção 2.2.2, mostramos que bons Métodos de Pontos Interiores baseiam-se na utilização de vizinhanças que fazem com que os iterados estejam a distâncias razoáveis da trajetória central. Nesta seção mostraremos que, se um ponto pertence a um conjunto viável gerado por funções polinomiais de terceiro grau nas variáveis  $(\alpha, \mu, \sigma)$ , construídas a partir da vizinhança  $\mathcal{N}_{-\infty}$ , então esse ponto também pertencerá a essa vizinhança. Com isso, tais funções serão usadas como restrições do problema de otimização global da função de mérito  $\hat{\varphi}$ , garantindo que o próximo iterado não só reduza o valor da função de mérito, mas também que o próximo ponto tenha as boas propriedades que a vizinhança  $\mathcal{N}_{-\infty}$  possui.

Vamos considerar a relação existente entre o resíduo linear escalado  $\rho_L$  na iteração  $k$  com o resíduo linear da iteração  $k + 1$ , dada no Teorema 3.3. É possível, generalizar tal relação usando a proposição seguinte que trata tanto dos resíduos lineares de KKT, quanto dos resíduos do KKT escalado.

**Proposição 3.11.** *Seja  $\{(x^k, y^k, z^k)\}$  gerada pelo Algoritmo 3.1. Então para  $k \geq 0$ , vale*

$$\begin{aligned} r_P^{k+1} &= (1 - \alpha_k)r_P^k = \nu_{k+1}r_P^0, \\ r_D^{k+1} &= (1 - \alpha_k)r_D^k = \nu_{k+1}r_D^0, \end{aligned} \tag{3.37}$$

e além disso

$$\rho_L^{k+1} = (1 - \alpha_k)\rho_L^k = \nu_{k+1}\rho_L^0, \tag{3.38}$$

em que  $\nu_0 = 1$  e

$$\nu_{k+1} = (1 - \alpha_k)\nu_k = \prod_{j=0}^k (1 - \alpha_j) \geq 0. \tag{3.39}$$

*Demonstração.* Pela definição de resíduos dada em (3.2) e pela Definição 3.2, temos que para todo  $k$ ,  $\rho_P^k = H_P r_P^k$  e  $\rho_D^k = H_D r_D^k$ .

Como  $H_P$  e  $H_D$  são não-singulares, e como  $\rho_L^k = (\rho_P^k, \rho_D^k)$ , da primeira equação de (3.21) do

Teorema 3.3, segue que  $\rho_P^k = (1 - \alpha_{k-1})\rho_P^{k-1}$  e logo  $H_P r_P^k = (1 - \alpha_{k-1})H_P r_P^{k-1}$  concluindo que  $H_P^{-1} H_P r_P^k = (1 - \alpha_{k-1})r_P^{k-1}$ , isto é,  $r_P^k = (1 - \alpha_{k-1})r_P^{k-1}$ .

Daí, por raciocínio indutivo segue que

$$\begin{aligned} r_P^k &= (1 - \alpha_{k-1})r_P^{k-1} \\ &= (1 - \alpha_{k-1}) \cdots (1 - \alpha_0)r_P^0 \\ &= \nu_k r_P^0. \end{aligned}$$

Similarmente,  $\rho_D^k = (1 - \alpha_{k-1})\rho_D^{k-1}$  e logo  $H_D r_D^k = (1 - \alpha_{k-1})H_D r_D^{k-1}$ , o que significa que  $r_D^k = (1 - \alpha_{k-1})r_D^{k-1}$ . Novamente, por raciocínio indutivo segue que

$$\begin{aligned} r_D^k &= (1 - \alpha_{k-1})r_D^{k-1} \\ &= (1 - \alpha_{k-1}) \cdots (1 - \alpha_0)r_D^0 \\ &= \nu_k r_D^0. \end{aligned}$$

Finalmente, temos que

$$\begin{aligned} \rho_L^k &= (\rho_P^k, \rho_D^k) = (H_P r_P^k, H_D r_D^k) = \\ &= (H_P \nu_k r_P^0, H_D \nu_k r_D^0) = \nu_k (H_P r_P^0, H_D r_D^0) = \nu_k (\rho_P^0, \rho_D^0) = \nu_k \rho_L^0, \end{aligned}$$

o que finaliza a demonstração. ■

Como  $\nu_k$  é uma constante não negativa, uma consequência desta proposição é que se  $r_L^0 \neq 0$ , o que implica que  $\rho_L^0 \neq 0$ , para toda norma-p, vale

$$\nu_k = \frac{\|r_L^k\|_p}{\|r_L^0\|_p} = \frac{\|\rho_L^k\|_p}{\|\rho_L^0\|_p}.$$

Logo, pela definição de *operador média*, segue que

$$\nu_k = \overline{\rho_L^k} / \overline{\rho_L^0}. \quad (3.40)$$

Caso  $\rho_L^0 = 0$ , então encontramos um ponto inicial factível e o Algoritmo resolve o problema como um método factível o faz. Na prática, porém, essa situação é rara.

No Capítulo 2, Equação (2.20), mostramos que Kojima, Megiddo e Mizuno [44], assim como Wright [77, pg. 110], utilizaram em seu método infactível a vizinhança de passo largo infactível  $\mathcal{N}_{-\infty}(\gamma, \beta)$ . Com essa definição, podemos fazer as seguintes ilações:

- $x_i z_i$  é não só o par de complementaridade, mas também o resíduo atual do sistema KKT na parte da complementaridade – vide Equação (3.2c).
- O termo

$$\tau = \frac{x^T z}{n} = \frac{1}{n} \sum_{i=1}^n x_i z_i$$

pode ser entendido como a média dos resíduos, no que diz respeito à parte complementar de KKT. Em nossa notação,  $\tau = \overline{\rho_C}$ . Além disso,  $x_i z_i = (\rho_C)_i$ .

Com tudo isso, mais a Definição 3.2 e a Proposição 3.11, é possível rescrever a vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$  para o sistema KKT escalado como

$$\mathcal{N}_{-\infty}(\gamma, \beta) = \left\{ (x, y, z) \in \mathcal{Q}^+ : \frac{\overline{\rho_L}}{\rho_L^0} \leq \beta \frac{\overline{\rho_C}}{\rho_C^0}, (\rho_C)_i \geq \gamma \overline{\rho_C}, \forall i = 1, \dots, n \right\}, \quad (3.41)$$

em que  $\gamma \in (0, 1)$  e  $\beta \geq 1$ .

Por conta de (3.40) e (3.41), segue que para qualquer ponto  $(x^k, y^k, z^k) \in \mathcal{N}_{-\infty}(\gamma, \beta)$  vale

$$\nu_k \leq \beta \overline{\rho_C^k} / \overline{\rho_C^0}. \quad (3.42)$$

Até agora, o método proposto resolve o sistema KKT escalado (3.4) e além disso usa a função

de mérito  $\hat{\varphi}$  dada em (3.33) como guia para a escolha das variáveis  $(\alpha, \mu, \sigma)$  e por conseguinte do próximo ponto. Por conta do exposto acima, exigir que o próximo ponto pertença à vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$ , equivale a dar uma garantia de que este ponto estará a uma distância adequada da trajetória central.

Para tanto, é necessário escrever funções nas variáveis  $(\alpha, \mu, \sigma)$  que sirvam de restrições para o subproblema de minimização global da função de mérito  $\hat{\varphi}$  e que garantam que o ponto escolhido esteja em  $\mathcal{N}_{-\infty}(\gamma, \beta)$ .

Notando que  $\rho_C = xz$  e portanto  $(\rho_C)_i = x_i z_i$  e  $\overline{\rho_C} = x^T z / n$ , e usando a Proposição 3.11 e a equação (3.40), é possível ver que um ponto  $(x, y, z) \in \mathcal{Q}^+$ , gerado pelo Algoritmo 3.1, cumpre as condições da vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$  para o sistema KKT escalado, se valerem as desigualdades

$$(\rho_C)_i(x, y, z) - \gamma \overline{\rho_C}(x, y, z) \geq 0, \quad \forall i = 1, \dots, n \quad (3.43a)$$

$$\overline{\rho_C}(x, y, z) - \frac{\nu_k}{\beta} \overline{\rho_C}^0 \geq 0. \quad (3.43b)$$

A escolha de  $(\alpha, \mu, \sigma)$  será feita através do seguinte subproblema de otimização: minimizar a função de mérito para o próximo ponto  $\hat{\varphi}$ , dada pelo Teorema 3.9, restrita às desigualdades (3.43), também para o próximo para o próximo ponto, sempre levando em conta o teste da razão.

Para expressar tal problema de forma apropriada, assim como feito com  $\hat{\varphi}$ , vamos explicitar as desigualdades (3.43) para o próximo ponto em termos de  $(\alpha, \mu, \sigma)$ . De fato, para que o próximo ponto  $(\hat{x}, \hat{y}, \hat{z})$  gerado pelo Algoritmo 3.1, definimos os lados esquerdos de (3.43) como

$$g_C^i(\hat{x}, \hat{y}, \hat{z}) := (\hat{\rho}_C(\hat{x}, \hat{y}, \hat{z}))_i - \gamma \overline{\hat{\rho}_C}(\hat{x}, \hat{y}, \hat{z}), \quad i = 1, \dots, n, \quad (3.44a)$$

$$g_L(\hat{x}, \hat{y}, \hat{z}) := \overline{\hat{\rho}_C}(\hat{x}, \hat{y}, \hat{z}) - \frac{\hat{\nu}}{\beta} \overline{\hat{\rho}_C}^0. \quad (3.44b)$$

Assim, usando os Teoremas 3.3 e 3.9 em conjunto com (3.44a), obtemos  $g_C^i$ , para  $i = 1, \dots, n$ , em função de  $(\alpha, \mu, \sigma)$  como

$$g_C^i(\alpha, \mu, \sigma) = (1 - \alpha) ((\rho_C)_i - \gamma \overline{\rho_C}) + \alpha \mu (1 - \gamma) + \\ + \alpha(\alpha - \sigma) \left( (L_{0,0})_i - \gamma \overline{L_{0,0}} \right) + \alpha^2 \left( \Lambda(\mu, \sigma)_i - \gamma \overline{\Lambda(\mu, \sigma)} \right). \quad (3.45a)$$

Por outro lado, considerando que  $\hat{\nu} = \nu_{k+1} = (1 - \alpha)\nu_k$  e levando em conta novamente os Teoremas 3.3 e 3.9 em conjunto com (3.44b), obtemos  $g_L$ , também em função de  $(\alpha, \mu, \sigma)$  com a seguinte expressão

$$g_L(\alpha, \mu, \sigma) = (1 - \alpha) (\overline{\rho_C} - \beta_L \nu_k) + \alpha \mu + \alpha(\alpha - \sigma) \overline{L_{0,0}} + \alpha^2 \overline{\Lambda(\mu, \sigma)}, \quad (3.45b)$$

em que  $\beta_L = \frac{\overline{\rho_C^0}}{\beta}$ .

Desta forma, exigindo que as desigualdades

$$g_C^i(\alpha, \mu, \sigma) \geq 0 \quad \forall i = 1, \dots, n$$

$$g_L(\alpha, \mu, \sigma) \geq 0$$

sejam satisfeitas para alguma tripla  $(\alpha, \mu, \sigma)$ , garantimos que o próximo ponto estará na vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$ . Neste sentido, essas desigualdades se tornam parte das restrições de nosso subproblema de otimização global, o qual leva a escolha, em cada iteração, de  $(\alpha, \mu, \sigma)$ .

Resumidamente, o subproblema de otimização global pode ser escrito como

$$\begin{array}{ll} \min_{(\alpha, \mu, \sigma)} & \hat{\varphi}(\alpha, \mu, \sigma) \\ \text{s. a.} & \begin{cases} g_C^i(\alpha, \mu, \sigma) \geq 0 \quad \forall i = 1, \dots, n \\ g_L(\alpha, \mu, \sigma) \geq 0 \\ 0 \leq (\alpha, \mu, \sigma) \leq u, \end{cases} \end{array} \quad (3.46)$$

em que  $u \in \mathbb{R}^3$  é um vetor de limitantes superiores para  $(\alpha, \mu, \sigma)$ . Para  $\alpha$ ,  $u_1 < 1$ , pois não queremos que o próximo ponto saia do interior do ortante positivo. Com efeito, também não queremos  $\alpha = 0$ . Por isso mesmo, durante a demonstração de convergência daremos garantia de que é possível reduzir a função de mérito usando  $\alpha > 0$ . O teste da razão é sempre utilizado em cada iteração para garantir isso. Por outro lado, para  $\mu$  e  $\sigma$  os limitantes superiores  $u_2$  e  $u_3$  são escalares positivos que podem ou não ser diferentes para esses dois parâmetros.

O problema (3.46) é um Problema de Otimização de Polinômios (POP) e, em geral, é difícil de ser resolvido. Considere que o conjunto dos polinômios aproxima o conjunto de todas as funções contínuas em conjuntos compactos e, por isso, minimizar polinômios globalmente não é uma tarefa trivial. Com efeito, para alguns exemplares, um POP é NP-Hard [46, 47].

No entanto, em nosso caso, Villas-Bôas et al. [72] resolvem um problema com características parecidas com (3.46) em tempo computacional muito baixo, o que permite sua utilização em termos práticos. Um resumo dessa estratégia, que foi utilizada em nossa implementação, está no Capítulo 5.

No presente capítulo estabelecemos as bases do Método de Escolha Otimizada de Parâmetros, as quais se encontram resumidas no Algoritmo 3.1. Nos capítulos que seguem estabeleceremos os resultados teóricos dessa abordagem, garantindo convergência e polinomialidade do algoritmo, bem como os resultados dos experimentos computacionais de nosso método. Mais uma vez ressaltamos que nossa abordagem é tratar o MEOP tanto do ponto de vista teórico, quanto do de implementação, garantindo que os resultados da análise teórica sejam estabelecidos para o algoritmo que está, de fato, sendo implementado.



## Capítulo 4

# Convergência polinomial do Método de Escolha Otimizada de Parâmetros

“Nothing ever comes to one, that is worth having, except as a result of hard work.”

---

*(Booker T. Washington)*

Conforme esclarecem Zhang e Tapia [83], do ponto de vista matemático, os conceitos de polinomialidade de um algoritmo e de taxa de convergência são incompatíveis. Enquanto a polinomialidade tem sentido apenas quando o algoritmo termina em um número finito de passos, a taxa de convergência está definida apenas para métodos que tomam infinitos passos para convergir. Quando se diz que um Método de Pontos Interiores é polinomial, tem-se em mente dados inteiros ou racionais e terminação finita. Por outro lado, quando se diz que o mesmo algoritmo tem convergência linear, por exemplo, isso é feito no sentido tradicional da análise numérica [59]. Para uma abordagem mais completa de teoria de complexidade em Métodos de Pontos Interiores veja o texto de Wright [77, cap. 3].

Da mesma forma que esses autores, discutiremos aqui tanto polinomialidade quanto taxa de convergência do Método de Escolha Otimizada de Parâmetros. O objetivo principal deste

capítulo é provar que o Algoritmo 3.1 converge em uma taxa Q-linear e tem ordem polinomial de iterações.

Para tanto, fixaremos os parâmetros  $(\mu, \sigma)$ . O motivo de fazermos isso é facilitar as contas. Neste sentido escolheremos estrategicamente para o parâmetro  $\mu$  o valor

$$\hat{\mu} = \eta \frac{x^T z}{n},$$

em que  $\eta \in (\eta_{\min}, \eta_{\max})$ , enquanto o parâmetro  $\sigma$  será fixado como  $\hat{\sigma} = 0$ . Em alguns momentos, utilizaremos apenas  $\hat{\sigma}$  ao invés de 0, pois derivaremos desigualdades importantes que vão valer para qualquer valor de  $\eta$  e  $\sigma$ . Essas escolhas fazem com que nosso método resolva sistemas lineares similares aos dos métodos seguidores de caminho, vistos no Capítulo 2. Os valores dos limitantes de  $\eta$  para estes  $\hat{\mu}$  e  $\hat{\sigma}$  fixados são determinados no Teorema 4.10.

A fixação desses dois parâmetros permite as demonstrações que seguem e a consequente prova da convergência do método de maneira mais fácil. No entanto, na prática, o método tem desempenho melhor, já que se busca, em cada iteração  $k$ , o minimizador global de  $\hat{\varphi}$  através da solução do problema (3.46), dado por  $(\alpha^*, \mu^*, \sigma^*)$ , e portanto  $\hat{\varphi}(\alpha^*, \mu^*, \sigma^*) \leq \hat{\varphi}(\hat{\alpha}, \hat{\mu}, \hat{\sigma})$ , para todo  $\hat{\alpha}$ .

De fato, com  $\mu = \hat{\mu}$  e  $\sigma = \hat{\sigma}$ , vamos determinar que existe  $\hat{\alpha}$  no intervalo  $(0, 1)$  que cumpre com as condições da vizinhança  $N_{-\infty}(\gamma, \beta)$ , para algum  $\gamma$  e  $\beta$ . Por último, vamos mostrar que  $\hat{\alpha}$  é grande o suficiente para fazer com que a função de mérito  $\hat{\varphi}$  decresça suficientemente. Vejamos como isso se dá.

A fim de poder resolver o problema primal-dual (2.1-2.2), vamos considerar válido o Pressuposto 2.1, isto é, que o interior da região factível é não vazio. Além disso, vamos considerar que o PL em questão possui ao menos uma solução ótima.

Pelo Teorema 3.9, a função de mérito para o próximo ponto é, nas variáveis  $(\alpha, \mu, \sigma)$ ,

$$\hat{\varphi}(\alpha, \mu, \sigma) = (1 - \alpha)(\overline{\rho_L} + \overline{\rho_C^k}) + \alpha\mu + \alpha(\alpha - \sigma)\overline{L_{0,0}} + \alpha^2\overline{\Lambda(\mu, \sigma)},$$

em que

$$\overline{\Lambda(\mu, \sigma)} = \mu^2 \overline{L_{2,0}} + \mu \overline{L_{1,0}} + \mu \sigma \overline{L_{1,1}} + \sigma^2 \overline{L_{0,2}} + \sigma \overline{L_{0,1}}.$$

Como vimos na Observação 3.10,  $\overline{L_{2,0}} = \overline{L_{0,2}} = 0$ . Fixando  $\mu = \hat{\mu} = \eta \frac{x^T z}{n} = \eta \overline{\rho_C^k}$  e  $\sigma = \hat{\sigma} = 0$  e usando a Proposição 3.11, podemos rescrever a função de mérito para o próximo ponto, dependendo apenas de uma escolha de  $\alpha$  como

$$\hat{\varphi}(\alpha) = (1 - \alpha)(\nu_k \overline{\rho_L^0} + \overline{\rho_C^k}) + \alpha \eta \overline{\rho_C^k} + \alpha^2 (\overline{L_{0,0}} + \eta \overline{\rho_C^k} \overline{L_{1,0}}). \quad (4.1)$$

Apenas para facilitar a notação, dada uma iteração  $k$  qualquer, definimos  $\hat{\varphi} = \varphi_{k+1}$  e  $\varphi = \varphi_k$ .

Seja a função auxiliar dada por

$$\theta(\alpha) = \frac{\alpha \left[ \nu_k \overline{\rho_L^0} + (1 - \eta) \overline{\rho_C^k} - \alpha (\overline{L_{0,0}} + \eta \overline{\rho_C^k} \overline{L_{1,0}}) \right]}{\nu_k \overline{\rho_L^0} + \overline{\rho_C^k}}. \quad (4.2)$$

Usando tal definição, derivamos a seguinte relação entre a função de mérito atual  $\varphi$  e a da próxima iteração  $\hat{\varphi}$ :

$$\hat{\varphi} = (1 - \theta(\alpha))\varphi. \quad (4.3)$$

Como é necessário garantir que  $\hat{\varphi}$  seja não negativa, devemos escolher um tamanho de passo  $\alpha_k$ , em cada iteração  $k$ , tal que  $\theta_k = \theta(\alpha_k) < 1$ . Além disso, se existir um escalar  $\theta > 0$ , tal que  $\theta = \liminf(\theta_k)$ , então para  $k$  suficientemente grande

$$\frac{\varphi_{k+1}}{\varphi_k} < (1 - \theta) < 1 \quad (4.4)$$

e, logo, a sequência  $\{\varphi_k\}$ , gerada pelo Algoritmo 3.1, converge para zero Q-linearmente [59].

Como vimos, de modo a garantir a convergência dos pontos gerados pelo Algoritmo para uma solução ótima  $(x^*, y^*, z^*)$  do problema de PL, além de ser crucial que exista  $\hat{\alpha} > 0$  tal que, para toda iteração  $k$ , exista  $\alpha_k \in (0, \hat{\alpha}]$ , de modo que a equação (4.3) seja válida, é preciso que todo

ponto  $(x^k, y^k, z^k)$  pertença à vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$ .

Para tanto, sejam

$$\begin{cases} \hat{\alpha}_C^i = \max_{\alpha \in (0,1]} \{\alpha : g_C^i(v, \hat{\mu}, \hat{\sigma}) \geq 0 \text{ para todo } 0 \leq v \leq \alpha\}, & i = 1, \dots, n, \\ \hat{\alpha}_C = \min_{1 \leq i \leq n} \{\hat{\alpha}_C^i\}, \\ \hat{\alpha}_L = \max_{\alpha \in (0,1]} \{\alpha : g_L(v, \hat{\mu}, \hat{\sigma}) \geq 0 \text{ para todo } 0 \leq v \leq \alpha\}, \end{cases} \quad (4.5)$$

em que  $g_C^i$  e  $g_L$  são dadas pelas equações (3.45a) e (3.45b), já tendo sido fixados os parâmetros  $\mu = \hat{\mu}$  e  $\sigma = \hat{\sigma}$ .

Com isso, escolheremos  $\hat{\alpha}$  que garanta decréscimo suficiente de  $\hat{\varphi}$  ao mesmo tempo em que assegure a pertinência de  $(x^k, y^k, z^k)$  à vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$  por sujeitar  $\hat{\varphi}$  tanto à (3.45a) quanto a (3.45b). Isso será feito simplesmente por escolher

$$\hat{\alpha} = \min\{\hat{\alpha}_C, \hat{\alpha}_L\}, \quad (4.6)$$

em que  $\hat{\alpha}_C$  e  $\hat{\alpha}_L$  são definidos nos Lemas 4.8 e 4.9.

Veremos que essa escolha garante que o Algoritmo 3.1 gere uma sequência  $\{\varphi_k\}$  que decresce de maneira suficiente a cada passo, tal que  $\varphi_k \rightarrow 0$ , quando  $k \rightarrow \infty$ . Assim, o Algoritmo gera pontos  $(x^k, y^k, z^k)$  que tem as boas propriedades da vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$  e ainda faz com que  $(x^k, y^k, z^k)$  convirja para alguma solução ótima  $(x^*, y^*, z^*)$  do problema primal-dual (2.1-2.2).

Em seguida, fazemos a demonstração da convergência do Algoritmo proposto. A análise que segue é baseada naquelas feitas por Wright [77, cap. 6], Zhang [82] e Zhang e Zhang [85].

## 4.1 Resultados técnicos

A fim de demonstrarmos a convergência e a complexidade do Algoritmo proposto, na presente seção provaremos alguns lemas técnicos que serão utilizados para estabelecer nossos principais

resultados de convergência.

Primeiramente, é necessário escolher  $\gamma \in (0, 1)$  e  $\beta \geq 1$  adequados, para construir a vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$ . Para o primeiro parâmetro, várias escolhas são possíveis como, por exemplo, a de Colombo e Gondzio [9] que utilizam  $\gamma = 1/10$ . Vamos escolher  $\gamma$  tal que

$$\gamma \leq \min \left\{ \frac{\min_i(x_i^0 z_i^0)}{(x^0)^T z^0 / n}, \frac{1}{10} \right\},$$

o que garante que o ponto inicial satisfaz a desigualdade (3.43a).

Usando um raciocínio similar, para qualquer  $\beta \geq 1$ , o ponto inicial sempre satisfaz (3.43b). Quanto menor for  $\beta$ , mais acelerada é a redução das médias dos resíduos de complementaridade, dadas por  $\overline{\rho_C^k}$ , em relação à redução da média dos resíduos de linearidade  $\overline{\rho_L^k}$ . Para facilitar o raciocínio, do ponto de vista teórico, vamos escolher o menor valor possível para  $\beta$ , isto é,  $\beta = 1$ .

Além disso a seguinte observação será útil.

*Observação 4.1.* Seja  $(\Delta x, \Delta y, \Delta z)$  uma terna qualquer de direções. Se

$$A\Delta x = 0 \text{ e } A^T \Delta y + \Delta z = 0, \quad (4.7)$$

então

$$\Delta x^T \Delta z = -\Delta x^T (A^T \Delta y) = -(A\Delta x)^T \Delta y = 0. \quad (4.8)$$

Em demonstrações de convergência e polinomialidade de Métodos de Pontos Interiores infactíveis, em particular no métodos do tipo Mehrotra como o que propomos, é necessária a aceitação de algumas hipóteses ou condições sobre o ponto inicial e sobre uma solução ótima. Veja por exemplo os trabalhos de Wright [76], Wright [77, cap. 6], Wright e Zhang [78], Zhang [82] e Zhang e Zhang [84, 85]. Tais hipóteses permitem estabelecer um limitante para a sequência  $\{(x^k, z^k)\}$ , o que é de vital importância para as demonstrações.

No entanto, embora as hipóteses utilizadas nos trabalhos acima produzam uma ordem de

convergência polinomial, estas são impraticáveis do ponto de vista da implementação, já que utilizam pontos iniciais que geram péssimo comportamento quando utilizados e, ao mesmo tempo, exigem o conhecimento *a priori* de um limitante para a solução ótima.

Nesse trabalho, preferimos obter uma ordem de complexidade um pouco maior, embora ainda polinomial, porém com um ponto inicial cuja eficácia é bem demonstrada na prática. Para tanto propomos a Condição 4.2, que envolve o ponto inicial, uma solução ótima do PL e uma medida do *tamanho dos dados* do problema. Tal Condição foi satisfeita para todos os problemas do conjunto de testes que utilizamos – veja Seção 5.1, com ponto inicial  $(x^0, y^0, z^0)$ , dado em (2.30), obtido a partir da heurística de Mehrotra [49] – veja Seção 2.3.3. Estamos fazendo isso para garantir que a análise de convergência seja feita no Algoritmo que é de fato implementado.

A medida do tamanho dos dados vai ser dada pela constante positiva

$$\varsigma = \max \left\{ |A_{ij}|, |b_i|, |c_j|, \text{ para } 1 \leq i \leq m \text{ e } 1 \leq j \leq n \right\}, \quad (4.9)$$

que corresponde ao maior valor absoluto de todos os dados de entrada do problema (2.1-2.2). Podemos supor sem perda de generalidade que  $\varsigma \geq 1$ , bastando para isso fazer um escalonamento trivial. Entretanto, na prática, usualmente são feitos escalonamento mais sofisticados, como o de Curtis e Reid [10]. Em nossa implementação, mantivemos o escalonamento de Curtis e Reid, que é usado no PCx.

A Condição 4.2, que enunciamos a seguir, é uma hipótese teórica necessária para as demonstrações de convergência e complexidade que estamos procedendo, porém não será necessária na implementação do MEOP. Como dissemos, ela é válida para o ponto inicial que escolhemos na implementação, porém é possível que estudos futuros determinem outros pontos iniciais que, ao satisfazerem tal condição, têm garantia de convergência do MEOP.

**Condição 4.2.** *Para o ponto inicial dado por (2.30), com  $(x^0, z^0) > 0$ , existe uma solução ótima*

$(x^*, y^*, z^*)$  do problema (2.1-2.2) que satisfaz

$$\frac{2(x^0)^T z^0 + (x^0)^T z^* + (x^*)^T z^0}{(x^0)^T z^0 \min_i \{x_i^0, z_i^0\}} \|(x^0, z^0) - (x^*, z^*)\| < \varsigma^4, \quad (4.10)$$

em que  $\varsigma > 0$  é calculado por (4.9).

O primeiro lema a seguir mostra a existência de um limitante para  $\nu_k \|(x^k, z^k)\|_1$ . Uma demonstração similar, porém com condições para o ponto inicial e para a solução ótima diferentes da que aceitamos aqui, pode ser encontrada em [77, Lema 6.1], fonte da qual a transcrevemos com as adaptações pertinentes à nossa notação bem como à hipótese que aqui assumimos.

**Lema 4.3.** *Suponha que a Condição 4.2 seja válida. Então, para qualquer iterado  $(x^k, y^k, z^k)$  gerado pelo Algoritmo 3.1, vale*

$$\nu_k \|(x^k, z^k)\|_1 \|(x^0, z^0) - (x^*, z^*)\| \leq \varsigma^4 n \overline{\rho_C^k}, \quad (4.11)$$

em que  $\nu_k$  é definido em (3.39).

*Demonstração.* Seja o ponto auxiliar

$$(\tilde{x}, \tilde{y}, \tilde{z}) = \nu_k(x^0, y^0, z^0) + (1 - \nu_k)(x^*, y^*, z^*) - (x^k, y^k, z^k).$$

em que  $(x^0, y^0, z^0)$  é ponto inicial dado por (2.30) e  $(x^*, y^*, z^*)$  é uma solução ótima, para os quais é válido (4.10).

Note que

$$\begin{aligned} A\tilde{x} &= A(\nu_k x^0 + (1 - \nu_k)x^* - x^k) = \nu_k(Ax^0 - Ax^*) + Ax^* - Ax^k \\ &= \nu_k(Ax^0 - b) + (b - Ax^k) = \nu_k r_P^0 - r_P^k \\ &= r_P^k - r_P^k = 0, \end{aligned}$$

e que

$$\begin{aligned}
A^T \tilde{y} + \tilde{z} &= A^T(\nu_k y^0 + (1 - \nu_k)y^* - y^k) + (\nu_k z^0 + (1 - \nu_k)z^* - z^k) \\
&= \nu_k [(A^T y^0 + z^0) - (A^T y^* + z^*)] + [(A^T y^* + z^*) - (A^T y^k + z^k)] \\
&= \nu_k r_D^0 - r_D^k \\
&= r_D^k - r_D^k = 0.
\end{aligned}$$

Portando,  $(\tilde{x}, \tilde{y}, \tilde{z})$  também satisfaz (4.7) e então vale (4.8), isto é,  $(\tilde{x})^T \tilde{z} = 0$ .

Logo temos que

$$\begin{aligned}
0 = \tilde{x}^T \tilde{z} &= (\nu_k x^0 + (1 - \nu_k)x^* - x^k)^T (\nu_k z^0 + (1 - \nu_k)z^* - z^k) \\
&= \nu_k^2 (x^0)^T z^0 + (1 - \nu_k)^2 (x^*)^T z^* + \nu_k(1 - \nu_k)((x^0)^T z^* + (x^*)^T z^0) \\
&\quad + (x^k)^T z^k - \nu_k((x^k)^T z^0 + (x^0)^T z^k) - (1 - \nu_k)((x^k)^T z^* + (x^*)^T z^k). \tag{4.12}
\end{aligned}$$

Como  $(x^k, z^k) > 0$  e  $(x^*, z^*) \geq 0$ , vale  $((x^k)^T z^* + (x^*)^T z^k) \geq 0$ . Além disso,  $(x^*, y^*, z^*)$  é uma solução ótima, e por isso  $(x^*)^T z^* = 0$ . Usando essas observações e levando em conta que  $\nu_k \in (0, 1)$ , podemos reorganizar a equação (4.12) como

$$\nu_k((x^k)^T z^0 + (x^0)^T z^k) \leq \nu_k^2 (x^0)^T z^0 + (x^k)^T z^k + \nu_k(1 - \nu_k)((x^0)^T z^* + (x^*)^T z^0). \tag{4.13}$$

Note que como  $(x^k, z^k) > 0$ , segue que

$$\begin{aligned}
\min_i \{x_i^0, z_i^0\} \|(x^k, z^k)\|_1 &= \min_i \{x_i^0, z_i^0\} \left( \sum_{i=1}^n x_i^k + \sum_{i=1}^n z_i^k \right) \\
&\leq \min_i (z_i^0) \|x^k\|_1 + \min_i (x_i^0) \|z^k\|_1 \leq (x^k)^T z^0 + (x^0)^T z^k,
\end{aligned}$$

o que implica que  $\nu_k \|(x^k, z^k)\|_1 \leq \nu_k((x^k)^T z^0 + (x^0)^T z^k) \frac{1}{\min_i \{x_i^0, z_i^0\}}$ .

Comparando o primeiro termo do lado direito dessa desigualdade com (4.13) obtemos

$$\nu_k \|(x^k, z^k)\|_1 \leq \left[ \nu_k^2 (x^0)^T z^0 + (x^k)^T z^k + \nu_k(1 - \nu_k)((x^0)^T z^* + (x^*)^T z^0) \right] \frac{1}{\min_i \{x_i^0, z_i^0\}}.$$

Agora,  $\nu_k \in (0, 1)$  implica que  $\nu_k^2 < \nu_k$  e que  $\nu_k(1 - \nu_k) < \nu_k$ . Além disso, para qualquer  $k$ ,  $(x^k)^T z^k = n\overline{\rho_C^k}$ . Usando esses argumentos na última desigualdade obtemos

$$\nu_k \|(x^k, z^k)\|_1 \leq \left[ n\nu_k \overline{\rho_C^0} + n\overline{\rho_C^k} + \nu_k((x^0)^T z^* + (x^*)^T z^0) \right] \frac{1}{\min_i \{x_i^0, z_i^0\}}.$$

Considerando que  $(x^k, y^k, z^k)$  pertence à vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$  e portanto vale (3.42), e considerando ainda que  $\beta = 1$ , temos  $\nu_k \leq \overline{\rho_C^k} / \overline{\rho_C^0}$ .

Por tudo isso, obtemos

$$\begin{aligned} \nu_k \|(x^k, z^k)\|_1 &\leq \left[ n\overline{\rho_C^k} + n\overline{\rho_C^k} + \left( \overline{\rho_C^k} / \overline{\rho_C^0} \right) ((x^0)^T z^* + (x^*)^T z^0) \right] \frac{1}{\min_i \{x_i^0, z_i^0\}} \\ &= \left[ 2n\overline{\rho_C^k} + \frac{n\overline{\rho_C^k}}{(x^0)^T z^0} ((x^0)^T z^* + (x^*)^T z^0) \right] \frac{1}{\min_i \{x_i^0, z_i^0\}} \\ &= \frac{2(x^0)^T z^0 + ((x^0)^T z^* + (x^*)^T z^0)}{(x^0)^T z^0 \min_i \{x_i^0, z_i^0\}} n\overline{\rho_C^k}. \end{aligned}$$

Multiplicando ambos os lados dessa última desigualdade por  $\|(x^0, z^0) - (x^*, z^*)\|$  e utilizando a Equação (4.10) da Condição 4.2 finalmente obtemos

$$\begin{aligned} \nu_k \|(x^k, z^k)\|_1 \|(x^0, z^0) - (x^*, z^*)\| &\leq \frac{2(x^0)^T z^0 + ((x^0)^T z^* + (x^*)^T z^0)}{(x^0)^T z^0 \min_i \{x_i^0, z_i^0\}} \|(x^0, z^0) - (x^*, z^*)\| n\overline{\rho_C^k} \\ &< \varsigma^4 n\overline{\rho_C^k}, \end{aligned}$$

como queríamos demonstrar. ■

A próxima proposição será utilizada nas demonstrações que seguem e é um fato conhecido e bastante utilizado em MPI.

**Proposição 4.4.** *Seja  $D$  uma matriz diagonal não-singular de ordem  $n$  e sejam  $u, v \in \mathbb{R}^n$  tais que  $u^T v \geq 0$ , então*

$$\|uv\|_1 \leq \|Du\| \|D^{-1}v\| \leq \frac{1}{2} \left( \|Du\|^2 + \|D^{-1}v\|^2 \right). \quad (4.14)$$

*Demonstração.* Para a primeira desigualdade vejamos que

$$\begin{aligned} \|uv\|_1^2 &= \left( \sum_{i=1}^n |u_i v_i| \right)^2 \leq \sum_{i=1}^n |u_i v_i|^2 \\ &= \|uv\|^2 = (uv)^T (uv) = v^T D^{-T} D^T u^T D D^{-1} uv \\ &= v^T D^{-T} D^T u^T D u D^{-1} v = \|Du\|^2 \|D^{-1}v\|^2. \end{aligned}$$

e logo a segunda desigualdade de (4.14) está provada.

Com relação a terceira desigualdade, note que

$$0 \leq \left( \|Du\| - \|D^{-1}v\| \right)^2 = \|Du\|^2 - 2\|Du\| \|D^{-1}v\| + \|D^{-1}v\|^2$$

e portanto temos

$$\|Du\| \|D^{-1}v\| \leq \frac{1}{2} \left( \|Du\|^2 + \|D^{-1}v\|^2 \right). \quad \blacksquare$$

De agora em diante, dados  $(x^k, z^k)$  gerado pelo Algoritmo, definimos a matriz  $D^k$  como sendo

$$D^k = (X^k)^{-1/2} (Z^k)^{1/2}.$$

Podemos agora enunciar o seguinte lema.

**Lema 4.5.** *Suponha válida a Condição 4.2. Então*

$$\|(\Delta x^{af})^k (\Delta z^{af})^k\| \leq \|(\Delta x^{af})^k (\Delta z^{af})^k\|_1 \leq \frac{(1 + 2\zeta^4)^2}{\gamma} n^2 \overline{\rho_C^k}, \quad (4.15)$$

em que  $\frac{(1 + 2\zeta^4)^2}{\gamma} > 1$ .

*Demonstração.* Seja a direção auxiliar

$$(\Delta x, \Delta y, \Delta z) = ((\Delta x^{af})^k, (\Delta y^{af})^k, (\Delta z^{af})^k) + \nu_k(x^0 - x^*, y^0 - y^*, z^0 - z^*),$$

em que  $(x^0, y^0, z^0)$  é o ponto inicial dado por (2.30) e  $(x^*, y^*, z^*)$  é uma solução ótima, satisfazendo (4.10).

Note que

$$\begin{aligned} A\Delta x &= A((\Delta x^{af})^k + \nu_k(x^0 - x^*)) \\ &= A(\Delta x^{af})^k + \nu_k(Ax^0 - Ax^*) = A(\Delta x^{af})^k + \nu_k(Ax^0 - b) \\ &= -r_P^k + \nu_k r_P^0 = -r_P^k + r_P^k = 0, \end{aligned}$$

e que

$$\begin{aligned} A^T \Delta y + \Delta z &= A^T((\Delta y^{af})^k + \nu_k(y^0 - y^*)) + ((\Delta z^{af})^k + \nu_k(z^0 - z^*)) \\ &= (A^T(\Delta y^{af})^k + (\Delta z^{af})^k) + \nu_k(A^T y^0 + z^0 - (A^T y^* + z^*)) \\ &= -r_D^k + \nu_k(A^T y^0 + z^0 - c) \\ &= -r_D^k + \nu_k r_D^0 = -r_D^k + r_D^k = 0. \end{aligned}$$

Portando,  $(\Delta x, \Delta y, \Delta z)$  satisfaz (4.7) e logo vale (4.8), isto é,  $\Delta x^T \Delta z = 0$ . Assim,

$$0 = \Delta x^T \Delta z = \left( (\Delta x^{af})^k + \nu_k(x^0 - x^*) \right)^T \left( (\Delta z^{af})^k + \nu_k(z^0 - z^*) \right). \quad (4.16)$$

Usando essa direção auxiliar e equação (3.6c), obtemos

$$\begin{aligned} Z^k \left( (\Delta x^{\text{af}})^k + \nu_k(x^0 - x^*) \right) + X^k \left( (\Delta z^{\text{af}})^k + \nu_k(z^0 - z^*) \right) \\ = -x^k z^k + \nu_k Z^k(x^0 - x^*) + \nu_k X^k(z^0 - z^*). \end{aligned}$$

Multiplicando toda essa expressão por  $(X^k Z^k)^{-1/2}$ , e notando que  $D^k = (X^k Z^k)^{-1/2} Z^k$  e que  $(D^k)^{-1} = (X^k Z^k)^{-1/2} X^k$ , resulta em

$$\begin{aligned} D^k \left( (\Delta x^{\text{af}})^k + \nu_k(x^0 - x^*) \right) + (D^k)^{-1} \left( (\Delta z^{\text{af}})^k + \nu_k(z^0 - z^*) \right) = \\ - (x^k z^k)^{1/2} + \nu_k D^k(x^0 - x^*) + \nu_k (D^k)^{-1}(z^0 - z^*). \quad (4.17) \end{aligned}$$

Note que, como vale (4.16), utilizando o Teorema de Pitágoras para norma-2 de vetores e tomando a norma-2 ao quadrado do lado esquerdo de (4.17) obtemos

$$\begin{aligned} \left\| D^k \left( (\Delta x^{\text{af}})^k + \nu_k(x^0 - x^*) \right) + (D^k)^{-1} \left( (\Delta z^{\text{af}})^k + \nu_k(z^0 - z^*) \right) \right\|^2 = \\ \left\| D^k \left( (\Delta x^{\text{af}})^k + \nu_k(x^0 - x^*) \right) \right\|^2 + \left\| (D^k)^{-1} \left( (\Delta z^{\text{af}})^k + \nu_k(z^0 - z^*) \right) \right\|^2. \end{aligned}$$

Se usarmos este resultado e a desigualdade triangular, após tomarmos a norma-2 ao quadrado de ambos os lados de (4.17), obtemos

$$\begin{aligned} \left\| D^k \left( (\Delta x^{\text{af}})^k + \nu_k(x^0 - x^*) \right) \right\|^2 + \left\| (D^k)^{-1} \left( (\Delta z^{\text{af}})^k + \nu_k(z^0 - z^*) \right) \right\|^2 \leq \\ \left\{ \left\| (x^k z^k)^{1/2} \right\| + \nu_k \left\| D^k(x^0 - x^*) \right\| + \nu_k \left\| (D^k)^{-1}(z^0 - z^*) \right\| \right\}^2. \end{aligned}$$

Como o segundo termo dessa inequação é não negativo, obtemos

$$\left\| D^k \left( (\Delta x^{\text{af}})^k + \nu_k(x^0 - x^*) \right) \right\| \leq \left\| (x^k z^k)^{1/2} \right\| + \nu_k \left\| D^k(x^0 - x^*) \right\| + \nu_k \left\| (D^k)^{-1}(z^0 - z^*) \right\|.$$

Uma aplicação direta da desigualdade triangular e a adição de um termo  $\nu_k \|(D^k)^{-1}(z^0 - z^*)\|$  extra resulta em

$$\begin{aligned} \|D^k(\Delta x^{\text{af}})^k\| &= \left\| D^k \left( (\Delta x^{\text{af}})^k + \nu_k(x^0 - x^*) - \nu_k(x^0 - x^*) \right) \right\| \\ &\leq \left\| D^k \left( (\Delta x^{\text{af}})^k + \nu_k(x^0 - x^*) \right) \right\| + \nu_k \|D^k(x^0 - x^*)\| \\ &\leq \|(x^k z^k)^{1/2}\| + 2\nu_k \|D^k(x^0 - x^*)\| + 2\nu_k \|(D^k)^{-1}(z^0 - z^*)\|. \end{aligned} \quad (4.18)$$

Vamos mostrar a existência de um limitante de magnitude  $\mathcal{O}\left(\overline{\rho_C^k}^{-1/2}\right)$  para cada termo do lado direito de (4.18). Nesse caso, o mesmo se aplicará a  $\|(D^k)^{-1}(\Delta z^{\text{af}})^k\|$ .

Para o primeiro termo, note que

$$\|(x^k z^k)^{1/2}\| = \left( \sum_{i=1}^n x_i^k z_i^k \right)^{1/2} = ((x^k)^T z^k)^{1/2} = n^{1/2} \overline{\rho_C^k}^{-1/2} \leq \frac{n}{\gamma^{1/2}} \overline{\rho_C^k}^{-1/2}, \quad (4.19)$$

já que  $\gamma \in (0, 1)$  e  $\sqrt{n} \leq n$ , para todo  $n$  natural.

A norma-2 da matriz  $D^k$  será

$$\|D^k\| = \max_{i=1, \dots, n} |D_{ii}^k| = \|D^k e\|_\infty = \|(X^k Z^k)^{-1/2} z^k\|_\infty \leq \|(X^k Z^k)^{-1/2}\| \|z^k\|_1,$$

e similarmente

$$\|(D^k)^{-1}\| \leq \|(X^k Z^k)^{-1/2}\| \|x^k\|_1.$$

Mais que isso, como  $(x^k, y^k, z^k) \in \mathcal{N}_{-\infty}(\gamma, \beta)$ , de (3.41), temos que

$$\|(X^k Z^k)^{-1/2}\| = \max_{i=1, \dots, n} \frac{1}{(x_i^k z_i^k)^{1/2}} \leq \frac{1}{\gamma^{1/2} \overline{\rho_C^k}^{-1/2}}. \quad (4.20)$$

Com essas desigualdades para norma de  $D^k$  e de  $(D^k)^{-1}$ , utilizando propriedades de norma

de matrizes e vetores segue que

$$\begin{aligned}
\nu_k \|D^k(x^0 - x^*)\| + \nu_k \|(D^k)^{-1}(z^0 - z^*)\| &\leq \nu_k \|D^k\| \|x^0 - x^*\| + \nu_k \|(D^k)^{-1}\| \|z^0 - z^*\| \\
&\leq \nu_k \left( \|D^k\| + \|(D^k)^{-1}\| \right) \|(x^0, z^0) - (x^*, z^*)\| \\
&\leq \nu_k \|(X^k Z^k)^{-1/2}\| \left( \|x^k\|_1 + \|z^k\|_1 \right) \|(x^0, z^0) - (x^*, z^*)\| \\
&\leq \nu_k \|(x^k, z^k)\|_1 \|(x^0, z^0) - (x^*, z^*)\| \|(X^k Z^k)^{-1/2}\|.
\end{aligned}$$

Do Lema 4.3, da Equação (4.20) e da desigualdade acima segue que

$$\nu_k \|D^k(x^0 - x^*)\| + \nu_k \|(D^k)^{-1}(z^0 - z^*)\| \leq \varsigma^4 n \overline{\rho_C^k} \frac{1}{\gamma^{1/2} \overline{\rho_C^k}^{1/2}} = \frac{\varsigma^4}{\gamma^{1/2}} \overline{\rho_C^k}^{1/2}. \quad (4.21)$$

Se usarmos as equações (4.19) e (4.21), comparando-as com (4.18), temos que

$$\begin{aligned}
\|D^k(\Delta x^{\text{af}})^k\| &\leq \frac{n}{\gamma^{1/2}} \overline{\rho_C^k}^{1/2} + \frac{2\varsigma^4}{\gamma^{1/2}} n \overline{\rho_C^k}^{1/2} \\
&\leq \frac{1 + 2\varsigma^4}{\gamma^{1/2}} n \overline{\rho_C^k}^{1/2},
\end{aligned}$$

em que  $\frac{1 + 2\varsigma^4}{\gamma^{1/2}} > 1$ , já que  $\gamma \in (0, 1)$ .

Com isso, obtemos que

$$\|D^k(\Delta x^{\text{af}})^k\| \leq \left( \frac{1 + 2\varsigma^4}{\gamma^{1/2}} \right) n \overline{\rho_C^k}^{1/2} \text{ e } \|(D^k)^{-1}(\Delta z^{\text{af}})^k\| \leq \left( \frac{1 + 2\varsigma^4}{\gamma^{1/2}} \right) n \overline{\rho_C^k}^{1/2}. \quad (4.22)$$

Usando as desigualdades da Proposição 4.4 e a Equação (4.22), completamos a demonstração, obtendo

$$\begin{aligned}
\|(\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k\| &\leq \|(\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k\|_1 \leq \frac{1}{2} \left( \|D^k(\Delta x^{\text{af}})^k\|^2 + \|(D^k)^{-1}(\Delta z^{\text{af}})^k\|^2 \right) \\
&\leq \left[ \left( \frac{1 + 2\varsigma^4}{\gamma^{1/2}} \right) n \overline{\rho_C^k}^{1/2} \right]^2 = \frac{(1 + 2\varsigma^4)^2}{\gamma} n^2 \overline{\rho_C^k}. \quad \blacksquare
\end{aligned}$$

**Lema 4.6.** *Suponha válida a Condição 4.2. Se*

$$\hat{\sigma} > \frac{\gamma(\sqrt{\gamma} - \eta)}{(1 + 2\zeta^4)^2}, \quad (4.23)$$

então

$$\|(\Delta x^c)^k (\Delta z^c)^k\| \leq \|(\Delta x^c)^k (\Delta z^c)^k\|_1 \leq \frac{[\eta\gamma + \hat{\sigma}(1 + 2\zeta^4)^2]^2}{2\gamma^3} n^4 \overline{\rho_C^k}, \quad (4.24)$$

em que  $\frac{[\eta\gamma + \hat{\sigma}(1 + 2\zeta^4)^2]^2}{2\gamma^3} > 1$ .

*Demonstração.* Note que por conta de (3.14), temos que  $((\Delta x^c)^k, (\Delta y^c)^k, (\Delta z^c)^k)$  satisfaz (4.7) e logo  $((\Delta x^c)^k)^T (\Delta z^c)^k = 0$ .

Além disso, também por causa de (3.14), temos que

$$Z^k (\Delta x^c)^k + X^k (\Delta z^c)^k = \mu e - \sigma (\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k.$$

Multiplicando toda equação a anterior por  $(X^k Z^k)^{-1/2}$ , para algum  $\mu$  e  $\sigma$  a ser escolhido, segue que

$$D^k (\Delta x^c)^k + (D^k)^{-1} (\Delta z^c)^k = (X^k Z^k)^{-1/2} (\mu e - \sigma (\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k).$$

Por um lado, temos que

$$\begin{aligned} \|D^k (\Delta x^c)^k + (D^k)^{-1} (\Delta z^c)^k\|^2 &= \|D^k (\Delta x^c)^k\|^2 + \|(D^k)^{-1} (\Delta z^c)^k\|^2 + 2((\Delta x^c)^k)^T (\Delta z^c)^k \\ &= \|D^k (\Delta x^c)^k\|^2 + \|(D^k)^{-1} (\Delta z^c)^k\|^2. \end{aligned}$$

Assim, usando consistência de normas, obtemos

$$\begin{aligned} \|D^k (\Delta x^c)^k\|^2 + \|(D^k)^{-1} (\Delta z^c)^k\|^2 &= \|(X^k Z^k)^{-1/2} (\mu e - \sigma (\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k)\|^2 \\ &\leq \|(X^k Z^k)^{-1/2}\|^2 \|(\mu e - \sigma (\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k)\|^2. \end{aligned}$$

A Equação (4.20) e o fato de que  $(x^k, y^k, z^k) \in \mathcal{N}_{-\infty}(\gamma, \beta)$  permitem que derivemos, a partir da desigualdade acima,

$$\begin{aligned} \|D^k(\Delta x^c)^k\|^2 + \|(D^k)^{-1}(\Delta z^c)^k\|^2 &\leq \left[ \min \{x_i^k z_i^k\} \right]^{-1} \left\| (\mu e - \sigma(\Delta x^{\text{af}})^k(\Delta z^{\text{af}})^k) \right\|^2 \\ &\leq \left[ \min \{x_i^k z_i^k\} \right]^{-1} \left( \mu \|e\| + \sigma \left\| (\Delta x^{\text{af}})^k(\Delta z^{\text{af}})^k \right\| \right)^2 \\ &\leq (\gamma \overline{\rho_C^k})^{-1} \left( \mu \sqrt{n} + \sigma \left\| (\Delta x^{\text{af}})^k(\Delta z^{\text{af}})^k \right\| \right)^2. \end{aligned} \quad (4.25)$$

Para utilizar (4.15), definimos a variável local  $t = \frac{(1 + 2\varsigma^4)^2}{\gamma}$ , tal que

$$\left\| (\Delta x^{\text{af}})^k(\Delta z^{\text{af}})^k \right\| \leq tn^2 \overline{\rho_C^k}.$$

Agora, fixamos  $\mu = \hat{\mu} = \eta \overline{\rho_C^k}$  e  $\sigma = \hat{\sigma}$ , este último ainda a ser determinado. Notando que para todo  $n$  natural vale  $\sqrt{n} \leq n^2$ , a desigualdade (4.25) pode ser transformada em

$$\begin{aligned} \|D^k(\Delta x^c)^k\|^2 + \|(D^k)^{-1}(\Delta z^c)^k\|^2 &\leq (\gamma \overline{\rho_C^k})^{-1} \left( \eta \overline{\rho_C^k} \sqrt{n} + \hat{\sigma} tn^2 \overline{\rho_C^k} \right)^2 \\ &\leq (\gamma)^{-1} \left( \eta \sqrt{n} + \hat{\sigma} tn^2 \right)^2 \overline{\rho_C^k} \\ &\leq (\gamma)^{-1} (\eta + \hat{\sigma} t)^2 n^4 \overline{\rho_C^k}. \end{aligned} \quad (4.26)$$

Observe que  $\gamma \in (0, 1)$  e, logo,  $t > 1$ . Daí, se  $\eta \in [0, 1]$  e se (4.23) for satisfeita, então

$$\hat{\sigma} > \frac{\sqrt{\gamma} - \eta}{t}$$

e por isso garantimos que  $\frac{(\eta + \hat{\sigma} t)^2}{\gamma} > 1$ , como queríamos.

Ao substituírmos o valor de  $t$  nessa última desigualdade, temos

$$\frac{(\eta + \hat{\sigma} t)^2}{\gamma} = \frac{\left[ \eta + \hat{\sigma} \frac{(1 + 2\varsigma^4)^2}{\gamma} \right]^2}{\gamma} = \frac{[\eta\gamma + \hat{\sigma}(1 + 2\varsigma^4)]^2}{\gamma^3}.$$

Substituindo essa expressão em (4.26), obtemos

$$\|D^k(\Delta x^c)^k\|^2 + \|(D^k)^{-1}(\Delta z^c)^k\|^2 \leq \frac{[\eta\gamma + \hat{\sigma}(1 + 2\zeta^4)^2]^2}{\gamma^3} n^4 \overline{\rho_C^k}. \quad (4.27)$$

Pra finalizarmos, como consequência das desigualdades dadas pela Proposição 4.4 e da Inequação (4.27) temos

$$\begin{aligned} \|(\Delta x^c)^k(\Delta z^c)^k\| &\leq \|(\Delta x^c)^k(\Delta z^c)^k\|_1 \leq \frac{1}{2} \left( \|D^k(\Delta x^c)^k\|^2 + \|(D^k)^{-1}(\Delta z^c)^k\|^2 \right) \\ &\leq \frac{[\eta\gamma + \hat{\sigma}(1 + 2\zeta^4)^2]^2}{2\gamma^3} n^4 \overline{\rho_C^k}, \end{aligned}$$

como queríamos demonstrar. ■

**Lema 4.7.** *Suponha válida a Condição 4.2. Então vale*

$$\|(\Delta x^{af})^k(\Delta z^c)^k + (\Delta x^c)^k(\Delta z^{af})^k\|_1 \leq \frac{2(1 + 2\zeta^4) [\eta\gamma + \hat{\sigma}(1 + 2\zeta^4)^2]}{\gamma^2} n^3 \overline{\rho_C^k}. \quad (4.28)$$

*Demonstração.* Sejam as constantes locais

$$t_1 = \frac{1 + 2\zeta^4}{\gamma^{1/2}} \quad \text{e} \quad t_2 = \frac{[\eta\gamma + \hat{\sigma}(1 + 2\zeta^4)^2]^2}{\gamma^3}.$$

A partir das Inequações (4.22) e (4.24) e da inequação (4.14) segue que

$$\begin{aligned} \|(\Delta x^{af})^k(\Delta z^c)^k\|_1 &\leq \|D^k(\Delta x^{af})^k\| \|(D^k)^{-1}(\Delta z^c)^k\| \\ &\leq t_1 n \overline{\rho_C^k}^{1/2} (t_2 n^4 \overline{\rho_C^k})^{1/2} \\ &= (t_1 t_2^{1/2}) n^3 \overline{\rho_C^k}. \end{aligned}$$

Similarmente,  $\|(\Delta x^c)^k(\Delta z^{\text{af}})^k\|_1 \leq (t_1 t_2^{1/2}) n^3 \overline{\rho_C^k}$ . Com isso, temos que

$$\|(\Delta x^{\text{af}})^k(\Delta z^c)^k + (\Delta x^c)^k(\Delta z^{\text{af}})^k\|_1 \leq 2(t_1 t_2^{1/2}) n^3 \overline{\rho_C^k}.$$

Para finalizar, temos que

$$2(t_1 t_2^{1/2}) = 2 \left( \frac{1 + 2\zeta^4}{\gamma^{1/2}} \right) \cdot \left( \frac{\eta\gamma + \hat{\sigma}(1 + 2\zeta^4)^2}{\gamma^{3/2}} \right) = \frac{2(1 + 2\zeta^4) [\eta\gamma + \hat{\sigma}(1 + 2\zeta^4)^2]}{\gamma^2}. \quad \blacksquare$$

## 4.2 Prova de complexidade e convergência

A fim de que o Algoritmo 3.1 esteja bem definido é necessário que exista em cada iteração  $k$  uma tripla  $(\alpha_k, \mu_k, \sigma_k)$ , de modo que seja possível encontrar um próximo ponto  $(\hat{x}, \hat{y}, \hat{z})$ . Com efeito, considerando que fixamos os valores de  $\mu$  como  $\hat{\mu} = \overline{\eta\rho_C^k}$  e agora que faremos  $\sigma$  como  $\hat{\sigma} = 0$ , basta encontrar um tamanho de passo  $\alpha_k > 0$  tal que o próximo ponto  $(\hat{x}, \hat{y}, \hat{z})$  satisfaça as restrições da vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$  e, além disso, tenhamos  $0 < \theta(\alpha_k) < 1$ . É o que os próximos resultados garantem. Observe que, por conta de (4.23), como  $\hat{\sigma} = 0$ , teremos que  $\eta > \sqrt{\gamma}$ . Isso dará o primeiro limitante para  $\eta$ .

**Lema 4.8.** *Seja  $\hat{\alpha}_C$  dado em (4.5). Então*

$$\hat{\alpha}_C \geq \delta_1/n^4,$$

em que

$$\delta_1 = \frac{2\eta\gamma(1 - \gamma)}{\eta^2 + 8\eta(1 + 2\zeta^4) + 4(1 + 2\zeta^4)^2} \quad (4.29)$$

é uma constante independente de  $n$ .

*Demonstração.* Primeiramente, considere que, assim como na Equação (4.1), as escolhas de  $\hat{\mu}$  e  $\hat{\sigma}$  permitem rescrever a função  $g_C^i$ , para  $i = 1, \dots, n$ , que foi dada em (3.45a), somente

dependendo de uma escolha de  $\alpha$ . De fato, temos

$$g_C^i(\alpha) = (1 - \alpha)(\rho_C^k)_i + \alpha\eta\overline{\rho_C^k} + \alpha^2 \left[ (L_{0,0}^k)_i + \eta\overline{\rho_C^k}(L_{1,0}^k)_i + (\eta\overline{\rho_C^k})^2(L_{2,0}^k)_i \right] + \\ - \gamma \left[ (1 - \alpha)\overline{\rho_C^k} + \alpha\eta\overline{\rho_C^k} + \alpha^2 (\overline{L_{0,0}^k} + \eta\overline{\rho_C^k}L_{1,0}^k) \right]$$

Note que, por conta da Observação 3.10 na página 70,  $\overline{L_{2,0}^k} = 0$  e, por conta disso, essa constante não aparece no termo que é multiplicado por  $-\gamma$  na expressão acima. No entanto, não há qualquer substituição possível para  $(L_{2,0}^k)_i$  e, logo, ela continua aparecendo na expressão de  $g_C^i(\alpha)$ .

Sejam as constantes

$$\begin{aligned} \zeta_i &= (L_{0,0}^k)_i - \gamma\overline{L_{0,0}^k}, \\ \chi_i &= \eta\overline{\rho_C^k} \left( (L_{1,0}^k)_i - \gamma\overline{L_{1,0}^k} \right), \\ \xi_i &= (\eta\overline{\rho_C^k})^2(L_{2,0}^k)_i. \end{aligned} \tag{4.30}$$

Usando o fato de que o ponto atual pertence à vizinhança  $\mathcal{N}_{-\infty}(\gamma, \beta)$ , e as definições da Equação (4.30) rescrevemos  $g_C^i(\alpha)$  como

$$\begin{aligned} g_C^i(\alpha) &= \underbrace{(1 - \alpha)((\rho_C^k)_i - \gamma\overline{\rho_C^k})}_{\geq 0} + (1 - \gamma)\eta\overline{\rho_C^k}\alpha + (\zeta_i + \chi_i + \xi_i)\alpha^2 \\ &\geq (1 - \gamma)\eta\overline{\rho_C^k}\alpha + (\zeta_i + \chi_i + \xi_i)\alpha^2 \\ &\geq (1 - \gamma)\eta\overline{\rho_C^k}\alpha - (|\zeta_i| + |\chi_i| + |\xi_i|)\alpha^2 \\ &= \alpha \left[ (1 - \gamma)\eta\overline{\rho_C^k} - (|\zeta_i| + |\chi_i| + |\xi_i|)\alpha \right] = h^i(\alpha), \end{aligned}$$

em que  $h^i$  é uma quadrática côncava em função de  $\alpha$  com uma raiz nula e uma positiva.

Vamos agora obter limitantes para  $|\zeta_i|$ ,  $|\chi_i|$  e para  $|\xi_i|$ .

Considerando que nas substituições que, seguem, as definições dos vetores  $L_{\ell,j}^k$  dadas em (3.23)

fazem com que sejam válidas

$$\left| (L_{\ell,j}^k)_i \right| \leq \|L_{\ell,j}^k\|_1 \quad \text{e} \quad \left| \overline{L_{\ell,j}^k} \right| \leq \frac{1}{n} \|L_{\ell,j}^k\|_1. \quad (4.31)$$

Mais que isso, já que  $\gamma/n < 1$ , obtemos

$$\left| (L_{\ell,j}^k)_i - \gamma \overline{L_{\ell,j}^k} \right| \leq \left| (L_{\ell,j}^k)_i \right| + \left| \gamma \overline{L_{\ell,j}^k} \right| \leq \|L_{\ell,j}^k\|_1 + \frac{\gamma}{n} \|L_{\ell,j}^k\|_1 \leq 2 \|L_{\ell,j}^k\|_1. \quad (4.32)$$

Para o limitante de  $|\zeta_i|$  note que por conta de (4.30) e também pela desigualdade dada em (4.32), segue que

$$|\zeta_i| \leq 2 \|L_{0,0}^k\|_1 \leq 2 \|(\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k\|_1. \quad (4.33)$$

Para  $|\chi_i|$ , considere que  $\sigma = \hat{\sigma} = 0$ , que valem as definições dadas em (3.17) e que  $\mu = \hat{\mu} = \overline{\eta \rho_C^k}$  está fixado. Assim, temos que  $\Delta w^c = \hat{\mu} \Delta w^\mu + \hat{\sigma} \Delta w^\sigma = \overline{\eta \rho_C^k} \Delta w^\mu$  e por isso

$$\overline{\eta \rho_C^k} (\Delta x^\mu)^k = (\Delta x^c)^k \quad \text{e} \quad \overline{\eta \rho_C^k} (\Delta z^\mu)^k = (\Delta z^c)^k.$$

Assim,  $L_{1,0}^k = (\Delta x^{\text{af}})^k (\Delta z^\mu)^k + (\Delta z^{\text{af}})^k (\Delta x^\mu)^k = (\Delta x^{\text{af}})^k (\Delta z^c)^k + (\Delta z^{\text{af}})^k (\Delta x^c)^k$  e, novamente, usando (4.32) segue que

$$|\chi_i| = \left| \overline{\eta \rho_C^k} \left( (L_{1,0}^k)_i - \gamma \overline{L_{1,0}^k} \right) \right| \leq 2 \|(\Delta x^{\text{af}})^k (\Delta z^c)^k + (\Delta x^c)^k (\Delta z^{\text{af}})^k\|_1. \quad (4.34)$$

Finalmente para o limitante de  $|\xi_i|$ , se utilizarmos a equação (3.17), novamente com as escolhas  $\hat{\mu}$  e  $\hat{\sigma}$  acima, temos

$$(\Delta x^c)^k (\Delta z^c)^k = (\overline{\eta \rho_C^k} \Delta x^\mu) (\overline{\eta \rho_C^k} \Delta z^\mu) = (\overline{\eta \rho_C^k})^2 (L_{2,0}^k).$$

Portanto segue que

$$|\xi_i| = |(\eta \overline{\rho_C^k})^2 (L_{2,0}^k)_i| = |((\Delta x^c)^k (\Delta z^c)^k)_i| \leq \|(\Delta x^c)^k (\Delta z^c)^k\|_1. \quad (4.35)$$

Agora note que, para  $\alpha > 0$  e para todo  $i = 1, \dots, n$ , se  $h^i(\alpha) \geq 0$  então  $g_C^i(\alpha) \geq 0$ .

Com efeito, a única raiz positiva de  $h^i$  é dada por

$$\alpha_C^i = \frac{(1 - \gamma) \eta \overline{\rho_C^k}}{|\zeta_i| + |\chi_i| + |\xi_i|} \quad (4.36)$$

e  $h^i(\alpha) \geq 0$  sempre que  $\alpha \in [0, \alpha_C^i]$ .

Utilizando (4.33), (4.34) e (4.35) é possível encontrar um limitante para o denominador de (4.36). Primeiro considere que podemos escrever a desigualdade

$$|\zeta_i| + |\chi_i| + |\xi_i| \leq 2 \|(\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k\|_1 + 2 \|(\Delta x^{\text{af}})^k (\Delta z^c)^k + (\Delta x^c)^k (\Delta z^{\text{af}})^k\|_1 + \|(\Delta x^c)^k (\Delta z^c)^k\|_1.$$

Os Lemas 4.5, 4.6 e 4.7 fazem com que a desigualdade acima se transforme em

$$|\zeta_i| + |\chi_i| + |\xi_i| \leq \frac{2(1 + 2\varsigma^4)^2}{\gamma} n^2 \overline{\rho_C^k} + \frac{4(1 + 2\varsigma^4) [\eta\gamma + \hat{\sigma}(1 + 2\varsigma^4)^2]}{\gamma^2} n^3 \overline{\rho_C^k} + \frac{[\eta\gamma + \hat{\sigma}(1 + 2\varsigma^4)^2]^2}{2\gamma^3} n^4 \overline{\rho_C^k}.$$

Como  $\hat{\sigma} = 0$  e  $n^2 \leq n^3 \leq n^4$  e, definindo  $t = (1 + 2\varsigma^4)$ , temos que

$$\begin{aligned} |\zeta_i| + |\chi_i| + |\xi_i| &\leq \left[ \frac{2(1 + 2\varsigma^4)^2}{\gamma} + \frac{4\eta(1 + 2\varsigma^4)}{\gamma} + \frac{\eta^2}{2\gamma} \right] n^4 \overline{\rho_C^k} \\ &= \frac{1}{\gamma} [2t^2 + 4\eta t + \eta^2/2] n^4 \overline{\rho_C^k} \end{aligned} \quad (4.37)$$

Como  $\hat{\alpha}_C^i$ , dada em (4.5), é o maior número em  $(0, 1]$  tal que  $g_C^i(\alpha) \geq 0$  para  $\alpha \leq \hat{\alpha}_C^i$ , e além

disso para todo  $\alpha$  vale  $g_C^i(\alpha) \geq h^i(\alpha)$ , claramente  $\hat{\alpha}_C^i \geq \alpha_C^i$  para todo  $i = 1, \dots, n$ .

Por conta disso, e a partir de (4.36) e de (4.37) temos que

$$\begin{aligned} \hat{\alpha}_C^i \geq \alpha_C^i &\geq \frac{(1-\gamma)\eta\overline{\rho_C^k}}{\frac{1}{\gamma}(2t^2 + 4\eta t + \eta^2/2)\overline{\rho_C^k}n^4} \\ &= \frac{\gamma(1-\gamma)\eta}{2t^2 + 4\eta t + \eta^2/2} \frac{1}{n^4} = \frac{\delta_1}{n^4}, \end{aligned}$$

em que a substituição de  $t$  na penúltima igualdade acima nos dá a Equação (4.29).

O lema fica provado considerando que, para algum  $j \in \{1, \dots, n\}$ ,  $\alpha_C^j = \hat{\alpha}_C$ . ■

**Lema 4.9.** *Seja  $\hat{\alpha}_L$  dado em (4.5). Então*

$$\hat{\alpha}_L \geq \delta_2/n^2,$$

em que

$$\delta_2 = \frac{\gamma\eta}{(1 + 2\zeta^4)^2 + 2(1 + 2\zeta^4)\eta} \quad (4.38)$$

é uma constante independente de  $n$ .

*Demonstração.* Usando as mesmas substituições de  $\hat{\mu}$  e  $\hat{\sigma}$ , a função  $g_L$  dada em (3.45b) torna-se uma função que depende apenas de  $\alpha$ , nos seguintes termos

$$g_L(\alpha) = (1 - \alpha) \left( \overline{\rho_C^k} - \beta_L \nu_k \right) + \alpha \eta \overline{\rho_C^k} + \alpha^2 \left( \overline{L_{0,0}^k} + \eta \overline{\rho_C^k} \overline{L_{1,0}^k} \right).$$

Novamente, levando em conta a Observação 3.10,  $\overline{L_{2,0}^k} = 0$ .

Usando o fato de que o ponto atual pertence à vizinhança segue que

$$\begin{aligned} g_L(\alpha) &= (1 - \alpha) \underbrace{\left( \overline{\rho_C^k} - \beta_L \nu_k \right)}_{\geq 0} + \alpha \eta \overline{\rho_C^k} + \alpha^2 \left( \overline{L_{0,0}^k} + \eta \overline{\rho_C^k} \overline{L_{1,0}^k} \right) \\ &\geq \alpha \left[ \eta \overline{\rho_C^k} - \alpha \left( \left| \overline{L_{0,0}^k} \right| + \left| \eta \overline{\rho_C^k} \overline{L_{1,0}^k} \right| \right) \right]. \end{aligned}$$

Note que  $\hat{\alpha}_L \in (0, 1]$  – definido em (4.5) – é o maior número tal que  $g_L(\alpha) \geq 0$ , para  $\alpha \in [0, \hat{\alpha}_L]$ .

Por conta da última inequação acima, temos que

$$\hat{\alpha}_L \geq \frac{\overline{\eta\rho_C^k}}{\overline{L_{0,0}^k} + \overline{\eta\rho_C^k L_{1,0}^k}}. \quad (4.39)$$

Novamente, como na demonstração do Lema 4.8, usaremos (4.31) para obter

$$\overline{L_{0,0}^k} = \frac{1}{n} \|L_{0,0}^k\|_1 \leq \frac{1}{n} \|(\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k\|_1. \quad (4.40)$$

Além disso,  $\overline{\eta\rho_C^k} (\Delta x^\mu)^k = (\Delta x^c)^k$  e  $\overline{\eta\rho_C^k} (\Delta z^\mu)^k = (\Delta z^c)^k$ . Assim, segue que

$$\begin{aligned} \overline{\eta\rho_C^k L_{1,0}^k} &= \frac{1}{n} \|\overline{\eta\rho_C^k} L_{1,0}^k\|_1 \\ &= \frac{1}{n} \|(\Delta x^{\text{af}})^k (\Delta z^c)^k + (\Delta x^c)^k (\Delta z^{\text{af}})^k\|_1. \end{aligned} \quad (4.41)$$

Dos Lemas 4.7 e 4.5 em conjunto com as Inequações (4.39), (4.40) e (4.41) derivamos

$$\begin{aligned} \hat{\alpha}_L &\geq \frac{\overline{\eta\rho_C^k}}{\frac{1}{n} \left( \|(\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k\|_1 + \|(\Delta x^{\text{af}})^k \Delta z^c \Delta x^c (\Delta z^{\text{af}})^k\|_1 \right)} \\ &\geq \frac{\overline{\eta\rho_C^k}}{\frac{1}{n} \left( \frac{(1+2\zeta^4)^2}{\gamma} n^2 \overline{\rho_C^k} + \frac{2(1+2\zeta^4) [\eta\gamma + \hat{\sigma}(1+2\zeta^4)^2]}{\gamma^2} n^3 \overline{\rho_C^k} \right)} \\ &= \left( \frac{\gamma\eta}{(1+2\zeta^4)^2 + 2(1+2\zeta^4)\eta} \right) \frac{1}{n^2} = \frac{\delta_2}{n^2}, \end{aligned}$$

em que  $\delta_2$  é exatamente o dado na Equação (4.38) ■

Agora podemos enunciar e demonstrar o Teorema que mostra que a sequência  $\{\varphi_k\}$  gerada pelo nosso Algoritmo é convergente.

**Teorema 4.10** (Taxa de Convergência do Algoritmo 3.1). *A sequência  $\{\varphi_k\}$  gerada pelo Algo-*

ritmo 3.1 é tal que

$$\varphi_{k+1} \leq \left(1 - \frac{\hat{\delta}}{n^4}\right) \varphi_k, \quad (4.42)$$

para todo  $k$ , em que  $0 < \hat{\delta} < 1$  e  $\hat{\delta}$  é independente de  $n$ .

*Demonstração.* Note que  $(1 - \gamma) < 1$ , que  $(2t^2 + 4\eta t + \eta^2/2) > (t^2 + 2\eta t)$ , para  $t = (1 + 2\zeta^4)$  e  $\eta > 0$ , e que  $n^4 > n^2$ . Levando isso em conta em conjunto com os Lemas 4.8 e 4.9 e por conta da Equação (4.6), podemos escolher  $\hat{\alpha} = \delta_1/n^4$ , com  $\delta_1$  dado em (4.29).

Precisamos primeiro mostrar que para a função  $\theta$  dada em (4.2),  $\theta(\hat{\alpha}) = \mathcal{O}(1/n^4)$ . Para isso, novamente, seja  $t = (1 + 2\zeta^4)$  e  $t_1 = \frac{t}{\gamma}$ .

Considere agora que pelas definições dadas em (3.23) em conjunto com as Equações (4.40) e (4.41) e ainda os Lema 4.7 e 4.5 derivamos

$$-\overline{L_{0,0}^k} \geq -\frac{1}{n} \left\| (\Delta x^{\text{af}})^k (\Delta z^{\text{af}})^k \right\|_1 \geq -\frac{(1 + 2\zeta^4)^2}{\gamma} n \overline{\rho_C^k} \geq -\gamma t_1^2 n \overline{\rho_C^k}$$

e

$$-\eta \overline{\rho_C^k} \overline{L_{1,0}^k} \geq -\frac{1}{n} \left\| (\Delta x^{\text{af}})^k (\Delta z^{\text{c}})^k + (\Delta x^{\text{c}})^k (\Delta z^{\text{af}})^k \right\|_1 \geq -\frac{2(1 + 2\zeta^4)\eta}{\gamma} n^2 \overline{\rho_C^k} \geq -2\eta t_1 n^2 \overline{\rho_C^k}.$$

Daí, como  $-\frac{\overline{\rho_C^k}}{\nu_k \overline{\rho_{L_0}} + \overline{\rho_C^k}} \geq -1$ , temos que

$$\begin{aligned} \theta(\hat{\alpha}) &= \frac{1}{\nu_k \overline{\rho_{L_0}} + \overline{\rho_C^k}} \left[ (\nu_k \overline{\rho_{L_0}} + \overline{\rho_C^k}) \hat{\alpha} - \eta \overline{\rho_C^k} \hat{\alpha} - \hat{\alpha}^2 (\overline{L_{0,0}^k} + \eta \overline{\rho_C^k} \overline{L_{1,0}^k}) \right] \\ &\geq \frac{1}{\nu_k \overline{\rho_{L_0}} + \overline{\rho_C^k}} \left[ (\nu_k \overline{\rho_{L_0}} + \overline{\rho_C^k}) \hat{\alpha} - \eta \overline{\rho_C^k} \hat{\alpha} - \hat{\alpha}^2 (\gamma t_1^2 n \overline{\rho_C^k} + 2\eta t_1 n^2 \overline{\rho_C^k}) \right] \\ &\geq \hat{\alpha} - \frac{\overline{\rho_C^k}}{\nu_k \overline{\rho_{L_0}} + \overline{\rho_C^k}} \left[ \eta \hat{\alpha} + \hat{\alpha}^2 t_1 (\gamma t_1 n + 2\eta n^2) \right] \\ &\geq \hat{\alpha} \left[ 1 - \eta - \hat{\alpha} t_1 (\gamma t_1 n + 2\eta n^2) \right]. \end{aligned}$$

Substituindo  $\hat{\alpha}$  por  $\delta_1/n^4$  na desigualdade acima, segue que

$$\begin{aligned}\theta(\hat{\alpha}) &\geq \frac{\delta_1}{n^4} \left[ 1 - \eta - \frac{\delta_1}{n^4} t_1 (\gamma t_1 n + 2\eta n^2) \right] \\ &\geq \left[ (1 - \eta)\delta_1 - \delta_1^2 t_1 (\gamma t_1 + 2\eta) \right] \frac{1}{n^4} = \frac{\hat{\delta}}{n^4},\end{aligned}\quad (4.43)$$

com

$$\hat{\delta} = (1 - \eta)\delta_1 - \delta_1^2 t_1 (\gamma t_1 + 2\eta) = (1 - \eta)\delta_1 - \frac{\delta_1^2}{\gamma} \left( (1 + 2\varsigma^4)^2 + 2\eta(1 + 2\varsigma^4) \right). \quad (4.44)$$

Precisamos verificar que  $0 < \hat{\delta} < 1$ . Com efeito, utilizando a definição de  $\delta_1$  dada em (4.29) e de  $t$  dada acima, e substituindo em (4.44), encontramos o valor de  $\hat{\delta}$  a depender de  $\gamma$  e  $\eta$ , ou seja,

$$\hat{\delta} = (2\gamma\eta) \frac{(4t^2 + (8t - 6t^2 - 2t^2\gamma)\eta - (1 - 4t - 4t\gamma))\eta^2 - \eta^3}{(4t^2 + 8t\eta + \eta^2)^2}.$$

Para o primeiro termo do lado direito da igualdade acima, temos que  $0 < (2\gamma\eta) < 1$ . Quanto ao segundo termo, note que o denominador é sempre maior que o numerador, para qualquer valor de  $\gamma$  e  $\eta$ . Basta mostrar que o numerador deste segundo termo é positivo. Com efeito, tal numerador é o polinômio

$$p(\eta, \gamma) = 4t^2 + (8t - 6t^2 + 2t^2\gamma)\eta + (1 - 4t - 4t\gamma)\eta^2 - \eta^3.$$

Considerando que  $t \geq 3$ , pois  $\varsigma \geq 1$ , e que  $\sqrt{\gamma} < \eta < 1$ , para  $0 < \gamma \leq 1/10$ , segue que

$$p(\sqrt{1/10}, \gamma) > 0 \text{ e } p(1, \gamma) < 0.$$

Sob essas condições, escolhido  $\gamma$ ,  $p$  um é polinômio de grau 3 em  $\eta$ . Pelo Teorema do Valor Intermediário de Bolzano [3, Teorema 5.3.7], existe pelo menos uma raiz de  $p$  no intervalo

$(\sqrt{1/10}, 1)$ .

Seja  $\eta_1$  a menor dessas raízes. Nesse caso, por continuidade de  $p$ , para qualquer  $\eta < \eta_1$  e  $\gamma$  escolhido como acima, temos que  $p(\eta, \gamma) > 0$ .

Com isso, concluímos que, para  $0 < \gamma \leq 1/10$  e  $\sqrt{\gamma} < \eta < \eta_1$ , vale  $0 < \hat{\delta} < 1$  sempre que  $\eta < \eta_1 < 1$ .

Para finalizar, usando (4.43) e por conta da relação existente entre  $\varphi_{k+1}$  e  $\varphi_k$  dada em (4.3) obtemos, para esse  $\hat{\alpha}$

$$\varphi_{k+1} = (1 - \theta(\hat{\alpha})) \varphi_k \leq \left(1 - \frac{\hat{\delta}}{n^4}\right) \varphi_k,$$

como queríamos demonstrar. ■

O Teorema 4.10 significa que a relação (4.4) é válida. Portanto, a taxa de convergência da sequência  $\{\varphi_k\}$  é Q-linear. Usando esse fato, podemos agora enunciar e provar o Teorema de complexidade do número de iterações.

**Teorema 4.11** (Complexidade do Algoritmo 3.1). *Seja  $0 < \varepsilon < 1$  dado. Suponha que a sequência  $\{\varphi_k\}$  seja gerada pelo Algoritmo 3.1. Então existe um índice*

$$K_\varepsilon = \mathcal{O}\left(n^4 \ln \frac{1}{\varepsilon}\right),$$

*tal que  $\varphi_k \leq \varepsilon$  para todo  $k \geq K_\varepsilon$ .*

*Demonstração.* Para mostrarmos a complexidade polinomial do algoritmo, aplicamos o logaritmo natural em ambos os lados da inequação (4.42) dada no Teorema 4.10, obtendo

$$\ln \varphi_{k+1} \leq \ln \left(1 - \frac{\hat{\delta}}{n^4}\right) + \ln \varphi_k.$$

Usando um argumento indutivo segue que

$$\ln \varphi_k \leq k \ln \left( 1 - \frac{\hat{\delta}}{n^4} \right) + \ln \varphi_0 \quad (4.45)$$

Para

$$\kappa \geq \frac{\ln(1/\varphi_0)}{\ln(\varepsilon)}$$

temos que

$$\varphi_0 \leq \frac{1}{\varepsilon^\kappa}$$

Assim, utilizando esta última inequação em (4.45), obtemos

$$\ln \varphi_k \leq k \ln \left( 1 - \frac{\hat{\delta}}{n^4} \right) + \kappa \ln \frac{1}{\varepsilon}.$$

Conforme Wright [77, Lema 4.1, p. 68], temos que  $\ln(1+r) \leq r$ , sempre que  $r > -1$ . Assim

$$\ln \varphi_k \leq k \left( -\frac{\hat{\delta}}{n^4} \right) + \kappa \ln \frac{1}{\varepsilon}.$$

Para que o critério de convergência  $\varphi_k \leq \varepsilon$  seja satisfeito, devemos garantir que

$$k \left( -\frac{\hat{\delta}}{n^4} \right) + \kappa \ln \frac{1}{\varepsilon} \leq \ln \varepsilon.$$

De fato, tal inequação é válida para

$$k \geq \frac{n^4}{\hat{\delta}} (1 + \kappa) \ln \frac{1}{\varepsilon},$$

o que termina a demonstração. ■

O Teorema 4.11 implica que, para um  $k$  suficientemente grande, o valor de  $\varphi$  será tão pequeno

quanto se deseja. No entanto, tal resultado ainda não implica que algum critério de parada usual de Métodos de Pontos Interiores é satisfeito. Porém é possível garantir que o Critério de Parada dado em (2.32) seja satisfeito para uma escolha particular de  $\varepsilon$ . Tal garantia, dada pelo Corolário a seguir, permite concluir que a sequência  $\left\{ \left\| (r_P^k, r_D^k) \right\| \right\}$  tem taxa de convergência R-linear [59], já que tal sequência estará limitada superiormente pela sequência  $\{\varphi_k\}$ , que é Q-linear, como vimos no Teorema 4.10.

**Corolário 4.12.** *Suponha que escolhamos  $\text{tol} > 0$  e  $\varepsilon > 0$  tal que*

$$\varepsilon < \min \left\{ \frac{\text{tol}(1 + \|b\|)}{m + n}, \frac{\text{tol}(1 + \|c\|)}{m + n}, \frac{\text{tol}}{n} \right\}. \quad (4.46)$$

*Se para  $k$  suficientemente grande, a sequência  $\{\varphi_k\}$  gerada pelo Algoritmo 3.1 é tal que  $\varphi_k \leq \varepsilon$ , então o critério de parada dado na Equação (2.32) é satisfeito.*

*Demonstração.* Pelas Definições 3.2 e 3.7, vale

$$\varphi_k = \frac{\|\rho_L^k\|_1}{m + n} + \frac{(x^k)^T z^k}{n} = \frac{\|H_P(Ax^k - b)\|_1}{m + n} + \frac{\|H_D(A^T y^k + z^k - c)\|_1}{m + n} + \frac{(x^k)^T z^k}{n}. \quad (4.47)$$

Além disso, por equivalência de normas e pela definição de  $H_P$  e  $H_D$  dadas na Equação (3.3), temos que  $\|Ax^k - b\| \leq \|Ax^k - b\|_1 = \|H_P(Ax^k - b)\|_1$  e que  $\|H_D(A^T y^k + z^k - c)\| \leq \|A^T y^k + z^k - c\|_1 = \|H_D(A^T y^k + z^k - c)\|_1$ .

Ademais, se o Algoritmo 3.1 converge, existe  $k$  suficientemente grande tal que  $\varphi_k < \varepsilon$ . Assim, cada um dos termos da última parte de (4.47) é menor que  $\varepsilon$ .

Daí, pela Equação (4.46), segue que

$$\frac{\|Ax^k - b\|}{1 + \|b\|} \leq \frac{\|H_P(Ax^k - b)\|_1}{1 + \|b\|} < \frac{\varepsilon(m + n)}{1 + \|b\|} < \frac{\text{tol}(1 + \|b\|)}{(m + n)} \frac{(m + n)}{(1 + \|b\|)} = \text{tol},$$

e

$$\frac{\|A^T y^k + z^k - c\|}{1 + \|c\|} \leq \frac{\|H_D(A^T y^k + z^k - c)\|_1}{1 + \|c\|} < \frac{\varepsilon(m+n)}{1 + \|c\|} < \frac{\text{tol}(1 + \|c\|)}{(m+n)} \frac{(m+n)}{(1 + \|c\|)} = \text{tol}.$$

Além disso, como  $1 + |c^T x^k| \geq 1$ , temos que

$$\frac{(x^k)^T z^k}{1 + |c^T x^k|} < \frac{n \cdot \varepsilon}{1 + |c^T x^k|} \leq n \cdot \varepsilon < n \frac{\text{tol}}{n} = \text{tol}. \quad \blacksquare$$

No presente capítulo, estabelecemos os resultados de complexidade polinomial e convergência do Método de Escolha Otimizada de Parâmetros (MEOP), em relação à função de mérito que estamos usando. Isso foi feito através das ferramentas de análise de convergência utilizadas de praxe em Métodos de Pontos Interiores. Além disso, garantimos que é possível escolher um  $\varepsilon > 0$ , tal que, se a função de mérito em uma dada iteração é limitada superiormente por esse  $\varepsilon$ , o critério de parada usado em nossa implementação também é satisfeito. Nesse sentido, do ponto de vista teórico, garantimos que o MEOP converge assintoticamente para pontos com factibilidade e otimalidade, desde que o ponto inicial satisfaça a Condição 4.2. Nos capítulos que seguem, mostramos alguns detalhes da implementação de nosso método bem como seus resultados numéricos.



# Capítulo 5

## Implementação

“The best programs are written so that computing machines can perform them quickly and so that human beings can understand them clearly.”

---

*(Donald Knuth)*

Neste capítulo, fazemos alguns comentários sobre a implementação do Método de Escolha Otimizada de Parâmetros.

O Algoritmo 3.1 foi implementado em C/C++ e o código utilizado foi o de Villas-Bôas et al. [71, 72], que o adaptou ao MEOP. utilizando como base a implementação PCx [11] do algoritmo preditor-corretor de Mehrotra [49]. O PCx pode ser configurado para executar correções de ordem superior de Gondzio [25] e permitimos que fossem feitas no máximo 2 correções através do arquivo de configuração, além do refinamento da solução por gradiente conjugado presente na implementação original. Denominamos essa implementação de referência como PCx-r.

Chamaremos nossa implementação de PCx-EOP. Com efeito, a chamada de PCx-EOP coincide com a da implementação de referência e há a introdução do código de nosso algoritmo dentro do *loop* principal de PCx-r, retornando em seguida para a rotina principal e dando prosseguimento à sua execução usual. Isso foi feito para fazermos uma comparação justa entre nossos resultados

e os que o PCx-r produz.

Assim, herdamos todas as rotinas de préprocessamento – dentre as quais leitura de dados, condicionamento e escalonamento, incluindo escalonamento de Curtis e Reid [10], reordenação de linhas e colunas, tratamento de colunas densas e fatoração simbólica de Cholesky –, de ponto inicial, de critério de parada e saída de dados, bem como todas as rotinas de álgebra linear – incluindo a estratégia de Ng e Peyton [58] para resolução de sistemas lineares esparsos via fatoração de Cholesky e refinamento por gradientes conjugados.

Para problemas canalizados, a estrutura de dados do PCx é construída de forma a fazer o papel da matriz  $E$  e das transformações usadas em (1.3) e (1.4), tal que os problemas canalizados tenham sua resolução executada com as mesmas funcionalidades dos problemas não canalizados. Isso significa que todos os passos de PCx-r e PCx-MEOP são executados de modo sem prejuízo por conta das canalizações.

Além disso, para todos os códigos foram usadas as mesmas opções de compilação e o mesmo computador. Com essa estratégia, diferenças nos tempos de CPU ou no número de iterações podem ser atribuídas apenas à implementação de cada algoritmo.

## 5.1 Ponto Inicial

O ponto inicial dado por Mehrotra [49] foi genericamente descrito na seção 2.3.3. Ali, mostramos que tal estratégia encontra o ponto  $(\tilde{x}, \tilde{y}, \tilde{z})$ , que é solução de norma mínima que satisfaz as restrições primais e duais. Tal terna de pontos é encontrada através de

$$\tilde{x} = A^T(AA^T)^{-1}b, \quad \tilde{y} = (AA^T)^{-1}Ac \quad \text{e} \quad \tilde{z} = c - A^T\tilde{y}. \quad (5.1)$$

Além disso, o ponto é transladado para o ortante positivo através do uso de um vetor de

constantes  $\vartheta_x > 0$  e  $\vartheta_z > 0$  tais que

$$(x^0, y^0, z^0) = (\tilde{x} + \vartheta_x e, \tilde{y}, \tilde{z} + \vartheta_z e), \quad (5.2)$$

garantindo que  $(x^0, z^0) > 0$ .

A rotina `InitialPoint` do PCx é responsável por gerar o ponto inicial, e portanto calcular  $\vartheta_x$  e  $\vartheta_z$ . A implementação feita no PCx possui pequenas alterações em relação ao proposto por Mehrotra, mas que não estão documentadas no Manual de Utilização do PCx [12]. Por isso, para fins de clareza, documentaremos como o ponto inicial é de fato encontrado, deixando o mérito de tal estratégia para os autores citados.

Além disso, tal ponto inicial é utilizado em ambos PCx-r e PCx-EOP e a análise de convergência e complexidade relatada no Capítulo 4, em particular a Condição 4.2, continuam válidas ao utilizá-lo para iniciar nosso algoritmo.

O que a rotina `InitialPoint` faz é primeiramente encontrar  $(\tilde{x}, \tilde{y}, \tilde{z})$  conforme a Equação (5.1). Em seguida, são calculadas

$$\tilde{\vartheta}_x = \max \left\{ -1,5 \cdot \min_i \{ \tilde{x}_i \}, 10^{-2} \right\} \quad \text{e} \quad \tilde{\vartheta}_z = \max \left\{ -1,5 \cdot \min_i \{ \tilde{z}_i \}, 10^{-2} \right\}. \quad (5.3)$$

Só então é que se obtém as translações dadas por

$$\vartheta_x = \tilde{\vartheta}_x + 0,5 \cdot \frac{(\tilde{x} + \tilde{\vartheta}_x e)^T (\tilde{z} + \tilde{\vartheta}_z e)}{\|\tilde{z} + \tilde{\vartheta}_z e\|_1} \quad (5.4)$$

e

$$\vartheta_z = \tilde{\vartheta}_z + 0,5 \cdot \frac{(\tilde{x} + \tilde{\vartheta}_x e)^T (\tilde{z} + \tilde{\vartheta}_z e)}{\|\tilde{x} + \tilde{\vartheta}_x e\|_1} \quad (5.5)$$

a fim de finalmente produzir o ponto inicial como em (5.2).

No trabalho original de Mehrotra [49], o menor valor possível para  $\tilde{\vartheta}_x$  e  $\tilde{\vartheta}_z$  dados pela Equ-

ção (5.3) é 0, ao invés de  $10^{-2}$ , que é o valor utilizado pelo PCx original. Em nossos testes, tanto com PCx-r quanto com PCx-EOP, utilizamos  $10^{-1}$ .

## 5.2 Critério de Parada

O PCx tem dois critérios de parada que podem ser escolhidos pelo usuário. Sua documentação [12] declara que é utilizado o critério dado em (2.31). Além disso, é possível trocar o critério de parada relacionado à complementaridade, dentro do código. Conforme explanamos na Seção 2.3.4, uma escolha possível seria a dada na equação (2.32). Porém ao invés do teste dado pela Equação (2.32c) – ou por (2.31c) o que seria razoável –, a implementação do PCx usa

$$\frac{x^T z/n}{1 + |c^T x|} \leq \text{tol},$$

em que `tol` é a tolerância aceita. Essa escolha tem um viés dimensional, já que divide por  $n$  o *gap* de complementaridade, fazendo com que o programa acuse otimalidade antes do que deveria. Isso porque usar esse teste equivale a utilizar o critério dado em (2.32c) na forma  $x^T z / (1 + |c^T x|) \leq n \cdot \text{tol}$ .

Fizemos duas alterações no código, no momento em que testamos o critério de parada, que nos parecem contribuir para um melhor desempenho, pelo menos no que diz respeito à verificação da otimalidade do ponto em questão.

Primeiramente, reutilizamos a rotina `RecomputeDualVariables` do PCx. Tal rotina é utilizada no PCx original somente quando já há a indicação de otimalidade e o programa saiu do *loop* principal. Passamos a descrever a rotina em questão.

Seja  $(\tilde{x}, \tilde{y}, \tilde{z})$  o ponto que o critério de parada do PCx indicou como sendo ótimo. O que a rotina faz é primeiro calcular o vetor  $t = c - A^T \tilde{y}$ . É evidente que, por conta do modo como funcionam os MPI, deveríamos ter  $t \geq 0$  e, mais que isso, pela definição de problema dual, deveríamos ter  $\tilde{z} = t$ .

O código então primeiro testa, para cada  $i = 1, \dots, n$ , se  $t_i < 0$ . Nos casos afirmativos, o programa assinala  $t_i \leftarrow 0$ . Finalmente estabelece-se que  $\tilde{z} \leftarrow t$ , o que finaliza a rotina. Basicamente isso significa que, a rotina garante que a variável dual  $\tilde{z}$ , tenha o valor calculado pela definição do problema dual, e não pelo algoritmo. O teste para verificar se  $t_i$  é negativo serve para detectar possíveis valores inconvenientes, já que  $z^* \geq 0$ .

Pois bem, a alteração que fizemos utiliza os princípios da rotina `RecomputeDualVariables` em toda iteração  $k$ . Neste caso, o que fazemos é encontrar  $t = c - A^T y^k$ . Testamos então se para algum  $i$  ocorre  $t_i < 0$  e em caso afirmativo assinalamos  $t_i \leftarrow 0$ . Com isso, consideramos a terna  $(x^k, y^k, t)$  com a qual realizamos os testes de otimalidade dados por (2.32), porém com as seguintes modificações: o critério (2.32b) é transformado em

$$\frac{\|t\|}{1 + \|c\|} \leq \text{tol},$$

e o critério (2.32c) é transformado em

$$\frac{(x^k)^T t}{1 + |c^T t|} \leq \text{tol}.$$

Se esses testes são satisfeitos, então assinalamos otimalidade no algoritmo e saímos do *loop* principal. Isso é feito porque, como  $t$  não é mais interior,  $(x^k)^T t$  tem possivelmente valor menor que  $(x^k)^T z^k$ . Essa estratégia diz que  $(x^k, y^k, t)$  pode ser um ponto melhor que  $(x^k, y^k, z^k)$ .

Caso algum teste não seja satisfeito, continuamos a execução do programa com o ponto atual sendo  $(x^k, y^k, z^k)$ . Em ambos os casos  $t$  é descartado e o ponto utilizado é sempre  $(x^k, y^k, z^k)$ . Considere que se tivermos otimalidade, `RecomputeDualVariables` será chamada – pois estamos fora do *loop* e o programa assinala  $(x^k, y^k, z^k) \leftarrow (x^k, y^k, t)$ .

Uma outra alteração que fizemos no teste de parada tem relação com o escalonamento dos dados. No manual do PCx [12, p. 20], os problemas `brandy`, `ganges`, `grow15`, `grow22`, `perold`, `pilot.ja`, `pilot`, `pilot4`, `pilot87`, `pilotnov` e `scfxm1` estão indicados como problemas para

os quais houve convergência. No entanto, o resíduo relativo de infactibilidade primal-dual é maior que  $10^{-8}$ . Isso significa que o PCx parou antes do que deveria. De fato, descobrimos que o teste de parada, que é feito com os dados escalonados, era válido, porém, ao desescalonar, o critério já não era mais satisfeito. Além disso, obviamente o PCx faz o relatório de saída com os dados desescalonados.

A fim de evitar isso, usamos a rotina `ComputeUnscaledInfeasibilities` que testa a infactibilidade relativa dos dados desescalonados. Essa rotina tem como entrada cópia dos ponteiros da solução atual do PCx e logo não estamos modificando tal solução.

O teste de infactibilidade primal-dual é feito então com os dados desescalonados, através da mesma rotina do PCx que faz o desescalonamento no final do algoritmo. Se para os dados desescalonados o critério for satisfeito, então indicamos otimalidade. Caso contrário, continuamos a execução do algoritmo, com os dados escalonados.

O critério de parada usado na implementação em PCx-r e PCx-EOP é o dado na Equação (2.32), porém com as alterações relatadas acima. Note que PCx não convergia para `greenbeb` e `wood1p`, porém, com essas alterações, PCx-r converge para ambos os problemas.

### 5.3 Solução do subproblema de otimização

Em cada iteração de nosso método, precisamos resolver o problema dado em (3.46), isto é,

$$\begin{array}{ll}
 \min_{(\alpha, \mu, \sigma)} & \hat{\varphi}(\alpha, \mu, \sigma) \\
 \text{s. a.} & \left\{ \begin{array}{l} g_C^i(\alpha, \mu, \sigma) \geq 0 \quad \forall i = 1, \dots, n \\ g_L(\alpha, \mu, \sigma) \geq 0 \\ 0 \leq (\alpha, \mu, \sigma) \leq u, \end{array} \right. \quad .
 \end{array} \tag{5.6}$$

em que  $\varphi$  bem como  $g_C^i$  e  $g_L$  são polinômios de grau total 6 nas variáveis  $(\alpha, \mu, \sigma)$ . Este é um Problema de Otimização de Polinômios (POP) e é considerado um problema não linear altamente difícil de ser resolvido, alguns deles, NP-Hard [46, 47, 56].

Testamos algumas implementações especializadas em POP, dentre elas GloptiPoly [33] e SparsePOP [74], as quais se baseiam na transformação do POP em um problema de Programação Semidefinida [46]. Nenhuma delas, entretanto, foi capaz de resolver os subproblemas. A causa provável disso é quantidade de restrições do POP, que é  $n + 1$ , em que  $n$  é o número de variáveis do PL original.

O subproblema foi resolvido utilizando a implementação de Villas-Bôas et al. [71, 72], especialmente desenvolvida para resolver um problema com as mesmas características que o nosso. Tal estratégia é brevemente resumida a seguir, adaptada à nossa notação.

Seja

$$\Omega = \{(\alpha, \mu, \sigma) : g_C^i(\alpha, \mu, \sigma) \geq 0, \forall i = 1, \dots, n, g_L(\alpha, \mu, \sigma) \geq 0, 0 \leq (\alpha, \mu, \sigma) \leq u\}$$

o conjunto factível de (5.6). Note que pelas demonstrações que fizemos no Capítulo 4,  $\Omega$  é não-vazio. Considere então o problema alternativo

$$\min_{\mu} \left\{ \min_{\sigma} \left\{ \min_{\alpha} \{ \hat{\varphi}(\alpha, \mu, \sigma) : (\alpha, \mu, \sigma) \in \Omega \} \right\} \right\}, \quad (5.7)$$

ou mais convenientemente

$$\min_{\mu} \Psi(\mu),$$

em que  $\Psi(\mu) = \min_{\sigma} \Phi(\mu, \sigma)$  e

$$\Phi(\mu, \sigma) = \begin{cases} \min_{\alpha} \hat{\varphi}(\alpha, \mu, \sigma) \\ \text{s. a. } (\alpha, \mu, \sigma) \in \Omega \end{cases}.$$

Primeiramente, vamos verificar a relação existente entre os mínimos de (5.6) e de (5.7). Sejam  $(\alpha^*, \mu^*, \sigma^*)$  uma solução global do primeiro problema e  $(\bar{\alpha}, \bar{\mu}, \bar{\sigma})$  uma solução global de segundo. Então por definição  $\hat{\varphi}(\alpha^*, \mu^*, \sigma^*) \leq \hat{\varphi}(\bar{\alpha}, \bar{\mu}, \bar{\sigma})$ . Por outro lado

$$\hat{\varphi}(\bar{\alpha}, \bar{\mu}, \bar{\sigma}) = \min_{\alpha} \{ \min_{\sigma} \Phi(\bar{\mu}, \bar{\sigma}) \} \leq \Phi(\mu^*, \sigma^*) \leq \hat{\varphi}(\alpha^*, \mu^*, \sigma^*).$$

Assim, os dois problemas têm o mesmo mínimo. A segunda formulação, entretanto, permite encontrar aproximações numéricas do minimizador de  $\hat{\varphi}$  de maneira mais fácil.

Para esse problema alternativo, um gráfico típico de  $\Phi(\mu, \sigma)$  é dado na Figura 1, tomada na iteração 4 do problema `bn11`. O aspecto desse gráfico – com um *zoom* conveniente perto do mínimo – acaba se repetindo em todos os problema da `NETLIB-108` – veja Capítulo 6 – que estudamos, o que sugere que  $\Phi$  tenha derivadas contínuas por partes próximo ao mínimo.

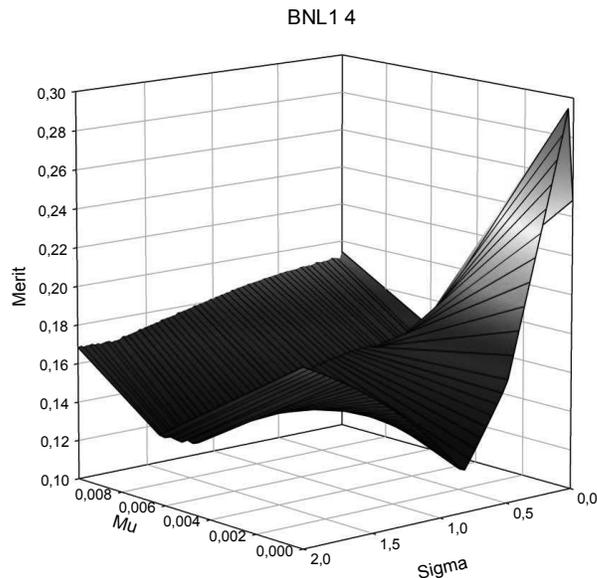


Figura 1: Gráfico de  $\Phi(\mu, \sigma)$  para  $\alpha$  fixo de `bn11`.

Uma característica interessante desse problema alternativo é que, em quase todos os problemas e iterações, a restrição de  $\Omega$  que estava ativa ao final da resolução do subproblema é exatamente aquela relacionada ao índice de bloqueio indicado pelo teste da razão.

Primeiramente considere um malha construída para o intervalo  $[0, u_\mu]$  no qual  $\mu$  está definido. Usando o fato de que o índice de bloqueio do teste da razão tem relação com a restrição ativa, uma partição  $\mathcal{P} = 0 < \sigma_1 < \sigma_2 < \dots < u_\sigma$  do intervalo  $[0, u_\sigma]$ , para a qual  $\sigma$  está definido, é construída tal que se  $\bar{\sigma} \in I_j = [\sigma_j, \sigma_{j+1}]$ , então existe um índice  $t$  tal que para este  $\bar{\sigma}$ , os testes da razão para as componentes não negativas das variáveis  $(x, z)$  podem ser substituídos por um único teste da razão, usando o índice  $t$ .

Dessa forma, para cada subintervalo  $I_j = [\sigma_j, \sigma_{j+1}]$  define-se o teste da razão principal. Esse teste da razão independente de  $\sigma$  é, segundo Villas-Bôas et al. [71, 72], a ferramenta principal para conseguir resolver o subproblema na medida em que define a partição  $\mathcal{P}$ .

Esse índice  $t$  permite que escolhamos a restrição  $g_C^t(\alpha, \sigma) \geq 0$ , para algum índice  $t$  que é mais restritiva que o teste da razão, sendo então declarada como *restrição principal* de  $I_j$ . De fato, a partição  $\mathcal{P}$  agora é reconstruída tal que cada ponto na fronteira do subintervalo está associado com a interseção de restrições principais, gerando então uma região de relevância.

Essa estratégia leva a descartar restrições que são irrelevantes, reduzindo o número das que precisam ser utilizadas na solução do subproblema. Para cada intervalo  $I_j$ , determina-se um tamanho de passo máximo  $\alpha_j^{\max}$  e o mínimo de  $\hat{\varphi}$  em cada intervalo. Isso pode ser feito de maneira analítica usando condições de primeira ordem, mas na implementação atual tal solução é encontrada usando *splines* cúbicos. Finalmente, o menor dos  $\hat{\varphi}$  encontrados para cada intervalo é declarado como ótimo.

Na prática, esse método funciona porque o tamanho da partição é muito menor que o tamanho dos problemas – para todas as iterações do conjunto NETLIB-108, a partição tem em média tamanho 3,7 com desvio padrão 2,9 e máximo 39. De fato, o tempo de CPU gasto na solução do subproblema para toda NETLIB-108 foi 1,37% do tempo total. Um detalhe é que, em grande parte dos casos, o melhor resultado de  $\hat{\varphi}$  foi encontrado com  $\mu = 0$ .



# Capítulo 6

## Resultados numéricos

“An ounce of practice is worth more  
than tons of preaching.”

---

*(Mahatma Gandhi)*

No presente capítulo descrevemos o conjunto de testes e relatamos os resultados de nosso método, comparando-o com o PCx [11].

O conjunto de testes que vamos utilizar foi retirado da NETLIB<sup>1</sup> [19]. Escolhemos primeiramente os 95 problemas de programação linear factíveis que estão no diretório `/lp/data/` de tal repositório. Além disso, do mesmo repositório escolhemos 12 dos 16 problemas Kennington [40] que se encontram em `/lp/data/kennington/`. Para finalizar, escolhemos um problema – `qap-8` – dos 3 do conjunto QAP, que estão disponíveis no diretório `/lp/generators/qap`. Os problemas QAP são construído via um gerador e aparecem a partir da linearização de um problema de designação quadrática. No total então, temos 108 problemas e, por isso, tal conjunto de testes é chamado aqui de NETLIB-108.

Alguns comentários sobre a escolha de NETLIB-108 cabem aqui. Primeiramente, os 4 problemas Kennington – `kennington-ken-13`, `kennington-ken-18`, `kennington-osa-60` e `kennington-pds-20` – e os 2 QAP – `qap12` e `qap15` – que foram deixados de fora dos testes, o

---

<sup>1</sup>Disponíveis em <http://www.netlib.org>.

foram pois suas estruturas e tamanhos não fazem sentido numa implementação que resolve as equações normais que aparecem em MPI via fatoração de Cholesky.

De fato, tal gama de problemas, originária de problemas reais, é consensual entre os usuários de PL. Com efeito, tais problemas, ou pelo menos parte deles, foram testados por Colombo e Gondzio [9], Gondzio [25], Jarre e Wechs [38], Mehrotra [49] e Mehrotra e Li [50], entre outros pesquisadores com trabalhos importantes da área, e estão inclusive na documentação do PCx [11, 12], com a finalidade de demonstrar os testes de seu desempenho. Nesse sentido e assim como esses autores, pensamos que tal conjunto é capaz de permitir testes adequados de nosso método.

A Tabela 1 mostra os 108 problemas com seus nomes na primeira coluna. As colunas 2 a 4 mostram o número de linhas, colunas e de canalizações de cada problema, após as rotinas de pré-processamento feitas pelo PCx. Indicamos que um problema não é canalizado ao colocar o número 0 na coluna correspondente. A coluna 5 tem o número de elementos não-nulos (NEM) da matriz  $A$  das restrições e a última coluna apresenta a densidade da matriz resultante da decomposição de de Cholesky utilizada em nossa implementação.

Tabela 1: Conjunto de testes NETLIB-108.

| <b>Problema</b> | <b>Linhas</b> | <b>Colunas</b> | <b>Canalizações</b> | <b>NEN</b> | <b>Densidade (<math>\times 10^{-1}</math>)</b> |
|-----------------|---------------|----------------|---------------------|------------|--|
| 25fv47          | 788           | 1843           | 0                   | 33809      | 1,0763   |
| 80bau3b         | 2140          | 11066          | 2968                | 41367      | 0,175 98                                       |
| adlittle        | 55            | 137            | 0                   | 404        | 2,4893   |
| afiro           | 27            | 51             | 0                   | 107        | 2,5652   |
| agg             | 390           | 477            | 0                   | 12297      | 1,5913   |
| agg2            | 514           | 750            | 0                   | 21482      | 1,6068   |
| agg3            | 514           | 750            | 0                   | 21482      | 1,6068   |
| bandm           | 240           | 395            | 0                   | 3936       | 1,3250   |
| beaconfd        | 86            | 171            | 0                   | 820        | 2,1011   |
| blend           | 71            | 111            | 0                   | 913        | 3,4815   |
| bn11            | 610           | 1491           | 0                   | 12089      | 0,633 38                                       |
| bn12            | 1964          | 4008           | 0                   | 81275      | 0,416 32                                       |
| boeing1         | 331           | 697            | 243                 | 5725       | 1,0149   |
| boeing2         | 126           | 265            | 73                  | 2029       | 2,4767   |
| bore3d          | 81            | 138            | 7                   | 1034       | 3,0285   |

Continua na próxima página.

Tabela 1: Conjunto de testes NETLIB-108 (continuação).

| Problema  | Linhas | Colunas | Canalizações | NEN     | Densidade ( $\times 10^{-1}$ ) |
|-----------|--------|---------|--------------|---------|--------------------------------|
| brandy    | 133    | 238     | 0            | 2755    | 3,0397                         |
| capri     | 241    | 436     | 131          | 3962    | 1,3228                         |
| cycle     | 1420   | 2773    | 77           | 56102   | 0,549 41                       |
| czprob    | 671    | 2779    | 0            | 3520    | 0,141 46                       |
| d2q06c    | 2132   | 5728    | 0            | 137349  | 0,599 65                       |
| d6cube    | 403    | 5443    | 0            | 54840   | 6,7285                         |
| degen2    | 444    | 757     | 0            | 16319   | 1,6331                         |
| degen3    | 1503   | 2604    | 0            | 120906  | 1,0638                         |
| df1001    | 5984   | 12143   | 13           | 1638085 | 0,913 25                       |
| e226      | 198    | 429     | 0            | 3229    | 1,5968                         |
| etamacro  | 334    | 669     | 131          | 10843   | 1,9140                         |
| ffffff800 | 322    | 826     | 0            | 9573    | 1,8155                         |
| finnis    | 438    | 935     | 33           | 4984    | 0,496 76                       |
| fit1d     | 24     | 1049    | 1026         | 296     | 9,8611                         |
| fit1p     | 627    | 1677    | 399          | 627     | 0,015 950                      |
| fit2d     | 25     | 10524   | 10500        | 324     | 9,9680                         |
| fit2p     | 3000   | 13525   | 7500         | 3000    | 0,003 330 0                    |
| forplan   | 121    | 447     | 22           | 3304    | 4,4307                         |
| ganges    | 1113   | 1510    | 397          | 29677   | 0,470 15                       |
| gfrd-pnc  | 590    | 1134    | 258          | 2112    | 0,104 40                       |
| greenbea  | 1933   | 4164    | 264          | 49055   | 0,257 40                       |
| greenbeb  | 1932   | 4154    | 268          | 47783   | 0,250 85                       |
| grow7     | 140    | 301     | 280          | 2730    | 2,7143                         |
| grow15    | 300    | 645     | 600          | 6090    | 1,3200                         |
| grow22    | 440    | 946     | 880          | 9058    | 0,913 02                       |
| israel    | 174    | 316     | 0            | 11488   | 7,5314                         |
| kb2       | 43     | 68      | 9            | 503     | 5,2082                         |
| lotfi     | 133    | 346     | 0            | 1369    | 1,4727                         |
| maros-r7  | 2152   | 7440    | 0            | 534188  | 2,3023                         |
| maros     | 655    | 1437    | 0            | 13454   | 0,611 92                       |
| modszk1   | 665    | 1599    | 0            | 10550   | 0,462 10                       |
| nesm      | 654    | 2922    | 1596         | 21776   | 1,0030                         |
| perold    | 593    | 1374    | 259          | 21782   | 1,2220                         |
| pilot.ja  | 810    | 1804    | 331          | 47924   | 1,4485                         |
| pilot.we  | 701    | 2814    | 287          | 15605   | 0,620 86                       |
| pilot     | 1368   | 4543    | 1040         | 200812  | 2,1388                         |
| pilot4    | 396    | 1022    | 247          | 14279   | 1,7959                         |

Continua na próxima página.

Tabela 1: Conjunto de testes NETLIB-108 (continuação).

| Problema | Linhas | Colunas | Canalizações | NEN    | Densidade ( $\times 10^{-1}$ ) |
|----------|--------|---------|--------------|--------|--------------------------------|
| pilot87  | 1971   | 6373    | 1578         | 425654 | 2,1863                         |
| pilotnov | 848    | 2117    | 332          | 46353  | 1,2774                         |
| recipe   | 64     | 123     | 56           | 277    | 1,1963                         |
| sc50a    | 49     | 77      | 0            | 242    | 1,8118                         |
| sc50b    | 48     | 76      | 0            | 231    | 1,7969                         |
| sc105    | 104    | 162     | 0            | 569    | 0,955 99                       |
| sc205    | 203    | 315     | 0            | 1156   | 0,511 78                       |
| scagr7   | 127    | 183     | 0            | 730    | 0,826 46                       |
| scagr25  | 469    | 669     | 0            | 2944   | 0,246 36                       |
| scfxm1   | 305    | 568     | 0            | 4430   | 0,919 65                       |
| scfxm2   | 610    | 1136    | 0            | 9284   | 0,482 61                       |
| scfxm3   | 915    | 1704    | 0            | 14138  | 0,326 81                       |
| scorpion | 340    | 412     | 0            | 2275   | 0,364 19                       |
| scrs8    | 421    | 1199    | 0            | 4477   | 0,481 43                       |
| scsd1    | 77     | 760     | 0            | 1392   | 4,5657                         |
| scsd6    | 147    | 1350    | 0            | 2545   | 2,2875                         |
| scsd8    | 397    | 2750    | 0            | 5879   | 0,720 83                       |
| sctap1   | 284    | 644     | 0            | 2435   | 0,568 59                       |
| sctap2   | 1033   | 2443    | 0            | 11736  | 0,210 28                       |
| sctap3   | 1408   | 3268    | 0            | 16100  | 0,155 32                       |
| seba     | 448    | 901     | 447          | 53711  | 5,3299                         |
| share1b  | 112    | 248     | 0            | 1417   | 2,1700                         |
| share2b  | 96     | 162     | 0            | 1041   | 2,1550                         |
| shell    | 487    | 1451    | 117          | 3983   | 0,315 34                       |
| ship04l  | 292    | 1905    | 0            | 2641   | 0,585 24                       |
| ship04s  | 216    | 1281    | 0            | 1778   | 0,715 88                       |
| ship08l  | 470    | 3121    | 0            | 4442   | 0,380 90                       |
| ship08s  | 276    | 1604    | 0            | 2295   | 0,566 32                       |
| ship12l  | 610    | 4171    | 0            | 5506   | 0,279 55                       |
| ship12s  | 340    | 1943    | 0            | 2507   | 0,404 33                       |
| sierra   | 1212   | 2705    | 2016         | 12862  | 0,166 87                       |
| stair    | 356    | 532     | 6            | 14682  | 2,2889                         |
| standata | 314    | 796     | 96           | 2395   | 0,453 97                       |
| standgub | 314    | 796     | 96           | 2395   | 0,453 97                       |
| standmps | 422    | 1192    | 96           | 3957   | 0,420 70                       |
| stocfor1 | 102    | 150     | 0            | 805    | 1,4494                         |
| stocfor2 | 1980   | 2868    | 0            | 22841  | 0,111 47                       |

Continua na próxima página.

Tabela 1: Conjunto de testes NETLIB-108 (continuação).

| Problema          | Linhas | Colunas | Canalizações | NEN     | Densidade ( $\times 10^{-1}$ ) |
|-------------------|--------|---------|--------------|---------|--------------------------------|
| stocfor3          | 15362  | 22228   | 0            | 177936  | 0,014 430                      |
| truss             | 1000   | 8806    | 0            | 53509   | 1,0602                         |
| tuff              | 257    | 567     | 25           | 7051    | 2,0962                         |
| vtp.base          | 72     | 111     | 32           | 505     | 1,8094                         |
| wood1p            | 171    | 1718    | 0            | 11645   | 7,9064                         |
| woodw             | 708    | 5364    | 0            | 30027   | 1,1839                         |
| kennington-cre-a  | 2994   | 6692    | 0            | 33212   | 0,070 760                      |
| kennington-cre-b  | 5336   | 36382   | 0            | 248629  | 0,172 77                       |
| kennington-cre-c  | 2375   | 5412    | 0            | 28528   | 0,096 940                      |
| kennington-cre-d  | 4102   | 28601   | 0            | 212094  | 0,249 66                       |
| kennington-ken-07 | 1437   | 2613    | 2613         | 10034   | 0,090 220                      |
| kennington-ken-11 | 10085  | 16740   | 16740        | 102906  | 0,019 240                      |
| kennington-osa-07 | 1081   | 25030   | 0            | 28276   | 0,474 69                       |
| kennington-osa-14 | 2300   | 54760   | 0            | 60795   | 0,225 50                       |
| kennington-osa-30 | 4313   | 104337  | 0            | 115081  | 0,121 41                       |
| kennington-pds-02 | 2609   | 7339    | 1914         | 44301   | 0,126 33                       |
| kennington-pds-06 | 9156   | 28472   | 8448         | 589339  | 0,139 51                       |
| kennington-pds-10 | 15648  | 48780   | 15125        | 1687660 | 0,137 21                       |
| qap8              | 912    | 1632    | 0            | 193944  | 4,6526                         |

Os testes computacionais foram efetuados em um computador com processador Intel i7 930 2.8GHz com 12GB de memória RAM em sistema operacional Windows 7 64bits e compilados utilizando Microsoft Visual Studio C++ 2010 com compilador Intel C++ Compiler XE 13.0.

A Tabela 2 mostra o número total de iterações, o tempo de CPU e a saída programa – 0 indica convergência e 3 indica que otimalidade não foi alcançada – para as implementações PCx-r e PCx-EOP.

Tabela 2: Resultados comparativos de PCx-r e PCx-EOP.

| Problema | PCx-r |         |        | PCx-EOP |         |        |
|----------|-------|---------|--------|---------|---------|--------|
|          | Iter. | CPU (s) | Status | Iter.   | CPU (s) | Status |
| 25fv47   | 24    | 0,193   | 0      | 27      | 0,382   | 0      |
| 80bau3b  | 39    | 0,473   | 0      | 38      | 0,882   | 0      |

Continua na próxima página.

Tabela 2: Resultados comparativos de PCx-r e PCx-EOP(continuação).

| Problema | PCx-r |         |        | PCx-EOP |         |        |
|----------|-------|---------|--------|---------|---------|--------|
|          | Iter. | CPU (s) | Status | Iter.   | CPU (s) | Status |
| adlittle | 10    | 0,038   | 0      | 18      | 0,163   | 0      |
| afiro    | 6     | 0,031   | 0      | 12      | 0,111   | 0      |
| agg      | 17    | 0,083   | 0      | 21      | 0,219   | 0      |
| agg2     | 20    | 0,118   | 0      | 23      | 0,268   | 0      |
| agg3     | 19    | 0,115   | 0      | 22      | 0,256   | 0      |
| bandm    | 15    | 0,065   | 0      | 24      | 0,237   | 0      |
| beaconfd | 10    | 0,059   | 0      | 16      | 0,189   | 0      |
| blend    | 8     | 0,033   | 0      | 18      | 0,168   | 0      |
| bnl1     | 33    | 0,169   | 0      | 50      | 0,602   | 3      |
| bnl2     | 29    | 0,551   | 0      | 31      | 0,749   | 0      |
| boeing1  | 21    | 0,099   | 0      | 29      | 0,304   | 0      |
| boeing2  | 12    | 0,053   | 0      | 21      | 0,233   | 0      |
| bore3d   | 14    | 0,053   | 0      | 23      | 0,212   | 0      |
| brandy   | 17    | 0,082   | 3      | 22      | 0,240   | 0      |
| capri    | 17    | 0,071   | 0      | 25      | 0,284   | 0      |
| cycle    | 28    | 0,304   | 0      | 24      | 0,414   | 0      |
| czprob   | 25    | 0,137   | 0      | 29      | 0,376   | 0      |
| d2q06c   | 27    | 0,812   | 0      | 28      | 1,060   | 0      |
| d6cube   | 17    | 0,298   | 0      | 23      | 0,586   | 0      |
| degen2   | 10    | 0,074   | 0      | 11      | 0,147   | 0      |
| degen3   | 23    | 0,637   | 0      | 15      | 0,533   | 0      |
| df1001   | 49    | 46,355  | 0      | 50      | 46,316  | 0      |
| e226     | 17    | 0,073   | 0      | 22      | 0,221   | 0      |
| etamacro | 27    | 0,121   | 0      | 31      | 0,332   | 0      |
| fffff800 | 29    | 0,139   | 0      | 35      | 0,422   | 0      |
| finnis   | 22    | 0,096   | 0      | 29      | 0,323   | 0      |
| fit1d    | 16    | 0,089   | 0      | 23      | 0,348   | 0      |
| fit1p    | 16    | 0,141   | 0      | 27      | 0,426   | 3      |
| fit2d    | 22    | 0,450   | 0      | 35      | 1,151   | 0      |
| fit2p    | 21    | 0,634   | 0      | 31      | 1,050   | 3      |
| forplan  | 20    | 0,087   | 0      | 28      | 0,322   | 0      |

Continua na próxima página.

Tabela 2: Resultados comparativos de PCx-r e PCx-EOP(continuação).

| Problema | PCx-r |         |        | PCx-EOP |         |        |
|----------|-------|---------|--------|---------|---------|--------|
|          | Iter. | CPU (s) | Status | Iter.   | CPU (s) | Status |
| ganges   | 15    | 0,128   | 0      | 21      | 0,313   | 0      |
| gfrd-pnc | 15    | 0,064   | 0      | 19      | 0,199   | 0      |
| greenbea | 49    | 0,511   | 3      | 31      | 0,551   | 3      |
| greenbeb | 38    | 0,401   | 0      | 37      | 0,689   | 0      |
| grow7    | 13    | 0,057   | 0      | 21      | 0,215   | 0      |
| grow15   | 16    | 0,081   | 0      | 25      | 0,276   | 0      |
| grow22   | 17    | 0,101   | 0      | 27      | 0,365   | 0      |
| israel   | 19    | 0,130   | 0      | 21      | 0,232   | 0      |
| kb2      | 10    | 0,039   | 0      | 25      | 0,236   | 0      |
| lotfi    | 14    | 0,056   | 0      | 22      | 0,253   | 0      |
| maros-r7 | 13    | 2,006   | 0      | 20      | 2,959   | 0      |
| maros    | 17    | 0,095   | 0      | 23      | 0,270   | 0      |
| modszk1  | 18    | 0,095   | 0      | 32      | 0,362   | 0      |
| nesm     | 28    | 0,197   | 0      | 29      | 0,463   | 0      |
| perold   | 32    | 0,200   | 0      | 47      | 0,587   | 0      |
| pilot.ja | 33    | 0,390   | 0      | 38      | 0,694   | 0      |
| pilot.we | 44    | 0,286   | 0      | 50      | 0,714   | 0      |
| pilot    | 34    | 1,879   | 0      | 36      | 2,155   | 0      |
| pilot4   | 35    | 0,234   | 0      | 43      | 0,522   | 3      |
| pilot87  | 33    | 4,804   | 0      | 35      | 5,155   | 0      |
| pilotnov | 16    | 0,173   | 0      | 16      | 0,289   | 0      |
| recipe   | 7     | 0,030   | 0      | 15      | 0,148   | 0      |
| sc50a    | 7     | 0,029   | 0      | 14      | 0,131   | 0      |
| sc50b    | 6     | 0,026   | 0      | 10      | 0,093   | 0      |
| sc105    | 9     | 0,037   | 0      | 19      | 0,177   | 0      |
| sc205    | 9     | 0,038   | 0      | 18      | 0,210   | 0      |
| scagr7   | 12    | 0,047   | 0      | 22      | 0,224   | 0      |
| scagr25  | 16    | 0,068   | 0      | 24      | 0,238   | 0      |
| scfxm1   | 16    | 0,068   | 0      | 23      | 0,278   | 0      |
| scfxm2   | 19    | 0,106   | 3      | 27      | 0,356   | 0      |
| scfxm3   | 18    | 0,103   | 0      | 26      | 0,300   | 0      |

Continua na próxima página.

Tabela 2: Resultados comparativos de PCx-r e PCx-EOP(continuação).

| Problema | PCx-r |         |        | PCx-EOP |         |        |
|----------|-------|---------|--------|---------|---------|--------|
|          | Iter. | CPU (s) | Status | Iter.   | CPU (s) | Status |
| scorpion | 10    | 0,063   | 0      | 12      | 0,142   | 0      |
| scrs8    | 21    | 0,091   | 0      | 33      | 0,420   | 0      |
| scsd1    | 8     | 0,035   | 0      | 17      | 0,171   | 0      |
| scsd6    | 10    | 0,046   | 0      | 19      | 0,228   | 0      |
| scsd8    | 9     | 0,048   | 0      | 19      | 0,221   | 0      |
| sctap1   | 13    | 0,055   | 0      | 18      | 0,181   | 0      |
| sctap2   | 12    | 0,069   | 0      | 21      | 0,275   | 0      |
| sctap3   | 13    | 0,082   | 0      | 17      | 0,280   | 0      |
| seba     | 11    | 0,191   | 0      | 14      | 0,329   | 0      |
| share1b  | 17    | 0,093   | 0      | 23      | 0,220   | 0      |
| share2b  | 16    | 0,063   | 0      | 23      | 0,221   | 0      |
| shell    | 20    | 0,114   | 0      | 22      | 0,287   | 0      |
| ship04l  | 11    | 0,051   | 0      | 19      | 0,199   | 0      |
| ship04s  | 11    | 0,048   | 0      | 19      | 0,195   | 0      |
| ship08l  | 13    | 0,067   | 0      | 23      | 0,325   | 0      |
| ship08s  | 10    | 0,045   | 0      | 16      | 0,166   | 0      |
| ship12l  | 14    | 0,079   | 0      | 19      | 0,238   | 0      |
| ship12s  | 12    | 0,056   | 0      | 17      | 0,190   | 0      |
| sierra   | 18    | 0,144   | 0      | 20      | 0,316   | 0      |
| stair    | 15    | 0,082   | 0      | 20      | 0,264   | 0      |
| standata | 12    | 0,050   | 0      | 16      | 0,159   | 0      |
| standgub | 12    | 0,051   | 0      | 16      | 0,160   | 0      |
| standmps | 22    | 0,095   | 0      | 30      | 0,349   | 0      |
| stocfor1 | 10    | 0,039   | 0      | 19      | 0,236   | 0      |
| stocfor2 | 19    | 0,136   | 0      | 25      | 0,368   | 0      |
| stocfor3 | 30    | 0,801   | 0      | 36      | 1,483   | 0      |
| truss    | 17    | 0,197   | 0      | 27      | 0,503   | 0      |
| tuff     | 20    | 0,090   | 0      | 23      | 0,299   | 0      |
| vtp.base | 9     | 0,042   | 0      | 17      | 0,157   | 0      |
| wood1p   | 22    | 0,304   | 0      | 23      | 0,448   | 0      |
| woodw    | 31    | 0,264   | 0      | 34      | 0,545   | 0      |

Continua na próxima página.

Tabela 2: Resultados comparativos de PCx-r e PCx-EOP(continuação).

| Problema          | PCx-r |         |        | PCx-EOP |         |        |
|-------------------|-------|---------|--------|---------|---------|--------|
|                   | Iter. | CPU (s) | Status | Iter.   | CPU (s) | Status |
| kennington-cre-a  | 25    | 0,274   | 0      | 23      | 0,494   | 0      |
| kennington-cre-b  | 43    | 2,420   | 0      | 40      | 3,035   | 0      |
| kennington-cre-c  | 25    | 0,245   | 0      | 25      | 0,453   | 0      |
| kennington-cre-d  | 41    | 1,998   | 0      | 38      | 2,529   | 0      |
| kennington-ken-07 | 13    | 0,107   | 0      | 20      | 0,314   | 0      |
| kennington-ken-11 | 20    | 0,489   | 0      | 23      | 1,034   | 0      |
| kennington-osa-07 | 21    | 0,380   | 0      | 54      | 1,811   | 0      |
| kennington-osa-14 | 24    | 0,881   | 0      | 77      | 4,754   | 0      |
| kennington-osa-30 | 24    | 1,680   | 0      | 81      | 9,036   | 0      |
| kennington-pds-02 | 24    | 0,301   | 0      | 28      | 0,648   | 0      |
| kennington-pds-06 | 32    | 5,592   | 0      | 39      | 7,198   | 0      |
| kennington-pds-10 | 40    | 31,566  | 0      | 46      | 35,754  | 0      |
| qap8              | 7     | 0,435   | 0      | 8       | 0,476   | 0      |
| <b>Somatórios</b> | 2115  | 115,17  | -      | 2836    | 156,29  | -      |

Note que o PCx-r não resolveu 3 problemas: `brandy`, `greenbea` e `scfxm2`, enquanto PCx-EOP não resolveu 5: `bnl1`, `fit1p`, `fit2p`, `greenbea`, `pilot4`. Apenas `greenbea` não foi resolvido pelas duas implementações.

Os resultados mostram que PCx-EOP resolveu o conjunto de 108 problemas em tempo de CPU de 2min 36s e o PCx-r resolveu o mesmo conjunto em 1min 55s, com número de iterações comparável. Isso significa que os tempos de CPU tem a mesma ordem de grandeza, para ambas as implementações. Esse era o parâmetro mais importante para nós já que, em nossa opinião, é o tempo de CPU que valida nossa abordagem como prova de conceito.

Uma das possíveis causas do resultado acima é que o MEOP é um método do tipo *greedy* ou guloso, no sentido que busca a minimização das médias dos resíduos, levando em conta apenas a próxima iteração. Entretanto, note que qualquer informação de iterações mais avançadas depende, para cada iteração, de uma fatoração de Cholesky e, portanto, tem auto custo

computacional.

Um outro fator importante a ser ressaltado é que nosso método escolhe o mesmo tamanho de passo em ambas as direções primal e dual, e tal passo é o implementado em PCx-MEOP. Por outro lado, o método preditor-corretor de Mehrotra implementado em PCx-r utiliza passos diferentes para as direções primal e dual. Aliás, Wright [77, p. 195] explica que

The idea of different primal and dual step lengths [...] has been used in both theoretical and practical development of primal dual methods since the earliest days. Computationally, this enhancement saves about 10%-20% on a typical problem.

Mesmo assim, em nosso método utilizamos os mesmo tamanho de passo, por conta da complexidade algébrica que isso envolveria. Considere que as restrições do subproblema que resolvemos no MEOP, dadas por  $g_C^i$  e  $g_L$  – veja (3.46), seriam agora dependentes de  $\alpha_P$  e  $\alpha_D$ , além de  $\mu$  e  $\sigma$ . Fixados  $\mu$  e  $\sigma$ , cada uma das  $\mathcal{O}(n)$  desigualdades é uma região do plano  $\mathbb{R}^2$  definido por uma hipérbole com eixos paralelos aos eixos  $\alpha_P$  e  $\alpha_D$ . Para que tais desigualdades fossem satisfeitas, precisaríamos computar  $\mathcal{O}(n)$  intersecções dessas regiões. O resultado seria uma região não convexa e podem ter até  $\mathcal{O}(n)$  pontos extremos.

Tanto do ponto de vista do cálculo algébrico quanto do método numérico, tais intersecções são causam grandes problemas. Deixamos tal escolha para um desenvolvimento futuro.

Neste capítulo apresentamos o conjunto de testes NETLIB-108 nem como os resultados numéricos do MEOP, comparando sua implementação, chamada PCx-EOP com uma implementação do método preditor-corretor de Mehrotra, denominada PCx-r. Tais resultado mostram que o MEOP é robusto e tem tempo de CPU comparável com PCx-r. Futuros desenvolvimentos, tanto da solução do subproblema quanto da implementação, podem ser feitos e se mostram promissores.

## Considerações finais

Neste trabalho procuramos responder a algumas das questões atuais em Métodos de Pontos Interiores primais duais, tais como a utilização de vários tipos de direções e o modo como se deve combiná-las. Neste sentido, propomos e implementamos um Método de Pontos Interiores preditor-corretor com pontos inactíves para PL, do tipo seguidor de caminho, fazendo uso de polinômios reais nas variáveis  $(\alpha, \mu, \sigma)$ , em que  $\alpha$  é o tamanho do passo,  $\mu$  é o parâmetro que define a trajetória central, e  $\sigma$  modela o peso que uma direção corretora deve ter.

O método proposto foi denominado Método de Escolha Otimizada de Parâmetros (MEOP) já que os parâmetros  $(\alpha, \mu, \sigma)$  foram tratados como variáveis e foram encontrados através da solução de um subproblema que *minimiza* uma função de mérito polinomial de grau total máximo 6 nessas três variáveis e que está sujeita a restrições que impõem que o próximo iterando esteja dentro de uma vizinhança da trajetória central.

Além disso, demonstramos resultados de convergência e complexidade do MEOP. Para tanto, estabelecemos a Condição 4.2, a qual leva em conta o tamanho dos dados do problema a ser resolvido, e a distância entre o ponto inicial e uma solução ótima. A importância dessa condição surgiu a partir da análise teórica do MEOP, embora ela também tenha implicações práticas. Ademais, essa condição vale também para outros métodos de pontos interiores inactíves, em particular aqueles que necessitam em suas análises de limitantes para  $\left\{ \left\| (x^k, z^k) \right\| \right\}$ .

O ponto inicial foi escolhido conforme a heurística de Mehrotra [49] e todo o conjunto de testes que utilizamos obedece a Condição 4.2. Essa escolha leva a um resultado de complexidade

um pouco pior em relação aos resultados encontrados por outros autores [75, 78, 82, 85], embora ainda polinomial.

De fato, a melhor complexidade de ordem de iterações para métodos de pontos interiores preditores-corretores do tipo Mehrotra com ponto inicial inactível é dado por Zhang e Zhang [81], com ordem de iterações  $\mathcal{O}(n^{1.5})$  e taxa de convergência Q-subquadrática. Outras complexidades, com ordem de iterações  $\mathcal{O}(n^3)$  foram dadas por Wright e Zhang [78] e Zhang [82]. Nossa análise demonstrou que temos uma ordem de iterações um pouco maior,  $\mathcal{O}(n^4)$ , mas ainda polinomial, e temos convergência Q-linear.

Alguns comentários são importantes. Tanto quanto sabemos, os algoritmos de Wright e Zhang [78], Zhang e Zhang [81] e Zhang [82] não foram implementados. Além disso, Gertz e Wright [23] utilizam em sua implementação um ponto inicial que também tem alguma relação com o tamanho dos dados do problema, porém, não há qualquer análise de complexidade ou convergência nesse trabalho. Outro detalhe importante é que grande parte das análises de convergência recentes [27] supõe que o ponto inicial é factível.

Embora os resultados teóricos de nosso método não superem os melhores resultados teóricos existentes, cabe ressaltar que, no nosso caso, a análise feita acima é do algoritmo que foi de fato implementado, incluindo-se aí o tratamento do pontos inactíveis, o ponto inicial de Mehrotra e os critérios de parada detalhados. Com isso, conseguimos um resultado teórico considerável de um algoritmo que tem seu funcionamento na prática demonstrado.

Foram feitos então experimentos computacionais com o MEOP. Para os testes, utilizamos o conjunto de testes NETLIB-108, escolhido entre os problemas do respostório NETLIB. Comparamos nosso método com o PCx [11], uma implementação do método preditor-corretor de Mehrotra. De fato, relatamos também as modificações que fizemos na escolha do ponto inicial e no critério de parada do PCx, de modo a melhorar seu desempenho. Essas modificações foram utilizadas tanto na nossa implementação, a qual chamamos PCx-EOP quanto a de referência, que denominamos PCx-r. Os testes mostraram que nosso algoritmo é competitivo e robusto,

quando comparado com uma implementação já bem testada e madura, como o PCx.

### **Perspectivas futuras**

O desenvolvimento do Método de Escolha Otimizada de Parâmetros trouxe algumas questões que não foram abordadas nessa tese. Algumas dessas questões podem servir como guia para futuras pesquisas.

Uma ideia interessante é concernente às matrizes  $H_P$  e  $H_D$  – veja Equação (3.3). Tais matrizes podem ser adaptadas, não só garantido a não-negatividade dos resíduos primais e duais, mas servindo, de certa maneira de fator de escalonamento do sistema o que geraria escalamento da função de mérito. Assim, poderíamos inclusive incluir um escalonamento da parte da complementaridade do sistema KKT e verificar como tal escalonamento impactaria no despenho do MEOP.

Quanto à análise do método, a fim de atacar o problema escolhemos um valor para  $\mu \neq 0$  e para  $\sigma = 0$ , de modo a garantir que o conjunto  $\Omega$  das restrições do subproblema resolvido em cada iteração fosse não-vazio. No entanto, quando da solução de problemas práticos, em grande parte das iterações o método utilizado encontrou ótimo para  $\mu = 0$  e  $\sigma \neq 0$ . Embora a demonstração continue válida, a utilização de outros valores para  $\mu$  e  $\sigma$  pode fazer com que a complexidade seja melhorada. Além disso, outro caminho na análise seria tentar verificar se é possível demonstrar convergências de ordem maior, como quadrática ou subquadrática.

Uma investigação que pretendemos fazer é quanto à Condição 4.2. Essa condição permite que estudemos relações entre o ponto inicial, a solução ótima e o tamanho dos dados. Embora tenhamos escolhido o ponto inicial de Mehrotra, qualquer outro ponto inicial que satisfaça essa Condição tem garantia de convergência do MEOP. Ademais, conforme dissemos, poderemos utilizar essa condição em outras análises de convergência. Daí testes podem surgir com outros pontos iniciais no PCx-EOP ou em outras implementações de pontos interiores.

Em resumo, podemos dizer que através das ideias desenvolvidas nesta tese alcançamos os objetivos a que nos propusemos e, mesmo assim, tais ideias nos deram outros caminhos a serem

percorridos. Isso mostra que tanto a teoria quanto a implementação de Métodos de Pontos Interiores ainda representam um vasto campo de pesquisa a ser explorado. De fato, poucos trabalhos têm desenvolvido esses dois vieses de forma conjunta, o que foi feito em nossa pesquisa. Tal fato, por si só, é suficiente para demonstrar a contribuição desta tese para o estado da arte na área de pontos interiores.

# Referências

- [1] Ilan Adler, Mauricio G C Resende, Geraldo Veiga e Narendra Karmarkar. “An implementation of Karmarkar’s algorithm for linear programming”. Em: *Mathematical Programming* 44.1-3 (1989), pp. 297–335.
- [2] Erling D Andersen, Jacek Gondzio, Csaba Mészáros e Xiaojie Xu. “Implementation of interior point methods for large scale linear programming”. Em: *Interior Point Methods of Mathematical Programming*. Ed. por Tamás Terlaky. Springer, 1996, pp. 189–252.
- [3] Robert G Bartle e Donald R Sherbert. *Introduction to Real Analysis*. 4<sup>a</sup> ed. John Wiley & Sons, jan. de 2011.
- [4] Dave A Bayer e Jeffrey C Lagarias. “The Nonlinear Geometry of Linear Programming. I Affine and Projective Scaling Trajectories”. Em: *Transactions of the American Mathematical Society* 314.2 (ago. de 1989), pp. 499–526.
- [5] Dave A Bayer e Jeffrey C Lagarias. “The Nonlinear Geometry of Linear Programming. II Legendre Transform Coordinates and Central Trajectories”. Em: *Transactions of the American Mathematical Society* 314.2 (jan. de 1989), pp. 527–581.
- [6] Mokhtar S Bazaraa, John J Jarvis e Hanif D Sherali. *Linear Programming and Network Flows*. 4<sup>a</sup> ed. New York: Wiley, dez. de 2009.
- [7] Robert G Bland, Donald Goldfarb e Michael J Todd. “The ellipsoid method: A survey”. Em: *Operations Research* 29.6 (1981), pp. 1039–1091.
- [8] Marco Colombo. “Advances in Interior Point Methods for Large-Scale Linear Programming”. Tese de doutorado. Edinburgh: School of Mathematics/University of Edinburgh, 2008.
- [9] Marco Colombo e Jacek Gondzio. “Further development of multiple centrality correctors for interior point methods”. Em: *Computational Optimization and Applications* 41.3 (2008), pp. 277–305.
- [10] Andrew R Curtis e Jonh K Reid. “On the Automatic Scaling of Matrices for Gaussian Elimination”. Em: *IMA Journal of Applied Mathematics* 10.1 (1972), pp. 118–124.
- [11] Joseph Czyzyk, Sanjay Mehrotra, Michael Wagner e Stephen J Wright. “PCx: an interior-point code for linear programming”. Em: *Optimization Methods and Software* 11.1 (1999), pp. 397–430.
- [12] Joseph Czyzyk, Stephen J Wright, Michael Wagner e Sanjay Mehrotra. *PCx User Guide: Supplement for PCxGUI*. Rel. téc. Mathematics e Computer Science Division, Aragonne National Laboratory, US Departament of Energy, nov. de 1997.

- [13] George B Dantzig. “Maximization of a linear function of variables subject to linear inequalities”. Em: *Activity Analysis of Production and Allocation: Proceedings of a Conference*. Ed. por T C Koopmans. New York: John Wiley & Sons, 1951, pp. 339–347.
- [14] George B Dantzig. *Linear Programming and Extensions*. 1ª ed. Princeton, New Jersey: Princeton University Press, 1963.
- [15] George B Dantzig. “Linear programming”. Em: *Operations Research* 50.1 (2002), pp. 42–47.
- [16] Marco D’Apuzzo, Valentina Simone e Daniela Serafino. “Starting-point strategies for an infeasible potential reduction method”. Em: *Optimization Letters* 4.1 (out. de 2009), pp. 131–146.
- [17] Antoine Deza, Eissa Nematollahi, Reza Peyghami e Tamás Terlaky. “The central path visits all the vertices of the Klee-Minty cube”. Em: *Optimization Methods and Software* 21.5 (2006), pp. 851–865.
- [18] Ilya I Dikin. “Iterative solution of problems of linear and quadratic programming”. Em: *Soviet Mathematics Doklady* 8.3 (1967), pp. 674–675.
- [19] Jack J Dongarra e Eric Grosse. “Distribution of mathematical software via electronic mail”. Em: *Communications of the ACM* 30.5 (maio de 1987), pp. 403–407.
- [20] Shu-Cherng Fang e Sarat Puthenpura. *Linear optimization and extensions. theory and algorithms*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.
- [21] David M Gay. “Electronic mail distribution of linear programming test problems”. Em: *Mathematical Programming Society COAL Newsletter* 13 (dez. de 1985), pp. 10–12.
- [22] E Michael Gertz, Jorge Nocedal e Annick Sartenaer. “A starting point strategy for nonlinear interior methods”. Em: *Applied Mathematics Letters* 17.8 (2004), pp. 945–952.
- [23] E Michael Gertz e Stephen J Wright. “Object-oriented software for quadratic programming”. Em: *ACM Transactions on Mathematical Software* 29.1 (mar. de 2003), pp. 58–81.
- [24] Jacek Gondzio. “HOPDM (version 2.12) — A fast LP solver based on a primal-dual interior point method”. Em: *European Journal of Operational Research* 85.1 (ago. de 1995), pp. 221–225.
- [25] Jacek Gondzio. “Multiple centrality corrections in a primal-dual method for linear programming”. Em: *Computational Optimization and Applications* 6.2 (1996), pp. 137–156.
- [26] Jacek Gondzio. “Matrix-Free Interior Point Method”. Em: *Computational Optimization and Applications* (2010).
- [27] Jacek Gondzio. “Interior point methods 25 years later”. Em: *European Journal of Operational Research* 218.3 (maio de 2012), pp. 587–601.
- [28] Jacek Gondzio e Tamás Terlaky. “A computational view of interior point methods”. Em: *Advances in linear and integer programming*. Ed. por John E Beasley. Oxford, UK: Oxford University Press, Inc., 1996, pp. 103–144.

- [29] Clóvis C Gonzaga. “Path-Following Methods for Linear Programming”. Em: *Siam Review* 34.2 (1992), pp. 167–224.
- [30] Osman Güler. “Generalized Linear Complementarity Problems”. Em: *Mathematics of Operations Research* 20.2 (maio de 1995), pp. 441–448.
- [31] Osman Güler, Cornelius Roos, Tamás Terlaky e Jean-Philippe Vial. “A survey of the implications of the behavior of the central path for the duality theory of linear programming”. Em: *Management science* 41.12 (maio de 1995), pp. 1992–1934.
- [32] Julian A J Hall e Ken I M McKinnon. “Hyper-sparsity in the revised simplex method and how to exploit it”. Em: *Computational Optimization and Applications* 32.3 (2005), pp. 259–283.
- [33] Didier Henrion, Jean-Bernard Lasserre e Johan Löfberg. “GloptiPoly 3: moments, optimization and semidefinite programming”. Em: *Optimization Methods and Software* 24.4-5 (2009), pp. 761–779.
- [34] Roger A Horn e Charles R Johnson. *Matrix Analysis*. 1ª ed. Cambridge; New York: Cambridge University Press, 1985.
- [35] Pi-Fang Hung e Yinyu Ye. “An Asymptotical  $O(\sqrt{n}L)$ -Iteration Path-Following Linear Programming Algorithm That Uses Wide Neighborhoods”. Em: *SIAM Journal on Optimization* 6.3 (ago. de 1996), pp. 570–586.
- [36] Benjamin Jansen. *Interior Point Techniques in Optimization: Complementarity, Sensitivity and Algorithms*. Ed. por Panos M Pardalos e Donald Hearn. 1ª ed. Vol. 6. Applied Optimziation. Dordrecht: Kluwer Academic Publishers, 1997.
- [37] Benjamin Jansen, Cornelius Roos e Tamás Terlaky. “A family of polynomial affine scaling algorithms for positive semidefinite linear complementarity problems”. Em: *SIAM Journal on Optimization* 7.1 (1997), pp. 126–140.
- [38] Florian Jarre e Martin Wechs. “Extending Mehrotra’s corrector for linear programs”. Em: *Advanced Modeling and Optimization* 1.2 (1999), pp. 38–60.
- [39] Narendra Karmarkar. “A new polynomial-time algorithm for linear programming”. Em: *Combinatorica* 4.4 (1984), pp. 373–395.
- [40] Jeffery L Kennington, Sandra Niemi e Stephen J Wichmann. “An empirical evaluation of the KORBX algorithms for military airlift applications”. Em: *Operations Research* 38.2 (abr. de 1990).
- [41] Lev G Khachiyan. “A polynomial algorithm in linear programming”. Em: *Soviet Mathematics Doklady* 20 (1979), pp. 191–194.
- [42] Tomonari Kitahara e Takashi Tsuchiya. “A Simple Variant of the Mizuno–Todd–Ye Predictor-Corrector Algorithm and Its Objective-Function-Free Complexity”. Em: *SIAM Journal on Optimization* 23.3 (jul. de 2013), pp. 1890–1903.
- [43] Victor Klee e George J Minty. “How good is the simplex algorithm?” Em: *Inequalities III: Proceedings of the Third Symposium on Inequalities*. Ed. por Oved Shisha. New York: Academic Press, 1972, pp. 159–175.

- [44] Masakazu Kojima, Nimrod Megiddo e Shinji Mizuno. “A primal-dual infeasible-interior-point algorithm for linear programming”. Em: *Mathematical Programming* 61.3 (1993), pp. 263–280.
- [45] Masakazu Kojima, Shinji Mizuno e Akiko Yoshise. “A polynomial-time algorithm for a class of linear complementarity problems”. Em: *Mathematical Programming* 44 (1989), pp. 1–26.
- [46] Jean-Bernard Lasserre. “Global Optimization with Polynomials and the Problem of Moments”. Em: *SIAM Journal on Optimization* 11.3 (2001), p. 796.
- [47] Monique Laurent. “Sums of squares, moment matrices and optimization over polynomials”. Em: *Emerging Applications of Algebraic Geometry*. Ed. por Mihai Putinar, Seth Sullivant, Douglas N Arnold e Fadil Santosa. New York, NY: Springer, 2010, pp. 157–270.
- [48] Nimrod Megiddo. “Pathways to the optimal set in linear programming”. Em: *Progress in Mathematical Programming Interior-point and related methods*. Ed. por Nimrod Megiddo. New York: Springer-Verlag, 1989, pp. 131–158.
- [49] Sanjay Mehrotra. “On the Implementation of a Primal-Dual Interior Point Method”. Em: *SIAM Journal on Optimization* 2.4 (1992), pp. 575–601.
- [50] Sanjay Mehrotra e Zhifeng Li. “Convergence Conditions and Krylov Subspace-Based Corrections for Primal-Dual Interior-Point Method”. Em: *SIAM Journal on Optimization* 15.3 (2005), pp. 635–653.
- [51] Renato D C Monteiro e Ilan Adler. “Interior path following primal-dual algorithms. Part I: Linear programming”. Em: *Mathematical Programming* 44 (1989), pp. 27–41.
- [52] Renato D C Monteiro, Ilan Adler e Mauricio G C Resende. “A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension”. Em: *Mathematics of Operations Research* 15.2 (1990), pp. 191–214.
- [53] Pedro Augusto Munari Junior. “Theoretical and computational issues for improving the performance of linear optimization methods”. Tese de doutorado. São Carlos: ICMC/USP, jan. de 2013.
- [54] John Lawrence Nazareth. “Homotopy techniques in linear programming”. Em: *Algorithmica* 1.1-4 (1986), pp. 529–535.
- [55] George L Nemhauser e Laurence A Wolsey. *Integer and combinatorial optimization*. Integer and combinatorial optimization. New York: John Wiley & Sons, 1999.
- [56] Yurii Nesterov. “Squared Functional Systems and Optimization Problems”. Em: *High performance optimization*. Ed. por Hans Frenk, Kees Roos, Tamás Terlaky e Shuzhong Zhang. Boston, MA: Springer US, 2000, pp. 405–440.
- [57] Yurii Nesterov. *Introductory Lectures on Convex Optimization: a basic course*. 1<sup>a</sup> ed. Applied Optimization. Boston: Kluwer Academic Publishers, dez. de 2003.
- [58] Esmond G Ng e Barry W Peyton. “Block sparse Cholesky algorithms on advanced uni-processor computers”. Em: *SIAM Journal on Scientific Computing* 14.5 (1993), pp. 1034–1056.

- [59] James M Ortega e Werner C Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. 1<sup>a</sup> ed. Classics in applied mathematics. Philadelphia, PA: SIAM, jan. de 2000.
- [60] Florian A Potra. “A quadratically convergent predictor-corrector method for solving linear programs from infeasible starting points”. Em: *Mathematical Programming* 67.1-3 (out. de 1994), pp. 383–406.
- [61] James Renegar. “A polynomial-time algorithm, based on Newton’s method, for linear programming”. Em: *Mathematical Programming* 40.1, (Ser. A) (1988), pp. 59–93.
- [62] Mauricio G C Resende e Geraldo Veiga. “An Implementation of the Dual Affine Scaling Algorithm for Minimum-Cost Flow on Bipartite Uncapacitated Networks”. Em: *SIAM Journal on Optimization* 3.3 (ago. de 1993), pp. 516–537.
- [63] Maziar Salahi, Jiming Peng e Tamás Terlaky. “On Mehrotra-type predictor-corrector algorithms”. Em: *SIAM Journal on Optimization* 18.4 (2007), pp. 1377–1397.
- [64] Alexander Schrijver. *Theory of Linear and Integer Programming*. New York: John Wiley & Sons, 1986.
- [65] Ron Shamir. “The Efficiency of the Simplex Method: A Survey”. Em: *Management science* 33.3 (mar. de 1987), pp. 301–334.
- [66] Gy Sonnevend. “An “analytical centre” for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming”. Em: *System Modelling and Optimization: proceedings of the 12th IFIP Conference, Budapest, Hungary, September 2-6, 1985*. Ed. por András Prékopa, J Szelezsáan e Beátra Strazicky. Berlin/Heidelberg: Springer-Verlag, 1986, pp. 866–875.
- [67] Robert J Vanderbei. *Linear programming: foundations and extensions*. 3<sup>a</sup> ed. International Series in Operations Research and Management Science. New York: Springer-Verlag, 2007.
- [68] Robert J Vanderbei e David F Shanno. “An Interior-Point Algorithm for Nonconvex Nonlinear Programming”. Em: *Computational Optimization and Applications* 13.1-3 (abr. de 1999).
- [69] Stephen A Vavasis e Yinyu Ye. “A primal-dual interior point method whose running time depends only on the constraint matrix”. Em: *Mathematical Programming* 74.1 (jul. de 1996), pp. 79–120.
- [70] Fernando Rocha Villas-Bôas. “Escolha adiada do parâmetro de penalização e do tamanho de passo em algoritmos de Pontos Interiores ”. Tese de doutorado. Campinas: IMECC/Unicamp, 2000.
- [71] Fernando Rocha Villas-Bôas, Aurelio Ribeiro Leite Oliveira, Clóvis Perin e Luiz-Rafael Santos. “Polynomial Inequality Systems in Neighborhoods of the Central Path”. Em: *Proceedings of the 3rd IMA Conference on Numerical Linear Algebra and Optimisation*. IMA, University of Birmingham. Birmingham, UK, set. de 2012, p. 26.

- [72] Fernando Rocha Villas-Bôas, Aurelio Ribeiro Leite Oliveira, Clóvis Perin e Luiz-Rafael Santos. *Predictive Polynomials in Interior Point Methods*. Rel. téc. IMECC/Unicamp, 2013.
- [73] Fernando Rocha Villas-Bôas e Clóvis Perin. “Postponing the choice of penalty parameter and step length”. Em: *Computational Optimization and Applications* 24.1 (2003), pp. 63–81.
- [74] Hayato Waki, Sunyoung Kim, Masakazu Kojima, Masakazu Muramatsu e Hiroshi Sugimoto. “Algorithm 883: SparsePOP—A Sparse Semidefinite Programming Relaxation of Polynomial Optimization Problems”. Em: *ACM Transactions on Mathematical Software* 35.2 (jul. de 2008), Art. 15, 13–13.
- [75] Stephen J Wright. “A path-following infeasible-interior-point algorithm for linear complementarity problems”. Em: *Optimization Methods and Software* 2.2 (jan. de 1993), pp. 79–106.
- [76] Stephen J Wright. “An infeasible-interior-point algorithm for linear complementarity problems”. Em: *Mathematical Programming* 67.1-3 (out. de 1994), pp. 29–51.
- [77] Stephen J Wright. *Primal-dual interior point methods*. 1<sup>a</sup> ed. Philadelphia, PA: SIAM, 1997.
- [78] Stephen J Wright e Yin Zhang. “A superquadratic infeasible-interior-point method for linear complementarity problems”. Em: *Mathematical Programming* 73.3 (jan. de 1996), pp. 269–289.
- [79] Yinyu Ye, Michael J Todd e Shinji Mizuno. “An  $\mathcal{O}(\sqrt{nl})$ -iteration homogeneous and self-dual linear programming algorithm”. Em: *Mathematics of Operations Research* 19.1 (fev. de 1994), pp. 53–67.
- [80] E Alper Yildirim e Michael J Todd. “Sensitivity analysis in linear programming and semidefinite programming using interior-point methods”. Em: *Mathematical Programming* 90.2, Ser. A (2001), pp. 229–261.
- [81] Detong Zhang e Yin Zhang. “A Mehrotra-type predictor-corrector algorithm with polynomiality and Q-subquadratic convergence”. Em: *Annals of Operations Research* 62.1 (dez. de 1996), pp. 131–150.
- [82] Yin Zhang. “On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem”. Em: *SIAM Journal on Optimization* 4.1 (1994), pp. 208–227.
- [83] Yin Zhang e Richard A Tapia. “A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming”. Em: *SIAM Journal on Optimization* 3.1 (1993), pp. 118–133.
- [84] Yin Zhang e Detong Zhang. “Superlinear convergence of infeasible-interior-point methods for linear programming”. Em: *Mathematical Programming* 66.1-3 (ago. de 1994), pp. 361–377.

- [85] Yin Zhang e Detong Zhang. “On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms”. Em: *Mathematical Programming* 68.3, Ser. A (1995), pp. 303–318.