



Universidade Estadual de Campinas  
Instituto de Matemática Estatística e Computação Científica  
Departamento de Matemática Aplicada



## **2o Exame de Qualificação:**

### **Escolha adiada de parâmetros em métodos de pontos interiores para programação linear**

**Luiz Rafael dos Santos**

Doutorado em Matemática Aplicada

Orientador: Prof. Dr. Aurelio Ribeiro Leite de Oliveira

Coorientadores: Dr. Fernando Villas-Bôas e Prof. Dr. Clóvis Perin

Este trabalho contou com suporte financeiro da FAPESP (processo 2008/09685-3) e do CNPq.

Campinas-SP

# Sumário

<b>1. Introdução e Fundamentos</b>	<b>1</b>	
1.1. Programação linear . . . . .	1	
1.1.1. Método simplex . . . . .	3	
1.1.2. Método elipsoide . . . . .	3	
1.1.3. Métodos de pontos interiores . . . . .	4	
1.2. Objetivos e estrutura desde trabalho . . . . .	5	
<b>2. Métodos de Pontos Interiores Primais-Duais para Programação Linear</b>	<b>6</b>	
2.1. Um método de pontos interiores . . . . .	6	
2.1.1. Método Afim-escala . . . . .	9	
2.1.2. O problema de barreira . . . . .	11	
2.2. Métodos seguidores de caminho . . . . .	13	
2.2.1. Modelo geral para métodos seguidores de caminho . . . . .	13	
2.2.2. Vizinhanças da Trajetória Central . . . . .	14	
Vizinhança Simétrica . . . . .	16	
2.2.3. Ponto inicial infactível . . . . .	17	
Métodos Infactíveis . . . . .	18	
2.3. Métodos de Pontos Interiores na Prática . . . . .	19	
2.3.1. Método Preditor-Corretor de Mehrotra . . . . .	19	
Direção com correção de Segunda Ordem . . . . .	21	
2.3.2. Método das múltiplas correções de centralização . . . . .	23	
2.3.3. Ponto Inicial . . . . .	25	
2.3.4. Critério de Parada . . . . .	27	
<b>3. Uma função de mérito polinomial em Métodos de Pontos Interiores (MPI)</b>	<b>29</b>	{acro:MPI}
3.1. Um KKT escalado . . . . .	29	
3.2. Direções de busca . . . . .	31	
3.2.1. Direção Afim-escala . . . . .	31	
3.2.2. A direção ideal . . . . .	32	
3.2.3. Combinando direções . . . . .	34	

---

3.3. O próximo resíduo . . . . .	36
3.4. Uma função de mérito polinomial . . . . .	37
3.5. Vizinhança simétrica como conjunto de funções polinomiais . . . . .	41
3.6. Subproblema de Otimização de Polinômios . . . . .	43
<b>4. Resultados de Convergência</b>	<b>46</b>
<b>5. Experimentos Numéricos</b>	<b>47</b>
<b>Considerações Finais</b>	<b>48</b>
<b>Referências Bibliográficas</b>	<b>49</b>
<b>A. Resultados Técnicos</b>	<b>53</b>

# Capítulo 1.

## Introdução e Fundamentos

Neste capítulo será feita uma breve introdução aos problemas de Programação Linear (PL), bem como aspectos preliminares de alguns métodos criados para resolvê-lo. Posteriormente são definidos os objetivos deste trabalho e modo como o mesmo foi estruturado. {acro:PL}

### 1.1. Programação linear

Programação Linear é um tópico que tem sido amplamente estudado no escopo da otimização e que ganhou relevância a partir da 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 mesmo Dantzig, em um texto comemorativo de 1991, faz um resumo histórico do desenvolvimento da PL e a define de maneira genérica:

“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.”[12].

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 puderam 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 [11, 39]

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

---

<sup>1</sup>Usaremos problema de PL e PL como sinônimos.

exemplo em problemas de economia, logística, planejamento e controle da produção entre outros. 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*:

$$\begin{array}{ll} \underset{x}{\text{minimizar}} & c^T x \\ \text{sujeito a} & \begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{array} \quad (1.1) \quad \{\text{eq:PL}\}$$

em que  $A \in \mathbb{R}^{m \times n}$  e  $c$ ,  $x$  e  $b$  tem dimensão compatível. A região  $\mathcal{P} = \{x \in \mathbb{R}^n | Ax = b \text{ e } x \geq 0\}$  é conhecida como região factível do problema.

Como uma função linear é convexa, problemas de PL tem a particularidade dada pelo seguinte teorema [4, cap. 3]:

{teo:fundament

**Teorema 1.1** (Teorema Fundamental da Programação Linear). *Em um problema de Programação Linear cuja região factível  $\mathcal{P}$  possua pelo menos um ponto extremo, ou o valor ótimo da função objetivo é ilimitado ou então este valor será atingido em um ponto extremo de  $\mathcal{F}$ .*

Um conjunto de restrições lineares define um *poliedro* que constitui então a *região factível*. De acordo com o Teorema 1.1, uma maneira 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

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

Embora este valor seja superestimado, já que nem todos os vértices são factíveis, ainda assim nosso método 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 este número é exponencial já que

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

Assim, encontrar o *vértice ótimo* de maneira inteligente e eficiente é o segredo dos bons métodos para resolver problemas de PL.

### 1.1.1. Método simplex

O Método Simplex foi apresentado por Dantzig [10] 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 a busca da solução do PL caminhando pelos vértices de maneira a utilizar o valor da função objetivo – caso não degenerado – 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 ocorre quando um vértice em  $\mathbb{R}^n$  é definido por  $p > n$  restrições, e um passo de tamanho zero pode ser produzido pelo método. Desta maneira, o método simplex não sairia do lugar, e logo nenhuma melhora na função objetivo seria alcançada.

O método simplex se mostrou, e ainda se mostra, robusto e eficiente em grande quantidade de problemas, e por isso mesmo foi considerado *o método* para se resolver PL's. Entretanto mostrou-se que o método simplex possui complexidade exponencial. Isto significa que há a possibilidade de todos os vértices do poliedro serem visitados para que se encontre a solução ótima. De fato um exemplo patológico criado por Klee and Minty [30], precisou de  $2^m$  iterações do método simplex para ser resolvido. Apesar disso nenhum caso com número exponencial de iterações foi encontrado na vida real, e geralmente apenas uma pequena parte de vértices é consultada até que se chegue a uma solução ótima. Um resumo com a eficiência do método simplex pode ser encontrada em [40].

### 1.1.2. Método 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 [29] através de seu *Método Elipsoide*. Este método, 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 um PL faz parte dos problemas polinomiais, isto é, que pode ser resolvido por um algoritmo de complexidade polinomial. Provou-se que a complexidade deste método é  $\mathcal{O}(n^2L)$ . Todavia, do ponto de vista prático, o método elipsoide não conseguiu competir com o método simplex [5], 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. Consequentemente, embora desafortunadamente o método simplex tenha complexidade exponencial na análise do *pior caso*, provou-se na prática, i.e., no caso médio,

que era absolutamente superior ao método elipsoide.

### 1.1.3. 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 [28] inicia uma nova abordagem que ficou conhecida como MPI.

De fato, uma busca histórica mostra que em 1955 o primeiro *Métodos de Pontos Interiores* é atribuído a Frisch [16]. Este trabalho implementa uma função barreira-logarítmica para encontrar o mínimo local de uma função não linear, sujeita a restrições de desigualdade. O método foi extensamente estudado por Fiacco and McCormick [15] porém caiu em desuso por conta de sua ineficiência bem como a presença de competidores muito fortes como a programação quadrática sequencial.

Em 1967 Dikin [14] publicou um trabalho independentemente de Karmarkar, que resolvia PL usando um método de pontos interiores, que 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 desta região. Isto é feito por criar uma família parametrizada de soluções que convergem assintoticamente para a solução exata. Consequentemente, usa-se uma abordagem não-linear para resolver um problema linear, escapando então da dificuldade que a dimensão do problema apresenta ao se lidar com as características combinatoriais do PL.

Além disso, já vimos que no método simplex a quantidade de iterações cresce e de acordo com o tamanho do problema, o que não se repete nos MPI. Do ponto de vista da complexidade computacional, este método também é polinomial, da ordem  $\mathcal{O}(n^3)$  [28].


Ao contrário do método elipsoide, o método de Karmarkar tem uma performance muito melhor e compete com o simplex, sendo consideravelmente melhor que este em problemas de larga-escala. Uma variante do método de Karmarkar foi implementada por Adler, Resende, Veiga, and Karmarkar [1] e desde então a compreensão do funcionamento deste métodos aumentou consideravelmente, ao mesmo tempo em que métodos variantes tem sido propostos, muitos deles mostrando-se alternativas computacionais viáveis ao método simplex.

Na prática há classes de problemas que são melhores resolvidos pelo método simplex e outras para os quais os métodos de pontos interiores são mais adequados. Tamanho, 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. Este fato entretanto não vale para casos hiper-esparcos, em que o método simplex é virtualmente imbatível [25], e para problemas em rede, nos quais uma especialização

do método simplex permite explorar a estrutura do problema de maneira muito eficiente [38].

Embora esta tese trate de MPI para problemas lineares, aqueles podem ser utilizados para resolver problemas não-lineares. Em particular problemas de Programação Quadrática (QP) {acro:QP} convexa são largamente resolvidos com códigos de Pontos Interiores, em particular quando são de larga escala. Com efeito, um QP convexo pode ser escrito como

$$\begin{array}{ll} \underset{x}{\text{minimizar}} & x^T Q x + c^T x \\ \text{sujeito a} & \begin{cases} Ax = b \\ x \geq 0 \end{cases}, \end{array} \quad (1.2) \quad \text{{eq:QP}}$$

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

## 1.2. Objetivos e estrutura desde trabalho

A ser escrito.



## Capítulo 2.

# Métodos de Pontos Interiores Primais-Duais para Programação Linear

{chap:mpis}

“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” [28]

(N. 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 [42, 46]. Recentemente Gondzio [21] fez um resumo histórico no qual resgata as principais contribuições no tema. Estas referências servem como texto introdutório e foi com base nelas, juntamente com [6, 44], que este capítulo foi escrito.

### 2.1. Um método de pontos interiores

O problema de PL na forma padrão,

$$(P) \quad \begin{array}{ll} \underset{x}{\text{minimizar}} & c^T x \\ \text{sujeito a} & \begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{array} \quad (2.1) \quad \text{\{eq:primal\}}$$

é chamado *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

$$(D) \quad \begin{array}{ll} \underset{(y,z)}{\text{maximizar}} & b^T y \\ \text{sujeito a} & \begin{cases} A^T y + z = b \\ z \geq 0, y \text{ livre} \end{cases} \end{array} \quad (2.2) \quad \{\text{eq:dual}\}$$

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 os componentes de  $y$  de *variáveis duais* enquanto que  $z$  é o vetor das *folgas duais*.

Definimos os conjuntos de pontos factíveis primais e duais respectivamente como

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

e usando esta notação o par primal-dual de (2.1-2.2) pode ser reescrito como

$$\min_x c^T x \quad \text{s.a.} \quad x \in \mathcal{P}; \quad \max_{(y,z)} b^T y \quad \text{s.a.} \quad (y, z) \in \mathcal{D}; \quad (2.4) \quad \{\text{eq:primal-dual}\}$$

Seja  $\mathcal{F} = \mathcal{P} \times \mathcal{D}$  o conjuntos dos pontos primais-duais factíveis. Como  $A$  tem posto completo, note que existe uma relação unívoca entre  $y$  e  $z$ . Para fins de notação, podemos escrever  $w = (x, y, z)$  e portanto poderemos dizer que  $w \in \mathcal{F}$ .

Neste trabalho, vamos assumir os seguintes pressupostos:

$\mathcal{P}1$ .  $\mathcal{F}$  é não-vazio, isto é, existe ponto factível.

$\mathcal{P}2$ . Para quaisquer  $(x, y, z) \in \mathcal{F}$  e  $(\bar{x}, \bar{y}, \bar{z}) \in \mathcal{F}$ , temos  $(x - \bar{x})^T (x - \bar{z}) \geq 0$ .

Sabe-se que  $\mathcal{P}1$  implica que sempre é possível a existência de solução para o PL 2.1 (veja [23, Teo. 3.1]).

Além disso, seja

$$\mathcal{F}^+ = \{w \in \mathcal{F} : (x, z) > 0\}$$

o conjunto dos pontos primais-duais interiores. Em pontos interiores, dedica-se especialmente a estudar iterandos que estejam contidos em  $\mathcal{F}^+$ .

Apresentamos abaixo alguns resultados bem conhecidos sobre a relação entre o par primal-dual. Para maiores detalhes consulte-se [6, 24, 46], textos que inspiraram esta apresentação.

$\{\text{lema:weak-dual}\}$

**Lema 2.1** (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

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

■

O significado da Dualidade Fraca é que o valores objetivo primal serve de limite 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  $(x, y, z)$ , 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 2.2** (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  $(P)$  tem uma solução factível é que  $\mathcal{P} \neq \emptyset$ . Além disso, se  $\mathcal{P} \neq \emptyset$  e  $\mathcal{D} \neq \emptyset$  então ambos  $(P)$  e  $(D)$  admitem solução ótima  $(x^*, y^*, z^*)$  e pelo Lema 2.2 o valor da função objetivo coincide neste ponto. Por outro lado, se os conjuntos  $\mathcal{P}$  ou  $\mathcal{D}$  forem vazios, 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  $(P)$  e  $(D)$ . Tais condições no ajudarão 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 Karush-Kuhn-Tucker (KKT) expressam uma condição de otimalidade de primeira ordem para um PL e podem ser escritas como



$$Ax = b,$$

(2.5a) {eq:KKT-fac-pr}

$$A^T y + z = c,$$

(2.5b) {eq:KKT-fac-du}

$$XZe = 0,$$

(2.5c) {eq:KKT-comple}

$$(x, z) \geq 0.$$

(2.5d) {eq:KKT-nao-ne}

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

As duas primeiras equações de (2.5) são conhecidas como factibilidade primal e dual, respectivamente<sup>1</sup>. A equação (2.5c) significa que  $x_i z_i = 0$  para todo  $i = 1, 2, \dots, n$  e é chamada complementaridade. A última equação é chamada não-negatividade. Note ainda que se  $(x, y, z) \in \mathcal{F}$  então ele satisfaz (2.5a), (2.5b) e (2.5d). 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:

$$x^T z = c^T x - b^T y \tag{2.6} \text{ {eq:gap-comple}}$$

<sup>1</sup>Para fins de simplificação de notação, para qualquer  $u, v \in \mathbb{R}^n$ , o vetor  $UVe$  poderá ser representado por  $uv$ .

A valor  $x^T z$  é chamado *gap de complementaridade*. Quando este valor converge para zero, então estamos em uma solução ótima. Além disso, a igualdade entre o gap de complementaridade e o gap de dualidade mostrado em (2.6) só vale quando o ponto é factível.

As equações (2.5) dão uma condição necessária e suficiente para que  $w^* = (x^*, y^*, z^*)$  seja solução de (2.1-2.2), isto é,  $x^*$  é solução de (P) então  $(y^*, z^*)$  é solução de (D). Consequentemente, um corolário de KKT para PL é o Lema 2.2.

Uma hipótese padrão que consideraremos, chamada *hipótese de ponto interior*, diz que tanto  $\mathcal{P}^0$  quanto  $\mathcal{D}^0$  sejam não vazios [24]. Caso essa hipótese não seja satisfeita, considera-se permitir que o método aceite iterandos infactíveis ou introduzindo perturbações que aumentem o conjunto  $\mathcal{F}$ .

### 2.1.1. Método Afim-escala

{sec:affine-sc

Os métodos de pontos interiores primais-duais para PL basicamente encontram  $w^*$  resolvendo as equações (2.5). Note que a única equação não-linear de (2.5) é  $XZe = 0$  e por conta dela é aplica-se alguma variante do método de Newton, modificando a direção de busca e o tamanho de passo de modo que (2.5d) seja satisfeita *estritamente* em toda iteração.

Podemos reescrever (2.5) utilizando-nos de uma aplicação  $F : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^{2n+m}$  da seguinte maneira:

{eq:KKT-Newton

$$F(w) = \begin{bmatrix} Ax - b \\ A^T y + z - c \\ XZe \end{bmatrix} = 0, \quad (2.7a) \quad \text{{eq:KKT-Newton}}$$

$$(x, s) \geq 0. \quad (2.7b) \quad \text{{eq:KKT-Newton}}$$

Todos os métodos primais-duais geram iterandos  $w^k = (x^k, y^k, z^k)$  que satisfazem (2.7b) estritamente, isto é,  $(x^k, z^k) > 0$ . Esta propriedade dá origem ao termo *ponto interior*. Ao respeitar estes pontos, impede-se iterandos tais que  $F(w^k) = 0$ , mas que não satisfaçam  $(x, s) \geq 0$ . Este tipos de soluções são facilmente encontradas, no entanto não nos dão informações úteis para a solução de (P) e (D). A maioria dos métodos primais-duais exige que  $w^k \in \mathcal{F}^0$ , ou seja, que o ponto seja estritamente factível.

O método de Newton aplicado a (2.7a) 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 (1.2), a diferença para estaria justamente na matriz jacobiana, que seria que dada por

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

Note que o passo Newton é encontrado por resolver

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - z \\ -XZe \end{bmatrix} = \begin{bmatrix} r_P \\ r_D \\ r_C \end{bmatrix}. \quad (2.8) \quad \text{\texttt{eq:KKT-Newton}}$$

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 (2.7b). Para evitar essa dificuldade pode ser feita uma busca linear na direção de Newton tal que o novo iterando será

$$w + \alpha \Delta w$$

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

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

- (i) Enviesar a direção de busca de modo a fazê-la apontar para dentro do interior do ortante não-negativo  $(x, z) \geq 0$ , de tal forma que seja possível mover-se suficientemente através desta direção 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 busca calculadas através de pontos próximos a este limite tendem a ser destorcidas e pouco progresso pode ser feito por elas.

### 2.1.2. O problema de barreira

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

$$(P_\tau) \quad \begin{aligned} \min_x \quad & c^T x - \tau \sum_{i=1}^n \ln x \\ \text{s.a.} \quad & x \in \mathcal{P}^0 \end{aligned} \quad (2.9)$$

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 iterando a ficar no interior da região factível, 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 iterando para o limite, ou seja, quando  $\tau \rightarrow 0$ , o problema  $(P_\tau)$  cada vez mais se parece com o problema  $(P)$ . Esta estratégia só é possível se  $\mathcal{P}^0 \neq \emptyset$ . Além disso se  $\mathcal{P}$  for limitado então tanto  $(P)$  quanto  $(P_\tau)$  admitem solução ótima.

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

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

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

$$Ax = b, \quad (2.10a)$$

$$A^T y + z = c, \quad (2.10b)$$

$$XZe = \tau e, \quad (2.10c)$$

$$(x, z) > 0. \quad (2.10d)$$

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 (2.5) 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 a 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 pode ser encontrados em Bayer and Lagarias [2, 3], Megiddo [33], Sonnevend [41], 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 (2.6) como função do parâmetro de barreira:

$$g(\tau) = c^T x(\tau) - b^T y(\tau) = x(\tau)^T(\tau), \quad (2.11) \quad \text{\texttt{eq:gap-dualit}}$$

isto é, para cada valor de  $\tau$ , o gap de dualidade corresponde ao gap de complementaridade e logo reduzir um significa reduzir o outro. Além disso, como  $XZe - \tau e = 0$  por (2.10c), então  $x_i z_i = \tau, i = 1, \dots, n$  e logo

$$g(\tau) = x(\tau)^T z(\tau) = \sum_{i=1}^n x_i(\tau) z_i(\tau) = n\tau. \quad (2.12) \quad \text{\texttt{eq:gap-dualit}}$$

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

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

Assim o valor objetivo do problema perturbado convergem para aqueles obtidos por uma solução ótima  $(x^*, y^*, z^*)$  do problema original. Mais que isso, vale o seguinte resultado [33]:

**Teorema 2.3.** *Se um problema de PL for primal e dual factível e  $A$  for de posto completo, então*

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

*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*.

Um outro fato que vale a pena chamar atenção é que nos casos em que o PL tenha múltiplas soluções, um MPI termina no centro analítico da face ótima ao invés de em um vértice, como no método simplex. Isso significa que de certa maneira, podemos interpretar esta 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.

Vavasis and Ye [43] 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 [33].

## 2.2. Métodos seguidores de caminho

{sec:path-fol}

### 2.2.1. Modelo geral para métodos seguidores de caminho

A maioria dos métodos primais-duais seguidores de caminho dão 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 (2.5) é resolvido por fazer os pares complementares se alinhem (veja Equação (2.10c)) ao mesmo tempo em que  $(x, z) > 0$ . Como estes passos são enviesados 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 em cada iteração  $\tau$  decresça monotonicamente.

As equações para o encontrar a direção de busca tornam-se

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

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 neste caso, direção de busca encontrada  $\Delta w$  nem sempre podem ser utilizada e o novo iterando é calculado novamente usando um teste da razão, isto é, escolhendo  $\alpha_k$ , tal que

$$w^{k+1} = w^k + \alpha_k \Delta w \quad (2.14) \quad \text{{eq:updating-i}}$$

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 (2.8). 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

$$\tau_k = \frac{(x^k)^T z^k}{n}. \quad (2.15) \quad \text{\texttt{\{eq:tau\_k\}}}$$

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 (2.7).

Com estes conceitos e ideias em mão, podemos definir um modelo geral para métodos primais-duais, dado no Pseudo-Código 1.

---

**Pseudo-Código 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 iterando conforme (2.14).

$k \leftarrow k + 1$

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

---

\texttt{\{alg:modelo-ge\}}

### 2.2.2. Vizinhanças da Trajetória Central

\texttt{\{subsec:neighb\}}

Um método seguidor de caminho segue a trajetória central  $\mathcal{C}$  no interior da região factível em direção a solução ótima. No entanto manter um iterando exatamente sobre  $\mathcal{C}$  é um objetivo muito difícil, senão impossível. Isto porque encontrar um ponto que resolve a condição de complementaridade perturbada (2.10c) para um  $\tau$  específico é um problema tão difícil quanto resolver o próprio PL.

Assim, de modo a ser mais razoável, um método seguidor de caminho faz com que os iterandos fiquem em torno de  $\mathcal{C}$ . Isto significa restringir os iterandos 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 progridem, 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

$$\mathcal{N}_2(\theta) = \{w \in \mathcal{F}^0 : \|XZe - \tau e\|_2 \leq \theta\tau\},$$

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

$$\mathcal{N}_\infty(\gamma) = \{w \in \mathcal{F}^0 : x_i z_i \geq \gamma\tau, \text{ para todo } i = 1, \dots, n\}, \quad (2.16) \quad \text{\texttt{feq:infy-neig}}$$

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

A vizinhança  $\mathcal{N}_2(\theta)$  é mais restrita e portanto define pontos que estão mais próximos da trajetória central. Portanto certos pontos de  $\mathcal{F}^0$  não pertencem à  $\mathcal{N}_2(\theta)$ , não importando quanto façamos  $\theta$  se aproximar de 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  $\tau$ . Ao manter todos os iterandos nos limites destas vizinhanças, métodos seguidores de caminho reduzem todos os produtos  $x_i z_i$  a zero mais ou menos na mesma taxa.

MPI seguidores de caminho são métodos de homotopia continuada [37] 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 mantem-se 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 convergência existentes para PL: algoritmos de passo curto tem complexidade  $\mathcal{O}(\sqrt{n} \log 1/\varepsilon)$ . 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, and Yoshise [31] e Monteiro and Adler [35]. Na prática, como a redução o parâmetro de barreira é muito pequena, estes tipos de métodos são limitados.

Métodos baseados na vizinhança  $\mathcal{N}_\infty(\gamma)$  permitem iterandos que seguem a trajetória central de maneira mais relaxada pois obtém-se mais liberdade para manobrar e é possível se aproximar da fronteira da região factível. De fato, para  $\gamma$  pequeno quase todo conjunto  $\mathcal{F}^0$  está contido em  $\mathcal{N}_\infty(\gamma)$ . Por outro lado, direções computadas a partir de pontos desta vizinhança

---

<sup>2</sup>Tipicamente são usados os seguintes valores para os parâmetros:  $\theta = 0.5$  e  $\gamma = 10^{-3}$  [46, pg. 9].

tem 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. São chamados de *métodos de passo-longo* e implementações eficientes de MPI baseiam-se em alguma variação de métodos de passo-longo.

### Vizinhança Simétrica

{sec:symmetric-

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 iterandos produzam produtos  $x_i z_i$  muito diferentes, já que não impõe uma limite superior para a complementaridade.

Esta dificuldade quando ultrapassada permite resultados muito melhores na prática, já que os pares  $x_i z_i$ , quando mal escalados, influenciam no mau comportamento do método de Newton.

A razão

$$\varrho(XZe) = \frac{\min(XZe)}{\max(XZe)} \quad (2.17) \quad \text{feq:ratio-comp}$$

dá uma medida de proporção entre o maior e o menor par de complementaridade. Claramente  $\varrho(XZe) \in (0, 1)$ , quando o iterando é perfeitamente centrado o temos igual a 1. Uma análise da medida (2.17) pode ser encontrada em [26].

Gondzio [20] e posteriormente Colombo [6], Colombo and Gondzio [7] definiram uma vizinhança que tenta fazer com que em cada iteração tenha-se uma certa *centralidade* dos iterandos. Por centralidade entenda-se a dispersão em relação à média os pares  $x_i z_i$ . Quando a discrepância nos valores dos pares complementares é muito grande, e portanto tem-se pouca centralidade, 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 [6, pp. 26] diz 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 e é definida como

$$\mathcal{N}_s(\gamma) = \left\{ w \in \mathcal{F}^0 : \gamma\tau \leq x_i z_i \leq \frac{1}{\gamma}\tau, i = 1, \dots, n \right\}, \quad (2.18) \quad \text{feq:symmetric-}$$

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

A vizinhança simétrica (2.18) pode ser considerada uma extensão de  $\mathcal{N}_{-\infty}$ , no entanto, ao contrário desta, 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 [6] determinou o valor de  $n$  para o qual a vizinhança simétrica impõe um limite superior mais estreito:

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

Para um  $\gamma = 0.1$ , 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  $\mathcal{O}(n \ln(1/\varepsilon))$ , em que  $\varepsilon$  é a tolerância para convergência [7]. 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 iterandos mais centrados na prática.

De fato, Zhang and Tapia [49], ao provar 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 foi demonstrada errônea [6, 7, 20]

### 2.2.3. Ponto inicial infactível

Note que até agora, estamos assumindo que o método inicia um ponto estritamente factível  $w^0 \in \mathcal{F}^0$ . 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

$$A\Delta x = 0 \quad \text{e} \quad A^T \Delta y + \Delta z = 0. \quad (2.19) \quad \text{\texttt{\{eq:fact-direc}}}$$

Consequentemente a factibilidade de todos os iterandos 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*.

Ao avaliar o gap de complementaridade quando tomando um passo de tamanho  $\alpha$  na direção  $(\Delta x, \Delta y, \Delta z)$  obtemos

$$x(\alpha)^T z(\alpha) = x^T z + \alpha (z^T \Delta x + x^T \Delta z + \alpha^2 \Delta x^T \Delta z) = (1 - \alpha(1 - \eta)) x^T z,$$

em que usamos o fato de que  $\Delta x^T \Delta z = 0$ ,  $x^T z = n\tau$  e que  $z^T \Delta x + x^T \Delta z = -x^T z + n\tau\eta$ .

Assim dividindo por  $n$  obtemos

$$\tau(\alpha) = x(\alpha)^T z(\alpha) / n = (1 - \alpha(1 - \eta))\tau. \quad (2.20) \quad \text{feq:progress-o}$$

A equação (2.20) mostra que o progresso na otimização depende dos parâmetros  $\tau$  e  $\eta$  bem como do tamanho do passo  $\alpha$ . Este fato motiva a escolha adequada destes valores de modo a obter um iterando que progrida em direção ao ótimo de maneira mais rápida.

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 [47], mas neste caso, para tornar o problema sempre factível, aumenta-se o tamanho do problema, o que de certa forma aumenta o tempo computacional por exigir dois *backsolves* extra.

### Métodos Infactíveis

Uma alternativa que permite que essas dificuldades sejam contornadas é utilizar *MPI infactíveis*. De fato métodos implementados que tem importância prática fazem uso pontos infactíveis.

Para fins de notação, dado  $w = (x, y, z)$ , definimos os conjuntos

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

e

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

respectivamente como os conjuntos dos pontos de  $\mathbb{R}^{2n+m}$  não-negativos e positivos nas variáveis  $x$  e  $z$ . Em geral métodos infactíveis exigem apenas que  $w^0 \in \mathcal{Q}^+$ , isto é  $(x^0, z^0)$  esteja no ortante positivo. Neste caso  $r_P^0$  e  $r_D^0$  são não-nulos.

A direção de busca ainda é feita segundo (2.13), e portanto temos um passo de Newton em direção ao ponto  $(x(\eta\tau), y(\eta\tau), z(\eta\tau)) \in \mathcal{C}$ . De fato é fácil provar que se pudéssemos dar um passo completo conseguiríamos eliminar a infactibilidade. No entanto, na prática, o limite da sequência de iterandos é o ponto ótimo e somente quando se está na vizinhança deste ótimo é que se dá um passo que zera os resíduos.

Kojima, Megiddo, and Mizuno [32] 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 utilizar-se uma vizinhança similar à  $\mathcal{N}_{-\infty}(\gamma)$  (veja equação (2.16)),

dada por

$$\mathcal{N}_{-\infty}(\gamma, \beta) = \left\{ w \in \mathcal{Q}^+ : \frac{\|(r_P, r_D)\|}{\tau} \leq \beta \frac{\|(r_P^0, r_D^0)\|}{\tau_0}, x_i z_i \geq \gamma \tau, \forall i = 1, \dots, n \right\}, \quad (2.21) \quad \text{\texttt{eq:infeasible}}$$

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 a factibilidade em cada iteração temos um conjunto limitado superiormente pela medida de complementaridade  $\tau$ . Assim ao reduzirmos  $\tau$  forçamos os resíduos para zero, e portanto reduzindo a complementaridade e a infactibilidade na mesma taxa. 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 reduz linearmente com  $\alpha$ . Assim, como dissemos, se  $\alpha = 1$ , então os resíduos  $r_P$  e  $r_D$  tornam-se nulos, portanto a factibilidade é restaurada e subsequentemente o método infactível torna-se igual a um método factível já que todos os próximos iterandos estarão em  $\mathcal{F}^0$ . Kojima et al. [32] provaram a convergência global de um método com passos infactíveis enquanto Zhang [48] provou a complexidade de  $\mathcal{O}(n^2 \ln(1/\varepsilon))$  para um método deste tipo. Para uma explanação completa de métodos infactíveis veja Wright [46, 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 baseam-se em alguma variação do Método Preditor-Corretor (MPC) devido a Mehrotra [34]. O método de Mehrotra baseia-se no modelo geral para Métodos Seguidores de Caminho, dado no Pseudo-código 1, mas altera a busca de direção puramente Newton com correções que são muito baratas computacionalmente mas que auxiliam a obter uma direção melhor. Além disso, um escolha adaptativa do parâmetro de centragem  $\eta$  é feita em cada iteração. Atualmente, nos códigos mais utilizados, tanto acadêmicos como BPMPD, HOPDM, OOPS, OOQP, PCx, bem como comerciais tais quais Cplex, Mosek and Xpress, algumas heurísticas e ideias de otimização também foram incorporadas, de modo que todos são variantes do MPC.

Este método usa aproximações de ordem maior da trajetória central  $\mathcal{C}$  – abordagem começada por Megiddo [33] e mais tarde desenvolvidas por Monteiro, Adler, and Resende [36] – e o uso de pontos infactíveis. Segundo Wright [46, pg 194], “a contribuição 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, quase a totalidade das implementações do método de Mehrotra usa passos diferentes para as variáveis primais e duais. Esta ideia não tem origem em Mehrotra, mas tem sido usada em quase todos os desenvolvimentos práticos de métodos primais-duais e esta estratégia contribui em até 10% no número de iterações Netlib [46, pg. 195].

Como queremos que  $(x, z)$  permaneçam positivos, empregamos uma busca linear 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,

$$\alpha_P = \alpha_0 \min \left\{ -\frac{x_i}{\Delta x_i} : \Delta x_i < 0 \right\}, \quad \alpha_D = \alpha_0 \min \left\{ -\frac{z_i}{\Delta z_i} : \Delta z_i < 0 \right\} \quad (2.22) \quad \text{\texttt{\{eq:size-step\}}}$$

em que  $\alpha_0$  é um fator que garante positividade estrita e em geral vale 0.9995. Além disso, embora use-se sempre algum tipo de vizinhança, muitas vezes permite-se que o ponto esteja fora da mesma e por isso, perde-se a propriedade da convergência global em favor da eficiência computacional.

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

- (i) Uma *direção preditora*;
- (ii) Um termo de centralização, cujo tamanho é controlado pela escolha de forma adaptativa do parâmetro  $\eta$ ;
- (iii) Uma direção *corretora*.

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:

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

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

$$\begin{bmatrix} b - Ax \\ c - A^T y - z \\ -XZe \end{bmatrix} = \begin{bmatrix} r_P \\ r_D \\ r_C \end{bmatrix} \quad (2.23) \quad \text{\texttt{\{eq:afinne-sca\}}}$$

isto é,  $(\Delta x^{\text{af}}, \Delta y^{\text{af}}, \Delta z^{\text{af}})$  corresponde a direção afim-escala dada em (2.8). Como vimos, esta 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 distraídas por pontos que tem complementaridade pequena, mas que não são ótimos.

### Direção com correção de Segunda Ordem

O MPC de Mehrotra 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  $\tau_a$  como sendo a média dos valores hipotético para os pares complementares resultantes de um passo na direção afim-escala, dado por

$$\tau_a = \frac{g_a}{n} = \frac{(x + \alpha_P^{\text{af}} \Delta x^{\text{af}})^T (z + \alpha_D^{\text{af}} \Delta z^{\text{af}})}{n}. \quad (2.24) \quad \text{\texttt{\{eq:tau-affine\}}}$$

em que  $g_a$  é o gap de complementaridade previsto para a direção afim-escala. Note que  $\tau_a/\tau \in (0, 1)$ . Se  $\tau_a \ll \tau$  então a direção afim-escala é boa e permite uma significativa redução do gap de complementaridade. Por outro lado se a razão  $\tau_a/\tau$  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. Esta escolha faz com que ocorra uma forte centralização, isto é move-se para mais perto de  $\mathcal{C}$ , e logo na próxima iteração, estaremos melhores posicionados para reduzir o gap de complementaridade.

Em [34], a seguinte heurística na escolha do parâmetro de centralização foi usada:

$$\eta = \left( \frac{\tau_a}{\tau} \right)^3 \quad (2.25) \quad \text{\texttt{\{eq:Mehrotra-c\}}}$$

Esta escolha sugere um parâmetro de barreira dado por

$$\tau = \left( \frac{g_a}{x^T z} \right)^2 \tau_a = \left( \frac{g_a}{x^T z} \right)^3 \frac{x^T z}{n}. \quad (2.26) \quad \text{\texttt{\{eq:Mehrotra-b\}}}$$

Assim, se o passo preditor progride bem,  $\eta$  é escolhido pequeno e teremos pouca centralização. Caso contrário,  $\eta$  é escolhido próximo a um e teremos uma forte centralização. Mais geralmente podemos escolher o parâmetro de centralização como

$$\eta = \left( \frac{\tau_a}{\tau} \right)^p = \left( \frac{g_a}{x^T z} \right)^p.$$



Mehrotra [34] estudou o efeito de diferentes valores de  $p = 1, 2, 3, 4$  em um subconjunto da de problemas da **Netlib** e conclui que não havia muita diferença para  $p$  entre 2 e 4.

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

A direção afim-escala corresponde a 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 os pares complementares são dados por

$$(x_i + \Delta x_i^{\text{af}})(z_i + \Delta z_i^{\text{af}}) = x_i z_i + x_i \Delta z_i^{\text{af}} + z_i \Delta x_i^{\text{af}} + \Delta x_i^{\text{af}} \Delta z_i^{\text{af}} = \Delta x_i^{\text{af}} \Delta z_i^{\text{af}}$$

já que ao usarmos o lado direito (2.23) para resolver o sistema de Newton, temos que  $X \Delta Z^{\text{af}} e + Z \Delta X^{\text{af}} e = -X Z e$ . Isso significa que quando um passo completo é dado, os produtos  $x_i z_i$  se transformam em  $\Delta x_i^{\text{af}} \Delta z_i^{\text{af}}$  a 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 compensar este erro.

Podemos ainda ver que idealmente gostaríamos que em nosso iterando, tivéssemos

$$(X + \Delta X)(Z + \Delta Z)e = \eta\tau e$$

o que é equivalente a resolver o sistema não linear

$$Z \Delta x + X \Delta z = -X Z e + \eta\tau e - \Delta X \Delta Z e.$$

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

$$\begin{bmatrix} 0 \\ 0 \\ -\Delta X^{\text{af}} \Delta Z^{\text{af}} e + \eta\tau e \end{bmatrix} \quad (2.27) \quad \text{\texttt{\{eq:corrector-d\}}}$$

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 estão computadas elas são simplesmente adicionadas de maneira a ter a direção desejada:

$$(\Delta w, \Delta y, \Delta z) = (\Delta x^{\text{af}}, \Delta y^{\text{af}}, \Delta z^{\text{af}}) + (\Delta x^c, \Delta y^c, \Delta z^c), \quad (2.28) \quad \text{\texttt{\{eq:Mehrotra-d\}}}$$

e o próximo iterando será dado por

$$(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k, y^k, z^k) + (\alpha_P \Delta x, \alpha_D \Delta y, \alpha_D \Delta z), \quad (2.29) \quad \{\text{eq:Mehrotra-u}\}$$

em que  $\alpha_P$  e  $\alpha_D$  são os tamanhos de passo primal e dual dado por (2.22).

Resumimos o método preditor-corretor de Mehrotra no Pseudo-Código 2.

---

**Pseudo-Código 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 (2.23).

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

Encontre  $\tau$ , dado por (2.26) e calcule  $\eta$  de acordo com (2.25).

Resolva o sistema de Newton com o lado direito dado por (2.27).

Encontre tamanho de passo  $\alpha_P$  e  $\alpha_D$  tal que possa se dar um passo factível na direção

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

Atualize o iterando conforme (2.29).

$k \leftarrow k + 1$

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

---

A principal vantagem do MPC é que na prática produz tamanho de passos maiores antes de violar as restrições de não-negatividade. Isso normalmente significa que se está salvando número de iterações: de fato Mehrotra relata que temos economia da ordem de 35%-50% quando comparada com outras estratégias [34]. Neste caso, o custo é apenas de um *backsolve* 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 são alcançados por essa técnica. De fato a o método de Mehrotra é vantajoso em relação a todas as implementações para programação linear que usam métodos diretos para computar as direções de Newton [6, pg. 40].

### 2.3.2. Método das múltiplas correções de centralização

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

O *método das múltiplas correções* deve-se à Gondzio [20] e é uma tentativa de melhorar a centralização do iterando atual. Esta abordagem busca produzir produtos  $x_i z_i$  mais homogê-

neos e forçar um aumento do tamanho do passo corrigindo o parâmetro de centralização de Mehrotra.

Implicitamente Gondzio [20] utiliza a Vizinhança Simétrica dada em (2.18), de modo a ter produtos perto o suficiente da trajetória central. Nesta descrição o passo será dado na direção

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

em que  $\Delta w^m$  é a direção de Mehrotra (2.28) a qual uma ou mais correções  $\Delta w^g$  podem ser aplicadas. Outras escolhas da direção preditora  $\Delta w^m$  podem ser feitas [7].

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. Computamos o ponto de tentativa

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

e os respectivos produtos complementares  $\tilde{t} = \tilde{X} \tilde{Z} e \in \mathbb{R}^n$ . Note que por aumentar o tamanho do passo, o ponto de tentativa é ineficaz (exterior) mas em compensação é utilizado apenas para buscar o corretor da centralização e logo não é necessário preocupar-se com este fato.

Obviamente os produtos  $\tilde{t}$  dificilmente serão iguais à  $\tau$  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 checamos as componentes as quais 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 iterandos para dentro de  $\mathcal{N}_s(\gamma)$ .

Neste caso o termo  $\Delta w^g$  será encontrado por resolver o sistema de Newton com o lado direito dado por

$$r = \begin{bmatrix} 0 \\ 0 \\ \xi \end{bmatrix} \tag{2.30} \quad \text{\texttt{\{eq:multiply-c}}}$$

em que o alvo  $\xi$  é definido 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 o lado o alvo dado por (2.30) é realmente conseguido.

Dada uma nova direção poderemos repetir o processo de atualizá-la tantas vezes quantas forem convenientes. Basta fazer-se  $\Delta^m \leftarrow \Delta$  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 [20] de modo heurístico. Note-se que o corretor computado é aceito com tanto que o tamanho do passo tenha aumentado de uma fração desejada.

Em códigos como o PCx [9], 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 o *backsolve*. Além disso estima-se a redução do número de iterações. Em [20] o custo da fatoração não foi considerado.

As experiências computacionais apresentadas em [20] mostram que a estratégia das múltiplas correções realmente é efetiva pois os passos primais e duais computados para a direção composta são maiores que aqueles correspondentes à direção preditora. Isso resulta numa diminuição 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-se desta estratégia [46, Apêndice B]. Uma generalização desta pode ser encontrada em [7], no qual a direção de busca é dada por  $\Delta w = \Delta w^m + \omega \Delta w^g$  em que  $\omega$  é um parâmetro de peso  $\omega \in (0, 1]$ . Este peso é encontrado através de uma busca linear tal que a solução  $\hat{\omega}$  maximiza o tamanho de passo  $\alpha$ .

### 2.3.3. Ponto Inicial

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

De fato a escolha de um iterando inicial com boas propriedades é crítica e é uma tarefa difícil tanto para métodos factíveis quanto infactíveis. Pedese que o ponto possua boas propriedades nas seguintes características: centralidade do ponto e magnitude das infactibilidades.

Mehrotra [34] introduziu uma heurística para encontrar um ponto inicial que satisfaça essas hipóteses. Nesta, resolve-se os seguintes problemas de Quadrados Mínimos

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


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

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

em que  $\vartheta_x$ ,  $\vartheta_y$  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 escalamento 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 ótimo, além de ter custo computacional equivalente a uma iteração de um MPI.

Gertz and Wright [17] 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 calcula-se a *norma dos dados*, que é definida como à 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, faz-se  $x$  e  $z$  iguais a essa norma dos dados e  $y$  igual a zero e a partir desse ponto, resolve-se o sistema de Newton (2.8) e dá-se um passo completo na direção afim-escala, encontrando-se o ponto  $(\tilde{x}, \tilde{y}, \tilde{z})$ . Certamente esse procedimento resulta num ponto infactível. Proceda-se então de forma similar à heurística de Mehrotra, isto é, translada-se  $(\tilde{x}, \tilde{y}, \tilde{z})$  para o ortante positivo. Para tal, calcula-se valor o máximo da violação dos limites das variáveis, 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 translada-se o

ponto com o valor

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

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-se 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 que tal que os produtos complementares  $x_i z_i$  sejam maiores que uma fração adequada de  $\tau$  e que a razão

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

não seja muito grande.

Aliás, note-se que se está utilizando a *métrica* dada pela vizinhança  $\mathcal{N}(\gamma, \beta)$  dada em (2.21). Com isso, essas heurísticas tem como objetivo prevenir 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  $\mathcal{N}(\gamma, \beta)$ , deseja-se que a infactibilidade seja levada à zero na mesma taxa que o *gap* de dualidade. Para uma análise aprofundada de estratégias de pontos iniciais em MPI veja [13, 18].

#### 2.3.4. Critério de Parada

Os MPI, ao contrário do simplex, por exemplo, encontram a solução apenas assintoticamente. Por conta do parâmetro de barreira  $\tau$  ele nunca encontra a solução exata do problema de PL. Com isto é necessário um critério de parada que possa decidir quando o iterando atual está próximo o suficiente do conjunto solução. Em códigos práticos frequentemente há uma fase de terminação finita, a qual faz a solução aproximada *pular* para a solução exata, encontrando-se inclusive a base ótima (veja [46, Cap. 7]).

A maioria das implementações de pontos interiores, por estar trabalhando com precisão finita, garante apenas um certo grau de exatidão. Tipicamente a maioria dos códigos utiliza

os seguintes critérios [22]:

$$\frac{\|Ax - b\|}{1 + \|x\|_\infty} \leq 10^{-\beta}, \quad (2.31a)$$

$$\frac{\|A^T y + z - c\|}{1 + \|z\|_\infty} \leq 10^{-\beta}, \quad (2.31b)$$

$$\frac{|c^T x - b^T y|}{1 + |b^T y|} \leq 10^{-\nu}. \quad (2.31c)$$

Os valores de  $\beta, \nu \in \mathbb{N}$  dependem da precisão desejada. Comumente na literatura escolhe-se  $\beta = \nu = 8$ .

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 2.2 (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) e que é utilizado no PCx [9] é

$$\frac{\|Ax - b\|}{1 + \|b\|} \leq 10^{-\beta}, \quad (2.32a)$$

$$\frac{\|A^T y + z - c\|}{1 + \|c\|} \leq 10^{-\beta}, \quad (2.32b)$$

$$\frac{\tau}{1 + |c^T x|} \leq 10^{-\nu}. \quad (2.32c)$$

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  $y$  em cada iteração. O valor de  $\tau$ , é usado em (2.32c) ao invés do gap dual como em (2.31c), pois no ponto ótimo, gap dual e gap de complementaridade são iguais e já que  $\tau$  está calculado, temos mais uma vez uma economia de operações.

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.

# Uma função de mérito polinomial em MPI

{chap:merit-fun

### 3.1. Um KKT escalado

Este trabalho tem por objetivo propor um método para resolver o problema PL. Para isso considere o par primal-dual na sua forma padrão, i.e.,

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

e

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

conforme as equações (2.1) e (2.2).

Como visto no capítulo 2, as condições KKT para este problema são

$$\begin{cases} Ax = b, & (3.1a) \quad \text{\texttt{\{eq:KKT-fac-pr}}} \\ A^T y + z = c, & (3.1b) \quad \text{\texttt{\{eq:KKT-fac-du}}} \\ XZe = 0, & (3.1c) \quad \text{\texttt{\{eq:KKT-comple}}} \\ (x, z) \geq 0. & (3.1d) \quad \text{\texttt{\{eq:KKT-nao-ne}}} \end{cases}$$

Dado qualquer  $w = (x, y, z)$ , os vetores dos resíduos de (3.1),  $r_P, r_D$  e  $r_C$ , podem ser



definidos como

{eq:KKT-residu

$$r_P = Ax - b, \quad (3.2a) \quad \text{{eq:KKT-residu}}$$

$$r_D = A^T y + z - c, \quad (3.2b) \quad \text{{eq:KKT-residu}}$$

$$r_C = XZe. \quad (3.2c) \quad \text{{eq:KKT-residu}}$$

Como o pressuposto  $\mathcal{P}1$  é assumido como verdadeiro, seja  $(x^0, y^0, z^0)$  um ponto inicial que pertença à  $\mathcal{F}^+$ . Então

$$\begin{aligned} r_P^0 &= Ax^0 - b, \\ r_D^0 &= A^T y^0 + z^0 - c, \\ r_C^0 &= X^0 Z^0 e > 0. \end{aligned}$$

No método proposto, precisa-se garantir que ambos  $r_P^0$  e  $r_D^0$  sejam não-negativos – os motivos para isso ficarão mais claros abaixo. Para esse fim, sejam  $H_P$  e  $H_D$  matrizes diagonais, tais que cada entrada de suas diagonais é formada segundo as seguintes regras:

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

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

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

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

Consequentemente,

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

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

$$\begin{cases} H_P(Ax - b) = 0, & (3.3a) \quad \text{{eq:ScaledKKT-}} \\ H_D(A^T y + z - c) = 0, & (3.3b) \quad \text{{eq:ScaledKKT-}} \\ XZe = 0, & (3.3c) \quad \text{{eq:ScaledKKT-}} \\ (x, z) \geq 0, & (3.3d) \quad \text{{eq:ScaledKKT-}} \end{cases}$$

é o mesmo. De fato, note que apenas se está multiplicando cada linha do sistema KKT original por um escalar.

A otimalidade do sistema (3.1) – bem como de (3.3) – é alcançada quando todos os resíduos são nulos, ou melhor dizendo, sempre que esses resíduos forem menores ou iguais a uma – ou mais – tolerância pré-estabelecida. Assim propõe-se um método para resolver o sistema KKT escalado que resolva aproximadamente, em cada iteração, para  $(x, y, z) \in \mathcal{F}^+$  e algum  $\mu > 0$  o sistema

$$\begin{cases} H_P(Ax - b) = 0, & (3.4a) \quad \text{\texttt{\{eq:Homotopy-f}}} \\ H_D(A^T y + z - c) = 0, & (3.4b) \quad \text{\texttt{\{eq:Homotopy-f}}} \\ XZe = \mu e, & (3.4c) \quad \text{\texttt{\{eq:Homotopy-c}}} \\ (x, z) > 0. & (3.4d) \quad \text{\texttt{\{eq:Homotopy-n}}} \end{cases}$$

## 3.2. Direções de busca

### 3.2.1. Direção Afim-escala

\texttt{\{sec:affine-sc}}

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

$$\begin{cases} A\Delta x^{\text{af}} + r_P = 0 & (3.5a) \quad \text{\texttt{\{eq:affine-sca}}} \\ A^T \Delta y^{\text{af}} + \Delta z^{\text{af}} + r_D = 0 & (3.5b) \quad \text{\texttt{\{eq:affine-sca}}} \\ Z\Delta x^{\text{af}} + X\Delta z^{\text{af}} + r_C = 0 & (3.5c) \quad \text{\texttt{\{eq:affine-sca}}} \end{cases}$$

Por meio da equação (3.5c) pode-se encontrar  $\Delta z^{\text{af}}$  como

$$\Delta z^{\text{af}} = -X^{-1}(Z\Delta x^{\text{af}} + r_C). \quad (3.6) \quad \text{\texttt{\{eq:de_z\}}}$$

Se esta equação for usada e além disso  $\Delta z^{\text{af}}$  for substituído na equação (3.5b) obtém-se 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

$$\begin{aligned}
 \Delta x^{\text{af}} &= -D \left( -r_D + X^{-1}r_C - A^T \Delta y^{\text{af}} \right) \\
 &= D \left( r_D - X^{-1}r_C + A^T \Delta y^{\text{af}} \right) \\
 &= D \left( A^T \Delta y^{\text{af}} - t \right),
 \end{aligned} \tag{3.7} \quad \text{\texttt{\{eq:de\_x\}}}$$

em que

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

Usando a fórmula para  $\Delta x^{\text{af}}$ , encontra-se  $\Delta y^{\text{af}}$  e neste caso a equação (3.5c) torna-se  $AD(A^T \Delta y^{\text{af}} - t) + r_P = 0$  e logo

$$\Delta y^{\text{af}} = B(ADt - r_P), \tag{3.8} \quad \text{\texttt{\{eq:de\_y\}}}$$

em que  $B^{-1} = ADA^T$  e  $B^{-1}$  sempre simétrica definida positiva. Consequentemente, para encontrar  $\Delta w^{\text{af}}$  usa-se uma fatoração de Cholesky de  $B^{-1}$  e um *backsolve*.

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

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

a solução do sistema

$$\begin{cases} A\hat{x} - b = 0 \\ A^T \hat{y} + \hat{z} - c = 0 \\ \hat{X} \hat{Z} e = \mu e \end{cases} \tag{3.9} \quad \text{\texttt{\{eq:ideal-syst\}}}$$

A estratégia adotada aqui 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 (3.9), tem-se

$$A(x + \Delta x) - b = A(x + \Delta x^{\text{af}} + \Delta x^c) - b = \underbrace{(Ax - b) + A\Delta x^{\text{af}}}_{=0 \text{ por (3.5a)}} + A\Delta x^c = A\Delta x^c$$

e

$$\begin{aligned} A^T(y + \Delta y) + (z + \Delta z) - c &= A^T(y + \Delta y^{\text{af}} + \Delta y^c) + (z + \Delta z^{\text{af}} + \Delta z^c) - c \\ &= \underbrace{A^T y + z - c + A^T \Delta y^{\text{af}} + \Delta z^{\text{af}}}_{=0 \text{ por (3.5b)}} + A^T \Delta y^c + \Delta z^c \\ &= A^T \Delta y^c + \Delta z^c. \end{aligned}$$

Por outro lado, para a parte da complementaridade de (3.9), é possível escrever

$$\begin{aligned} \hat{X}\hat{Z}e &= (X + \Delta X)(Z + \Delta Z)e = XZe + X\Delta z + Z\Delta x + \Delta X\Delta Ze \\ &= \underbrace{XZe + X\Delta z^{\text{af}} + Z\Delta x^{\text{af}}}_{=0 \text{ por (3.5c)}} + Z\Delta x^c + X\Delta z^c + (\Delta X^{\text{af}} + \Delta X^c)(\Delta Z^{\text{af}} + \Delta Z^c)e \\ &= X\Delta z^c + Z\Delta x^c + \Delta X^c\Delta z^c + \Delta X^{\text{af}}\Delta z^{\text{af}} + \Delta X^{\text{af}}\Delta z^c + \Delta Z^{\text{af}}\Delta x^c \\ &= X\Delta z^c + Z\Delta x^c + \Delta X\Delta z \end{aligned}$$

Usando tais simplificações, obtém-se o seguinte sistema não linear

$$\begin{cases} A\Delta x^c = 0 \\ A^T \Delta y^c + \Delta z^c = 0 \\ X\Delta z^c + Z\Delta x^c + \Delta X\Delta z = \mu e \end{cases} \quad (3.10) \quad \text{\texttt{\{eq:corrector-1}}}$$

O vetor  $\Delta X\Delta z$  corresponde a uma correção de segunda ordem, nos moldes dos trabalhos de Gondzio [20], Mehrotra [34].

A proposta deste trabalho é utilizar uma forma de generalização das correções de ordem superior usadas por esses autores. Essa generalização será feita supondo-se que para algum escalar  $\sigma \in [0, \zeta]$ ,  $\zeta > 0$ , a aproximação

$$\Delta X\Delta z \approx \sigma \Delta X^{\text{af}}\Delta z^{\text{af}} \quad (3.11) \quad \text{\texttt{\{eq:aproximaca}}}$$

seja aceitável<sup>1</sup>.

Em particular note que se  $\sigma = 1$  e  $\mu = \tau_a^3/\tau$  – em que  $\tau_a$  é dado na equação (2.24) e  $\tau = x^T z/n$  – tem-se o método de Mehrotra [34]. Por outro lado, para Gondzio [20],  $\mu$  é

---

<sup>1</sup>Neste caso, a notação  $u \equiv \sigma v$  significa que o  $u$  é um vetor que pertence ao subespaço gerado por  $v$ .

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  $N_s(\gamma)$ . De fato a escolha do valor  $\mu$  e a utilização de correções de ordem superior definem os diferentes tipos de MPI que tem sido utilizados atualmente [46].

### 3.2.3. Combinando direções

Alguns autores como Colombo and Gondzio [7] e também Jarre and Wechs [27] combinam direções de correção utilizando-se de pesos. Os primeiros generalizam e estendem o trabalho de Gondzio [20], fazendo com que a combinação das correções múltiplas tenham um peso, que é escolhido fazendo-se uma busca linear. Já os segundos, propõe um subprograma 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.


Neste trabalho, e baseando-se fortemente nas ideias de Villas-Bôas and Perin [45], a escolha do valor de  $\mu$  e de  $\sigma$  será feita de maneira adiada, e por isso, tais parâmetros bem como o tamanho do passo  $\alpha$ , estão sendo tratados como *variáveis*. Para a escolha desses parâmetros, far-se-á uso de uma função de mérito, a qual será descrita abaixo.

Antes disso, note que a aproximação dada em (3.11) transforma o sistema não linear (3.10) no sistema linear

$$\begin{cases} A\Delta x^c = 0 \\ A^T \Delta y^c + \Delta z^c = 0 \\ X\Delta z^c + Z\Delta x^c = \mu e - \sigma \Delta X^{\text{af}} \Delta z^{\text{af}} \end{cases}$$

ou expressando tal sistema na forma matricial tem-se

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^c \\ \Delta y^c \\ \Delta z^c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mu e - \sigma \Delta X^{\text{af}} \Delta z^{\text{af}} \end{bmatrix}. \quad (3.12) \quad \text{\texttt{\{eq:corrector-1\}}}$$

Note que as matrizes do lado esquerdo das equações em (3.5) e (3.12) são as mesmas. Com isso podemos resolver (3.12) utilizando a mesma fatoração de Cholesky de  $B^{-1}$ . 

De fato, o lado direito de (3.12) pode ser reescrito como

$$\mu \begin{bmatrix} 0 \\ 0 \\ e \end{bmatrix} + \sigma \begin{bmatrix} 0 \\ 0 \\ -\Delta X^{\text{af}} \Delta z^{\text{af}} \end{bmatrix}. \quad (3.13) \quad \text{\texttt{\{eq:corrector-2\}}}$$

Portanto, para  $(\mu, \sigma)$  qualquer – ainda não escolhidos –, é possível encontrar  $\Delta w^c$  por separá-lo

em

$$\Delta w^c = \mu \Delta w^\mu + \sigma \Delta w^\sigma \quad (3.14) \quad \text{\texttt{\{eq:Corrector-\}}}$$

e resolver dois sistemas lineares, similares ao da seção 3.2.1. De fato encontra-se  $\Delta w^\mu$  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, tem-se

$$\begin{aligned} \Delta y^\mu &= B(ADt^\mu) & \Delta y^\sigma &= B(ADt^\sigma) \\ \Delta x^\mu &= D(A^T \Delta y^\mu - t^\mu) & \Delta x^\sigma &= D(A^T \Delta y^\sigma - t^\sigma) \\ \Delta z^\mu &= -X^{-1}(Z \Delta x^\mu - e) & \Delta z^\sigma &= -X^{-1}(Z \Delta x^\sigma + \Delta X^{\text{af}} \Delta z^{\text{af}}) \end{aligned} \quad (3.15) \quad \text{\texttt{\{eq:De\_C\}}}$$

em que

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

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

Define-se 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 como

\text{\texttt{\{eq:next-itera\}}}

$$\hat{x} = x + \alpha(\Delta x^{\text{af}} + \mu \Delta x^\mu + \sigma \Delta x^\sigma) \quad (3.16a)$$

$$\hat{y} = y + \alpha(\Delta y^{\text{af}} + \mu \Delta y^\mu + \sigma \Delta y^\sigma) \quad (3.16b)$$

$$\hat{z} = z + \alpha(\Delta z^{\text{af}} + \mu \Delta z^\mu + \sigma \Delta z^\sigma) \quad (3.16c)$$

Até o presente momento, a tripla  $(\alpha, \mu, \sigma)$ , em que  $\alpha$  é o tamanho do passo ainda não foi escolhida. Para construir a função de mérito dada acima, a abordagem proposta neste trabalho é tratar algebricamente  $(\alpha, \mu, \sigma)$  como uma tripla de variáveis reais, utilizar no máximo três *backsolves* para escolhe-las e finalmente usar a combinação linear das direções  $\Delta w^{\text{af}}$ ,  $\Delta w^\mu$  e  $\Delta w^\sigma$  porém usando a tripla  $(\alpha, \mu, \sigma)$  para determinar as constantes – ou pesos – de tal combinação.

A função de mérito que será proposta mais abaixo, será formada a partir dos resíduos do sistema KKT escalado (3.3). Para tanto, a seção a seguir mostrará que é possível prever, a depender de uma escolha de  $(\alpha, \mu, \sigma)$ , o próximo resíduo deste sistema.

### 3.3. O próximo resíduo

**Definição 3.1.** Definimos  $\rho$ , *vetor de resíduos do sistema* KKT escalado (3.3) para um ponto  $(x, y, z)$  como

$$\rho(x, y, z) = \begin{cases} \rho_P(x, y, z) = H_P(Ax - b) \\ \rho_D(x, y, z) = H_D(A^T y + z - c) \\ \rho_C(x, y, z) = XZe \end{cases} \quad (3.17) \quad \text{\texttt{\{eq:residuals\}}}$$

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  $(x^0, y^0, z^0)$ . Pode-se denotar o *vetor (preditivo) dos resíduos para a próxima iteração  $k+1$*  como

$$\rho^{k+1} \rho(x^{k+1}, y^{k+1}, z^{k+1}).$$

**Observação 3.2.** Se o contexto permitir, pode-se representar o resíduo preditivo  $\rho^{k+1}$  como  $\hat{\rho}$  ou ainda  $\hat{\rho}(\alpha, \mu, \sigma)$  já que, como veremos no Teorema 3.3 a seguir, a escolha de  $(\alpha, \mu, \sigma)$  determinará o próximo resíduo.

Também poderemos fazer algum abuso de notação para fins de clareza. Qualquer vetor  $v$  pode ser representado também pela matriz diagonal  $V = \text{diag}(v)$  e ambas notações serão usadas indistintamente e com o mesmo significado, sempre respeitando as dimensões de maneira adequada. Também, para quaisquer vetores  $u$  e  $v$  se para as matrizes diagonais  $U = \text{diag}(u)$  e  $V = \text{diag}(v)$ , poderemos utilizar a seguinte equivalência  $UVe \equiv uv$ . Ainda fazendo abuso de notação, poderemos substituir  $\xi e$  por  $\eta$ , em que  $\xi$  é um escalar e  $e = (1, \dots, 1)$ , sempre respeitando e adequando as dimensões.

Usando a Definição 3.1, dado um ponto  $(x, y, z)$ , podemos prever o próximo resíduo  $\hat{\rho}$ , a depender de uma escolha de  $(\alpha, \mu, \sigma)$ , através do teorema a seguir.

**Teorema 3.3.** O próximo resíduo para o sistema KKT escalado (3.3) é escrito, em termos da tripla  $(\alpha, \mu, \sigma)$ , como

$$\hat{\rho}(\alpha, \mu, \sigma) = \begin{cases} (\hat{\rho}_L)_\ell = (1 - \alpha)(\rho_L)_\ell, \\ \text{for } \ell = 1, \dots, n + m. \\ (\hat{\rho}_C)_j = (1 - \alpha)(\rho_C)_j + \alpha\mu + \alpha(\alpha - \sigma)(L_{0,0})_j + \alpha^2\Lambda(\mu, \sigma)_j, \\ \text{for } j = 1, \dots, n. \end{cases} \quad (3.18) \quad \text{\texttt{\{eq:next-resid\}}}$$

em que

$$\Lambda(\mu, \sigma) = (\mu^2 L_{2,0} + \mu L_{1,0} + \mu\sigma L_{1,1} + \sigma^2 L_{0,2} + \sigma L_{0,1}) \quad (3.19) \quad \text{\texttt{\{eq:Lambda-mu\}}}$$

e

$$\begin{aligned}
L_{0,0} &= \Delta x^{af} \Delta z^{af} & L_{1,1} &= \Delta x^\mu \Delta z^\sigma + \Delta x^\sigma \Delta z^\mu \\
L_{1,0} &= \Delta x^{af} \Delta z^\mu + \Delta z^{af} \Delta x^\mu & L_{0,1} &= \Delta x^{af} \Delta z^\sigma + \Delta z^{af} \Delta x^\sigma \\
L_{2,0} &= \Delta x^\mu \Delta z^\mu & L_{0,2} &= \Delta x^\sigma \Delta z^\sigma
\end{aligned} \tag{3.20}$$

A prova do Teorema 3.3 é consequência direta da aplicação dos Lemas A.1 e A.2, que atacam a parte linear e não linear de  $\hat{\rho}$  de forma separada. Esses lemas e suas demonstrações encontram-se no Apêndice A.

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

**Corolário 3.4.** *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 (3.3), pelo Lema A.1,  $\hat{\rho}_L(\alpha, \mu, \sigma) = (1 - \alpha)\rho_L$ . 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  $(x, y, z) \in \mathcal{Q}^+$ . Além disso, faremos a cada iteração o teste da razão, isto é, escolheremos  $\alpha_k$ , tal que

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

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

### 3.4. Uma função de mérito polinomial

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 [48], por exemplo, quer escolher o tamanho de passo de seu algoritmo, tal que a complementaridade e a factibilidade primal-dual sejam reduzidas. Para isso, o 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.

Escrever sobre a função de mérito do Mehrotra [34] e do PCx Czyzyk, Wright, Wagner, and Mehrotra [8], Czyzyk et al. [9]

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, será usada a soma entre a norma-1 da factibilidade – esta escolha ficará mais clara mais adiante – e a média do *gap* de complementaridade.



**Definição 3.5** (Função de Mérito). Definimos a *função de mérito* de um ponto  $(x, y, z)$  como

$$\varphi(x, y, z) = \frac{1}{m+n} \|\rho_L\|_1 + \frac{x^T z}{n} \quad (3.21)$$

em que  $\rho_L$  é dado pela (3.17) no ponto  $(x, y, z)$ .

A Definição 3.5 é uma das possíveis maneiras de medir quão perto da solução está um ponto  $(x, y, z)$ . Em particular, esta possui algumas propriedades que serão exploradas para escolher-se a tripla  $(\alpha, \mu, \sigma)$ , a saber:

- (i) Se  $(x^*, y^*, z^*)$  é solução de (3.3), então  $\varphi(x^*, y^*, z^*) = 0$ ;
- (ii) A utilização das matrizes  $H_P$  e  $H_D$  e o Corolário 3.4 garante que dado  $(x, y, z) \in \mathcal{Q}^+$ , então  $\rho_L(x, y, z) \geq 0$ . Por isso,

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

e portanto não é preciso se preocupar com a não-diferenciabilidade da norma-1;

- (iii) Pode-se reescrever o *gap* de complementaridade como

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

- (iv) A média do *gap* de complementaridade,  $x^T z/n$ , é usada em geral como valor para o parâmetro de barreira (vide Seção 2.1.2) e logo, em uma solução ótimo, este valor é nulo;
- (v) Também por causa do Corolário 3.4, dado  $(x, y, z) \in \mathcal{Q}^+$ , vale

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

Por conta da propriedade (iii), podemos reescrever (3.21) como

$$\varphi(x, y, z) = \frac{1}{n+m} \sum_{\ell=1}^{n+m} (\rho_L)_\ell + \frac{1}{n} \sum_{i=1}^n (\rho_C)_i \quad (3.22)$$

Para fins de notação, define-se 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, pode-se representar a função de mérito (3.22) como

$$\varphi(x, y, z) = \overline{\rho_L} + \overline{\rho_C} \quad (3.23) \quad \text{\texttt{\{eq:merit-func\}}}$$

Em vista disso e do Teorema 3.3, pode-se definir a função mérito para o próximo ponto  $(\hat{x}, \hat{y}, \hat{z})$  a depender de uma escolha de  $(\alpha, \mu, \sigma)$ , i.e., temos a seguinte definição. \texttt{\{def:predictiv\}}

**Definição 3.6** (Função de Mérito Preditiva). A *função de mérito preditiva*  $\hat{\varphi}$ , na iteração  $k$  será

$$\hat{\varphi}(x^k, y^k, z^k) = \overline{\hat{\rho}_L}(x^k, y^k, z^k) + \overline{\hat{\rho}_C}(x^k, y^k, z^k).$$

Como, por conta da equação (3.18) sabe-se a expressão do próximo resíduo, a depender de uma escolha de  $(\alpha, \mu, \sigma)$ , reescreve-se a função de mérito preditiva como

$$\hat{\varphi}(\alpha, \mu, \sigma) = \overline{\hat{\rho}_L}(\alpha, \mu, \sigma) + \overline{\hat{\rho}_C}(\alpha, \mu, \sigma) \quad (3.24) \quad \text{\texttt{\{eq:merit-func\}}}$$

Finalmente, usando as propriedades (i-v) acima descritas e o próximo teorema, pode-se escrever  $\hat{\varphi}$  como uma função polinomial que depende das variáveis  $(\alpha, \mu, \sigma)$ . Com isso, alcança-se o objetivo de encontrar uma função de mérito que tenha boas propriedades matemáticas e que ao mesmo tempo dê uma medida adequada da qualidade solução. \texttt{\{thm:varphi\}}

**Teorema 3.7.** A *função de mérito preditiva* 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:

$$\hat{\varphi}(\alpha, \mu, \sigma) = (1 - \alpha)(\overline{\rho_{L_k}} + \overline{\rho_{C_k}}) + \alpha\mu + \alpha(\alpha - \sigma)\overline{L_{0,0_k}} + \alpha^2\overline{\Lambda(\mu, \sigma)_k} \quad (3.25) \quad \text{\texttt{\{eq:merit-func\}}}$$

em que

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

*Demonstração.* Aplicando diretamente o Teorema 3.3 na Definição 3.6 temos que

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

■

*Observação 3.8.* 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 (3.12) e (3.13), tanto  $\Delta x^\mu$  e  $\Delta z^\mu$  quanto  $\Delta x^\sigma$  e  $\Delta z^\sigma$  são vetores ortogonais. Nada obstante, na prática ter-se-á  $\overline{L_{2,0}} \geq \text{tol}$  e  $\overline{L_{0,2}} \geq \text{tol}$ , já que só é possível resolver os sistemas (3.12) através de aritmética de ponto flutuante[19]. Com isso, por enquanto, tais escalares serão considerados não nulos. Posso falar disso depois, na parte de implementação?

Definindo os coeficientes

$$\begin{cases} a_{0,0,0} = (\overline{\rho_L} + \overline{\rho_C}) & a_{2,1,1} = \overline{L_{1,1}} \\ a_{1,0,0} = -(\overline{\rho_L} + \overline{\rho_C}) & a_{2,1,0} = \overline{L_{1,0}} \\ a_{1,1,0} = 2 & a_{2,2,0} = \overline{L_{2,0}} = 0 \\ a_{1,0,1} = -\overline{L_{0,0}} & a_{2,0,1} = \overline{L_{0,1}} \\ a_{2,0,0} = \overline{L_{0,0}} & a_{2,0,2} = \overline{L_{0,2}} = 0 \end{cases}$$

e adotando a convenção de que se o coeficiente  $a_{\ell,p,s}$  não está definido acima, então será nulo, pode-se 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

$$\begin{aligned}\hat{\varphi}(\alpha, \mu, \sigma) = & a_{0,0,0} + a_{1,0,0}\alpha + a_{1,1,0}\alpha\mu + a_{1,0,1}\alpha\sigma \\ & + a_{2,0,0}\alpha^2 + a_{2,1,0}\alpha^2\mu + a_{2,0,1}\alpha^2\sigma \\ & + a_{2,2,0}\alpha^2\mu^2 + a_{2,0,2}\alpha^2\sigma^2 \\ & + a_{2,1,1}\alpha^2\mu\sigma\end{aligned}\quad (3.26) \quad \{\text{eq:varphi-pol}\}$$

Estes 10 (8) coeficientes de três índices que são não nulos são as únicas coisas que são computadas nesta implementação, além das direções.

O método proposto encontra em cada iteração o argumento do mínimo global  $(\alpha^*, \mu^*, \sigma^*)$  para a função polinomial  $\hat{\varphi}(\alpha, \mu, \sigma)$ . 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.

Na seção 2.2.2, ficou claro que bons métodos baseiam-se na utilização de vizinhanças que fazem com que os iterandos estejam a distâncias razoáveis da trajetória central. Na próxima seção será mostrado que é possível escrever tais vizinhanças, em particular a vizinhança simétrica  $\mathcal{N}_s$  como um conjunto de funções polinomiais  $\psi_\ell : \mathbb{R}^3 \rightarrow \mathbb{R}$  nas variáveis  $(\alpha, \mu, \sigma)$ . Tais funções serão usadas como restrições do problema de otimização global exposto acima, garantindo que o próximo iterando não só reduza a função de mérito, mas também que o próximo ponto tenha boas propriedades.

### 3.5. Vizinhança simétrica como conjunto de funções polinomiais

Como visto no capítulo anterior, Colombo and Gondzio [7] propuseram a vizinhança simétrica  $\mathcal{N}_s(\gamma)$  dada na Equação (2.18) – baseada na vizinhança  $\mathcal{N}_{-\infty}(\gamma)$  – com objetivo de melhorar a *centralidade* de um ponto, i.e., como o espalhamento dos produtos complementares  $x_i z_i$ , for  $i = 1, \dots, n$ . Estes autores entendem que tal vizinhança é a pedra angular através da qual suas implementações obtém boas performances, já que tais condições impedem que os pares da complementaridade torne-se muito grandes ou muito pequenos antes do tempo adequado.

Pode-se interpretar  $\mathcal{N}_s(\gamma)$  da seguinte maneira:

- $x_i z_i$  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.

Como este trabalho baseia-se em pontos infactíveis mas ao mesmo tempo pretende utilizar as boas propriedades da vizinhaça  $\mathcal{N}_s(\gamma)$ , é necessário estendê-la. Como visto no Capítulo anterior, Kojima et al. [32] assim como Wright [46, pg. 110] utilizaram em seus métodos infactíveis a vizinhaça  $\mathcal{N}_{-\infty}(\gamma, \beta)$ .

Assim, usando a Definição 3.1, definimos a vizinhaça simétrica infactível  $\mathcal{N}_s(\gamma, \beta)$ , para o sistema KKT escalado (3.3) como

$$\mathcal{N}_s(\gamma, \beta) = \left\{ (x, y, z) \in \mathcal{Q}^+ : \frac{\|\rho_L\|}{\tau} \leq \beta \frac{\|\rho_L^0\|}{\tau_0}, \gamma\tau \leq x_i z_i \leq \frac{1}{\gamma}\tau, \forall i = 1, \dots, n \right\}, \quad (3.27)$$

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

Até agora, o método proposto resolverá o KKT escalado (3.3) e além disso usará a função de mérito (3.25) como guia para a escolha das variáveis  $(\alpha, \mu, \sigma)$  e por conseguinte dodo próximo ponto. Por conta do exposto acima, fazer este próximo ponto pertencer à vizinhaça  $\mathcal{N}_s(\gamma, \beta)$  é uma garantia de que os iterandos estarão numa boa distância da trajetória central. Para isso basta que seja possível escrever funções nas variáveis  $(\alpha, \mu, \sigma)$  que sirvam de restrições para o subproblema de minimização da função de mérito  $\varphi$  e que garantam que o ponto escolhido esteja em  $\mathcal{N}_s(\gamma, \beta)$ . Isso é possível já que tal vizinhaça é definida com base nos resíduos lineares e complementares –  $\|\rho_L\|$  e  $x_i z_i$  – de KKT. O seguinte Teorema garante isso.

**Teorema 3.9.** *Um ponto  $(x, y, z) \in \mathcal{Q}^+$  pertence à  $\mathcal{N}_s(\gamma, \beta)$  para o sistema KKT escalado se valerem as seguintes desigualdades*

$$\overline{\rho_L}(x, y, z) \leq \beta_L \overline{\rho_C}(x, y, z), \quad (3.28a)$$

$$\gamma \overline{\rho_C}(x, y, z) \leq (\rho_C)_i(x, y, z) \leq \frac{1}{\gamma} \overline{\rho_C}(x, y, z), \quad (3.28b)$$

em que

$$\beta_L = \frac{\beta}{n} \frac{\|\rho_L^0\|}{\overline{\rho_{C_0}}}.$$

*Demonstração.* Para facilitar a compreensão, deixamos de mostrar que os vetores em questão dependem do ponto  $(x, y, z)$ . Para demonstrar, comece por considerar que  $\overline{\rho_L} = \|\rho_L\|_1/n$ . Assim,  $\overline{\rho_L} \leq \beta_L \overline{\rho_C}$  implica que  $\|\rho_L\|_1 \leq \beta \frac{\|\rho_L^0\|}{\overline{\rho_{C_0}}} \overline{\rho_C}$ . Por equivalência de normas temos que

$$\|\rho_L\| \leq \|\rho_L\|_1 \leq \beta \frac{\|\rho_L^0\|}{\overline{\rho_{C_0}}} \overline{\rho_C}.$$

Note finalmente que  $\rho_C = XZe$  e portanto  $(\rho_C)_i = x_i z_i$  e  $\overline{\rho_C} = x^T z/n$ . Claramente, se (3.29)

for satisfeita então a primeira propriedade que define  $\mathcal{N}_s(\gamma, \beta)$ . Por sua vez (3.30) corresponde à vizinhança simétrica, o que finaliza a demonstração. ■

Quando cada ponto  $(x, y, z)$  pertence à sequência  $w^k$ , então podemos escrever  $\overline{\rho_{C_{k+1}}} = \hat{\varphi}_C$  e  $\overline{\rho_{L_{k+1}}} = \hat{\varphi}_L$ , em portanto

$$\hat{\varphi}_L = (1 - \alpha)\overline{\rho_{L_k}}$$

e

$$\hat{\varphi}_C = (1 - \alpha)\overline{\rho_{C_k}} + \alpha\mu + \alpha(\alpha - \sigma)\overline{L_{0,0}} + \alpha^2\overline{\Lambda(\mu, \sigma)}.$$

Assim as restrições da Teorema 3.9 podem ser reescritas como

$$\hat{\varphi}_L \leq \beta_L \hat{\varphi}_C, \quad (3.29) \quad \text{\texttt{eq:symmetric-}}$$

$$\gamma \hat{\varphi}_C \leq (\rho_C)_i \leq \frac{1}{\gamma} \hat{\varphi}_C. \quad (3.30) \quad \text{\texttt{eq:symmetric-}}$$

### 3.6. Subproblema de Otimização de Polinômios

A escolha de  $(\alpha, \mu, \sigma)$  será feita através do seguinte subproblema de otimização: minimizar a função de mérito  $\hat{\varphi}$ , dada pelo Teorema 3.7 restrita às desigualdades dadas no Teorema 3.9 – sempre levando em conta o teste da razão.

Portanto, para expressar tal problema de forma apropriada, note que assim como  $\varphi$ , as inequações (3.28) podem ser computadas em termos de  $(\alpha, \mu, \sigma)$ , utilizando-se das desigualdades dadas em 3.30, produzindo um conjunto de  $2n + 1$  restrições. Com efeito, expandindo 3.30 tem-se:

\text{\texttt{eq:explicit-s}}

$$(1 - \alpha)(\rho_C)_i + \alpha\mu + \alpha(\alpha - \sigma)(L_{0,0})_i + \alpha^2\Lambda(\mu, \sigma)_i - \frac{1}{\gamma}\hat{\varphi}_C(\alpha, \mu, \sigma) \leq 0 \quad (3.31a)$$

$$\gamma\hat{\varphi}_C(\alpha, \mu, \sigma) - [(1 - \alpha)(\rho_C)_i + \alpha\mu + \alpha(\alpha - \sigma)(L_{0,0})_i + \alpha^2\Lambda(\mu, \sigma)_i] \leq 0 \quad (3.31b)$$

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

$$(1 - \alpha)\overline{\rho_L} - \beta_L \hat{\varphi}(\alpha, \mu, \sigma) \leq 0 \quad (3.31c)$$

Seja  $\psi : \mathbb{R}^3 \rightarrow \mathbb{R}^{2q+1}$ ,  $(\alpha, \mu, \sigma) \mapsto (\psi_1(\alpha, \mu, \sigma), \dots, \psi_{2n+1}(\alpha, \mu, \sigma))$  a função vetorial que representa o lado esquerdo das inequações (3.31). Note que cada  $\psi_\ell$  é um polinômio nas

variáveis  $(\alpha, \mu, \sigma)$ . De fato é possível escrever cada  $\psi_\ell$  como o seguinte polinômio

$$\psi_\ell(\alpha, \mu, \sigma) = \sum_{i=0}^2 \sum_{j=0}^2 \sum_{p=0}^2 b_{i,j,p} \alpha^i \mu^j \sigma^p,$$

onde nem todos os coeficientes  $b_{i,j,p}$  são diferentes de zero. Mais que isso, os coeficientes que são diferentes de zero são os que estão relacionados com os mesmos monômios dados na Equação (3.26), i.e.,  $\psi_\ell$  pode ser expandido como

$$\begin{aligned} \psi_\ell(\alpha, \mu, \sigma) = & b_{0,0,0} + b_{1,0,0}\alpha + b_{1,1,0}\alpha\mu + b_{1,0,1}\alpha\sigma \\ & + b_{2,0,0}\alpha^2 + b_{2,1,0}\alpha^2\mu + b_{2,0,1}\alpha^2\sigma \\ & + b_{2,2,0}\alpha^2\mu^2 + b_{2,0,2}\alpha^2\sigma^2 \\ & + b_{2,1,1}\alpha^2\mu\sigma \end{aligned} \quad (3.32) \quad \text{\texttt{\{eq:psi-poly-f}}}$$

Portanto, o subproblema de otimização global pode ser escrito como

$$\begin{aligned} \min_{(\alpha, \mu, \sigma)} \quad & \hat{\varphi}(\alpha, \mu, \sigma) \\ \text{s.t.} \quad & \begin{cases} \psi(\alpha, \mu, \sigma) \leq 0 \\ l \leq (\alpha, \mu, \sigma) \leq u \end{cases} \end{aligned} \quad (3.33) \quad \text{\texttt{\{eq:pop-subpro}}}$$

em que  $l$  e  $u$  são as canalizações de  $(\alpha, \mu, \sigma)$ .

O algoritmo proposto neste capítulo pode ser resumido no Pseudo-código 3.

---

**Pseudo-Código 3** Resumo do Método de Escolha Adiada.

---

```

1: procedure 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:   for  $k=1,2,3$  do
5:     Resolva (3.5) e encontre  $((\Delta x^{\text{af}})^k, (\Delta y^{\text{af}})^k, (\Delta z^{\text{af}})^k)$ .
6:     Encontre  $((\Delta x^\mu)^k, (\Delta y^\mu)^k, (\Delta z^\mu)^k)$  w  $((\Delta x^\sigma)^k, (\Delta y^\sigma)^k, (\Delta z^\sigma)^k)$  utilizando (3.15).
7:     Calcule  $\hat{\varphi}(\alpha^k, \mu^k, \sigma^k)$  usando (3.24) e  $\psi(\alpha^k, \mu^k, \sigma^k)$  usando (??).
8:     Encontre  $(\alpha^k, \mu^k, \sigma^k)$  por resolver o subproblema de otimização global (3.33).
9:     De próximo o passo conforme

$$x^{k+1} = x^k + \alpha^k((\Delta x^{\text{af}})^k + \mu_k(\Delta x^\mu)^k + \sigma_k(\Delta x^\sigma)^k)$$


$$y^{k+1} = y^k + \alpha^k((\Delta y^{\text{af}})^k + \mu_k(\Delta y^\mu)^k + \sigma_k(\Delta y^\sigma)^k) .$$


$$z^{k+1} = z^k + \alpha^k((\Delta z^{\text{af}})^k + \mu^k(\Delta z^\mu)^k + \sigma_k(\Delta z^\sigma)^k)$$

10:     $k \leftarrow k + 1$ 
11:   end for
12: end procedure

```

---

{alg:optimized}



## **Capítulo 4.**

### **Resultados de Convergência**

## **Capítulo 5.**

### **Experimentos Numéricos**

## **Considerações Finais**

Aqui deve-se colocar as considerações finais e conclusões da tese.

## Referências Bibliográficas

- [1] I. Adler, M. G. C. Resende, G. Veiga, and N. Karmarkar. An implementation of Karmarkar's algorithm for linear programming. *Mathematical Programming*, 44(1-3):297–335, 1989.
- [2] D. A. Bayer and J. C. Lagarias. The Nonlinear Geometry of Linear Programming. I Affine and Projective Scaling Trajectories. *Transactions of the American Mathematical Society*, 314(2):499–526, Aug. 1989.
- [3] D. A. Bayer and J. C. Lagarias. The Nonlinear Geometry of Linear Programming. II Legendre Transform Coordinates and Central Trajectories. *Transactions of the American Mathematical Society*, 314(2):527–581, Jan. 1989.
- [4] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows*. Wiley, New York, 4 edition, Dec. 2009.
- [5] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: A survey. *Operations Research*, 29(6):1039–1091, 1981.
- [6] M. Colombo. *Advances in Interior Point Methods for Large-Scale Linear Programming*. PhD thesis, University of Edinburgh, Edinburgh, 2008.
- [7] M. Colombo and J. Gondzio. Further development of multiple centrality correctors for interior point methods. *Computational Optimization and Applications*, 41(3):277–305, 2008.
- [8] J. Czyzyk, S. J. Wright, M. Wagner, and S. Mehrotra. PCx User Guide: Supplement for PCxGUI. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, US Department of Energy, Nov. 1997.
- [9] J. Czyzyk, S. Mehrotra, M. Wagner, and S. J. Wright. PCx: an interior-point code for linear programming. *Optimization Methods and Software*, 11(1):397–430, 1999.
- [10] G. B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation: Proceedings of a Conference.*, pages 339–347. John Wiley & Sons, New York, 1951.

- [11] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey, 1 edition, 1963.
- [12] G. B. Dantzig. Linear programming. *Operations Research*, 50(1):42–47, 2002.
- [13] M. D’Apuzzo, V. Simone, and D. Serafino. Starting-point strategies for an infeasible potential reduction method. *Optimization Letters*, 4(1):131–146, Oct. 2009.
- [14] I. I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Mathematics Doklady*, 8(3):674–675, 1967.
- [15] A. V. Fiacco and G. P. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*. John Wiley & Sons, New York, 1968.
- [16] K. R. Frisch. The logarithmic potential method of convex programming. Technical report, University Institute of Economics, Oslo, 1955.
- [17] E. M. Gertz and S. J. Wright. Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software*, 29(1):58–81, Mar. 2003.
- [18] E. M. Gertz, J. Nocedal, and A. Sartendar. A starting point strategy for nonlinear interior methods. *Applied Mathematics Letters*, 17(8):945–952, 2004.
- [19] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, 3 edition, Oct. 1996.
- [20] J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6(2):137–156, 1996.
- [21] J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, May 2012.
- [22] J. Gondzio and T. Terlaky. A computational view of interior point methods. In J. E. Beasley, editor, *Advances in linear and integer programming*, pages 103–144. Oxford University Press, Inc., Oxford, UK, 1996.
- [23] O. Güler. Generalized Linear Complementarity Problems. *Mathematics of Operations Research*, 20(2):441–448, May 1995.
- [24] O. Güler, C. Roos, T. Terlaky, and J.-P. Vial. A survey of the implications of the behavior of the central path for the duality theory of linear programming. *Management science*, 41(12):1992–1934, May 1995.

- [25] J. A. J. Hall and K. I. M. McKinnon. Hyper-sparsity in the revised simplex method and how to exploit it. *Computational Optimization and Applications*, 32(3):259–283, 2005.
- [26] B. Jansen, C. Roos, and T. Terlaky. A family of polynomial affine scaling algorithms for positive semidefinite linear complementarity problems. *SIAM Journal on Optimization*, 7(1):126–140, 1997.
- [27] F. Jarre and M. Wechs. Extending Mehrotra’s corrector for linear programs. *Advanced Modeling and Optimization*, 1(2):38–60, 1999.
- [28] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.
- [29] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.
- [30] V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities III: Proceedings of the Third Symposium on Inequalities*, pages 159–175. Academic Press, New York, 1972.
- [31] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial-time algorithm for a class of linear complementarity problems. *Mathematical Programming*, 44:1–26, 1989.
- [32] M. Kojima, N. Megiddo, and S. Mizuno. A primal-dual infeasible-interior-point algorithm for linear programming. *Mathematical Programming*, 61(3):263–280, 1993.
- [33] N. Megiddo. Pathways to the optimal set in linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming Interior-point and related methods*, pages 131–158. Springer-Verlag, New York, 1989.
- [34] S. Mehrotra. On the Implementation of a Primal-Dual Interior Point Method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.
- [35] R. D. C. Monteiro and I. Adler. Interior path following primal-dual algorithms. Part I: Linear programming. *Mathematical Programming*, 44:27–41, 1989.
- [36] R. D. C. Monteiro, I. Adler, and M. G. C. Resende. A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, 15(2):191–214, 1990.
- [37] J. L. Nazareth. Homotopy techniques in linear programming. *Algorithmica*, 1(1-4):529–535, 1986.

- [38] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. Integer and combinatorial optimization. John Wiley & Sons, New York, 1999.
- [39] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, New York, 1986.
- [40] R. Shamir. The Efficiency of the Simplex Method: A Survey. *Management science*, 33(3):301–334, Mar. 1987.
- [41] G. Sonnevend. An “analytical centre” for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In A. Prékopa, J. Szelezsáan, and B. Strazicky, editors, *System Modelling and Optimization: proceedings of the 12th IFIP Conference, Budapest, Hungary, September 2-6, 1985*, pages 866–875. Springer-Verlag, Berlin/Heidelberg, 1986.
- [42] R. J. Vanderbei. *Linear programming: foundations and extensions*. International Series in Operations Research and Management Science. Springer-Verlag, New York, 3 edition, 2007.
- [43] S. A. Vavasis and Y. Ye. A primal-dual interior point method whose running time depends only on the constraint matrix. *Mathematical Programming*, 74(1):79–120, July 1996.
- [44] F. R. Villas-Bôas. *Escolha adiada do parâmetro de penalização e do tamanho de passo em algoritmos de Pontos Interiores*. PhD thesis, IMECC/Unicamp, Campinas, 2000.
- [45] F. R. Villas-Bôas and C. Perin. Postponing the choice of penalty parameter and step length. *Computational Optimization and Applications*, 24(1):63–81, 2003.
- [46] S. J. Wright. *Primal-dual interior point methods*. SIAM, Philadelphia, PA, 1 edition, 1997.
- [47] Y. Ye, M. J. Todd, and S. Mizuno. An  $O(nL)$ -Iteration Homogeneous and Self-Dual Linear Programming Algorithm. *Mathematics of Operations Research*, 19(1):53–67, Feb. 1994.
- [48] Y. Zhang. On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem. *SIAM Journal on Optimization*, 4(1):208–227, 1994.
- [49] Y. Zhang and R. A. Tapia. A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming. *SIAM Journal on Optimization*, 3(1):118–133, 1993.

## Apêndice A.

### Resultados Técnicos

A prova do Teorema 3.3 é consequência direta dos Lemas A.1 e A.2, cujas demonstrações seguem abaixo.

**Lema A.1.** *O resíduo da parte linear de (3.3), para o próximo iterando é escrito como*

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

*Demonstração.* Para a equação (3.3a) temos

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

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

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

Por outro lado, pela equação (3.10),  $A\Delta x^{\text{c}} = 0$ . Logo  $\hat{\rho}_P = (1 - \alpha)\rho_P$ .

A prova para parte dual da factibilidade é similar. Com efeito,



$$\begin{aligned}
\hat{\rho}_D &= H_D(A^T \hat{y} + \hat{z} - c) \\
&= H_D \left[ A^T \left( y + \alpha(\Delta y^{\text{af}} + \Delta y^c) \right) + \left( z + \alpha(\Delta z^{\text{af}} + \Delta z^c) \right) - c \right] \\
&= H_D \left[ A^T y + z - c + \alpha(A^T \Delta y^{\text{af}} + \Delta z^{\text{af}}) \right] + \\
&\quad + H_D \underbrace{\left( A^T \Delta y^c + \Delta z^c \right)}_{=0 \text{ por (3.10)}}
\end{aligned} \tag{A.1}$$

Como  $A^T \Delta y^{\text{af}} + \Delta z^{\text{af}} = c - A^T y - z$ , a equação (A.1) 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,

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

■ \lemma:nonline

**Lema A.2.** *O resíduo da parte da complementaridade de (3.3), para o próximo iterando é escrito como*

$$\hat{\rho}_C = (1 - \alpha)\rho_C + \alpha\mu e + \alpha(\alpha - \sigma)L_{0,0} + \alpha^2\Lambda(\mu, \sigma). \tag{A.2} \quad \text{\eq:next-resid}$$

em que os vetores  $L_{i,k}$ ,  $i, k \in 0, 1, 2$  são definidos por (3.20).

*Demonstração.* Para encontrar o resíduo da parte complementar para o próximo iterando, considere que pode-se escrever  $\hat{\rho}_C = \hat{x}\hat{z}$ . Então

$$\begin{aligned}
\hat{x}\hat{z} &= \left( x + \alpha(\Delta x^{\text{af}} + \Delta x^c) \right) \left( z + \alpha(\Delta z^{\text{af}} + \Delta z^c) \right) \\
&= \alpha^2(\Delta x^{\text{af}} + \Delta x^c)(\Delta z^{\text{af}} + \Delta z^c) + \alpha \left[ (x\Delta z^{\text{af}} + z\Delta x^{\text{af}}) + (x\Delta z^c + z\Delta x^c) \right] + xz
\end{aligned} \tag{A.3} \quad \text{\eq:next-resid}$$

Note que  $(x\Delta z^{\text{af}} + z\Delta x^{\text{af}}) = -xz$  por (3.5). Além disso usando a equação (3.14), 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  $(x\Delta z^c + z\Delta x^c) = \mu e - \sigma\Delta x^{\text{af}}\Delta z^{\text{af}}$ .

Assim, a equação (A.3) torna-se

$$\begin{aligned}
\hat{x}\hat{z} &= \alpha^2(\Delta x^{\text{af}} + \Delta x^c)(\Delta z^{\text{af}} + \Delta z^c) + \alpha(-xz + \mu e - \sigma\Delta x^{\text{af}}\Delta z^{\text{af}}) + xz \\
&= (1 - \alpha)xz + \alpha\mu e + \alpha(\alpha - \sigma)\Delta x^{\text{af}}\Delta z^{\text{af}} +
\end{aligned} \tag{A.4a} \quad \text{\eq:next-resid}$$

$$+ \alpha^2 \left( \Delta x^{\text{af}}\Delta z^c + \Delta z^{\text{af}}\Delta x^c + \Delta x^c\Delta z^c \right). \tag{A.4b} \quad \text{\eq:next-resid}$$

Note que 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

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

Consequentemente a equação (A.4b) pode ser expressa como

$$\begin{aligned}\alpha^2 (\mu^2 \Delta x^\mu \Delta z^\mu + \mu \sigma (\Delta x^\mu \Delta z^\sigma + \Delta x^\sigma \Delta z^\mu) + \\ + \mu (\Delta x^{\text{af}} \Delta z^\mu + \Delta z^{\text{af}} \Delta x^\mu) + \sigma (\Delta x^{\text{af}} \Delta z^\sigma + \Delta z^{\text{af}} \Delta x^\sigma) + \sigma^2 \Delta x^\sigma \Delta z^\sigma). \quad (\text{A.5}) \quad \text{\texttt{eq:next-resid}}\end{aligned}$$

Nos agora somamos as a equação (A.5) com a equação (A.4a). Definindo os vetores  $\Lambda(\mu, \sigma)$  como em (3.19) e  $L_{i,j}$  como em (3.20) e substituindo onde for possível, finalmente encontramos a equação (A.2) levando em conta que  $xz = \rho_C$ . ■