

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO  
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

Resolução de Problemas de Programação Linear por Partes Via Algoritmos de  
Pontos Interiores

Por: Mario Conrado Cavichia  
Orientador: Marcos Nereu Arenales  
Co-Orientador: Christiano Lyra Filho

Dissertação submetida à  
Faculdade de Engenharia  
Elétrica e de Computação  
da Universidade Estadual  
de Campinas, para preen-  
chimento dos pré-requi-  
sitos parciais para obtenção  
do Título de Doutor em  
Engenharia Elétrica.

Este exemplar corresponde à redação final da tese  
defendida por Mario Conrado Cavichia  
Julgada em 11 de 07 de 1997 pela Comissão  
Francisco Lyra Filho  
Orientador

Julho 1997

UNIDADE	BC
N.º CHAMADA:	Unicamp
	C316r
V.	Et.
TIPO DE	31870
PROJ.	281/97
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	7811,00
DATA	18/10/97
N.º CPD	

CM-00102685-0

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

C316r Cavichia, Mario Conrado  
Resolução de problemas de programação linear por partes via algoritmos de pontos interiores / Mario Conrado Cavichia.--Campinas, SP: [s.n.], 1997.

Orientador: Marcos Nereu Arenales.  
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Programação não-linear. 2. Programação linear. 3. Otimização matemática. I. Arenales, Marcos Nereu. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

A minha esposa Lia....  
Aos meus pais Mario (In memoriam ) e Irene....

## **Agradecimentos**

Ao Prof. Marcos Nereu Arenales, pela paciência e dedicação constantes na orientação, durante todo o desenvolvimento deste trabalho.

Aos professores Francisco Antonio Menezes e Renato Soliani, pelas sugestões e incentivo.

À Faculdade de Engenharia Elétrica e de Computação, em particular ao Prof. Christiano Lyra Filho, pelo apoio recebido durante todo o programa de doutoramento.

Aos professores e funcionários da Faculdade de Engenharia Civil da Unicamp, pelos quais de modo direto ou não, senti-me apoiado.

A minha família , pelo estímulo e compreensão.

Ao Programa de Incentivo à Capacitação Docente da Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior.

## Resumo

Este trabalho tem como objetivo o desenvolvimento de algoritmos de pontos interiores, visando a resolução de problemas de minimização de funções objetivos lineares por partes, separáveis e convexas, sujeitas ainda a restrições lineares. Os métodos disponíveis na literatura para esta classe de problemas são do tipo simplex, com exceção de casos particulares.

Uma prática comum para a resolução de programas lineares por partes consiste em transformá-lo num programa linear e explorar suas propriedades. Mostramos que esta estratégia pode ser adequada para métodos do tipo simplex, porém inadequada para métodos de pontos interiores.

Neste trabalho abordamos o programa linear por partes diretamente como um problema de programação não linear. Para tanto apresentamos o que convencionamos chamar *algoritmo linear por partes interior*, isto é, um algoritmo que gera pontos interiores no problema original, embora a solução transformada esteja na fronteira. Mostramos que não se trata de uma simples extensão de um algoritmo de ponto interior aplicado ao programa linear transformado. Apresentamos também uma breve experiência computacional.

O algoritmo proposto é aplicado em alguns problemas, sendo alguns elaborados a partir de exemplos-teste retirados da NetLib.

Antes da apresentação do algoritmo linear por partes interior, fazemos uma revisão dos vários métodos de pontos interiores, apresentando-os sob um ponto de vista unificado, dando uma pequena contribuição quando da apresentação do algoritmo primal para problemas canalizados, sob tal ponto de vista. Em seguida, o método simplex linear por partes é apresentado para efeito de complementação de informações.

Como proposta de estudo futuro, introduzimos um problema que pode ser visto como de programação linear por partes: o problema conhecido como minmax.

No decorrer do trabalho, uma relação de outras aplicações que podem ser tratadas sob a ótica aqui abordada é apresentada.

## **Abstract**

The objective of this work is the development of an interior point algorithm for a piecewise linear programming problem (PLP). In contrast to the most papers which prefer to transform a PLP in a linear programming problem (LP) and then take advantage of a specific structure now created or considering the problem as an extension of the linear programming problem, using now a piecewise linear simplex algorithm.

The PLP will be considered as a problem of non-linear programming and in this context will be proposed an algorithm of interior point in order to solve it.

The proposed algorithm is applied to problems, with some of them constructed using examples from NetLib.

Before the main algorithm, a review of several interior point methods is presented, under an unified point of view. A review of this nature gives a small contribution when the primal algorithm for bounded linear problems is presented. The piecewise linear simplex method is then developed.

## **Conteúdo**

AGRADECIMENTOS	ii
RESUMO	iii
ABSTRACT	v
CONTEÚDO	vi
INTRODUÇÃO	1
<b>1 Uma Visão Geral dos Métodos de Pontos Interiores.</b>	<b>4</b>
1.1 Introdução.....	4
1.2 Preliminares.....	7
1.2.1 Dual de um Problema Canalizado.....	8
1.2.2 Condições de Otimalidade.....	10
1.2.2.1 Condições Necessárias de Primeira Ordem.....	10
1.2.2.2 Condições Suficientes de Segunda Ordem.....	11
1.2.3 Centro de Região de Factibilidade.....	11

1.3	Métodos de Pontos Interiores - Uma Visão Unificada.....	13
1.3.1	Métodos Projetivos.....	14
1.3.2	Métodos Afim - Escala.....	19
1.3.2.1	Primal.....	20
1.3.2.2	Primal Canalizado.....	24
1.3.2.3	Dual.....	27
1.3.2.4	Primal - Dual.....	30
1.3.3	Métodos de Trajetória Central.....	32
1.4	Sobre o Cálculo das Direções.....	35
1.5	Problema com Variáveis Canalizadas.....	36
<b>2</b>	<b>Solução de Sistemas Lineares Inconsistentes no Senso da Norma <math>L_1</math></b>	<b>49</b>
2.1	Introdução.....	49
2.2	Preliminares.....	51
2.1.1	Introdução.....	51
2.2.2	Método de Newton Composto de Nível-m Modificado e Perturbado.....	53
2.3	Métodos de Pontos Interiores.....	54
<b>3</b>	<b>Algoritmo Linear por Partes Via Método Simplex</b>	<b>66</b>
3.1	Introdução.....	66
3.2	Conversão para um Problema de Programação Linear.....	71
3.3	Método Simplex Linear por Partes.....	78
<b>4</b>	<b>Algoritmo Linear Por Partes Via Pontos Interiores</b>	<b>89</b>
4.1	Introdução.....	89
4.2	A Direção de Busca e um Critério de Parada.....	90
4.3	A Busca Unidimensional Linear por Partes.....	94

4.4	Algoritmo Linear por Partes Interior.....	95
4.4.1	Restrições com Igualdade.....	99
4.4.2	Restrições com Desigualdades.....	100
<b>5</b>	<b>O Funcionamento do Método Proposto no PL Equivalente</b>	<b>102</b>
<b>6</b>	<b>Experimentos Computacionais</b>	<b>106</b>
	<b>Conclusão e Perspectivas</b>	<b>129</b>
	<b>Bibliografia</b>	<b>135</b>

## INTRODUÇÃO

O trabalho de Karmarkar[26], apresentado em 1984, propondo um método de pontos interiores para programação linear, estimulou intensas pesquisas de novos métodos nesta área (veja Gonzaga[102], Hertog e Roos[27]). Os princípios dos métodos de pontos interiores foram mais tarde estendidos a problemas de otimização não lineares (veja Monteiro e Adler[103], Tapia[104]). Deve-se observar que as técnicas utilizadas para desenvolver métodos de pontos interiores em programação linear (veja Shanno e Bagchi[68]), eram bem difundidas na comunidade de pesquisadores em programação não-linear, tais como funções barreiras, resolução de sistemas não-lineares decorrentes das condições de primeira ordem, direção de Newton, etc..

Entre as classes de problemas de programação linear e não-linear, temos a classe dos *programas lineares por partes* (PLP), em particular, com a função objetivo linear por partes separável e convexa, sujeita a restrições lineares, que será objeto de estudo neste trabalho. Esta classe de programas lineares por partes foi bem estudada por Golstein e Youdine[87] (1966), sob a ótica do método simplex e, mais recentemente, a prestigiosa revista *Mathematical Programming* publicou artigos de Robert Fourer(veja [95], [105],[106]), sobre a extensão do método simplex para problemas de programação linear por partes, o que confirma a relevância ( e evidencia a originalidade ) do presente trabalho. Os artigos publicados na *Mathematical Programming* vêm confirmar a

importância de lidarmos com o problema diretamente na forma PLP, sem convertê-lo para um programa linear, como na maior parte dos trabalhos pesquisados.

A literatura sobre pontos interiores para programação linear por partes é bastante escassa, limitando-se a alguns estudos de problemas particulares como solução de sistemas inconsistentes nas normas  $L_1$  ou  $L_\infty$  (veja Zhang[23], Ruzinski[21]). Uma técnica usual para desenvolver algoritmos especializados em programas lineares por partes consiste basicamente em transformar o problema num programa linear equivalente (implicando num aumento do número de variáveis) para então aplicar um método da programação linear, explorando-se suas particularidades (veja Barrodale e Young[13], Zhang[23], ). Nos métodos do tipo simplex esta estratégia pode ser adequada, no sentido de que é possível obter desta forma um método equivalente de abordagem direta ao programa linear por partes, como será mostrado neste trabalho. Entretanto, a aplicação de métodos de pontos interiores ao programa linear equivalente não pode ser especializada, como nos métodos tipo simplex, para a obtenção de métodos de abordagem direta, uma vez que como veremos, estes produzem *métodos de pontos interiores híbridos* ao programa linear equivalente, isto é, algumas variáveis estarão no interior de seus intervalos de definição enquanto outras variáveis estarão em seus limites e, portanto, na fronteira.

A importância do estudo em programação linear por partes decorre da variedade de aplicações que são encontradas na literatura, mas principalmente nos problemas reais. Problemas de programação linear por partes aparecem no planejamento de produção,

expansão de redes telefônicas etc e, no campo teórico, no cálculo da solução inicial de problemas de programação linear(fase1), aproximações de problemas não lineares, etc.

Neste trabalho propomos um método de pontos interiores de abordagem direta ao programa linear por partes e um breve estudo de seu comportamento computacional.

No capítulo 1 apresentamos uma visão geral dos métodos de pontos interiores, estudando o dual de um problema canalizado, condições de otimalidade, centro de região de factibilidade e apresentamos alguns métodos de pontos interiores sob uma única ótica, ou seja, a combinação de uma direção de centralização com uma direção de descida. Na última sessão do capítulo 1 é apresentado um algoritmo de pontos interiores para a solução de um problema de programação linear canalizado.

No capítulo 2 apresentamos um estudo sobre a solução de sistemas lineares inconsistentes no senso da norma  $L_1$ , problema este que pode ser tratado como um caso particular de problema linear por partes, e como são, na literatura pesquisada, resolvidos por métodos de pontos interiores.

No capítulo 3 tratamos de um processo de conversão de um problema de programação linear por partes para um problema de programação linear, fazemos uma explanação do algoritmo simplex linear por partes.

Já no capítulo 4, propomos um algoritmo linear por partes de pontos interiores para resolver o problema. Apresentamos uma pequena experiência computacional sobre o comportamento do algoritmo, demonstramos algumas propriedades e ainda discutimos rapidamente o problema minmax sob a ótica do algoritmo linear por partes interior.

## CAPÍTULO 1

### Uma Visão Geral dos Métodos de Pontos Interiores

#### 1.1 Introdução

Consideremos o problema de programação linear na forma padrão:

$$\begin{aligned} \text{Min } & \mathbf{c}^t \mathbf{x} \\ \text{s.a. } & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{1}$$

onde  $\mathbf{A}$  é uma matriz  $m \times n$  de posto completo,  $m < n$ .

A região de factibilidade do problema (1) é denotada por  $\mathbf{F} = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$  e seu interior relativo  $\mathbf{F}^o = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{Ax} = \mathbf{b}, \mathbf{x} > \mathbf{0}\}$ .(2)

Sem perda de generalidade, assumiremos que  $\mathbf{F}$  é fechado e limitado e que  $\mathbf{F}^o$  é não vazio, desde que transformações padrão podem dar origem a tais restrições.

Desde a concepção do método *Simplex*, por Dantzig, em 1947, este, e seus variantes especializados, vem sendo preferencialmente utilizados para resolver (1).

Em 1984, N.K.Karmarkar apresentou um novo algoritmo, usando uma nova estratégia para resolver um problema de programação linear(PL). A fim de evitar a teórica complexidade exponencial dos métodos do tipo simplex, que propunham a resolução do problema de programação linear acima, buscando um ótimo através de soluções básicas adjacentes, Karmarkar resolveu o PL particular

$$\begin{aligned}
& \text{Min } \mathbf{c}^t \mathbf{x} \\
& \text{s.a. } \mathbf{A} \mathbf{x} = \mathbf{0} \\
& \quad \mathbf{e}^t \mathbf{x} = 1 \\
& \quad \mathbf{x} \geq \mathbf{0} ,
\end{aligned} \tag{3}$$

onde  $\mathbf{e}^t = (1, 1, \dots, 1)$ .

A peculiaridade da forma apresentada em nada diminui a relevância do fato, pois o próprio Karmarkar apresentou uma forma de colocar qualquer PL na forma acima.

Na solução de (3), Karmarkar exigindo ainda que o valor ótimo da função objetivo fosse nulo, ou seja,  $\mathbf{c}^t \mathbf{x}^* = 0$ , provou que seu método era de complexidade polinomial (vale destacar que ainda hoje o problema de encontrar um algoritmo tipo simplex, de complexidade polinomial, está em aberto).

O processo iterativo apresentado por Karmarkar é baseado na transformação projetiva definida por  $\hat{\mathbf{x}} = \frac{\mathbf{D}^{-1} \mathbf{x}}{\mathbf{e}^t \mathbf{D}^{-1} \mathbf{x}}$ , levando-nos a trabalhar num outro espaço, de modo que se  $\mathbf{D}$  é uma matriz diagonal formada pelas coordenadas da solução corrente estritamente positiva, esta é transformada no ponto  $\frac{\mathbf{e}}{n}$ . Daí um novo ponto é calculado por  $\hat{\mathbf{x}}' = \frac{\mathbf{e}}{n} - \alpha \mathbf{P} \mathbf{D} \mathbf{c}$ , onde  $\mathbf{P} = \mathbf{I} - \mathbf{B}^t (\mathbf{B} \mathbf{B}^t)^{-1} \mathbf{B}$  é a matriz de projeção sobre o núcleo de  $\mathbf{B}$ , com  $\mathbf{B} = \begin{pmatrix} \mathbf{A} \mathbf{D} \\ \mathbf{e}^t \end{pmatrix}$ . Isso produz no espaço original a nova solução  $\mathbf{x}' = \frac{\mathbf{D} \hat{\mathbf{x}}'}{\mathbf{e}^t \mathbf{D} \hat{\mathbf{x}}'}$ .

Uma limitação do método, eliminada logo depois por Todd e Burrell[100], era a exigência que o valor ótimo da função objetivo fosse zero, mas nada suplantaria a grande

novidade que foi a procura do ótimo caminhando através do interior da região de factibilidade.

Usando a função potencial  $f(\mathbf{x}) = n \log(\mathbf{c}^t \mathbf{x}) - \sum_{i=1}^n \log(x_i)$ , Karmarkar mostrou a convergência de seu método verificando a redução da mesma a cada iteração. Isto incentivaria o surgimento de um grande número de algoritmos aplicados na solução de PLs usando métodos de programação não-linear.

Após Karmarkar[26] apresentar seu algoritmo de pontos interiores para programação linear, muitos outros apareceram tentando explicar, melhorar, ou simplificar esse algoritmo, dando origem a muitos outros e a novas estratégias. Alguns autores vêm tentando classificar tais métodos segundo as diferentes estratégias utilizadas na obtenção de cada um. A classificação adotada aqui será a de Hertog e Roos[27], onde todos esses variantes podem ser colocados em quatro grupos principais:

i-) *Métodos projetivos*: o originalmente proposto por Karmarkar[26] e os estudados e ou, modificados por Anstreicher[28,29], Gay[30], De Ghellinck e Vial[31], Gonzaga[32], Yamashita[33], Ye e Kojima[35].

ii-) *Métodos afim escala* (primal, dual, primal-dual): originalmente proposto por Dikin[36] e depois investigados por Adler, Karmarkar, Resende e Veiga[37], Meggido e Shub[38], Barnes[39], Monna e Morton[40], Monteiro, Adler e Resende[41], Vanderbei, Meketon e Freedman[42].

iii-) *Métodos de trajetória central*: de passos curtos - Gill et al.[43], Gonzaga[44], Kojima, Mizuno e Yoshise[62], Monteiro e Adler[70], Renegar[47], Roos[48,49], Roos e Hertog[50], Roos e Vial[51], Todd e Ye[52], Vaidya[53]; de passos longos - Gonzaga[54], Den Hertog, Roos e Terlaky[55], Roos e Vial[56].

iv-) *Método de redução potencial afim*: Anstreicher e Bosch[57], Freund[58], Gonzaga[59], Kojima, Mizuno, Yoshise[60].

## 1.2 Preliminares

Consideremos o problema de programação linear primal e o seu problema dual dados abaixo

$$(P) \begin{cases} \text{Min} & \mathbf{c}^t \mathbf{x} \\ \text{s.a.} & \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0} \end{cases} \quad (D) \begin{cases} \text{Max} & \mathbf{b}^t \mathbf{y} \\ \text{s.a.} & \mathbf{A}^t \mathbf{y} \leq \mathbf{c}, \mathbf{y} \geq \mathbf{0} \end{cases}$$

Introduzindo as variáveis de folga primal e dual,  $\mathbf{s}$  e  $\mathbf{r}$ , onde

$$\mathbf{s} = \mathbf{Ax} - \mathbf{b} \geq \mathbf{0} \quad \text{e} \quad \mathbf{r} = \mathbf{c} - \mathbf{A}^t \mathbf{y} \geq \mathbf{0},$$

observemos que

$$\begin{aligned} 0 \leq \mathbf{r}^t \mathbf{x} + \mathbf{s}^t \mathbf{y} &= (\mathbf{c} - \mathbf{A}^t \mathbf{y})^t \mathbf{x} + (\mathbf{Ax} - \mathbf{b})^t \mathbf{y} \\ &= \mathbf{c}^t \mathbf{x} - \mathbf{y}^t \mathbf{Ax} + \mathbf{x}^t \mathbf{A}^t \mathbf{y} - \mathbf{b}^t \mathbf{y} \\ &= \mathbf{c}^t \mathbf{x} - \mathbf{b}^t \mathbf{y} \end{aligned}$$

Aqui  $\mathbf{r}^t\mathbf{x} + \mathbf{s}^t\mathbf{y}$  é o chamado *gap* de dualidade. O gap é nulo se  $\mathbf{r}^t\mathbf{x} = 0$  e  $\mathbf{s}^t\mathbf{y} = 0$ . Logo se  $\mathbf{x}$  for primal factível e  $\mathbf{y}$  dual factível para (P) e (D), respectivamente, então  $\mathbf{x}$  e  $\mathbf{y}$  serão as soluções ótimas se e só se forem verificadas as chamadas condições de folgas complementares  $\mathbf{r}^t\mathbf{x} = 0$  e  $\mathbf{s}^t\mathbf{y} = 0$ , isto é,  $r_j = 0$  ou  $x_j = 0 \quad \forall j$  e,  $s_j = 0$  ou  $y_j = 0 \quad \forall j$ .

Esses conceitos são importantes, pois em muitos algoritmos o critério de parada é a busca da primal ou dual factibilidade, mantendo-se a factibilidade dual ou primal e esperando-se que se cumpram as condições de folgas complementares para uma precisão pré-estabelecida, isto é, esperando-se que o gap seja nulo. Temos ainda outros algoritmos guiados pela busca do cumprimento das folgas complementares, enquanto a primal e a dual factibilidades são mantidas.

Quando (P) estiver na forma padrão (restrições de igualdade), as folgas complementares se resumem a  $\mathbf{r}^t\mathbf{x} = \mathbf{0}$ , visto que  $\mathbf{s} = \mathbf{0}$ .

### 1.2.1 Dual de um problema canalizado

Um problema de programação linear na forma

$$\begin{aligned} \text{Min } & \mathbf{c}^t\mathbf{x} \\ \text{s.a. } & \mathbf{Ax} = \mathbf{b} \\ & -\mathbf{e} \leq \mathbf{x} \leq \mathbf{e} \end{aligned}$$

é chamado canalizado. Esta não é a forma mais abrangente de um problema canalizado, mas servirá aos propósitos deste trabalho. Como o dual deste particular problema

canalizado será utilizado em algumas discussões adiante, faremos a seguir sua apresentação.

Tomando o lagrangiano do problema acima teremos

$$L(\mathbf{x}, \mathbf{y}) = \mathbf{c}^t \mathbf{x} + \mathbf{y}^t (\mathbf{b} - \mathbf{A}\mathbf{x}), \quad \text{com } -\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}.$$

Assim a função dual será

$$\tilde{h}(\mathbf{y}) = \underset{-\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}}{\text{Min}} L(\mathbf{x}, \mathbf{y}) = (\mathbf{c}^t - \mathbf{y}^t \mathbf{A})\mathbf{x} + \mathbf{y}^t \mathbf{b}.$$

Fazendo

$$\mathbf{z}^t = \mathbf{c}^t - \mathbf{y}^t \mathbf{A}$$

podemos dizer que a função dual é agora

$$h(\mathbf{y}, \mathbf{z}) = \begin{cases} \underset{-\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}}{\text{Min}} L(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{z}^t \mathbf{x} + \mathbf{y}^t \mathbf{b} \\ \mathbf{z} = \mathbf{c} - \mathbf{A}^t \mathbf{y} \end{cases}.$$

Assim o mínimo será atingido em  $x_i = -1$ , para  $z_i > 0$  e em  $x_i = 1$ , para  $z_i < 0$ ,

enquanto  $x_i \in [-1, 1]$  para  $z_i = 0$ .

Portanto o dual será

$$\begin{aligned} \text{Max } h(\mathbf{y}, \mathbf{z}) &= \mathbf{y}^t \mathbf{b} - \sum_{z_i > 0} z_i + \sum_{z_i < 0} z_i \\ \text{s.a. } &\mathbf{A}^t \mathbf{y} + \mathbf{z} = \mathbf{c}. \end{aligned}$$

Ou ainda

$$\begin{aligned} \text{Max } &\mathbf{b}^t \mathbf{y} - \sum_{i=1}^n |z_i| \\ \text{s.a. } &\mathbf{A}^t \mathbf{y} + \mathbf{z} = \mathbf{c}. \end{aligned}$$

Observamos que o dual obtido acima é um problema onde a função objetivo é linear por partes e que a dual factibilidade é alcançada quando

$$\mathbf{z} = \mathbf{c} - \mathbf{A}^t \mathbf{y}$$

e

$$z_i x_i \leq 0, \quad \text{para } i = 1, 2, \dots, n.$$

### 1.2.2 Condições de Otimalidade

Veremos abaixo condições de otimalidade necessárias e suficiente para um problema geral de otimização. Condições estas utilizadas em grande parte deste trabalho.

#### 1.2.2.1 Condições necessárias de primeira ordem

Vejamos agora um dos resultados que formam a base da programação não-linear, servindo em muitos casos para o desenvolvimento de novos algoritmos e estabelecimento de critérios de parada para tantos outros.

Consideremos  $\mathbf{E} \subset \mathfrak{R}^n$  um conjunto não vazio, as funções  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$ ,  $g_i: \mathfrak{R}^n \rightarrow \mathfrak{R}$ , com  $i = 1, \dots, r$ , e  $h_i: \mathfrak{R}^n \rightarrow \mathfrak{R}$ , com  $i = 1, \dots, s$ , diferenciáveis e o seguinte problema:

$$\begin{aligned} & \text{Min } f(\mathbf{x}) \\ & \text{s.a. } g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, r \\ & \quad h_i(\mathbf{x}) = 0, \quad i = 1, \dots, s \\ & \quad \mathbf{x} \in \mathbf{E} \end{aligned}$$

Tomemos agora uma solução factível  $\bar{\mathbf{x}}$  e  $I = \{g_i(\bar{\mathbf{x}}) = 0\}$  e suponha que  $\nabla g_i(\bar{\mathbf{x}})$ , para  $i \in I$ , e  $\nabla h_i(\bar{\mathbf{x}})$ , para  $i = 1, \dots, s$ , são linearmente independentes. Então se  $\bar{\mathbf{x}}$  resolve localmente o problema acima, teremos escalares  $u_i$ ,  $i \in I$  e  $v_i$ ,  $i = 1, \dots, s$ , com

$$\nabla f(\bar{\mathbf{x}}) + \sum_{i \in I} u_i \nabla g_i(\bar{\mathbf{x}}) + \sum_{i=1}^s v_i \nabla h_i(\bar{\mathbf{x}}) = \mathbf{0}$$

$$u_i \geq 0, \quad i \in I$$

ou

$$\nabla f(\bar{\mathbf{x}}) + \sum_{i=1}^r u_i \nabla g_i(\bar{\mathbf{x}}) + \sum_{i=1}^s v_i \nabla h_i(\bar{\mathbf{x}}) = \mathbf{0}$$

$$u_i g_i(\bar{\mathbf{x}}) = 0, \quad i = 1, \dots, r$$

$$u_i \geq 0, \quad i = 1, \dots, r$$

ou ainda na forma vetorial

$$\nabla f(\bar{\mathbf{x}}) + \nabla \mathbf{g}(\bar{\mathbf{x}}) \mathbf{u} + \nabla \mathbf{h}(\bar{\mathbf{x}}) \mathbf{v} = \mathbf{0}$$

$$\mathbf{u}^t \mathbf{g}(\bar{\mathbf{x}}) = 0$$

$$\mathbf{u} \geq \mathbf{0} .$$

### 1.2.2.2 Condições suficientes de primeira ordem

Assumindo hipóteses de convexidade sobre  $f$ ,  $g_i$  e  $h_i$ , funções lineares, as condições dadas acima são também suficientes para a otimalidade global.

### 1.2.3 Centro da região de factibilidade

Um conceito muito utilizado em otimização, via pontos interiores, é o de centragem, isto é, o esforço de nos mantermos o mais próximo possível do centro da

região de factibilidade. O centro da região de factibilidade F do problema (1) é aqui definido como a solução do seguinte problema:

$$\begin{aligned} \text{Min} \quad & - \sum_{i=1}^n \ln(x_i) \\ \text{s. a.} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} > 0. \end{aligned} \tag{4}$$

Para solução deste problema aplicamos as condições de primeira ordem vistas acima, o que implicam na resolução do sistema não-linear

$$\begin{aligned} -\mathbf{X}^{-1}\mathbf{e} - \mathbf{A}^t\mathbf{y} &= 0 \\ -\mathbf{Ax} + \mathbf{b} &= 0, \end{aligned} \tag{5}$$

onde  $\mathbf{X} = \text{diag}(x_i)$ .

Como (5) não tem solução analítica, um passo de Newton é aplicado e então obtemos

$$\mathbf{J}(\mathbf{x}, \mathbf{y}) \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{pmatrix} = -\mathbf{F}(\mathbf{x}, \mathbf{y}),$$

onde  $\mathbf{J}(\mathbf{x}, \mathbf{y})$  representa a matriz jacobiana de  $\mathbf{F}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} -\mathbf{X}^{-1}\mathbf{e} - \mathbf{A}^t\mathbf{y} \\ -\mathbf{Ax} + \mathbf{b} \end{pmatrix}$

Temos então que:

$$\begin{pmatrix} \mathbf{X}^{-2} & -\mathbf{A}^t \\ -\mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{X}^{-1}\mathbf{e} + \mathbf{A}^t\mathbf{y} \\ \mathbf{0} \end{pmatrix}.$$

Assim

$$\mathbf{X}^{-2}\Delta \mathbf{x} - \mathbf{A}^t\Delta \mathbf{y} = \mathbf{X}^{-1}\mathbf{e} + \mathbf{A}^t\mathbf{y} \tag{6}$$

$$-\mathbf{A}\Delta \mathbf{x} = \mathbf{0}. \tag{7}$$

De (6) vem que

$$\Delta \mathbf{x} = \mathbf{X}^2 \mathbf{A}^t \Delta \mathbf{y} + \mathbf{X} \mathbf{e} + \mathbf{X}^2 \mathbf{A}^t \mathbf{y} = \mathbf{X} \mathbf{e} + \mathbf{X}^2 \mathbf{A}^t (\mathbf{y} + \Delta \mathbf{y}). \quad (8)$$

Multiplicando agora (6) por  $\mathbf{A} \mathbf{X}^2$  e usando que  $-\mathbf{A} \Delta \mathbf{x} = \mathbf{0}$ , segue que

$$\begin{aligned} -\mathbf{A} \mathbf{X}^2 \mathbf{A}^t \Delta \mathbf{y} &= \mathbf{A} \mathbf{X} \mathbf{e} + \mathbf{A} \mathbf{X}^2 \mathbf{A}^t \mathbf{y} \\ \text{ou} \\ \mathbf{y} + \Delta \mathbf{y} &= -(\mathbf{A} \mathbf{X}^2 \mathbf{A}^t)^{-1} \mathbf{A} \mathbf{X} \mathbf{e}. \end{aligned}$$

De (8) tiramos então que

$$\Delta \mathbf{x} = \mathbf{X} \mathbf{e} - \mathbf{X}^2 \mathbf{A}^t (\mathbf{A} \mathbf{X}^2 \mathbf{A}^t)^{-1} \mathbf{A} \mathbf{X} \mathbf{e}$$

ou

$$\Delta \mathbf{x} = \mathbf{X} (\mathbf{I} - \mathbf{X} \mathbf{A}^t (\mathbf{A} \mathbf{X}^2 \mathbf{A}^t)^{-1} \mathbf{A} \mathbf{X}) \mathbf{e} = \mathbf{X} \mathbf{P} \mathbf{e}, \quad (9)$$

$\mathbf{P} = \mathbf{I} - \mathbf{X} \mathbf{A}^t (\mathbf{A} \mathbf{X}^2 \mathbf{A}^t)^{-1} \mathbf{A} \mathbf{X}$  sendo a matriz de projeção no núcleo de  $\mathbf{A} \mathbf{X}$ .

Vemos então que  $\Delta \mathbf{x} = \mathbf{X} \mathbf{P} \mathbf{e}$  nada mais é que a direção no qual devemos caminhar se quisermos atingir o centro da região de factibilidade.

### 1.3 Métodos de pontos interiores - Uma visão unificada.

Tomaremos por base a classificação mencionada acima dada por Hertog e Roos[27] e a estratégia, devido a Shanno e Bagchi[68], para unificar diferentes abordagens que deram origem a diferentes métodos de pontos interiores.

A unificação acima citada é no sentido que, na busca do ótimo, caminharemos em uma direção que será a combinação de um vetor de centralização e do vetor projeção do gradiente (vetor de otimização). Todos os métodos apresentados são vistos como variações do método de barreira logarítmica aplicados ao problema primal ou ao dual ou em ambos,

incluindo um algoritmo primal afim canalizado, como extensão ao trabalho de Shanno e Bagchi.

Abaixo apresentamos tal uniformização escolhendo alguns casos. Tendo isto em mente, vejamos:

### 1.3.1 Métodos projetivos

Sob a ótica anunciada, vamos considerar o método projetivo de Karmarkar

Gonzaga[102] foi o primeiro a mostrar a equivalência do método projetivo de Karmarkar com um método de barreira logarítmica. Aqui, entretanto, usaremos uma técnica devida a Gay[30] aplicada em (1), para transformá-lo na forma explorada por Karmarkar, que mostrará que a direção de busca será a composição de um vetor de centralização com um vetor de máxima descida. Para tanto definamos a transformação

$$\mathbf{x}' = (\mathbf{e}^t \mathbf{X}^{-1} \mathbf{x} + 1)^{-1} \begin{pmatrix} \mathbf{X}^{-1} \mathbf{x} \\ 1 \end{pmatrix} \quad (10)$$

e consideremos a matriz

$$(\mathbf{X} \ \mathbf{1}) = \begin{pmatrix} x_1 & 0 & 0 & \dots & 0 \\ 0 & x_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & x_n & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

Multiplicando (10) por  $(\mathbf{X} \ \mathbf{1})$ , vem

$$(\mathbf{X} \ \mathbf{1})\mathbf{x}' = (\mathbf{e}^t \mathbf{X}^{-1} \mathbf{x} + 1)^{-1} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

e denotando  $\mathbf{x}'_{1,2,\dots,n}$  o vetor formado pelas  $n$  primeiras componentes de  $\mathbf{x}'$ , temos:

$$\mathbf{X}\mathbf{x}'_{1,2,\dots,n} = (\mathbf{e}^t \mathbf{X}^{-1} \mathbf{x} + 1)^{-1} \mathbf{x}$$

e

$$x'_{n+1} = (\mathbf{e}^t \mathbf{X}^{-1} \mathbf{x} + 1)^{-1}.$$

Portanto

$$\mathbf{x} = x'_{n+1} \mathbf{X} \mathbf{x}'_{1,2,\dots,n}. \quad (11)$$

Com o que foi visto acima, (1) pode ser colocado na forma exigida em (3), a saber

$$\begin{pmatrix} \mathbf{A}\mathbf{X} & -\mathbf{b} \\ \mathbf{e}^t & 1 \end{pmatrix} \mathbf{x}' = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix}.$$

Como no espaço transformado  $\mathbf{c}^t \mathbf{x} = \frac{\mathbf{c}^t \mathbf{X} \mathbf{x}'_{1,2,\dots,n}}{x'_{n+1}}$ , temos que o gradiente da função objetivo

é

$$\frac{1}{x'_{n+1}} \begin{pmatrix} \mathbf{X}\mathbf{c} \\ -\frac{(\mathbf{X}\mathbf{c})^t \mathbf{x}'_{1,2,\dots,n}}{x'_{n+1}} \end{pmatrix} = \frac{\mathbf{1}}{x'_{n+1}} \begin{pmatrix} \mathbf{X}\mathbf{c} \\ -\mathbf{c}^t \mathbf{x} \end{pmatrix} = \frac{1}{x'_{n+1}} \begin{pmatrix} \mathbf{X}\mathbf{c} \\ \beta \end{pmatrix}, \text{ sendo } \beta \text{ então um limitante}$$

inferior para o valor ótimo da função objetivo.

Observemos que para a solução corrente  $\mathbf{x}$  temos  $\mathbf{X}^{-1} \mathbf{x} = \mathbf{e}$ , e então de (10) segue que

$$\mathbf{x}' = (\mathbf{e}^t \mathbf{e} + 1)^{-1} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

Assim  $\mathbf{x}' = \frac{1}{n+1} \mathbf{e}$  (o vetor  $\mathbf{e}$  tem aqui  $n+1$  componentes de 1's).

Então, no espaço transformado, uma nova solução  $\hat{\mathbf{x}}'$  é

$$\hat{\mathbf{x}}' = \frac{1}{n+1} \mathbf{e} - \alpha \hat{\mathbf{d}}, \quad (12)$$

onde

$$\hat{\mathbf{d}} = \mathbf{P}_B \begin{pmatrix} \mathbf{Xc} \\ \beta \end{pmatrix} = \mathbf{P}_B (\hat{\mathbf{X}}\mathbf{c})$$

é a projecção do gradiente no núcleo da matriz  $B$ . Assim

$$\mathbf{P}_B = \mathbf{I} - \mathbf{B}^t (\mathbf{B}\mathbf{B}^t)^{-1} \mathbf{B},$$

onde

$$\mathbf{B} = \begin{pmatrix} \mathbf{A}\mathbf{X} & -\mathbf{b} \\ \mathbf{e}^t & 1 \end{pmatrix}.$$

Com o intuito de calcularmos  $\hat{\mathbf{d}}$ , notemos que

$$\begin{aligned} \mathbf{B}\mathbf{B}^t &= \begin{pmatrix} \mathbf{A}\mathbf{X}^2\mathbf{A}^t + \mathbf{b}\mathbf{b}^t & \mathbf{A}\mathbf{X}\mathbf{e} - \mathbf{b} \\ \mathbf{e}^t\mathbf{X}^t\mathbf{A}^t - \mathbf{b}^t & n+1 \end{pmatrix} = \begin{pmatrix} \mathbf{A}\mathbf{X}^2\mathbf{A}^t + \mathbf{b}\mathbf{b}^t & \mathbf{A}\mathbf{x} - \mathbf{b} \\ (\mathbf{A}\mathbf{x} - \mathbf{b})^t & n+1 \end{pmatrix} = \\ &= \begin{pmatrix} \mathbf{A}\mathbf{X}^2\mathbf{A}^t + \mathbf{b}\mathbf{b}^t & \mathbf{0} \\ \mathbf{0} & n+1 \end{pmatrix}, \end{aligned}$$

pois  $\mathbf{A}\mathbf{X}\mathbf{e} = \mathbf{A}\mathbf{x} = \mathbf{b}$ , desde que  $\mathbf{x}$  é factível. Lembrando que se  $M$  é uma matriz inversível

então

$$(\mathbf{a}\mathbf{h}^t + M)^{-1} = M^{-1} - \frac{M^{-1}\mathbf{a}\mathbf{h}^t M^{-1}}{1 + \mathbf{h}^t M^{-1}\mathbf{a}}, \quad (13)$$

temos

$$\begin{aligned}
(\mathbf{B}\mathbf{B}^t)^{-1} &= \begin{pmatrix} (\mathbf{A}\mathbf{X}^2\mathbf{A}^t + \mathbf{b}\mathbf{b}^t)^{-1} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n+1} \end{pmatrix} = \\
&= \begin{pmatrix} (\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1} - \frac{(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}}{1 + \mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n+1} \end{pmatrix}
\end{aligned}$$

pois

$$\begin{aligned}
&(\mathbf{A}\mathbf{X}^2\mathbf{A}^t + \mathbf{b}\mathbf{b}^t) \left[ (\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1} - \frac{(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}}{1 + \mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}} \right] = \\
&= \mathbf{I} - \frac{\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}}{1 + \mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}} + \mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1} - \frac{\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}}{1 + \mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}} \\
&= \mathbf{I} - \frac{\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t) - \mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1} - \mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}}{1 + \mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}} + \\
&\quad + \frac{\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}}{1 + \mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}} \\
&= \mathbf{I} - \frac{\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}(-\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1} + \mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1})}{1 + \mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}} = \mathbf{I}
\end{aligned}$$

Assim

$$\begin{aligned}
\hat{\mathbf{d}} &= (\mathbf{I} - \mathbf{B}^t(\mathbf{B}\mathbf{B}^t)^{-1}\mathbf{B}) \begin{pmatrix} \mathbf{X}\mathbf{c} \\ \beta \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{X}\mathbf{c} \\ \beta \end{pmatrix} - \begin{pmatrix} \mathbf{X}^t\mathbf{A} & \mathbf{e} \\ -\mathbf{b}^t & 1 \end{pmatrix} \begin{pmatrix} (\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1} - \frac{(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{b}\mathbf{b}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}}{1 + \mathbf{e}^t\tilde{\mathbf{P}}\mathbf{e}} & \mathbf{0} \\ \mathbf{0} & \frac{1}{n+1} \end{pmatrix} \begin{pmatrix} \mathbf{A}\mathbf{X} & -\mathbf{b} \\ \mathbf{e}^t & 1 \end{pmatrix} \begin{pmatrix} \mathbf{X}\mathbf{c} \\ \beta \end{pmatrix}
\end{aligned}$$

$$= \begin{pmatrix} \mathbf{Xc} \\ \beta \end{pmatrix} - \begin{pmatrix} \mathbf{k}_{11} & \mathbf{k}_{12} \\ \mathbf{k}_{21} & \mathbf{k}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{Xc} \\ \beta \end{pmatrix},$$

onde

$$\mathbf{k}_{11} = \mathbf{XA}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1} - \frac{\mathbf{XA}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{bb}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{AX}}{1 + \mathbf{b}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{b}} + \frac{\mathbf{ee}^t}{n+1}$$

$$\mathbf{k}_{12} = -\mathbf{XA}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{b} + \frac{\mathbf{XA}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{bb}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{b}}{1 + \mathbf{b}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{b}} + \frac{\mathbf{e}}{n+1}$$

$$\mathbf{k}_{21} = -\mathbf{b}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{AX} + \frac{\mathbf{b}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{bb}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{AX}}{1 + \mathbf{b}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{b}} + \frac{\mathbf{e}^t}{n+1}$$

$$\mathbf{k}_{22} = \mathbf{b}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{b} - \frac{\mathbf{b}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{bb}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{b}}{1 + \mathbf{b}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{b}} + \frac{1}{n+1}.$$

Como  $\mathbf{b} = \mathbf{AXe}$ , desde que  $\mathbf{x}$  é factível, e  $\tilde{\mathbf{P}} = \mathbf{XA}^t(\mathbf{AX}^2\mathbf{A}^t)^{-1}\mathbf{AX}$ , concluímos que

$$\hat{\mathbf{d}} = \begin{pmatrix} \mathbf{PXc} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \beta + \mathbf{e}^t\tilde{\mathbf{P}}\mathbf{Xc} \\ 1 + \mathbf{e}^t\tilde{\mathbf{P}}\mathbf{e} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{P}}\mathbf{e} \\ 1 \end{pmatrix} - \begin{pmatrix} \mathbf{c}^t\mathbf{x} + \beta \\ n+1 \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix}, \quad (14)$$

onde  $\mathbf{P} = \mathbf{I} - \hat{\mathbf{P}}$ .

Vejamus que a direção obtida em (14) é uma combinação de um vetor de descida mais um vetor centralizador.

De (12) vem

$$\begin{aligned} \hat{\mathbf{x}}' &= \frac{1}{n+1}\mathbf{e} - \alpha \left[ \begin{pmatrix} \mathbf{PXc} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \beta + \mathbf{e}^t\tilde{\mathbf{P}}\mathbf{Xc} \\ 1 + \mathbf{e}^t\tilde{\mathbf{P}}\mathbf{e} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{P}}\mathbf{e} \\ 1 \end{pmatrix} - \begin{pmatrix} \mathbf{c}^t\mathbf{x} + \beta \\ n+1 \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} \right] \\ &= \frac{1}{n+1}\mathbf{e} + \alpha \begin{pmatrix} \mathbf{c}^t\mathbf{x} + \beta \\ n+1 \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} - \alpha \begin{pmatrix} \beta + \mathbf{e}^t\tilde{\mathbf{P}}\mathbf{Xc} \\ 1 + \mathbf{e}^t\tilde{\mathbf{P}}\mathbf{e} \end{pmatrix} \begin{pmatrix} (\mathbf{I} - \hat{\mathbf{P}})\mathbf{e} \\ 1 \end{pmatrix} - \alpha \begin{pmatrix} \mathbf{PXc} \\ \mathbf{0} \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{n+1} \mathbf{e} + \alpha \begin{pmatrix} \mathbf{c}^t \mathbf{x} + \beta \\ n+1 \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} - \alpha \begin{pmatrix} \beta + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{X} \mathbf{c} \\ 1 + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{e} \end{pmatrix} \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix} + \\
&\quad + \alpha \begin{pmatrix} \beta + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{X} \mathbf{c} \\ 1 + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{e} \end{pmatrix} \begin{pmatrix} \mathbf{P} \mathbf{e} \\ 0 \end{pmatrix} - \alpha \begin{pmatrix} \mathbf{P} \mathbf{X} \mathbf{c} \\ 0 \end{pmatrix} \\
&= \left\{ \frac{1}{n+1} \mathbf{e} + \alpha \left[ \begin{pmatrix} \mathbf{c}^t \mathbf{x} + \beta \\ n+1 \end{pmatrix} - \begin{pmatrix} \beta + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{X} \mathbf{c} \\ 1 + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{e} \end{pmatrix} \right] \right\} \mathbf{e} + \\
&\quad + \alpha \begin{pmatrix} \beta + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{X} \mathbf{c} \\ 1 + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{e} \end{pmatrix} \begin{pmatrix} \mathbf{P} \mathbf{e} \\ 0 \end{pmatrix} - \alpha \begin{pmatrix} \mathbf{P} \mathbf{X} \mathbf{c} \\ 0 \end{pmatrix}
\end{aligned}$$

Fazendo  $v = \frac{\mathbf{c}^t \mathbf{x} + \beta}{n+1}$  e  $\eta = \frac{\mathbf{e}^t \tilde{\mathbf{P}} \mathbf{X} \mathbf{c} + \beta}{1 + \mathbf{e}^t \tilde{\mathbf{P}} \mathbf{e}}$ , teremos

$$\hat{\mathbf{x}}' = \left[ \frac{1}{n+1} + \alpha(v - \eta) \right] \mathbf{e} + \alpha \eta \begin{pmatrix} \mathbf{P} \mathbf{e} \\ 0 \end{pmatrix} - \alpha \begin{pmatrix} \mathbf{P} \mathbf{X} \mathbf{c} \\ 0 \end{pmatrix}$$

Portanto a  $(n+1)$ -ésima componente do vetor acima é dada por

$$\hat{x}'_{n+1} = w = \frac{1}{n+1} + \alpha(v - \eta)$$

Assim, utilizando (11) vemos que

$$\begin{aligned}
\mathbf{x}' &= \mathbf{w}^{-1} \mathbf{X} \mathbf{w} \mathbf{e} + \frac{\alpha}{w} (\eta \mathbf{X} \mathbf{P} \mathbf{e} - \mathbf{X} \mathbf{P} \mathbf{X} \mathbf{c}) \\
&= \mathbf{x} + \frac{\alpha}{w} (\eta \mathbf{X} \mathbf{P} \mathbf{e} - \mathbf{X} \mathbf{P} \mathbf{X} \mathbf{c}),
\end{aligned}$$

o que confirma o proposto.

### 1.3.2 Métodos Afim-escala

Os algoritmos afim escala hoje estudados tiveram origem num trabalho apresentado por Dikin[36] em 1967, onde apareceu o chamado algoritmo afim escala básico. Em 1985,

tanto Barnes[39] como Vanderbei, Meketon, Freedman[42], em trabalhos independentes, propuseram, para resolver problemas de programação linear na forma padrão, algoritmos afim escala primal. Um algoritmo similar, o afim escala dual, foi apresentado e implementado por Adler, Karmarkar, Resende e Veiga[37] para problemas lineares com desigualdades. Já em 1987, Monteiro e Adler[70] e Kojima, Mizuno e Yoshise[62] apresentaram o então chamado algoritmo primal - dual afim escala.

Os algoritmos afim escala firmam-se sobre uma idéia básica, também explorada nos algoritmos projetivos: manter-se “longe” da fronteira, para que quando escolhida uma direção de movimento, possamos caminhar o máximo possível. A meta de nos mantermos longe da fronteira é atingida utilizando-se uma transformação afim. Daí originou parte do nome dado a esses algoritmos.

O que veremos abaixo são exemplos de algoritmos afins, nos três casos descritos acima, a saber, primal, dual ou primal-dual, onde a combinação de um vetor de otimização com um vetor centralizador aparece naturalmente.

### 1.3.2.1 Primal

Consideremos o problema de programação linear na forma padrão apresentado em (1), ou seja:

$$\begin{aligned} \text{Min } & \mathbf{c}^t \mathbf{x} \\ \text{s. a. } & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} . \end{aligned}$$

Para construção do algoritmo afim escala primal, na forma que foi originalmente deduzido, numa iteração  $k$ , seja  $\mathbf{x}^k$  uma solução factível. Consideremos então a transformação afim  $\mathbf{x}' = T(\mathbf{x}) = \mathbf{X}_k^{-1}\mathbf{x}$ . Observemos que  $T(\mathbf{x}^k) = \mathbf{e}$ , i.é, a transformação  $T$  muda o espaço original de trabalho de modo que a solução corrente seja "centralizada" e assim possamos trabalhar "longe" da fronteira ( $\mathbf{x} = \mathbf{0}$ ). Devemos então resolver o problema equivalente a (1)

$$\begin{aligned} \text{Min } & (\bar{\mathbf{c}})' \mathbf{x}' \\ \text{s. a. } & \bar{\mathbf{A}} \mathbf{x}' = \mathbf{b} \\ & \mathbf{x}' \geq \mathbf{0}, \end{aligned} \tag{15}$$

onde  $\bar{\mathbf{c}} = \mathbf{X}_k \mathbf{c}$  e  $\bar{\mathbf{A}} = \mathbf{A} \mathbf{X}_k$ .

Façamos então  $\bar{\mathbf{x}}' = \mathbf{e} + \alpha \mathbf{d}'$ , onde  $\alpha$  é escolhido de modo que  $\mathbf{e} + \alpha \mathbf{d}' > \mathbf{0}$ . Como devemos ter  $\bar{\mathbf{A}} \mathbf{d}' = \mathbf{0}$ , tomamos a direção de máxima descida projetada no núcleo de  $\bar{\mathbf{A}}$ ,  $\mathbf{d}' = -\mathbf{P} \bar{\mathbf{c}}$ , onde  $\mathbf{P}$  é a matriz de projeção

$$\mathbf{P} = \mathbf{I} - \bar{\mathbf{A}}' (\bar{\mathbf{A}} \bar{\mathbf{A}}')^{-1} \bar{\mathbf{A}}.$$

No espaço original teremos  $\mathbf{x}^{k+1} = \mathbf{X}_k \mathbf{x}'$  e a direção de descida é agora

$$\mathbf{d} = -\mathbf{X}_k \mathbf{P} \bar{\mathbf{c}} = -\mathbf{X}_k \mathbf{P} \mathbf{X}_k \mathbf{c}. \tag{16}$$

Sob o aspecto da unificação pretendida, consideremos o problema de barreira logarítmica

$$\begin{aligned} \text{Min } f(\mathbf{x}) &= \mathbf{c}' \mathbf{x} - \mu \sum_{i=1}^n \ln(x_i) \\ \text{s. a. } & \mathbf{A} \mathbf{x} = \mathbf{b}. \end{aligned} \tag{17}$$

Dado então uma solução interior  $\mathbf{x}$  factível, i.e.,  $\mathbf{Ax} = \mathbf{b}$  e  $\mathbf{x} > \mathbf{0}$ , pretendemos então encontrar uma nova estimativa interior factível para  $\mathbf{x}$ . Para tanto, aplicamos as condições de primeira ordem em (17), para obtermos

$$\begin{aligned} \mathbf{c} - \mu\mathbf{X}^{-1}\mathbf{e} - \mathbf{A}^t\mathbf{y} &= \mathbf{0} \\ -\mathbf{Ax} + \mathbf{b} &= \mathbf{0}, \end{aligned} \tag{18}$$

lembrando novamente que  $\mathbf{X} = \text{diag}(x_i)$  é uma matriz inversível e que  $\mathbf{e} = (1 \ 1 \ 1 \dots 1)^t$ .

O sistema não-linear (18) novamente não tem solução analítica. Com intenção de resolvê-lo, apliquemos uma iteração do método de Newton para obtermos:

$$\begin{pmatrix} \mu\mathbf{X}^{-2} & -\mathbf{A}^t \\ -\mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{pmatrix} = \begin{pmatrix} -\mathbf{c} + \mu\mathbf{X}^{-1}\mathbf{e} + \mathbf{A}^t\mathbf{y} \\ \mathbf{0} \end{pmatrix} \tag{19}$$

desde que

$$\mathbf{J}(\mathbf{x}, \mathbf{y}) \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{pmatrix} = -\mathbf{F}(\mathbf{x}, \mathbf{y}),$$

onde

$$\mathbf{J}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \mu\mathbf{X}^{-2} & -\mathbf{A}^t \\ -\mathbf{A} & \mathbf{0} \end{pmatrix} \text{ é a matriz jacobiana com } \mathbf{x} \text{ primal factível}$$

e

$$\mathbf{F}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \mathbf{c} - \mu\mathbf{X}^{-1}\mathbf{e} - \mathbf{A}^t\mathbf{y} \\ -\mathbf{Ax} + \mathbf{b} \end{pmatrix}.$$

De (19) vem que

$$\mu\mathbf{X}^{-2}\Delta\mathbf{x} - \mathbf{A}^t\Delta\mathbf{y} = -\mathbf{c} + \mu\mathbf{X}^{-1}\mathbf{e} + \mathbf{A}^t\mathbf{y} \tag{20}$$

$$-\mathbf{A}\Delta\mathbf{x} = \mathbf{0} \quad (\Delta\mathbf{x} \in \mathbf{N}(\mathbf{A})), \tag{21}$$

sendo que em (21),  $N(A)$  representa o núcleo de  $A$ .

Multiplicando (20) por  $X^2$  e isolando  $\Delta x$ , teremos

$$\begin{aligned}\Delta x &= (-X^2c + \mu Xe + X^2A^t y + X^2A^t \Delta y) \\ &= \left[-\frac{1}{\mu}X^2c + Xe + \frac{1}{\mu}X^2A^t(y + \Delta y)\right]\end{aligned}\quad (22)$$

Lembremos que  $AX^2A^t$  é inversível, desde que a matriz  $A$  tem as linhas linearmente independentes. Então multiplicando agora (20) por  $AX^2$  e usando (21) podemos escrever

$$\Delta y = (AX^2A^t)^{-1}AX^2c - \mu(AX^2A^t)^{-1}AXe - y$$

ou

$y' = y + \Delta y = (AX^2A^t)^{-1}(AX^2c - \mu AXe)$ , que substituído em (22) produz

$$\begin{aligned}\Delta x &= -\frac{1}{\mu}X^2c + Xe + \frac{1}{\mu}X^2A^t[(AX^2A^t)^{-1}(AX^2c - \mu AXe)] \\ &= -\frac{1}{\mu}X^2c + Xe + \frac{1}{\mu}X^2A^t(AX^2A^t)^{-1}AX^2c - X^2A^t(AX^2A^t)^{-1}AXe \\ &= -\frac{1}{\mu}(X^2c - X^2A^t(AX^2A^t)^{-1}AX^2c) + Xe - X^2A^t(AX^2A^t)^{-1}AXe \\ &= -\frac{1}{\mu}X(Xc - XA^t(AX^2A^t)^{-1}AX^2c) + X(e - XA^t(AX^2A^t)^{-1}AXe) \\ &= -\frac{1}{\mu}X[I - XA^t(AX^2A^t)^{-1}AX]Xc + X[I - XA^t(AX^2A^t)^{-1}AX]e\end{aligned}$$

Portanto

$$\Delta x = -\frac{1}{\mu}XPXc + XPe, \quad (23)$$

onde

$\mathbf{P} = \mathbf{I} - \mathbf{X}\mathbf{A}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1}\mathbf{A}\mathbf{X}$  é a matriz de projeção sobre o núcleo da matriz  $\mathbf{A}\mathbf{X}$ .

Vemos que a direção dada por (23) é uma combinação linear dos vetores  $-\mathbf{X}\mathbf{P}\mathbf{X}\mathbf{c}$  e  $\mathbf{X}\mathbf{P}\mathbf{e}$ , que são respectivamente a direção usada num algoritmo afim primal (ver, por exemplo, [101], cap.7) e a direção de busca do centro da região de factibilidade.

Quando  $\mu$  se aproxima de zero ( $\mu \rightarrow 0$ ) o que se obtém em (23) é, preponderantemente,  $\Delta\mathbf{x} = -\mathbf{X}\mathbf{P}\mathbf{X}\mathbf{c}$ , que é exatamente a direção de descida encontrada no algoritmo afim escala primal e ainda que  $\Delta\mathbf{x} = -\mathbf{X}\mathbf{P}\mathbf{X}\mathbf{c}$  é a projeção do vetor de máxima descida, visto agora no espaço original.

### 1.3.2.2 Primal Canalizado

Estendemos aqui o trabalho de Shanno e Bagchi para um algoritmo afim-primal para um problema canalizado.

Lembremos que em 1.5, pg36, voltaremos a tratar com mais detalhes o problema canalizado, apresentando e analisando a convergência de um algoritmo de ponto interiores para tais problemas, que será útil na dedução do algoritmo linear por partes proposto.

Consideremos então o problema primal canalizado, ou seja, um problema na forma

$$\text{Min } f(\tilde{\mathbf{x}}) = \tilde{\mathbf{c}}^t \tilde{\mathbf{x}}$$

$$\text{s.a. } \tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$$

$$\mathbf{u} \leq \tilde{\mathbf{x}} \leq \mathbf{v},$$

Uma mudança de variável, que será vista mais a frente, nos conduzirá ao estudo do problema

$$\text{Min } f(\mathbf{x}) = \mathbf{c}^t \mathbf{x},$$

$$\text{s.a. } \mathbf{Ax} = \mathbf{b}$$

$$-\mathbf{e} \leq \mathbf{x} \leq \mathbf{e},$$

Então aplicando as condições de primeira ordem, obtemos

$$\mathbf{c} - \mathbf{A}^t \lambda + \mu \mathbf{XD}^{-1} = \mathbf{0}$$

$$-\mathbf{Ax} + \mathbf{b} = \mathbf{0}$$

No sistema não-linear acima,  $\mathbf{D} = \text{diag}(1 - x_i^2)$  e  $\mathbf{X} = \text{diag}(2x_i)$ .

Nossa estratégia de resolução deste sistema é aplicar novamente um passo de Newton e então usar a direção obtida como direção de descida. Tal estratégia nos leva a:

$$\begin{pmatrix} \mu(2\mathbf{D}^{-1} + \mathbf{X}^2\mathbf{D}^{-2}) & -\mathbf{A}^t \\ -\mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_\lambda \end{pmatrix} = \begin{pmatrix} -\mathbf{c} + \mathbf{A}^t \lambda - \mu \mathbf{XD}^{-1} \mathbf{e} \\ \mathbf{0} \end{pmatrix}$$

Fazendo  $\bar{\mathbf{D}}^{-1} = 2\mathbf{D}^{-1} + \mathbf{X}^2\mathbf{D}^{-2}$ , segue que

$$\mu \bar{\mathbf{D}}^{-1} \mathbf{d}_x - \mathbf{A}^t \mathbf{d}_\lambda = -\mathbf{c} + \mathbf{A}^t \lambda - \mu \mathbf{XD}^{-1} \mathbf{e} \quad (24)$$

$$-\mathbf{Ad}_x = \mathbf{0}$$

Multiplicando (24) por  $\bar{\mathbf{A}}\mathbf{D}$ , vem

$$\mu \bar{\mathbf{A}} \mathbf{d}_x - \bar{\mathbf{A}} \mathbf{D} \mathbf{A}^t \mathbf{d}_\lambda = -\bar{\mathbf{A}} \mathbf{D} \mathbf{c} + \bar{\mathbf{A}} \mathbf{D} \mathbf{A}^t \lambda - \mu \bar{\mathbf{A}} \mathbf{D} \mathbf{X} \mathbf{D}^{-1} \mathbf{e}.$$

Como sabemos que  $-\mathbf{A}\mathbf{d}_x = \mathbf{0}$ , vem

$$\mathbf{d}_\lambda + \lambda = (\mathbf{A}\bar{\mathbf{D}}\mathbf{A}^t)^{-1}\mathbf{A}\bar{\mathbf{D}}\mathbf{c} + \mu(\mathbf{A}\bar{\mathbf{D}}\mathbf{A}^t)^{-1}\mathbf{A}\bar{\mathbf{D}}\mathbf{X}\mathbf{D}^{-1}\mathbf{e} \quad (25)$$

Multiplicando agora (24) por  $\bar{\mathbf{D}}$ , teremos

$$\mu\mathbf{d}_x = -\bar{\mathbf{D}}\mathbf{c} - \mu\bar{\mathbf{D}}\mathbf{X}\mathbf{D}^{-1}\mathbf{e} + \bar{\mathbf{D}}\mathbf{A}^t(\lambda + \mathbf{d}_\lambda) \quad (26)$$

Substituindo (25) em (26), vem

$$\mathbf{d}_x = -\frac{1}{\mu} \left[ \bar{\mathbf{D}}\mathbf{c} - \bar{\mathbf{D}}\mathbf{A}^t(\mathbf{A}\bar{\mathbf{D}}\mathbf{A}^t)^{-1}\mathbf{A}\bar{\mathbf{D}}\mathbf{c} \right] - \bar{\mathbf{D}}\mathbf{X}\mathbf{D}^{-1}\mathbf{e} + \bar{\mathbf{D}}\mathbf{A}^t(\mathbf{A}\bar{\mathbf{D}}\mathbf{A}^t)^{-1}\mathbf{A}\bar{\mathbf{D}}\mathbf{X}\mathbf{D}^{-1}\mathbf{e}$$

$$\begin{aligned} \mathbf{d}_x = & -\frac{1}{\mu} \left[ \bar{\mathbf{D}}^{1/2}(\mathbf{I} - \bar{\mathbf{D}}^{1/2}\mathbf{A}^t(\mathbf{A}\bar{\mathbf{D}}\mathbf{A}^t)^{-1}\mathbf{A}\bar{\mathbf{D}}^{1/2}) \bar{\mathbf{D}}^{1/2}\mathbf{c} - \right. \\ & \left. - \bar{\mathbf{D}}^{1/2} \left[ \mathbf{I} - \bar{\mathbf{D}}^{1/2}\mathbf{A}^t(\mathbf{A}\bar{\mathbf{D}}\mathbf{A}^t)^{-1}\mathbf{A}\bar{\mathbf{D}}^{1/2} \right] \bar{\mathbf{D}}^{1/2}\mathbf{X}\mathbf{D}^{-1}\mathbf{e} \right] \end{aligned}$$

ou

$$\mathbf{d}_x = -\frac{1}{\mu} \bar{\mathbf{D}}^{1/2}\mathbf{P}\bar{\mathbf{D}}^{1/2}\mathbf{c} - \bar{\mathbf{D}}^{1/2}\mathbf{P}\bar{\mathbf{D}}^{1/2}\mathbf{X}\mathbf{D}^{-1}\mathbf{e}, \quad (27)$$

com  $\mathbf{P} = \mathbf{I} - \bar{\mathbf{D}}^{1/2}\mathbf{A}^t(\mathbf{A}\bar{\mathbf{D}}\mathbf{A}^t)^{-1}\mathbf{A}\bar{\mathbf{D}}^{1/2}$ , a matriz de projeção sobre o núcleo de  $\mathbf{A}\bar{\mathbf{D}}^{1/2}$ .

Podemos observar que  $-\bar{\mathbf{D}}^{1/2}\mathbf{P}\bar{\mathbf{D}}^{1/2}\mathbf{X}\mathbf{D}^{-1}\mathbf{e}$ , que aparece em (27), representa o vetor que aponta na direção do centro da região de factibilidade  $\{\mathbf{A}\mathbf{x} = \mathbf{b}, -\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}\}$ .

Senão, vejamos: Achar o centro da região de factibilidade  $\{\mathbf{A}\mathbf{x} = \mathbf{b}, -\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}\}$ ,

consiste em se resolver o seguinte problema:

$$\begin{aligned} & \text{Min}(\sum \ln(1 - x_i) + \sum \ln(1 + x_i)) \\ & \text{s.a. } \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned}$$

Aplicando as condições de primeira ordem, teremos

$$\begin{aligned} \mathbf{XD}^{-1} + \mathbf{A}^t \lambda &= \mathbf{0} \\ -\mathbf{Ax} + \mathbf{b} &= \mathbf{0} . \end{aligned}$$

Aqui, novamente, tomamos  $\mathbf{X}$  e  $\mathbf{D}$  como definidas acima.

Agora ,um passo de Newton nos leva a:

$$\begin{pmatrix} 2\mathbf{D}^{-1} + \mathbf{X}^2\mathbf{D}^{-2} & \mathbf{A}^t \\ -\mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{d}_x \\ \mathbf{d}_\lambda \end{pmatrix} = \begin{pmatrix} -\mathbf{XD}^{-1} - \mathbf{A}^t \lambda \\ \mathbf{0} \end{pmatrix}$$

Trabalhando no sistema acima encontramos

$$\mathbf{d}_x = -\bar{\mathbf{D}}^{1/2} \mathbf{P} \bar{\mathbf{D}}^{1/2} \mathbf{XD}^{-1} \mathbf{e} ,$$

a segunda parcela do vetor encontrado em (27).

### 1.3.2.3 Dual

Recordemos que o dual de (1) é dado por

$$\begin{aligned} \text{Max } \mathbf{b}^t \mathbf{y} \\ \text{s.a. } \mathbf{A}^t \mathbf{y} + \mathbf{z} &= \mathbf{c} \\ \mathbf{z} &\geq \mathbf{0} \end{aligned} \tag{28}$$

Primeiramente vejamos o vetor de movimento obtido por Adler, Karmarkar, Resende e Veiga[37], na apresentação do algoritmo dual afim escala.

Podemos fazer uso das mesmas idéias empregadas na dedução do vetor (16) no caso primal, mas lembrando que o escalamento é feito nas variáveis de folga  $\mathbf{z}$ .

Tomemos então, em uma iteração  $k$ , uma solução dual interior  $(\mathbf{y}^k, \mathbf{z}^k)$ , i.e.,

$$\mathbf{A}^t \mathbf{y}^k + \mathbf{z}^k = \mathbf{c} \quad \text{e} \quad \mathbf{z}^k > \mathbf{0} . \text{ Iremos determinar um vetor de movimento } \mathbf{d} = \begin{bmatrix} \mathbf{d}_y \\ \mathbf{d}_z \end{bmatrix} , \text{ de}$$

maneira que a nova solução obtida  $(\mathbf{y}^{k+1}, \mathbf{z}^{k+1})$ , com

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \varepsilon_k \mathbf{d}_y^k$$

e

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \varepsilon_k \mathbf{d}_z^k,$$

satisfaça

$$\begin{aligned} \mathbf{A}^t \mathbf{y}^{k+1} + \mathbf{z}^{k+1} &= \mathbf{c} \\ \mathbf{z}^{k+1} &> \mathbf{0} \\ \mathbf{b}^t \mathbf{y}^{k+1} &\geq \mathbf{b}^t \mathbf{y}^k. \end{aligned} \tag{30}$$

Usando (29) e (30) concluímos que

$$\begin{aligned} \mathbf{A}^t \mathbf{d}_y^k + \mathbf{d}_z^k &= \mathbf{0} \\ \mathbf{b}^t \mathbf{d}_y^k &\geq \mathbf{0}. \end{aligned}$$

Semelhante ao caso primal, procedemos a um escalamento a fim de levar  $\mathbf{z}^k$  (do espaço original) em  $\mathbf{e} = (1, 1, \dots, 1)^t \in \mathfrak{R}^n$  (no espaço escalado). Para tanto fazemos

$$\mathbf{s} = \mathbf{Z}_k^{-1} \mathbf{z}.$$

Temos que se  $\mathbf{d}_s^k$  é uma direção de movimento adequada no espaço escalado, então no espaço original a direção correspondente será

$$\mathbf{d}_z^k = \mathbf{Z}_k \mathbf{d}_s^k. \tag{31}$$

Como  $\mathbf{A}^t \mathbf{d}_y^k + \mathbf{d}_z^k = \mathbf{0}$ , segue que

$$\mathbf{A}^t \mathbf{d}_y^k + \mathbf{Z}_k \mathbf{d}_s^k = \mathbf{0},$$

$$\mathbf{Z}_k^{-1} \mathbf{A}^t \mathbf{d}_y^k + \mathbf{d}_s^k = \mathbf{0},$$

revelando que a direção de movimento no espaço original é determinada pela direção no espaço escalado. Então se conseguirmos encontrar  $\mathbf{d}_s^k$  de modo que  $\mathbf{b}^t \mathbf{d}_y^k \geq 0$ , teremos atingido nosso intuito. Tomemos

$$\mathbf{d}_s^k = -\left(\mathbf{AZ}_k^{-2} \mathbf{A}^t\right)^{-1} \mathbf{AZ}_k^{-1} \mathbf{b} \text{ e}$$

fazendo  $\mathbf{Q}_k = \left(\mathbf{AZ}_k^{-2} \mathbf{A}^t\right)^{-1} \mathbf{AZ}_k^{-1}$ ,

teremos  $\mathbf{b}^t \mathbf{d}_y^k = -\mathbf{b}^t \mathbf{Q}_k \mathbf{d}_s^k = \mathbf{b}^t \mathbf{Q}_k \mathbf{Q}_k^t \mathbf{b} = \|\mathbf{b}^t \mathbf{Q}_k\|^2 \geq 0$ .

Assim

$$\mathbf{d}_y^k = \left(\mathbf{AZ}_k^{-2} \mathbf{A}^t\right)^{-1} \mathbf{b}. \quad (32)$$

E como  $\mathbf{d}_z^k = -\mathbf{A}^t \mathbf{d}_y^k$

$$\mathbf{d}_z^k = -\mathbf{A}^t \left(\mathbf{AZ}_k^{-2} \mathbf{A}^t\right)^{-1} \mathbf{b}, \quad (33)$$

e a direção  $\mathbf{d} = \begin{bmatrix} \mathbf{d}_y \\ \mathbf{d}_z \end{bmatrix}$  fica completamente determinada.

Retornando à intenção primeira deste capítulo, voltemos nossa atenção para o problema (28), considerando o correspondente penalizado, ou seja

$$\begin{aligned} \text{Max } & \mathbf{b}^t \mathbf{y} - \mu \sum_{i=1}^n \ln(\mathbf{c}_i - \mathbf{a}_i^t \mathbf{y}) \\ \text{s.a. } & \mathbf{A}^t \mathbf{y} + \mathbf{z} = \mathbf{c}, \end{aligned}$$

onde  $\mathbf{a}_i^t$  é a  $i$ -ésima coluna da matriz  $\mathbf{A}$ .

As condições de 1ª ordem aplicadas fornecem

$$\mathbf{b} - \mu \mathbf{AZ}^{-1} \mathbf{e} = \mathbf{0}$$

Notando que a matriz jacobiana de  $\mathbf{F}(\mathbf{y}) = \mathbf{b} - \mu \mathbf{AZ}^{-1} \mathbf{e}$  é

$$\mathbf{J}(\mathbf{y}) = -\mu \mathbf{AZ}^{-2} \mathbf{A}^t,$$

um passo de Newton na solução de  $\mathbf{F}(\mathbf{y}) = \mathbf{0}$  leva-nos a

$$-\mu \mathbf{AZ}^{-2} \mathbf{A}^t \Delta \mathbf{y} = -\mathbf{b} + \mu \mathbf{AZ}^{-1} \mathbf{e}$$

ou

$$\mathbf{AZ}^{-2} \mathbf{A}^t \Delta \mathbf{y} = \frac{\mathbf{b}}{\mu} - \mathbf{AZ}^{-1} \mathbf{e}.$$

Assim temos

$$\Delta \mathbf{y} = \frac{1}{\mu} (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} \mathbf{b} - (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} \mathbf{AZ}^{-1} \mathbf{e}. \quad (34)$$

Novamente, analisando (34), notamos que temos a combinação linear dos vetores de máxima subida (32) e o vetor centralizador obtido pela aplicação da técnica apresentada na seção 1.2.3, na região de factibilidade do problema (28).

#### 1.3.2.4 Primal-Dual

Alguns métodos de pontos interiores dão passos simultaneamente nos espaços das variáveis primais e duais. São os chamados métodos primais-duais. Eles têm suas raízes em Megiddo[61] e uma análise e desenvolvimento destes métodos podem vistos, mais detalhadamente, por exemplo em [62],[63],[64],[65].

Veremos um pouco dos métodos primais-duais em seguida.

Aqui o método é derivado da aplicação das condições de primeira ordem ao problema dual (28), utilizando ainda o fato que os multiplicadores de Lagrange associados ao problema (28) são as variáveis primais.

Das condições de primeira ordem, segue que

$$\begin{aligned} \mathbf{b} - \mathbf{A}\mathbf{x} &= \mathbf{0} & \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mu\mathbf{Z}^{-1}\mathbf{e} - \mathbf{x} &= \mathbf{0} & \text{ou} & \mu\mathbf{e} = \mathbf{Z}\mathbf{x} = \mathbf{Z}\mathbf{X}\mathbf{e} \\ \mathbf{A}^t\mathbf{y} + \mathbf{z} - \mathbf{c} &= \mathbf{0} & \mathbf{A}^t\mathbf{y} + \mathbf{z} - \mathbf{c} &= \mathbf{0} \end{aligned} \quad (35)$$

com  $\mathbf{x} \geq \mathbf{0}$  e  $\mathbf{z} \geq \mathbf{0}$ .

Fazendo

$$\mathbf{F}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \begin{pmatrix} \mathbf{A}\mathbf{x} - \mathbf{b} \\ \mathbf{A}^t\mathbf{y} + \mathbf{z} - \mathbf{c} \\ \mathbf{Z}\mathbf{X}\mathbf{e} - \mu\mathbf{e} \end{pmatrix} = \mathbf{0} \quad (36)$$

e aplicando um passo de Newton obtemos

$$\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^t & \mathbf{I} \\ \mathbf{Z} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{X}\mathbf{Z}\mathbf{e} + \mu\mathbf{e} \end{pmatrix}. \quad (37)$$

De (37) podemos tirar que

$$\mathbf{Z}\Delta\mathbf{x} + \mathbf{X}\Delta\mathbf{z} = -\mathbf{X}\mathbf{Z}\mathbf{e} + \mu\mathbf{e} \quad (38)$$

$$\mathbf{A}\Delta\mathbf{x} = \mathbf{0} \quad (39)$$

$$\mathbf{A}^t\Delta\mathbf{y} + \Delta\mathbf{z} = \mathbf{0}. \quad (40)$$

Multiplicando (38) por  $\mathbf{A}\mathbf{Z}^{-1}$  e utilizando (39) e (40), segue que

$$\Delta\mathbf{y} = (\mathbf{A}\mathbf{Z}^{-1}\mathbf{X}\mathbf{A}^t)^{-1}\mathbf{A}\mathbf{Z}^{-1}(\mathbf{X}\mathbf{Z}\mathbf{e} - \mu\mathbf{e}). \quad (41)$$

Agora

$$AZ^{-1}(XZe - \mu e) = AZ^{-1}XZe - \mu AZ^{-1}e = AZ^{-1}ZXe - \mu AZ^{-1}e = \mathbf{b} - \mu AZ^{-1}e$$

Portanto de (41)

$$\Delta \mathbf{y} = (AZ^{-1}XA^t)^{-1} \mathbf{b} - \mu (AZ^{-1}XA^t)^{-1} AZ^{-1}e \quad (42)$$

Novamente temos em (42) um vetor de busca da mesma forma que os obtidos anteriormente, ou seja, uma combinação linear de um vetor de máxima subida com um de centralização.

### 1.3.3 Métodos de Trajetória Central

O algoritmo apresentado por Renegar[47] baseia-se no cálculo de uma seqüência de centros de regiões, obtidas da região de factibilidade através de cortes, e ainda incorporando a função objetivo como parte do conjunto de restrições. Renegar se inspirou num trabalho de Huard [83], que resolvia um problema de programação matemática com restrições não lineares via método dos centros. Renegar partiu do seguinte problema:

$$\mathbf{max} \mathbf{f}(\mathbf{x}, \beta) = l \cdot \ln(\mathbf{b}^t \mathbf{y} - \beta) + \sum_{i=1}^n \ln(c_i - \mathbf{a}_i^t \mathbf{y}), \quad (43)$$

onde  $\beta$  é uma estimativa para o valor ótimo da função objetivo e  $l$  um parâmetro livre, embora, como Renegar comentou,  $l = n + 1$  seria uma boa escolha.

O algoritmo, seguindo a citada estratégia de cortes na região de factibilidade, com o cálculo dos respectivos centros, produz uma seqüência convergente para o ótimo procurado.

As maiores diferenças entre o algoritmo ora proposto e o método de barreiras são a eliminação do parâmetro  $\mu$  e a inclusão da subestimativa para o valor ótimo  $\mathbf{c}^t \mathbf{x}^*$ .

Vejamos o método de Renegar sob o ponto de vista da unificação.

As condições de primeira ordem aplicadas em (43) produzem

$$\frac{\mathbf{l}}{\mathbf{b}^t \mathbf{y} - \beta} \mathbf{b} - \mathbf{AZ}^{-1} \mathbf{e} = \mathbf{0}$$

cuja matriz das derivadas segundas é

$$\nabla^2 \mathbf{f} = -\mathbf{l}(\mathbf{b}^t \mathbf{y} - \beta)^{-2} \mathbf{b} \mathbf{b}^t - \mathbf{AZ}^{-2} \mathbf{A}^t$$

e assim

$$-\nabla^2 \mathbf{f} = \mathbf{l}(\mathbf{b}^t \mathbf{y} - \beta)^{-2} \mathbf{b} \mathbf{b}^t + \mathbf{AZ}^{-2} \mathbf{A}^t. \quad (44)$$

Tomando  $\mathbf{a} = \mathbf{l}(\mathbf{b}^t \mathbf{y} - \beta)^{-2} \mathbf{b}$ ,  $\mathbf{h}^t = \mathbf{b}^t$  e  $\mathbf{M} = \mathbf{AZ}^{-2} \mathbf{A}^t$  e usando (13), teremos

$$\begin{aligned} (-\nabla^2 \mathbf{f})^{-1} &= (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} - \frac{(\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} \mathbf{l}(\mathbf{b}^t \mathbf{y} - \beta)^{-2} \mathbf{b} \mathbf{b}^t (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1}}{1 + \mathbf{b}^t (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} \mathbf{l}(\mathbf{b}^t \mathbf{y} - \beta)^{-2} \mathbf{b}} \\ &= (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} - \frac{(\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} \mathbf{b} \mathbf{b}^t (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1}}{\frac{(\mathbf{b}^t \mathbf{y} - \beta)^2}{\mathbf{l}} + \mathbf{b}^t (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} \mathbf{b}} \end{aligned}$$

Aplicando um passo de Newton, vem:

$$\Delta \mathbf{y} = -(\nabla^2 \mathbf{f})^{-1} \nabla \mathbf{f} = (-\nabla^2 \mathbf{f}^2)^{-1} \nabla \mathbf{f}$$

$$= \left( (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} - \frac{(\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} \mathbf{b} \mathbf{b}^t (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1}}{\frac{(\mathbf{b}^t \mathbf{y} - \beta)^2}{\mathbf{l}} + \mathbf{b}^t (\mathbf{AZ}^{-2} \mathbf{A}^t)^{-1} \mathbf{b}} \right) \left( \frac{\mathbf{l} \mathbf{b}}{\mathbf{b}^t \mathbf{y} - \beta} - \mathbf{AZ}^{-1} \mathbf{e} \right)$$

$$\begin{aligned}
&= -(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{AZ}^{-1}\mathbf{e} + \frac{(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{lb} \frac{(\mathbf{b}^t\mathbf{y} - \beta)^2}{\mathbf{l}} + (\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{lb}\mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b}}{(\mathbf{b}^t\mathbf{y} - \beta) \left( \frac{(\mathbf{b}^t\mathbf{y} - \beta)^2}{\mathbf{l}} + \mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b} \right)} \\
&\quad - \frac{(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{bb}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}}{(\mathbf{b}^t\mathbf{y} - \beta) \left( \frac{(\mathbf{b}^t\mathbf{y} - \beta)^2}{\mathbf{l}} + \mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b} \right)} + \frac{(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{bb}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{AZ}^{-1}\mathbf{e}}{\frac{(\mathbf{b}^t\mathbf{y} - \beta)^2}{\mathbf{l}} + \mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b}} \\
&= -(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{AZ}^{-1}\mathbf{e} + \frac{(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b}(\mathbf{b}^t\mathbf{y} - \beta) + (\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{bb}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{AZ}^{-1}\mathbf{e}}{\frac{(\mathbf{b}^t\mathbf{y} - \beta)^2}{\mathbf{l}} + \mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b}} \\
&= -(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{AZ}^{-1}\mathbf{e} + \frac{(\mathbf{b}^t\mathbf{y} - \beta + \mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{AZ}^{-1}\mathbf{e})(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b}}{\frac{(\mathbf{b}^t\mathbf{y} - \beta)^2}{\mathbf{l}} + \mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b}}.
\end{aligned}$$

Assim

$$\Delta\mathbf{y} = -(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{AZ}^{-1}\mathbf{e} + \gamma(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b}, \quad (45)$$

onde

$$\gamma = \frac{(\mathbf{b}^t\mathbf{y} - \beta + \mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{AZ}^{-1}\mathbf{e})}{\frac{(\mathbf{b}^t\mathbf{y} - \beta)^2}{\mathbf{l}} + \mathbf{b}^t(\mathbf{AZ}^{-2}\mathbf{A}^t)^{-1}\mathbf{b}}.$$

Analisando (45) percebemos que a direção de busca é uma combinação linear de um vetor de máxima descida com um vetor centralizador.

## 1.4 Observações Sobre os Cálculos das Direções

Em todos os métodos analisados, vimos que a direção de movimento tem como componentes principais um vetor de otimização e um vetor centralizador. Mais ainda, podemos observar que o trabalho computacional relevante, no cálculo de uma dada direção, é a resolução de sistemas lineares  $\mathbf{B}\mathbf{x} = \gamma$ , onde a matriz B é definida positiva, ou a esta reduzida e do tipo:

$$\mathbf{A}\Theta^2\mathbf{A}^t,$$

com  $\Theta$  matriz diagonal, onde:

$$\Theta^2 = \mathbf{X}^2, \text{ no caso primal,}$$

$$\Theta^2 = \mathbf{Z}^{-2}, \text{ no caso dual,}$$

$$\Theta^2 = \mathbf{Z}^{-1}\mathbf{X}, \text{ para algoritmos primais-duais,}$$

As diferentes estratégias usadas na obtenção do sistema  $\mathbf{B}\mathbf{x} = \gamma$  é que diferenciam muitos dos algoritmos conhecidos. Consideremos, por exemplo,

$$\begin{pmatrix} \mathbf{Z} & \mathbf{0} & \mathbf{X} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^t & \mathbf{I} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{z} \end{pmatrix} = \begin{pmatrix} -\mathbf{X}\mathbf{Z}\mathbf{e} + \mu\mathbf{e} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix},$$

dado em (37).

Observemos que, através de cálculos algébricos, podemos reduzir (37) para

$$\begin{bmatrix} -\mathbf{S}\mathbf{X}^{-1} & \mathbf{A}^t \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{w} \end{bmatrix} = \begin{bmatrix} \kappa \\ \upsilon \end{bmatrix} \quad (46)$$

ou

$$(\mathbf{AXS}^{-1}\mathbf{A}^t)[\Delta\mathbf{w}] = \zeta \quad (47)$$

Alguns autores, dentre os quais destacamos Vanderbei and Carpenter[66], Fourer and Mehrotra[67], apesar de não trabalharem diretamente com o sistema indefinido (37), não chegam ao ponto de utilizarem (47), a fim de evitarem o preenchimento("fill-in"), que acontece principalmente quando a matriz  $\mathbf{A}$  tem uma ou mais colunas densas. Uma vantagem que aparece quando se utiliza (47) é que, sendo a matriz  $\mathbf{AXZ}^{-1}\mathbf{A}^t$  simétrica positiva definida, a ordem dos pivôs pode ser escolhida de antemão sem nos preocuparmos com problemas numéricos.

### 1.5- Problema com Variáveis Canalizadas

Daremos uma atenção especial a um trabalho apresentado por Sherali, Skarpness e Kim[20], que trata da solução do já citado problema canalizado, sob o ponto de vista de pontos interiores. Tal problema desempenhará um papel importante na obtenção do algoritmo linear por partes a ser proposto no capítulo 4.

Consideremos assim o problema canalizado

$$\begin{aligned} \text{Min } & \tilde{\mathbf{c}}^t \tilde{\mathbf{x}} \\ & \tilde{\mathbf{A}} \tilde{\mathbf{x}} = \tilde{\mathbf{b}} \quad , \\ & \mathbf{l} \leq \tilde{\mathbf{x}} \leq \mathbf{u} \end{aligned}$$

com  $\tilde{\mathbf{A}}$  matriz  $m \times n$ , tendo posto máximo  $m$ .

Efetuada a mudança de variáveis

$$x_i = (2\tilde{x}_i - (l_i + u_i)) / (u_i - l_i), \quad i = 1, 2, \dots, n,$$

podemos trabalhar com o problema canalizado

$$\begin{aligned} \text{Min } \mathbf{c}^t \mathbf{x} \\ \mathbf{Ax} = \mathbf{b} \\ -\mathbf{e} \leq \mathbf{x} \leq \mathbf{e}, \end{aligned} \tag{48}$$

sendo  $\mathbf{e}$ , como anteriormente, o vetor formado por 1's.

Vamos utilizar a mesma técnica empregada na obtenção do vetor de descida para o algoritmo primal afim escala, deduzido no capítulo 1, seção 1.3.2.1, mas tomando agora matriz de escalamento:

$$\begin{aligned} \mathbf{X} &= \text{diag}(X_1, X_2, \dots, X_n), \\ \text{com} \\ X_i &= \min\{1 + x_i, 1 - x_i\}, \quad i = 1, 2, \dots, n. \end{aligned} \tag{49}$$

Então as condições de primeira ordem aplicadas no problema penalizado

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= \text{Min } \mathbf{c}^t \mathbf{x} - \mu \sum_{i=1}^n [(\log(1 + x_i) + \log(1 - x_i))] \\ &= \text{Min } \mathbf{c}^t \mathbf{x} - \mu \sum_{i=1}^n [(\log(1 - x_i^2))] \end{aligned} \tag{50}$$

$$\text{sujeito a } \mathbf{Ax} = \mathbf{b} \tag{51}$$

nos conduzem ao sistema

$$\mathbf{c} - \mathbf{A}^t \mathbf{y} - \mu \mathbf{X}^{-1} \mathbf{v}(\mathbf{x}) = \mathbf{0} \tag{52}$$

e

$$\mathbf{Ax} - \mathbf{b} = \mathbf{0}, \tag{53}$$

onde o vetor  $\mathbf{v}$  é tal que

$$v_i(x_i) = \frac{-2X_i x_i}{(1-x_i^2)}, \quad i = 1, 2, \dots, n. \quad (54)$$

Para a obtenção de (52) foi usado o fato que a  $i$ -ésima coordenada do vetor gradiente da

função  $f(\mathbf{x})$ , dada em (50), é  $\mathbf{c}_i - \mathbf{a}_i^t \mathbf{y} - \mu \left( \frac{-2x_i}{1-x_i^2} \right)$ ,

que pode ser reescrito como

$$\mathbf{c}_i - \mathbf{a}_i^t \mathbf{y} - \mu X_i^{-1} \left( \frac{-2X_i x_i}{1-x_i^2} \right),$$

com  $X_i$  definida em (49).

Agora, de (52) vem

$$\mathbf{A}^t \mathbf{y} = \mathbf{c} - \mu \mathbf{X}^{-1} \mathbf{v}(\mathbf{x}),$$

cujas soluções por quadrados mínimos é

$$\mathbf{y} = (\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1} \mathbf{A}\mathbf{X}(\mathbf{X}\mathbf{c} - \mu\mathbf{v}(\mathbf{x})). \quad (55)$$

De (49) segue que o gradiente de (50) é

$$\mathbf{g} = \mathbf{c} - \mathbf{A}^t \mathbf{y} - \mu \mathbf{X}^{-1} \mathbf{v}(\mathbf{x}). \quad (56)$$

Assim calculando o gradiente projetado

$$\mathbf{p} = \mathbf{X}\mathbf{g} = [\mathbf{I} - \mathbf{X}\mathbf{A}^t(\mathbf{A}\mathbf{X}^2\mathbf{A}^t)^{-1} \mathbf{A}\mathbf{X}(\mathbf{X}\mathbf{c} - \mu\mathbf{v}(\mathbf{x}))], \quad (57)$$

podemos tomar como direção de descida

$$\mathbf{d} = -\mathbf{X}\mathbf{p}. \quad (58)$$

Considere então o seguinte algoritmo para resolver (48):

### Algoritmo 1

Considerando que na iteração  $k = 0$  tenhamos disponível uma solução factível inicial, com  $-\mathbf{e} \leq \mathbf{x}_0 \leq \mathbf{e}$ , tomando  $q \in [0.97, 0.99]$  e  $\varepsilon > 0$ , pequeno, façamos:

Passo 1- Calculemos

$$\mathbf{y}_k = (\mathbf{A}\mathbf{X}_k^2\mathbf{A}^t)^{-1}\mathbf{A}\mathbf{X}_k(\mathbf{X}_k\mathbf{c} - \mu\mathbf{v}(\mathbf{x}_k)),$$

$$\mathbf{g}_k = \mathbf{c} - \mathbf{A}^t\mathbf{y}_k - \mu\mathbf{X}_k^{-1}\mathbf{v}(\mathbf{x}_k),$$

$$\mathbf{p}_k = \mathbf{X}_k\mathbf{g}_k = [\mathbf{I} - \mathbf{X}_k\mathbf{A}^t(\mathbf{A}\mathbf{X}_k^2\mathbf{A}^t)^{-1}\mathbf{A}\mathbf{X}_k(\mathbf{X}_k\mathbf{c} - \mu\mathbf{v}(\mathbf{x}_k))],$$

$$\mathbf{d}_k = -\mathbf{X}_k\mathbf{p}_k.$$

Passo 2- Critério de parada

$$\|\mathbf{p}_k\| \leq \varepsilon.$$

Passo 3- Cálculo da nova iteração

Façamos

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k\mathbf{d}_k,$$

com passo

$$\lambda_k = \text{Min}\left\{q\lambda_{\max}, \frac{(1-q^2)^2}{4m}\right\}, \quad (59)$$

onde

$$\frac{1}{\lambda_{\max}} = \max_{i=1,\dots,n} \left[ \max\left\{\frac{d_{ki}}{1-x_{ki}}, \frac{-d_{ki}}{1+x_{ki}}\right\} \right] > 0. \quad (60)$$

Incrementamos e voltamos ao passo 1.

### Algumas observações sobre o algoritmo acima

i-) Uma escolha adequada de  $\mu$  permite fixá-lo, sem que haja a preocupação de irmos diminuindo-o a cada iteração.

ii-) O valor de  $y$  calculado em (55) pode ser considerado como uma estimativa dual, pois o que temos é uma aproximação da solução de  $A^t y = c - \mu X^{-1} v(x)$ . Fazendo então

$$z = c - A^t y = \mu X^{-1} v(x),$$

notamos que  $z_i x_i = \frac{-2\mu x_i^2}{1 - x_i^2} \leq 0$ . Assim, a dual factibilidade, vista no cap.I, item 1.2,

será satisfeita se  $y$  resolver (52) exatamente.

iii-) O segundo termo em (59) ficará claro no contexto da prova da convergência.

iv-) De (49) e (54) vemos que

$$v_i(x_{ki}) = \frac{X_{ki}}{1 + x_{ki}} - \frac{X_{ki}}{1 - x_{ki}} = \begin{cases} 1 - \frac{1 + x_{ki}}{1 - x_{ki}}, & \text{para } x_{ki} \leq 0 \\ \frac{1 - x_{ki}}{1 + x_{ki}} - 1, & \text{para } x_{ki} \geq 0 \end{cases} \quad (61)$$

De (61) decorre que

$$-1 < v_i(x_{ki}) < 1,$$

para todo  $i$ .

Assim  $\mu v(x_k)$  provoca uma perturbação muito pequena em  $X_k c$ , quando  $\mu$  é tomado suficientemente próximo de zero. Podemos então dizer que o algoritmo ora proposto é uma modificação do algoritmo afim primal.

Com intuito de mostrarmos que o algoritmo1 produz uma seqüência que converge para um par primal dual ótimo, consideremos:

**Lema 1:** Seja

$$h_i(\lambda) = \frac{X_{ki}}{1 - (x_{ki} + \lambda d_{ki})^2}, i = 1, 2, \dots, n,$$

com  $0 \leq \lambda \leq q\lambda_{\max}$  e  $q \in [0.97, 0.99]$ . Segue então que

$$h_i(\lambda) \leq \frac{1}{1 - q^2}. \quad (62)$$

Prova:

Observemos que para  $\lambda = 0$  temos

$$\begin{aligned} h_i(0) &= \frac{X_{ki}}{1 - x_{ki}^2} = \frac{X_{ki}}{(1 - x_{ki})(1 + x_{ki})} \\ &= \text{Min} \left\{ \frac{1}{1 - x_{ki}}, \frac{1}{1 + x_{ki}} \right\} \leq 1 \leq \frac{1}{1 - q^2} \end{aligned}$$

Tomemos então  $0 < \lambda \leq q\lambda_{\max}$  e consideremos dois casos.

**1º caso-** Seja  $d_{ki} \leq 0$ .

De (59) e (60)

$$\frac{1}{\lambda} \geq \frac{1}{q\lambda_{\max}} \geq \frac{-d_{ki}}{q(1 + x_{ki})}.$$

Assim

$$q + qx_{ki} \geq -\lambda d_{ki}$$

ou equivalentemente

$$q + qx_{ki} \geq -\lambda d_{ki} + x_{ki} - x_{ki},$$

ou ainda

$$x_{ki} + \lambda d_{ki} \geq x_{ki}(1 - q) - q. \quad (63)$$

Se  $x_{ki} + \lambda d_{ki} \geq 0$ , como  $d_{ki} \leq 0$ , devemos ter  $x_{ki} \geq 0$  e assim (49) implica que

$$X_{ki} = 1 - x_{ki}.$$

Logo

$$h_i(\lambda) = \frac{1 - x_{ki}}{1 - (x_{ki} + \lambda d_{ki})^2} \leq \frac{1 - x_{ki}}{1 - x_{ki}^2} = \frac{1}{1 + x_{ki}} \leq 1 \leq \frac{1}{1 - q^2}.$$

Supondo agora que  $x_{ki} + \lambda d_{ki} < 0$ , de (63) segue que

$$(x_{ki} + \lambda d_{ki})^2 \leq (x_{ki}(1 - q) - q)^2 < 1$$

e então

$$h_i(\lambda) \leq \frac{X_{ki}}{1 - (x_{ki}(1 - q) - q)^2} = g_i(x_{ki}). \quad (64)$$

Se  $0 \leq x_{ki} < 1$  então  $X_{ki} = 1 - x_{ki}$ . Portanto

$$g_i(x_{ki}) = \frac{1 - x_{ki}}{1 - (x_{ki}(1 - q) - q)^2}$$

e

$$\begin{aligned} g'_i(x_{ki}) &= \frac{-\{1 - [x_{ki}(1 - q) - q]^2\} + 2(1 - x_{ki})[x_{ki}(1 - q) - q](1 - q)}{\{1 - [x_{ki}(1 - q) - q]^2\}^2} \\ &= \frac{-1 - x_{ki}^2 - q^2 x_{ki}^2 + 3q^2 + 2x_{ki} - 4qx_{ki} - 2q + 2qx_{ki}^2 + 2q^2 x_{ki}}{\{1 - [x_{ki}(1 - q) - q]^2\}^2} \end{aligned}$$

$$\begin{aligned}
&= \frac{-1 - x_{ki}^2 - q^2 x_{ki}^2 + 3q^2 + x_{ki} + x_{ki} - 2qx_{ki} - 2qx_{ki} - 2q - q + q + 2qx_{ki}^2 + q^2 x_{ki} + q^2 x_{ki}}{\{1 - [x_{ki}(1 - q) - q]\}^2} \\
&= \frac{-1 - 3q + q + 3q^2 + x_{ki} - 2qx_{ki} + q^2 x_{ki} - x_{ki}^2 + 2qx_{ki}^2 - q^2 x_{ki}^2 + x_{ki} - 2qx_{ki} + q^2 x_{ki}}{\{1 - [x_{ki}(1 - q) - q]\}^2} \\
&= \frac{-(1 - q) - 3q(1 - q) + x_{ki}(1 - 2q + q^2) - x_{ki}^2(1 - 2q + q^2) + x_{ki}(1 - 2q + q^2)}{\{1 - [x_{ki}(1 - q) - q]\}^2} \\
&= \frac{-(1 - q)(1 + 3q) + (1 - 2q + q^2)(x_{ki} - x_{ki}^2 + x_{ki})}{\{1 - [x_{ki}(1 - q) - q]\}^2} \\
&= \frac{(1 - q)^2}{\{1 - [x_{ki}(1 - q) - q]\}^2} \left[ x_{ki}(2 - x_{ki}) - \frac{(1 + 3q)}{(1 - q)} \right].
\end{aligned}$$

Como  $x_{ki}(2 - x_{ki}) \leq 1$  e  $\frac{1 + 3q}{1 - q} > 1$ , temos que  $g'_i(x_{ki}) < 0$ , isto é,  $g_i$  é decrescente no

intervalo  $[0, 1)$  e portanto

$$g_i(x_{ki}) \leq g_i(0) = \frac{1}{1 - q^2},$$

e assim (62) é verificada.

Por outro lado devemos tomar agora  $-1 < x_{ki} \leq 0$ . Podendo ver de (49) que

$$X_{ki} = 1 + x_{ki}$$

e agora de (60) segue que

$$g_i(x_{ki}) = \frac{1 + x_{ki}}{1 - (x_{ki}(1 - q) - q)^2}$$

com

$$g'_i(x_{ki}) = \frac{(1-q)^2}{\{1 - [x_{ki}(1-q) - q]\}^2} [1 + x_{ki}(2 + x_{kji})].$$

Como  $x_{ki}(2 + x_{ki}) > -1$  temos  $g'_i(x_{ki}) > 0$ , e então

$$g_i(x_{ki}) \leq g_i(0) = \frac{1}{1-q^2},$$

concluindo assim o 1º caso.

**2º caso:**  $d_{ki} \geq 0$ .

A prova é similar à anterior.

Podemos então agora enunciar e provar o seguinte:

**Teorema 1-** O algoritmo1 dá origem a uma seqüência que converge finitamente para um par ótimo primal dual  $(\mathbf{x}_k, \mathbf{y}_k)$  satisfazendo , (52) e (53). Ou então uma seqüência é gerada satisfazendo:

a-)  $\{f(\mathbf{x}_k)\}$  definida em (50) é estritamente decrescente.

b-)  $\{\|\mathbf{p}_k\|\} \rightarrow 0$ ,  $\mathbf{p}_k$  dado em (57).

c-) Para qualquer subsequência convergente  $\{\mathbf{x}_k\} \rightarrow \mathbf{x}^*$ ,  $k \in K$ , com  $\mathbf{x}^*$  ótimo, temos  $-\mathbf{e} < \mathbf{x}^* < \mathbf{e}$ .

Além disso  $\{\mathbf{y}_k\}$ ,  $k \in K$ , converge para algum  $\mathbf{y}^*$ , com  $(\mathbf{x}^*, \mathbf{y}^*)$  satisfazendo (52), (53) e assim gerando um par primal dual ótimo.

**Prova:**

Se  $\mathbf{p}_k = \mathbf{0}$  em alguma interação  $k$ , com  $-\mathbf{e} < \mathbf{x}_k < \mathbf{e}$ , então usando (55)-(57) podemos dizer que  $\mathbf{g}_k = \mathbf{0}$ , e portanto  $(\mathbf{x}_k, \mathbf{y}_k)$  satisfaz as condições de primeira ordem (52),(53).

Vamos supor então que tenhamos gerado uma seqüência infinita  $\{\mathbf{x}_k\}$ . Então

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k + \lambda_k \mathbf{d}_k) = \mathbf{c}^t \mathbf{x}_k + \lambda_k \mathbf{c}^t \mathbf{d}_k - \mu \sum_{i=1}^n \log[1 - (x_{ki} + \lambda_k d_{ki})^2]. \quad (65)$$

De (56)-(58) segue que

$$\begin{aligned} \mathbf{c}^t \mathbf{d}_k &= \mathbf{g}_k^t \mathbf{d}_k + \mathbf{y}_k^t \mathbf{A} \mathbf{d}_k + \mu \mathbf{v}(\mathbf{x}_k)^t \mathbf{X}_k^{-1} \mathbf{d}_k = -\mathbf{g}_k^t \mathbf{X}_k^2 \mathbf{g}_k - \mu \mathbf{v}(\mathbf{x}_k)^t \mathbf{p}_k = \\ &= -\|\mathbf{p}_k\|^2 - \mu \mathbf{v}(\mathbf{x}_k)^t \mathbf{p}_k. \end{aligned} \quad (66)$$

Substituindo (66) em (65), teremos:

$$f(\mathbf{x}_{k+1}) = \mathbf{c}^t \mathbf{x}_k - \lambda_k \|\mathbf{p}_k\|^2 - \mu \lambda_k \mathbf{v}(\mathbf{x}_k)^t \mathbf{p}_k - \mu \sum_{i=1}^n \log[1 - (x_{ki} + \lambda_k d_{ki})^2]. \quad (67)$$

Tomemos agora

$$\phi(\lambda) = \sum_{i=1}^n \log[1 - (x_{ki} + \lambda d_{ki})^2] \text{ para } 0 \leq \lambda \leq q\lambda_{\max}. \quad (68)$$

Pelo Teorema de Taylor, vem que

$$\phi(\lambda) = \phi(0) + \lambda \phi'(0) + \frac{\lambda^2}{2} \phi''(\bar{\lambda}), \text{ onde } 0 \leq \bar{\lambda} \leq \lambda \leq \lambda_{\max}. \quad (69)$$

Usando (54), (58) e (68) concluímos que

$$\phi(0) = \sum_{i=1}^n \log(1 - x_{ki}^2)$$

$$\phi'(0) = \sum_{i=1}^n \frac{-2x_{ki}d_{ki}}{1-x_{ki}^2} = \sum_{i=1}^n \frac{2X_{ki}x_{ki}p_{ki}}{1-x_{ki}^2} = -v(x_k)^t p_k. \quad (70)$$

Ainda por (58) e (69), vem

$$\phi''(\bar{\lambda}) = \sum_{i=1}^n -2d_{ki}^2 \frac{[1+(x_{ki}+\bar{\lambda}d_{ki})^2]}{[1-(x_{ki}+\bar{\lambda}d_{ki})^2]^2} = \sum_{i=1}^n \frac{-2p_{ki}^2 X_{ki}^2 [1+(x_{ki}+\bar{\lambda}d_{ki})^2]}{[1-(x_{ki}+\bar{\lambda}d_{ki})^2]^2}.$$

Do lema-1 e do fato que  $[1+(x_{ki}+\bar{\lambda}d_{ki})^2] \leq 2$ , obtemos

$$\phi''(\bar{\lambda}) \geq \sum_{i=1}^n \frac{-4p_{ki}^2}{(1-q^2)^2} = \frac{-4}{(1-q^2)^2} \|p_k\|^2.$$

Temos então de (68) que

$$\phi(\lambda) \geq \sum_{i=1}^n \log(1-x_{ki}^2) - \lambda_k v(x_k)^t p_k - \frac{2\lambda^2}{(1-q^2)^2} \|p_k\|^2.$$

Assim

$$f(x_{k+1}) \leq c^t x_k - \lambda_k \|p_k\|^2 - \mu \lambda_k v(x_k)^t p_k - \mu \sum_{i=1}^n \log(1-x_{ki}^2) + \mu \lambda v(x_k)^t p_k + \frac{2\mu \lambda_k^2}{(1-q^2)^2} \|p_k\|^2$$

e

$$f(x_{k+1}) - f(x_k) \leq -\lambda_k \|p_k\|^2 \left[ 1 - \frac{2\mu \lambda_k}{(1-q^2)^2} \right] \leq -\frac{\lambda_k}{2} \|p_k\|^2 < 0,$$

o que mostra a parte a) do teorema.

Agora

$$\frac{1}{\lambda_{\max}} \leq \max_{i=1, \dots, n} \left\{ \frac{|d_{ki}|}{D_{ki}} \right\} = \|p_k\|_{\infty} \leq \|p_k\|.$$

Então de (59) podemos escrever

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{\lambda_k}{2} \|\mathbf{p}_k\|^2 \geq \begin{cases} \frac{q}{2} \|\mathbf{p}_k\|^2 & \text{se } \lambda_k = q\lambda_{\max} \\ \frac{(1-q^2)^2}{8\mu} \|\mathbf{p}_k\|^2 & \text{se } \lambda_k = \frac{(1-q^2)^2}{4\mu} \end{cases} \quad (71)$$

Como  $\{f(\mathbf{x}_k)\}$  é decrescente e inferiormente limitada, (71) implica que  $\{\|\mathbf{p}_k\|\} \rightarrow 0$ .

Com isso verificamos b-). Como assumimos que  $\{\mathbf{x}_k\}$  está contida num conjunto

compacto, deve existir uma subsequência  $\{\mathbf{x}_k\}$ ,  $k \in K$ , convergente para  $\mathbf{x}^*$ . Agora (50)

acarreta que  $-\mathbf{e} < \mathbf{x}^* < \mathbf{e}$ , pois caso contrário  $\{f(\mathbf{x}_k)\} \rightarrow +\infty$ ,  $k \in K$ . Novamente usando

(50), podemos dizer que  $\{\mathbf{v}(\mathbf{x}_k)\} \rightarrow \mathbf{v}(\mathbf{x}^*)$ ,  $k \in K$ , e que está bem definida e, assim por

(55), teremos que o limite  $\mathbf{y}^*$ , de  $\{\mathbf{y}_k\}$ ,  $k \in K$ , existe. Além disso, como  $\{\|\mathbf{p}_k\|\} \rightarrow 0$  e

$-\mathbf{e} < \mathbf{x}^* < \mathbf{e}$ , segue que  $\{\mathbf{g}_k\} \rightarrow 0$ ,  $k \in K$ . Portanto (56) implica na existência de um par

$(\mathbf{x}^*, \mathbf{y}^*)$  primal dual ótimo para o problema, completando assim a prova do teorema 1.  $\square$

**Corolário 1:** Seja  $\varepsilon$  a precisão escolhida no critério de parada e tomemos  $\mu < \frac{\varepsilon}{2\sqrt{n}}$ .

Enquanto o critério de parada não é verificado, tem-se  $\mathbf{c}^t \mathbf{x}_{k+1} < \mathbf{c}^t \mathbf{x}_k$  em cada iteração  $k$ .

**Prova:**

$$\begin{aligned} \mathbf{c}^t(\mathbf{x}_{k+1} - \mathbf{x}_k) &= \mathbf{c}^t(\mathbf{x}_k + \lambda_k \mathbf{d}_k - \mathbf{x}_k) = \lambda_k \mathbf{c}^t \mathbf{d}_k = -\lambda_k \|\mathbf{p}_k\|^2 - \mu \lambda_k \mathbf{v}(\mathbf{x}_k)^t \mathbf{p}_k \\ &\leq -\lambda_k \|\mathbf{p}_k\|^2 + \mu \lambda_k \|\mathbf{p}_k\|_1 \leq -\lambda_k \|\mathbf{p}_k\|^2 + \mu \sqrt{n} \lambda_k \|\mathbf{p}_k\| \\ &= -\lambda_k \|\mathbf{p}_k\| [\|\mathbf{p}_k\| - \mu \sqrt{n}] < 0, \end{aligned}$$

que segue de  $\|\mathbf{p}_k\| > \varepsilon$ , (61), (66) e de  $\frac{\|\mathbf{x}\|_1}{n} \leq \|\mathbf{x}\| \leq \sqrt{n}\|\mathbf{x}\|_\infty$ .  $\square$

## CAPÍTULO 2

### Solução de Sistemas Lineares Inconsistentes no Senso da Norma $L_1$

#### 2.1 Introdução

A motivação para o desenvolvimento de um método que resolvesse um problema linear por partes via pontos interiores, surgiu, exatamente, no estudo de um algoritmo aplicado à solução do problema sugerido no título deste capítulo. Ou seja

$$\text{Min } f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_1 = \sum_{i=1}^m |\mathbf{a}_i^t \mathbf{x} - b_i| = \sum_{i=1}^m f_i(\mathbf{x}) , \quad (72)$$

onde  $A$  uma matriz  $m \times n$ ,  $m > n$ , e  $f_i(\mathbf{x})$  é uma função linear por partes particular.

Segundo Eisenhart[1], o problema (72) já era conhecido na metade do século XVIII (sugerido por Boscovith), bem antes mesmo da introdução do Método dos Quadrados Mínimos. Em 1789, Laplace apresentou um algoritmo que resolvia (72) com  $m = 1$  e em 1809 Gauss estabeleceu, sem provas, propriedades que soluções de (72) deveriam satisfazer, o que geraria o aparecimento de algoritmos grosseiros. Só em 1887 Edgeworth apresentaria um procedimento de solução de (72) mas, devido à inviabilidade de uso dos algoritmos existentes, permaneceu um tanto esquecido, ressurgindo no século XX, com os trabalhos de Rhodes[2] em 1930 e de Singleton[3] em 1940. Nenhum desses processos viria a se tornar viável mesmo com o advento dos computadores. O grande alento na resolução de (72) seria dado com o surgimento, em 1947, do método **SIMPLEX**, devido a George B.Dantzig, para a solução de problemas de programação linear, pois em 1950

Harris[4] publicaria uma das primeiras versões de (72) na forma de um problema de programação linear. Seguindo a linha de Harris, tivemos Charnes, Cooper and Ferguson (1955)[5] e Wagner(1959)[6], este último propondo o primeiro algoritmo que resolvia (72) via programação linear. Tivemos também os trabalhos de Karst (1958)[7], Usow(1967)[8], Abdelmalek(1971)[9], Claerbaut and Muir(1973)[10], Schlossmacher(1973)[11], McCormick and Sposito(1976)[12], podendo ainda ser encontrado em Communications in Statistics, Vol. B6, N° 4,1977, uma considerável lista de referências.

O método Simplex, largamente usado na solução de problemas de programação linear, apresentou ótimos resultados quando utilizado na solução de (72). Destacaremos aqui alguns destes trabalhos: Barrodale e Young(1966)[13], Barrodale e Roberts(1973)[14](aprimoramento do trabalho anterior), Robers e Robers(1973)[15], apresentaram uma especialização do método Simplex para a solução de (72). Nesta mesma linha, com motivações diferentes, tivemos em 1978 o trabalho de Bartels, Conn e Sinclair[16]. Já em 1979 Armstrong, Frome e Kung[17], a partir da abordagem dada por Barrodale e Roberts, usaram a técnica de decomposição LU. Sendo seguido por Abdelmalek[18], 1980 ,com uma versão Dual Simplex Revisado, onde também aparecem técnicas de decomposição.

Por muito tempo o algoritmo Simplex e seus variantes foram utilizados na solução do problema ora proposto. Só a partir de 1987 começaram a aparecer trabalhos utilizando-se técnicas de *pontos interiores*, os quais serão melhor estudados na seção 2.3.

## 2.2 Preliminares

### 2.2.1 Introdução

Muitos algoritmos que resolvem (72), o fazem após transformá-lo num problema de programação linear. O que veremos agora é como isto pode ser feito. Existem várias formas para expressar (72) como um problema de programação linear. Aqui apresentamos duas delas, sendo que a primeira será a por nós utilizada.

Seguindo Witzgall[69], vemos que (72) pode ser escrito como:

$$\begin{aligned} \text{Min} \quad & \mathbf{e}^t \mathbf{w} = \sum_{i=1}^m w_i \\ \text{s.a.} \quad & w_i \geq |b_i - \mathbf{a}_i^t \mathbf{x}| \\ & w_i \text{ irrestrito,} \end{aligned}$$

ou

$$\begin{aligned} \text{Min} \quad & \mathbf{e}^t \mathbf{w} \\ \text{s.a.} \quad & w_i \geq b_i - \mathbf{a}_i^t \mathbf{x} \\ & w_i \geq \mathbf{a}_i^t \mathbf{x} - b_i, \end{aligned}$$

ou ainda

$$\begin{aligned} \text{Min} \quad & \begin{pmatrix} \mathbf{e}^t & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{x} \end{pmatrix} \\ \text{s.a.} \quad & \begin{pmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{I} & -\mathbf{A} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{x} \end{pmatrix} \geq \begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix}. \end{aligned} \tag{73}$$

Abaixo aparece outro modo de apresentar (72) como um problema linear, devido a Barrodale e Roberts[14]. Vejamos:

Em (72) fazemos  $\mathbf{r} = \mathbf{Ax} - \mathbf{b}$  e então ficamos com o problema

$$\begin{aligned} & \text{Min } \|\mathbf{r}\|_1 \\ & \text{s.a. } \mathbf{Ax} + \mathbf{r} = \mathbf{b}. \end{aligned}$$

Tomando  $\mathbf{r} = \mathbf{r}^+ - \mathbf{r}^-$ , com  $\mathbf{r}^+ \geq \mathbf{0}$ ,  $\mathbf{r}^- \geq \mathbf{0}$  e  $\mathbf{r}_j^+ \cdot \mathbf{r}_j^- = \mathbf{0}$ , teremos:

$$\begin{aligned} & \text{Min } \mathbf{e}^t \mathbf{r}^+ + \mathbf{e}^t \mathbf{r}^- \\ & \text{s.a. } \mathbf{Ax} + \mathbf{r}^+ - \mathbf{r}^- = \mathbf{b} \\ & \quad \mathbf{r}^+ \geq \mathbf{0}, \mathbf{r}^- \geq \mathbf{0}. \end{aligned}$$

Vemos então que (73) é um problema de programação linear, equivalente ao problema original (72) e abaixo apresentamos seu dual.

Considerando  $\mathbf{u}$ ,  $\mathbf{v}$  as variáveis duais, então teremos que o dual será

$$\begin{aligned} & \text{Max } \mathbf{u}^t \mathbf{b} - \mathbf{v}^t \mathbf{b} \\ & \text{s.a. } \begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{I} & -\mathbf{A} \end{bmatrix}^t \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{e} \\ \mathbf{0} \end{bmatrix} \\ & \quad \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}. \end{aligned}$$

ou

$$\begin{aligned} & \text{Max } (\mathbf{u} - \mathbf{v})^t \mathbf{b} \\ & \text{s.a. } \mathbf{A}^t (\mathbf{u} - \mathbf{v}) = \mathbf{0} \\ & \quad \mathbf{u} + \mathbf{v} = \mathbf{e} \\ & \quad \mathbf{u} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0}. \end{aligned}$$

Fazendo  $\mathbf{u} - \mathbf{v} = \mathbf{y}$  e observando que  $\mathbf{u} \geq \mathbf{0}$  e  $\mathbf{v} \geq \mathbf{0}$ , teremos

$$\begin{aligned}
& \text{Max } \mathbf{y}^t \mathbf{b} \\
& \text{s. a. } \mathbf{A}^t \mathbf{y} = \mathbf{0} \\
& -\mathbf{e} \leq \mathbf{y} \leq \mathbf{e},
\end{aligned}
\tag{74}$$

um problema linear canalizado.

### 2.2.2 Método de Newton composto nível-m modificado e perturbado.

Apenas por questão de completabilidade, apresentamos abaixo, resumidamente, uma variante do método de Newton, método este que será citado mais adiante num contexto apropriado.

Consideremos a  $k$ -ésima iteração na aplicação do método de Newton aplicado ao sistema  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$

$$\mathbf{x}^k = \mathbf{x}^{k-1} + \alpha^{k-1} \Delta \mathbf{x}^k,$$

onde

$$\Delta \mathbf{x}^k = -\mathbf{J}(\mathbf{x}^{k-1})\mathbf{F}(\mathbf{x}^{k-1}),$$

sendo  $\mathbf{J}(\mathbf{x}^{k-1})$  a matriz jacobiana de  $F$  em  $\mathbf{x}^{k-1}$ .

- Quando  $\alpha^{k-1} = 1$ , temos o conhecido método de Newton. Já quando  $\alpha^{k-1} \neq 1$ , dizemos que temos o método de Newton modificado.

- Podemos introduzir uma simplificação usando a matriz jacobiana inicial em todas as iterações seguintes, obtendo então o chamado método de Newton simplificado.

- Quando entre dois passos de Newton damos  $m$  passos de Newton simplificado, temos o método de Newton composto nível- $m$ .

- Na seção 2.3.1 aparecerá a justificativa do qualificativo "perturbado".

## 2.3 Métodos de Pontos Interiores

Apresentamos a seguir uma lista de artigos que tratam da solução de (72), sob a ótica de pontos interiores. Sobre alguns deles faremos uma apresentação mais detalhada logo à frente, explorando as principais idéias empregadas.

- 1- "Least Absolute Value Regression", Meketon[19](1987).
- 2- "An Assumption-Free Convergence Analysis for a Perturbation of the Scaling Algorithm for Linear Programs with Application to the  $L_1$  Estimation Problem"(1988), artigo publicado por Sherali, Skarpness e Kim[20].
- 3- " $L_1$  and  $L_\infty$  Minimization Via a Variant of Karmarkar's Algorithm", onde Ruzinky e Olsen[21] resolvem (72) através de um variante do algoritmo de Karmarkar(1989).
- 4- "A Global and Quadratic Affine Scaling Method for Linear  $L_1$  Problems", um trabalho não especificamente de programação linear, mas parcialmente inspirado em técnicas afim-escala, devido a Coleman e Li(1989)[22].
- 5- "A Primal - Dual Interior Point Approach for Computing the  $L_1$  and  $L_\infty$  Solutions of Overdetermined Linear Systems (1990). Um trabalho de Zhang[23] com uma abordagem primal-dual que, segundo alguns autores, tem a melhor performance do ponto de vista computacional, sem que isto tenha sido, entretanto, analisado pelo autor.

6-"An Interior Point Method Specialized to the  $L_1$  Regression Problem" (1991), um trabalho onde Oliveira, Nascimento e Lyra[24] tiram proveito da estrutura que se origina quando (72) é visto como um problema de programação linear.

7-"Primal-Dual Interior Point Algorithms for LSAD and LMAD Estimation "publicado por Duarte e Vanderbei[25] em 1994, onde algoritmos especializados são utilizados para resolver (72) via simplex e pontos interiores, comparando então a eficiência computacional.

Passamos agora a destacar diferentes abordagens na solução de (72), via pontos interiores.

**2.3.1-** Zhang[23], foi o primeiro a aplicar algoritmos de pontos interiores primal-dual na solução de (72), conseguindo prova de convergência super-linear ou quadrática, desde que a factibilidade primal e dual fosse cumprida em cada iteração. Ele fez uso de uma técnica muito utilizada em programação não linear, que corresponde à aplicação das condições de primeira ordem. Então, a partir do problema (73) obteve o sistema de equações não-lineares

$$F_1(\mathbf{u}, \mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{x}, \mathbf{w}) = \begin{bmatrix} \mathbf{A}^t(\mathbf{u} - \mathbf{v}) \\ \mathbf{u} + \mathbf{v} - \mathbf{e} \\ \mathbf{p} + \mathbf{Ax} - \mathbf{w} - \mathbf{b} \\ \mathbf{q} - \mathbf{Ax} - \mathbf{w} + \mathbf{b} \\ \mathbf{Up} \\ \mathbf{Vq} \end{bmatrix} = \mathbf{0} \quad (75)$$

onde

$\mathbf{u}, \mathbf{v}$  as variáveis duais,

$\mathbf{p}, \mathbf{q}$  as variáveis de folgas primais,

$\mathbf{U}, \mathbf{V}$  as matrizes diagonais formadas pelas coordenadas dos vetores  $\mathbf{u}$  e  $\mathbf{v}$ , respectivamente.

Agora, ao exigirmos a factibilidade estrita e que estejamos sobre a trajetória central, iremos obter um sistema  $\hat{\mathbf{F}}(\mathbf{x}, \mu)$  que é uma perturbação do sistema  $\mathbf{F}_1(\mathbf{x}) = 0$ , sendo  $\mu > 0$  e a perturbação  $\mu\mathbf{e}$  é adicionada à parte não-linear de  $\mathbf{F}_1(\mathbf{x})=0$ . Temos então

$$\hat{\mathbf{F}}(\mathbf{u}, \mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{x}, \mathbf{w}, \mu) = \begin{bmatrix} \mathbf{A}^t(\mathbf{u} - \mathbf{v}) \\ \mathbf{u} + \mathbf{v} - \mathbf{e} \\ \mathbf{p} + \mathbf{Ax} - \mathbf{w} - \mathbf{b} \\ \mathbf{q} - \mathbf{Ax} - \mathbf{w} + \mathbf{b} \\ \mathbf{UPe} - \mu\mathbf{e} \\ \mathbf{VQe} - \mu\mathbf{e} \end{bmatrix} = \mathbf{0} \quad (75a)$$

Se Newton é aplicado em  $\hat{\mathbf{F}}(\mathbf{x}, \mu)$ , o novo sistema obtido por uma perturbação aplicada em  $\mathbf{F}_1(\mathbf{x})=0$ , teremos o chamado método de Newton perturbado. Para um detalhamento maior ver Tapia, Zhang, Saltzman and Weiser[73] e Zhang, Tapia, Dennis[74].

Observemos que em (75) a parte linear representa exatamente as condições de factibilidade primal e dual e, a parte não-linear, as folgas complementares. Usando então o método de Newton composto nível-1 modificado e perturbado para resolver o sistema de equações não-lineares, Zhang propôs o algoritmo 2 abaixo, um algoritmo primal-dual de

simples implementação, segundo o autor, e com taxa de convergência superlinear que pode chegar a quadrática, dependendo da escolha dos parâmetros  $\sigma$  e  $\tau$  (ver [23]). A estrutura primal-dual que surge neste, e em vários outros algoritmos, pode ser obtida através de outras abordagens, como por exemplo, trajetória central como em Megiddo[61] ou Monteiro e Adler[70], ou ainda, redução potencial como em Gonzaga e Todd[71] ou Todd e Ye[52].

### Algoritmo 2

Passo 1. Dados  $x^0$ ,  $\varepsilon > 0$ ,  $\delta > 0$ , calcule  $\mathbf{r}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$  e inicialize com

$$\mathbf{u}_0 = \frac{1}{2}\mathbf{e},$$

$$\mathbf{v}_0 = \frac{1}{2}\mathbf{e},$$

$$\mathbf{w}_0 = |\mathbf{r}_0| + \delta\mathbf{e},$$

$$\mathbf{p}_0 = \mathbf{w}_0 - \mathbf{r}_0,$$

$$\mathbf{q}_0 = \mathbf{w}_0 + \mathbf{r}_0,$$

onde  $|\mathbf{r}_0|$  é o vetor dos valores absolutos das componentes de  $\mathbf{r}_0$ .

Passo 2. Seja  $g_k = \mathbf{p}_k^t \mathbf{u}_k + \mathbf{q}_k^t \mathbf{v}_k$  o gap de dualidade. Se  $g_k/2n \leq \varepsilon$ , então fim.

Passo 3. Para  $\sigma_k \in (0,1)$ ,  $\sigma_k = \text{Min}(0.1, g_k/2n)$ , seja  $\mu = \sigma_k g_k/2n$ . Calcule então o

passo de Newton  $d\mathbf{z}_k^0$  resolvendo

$$\mathbf{F}'_1(\mathbf{z}_k) d\mathbf{z}_k^0 = -\mathbf{F}_1(\mathbf{z}_k)$$

e em seguida

$$\mathbf{F}'_1(\mathbf{z}_k)\mathbf{dz}_k^1 = -\mathbf{F}_1(\mathbf{z}_k + \mathbf{dz}_k^0) + \mathbf{c}_k$$

para encontrar o passo de Newton centralizador  $\mathbf{dz}_k^1$ . O passo de Newton composto centralizado é então dado por

$$\mathbf{dz}_k = \mathbf{dz}_k^0 + \mathbf{dz}_k^1 \equiv (\mathbf{du}_k, \mathbf{dv}_k, \mathbf{dp}_k, \mathbf{dq}_k, \mathbf{dx}_k, \mathbf{dw}_k).$$

onde

$$\mathbf{z} = (\mathbf{u}, \mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{x}, \mathbf{w}),$$

$$\mathbf{dz} = (\mathbf{du}, \mathbf{dv}, \mathbf{dp}, \mathbf{dq}, \mathbf{dx}, \mathbf{dw})$$

e

$$\mathbf{c}_k^t = \mu_k(0\dots 01\dots 1)$$

é o chamado termo centralizador.

Passo 4- Nesse ponto é bom observar que a escolha dos parâmetros utilizados abaixo fica um pouco mais claro quando da prova que  $g_{k+1} \leq g_k$  na busca da convergência.

Seja então  $\tau_k \in (0,1)$ ,  $\tau_k = \text{Max}(0.99, 1 - (g_k/2n))$  e tome

$$\hat{\alpha}_k = \frac{-1}{\text{Min}(\mathbf{U}_k^{-1}\mathbf{du}_k, \mathbf{V}_k^{-1}\mathbf{dv}_k)}$$

e

$$\hat{\beta}_k = \frac{-1}{\text{Min}(\mathbf{P}_k^{-1}\mathbf{dp}_k, \mathbf{Q}_k^{-1}\mathbf{dq}_k)},$$

com  $\mathbf{P}$  e  $\mathbf{Q}$  sendo as matrizes diagonais formadas pelas coordenadas dos vetores  $\mathbf{p}$  e  $\mathbf{q}$ , respectivamente.

Faça

$$\hat{\beta}_k = \hat{\alpha}_k \text{ se } \hat{\beta}_k < 0$$

e

$$(\alpha_k, \beta_k) = \begin{cases} \tau_k(\hat{\alpha}_k, \hat{\alpha}_k), & \text{se } \hat{\beta}_k > \hat{\alpha}_k \text{ e } \mathbf{u}_k^t \mathbf{d}\mathbf{p}_k + \mathbf{v}_k^t \mathbf{d}\mathbf{q}_k > \mathbf{0} \\ \tau_k(\hat{\beta}_k, \hat{\beta}_k), & \text{se } \hat{\alpha}_k > \hat{\beta}_k \text{ e } \mathbf{p}_k^t \mathbf{d}\mathbf{u}_k + \mathbf{q}_k^t \mathbf{d}\mathbf{v}_k > \mathbf{0} \\ \tau_k(\hat{\alpha}_k, \hat{\beta}_k), & \text{caso contrário.} \end{cases}$$

Passo 5. Calcule

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{d}\mathbf{u}_k,$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \alpha_k \mathbf{d}\mathbf{v}_k,$$

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \beta_k \mathbf{d}\mathbf{p}_k,$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \beta_k \mathbf{d}\mathbf{q}_k,$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta_k \mathbf{d}\mathbf{x}_k,$$

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \beta_k \mathbf{d}\mathbf{w}_k.$$

Passo 6. Faça  $k = k+1$  e volte ao passo 1

**2.3.2** O artigo de Sherali, Skarpness e Kim já foi detalhado na seção 1.5. Eles apresentaram um algoritmo de pontos interiores para resolver um problema de programação linear canalizado. Na seção 2.2, vimos que , quando tomamos o PL equivalente a (72) e calculamos seu dual, obtemos um problema de programação linear com variáveis canalizadas (74), que assim pode ser resolvido pelo algoritmo 1, apresentado na seção 1.5.

**2.3.3** Na terceira abordagem a ser discutida, Coleman e Li[22], por outro lado, consideraram o problema (72) numa forma equivalente, baseando-se no trabalho de Seneta e Steiger[75]. A forma equivalente adotada foi :

$$\begin{aligned} \text{Min } \psi(\mathbf{r}) &= \|\mathbf{r}\|_1 \\ \text{s.a. } \mathbf{Zr} &= \mathbf{Zb} \end{aligned} \tag{76}$$

onde  $r_i = b_i - a_i^t x$  e  $\mathbf{Z}^t$  é tal que suas colunas formam uma base para o espaço nulo da matriz  $\mathbf{A}$ , ou seja,  $\mathbf{ZA} = \mathbf{0}$ . Coleman e Li apresentaram algoritmos onde a matriz  $\mathbf{Z}$  não necessariamente é calculada.

O primeiro algoritmo, de convergência linear, baseado num método afim escala, onde agora uma direção de descida é pesquisada analisando-se a distância da solução corrente às linhas de não-diferenciabilidade  $r_i = 0$ , e não mais a distância à fronteira. Ainda mais, o algoritmo permite que ultrapassemos as já citadas linhas de não-diferenciabilidade, quando da busca unidimensional linear por partes, contanto que continuemos numa direção de descida. Nesse caso, a adaptação do gradiente se faz necessária.

Supondo que estejamos numa região de diferenciabilidade ( $r_i \neq 0$ ,  $i = 1, 2, \dots, m$ ), uma direção de descida é encontrada resolvendo-se o seguinte problema de região de confiança:

$$\begin{aligned} \text{Min } \mathbf{g}^t \mathbf{d} \\ \text{s.a. } \mathbf{Zd} &= \mathbf{0} \\ \|\mathbf{D}^{-1} \mathbf{d}\| &\leq \delta, \end{aligned} \tag{77}$$

onde  $\delta$  reflete o tamanho da região de confiança,  $\mathbf{g}$  é o gradiente de  $\psi(\mathbf{r})$  e

$\mathbf{D} = \text{diag} \left\{ |r_i|^{1/2} \right\}$ , permitindo que o elipsóide se dilate nas direções correspondentes a  $|r_i|$  relativamente grandes e se contraia quando as componentes de  $|r_i|$  estiverem próximo de zero.

A solução de (77) (ver, por exemplo, Gonzaga[84]) é múltipla de

$$\mathbf{d} = -\mathbf{D}^2 (\mathbf{g} - \mathbf{Z}^t (\mathbf{Z} \mathbf{D}^2 \mathbf{Z}^t)^{-1} \mathbf{Z} \mathbf{D}^2 \mathbf{g}) \quad (78)$$

Ao invés de previamente escolher um valor para  $\delta$ , o autor preferiu calcular  $\mathbf{d}$  por (78) e então computar o tamanho do passo  $\alpha$  minimizando a função linear por partes  $\psi(\mathbf{r} + \alpha \mathbf{d})$  ao longo da direção  $\mathbf{d}$ .

Lembrando-se que as linhas de não-diferenciabilidade podem ser ultrapassadas, consideremos os pontos críticos não-negativos

$$\mathfrak{J} = \left\{ \alpha_i, \text{com } \alpha_i > 0, \alpha_i = -r_i / d_i \right\}.$$

Calculamos então o ponto *crítico ótimo*  $\alpha_*$ , de modo que

$$\psi(\mathbf{r} + \alpha_* \mathbf{d}) = \underset{\alpha > 0}{\text{Min}} \psi(\mathbf{r} + \alpha \mathbf{d}) = \underset{\alpha \in \mathfrak{J}}{\text{Min}} \psi(\mathbf{r} + \alpha \mathbf{d}).$$

Aqui  $\alpha_*$  é calculado considerando-se cada elemento de  $\mathfrak{J}$  e atualizando-se o gradiente sempre que ultrapassamos o ponto crítico. Por exemplo, sendo  $\alpha_0$  o menor elemento de  $\mathfrak{J}$ , então um passo ligeiramente além  $\alpha_0$  produz o seguinte gradiente :

$$\mathbf{g}^+ = \mathbf{g} - 2\mathbf{g}_0 \mathbf{e}_0.$$

Chegamos então no seguinte:

### Algoritmo 3a

Dado  $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$ :

Passo 1. Tomamos

$$\mathbf{D}_k = \text{diag}\left\{\left|r_i^k\right|^{1/2}\right\}$$

e

$$\mathbf{g}^k = \text{sgn}(\mathbf{r}^k).$$

Passo 2. Calculemos

$$\mathbf{d}^k = -(\mathbf{D}_k)^2 \left\{ \mathbf{g}^k - (\mathbf{Z}_k)^t (\mathbf{Z}_k (\mathbf{D}_k)^2 (\mathbf{Z}_k)^t)^{-1} \mathbf{Z}_k (\mathbf{D}_k)^2 \mathbf{g}^k \right\}.$$

Passo 3. Aplique o procedimento de busca linear por partes, abaixo descrito, para encontrar  $\alpha^k$  e faça

$$\mathbf{r}^{k+1} = \mathbf{r}^k + \alpha^k \mathbf{d}^k, \quad k \leftarrow k + 1 \text{ e volte ao passo 1}.$$

O procedimento citado no Passo 3 acima é agora descrito:

#### Busca Linear por Partes

Tendo determinado  $\mathbf{d}$  e tomando  $\tau$  com  $0 < \tau < 1$ :

Passo 1. Calcule  $\mathfrak{S} = \{\alpha_i : \alpha_i = -r_i/d_i, r_i d_i < 0\}$ .

Passo 2. Calcule o ponto crítico ótimo  $\alpha_*$  com

$$\psi(\mathbf{r} + \alpha_* \mathbf{d}) = \underset{\alpha > 0}{\text{Min}} \psi(\mathbf{r} + \alpha \mathbf{d}) = \underset{\alpha \in \mathfrak{S}}{\text{Min}} \psi(\mathbf{r} + \alpha \mathbf{d}).$$

Passo 3. Seja

$$\alpha_o = \text{Max}_{\mathbb{R} \cup \{0\}} \{ \alpha_i : 0 \leq \alpha_i < \alpha_* \} \quad (79)$$

e então calcule o tamanho do passo

$$\alpha = \alpha_o + \tau(\alpha_* - \alpha_o). \quad (80)$$

Coleman e Li apresentam um segundo algoritmo com uma nova maneira de calcular a matriz diagonal  $D$  e uma escolha para  $\tau$  que nos leva a um algoritmo de convergência quadrática, onde a matriz  $Z$  não é necessariamente calculada.

Sejam:

$$(\mathbf{D}_k)^2 = \left| \mathbf{D}_k^r (\mathbf{D}_k^\theta)^{-1} \right| \quad (81a)$$

$$\mathbf{D}_k^r = \text{diag}(\mathbf{r}^k), \quad (81b)$$

$$\mathbf{D}_k^\theta = \theta^k \text{diag}(\mathbf{g}^k) + (1 - \theta^k) \mathbf{D}_k^\lambda, \quad (81c)$$

onde  $\mathbf{D}_k^\lambda = \text{diag}(\mathbf{g} - \mathbf{Z}^t \mathbf{w})$ .

Observações:

- Para uma escolha adequada de  $\theta^k$  veja[22].
- Por  $|\mathbf{M}|$  entende-se a matriz obtida trocando-se os elementos  $m_{ij}$  por  $|m_{ij}|$ .

Agora calculemos  $\mathbf{d}^k$  de uma das seguintes formas:

i-) Resolvemos o sistema

$$\left[ \text{diag}(\mathbf{g}^k) \middle| \mathbf{D}_k^\theta \right] - \mathbf{D}_k^r \mathbf{Z}^t \begin{bmatrix} \mathbf{d}_x^k \\ \mathbf{d}_w^k \end{bmatrix} = -\mathbf{D}_k^r (\mathbf{g}^k - \mathbf{Z}^t \mathbf{w}^k) \quad (82)$$

e então fazemos  $\mathbf{d}^k = -\mathbf{A}^t \mathbf{d}_x^k$ . Também pelo métodos dos quadrados mínimos calculamos o valor de  $\mathbf{w}^{k+1}$  resolvendo o sistema

$$\mathbf{D}_k \mathbf{Z}^t \mathbf{w}^{k+1} = \mathbf{D}_k \mathbf{g}^k.$$

ii-)Resolvemos  $(\mathbf{D}_k)^{-1} \mathbf{A}^t \mathbf{d}_x^k = \mathbf{D}_k \mathbf{g}^k$  por quadrados mínimos

e então tomamos  $\mathbf{d}^k = -\mathbf{A}^t \mathbf{d}_x^k$ . (83)

A adaptação das variáveis duais é feita por

$$\lambda^{k+1} = \mathbf{g}^k + (\mathbf{D}_k)^{-2} \mathbf{d}^k. \quad (84)$$

iii-)Por quadrados mínimos resolvemos

$$\mathbf{D}_k \mathbf{Z}^t \mathbf{w}^{k+1} = \mathbf{D}_k \mathbf{g}^k$$

e então tomamos

$$\mathbf{d}^k = -(\mathbf{D}_k)^{-2} (\mathbf{g}^k - \mathbf{Z}^t \mathbf{w}^{k+1}). \quad (85)$$

Observação: Ao aplicarem o procedimento de busca linear por partes, afim de que se conseguisse convergência quadrática , os autores usaram, ao invés de  $\tau$  constante,

$$\tau^k = \mathbf{max}\{\tau, 1 - \theta^k\} \quad (86)$$

e portanto

$$\alpha^k = \alpha_o^k + \tau^k (\alpha_*^k - \alpha_o^k). \quad (87)$$

Podemos agora descrever:

### Algoritmo 3b

Passo 1. Calculado  $\theta^k$ , de (81) tiramos  $\mathbf{D}_k$ , considerando  $\mathbf{g}^k = \mathbf{sgn}(\mathbf{r}^k)$ .

Passo 2. Calculamos  $\mathbf{d}^k$  e  $\mathbf{w}^{k+1}$  usando uma das formas mencionadas acima.

Passo 3. Utilizamos o procedimento de busca linear por partes usando (86) e (87), e tomamos  $\mathbf{r}^{k+1} = \mathbf{r}^k + \alpha^k \mathbf{d}^k$ .

Na abordagem acima podemos observar que:

a-) exige-se a resolução de grandes sistemas e, para tanto, técnicas de fatoração são necessárias, juntamente com outras que mantenham a estabilidade numérica.

b-) em alguns casos o cálculo da matriz  $\mathbf{Z}$  é necessário(veja Coleman e Li[22]).

**2.3.4-** Duarte e Vanderbei [25] comparam o desempenho computacional de um algoritmo de ponto interior primal-dual especializado, aplicado na solução do par de problemas (73) e (74), com um eficiente algoritmo simplex especializado. Para tanto somos levados aos trabalhos de [14], [17], [76], [16], [77], [78], [79], [80]. Em [81] e [82] encontramos provas da complexidade polinomial para variantes nos quais a exigência da factibilidade dual não se faz necessária. No caso analisado por Duarte e Vanderbei, o algoritmo de pontos interiores mostrou uma maior eficiência na solução do problema proposto.

Como podemos observar, a maior parte dos algoritmos apresentados partem do entendimento que (72) pode ser visto como um problema de programação linear e então diferentes técnicas de pontos interiores são aplicadas.

## CAPÍTULO 3

### Algoritmo Linear Por Partes Via Método Simplex.

#### 3.1- Introdução

Neste capítulo, começaremos a estudar problemas de programação matemática que envolvem a minimização de uma função linear por partes, separável e convexa. Ou seja, trabalharemos com o seguinte problema:

$$\begin{aligned} \text{Min } f(\mathbf{x}) &= \sum_{j=1}^n f_j(x_j) \\ \text{s. a. } \mathbf{Ax} &= \mathbf{b} \end{aligned} \tag{88.1}$$

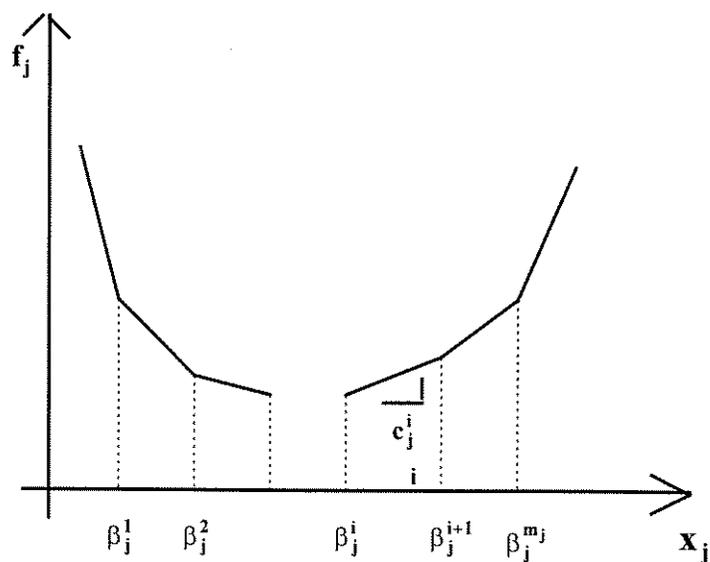
onde cada  $f_j(x_j)$  é uma função linear por partes definida por

$$f_j(x_j) = \begin{cases} c_j^1 x_j, & \beta_j^0 \leq x_j \leq \beta_j^1 \\ c_j^2 (x_j - \beta_j^1) + f_j(\beta_j^1), & \beta_j^1 \leq x_j \leq \beta_j^2 \\ \vdots \\ c_j^{m_j} (x_j - \beta_j^{m_j-1}) + f_j(\beta_j^{m_j-1}), & \beta_j^{m_j-1} \leq x_j \leq \beta_j^{m_j} \end{cases} \tag{88.2}$$

com

$$c_j^i < c_j^{i+1}, \beta_j^i < \beta_j^{i+1}, i = 1, \dots, m_j - 1.$$

Apresentamos a seguir a representação gráfica de uma particular  $f_j$ .



Os escalares  $\beta_j^i$ ,  $1 \leq i \leq m_j$ , são chamados pontos de quebra ou de não-diferenciabilidade de  $f_j$ .

Para ilustrar, considere a função linear por partes  $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$  definida por:

$$f_1(x_1) = \begin{cases} -3x_1 + 9 & 0 \leq x_1 \leq 1 \\ -2x_1 + 8 & 1 \leq x_1 \leq 2 \\ 4 & 2 \leq x_1 \leq 3 \\ x_1 + 1 & 3 \leq x_1 \leq 4 \end{cases}$$

$$f_2(x_2) = \begin{cases} -3x_2 + 3 & 0 \leq x_2 \leq 1 \\ -2x_2 + 2 & 1 \leq x_2 \leq 2 \\ x_2 - 4 & 2 \leq x_2 \leq 3 \\ 2x_2 - 7 & 3 \leq x_2 \leq 4, \end{cases}$$

cujo gráfico é mostrado na figura 1.

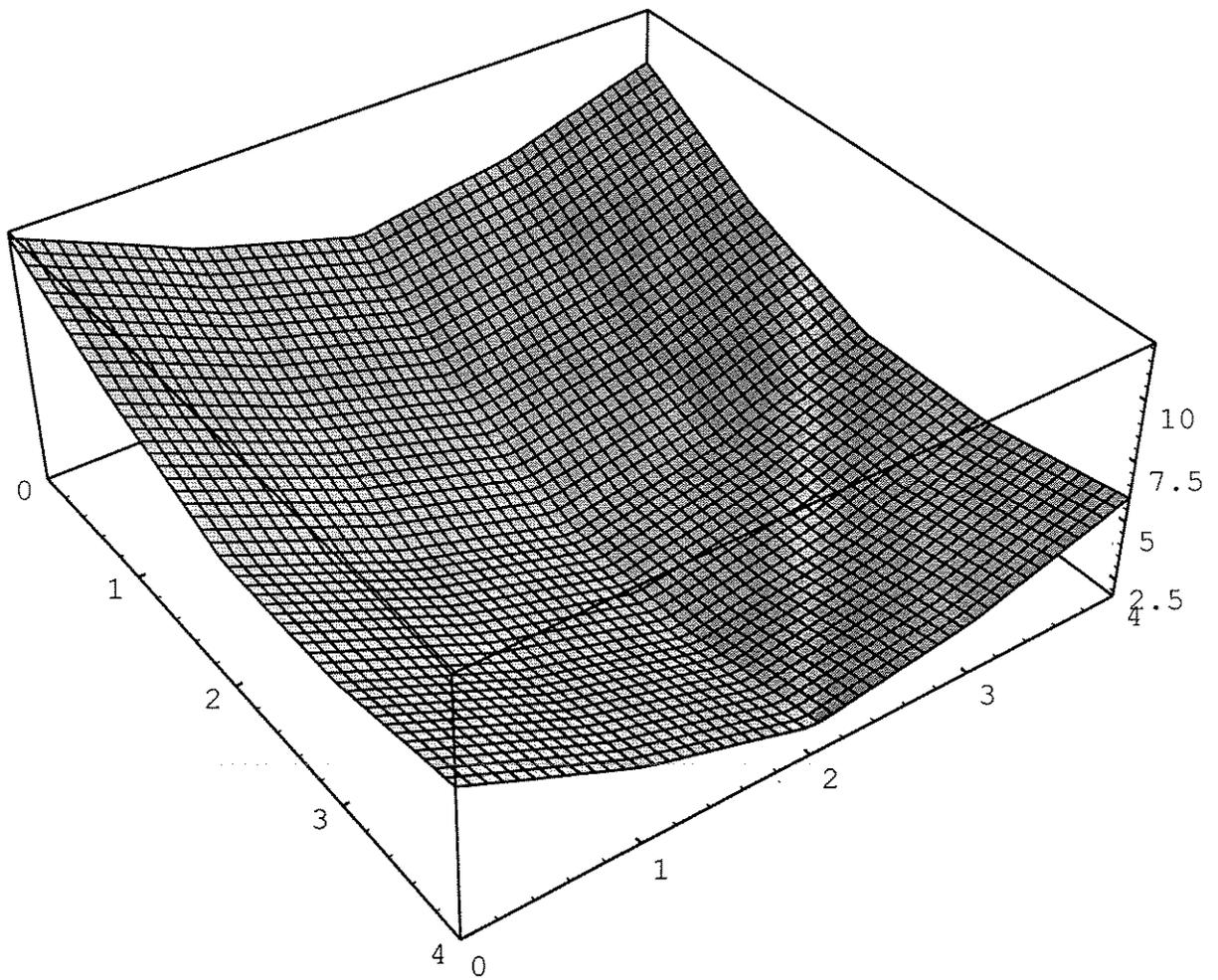


figura 1

Considere ainda as restrições lineares dadas por:

$$\begin{cases} -x_1 + x_2 \leq 2 \\ x_1 + 3x_2 \leq 14 \\ 3x_1 - x_2 \leq 12 \end{cases}$$

A Figura 2 ilustra a região de factibilidade (delimitadas pelas linhas em negrito), bem como as regiões onde a função objetivo  $f(\mathbf{x})$  é linear (delimitadas pelas linhas pontilhadas).

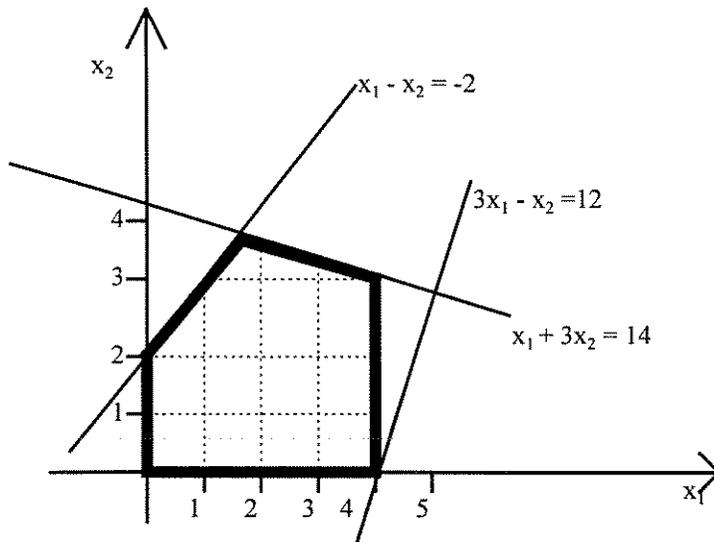


Figura 2.

Problemas de programação linear por partes aparecem no estudo de expansão de redes telefônicas (veja Nakagawa[85], França, Fernandes e Tavares[86]), de planejamento de produção (veja Golstein e Youdine[87] e, no campo teórico, no cálculo da solução inicial de problemas de programação linear(fase1)(veja [110]), aproximações de problemas não lineares(veja[34]), etc.

Algoritmos de pontos interiores estão fortemente amarrados ao conceito de fronteira, esta por sua vez, relacionada à idéia de escalamento. Para colocar o problema (88) neste contexto, introduzimos os conceitos de ponto interior e de fronteira virtual.

Observe no simples exemplo acima que a solução ótima pode ser um vértice, ou pertencente a uma aresta, ou ainda interior à região de factibilidade, como em problemas de programação não-linear. Entretanto, assumindo localmente as linhas de não-diferenciabilidade como *fronteiras virtuais*, a solução ótima ocorrerá num vértice. A seguinte definição caracteriza tais pontos que, por analogia à programação linear, chamaremos de soluções básicas.

**Definição 1.** (*solução básica factível*) Considere uma partição básica nas colunas de  $\mathbf{A}$ , isto é,  $\mathbf{A}=(\mathbf{B},\mathbf{N})$  onde  $\mathbf{B}$  é uma matriz  $m \times m$  não-singular. Esta partição é estendida ao vetor  $\mathbf{x}$ :  $\mathbf{x}=(\mathbf{x}_B, \mathbf{x}_N)$ . Sejam  $B_1, B_2, \dots, B_m$  os índices das colunas de  $\mathbf{A}$  que formam a matriz  $\mathbf{B}$  e  $N_1, N_2, \dots, N_{m-n}$  os índices das colunas de  $\mathbf{N}$ . Supondo que:  $x_{N_j} = \beta_{N_j}^i$  para algum  $i = 0, \dots, m_{N_j}, j=1, \dots, m-n$  e .

$$\mathbf{x}_B = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{N}\mathbf{x}_N),$$

então a solução é chamada *solução básica do programa linear por partes*, ou simplesmente, *solução básica linear por partes*. Se  $\mathbf{x}_B$  é tal que  $\beta_j^{i-1} < (x_B)_j < \beta_j^i$  então é *básica factível*.

Como já observado acima, esta solução não é necessariamente um vértice da região de factibilidade. Na verdade seria um vértice somente se  $x_{N_j} = \beta_{N_j}^0$  ou  $\beta_{N_j}^{m_j}$ ,  $j = 1, 2, \dots, n$ .

**Definição 2.** Diremos que  $\mathbf{x} \in \mathcal{R}^n$  é um *ponto interior linear por partes* se :

i-)  $\mathbf{Ax} = \mathbf{b}$

ii-)  $\beta_j^{i_{j-1}} < x_j < \beta_j^{i_j}$ , para todo  $j$ .

### 3.2 Conversão para um Problema de Programação Linear

Nesta seção apresentaremos um problema de programação linear equivalente ao problema (88).

Na solução de (88), a estratégia geralmente sugerida em livros é a conversão para problema de programação linear equivalente, e então emprega-se o método simplex padrão ou um de seus variantes especializados Dantzig[88], Charnes e Lemke[89], Dantzig, Johnson e White[90], Charnes, Cooper e Ferguson[5], semelhante ao apresentado no capítulo anterior. A dificuldade de tal estratégia é o aumento do número de variáveis em relação ao problema original. Uma abordagem eficiente é o uso de um algoritmo simplex linear por partes que explora o problema diretamente, como em Goldstein e Youdine[87], Fourer[95], Snyder[92] e Premoli[96]. Já como um problema típico de programação não linear, encontramos os trabalhos de Rockafellar[91], Snyder[92], Conn[93] e Bartels[94].

Vejamos a forma equivalente de (88), como um problema de programação linear, devido a Katta G. Murty[97].

Considerando (88.2), escrevemos as variáveis originais como somas de novas variáveis:

$$x_j = \sum_{i=1}^{m_j} x_j^i \quad \text{com } 0 \leq x_j^i \leq \beta_j^i - \beta_j^{i-1}, \quad i = 1, \dots, m_j \quad (89.1)$$

Chamamos  $x_j^i$  de *componente* de  $x_j$ . Além disso, impomos a condição:

$$x_j \in (\beta_j^k, \beta_j^{k+1}) \Leftrightarrow \begin{cases} x_j^i = \beta_j^{i+1} - \beta_j^i, & i < k - 1, \\ x_j^k = x_j - \beta_j^k \in (0, \beta_j^{k+1} - \beta_j^k), \\ x_j^i = 0, & i > k + 1. \end{cases} \quad (89.2)$$

Com isto podemos escrever o seguinte programa linear equivalente ao problema (88):

$$\begin{aligned} \text{Min } & \sum_{j=1}^n \sum_{i=1}^{m_j} c_j^i x_j^i \\ \text{s.a: } & \sum_{j=1}^n \sum_{i=1}^{m_j} a_j x_j^i = b, \\ & 0 \leq x_j^i \leq \beta_j^i - \beta_j^{i-1}, \quad i = 1, \dots, m_j, \quad j = 1, \dots, n, \end{aligned} \quad (89.3)$$

que possui  $m$  restrições,  $\sum_{j=1}^n m_j$  variáveis, a matriz de restrições tem a forma:

$$\tilde{A} = [\mathbf{a}_1 \ \mathbf{a}_1 \dots \mathbf{a}_1 \ \mathbf{a}_2 \ \mathbf{a}_2 \dots \mathbf{a}_2 \dots \mathbf{a}_n \ \mathbf{a}_n \dots \mathbf{a}_n],$$

com  $m_j$  colunas formadas por  $\mathbf{a}_j$ ,  $j = 1, 2, \dots, n$  e uma solução é dada por:

$$\tilde{\mathbf{x}} = (x_1^1, x_1^2 \dots x_1^{m_1}, x_2^1, x_2^2 \dots x_2^{m_2}, x_n^1 \dots x_n^{m_n}).$$

Vejamos um exemplo.

$$\begin{aligned} & \text{Min } f_1(x_1) + f_2(x_2) \\ & \text{s.a. } \begin{cases} 3x_1 - 2x_2 = 1 \\ 4x_1 + x_2 = 0 \end{cases} \end{aligned}$$

onde  $f_1$  e  $f_2$  são dadas por:

$$f_1(x_1) = \begin{cases} -3x_1 + 9 & 0 \leq x_1 \leq 1 \\ -2x_1 + 8 & 1 \leq x_1 \leq 2 \\ 4 & 2 \leq x_1 \leq 3 \\ x_1 + 1 & 3 \leq x_1 \leq 4 \end{cases}$$

$$f_2(x_2) = \begin{cases} -3x_2 + 3 & 0 \leq x_2 \leq 1 \\ -2x_2 + 2 & 1 \leq x_2 \leq 2 \\ x_2 - 4 & 2 \leq x_2 \leq 3 \\ 2x_2 - 7 & 3 \leq x_2 \leq 4, \end{cases}$$

Aplicando a técnica acima, obtemos o PL

$$\begin{aligned} \min \quad & -3x_1^1 - 2x_1^2 + 0x_1^3 + x_1^4 - 3x_2^1 - 2x_2^2 + x_2^3 + 2x_2^4 \\ & 3(x_1^1 + x_1^2 + x_1^3 + x_1^4) - 2(x_2^1 + x_2^2 + x_2^3 + x_2^4) = 1 \\ & 4(x_1^1 + x_1^2 + x_1^3 + x_1^4) + (x_2^1 + x_2^2 + x_2^3 + x_2^4) = 0 \end{aligned}$$

com

$$\begin{aligned} & 0 \leq x_1^1 \leq 1, 0 \leq x_1^2 \leq 1, 0 \leq x_1^3 \leq 1, 0 \leq x_1^4 \leq 1, 0 \leq x_2^1 \leq 1, 0 \leq x_2^2 \leq 1, \\ & 0 \leq x_2^3 \leq 1, 0 \leq x_2^4 \leq 1. \end{aligned}$$

A seguir mostramos algumas propriedades de uma solução ótima do problema (89).

**Propriedade 1.** Considere  $\tilde{x} = (x_1^1, x_1^2 \cdots x_1^{m_1}, x_2^1, x_2^2 \cdots x_2^{m_2}, x_n^1 \cdots x_n^{m_n})$  uma solução básica ótima não-degenerada do problema (89.3). Então:

- Se  $x_j^k$  for uma variável básica, então  $x_j^1, x_j^2 \dots x_j^{k-1}, x_j^{k+1} \dots x_j^{m_j}$  são necessariamente não-básicas, com  $x_j^i = \beta_j^{i+1} - \beta_j^i, i \leq k-1$  e  $x_j^i = 0, i \geq k+1$ ;
- se  $x_j^1 = 0$  então  $x_j^2 = \dots = x_j^{m_j} = 0$  (ou seja, se  $x_j^1$  for uma variável não-básica no limite inferior, então  $x_j^i$ , para todo  $i$ , são também variáveis não-básicas no limite inferior);
- Se  $x_j^{m_j} = \beta_j^{m_j} - \beta_j^{m_j-1}$ , então  $x_j^i = \beta_j^i - \beta_j^{i-1}, i = 1, \dots, m_j - 1$  (ou seja, se  $x_j^{m_j}$  for variável não-básica no limite superior, então  $x_j^i$ , para todo  $i$ , são também variáveis não-básicas no limite superior).

Prova:

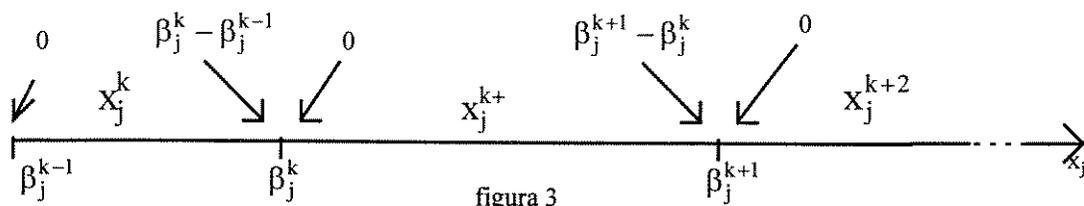
Para a prova da primeira afirmação, notemos que para  $x_j^k$  uma variável básica, segue que  $c_j^k - \pi^t a_j = 0$ , onde  $\pi$  é o vetor multiplicador simplex. Como para  $i = 0, 1, \dots, k-1$ , temos  $c_j^i < c_j^k$ , segue que  $c_j^i - \pi^t a_j < c_j^k - \pi^t a_j$ , o que implica  $c_j^i - \pi^t a_j < 0$ . Então  $x_j^i = \beta_j^{i+1} - \beta_j^i$  (caso contrário a solução não seria ótima).

Por outro lado, para  $i = k+1, \dots, m_j$ ,  $c_j^i > c_j^k$  implicará  $c_j^i - \pi^t a_j > 0$ . Então  $x_j^i = 0$ .

Na prova da segunda afirmação observamos que  $c_j^0 - \pi^t a_j \geq 0$ , pois  $x_j^0$  é uma variável não-básica no limite inferior. Como  $c_j^i > c_j^0$ , temos que  $c_j^i - \pi^t a_j > 0$ , ou seja, desde que a solução é ótima segue que  $x_j^i = 0$  (variáveis não-básicas no limite inferior).

Para a prova da última afirmação, usamos o fato de  $c_j^0 - \pi^t a_j \leq 0$ . Agora, como  $c_j^i < c_j^{mj}$ , segue que  $c_j^i - \pi^t a_j < 0$ . Ou seja, as demais componentes de  $x_j$  são não-básicas no limite superior,  $x_j^i = \beta_j^{i+1} - \beta_j^i$ .

O método simplex pode ser especializado ao problema (89.3) percorrendo apenas soluções básicas que satisfaçam a propriedade 1. A figura 3 mostra, por exemplo, que se  $x_j^k$  for uma variável básica e atingir seu limite superior com a alteração de uma variável não-básica, então será trocada por  $x_j^{k+1}$  (sem alterar a base, pois ambas as variáveis têm os mesmos coeficientes, a coluna  $a_j$ ). Isto significa que se  $x_j$  ultrapassar o ponto-de-quebra  $\beta_j^k$ , então a *variável-componente*  $x_j^k$  assume seu limite superior e  $x_j^{k+1}$  (antes no limite inferior) assume valores positivos.



Como o custo de  $x_j^{k+1}$  é diferente do custo de  $x_j^k$ , então o custo relativo da variável não-básica deve ser recalculado. Isto corresponde no problema original (88) a não se alterar a direção de busca, porém um ponto de não-diferenciabilidade foi ultrapassado. Entretanto no programa linear equivalente (89) a direção de busca é alterada da seguinte forma: Para exemplificar, suponha por simplicidade que  $x_1^1$  seja uma variável básica e atinja seu limite superior e seja trocada por  $x_1^2$  na base. Embora a base não tenha sido alterada, a direção de busca no problema (89.3) que antes era:  $(\underbrace{d_1, 0, \dots, 0}_{\text{comp. de } x_1}, \dots)$  passa a ser:

$(\underbrace{0, d_1, \dots, 0}_{\text{comp. de } x_1}, \dots)$ , onde  $d_1$  é a coordenada de direção simplex. Esta especialização do método

simplex é equivalente ao método simplex desenvolvido diretamente ao problema (88), percorrendo soluções básicas de acordo com a definição 1. Veremos, entretanto, que isto não será possível aos métodos de pontos interiores.

**Propriedade 2.** Sejam  $\tilde{x} = (x_1^1, x_1^2 \dots x_1^{m_1}, x_2^1, x_2^2 \dots x_2^{m_2}, x_n^1 \dots x_n^{m_n})$  uma solução ótima do problema (89.3) e  $x$  calculado por (89.1). Então a condição (89.2) é necessariamente verificada.

Prova - Suponha que (89.2) não seja verificada. Temos então três possibilidades:

i) existe uma variável  $x_j$  com componentes  $x_j^r$  e  $x_j^s$  tais que

$$x_j^r \in (0, \beta_j^{r+1} - \beta_j^r)$$

$$x_j^s \in (0, \beta_j^{s+1} - \beta_j^s).$$

ii) existe uma variável  $x_j$  com componentes

$$x_j^k \in (0, \beta_j^{k+1} - \beta_j^k)$$

e

$$x_j^r = 0, \text{ para } r < k.$$

iii) existe uma variável  $x_j$  com componentes

$$x_j^k \in (0, \beta_j^{k+1} - \beta_j^k)$$

e

$$x_j^s = \beta_j^{s+1} - \beta_j^s, \text{ para } s > k.$$

Se i) ocorre, então observe que  $\tilde{\mathbf{d}}$  tal que  $d_j^r = -d_j^s = 1$  e zero para as demais componentes, satisfaz  $\tilde{\mathbf{A}}\tilde{\mathbf{d}} = \mathbf{0}$ , e portanto é uma direção factível, pois  $x_j^r$  e  $x_j^s$  não estão em seus limites.

Seja agora  $\tilde{\mathbf{c}}$  o vetor de custos de (89.3). Então

$$\tilde{\mathbf{c}}^t \tilde{\mathbf{d}} = c_j^r - c_j^s < 0,$$

supondo  $s > r$ , sem perda de generalidade. Assim  $\tilde{\mathbf{d}}$  é uma direção factível e de descida e, portanto, a solução não é ótima. Absurdo, pois por hipótese supomos a solução ótima.

Se ii) ocorre, considere  $\tilde{\mathbf{d}}$  tal que  $d_j^k = -d_j^r = -1$ .

Como  $\tilde{\mathbf{A}}\tilde{\mathbf{d}} = \mathbf{0}$  e  $d_j^f = 1$  com  $x_j^f = 0$  (limite inferior), a direção escolhida é factível. Além disso,  $\tilde{\mathbf{c}}^t\tilde{\mathbf{d}} = -c_j^k + c_j^f < 0$ , contrariando a otimalidade. Analogamente a prova é feita se iii) ocorre.

A propriedade 2 mostra que a condição (89.2) pode ser abandonada, sem perda de generalidade.

Com intuito de tornar clara a diferença da abordagem aqui adotada, apresentamos a seguir, de uma forma breve, um algoritmo simplex linear por partes para resolver (88).

### 3.3 Método Simplex Linear por Partes

Um teorema fundamental da programação afirma que se um problema de programação linear tem solução ótima, então existe uma solução básica factível ótima. Este resultado pode ser estendido para programação linear por partes, onde agora uma solução básica factível é uma solução factível obtida a partir de uma partição,  $\mathbf{A} = (\mathbf{B}, \mathbf{N})$ , com  $\mathbf{B} \in \mathcal{R}^{m \times m}$  e  $\text{posto}(\mathbf{B}) = m$ , e as variáveis não-básicas assumindo valores iguais aos pontos de quebra. Assim  $\hat{\mathbf{x}} = (\hat{\mathbf{x}}_{\mathbf{B}}, \hat{\mathbf{x}}_{\mathbf{N}})$  será uma solução básica factível se

$$\hat{\mathbf{x}}_{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\hat{\mathbf{x}}_{\mathbf{N}}, \quad (\text{variáveis básicas}) \quad (90)$$

onde

$$\hat{x}_j = \beta_j^{k_j}, \quad (\text{variáveis não-básicas}) \quad (91)$$

para  $j \in \mathbf{N}$  e algum  $0 \leq k_j \leq m_j$ ,

com

$$\beta_j^0 \leq \hat{x}_j \leq \beta_j^{m_j}, j \in \mathbf{B}.$$

Por simplicidade de notação assumiremos que  $\beta_j^0 = 0$  e faremos  $\beta_j^{m_j+1} = \infty$ .

Notemos ainda que uma mesma base  $\mathbf{B}$  pode dar origem a muitas soluções básicas primais, bastando para isto que redefinamos os valores em (91) e quando  $\hat{\mathbf{x}}_{\mathbf{B}} \geq \mathbf{0}$ , diremos que a solução é primal factível.

Como no caso linear, definimos uma solução básica dual associada à  $\mathbf{B}$

$$(\hat{\lambda})^t = (\mathbf{c}_{\mathbf{B}})^t \mathbf{B}^{-1} \quad (92)$$

com  $\mathbf{c}_{\mathbf{B}}$  sendo o chamado vetor dos "custos" das variáveis  $\mathbf{x}_{\mathbf{B}} = \hat{\mathbf{x}}_{\mathbf{B}}$ . Podemos notar que para cada base  $\mathbf{B}$  escolhida, existem muitas soluções duais associadas pois, uma vez alterado  $\mathbf{x}_{\mathbf{B}}$ , alteramos também  $\mathbf{c}_{\mathbf{B}}$ , diferentemente do que acontecia no caso linear, onde a factibilidade dual podia ser explicitada independentemente da solução primal. A fim de estabelecermos tal factibilidade, definimos a função lagrangiano:

$$\begin{aligned} L(\mathbf{x}, \lambda) &= f(\mathbf{x}) + \lambda^t (\mathbf{b} - \mathbf{A}\mathbf{x}) \\ &= \sum_{j=1}^n (f_j(x_j) - \lambda^t \mathbf{A}^j x_j) + \lambda^t \mathbf{b}, \\ &= \sum_{j=1}^n L_j(x_j, \lambda) + \lambda^t \mathbf{b} \end{aligned} \quad (93)$$

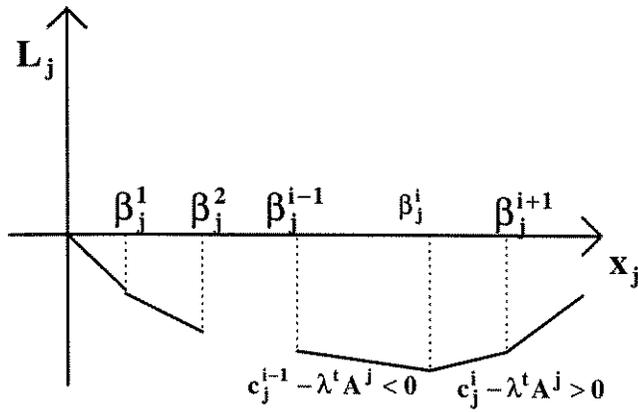
onde  $L_j(x_j, \lambda) = f_j(x_j) - \lambda^t \mathbf{A}^j x_j$ ,

sendo  $\lambda$  é o vetor das variáveis duais.

e

$$c_j^k - \lambda^t A^j > 0, \quad k = i, i+1, \dots, m_j.$$

Graficamente teremos

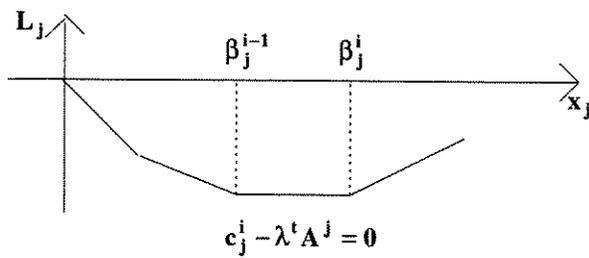


Assim a solução do subproblema  $j$  é

$$x_j^* = \beta_j^i.$$

Agora se  $\lambda^t A^j = c_j^i$ , a solução será, como pode ser visto no gráfico abaixo,

$$\beta_j^i \leq x_j^* \leq \beta_j^{i+1}.$$



Entretanto devemos exigir ainda que  $\lambda^t \mathbf{A}^j \leq c_j^{mj}$  pois, caso contrário, o mínimo em (94) não existiria, uma vez que supomos que  $x_j \geq 0$ . Podemos então dizer que vetor das variáveis duais deve ser tal que:

$$\lambda^t \mathbf{A} \leq (c_1^{m1}, c_2^{m2}, \dots, c_n^{mn}) = \mathbf{c}_{\max}.$$

e assim a função dual  $\phi$  pode ser avaliada tomando-se  $x_j^*$  por:

$$x_j^* = \beta_j^i \quad \text{se} \quad c_j^{i-1} < \lambda^t \mathbf{A}^j < c_j^i, \quad (96)$$

$$\beta_j^i \leq x_j^* \leq \beta_j^{i+1} \quad \text{se} \quad \lambda^t \mathbf{A}^j = c_j^i. \quad (97)$$

Portanto, se temos uma base  $\mathbf{B}$  e uma solução básica primal factível dada por (90) e (91),

diremos que  $\hat{\lambda} = \mathbf{c}_B^t \mathbf{B}^{-1}$  será dual factível se :

$$c_j^{ij-1} \leq (\hat{\lambda})^t \mathbf{A}^j \leq c_j^{ij}, \quad \forall j \in \mathbf{N}, \quad (98)$$

ou seja, a solução (90) - (91) resolve o problema lagrangeano  $\phi(\lambda)$ .

Finalmente, considerando que  $\hat{\mathbf{x}}$  seja uma solução básica primal factível,  $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$ ,

dada por (90) e (91), e tomando  $\hat{\lambda}$  como em (92), verificando (98), teremos:

$$\phi(\hat{\lambda}) = \underset{\mathbf{x} \geq 0}{\text{Min}} L(\mathbf{x}, \hat{\lambda}) = L(\hat{\mathbf{x}}, \hat{\lambda}) = f(\hat{\mathbf{x}}) \quad (99)$$

Com o que foi considerado acima podemos enunciar:

**Teorema:** Consideremos as soluções básicas primal e dual dadas por (90), (91), (92). Se  $\hat{\mathbf{x}}$  for primal factível e  $\hat{\lambda}$  for dual factível então a solução  $\hat{\mathbf{x}}$  é ótima para o problema primal e  $\hat{\lambda}$  resolve o problema dual:

$$\begin{aligned} \text{Max } \phi(\lambda) \\ \lambda^t \mathbf{A} \leq \mathbf{c}_{\max} \end{aligned}$$

Observemos que se  $\hat{\lambda}$  não for dual factível, então  $\hat{\mathbf{x}}$  não resolve o problema lagrangiano e

$$\phi(\hat{\lambda}) = \text{Min}_{\mathbf{x} \geq 0} L(\mathbf{x}, \hat{\lambda}) \leq L(\hat{\mathbf{x}}, \hat{\lambda}) = f(\hat{\mathbf{x}})$$

e assim podemos ainda diminuir o valor de  $f(\mathbf{x})$ .

Vejamus então como elaborar um algoritmo simplex linear por partes para resolver (88).

Vamos assumir que tenhamos uma solução básica primal factível  $\hat{\mathbf{x}}$ , mas que  $\hat{\lambda}$  não seja dual factível. Então sabemos que existe  $r \in N$  com  $\hat{x}_r = \beta_r^{k_r}$  e

$$\text{i) } \hat{c}_r^+ = c_r^{k_r} - (\hat{\lambda})^t \mathbf{A}^r < 0$$

ou (100)

$$\text{ii) } \hat{c}_r^- = -c_r^{k_r-1} + (\hat{\lambda})^t \mathbf{A}^r < 0 .$$

Considerando o primeiro caso, temos que  $\hat{x}_r = \beta_r^{k_r}$  não resolve (90) para  $\lambda = \hat{\lambda}$ , então modifiquemos  $\hat{\mathbf{x}}$  segundo a regra simplex

$$\mathbf{x}_N = \hat{\mathbf{x}}_N + \varepsilon \mathbf{e}_r$$

com  $\mathbf{e}_r$  sendo a  $r$ -ésima coluna da matriz identidade  $(n - m)$ .

Esta perturbação nas variáveis não - básicas produz nas variáveis básicas a perturbação

$$\mathbf{x}_B = \hat{\mathbf{x}}_B + \varepsilon \mathbf{d}_B,$$

onde

$$\mathbf{d}_B = -\mathbf{B}^{-1} \mathbf{N}^r, \text{ e } \mathbf{N}^r \text{ é a } r\text{-ésima coluna de } \mathbf{N}.$$

Desta forma a nova solução é

$$\mathbf{x}(\varepsilon) = \hat{\mathbf{x}} + \varepsilon \mathbf{d},$$

com

$$\mathbf{d} = \begin{pmatrix} -\mathbf{B}^{-1} \mathbf{N}^r \\ \mathbf{e}_r \end{pmatrix},$$

sendo a direção simplex.

Notemos que

$$\nabla f(\hat{\mathbf{x}}) \cdot \mathbf{d} = \begin{pmatrix} \mathbf{c}_B^t & \mathbf{c}_N^t \end{pmatrix} \begin{pmatrix} -\mathbf{B}^{-1} \mathbf{N}^r \\ \mathbf{e}_r \end{pmatrix} = c_r^{k_r} - (\hat{\lambda})^t \mathbf{A}^r < 0,$$

com a desigualdade vindo de (100), e ainda que com a perturbação simplex utilizada, o

“custo” de  $x_r$  é considerado  $c_r^{k_r}$ , uma vez que  $x_r(\varepsilon) \in [b_r^{k_r+1}, b_r^{k_r}]$ .

### **Cálculo de um Limitante para $\varepsilon$ :**

Resolveremos o seguinte problema unidimensional:

$$\text{Min}_{\varepsilon \geq 0} f(\hat{\mathbf{x}} + \varepsilon \mathbf{d}) = \sum_{j=1}^n f_j(\hat{x}_j + \varepsilon d_j)$$

Tomemos  $\varepsilon \geq 0$ ,  $\varepsilon \leq \beta_r^{k_r+1} - \beta_r^{k_r}$ , de modo que

$$(\hat{x}_j + \varepsilon d_j) \in [b_j^{k_j-1}, b_j^{k_j}], \forall j \text{ básico.} \quad (101)$$

Então

$$\begin{aligned} f(\mathbf{x}) &= \sum_{j=1}^n f_j(\hat{x}_j + \varepsilon d_j) = \sum_{j=1}^n c_j^{k_j-1} (\hat{x}_j + \varepsilon d_j - \beta_j^{k_j-1}) + f_j(\beta_j^{k_j-1}) \\ &= \left( \sum_{j=1}^n c_j^{k_j-1} (\hat{x}_j - \beta_j^{k_j-1}) + f_j(\beta_j^{k_j-1}) \right) + \varepsilon \sum_{j=1}^n c_j^{k_j-1} d_j \\ &= f(\hat{\mathbf{x}}) + \varepsilon (\nabla f(\hat{\mathbf{x}}))^t \mathbf{d}. \end{aligned}$$

Para  $\varepsilon \geq 0$ , seja  $c^0 = (\nabla f(\hat{\mathbf{x}}))^t \mathbf{d} < 0$ . Podemos observar que, quando  $\varepsilon$  cresce,  $f$  decresce.

No entanto, devido a (101), temos que:

$$\text{Se } d_j > 0 \text{ então } \hat{x}_j + \varepsilon d_j \leq \beta_j^{k_j} \text{ ou } \varepsilon \leq \frac{\beta_j^{k_j} - \hat{x}_j}{d_j}.$$

$$\text{Se } d_j < 0 \text{ teremos } \hat{x}_j + \varepsilon d_j \geq \beta_j^{k_j-1} \text{ ou } \varepsilon \leq \frac{\beta_j^{k_j-1} - \hat{x}_j}{d_j}.$$

Assim

$$0 \leq \varepsilon \leq \varepsilon_1, \text{ com } \varepsilon_1 = \text{Min}\{\varepsilon^+, \varepsilon^-, \beta_r^{k_r+1} - \beta_r^{k_r}\}$$

onde

$$\varepsilon^+ = \min\left\{\frac{\beta_j^{k_j} - \hat{x}_j}{d_j}, d_j > 0\right\},$$

$$\varepsilon^- = \min \left\{ \frac{\beta_j^{k_j-1} - \hat{x}_j}{d_j}, d_j < 0 \right\}.$$

Seja então  $\mathbf{x}^1 = \hat{\mathbf{x}} + \varepsilon_1 \mathbf{d}$ . Devemos salientar que aqui permitiremos que pontos de não-diferenciabilidade sejam ultrapassados, enquanto continuarmos a ter em  $\mathbf{d}$  uma direção de descida, o que implicará, evidentemente, numa atualização dos "custos". Então tomando  $\varepsilon > \varepsilon_1$ , teremos:

a) Para  $s$  tal que  $\varepsilon_1 = \varepsilon^+ = \frac{\beta_s^{k_s} - \hat{x}_s}{d_s}$

$$\mathbf{x}_s(\varepsilon) = (\hat{x}_s + \varepsilon d_s) \in [\beta_s^{k_s}, \beta_s^{k_s+1}]$$

e

$$\mathbf{x}_j(\varepsilon) \in [\beta_j^{k_j-1}, \beta_j^{k_j}], \text{ para todo } j \neq s, j \text{ básico.}$$

Observemos ainda que  $s_s^1 = \mathbf{x}_s(\varepsilon_1) = \beta_s^{k_s}$ .

Fazendo  $\varepsilon = \varepsilon_1 + \delta$ ,  $\delta \geq 0$ , suficientemente pequeno, vem

$$\begin{aligned} f(\mathbf{x}) &= \sum_{j=1}^n f_j(\hat{x}_j + \varepsilon d_j) = \sum_{j=1}^n c_j^{k_j-1} (\hat{x}_j + \varepsilon d_j) + f_j(\beta_j^{k_j-1}) \\ &= \sum_{\substack{j=1 \\ j \neq s}}^n c_j^{k_j-1} (\hat{x}_j + (\varepsilon_1 + \delta) d_j) + f_j(\beta_j^{k_j-1}) + c_s^{k_s} (\hat{x}_s + (\varepsilon_1 + \delta) d_s) + f_s(\beta_s^{k_s}) \\ &= \sum_{\substack{j=1 \\ j \neq s}}^n c_j^{k_j-1} x_j^1 + f_j(\beta_j^{k_j-1}) + c_s^{k_s} x_s^1 + f_s(\beta_s^{k_s}) + \delta \left[ \sum_{j \neq s}^n c_j^{k_j-1} d_j + c_s^{k_s} d_s \right] \end{aligned}$$

$$= \sum_{j=1}^n f_j(x_j^1) + \delta (\nabla f(\mathbf{x}^1))^t \mathbf{d}$$

$$\mathbf{c}^t(\mathbf{x}^1) = \mathbf{c}^t(\hat{\mathbf{x}}) + (0 \dots c_k^{i_k} - c_k^{i_k-1} \dots 0)^t.$$

Como

$$\nabla f(\mathbf{x}^1) = \nabla f(\hat{\mathbf{x}}) + (0 \dots c_s^{k_s} - c_s^{k_s-1} \dots 0)^t$$

vem que

$$c^1 = \nabla f(\mathbf{x}^1) \cdot \mathbf{d} = c^0 + (c_s^{k_s} - c_s^{k_s-1}) \cdot d_s > c^0 \quad (102)$$

pois em (102) temos

$$c_s^{k_s} - c_s^{k_s-1} > 0 \text{ e } d_s > 0.$$

Se  $c^1 < 0$ , então a direção  $\mathbf{d}$  ainda é de descida em  $\mathbf{x}^1$  e o intervalo de  $x_s$  é atualizado fazendo  $k_s \leftarrow k_s + 1$ . Caso contrário, a otimalidade é testada e, se for o caso, uma nova direção é calculada. Nesse momento a variável  $x_r$  torna-se básica e  $x_s$  deixa a base.

b) Para  $s$  tal que  $\varepsilon_1 = \varepsilon^- = \frac{\beta_s^{k_s-1} - \hat{x}_s}{d_s}$ ,

o desenvolvimento é análogo ao anterior e no final teremos

$$f(\mathbf{x}) = f(\mathbf{x}^1) + \delta \left( c^0 + (c_s^{k_s-1} - c_s^{k_s}) d_s \right),$$

onde agora  $c_s^{k_s-1} - c_s^{k_s} < 0$  e  $d_s < 0$  e toda argumentação se repete, com o intervalo de  $x_s$  sendo atualizado por  $k_s \leftarrow k_s - 1$ .

c) Caso  $\varepsilon_1 = \beta_r^{k_r+1} - \beta_r^{k_r}$ , os coeficientes das variáveis básicas  $\mathbf{c}_B$  ficam inalterados de modo que

$$\begin{aligned} c^1 &= \nabla f(\mathbf{x}^1) \cdot \mathbf{d} = \mathbf{c}_r^{k_r+1} - (\hat{\lambda})^t \mathbf{A}^r \\ &= (\mathbf{c}_r^{k_r+1} - \mathbf{c}_r^{k_r}) + [\mathbf{c}_r^{k_r} - (\hat{\lambda})^t \mathbf{A}^r] \\ &= \mathbf{c}^0 + (\mathbf{c}_r^{k_r+1} - \mathbf{c}_r^{k_r}). \end{aligned}$$

Se  $c^1 < 0$ , a direção  $\mathbf{d}$  é ainda de descida em  $\mathbf{x}^1$ , o intervalo de  $x_r$  é atualizado por  $k_r \leftarrow k_r + 1$ . Se  $c^1 > 0$ , a otimalidade é testada e, se for o caso, uma nova direção é calculada, porém sem troca de base ( a variável  $x_r$  continua não-básica, mas num outro ponto de quebra).

Encerrando esta seção, apresentamos um algoritmo simplex linear por partes.

#### **Algoritmo 4**

Passo1- Dado uma solução inicial primal factível  $\hat{\mathbf{x}}$  verifica-se se  $\hat{\lambda}$  é dual factível. Se sim, pare.

Passo2- Cálculo da direção de descida  $\mathbf{d}$  e atualização de  $\hat{\mathbf{x}}$ .

Passo3- Busca unidimensional: Cálculo dos limitantes para  $\varepsilon$ , nova atualização de  $\hat{\mathbf{x}}$  e volta ao passo1.

## Capítulo 4

### Algoritmo Linear Por Partes Via Pontos Interiores

#### 4.1 Introdução

Passaremos a desenvolver um algoritmo para resolver (88), sem no entanto transformá-lo em um problema linear ( veja seção 3.2) e nem fazer uso de alguma variante do método simplex. Desenvolveremos uma estratégia de pontos interiores para resolver o problema não linear dado, considerando técnicas de pontos interiores utilizadas em programação linear.

Tomemos então  $\mathbf{x}^0$  uma solução inicial, com  $\mathbf{Ax}^0 = \mathbf{b}$  e  $\beta_j^{i_j-1} < x_j < \beta_j^{i_j}$ ,  $1 \leq j \leq n$ . Quando restringimos as soluções tais que  $\beta_j^{i_j-1} < x_j < \beta_j^{i_j}$ , temos um problema de programação linear canalizado. Defrontamo-nos assim com a chamada fronteira virtual; que será uma fronteira de fato se  $i_j - 1 = 0$  ou  $i_j = m_j$ , pois então teremos  $x_j \geq \beta_j^0$  ou  $x_j \leq \beta_j^{m_j}$ , respectivamente. Passa a fazer sentido pensarmos na aplicação de um método de barreiras, como aqueles aplicados no capítulo 1, seção 1.3, com o qual encontramos uma direção de descida. Achada a direção de movimento, esquecemos a fronteira virtual e fazemos uma busca unidimensional, obtendo o mínimo da função objetivo nesta direção. O importante é que enquanto caminhamos na direção escolhida, podemos ultrapassar regiões de não-diferenciabilidade. No cálculo da direção de descida, utilizaremos o método desenvolvido por Sherali et al, na seção 1.3.

## 4.2 A direção de busca e um critério de parada

Seja  $\mathbf{x}^k$  um ponto interior linear por partes e  $i_j$ ,  $j=1,2,\dots,n$ , conforme a definição 2. Considere o seguinte programa linear:

$$\begin{aligned} \min f(\mathbf{x}) &= (\mathbf{c}^k)^t \mathbf{x}, \\ \text{s.a. } \mathbf{Ax} &= \mathbf{b} \\ \beta_j^{i_j-1} &\leq x_j \leq \beta_j^{i_j}, \quad j = 1,2,\dots,n, \end{aligned} \tag{103}$$

onde

$$\mathbf{c}^k = \nabla f(\mathbf{x}^k) = (\mathbf{c}_1^{i_1}, \mathbf{c}_2^{i_2}, \dots, \mathbf{c}_n^{i_n})^t.$$

Por simplicidade de notação, efetuamos a mudança de variável

$$x_j \leftarrow \frac{2x_j - (\beta_j^{i_j-1} + \beta_j^{i_j})}{(\beta_j^{i_j} - \beta_j^{i_j-1})}, \quad j = 1,2,\dots,n, \tag{104}$$

com a qual (103) fica:

$$\begin{aligned} \min f(\mathbf{x}) &= (\mathbf{c}^k)^t \mathbf{x}, \\ \text{s.a. } \mathbf{Ax} &= \mathbf{b} \\ -\mathbf{e} &\leq \mathbf{x} \leq \mathbf{e}, \end{aligned} \tag{105}$$

onde  $\mathbf{e} = (1 \ 1 \ \dots \ 1)^t$ .

Temos então que (105) é um problema linear canalizado como aquele apresentado na seção 1.5. A fim de evitarmos idas e vindas à citada seção, repetiremos algumas das deduções lá feitas.

Com a finalidade de nos mantermos longe da "fronteira", que em (105) não corresponde à fronteira do problema original, incorporamos à função objetivo uma função barreira obtendo:

$$\min (\mathbf{c}^k)^t \mathbf{x} - \mu \sum_{i=1}^n \ln(1 - x_i) - \mu \sum_{i=1}^n \ln(1 + x_i) \quad (106.1)$$

$$\text{s. a. } \mathbf{Ax} = \mathbf{b}$$

$$\text{s. a. } \mathbf{Ax} = \mathbf{b}. \quad (106.2)$$

Observamos que quando aplicamos as condições de primeira ordem ao lagrangeano de (106), obtemos:

$$\mathbf{c}^k - \mathbf{A}^t \lambda - \mu \mathbf{D}^{-1} \mathbf{v}(\mathbf{x}) = \mathbf{0} \quad (107.1)$$

$$\mathbf{Ax} = \mathbf{b} , \quad (107.2)$$

com

$$\mathbf{D} = \text{diag}(d_i), \text{ onde } d_i = \text{Min}\{1 + x_i, 1 - x_i\}$$

e

$$v_i(x_i) = \frac{2d_i x_i}{1 - x_i^2}, \quad i = 1, 2, \dots, n.$$

Se optássemos pela estratégia, que é bastante usual, de se aplicar um passo de Newton ao sistema não linear (107), com o intuito de se encontrar uma direção de descida, gerariamos um algoritmo para resolver (88) ( ver capítulo1, seção1.3). Nossa estratégia foi optar pela abordagem adotada por Sherali et al[20], que tenta resolver (107) usando quadrados mínimos. Mais ainda, poderíamos deduzir o algoritmo a ser proposto a partir de (106), mas com o intuito de torná-lo mais claro, efetuamos a mudança de escala em (105),  $\mathbf{x} = \mathbf{Dy}$ , para obtermos

$$\begin{aligned}
& \text{Min } (\mathbf{Dc}^k)^t \mathbf{y} \\
& \text{s. a. } \mathbf{ADy} = \mathbf{b} \\
& -\mathbf{1} \leq \mathbf{Dy} \leq \mathbf{1},
\end{aligned} \tag{108}$$

Incorporando uma função barreira à função objetivo em (106), obtemos o problema equivalente

$$\begin{aligned}
& \text{Min } (\mathbf{Dc}^k)^t \mathbf{y} - \mu \sum_{i=1}^n \ln(1 - d_i^2 y_i^2) = (\mathbf{Dc}^k)^t \mathbf{y} - \mu \sum_{i=1}^n \ln(1 + d_i y_i) - \mu \sum_{i=1}^n \ln(1 - d_i y_i) \\
& \text{s. a. } \mathbf{ADy} = \mathbf{b}
\end{aligned} \tag{109}$$

As condições de Karush-Kuhn-Tucker aplicadas no problema não-linear (109), nos leva a buscarmos  $\lambda$ , dual factível, e  $\mathbf{x}$ , primal factível, que satisfaçam o sistema não-linear:

$$\mathbf{Dc}^k - \mu \mathbf{Y} \mathbf{e} - (\mathbf{AD})^t \lambda = \mathbf{0} \tag{110}$$

$$\mathbf{ADy} = \mathbf{b} \tag{111}$$

onde  $\mathbf{Y}$  é a matriz diagonal, com cada elemento da diagonal dado por:

$$Y_i = \frac{2d_i^2 y_i}{(1 - d_i^2 y_i^2)}$$

De (110) segue que

$$\mathbf{DA}^t \lambda = \mathbf{Dc}^k - \mu \mathbf{Y} \mathbf{e} \tag{112}$$

Multiplicando (112) por  $\mathbf{AD}$ , vem

$$\mathbf{AD}^2 \mathbf{A}^t \lambda = \mathbf{AD}^2 \mathbf{c}^k - \mu \mathbf{ADY} \mathbf{e}$$

Assim, podemos tomar

$$\lambda = (\mathbf{AD}^2 \mathbf{A}^t)^{-1} \mathbf{AD}(\mathbf{Dc}^k - \mu \mathbf{Y} \mathbf{e}),$$

Se considerarmos que  $\mathbf{y} = \mathbf{D}^{-1}\mathbf{x}$ , teremos

$$\lambda = (\mathbf{AD}^2\mathbf{A}^t)^{-1}\mathbf{AD}(\mathbf{Dc}^k - \mu\mathbf{v}(\mathbf{x})). \quad (113)$$

Assim, dado uma solução primal factível  $\mathbf{x}^k$ , a solução quadrados mínimos para (112), dada pela equação (113), fornecerá uma estimativa para o multiplicador dual.

Considerando o gradiente do lagrangeano de (106),  $\mathbf{g}_k = \mathbf{c}^k - \mathbf{A}^t\lambda_k - \mu\mathbf{D}_k^{-1}\mathbf{v}(\mathbf{x}_k)$ , observamos que

$$\mathbf{p}_k = \mathbf{D}_k\mathbf{g}_k = \left[ \mathbf{I} - (\mathbf{AD}_k)^t (\mathbf{AD}_k^2\mathbf{A}^t)^{-1} \mathbf{AD}_k \right] [\mathbf{D}_k\mathbf{c} - \mu\mathbf{v}(\mathbf{x}_k)] = \mathbf{P}_k [\mathbf{D}_k\mathbf{c} - \mu\mathbf{v}(\mathbf{x}_k)],$$

sendo  $\mathbf{P}_k = \left[ \mathbf{I} - (\mathbf{AD}_k)^t (\mathbf{AD}_k^2\mathbf{A}^t)^{-1} \mathbf{AD}_k \right]$  a matriz de projeção sobre o núcleo  $\mathbf{AD}_k$ .

Assim,  $\mathbf{p}_k = \mathbf{D}_k\mathbf{g}_k$  é a projeção do gradiente da função objetivo (8.1), com mudança de escala  $\mathbf{x} \leftarrow \mathbf{D}^{-1}\mathbf{x}$ , no núcleo de  $\mathbf{AD}_k$ . Então

$$\mathbf{d}^k = -\mathbf{D}_k\mathbf{g}_k$$

é a direção factível de descida escolhida em  $\mathbf{x}^k$  e a nova solução será calculada por

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \varepsilon_k\mathbf{d}^k, \quad (114)$$

onde  $\varepsilon_k$  é o passo a ser determinado.

Sherali et al[20] utilizaram como critério de parada  $\|\mathbf{d}^k\| < \varepsilon$ , com  $\varepsilon > 0$  e pequeno. Apesar de não termos, em geral, um problema canalizado, o uso do mesmo critério foi aqui utilizado com sucesso.

### 4.3 Busca unidimensional linear por partes

A busca unidimensional linear por partes desempenha um papel fundamental no algoritmo a ser proposto, pois, como veremos adiante, é ela quem garante, nas primeiras iterações uma rápida aproximação do ótimo.

Considere inicialmente  $\mathbf{y}^1 = \mathbf{x}^k$  e faça  $r=1$  (uma sequência de pontos intermediários  $\mathbf{y}^r$  será gerada).

Determinamos  $\alpha$ , o maior valor tal que:  $f(\mathbf{y}^r + \varepsilon \mathbf{d}^k)$  seja linear,  $0 \leq \varepsilon \leq \alpha$ . Observe que  $\mathbf{c}^k \mathbf{d}^k < 0$  e, portanto, a função é decrescente neste intervalo.

Sejam  $\mathbf{y} = \mathbf{y}^r + \varepsilon \mathbf{d}^k$ , onde  $\varepsilon$  é um valor ligeiramente superior a  $\alpha$  e  $\nabla f(\mathbf{y}) \mathbf{d}^k$  a derivada direcional em  $\mathbf{y}$  na direção  $\mathbf{d}^k$ . Se  $\nabla f(\mathbf{y}) \mathbf{d}^k \geq 0$ , então caminhar além do primeiro ponto-de-quebra é desaconselhável, pois, naquela região,  $\mathbf{d}^k$  já não é mais uma direção de descida. Assim tomamos  $\varepsilon_k = 0,985\alpha$ , onde  $\alpha$  é o tamanho do passo para atingir o primeiro ponto-de-quebra. O novo ponto é definido  $\mathbf{x}^{k+1} = \mathbf{y}^r + \varepsilon_k \mathbf{d}^k$ , onde o vetor gradiente é mantido, porém uma nova direção deve ser calculada. Se  $\nabla f(\mathbf{y}) \mathbf{d}^k < 0$ , então ultrapassamos o ponto de não-diferenciabilidade e uma *iteração intermediária* é computada:  $\mathbf{y}^{r+1} = \mathbf{y}^r + 1,002\alpha \mathbf{d}^k$ ,  $r=r+1$ . Atualizamos o gradiente e com a mesma direção, a análise é repetida.

Podemos enunciar agora um algoritmo linear por partes interior para a solução do problema proposto.

#### 4.4 Algoritmo linear por partes interior (Algoritmo 5)

Considere uma solução factível inicial  $\mathbf{x}^0$ , com  $\beta_j^{i_j-1} < x_j^0 < \beta_j^{i_j}$ , faça a mudança de variável proposta acima, para obter  $-\mathbf{e} < \mathbf{x}^0 < \mathbf{e}$ , e tome  $k = 0$ .

Passo 1) *Determinação da direção*

Seja  $\mathbf{D}_k = \text{diag}\{\mathbf{d}_1^k, \mathbf{d}_2^k, \dots, \mathbf{d}_n^k\}$ ,  $d_i^k = \min\{1 - x_i^k, 1 + x_i^k\}$ . Calcule:

$$\lambda_k = (\mathbf{A}\mathbf{D}_k^2\mathbf{A}^t)^{-1} \mathbf{A}\mathbf{D}_k(\mathbf{D}_k\mathbf{c}^k - \mu\mathbf{v}(\mathbf{x}_k)) \{\text{estimativa dual}\}$$

$$\mathbf{g}_k = \mathbf{c}^k - \mathbf{A}^t\lambda_k - \mu\mathbf{D}_k^{-1}\mathbf{v}(\mathbf{x}_k) \{\text{gradiente do lagrangiano}\}$$

$$\mathbf{d}^k = -\mathbf{D}_k\mathbf{g}_k \{\text{direção de busca}\}$$

Passo 2) *Critério de parada*

Se  $\|\mathbf{d}_k\| < \varepsilon$  então pare.

Passo 3) *Busca linear por partes e cálculo da nova iteração:*

Aplique o algoritmo da seção 4.3, retornando com  $\mathbf{x}^{k+1} = \mathbf{x}^k + \varepsilon_k\mathbf{d}^k$ .

Faça  $k = k+1$  e volte ao passo 1.

Observações :

1ª) Um cuidado muito grande deve ser tomado na escolha do escalar  $\varepsilon_k$ , em (114), pois ele deverá refletir algumas tomadas de decisão. Vejamos:

i) Na busca linear por partes feita no passo 3, para  $j = 1, \dots, n$ , calculamos

$$\alpha_j^{i_j} = \frac{\beta_j^{i_j} - x_j^k}{d_j^k}, \text{ se } d_j^k > 0$$

e

$$\alpha_j^{i_j} = \frac{\beta_j^{i_j-1} - x_j^k}{d_j^k}, \text{ se } d_j^k < 0$$

e tomamos

$$\alpha_{j^*}^{i_{j^*}} = \text{Min}\{\alpha_j^{i_j}\}.$$

( $j^*$  é o índice da primeira variável que muda de intervalo).

ii) Na atualização do gradiente adotamos o seguinte procedimento:

Se  $d_{j^*}^k > 0$  então

$$\nabla f \leftarrow \nabla f + \left( c_{j^*}^{i_{j^*}+1} - c_{j^*}^{i_{j^*}} \right) \mathbf{e}_{j^*}.$$

Se  $d_{j^*}^k < 0$  então

$$\nabla f \leftarrow \nabla f + \left( c_{j^*}^{i_{j^*}-1} - c_{j^*}^{i_{j^*}} \right) \mathbf{e}_{j^*},$$

$$\text{com } \mathbf{e}_{j^*} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{posição } j^* .$$

iii) Após atualizarmos o gradiente, que é processado antes da atualização da solução corrente, devemos verificar se o vetor direção  $\mathbf{d}$  continua a ser de descida e agora sim, em qualquer caso, atualizamos da solução corrente. Vejamos.

Se  $\nabla \mathbf{f} \cdot \mathbf{d}^k \geq 0$  então a busca unidimensional é encerrada e o passo ótimo é  $\alpha_{j^*}^{i,j^*}$ .

Tal passo, entretanto, nos leva a um ponto de quebra ( uma fronteira virtual). Fazemos então:

$$\mathbf{x}^k \leftarrow \mathbf{x}^k + \varepsilon_k \mathbf{d}^k ,$$

com

$$\varepsilon_k = 0.985 \alpha_{j^*}^{i,j^*} .$$

Agora se  $\nabla \mathbf{f} \cdot \mathbf{d}^k < 0$  então

$$\mathbf{x}^k \leftarrow \mathbf{x}^k + \varepsilon_k \mathbf{d}^k ,$$

com

$$\varepsilon_k = 1.002 \alpha_{j^*}^{i,j^*}$$

e procedemos a seguinte atualização:

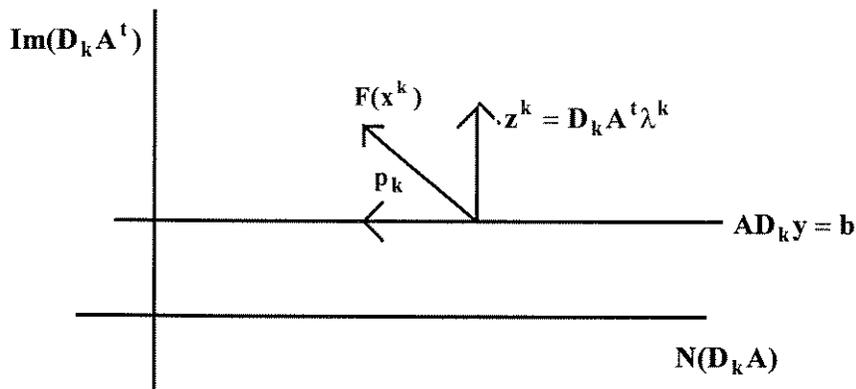
$$i_{j^*} \leftarrow i_{j^*} + 1 \text{ se } d_{j^*} > 0$$

ou

$$i_{j^*} \leftarrow i_{j^*} - 1 \text{ se } d_{j^*} < 0.$$

2ª) A escolha dos parâmetros 0,985 e 1,002 foi devida às experiências computacionais efetuadas com os exemplos analisados abaixo. A idéia de usarmos 0,985 surgiu do estudo do trabalho de Sherali et al[20], capítulo1, seção1.5, onde foi provado que para um PL canalizado, qualquer valor do parâmetro no intervalo  $[0,97,0,99]$  levaria-nos à convergência.

3ª) Geometricamente, o que buscamos com o algoritmo acima é fazer que  $F(x^k) = D_k c - \mu v(x_k)$  pertença à imagem de  $D_k A^t$ , visto que  $(AD_k)^t \lambda^k = D_k c - \mu v(x_k)$ . Isso será conseguido, como pode ser visto pelo gráfico abaixo, quando  $p_k = 0$ .



Embora desenvolvido a partir de restrições de igualdade, o algoritmo pode ser aplicado em qualquer caso, acrescentando-se variáveis artificiais e ,ou, de folga, e, mesmo

no caso de igualdade, a implementação é feita de modo que uma solução inicial sempre esteja definida. Vejamos:

#### 4.4.1 Restrições na forma de igualdade.

$$\begin{aligned} & \text{Min } \sum f_i(x_i) \\ & \text{s. a. } \mathbf{Ax} = \mathbf{b} \text{ ,} \\ & \quad \mathbf{x} \in \Omega \end{aligned}$$

$f_i$  uma função linear por partes e  $\Omega \subset \mathbb{R}^n$  o conjunto que define as fronteiras virtuais.

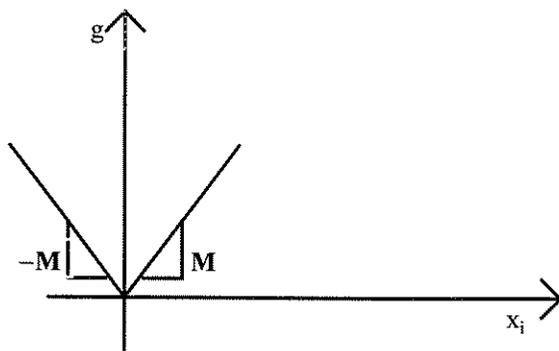
Neste caso aplicaríamos o algoritmo no problema equivalente

$$\text{Min } \sum_{i=1}^n f_i(x_i) + \sum_{i=n+1}^{n+m} g(x_i)$$

$$\begin{aligned} & \text{s. a. } \mathbf{Ax} + \mathbf{y} = \mathbf{b} \\ & \quad \mathbf{x} \in \Omega \text{ ,} \end{aligned}$$

onde  $\mathbf{y} = \begin{pmatrix} x_{n+1} \\ \vdots \\ x_{n+m} \end{pmatrix}$  e  $g(x_i) = \begin{cases} -Mx_i, & x_i < 0 \\ Mx_i, & x_i \geq 0 \end{cases}$ , para um certo  $M$  grande,

ilustrada abaixo.



#### 4.4 2 Restrições com desigualdades.

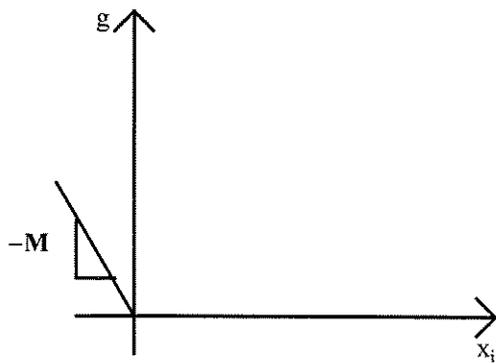
Suponhamos que queiramos resolver o seguinte problema:

$$\begin{aligned} & \text{Min } \sum f_i(x_i) \\ & \text{s.a. } \mathbf{Ax} \leq \mathbf{b} . \\ & \quad \mathbf{x} \in \Omega \end{aligned}$$

Tomamos então o problema equivalente

$$\begin{aligned} & \text{Min } \sum_{i=1}^n f_i(x_i) + \sum_{i=n+1}^{n+m} g(x_i) \\ & \text{s.a. } \mathbf{Ax} + \mathbf{y} = \mathbf{b} \\ & \quad \mathbf{x} \in \Omega , \end{aligned}$$

onde  $g(x_i) = \begin{cases} -Mx_i, & x_i < 0 \\ 0 & , x_i \geq 0 \end{cases}$  tem agora a forma abaixo.



Com a incorporação dessas funções lineares por partes  $g$ , o problema da fase 1 está superado, pois agora uma solução inicial está disponível e as penalizações embutidas nas funções  $g$  são tais que a solução final resolve também o problema original ou a existência de  $x_i$ ,  $i > n$ , com valor não nulo implica na inexistência de soluções factíveis para o problema original. Assim na implementação, uma fase 1 aparece embutida e programa fornece uma solução interior inicial toda vez que o algoritmo é inicializado.

## CAPÍTULO 5

### O funcionamento do método proposto no PL equivalente

Observamos na seção 3.2 que podemos estender um método da programação linear à programação linear por partes, pela aplicação deste ao programa linear equivalente (89.3). A estrutura particular do problema (89.3) permite um certo número de simplificações. Observamos também que o método simplex linear por partes (isto é, o método simplex desenvolvido diretamente para o problema (88)) pode ser deduzido desta forma.

Se aplicarmos um método de pontos interiores ao problema (89.3), podemos identificar através da transformação (89.1) a trajetória deste no problema original (88).

Tomemos, no espaço transformado, uma direção de descida  $\tilde{d}$ , factível em  $\hat{x}^k$ , obtida por qualquer método, dada por:

$$\tilde{d} = (d_1^0 \dots d_1^{m_1} d_2^0 \dots d_2^{m_2} \dots d_n^0 \dots d_n^{m_n})^t. \quad (115)$$

A próxima iteração será:

$$\tilde{x}^{k+1} = \tilde{x}^k + \alpha \tilde{d}. \text{ Assim, } (x_j^i)^{k+1} = (x_j^i)^k + \alpha d_j^i.$$

Lembrando que, por definição:

$$x_j^{k+1} = \sum_{i=1}^{m_j} (x_j^i)^{k+1},$$

no problema original teremos

$$x_j^{k+1} = \left[ \sum_{i=1}^{m_j} (x_j^i)^k \right] + \alpha \left[ \sum_{i=1}^{m_j} d_j^i \right] = x_j^k + \alpha d_j,$$

onde

$$d_j = \sum_{i=1}^{m_j} d_j^i \tag{116}$$

é a  $j$ -ésima coordenada da direção  $\mathbf{d}$  no espaço original. Esta relação mostra que um método no problema transformado tem seu funcionamento bem determinado no problema original (o inverso não é verdadeiro). Além disso, observamos que um ponto interior no problema (89.3) satisfaz:

$$0 < x_j^i < \beta_j^{i+1} - \beta_j^i,$$

para todo  $i, j$ , e portanto não satisfaz a condição (89.2), apesar de sabermos a priori que a maioria das componentes  $x_j^i$  estarão em seus limites na solução ótima (considerando uma solução básica linear por partes ótima, todas as componentes das variáveis não-básicas estarão em seus limites e apenas uma componente para cada variável básica será interior, a menos de degeneração). Esta maneira direta de estender um método de PL para um de PLP não nos parece adequada, pois no ótimo muitas componentes deverão estar em seus limites, evitados pelos métodos de pontos interiores, causando um grande número de iterações quando próximo de tais pontos (o efeito observado no gráfico 1 seria ampliado).

Em outras palavras as fronteiras virtuais estariam sendo consideradas de fato como fronteiras.

Podemos, entretanto, desenvolver um método de pontos interiores híbrido para o problema (89.3), que satisfaça a condição (89.2). Para tanto consideremos uma solução interior linear por partes,  $\mathbf{x}$ , (conforme a definição (2)), com

$$\beta_j^{i_{j-1}} < x_j^i < \beta_j^i, \quad j = 1, 2, \dots, n,$$

e a correspondente (única),  $\hat{\mathbf{x}}$ , no problema transformado satisfazendo a condição (89.2):

$$\begin{aligned} x_j^i &= \beta_j^i - \beta_j^{i-1}, & i &= 1, \dots, i_j - 1, \\ x_j^i &= 0, & i &= i_j + 1, \dots, n, \\ x_j^{i_j} &= x_j - \beta_j^{i_j-1} & &\in (0, \beta_j^{i_j} - \beta_j^{i_j-1}). \end{aligned} \quad (117)$$

Como podemos observar,  $\hat{\mathbf{x}}$  não é um ponto interior no problema (89.3). Uma direção factível  $\mathbf{d} = (d_1, d_2, \dots, d_n)^t$  em  $\mathbf{x}$  (problema original) tem uma correspondente direção factível em  $\hat{\mathbf{x}}$  (problema transformado) dada por:

$$d_j^i = \begin{cases} 0, & i \neq i_j, \\ d_j, & i = i_j. \end{cases} \quad (118)$$

Facilmente verificamos a seguinte propriedade:

**Propriedade 3:** Se  $\mathbf{d}$  for uma direção de descida factível em  $\mathbf{x}$ , então  $\tilde{\mathbf{d}}$ , dada por (118), será uma direção de descida factível em  $\hat{\mathbf{x}}$ , dada por (117).

Uma busca na direção  $\tilde{\mathbf{d}}$  pode fazer com que uma componente atinja um de seus limites. Por exemplo, suponha que  $x_1^1 \in (0, \beta_1^1 - \beta_1^0)$  e  $x_1^1$  atinja seu limite superior antes que as demais ( $\tilde{\mathbf{d}} = (\underbrace{d_1 0 \dots 0}_{\text{comp. } x_1}, \dots)$ ). Então

$$x_1^1 \leftarrow x_1^1 + \varepsilon d_1 = \beta_1^1 - \beta_1^0,$$

as demais componentes são recalculadas por  $\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} + \varepsilon \tilde{\mathbf{d}}$  e a direção  $\tilde{\mathbf{d}}$  torna-se inactivável para o novo ponto, necessitando ser recalculada. O cálculo de direções factíveis consiste no passo mais caro dos métodos de otimização, entretanto, a direção  $\mathbf{d}$ , no problema original, pode ainda ser factível e de descida se o ponto-de-quebra for ultrapassado (busca unidimensional linear por partes). Isto corresponde a mudar a direção  $\tilde{\mathbf{d}}$  por

$$\tilde{\mathbf{d}} \leftarrow (\underbrace{0 \ d_1 \ 0 \dots 0}_{\text{comp. } x_1}, \dots).$$

A nova direção é factível no novo ponto, e a propriedade 3 vale. Podemos repetir o processo, encontrando novo passo e etc.

Em outras palavras, o método apresentado neste trabalho corresponde a trabalhar no problema transformado com solução conforme (117), direções factíveis conforme (118) e mudanças de direções factíveis de acordo com o procedimento explicado acima. Assim, um método de pontos interiores aplicado ao problema linear equivalente, não produz o método linear por partes interior desenvolvido, pois uma solução interior no PLP pode ser vista no PL como admitindo componentes de  $x_j$  em seus limites (e portanto fronteira).

## CAPÍTULO 6

### Experimentos Computacionais

#### Exemplo 1

Vejamos como se comporta o algoritmo quando aplicado a problemas de duas variáveis, facilitando uma visão geométrica.

Considerando a função objetivo representada pela Figura 1 e as restrições dadas por

$$\begin{cases} -x_1 + x_2 \leq 2 \\ 2x_1 + x_2 \leq 8 \\ 2x_1 - x_2 \leq 4, \end{cases}$$

a Figura 4 mostra como o algoritmo procede na busca do ótimo, partindo de seis soluções iniciais, em um caso onde a solução ótima é única e se encontra no vértice (3,2) da fronteira da região de factibilidade.

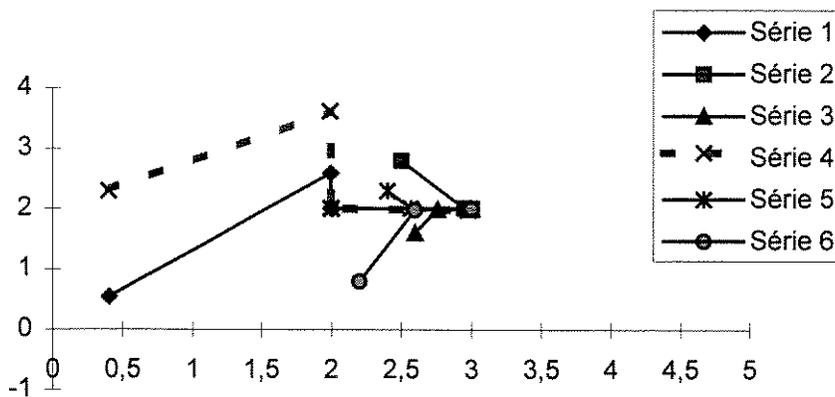


Figura 4

Na busca da prova de convergência do método acreditamos que existiria uma sub-região “atratora”, isto é, uma sub-região que, a partir de um certo número de iterações, seria atingida e então, sem abandoná-la, alcançaríamos o ótimo mas a figura acima vem contrapor tal crença.

O número de iterações intermediárias efetuadas em cada iteração principal, ou seja, quantas regiões de não-diferenciabilidade são ultrapassadas na busca linear por partes, para as séries 1, 3, 4, 6 é apresentado na figura 5. Podemos observar, examinando o gráfico devido à série 1, que partindo de uma solução inicial, o algoritmo realiza na primeira iteração principal, quatro iterações intermediárias, isto é, na busca linear por partes, passa por quatro regiões de não diferenciabilidade; na segunda iteração não abandona a face em questão; na terceira passa por duas faces e a partir de então não abandona mais a última face visitada. Já a série 4 mostra que, utilizando-se da primeira direção calculada, passa por três faces, muda a direção e caminha por outras duas faces, atualiza novamente a direção, mas permanece na mesma face, uma nova atualização é feita no vetor direção, com o qual passa por três faces e a partir de então permanece na última face visitada, até atingir o ótimo.

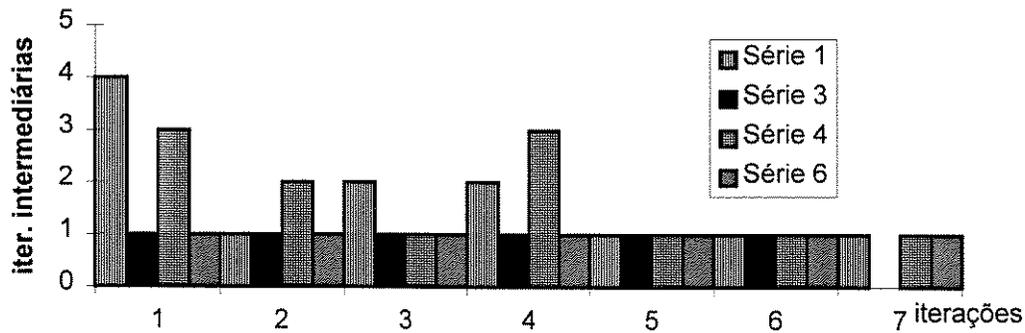


Figura 5

Apenas para efeito ilustrativo, os resultados da série1, gerados pela aplicação do algoritmo, são mostrados na Tabela1, indicando o número de iterações intermediárias em cada iteração principal, isto é, como se comporta a busca linear por partes.

iter	$x_1$	$x_2$	iter	$x_1$	$x_2$
s.i.	0.40400000000	0.54400000000	5	2.99968403888	1.99976528818
1	0.75835422481	1.00000009120	6	2.99977217683	1.99999647932
	1.00000004833	1.31096158072	7	2.99999516864	1.99999711917
	1.53544670387	2.00000013781	8	2.99999635158	1.99999995679
	1.99303170056	2.58884256049	9	2.99999635158	1.99999995679
2	1.99343458577	2.00883263841	10	2.99999992700	1.99999996281
3	2.00000000131	2.00289108208	11	2.99999994435	1.99999999944
	2.00314672532	2.00004336623	12	2.99999999892	1.99999999951
4	2.11731018435	1.99999999999	13	2.99999999916	1.99999999999
	2.98659454917	1.99966979281	14	2.99999999998	1.99999999999

Tabela 1

Na tabela acima, *iter* representa uma iteração principal e *s.i.* a solução inicial utilizada.

A prova da convergência nos algoritmos afim escala primal passa, em algum momento, pela verificação de que  $\lim_{k \rightarrow \infty} \|d_x^k\| = 0$  e aqui também este fato se deu em todos os exemplos testados. A tabela 2 mostra tal comportamento para o vetor direção calculado quando da evolução do algoritmo, referente à série 1 da figura 4, nas nove primeiras iterações.

1ª iteração	2ª iteração	3ª iteração	4ª iteração	5ª iteração
d[1]=1.15E-1	d[1]=2.25E-2	d[1]=6.29E-6	d[1]=9.07E-9	d[1]=8.81E-10
d[2]=-1.94E-1	d[2]=3.31E-2	d[2]=9.43E-6	d[2]=2.12E-9	d[2]=1.31E-9
d[3]=-7.36E-1	d[3]=-1.31E-3	d[3]=-2.68E-8	d[3]=-2.29E-8	d[3]=-5.1E-12
d[4]=4.68E-1	d[4]=-1.22E-1	d[4]=-3.45E-5	d[4]=-1.54E-8	d[4]=-4.83E-9
d[5]=-5.41E-1	d[5]=-3.44E-2	d[5]=-9.45E-6	d[5]=-2.51E-8	d[5]=-1.32E-9
6ª iteração	7ª iteração	8ª iteração	9ª iteração	
d[1]=1.63E-12	d[1]=1.33E-13	d[1]=3.31E-16	d[1]=3.06E-17	
d[2]=2.98E-13	d[2]=1.99E-13	d[2]=7.03E-17	d[2]=4.53E-17	
d[3]=-4.31E-12	d[3]=-1.05E-15	d[3]=-8.52E-16	d[3]=-1.31E-18	
d[4]=-2.53E-12	d[4]=-7.31E-13	d[4]=-5.42E-16	d[4]=-1.66E-16	
d[5]=-4.61E-12	d[5]=-4.61E-12	d[5]=-2.00E-13	d[5]=-9.22E-16	

Tabela 2

## Exemplo 2

Considerando a região de factibilidade definida por

$$\begin{cases} -x_1 + x_2 \leq 2 \\ x_1 + 3x_2 \leq 14 \\ 3x_1 - x_2 \leq 12 \end{cases}$$

e a função objetivo linear por partes dada por  $f(x) = f_1(x_1) + f_2(x_2)$ , com

$$f_1(x_1) = \begin{cases} -3x_1 + 9, & 0 \leq x_1 \leq 1 \\ -2x_1 + 8, & 1 \leq x_1 \leq 2 \\ -0,5x_1 + 5, & 2 \leq x_1 \leq 3 \\ x_1 + 0,5, & 3 \leq x_1 \leq 4 \end{cases}$$

$$f_2(x_2) = \begin{cases} -3x_2 + 3, & 0 \leq x_2 \leq 1 \\ -2x_2 + 2, & 1 \leq x_2 \leq 2 \\ x_2 - 4, & 2 \leq x_2 \leq 3 \\ 2x_2 - 7, & 3 \leq x_2 \leq 4, \end{cases}$$

a Figura 6 mostra a busca do ótimo quando se apresenta uma única solução ótima, e esta se encontra no interior da região de factibilidade( neste exemplo, no ponto (3,2) ).

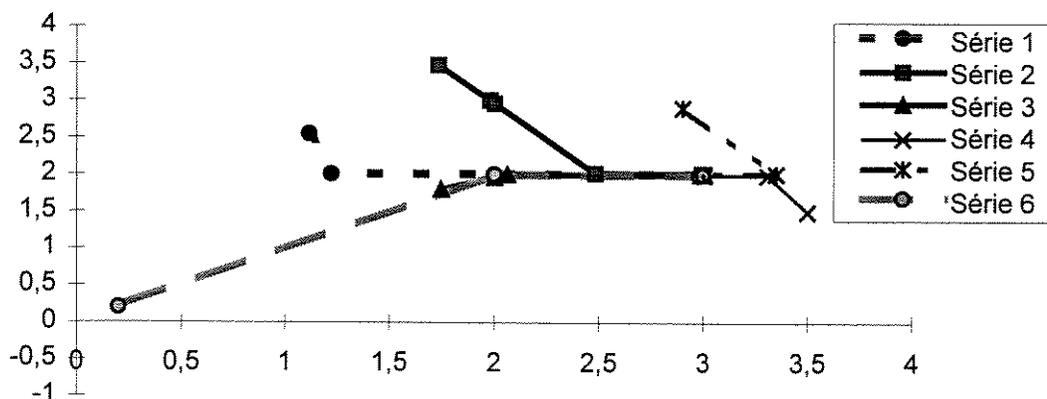


Figura 6

### Exemplo 3

Utilizando as restrições do Exemplo 2 e a função objetivo do exemplo 1, a figura 7 descreve o comportamento do algoritmo quando aplicado num problema de múltiplas soluções ótimas. Tais soluções são tais que  $2 \leq x_1 \leq 3$  e  $x_2 = 2$ .

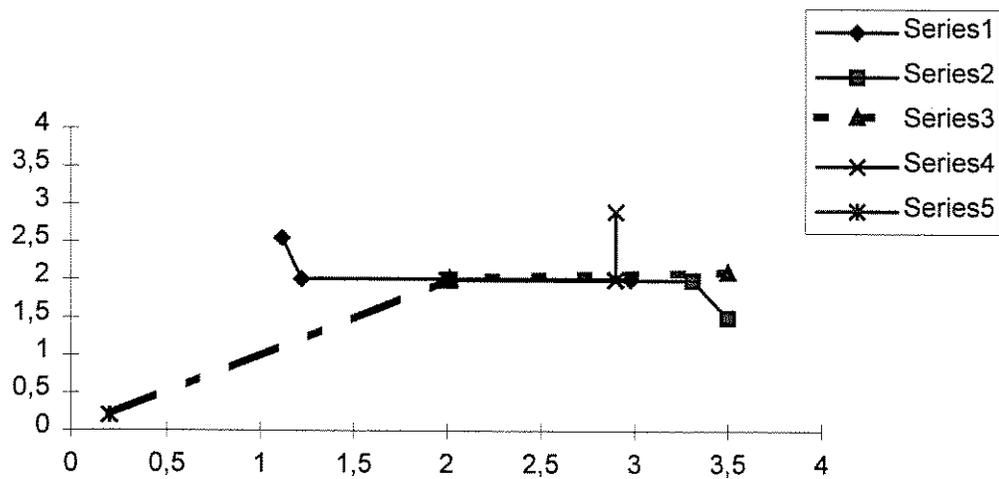


Figura 7

#### Exemplo 4

Consideremos o seguinte problema de programação linear por partes (Goldstein e Youdine):

$$\begin{aligned} \text{Min } F(\mathbf{x}) &= \sum_{i=1}^8 f_i(x_i) \\ \text{s.a. } \mathbf{Ax} &\leq \mathbf{b} \\ \mathbf{x} &\in \Omega, \end{aligned}$$

onde

$$f_1(x_1) = \begin{cases} -8x_1 & 0 \leq x_1 \leq 1, \\ -5x_1 - 3 & 1 \leq x_1 \leq 2, \\ -4x_1 - 5 & 2 \leq x_1 \leq 3, \\ -2x_1 - 11 & 3 \leq x_1 \leq 4. \end{cases}$$

$$f_2(x_2) = \begin{cases} -9x_2 & 0 \leq x_2 \leq 1, \\ -7x_2 - 2 & 1 \leq x_2 \leq 2, \\ -6x_2 - 4 & 2 \leq x_2 \leq 3, \\ -5x_2 - 7 & 3 \leq x_2 \leq 4, \\ -3x_2 - 15 & 4 \leq x_2 \leq 5. \end{cases}$$

$$f_3(x_3) = \begin{cases} -8x_3 & 0 \leq x_3 \leq 2, \\ -4x_3 - 8 & 2 \leq x_3 \leq 5, \\ -x_3 - 28 & 5 \leq x_3 \leq 6. \end{cases}$$

$$f_4(x_4) = \begin{cases} -6x_4 & 0 \leq x_4 \leq 8, \end{cases}$$

$$f_5(x_5) = \begin{cases} -8x_5 & 0 \leq x_5 \leq 1, \\ -5x_5 - 3 & 1 \leq x_5 \leq 2. \end{cases}$$

$$f_6(x_6) = \begin{cases} -9x_6 & 0 \leq x_6 \leq 2, \\ -8x_6 - 2 & 2 \leq x_6 \leq 20. \end{cases}$$

$$f_7(x_7) = \begin{cases} -8x_7 & 0 \leq x_7 \leq 2, \\ -7x_7 - 2 & 2 \leq x_7 \leq 4, \\ -5x_7 - 10 & 4 \leq x_7 \leq 7. \end{cases}$$

$$f_8(x_8) = \begin{cases} -11x_8 & 0 \leq x_8 \leq 1, \\ -10x_8 - 1 & 1 \leq x_8 \leq 2, \\ -8x_8 - 5 & 2 \leq x_8 \leq 3, \\ -5x_8 - 14 & 3 \leq x_8 \leq 4, \\ -4x_8 - 18 & 4 \leq x_8 \leq 5, \end{cases}$$

sujeito as seguintes restrições

$$\begin{aligned}
x_1 - 2x_2 + 3x_3 - x_4 + 5x_5 - 3x_6 + x_7 - x_8 &\leq 10, \\
3x_1 + x_2 - x_3 + 2x_4 + 2x_5 - 2x_6 - 3x_7 + 4x_8 &\leq 8, \\
-2x_1 + x_2 + x_3 + x_5 + x_6 - 2x_8 &\leq 7, \\
5x_1 - 4x_2 + 3x_4 - 2x_5 - x_7 &\leq 5
\end{aligned}$$

e  $\mathbf{x} \in \Omega$  é tal que

$$\begin{aligned}
0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 5, \quad 0 \leq x_3 \leq 6, \quad 0 \leq x_4 \leq 8, \quad 0 \leq x_5 \leq 2, \quad 0 \leq x_6 \leq 20, \\
0 \leq x_7 \leq 7, \quad 0 \leq x_8 \leq 5.
\end{aligned}$$

Os subintervalos que aparecem na definição de cada  $f_j(x_j)$  são os que definem a chamada fronteira virtual.

Se fôssemos aplicar uma técnica de transformação para obtermos um problema de programação linear, como a aplicada por Murty[97], no lugar de 8 variáveis teríamos que trabalhar com 25, uma para cada subintervalo de definição de cada variável. Um aumento expressivo de mais de 300%, sem contar com as variáveis de folga ou artificiais, características do algoritmo ora implementado.

Vejamos alguns resultados quando da solução do Exemplo 1.

Na Tabela 3 apresentamos cinco soluções iniciais do Exemplo 4, geradas aleatoriamente pelo programa computacional. Já a Tabela 4 mostra os progressivos valores da função objetivo, nas diferentes iterações, quando as diferentes soluções iniciais dadas são utilizadas. O comportamento do algoritmo, quando aplicado neste exemplo, fica mais claro se for acompanhado da análise do Gráfico1 abaixo. Podemos observar que apesar de alguns valores da função objetivo, no início das iterações, estarem bem longe do valor ótimo, isto é rapidamente revertido.

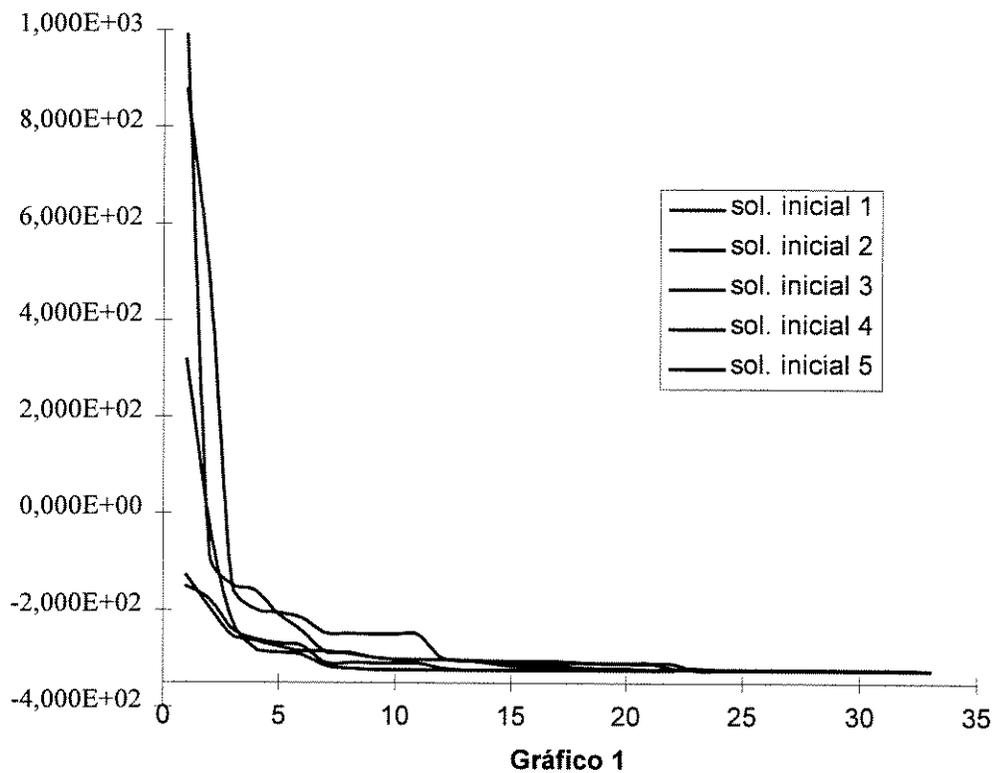
Soluções Iniciais					
x[1]=	6.880E-01	6.360E-01	1.304E+00	3.480E-01	6.840E-01
x[2]=	1.150E+00	2.040E+00	4.800E+00	6.100E-01	1.050E-01
x[3]=	5.556E+00	9.420E-01	2.640E+00	3.516E+00	4.290E+00
x[4]=	6.160E+00	4.800E-02	6.328E+00	5.920E+00	7.776E+00
x[5]=	1.610E+00	1.942E+00	1.078E+00	1.528E+00	3.500E-01
x[6]=	5.760E+00	1.304E+01	1.444E+01	1.856E+01	1.120E+00
x[7]=	4.193E+00	5.775E+00	2.982E+00	8.400E-02	3.024E+00
x[8]=	3.450E+00	1.810E+00	3.550E-01	1.130E+00	2.750E+00
x[9]=	9.591E+00	3.611E+01	5.201E+01	5.533E+01	5.768E+00
x[10]=	5.101E+00	3.718E+01	2.352E+01	2.782E+01	-5.807E+00
x[11]=	1.200E+00	-6.072E+00	-1.264E+01	-1.426E+01	8.003E+00
x[12]=	-4.907E+00	1.949E+01	3.834E+00	-8.921E+00	-1.760E+01

Tabela 3

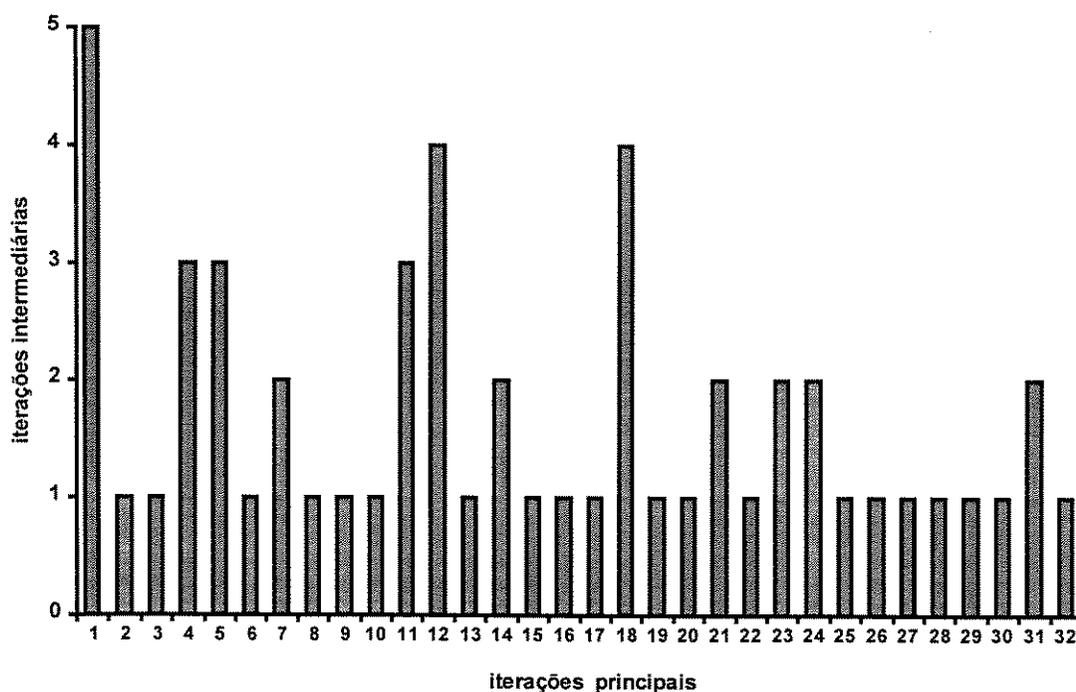
Iterações	Valor da Função Objetivo				
	sol. inicial 1	sol. inicial 2	sol. inicial 3	sol. inicial 4	sol. inicial 5
1	-1.280E+02	3.179E+02	8.780E+02	-1.511E+02	9.893E+02
2	-1.950E+02	-2.179E+01	4.820E+02	-1.777E+02	-8.538E+01
3	-2.520E+02	-2.216E+02	-1.480E+02	-2.386E+02	-1.477E+02
4	-2.630E+02	-2.793E+02	-1.980E+02	-2.607E+02	-1.591E+02
5	-2.750E+02	-2.873E+02	-2.060E+02	-2.684E+02	-2.081E+02
6	-2.840E+02	-2.927E+02	-2.170E+02	-2.735E+02	-2.432E+02
7	-2.840E+02	-3.154E+02	-2.470E+02	-3.084E+02	-2.852E+02
8	-2.900E+02	-3.203E+02	-2.480E+02	-3.088E+02	-2.867E+02
9	-2.970E+02	-3.229E+02	-2.490E+02	-3.090E+02	-2.976E+02
10	-3.010E+02	-3.230E+02	-2.490E+02	-3.090E+02	-3.015E+02
11	-3.010E+02	-3.230E+02	-2.500E+02	-3.092E+02	-3.020E+02
12	-3.020E+02	-3.230E+02	-2.960E+02	-3.190E+02	-3.020E+02
13	-3.030E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.022E+02
14	-3.080E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.044E+02

15	-3.150E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.079E+02
16	-3.150E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.080E+02
17	-3.150E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.080E+02
18	-3.170E+02	-3.230E+02	-3.070E+02	-3.230E+02	-3.080E+02
19	-3.190E+02	-3.230E+02	-3.090E+02	-3.230E+02	-3.080E+02
20	-3.190E+02	-3.230E+02	-3.090E+02	-3.230E+02	-3.080E+02
21	-3.220E+02		-3.090E+02	-3.230E+02	-3.080E+02
22	-3.220E+02		-3.090E+02	-3.230E+02	-3.190E+02
23	-3.220E+02		-3.230E+02		-3.195E+02
24	-3.220E+02		-3.230E+02		-3.220E+02
25	-3.220E+02		-3.230E+02		-3.220E+02
26	-3.220E+02		-3.230E+02		-3.220E+02
27	-3.230E+02		-3.230E+02		-3.220E+02
28	-3.230E+02		-3.230E+02		-3.220E+02
29	-3.230E+02		-3.230E+02		-3.220E+02
30	-3.230E+02		-3.230E+02		-3.220E+02
31	-3.230E+02		-3.230E+02		-3.220E+02
32	-3.230E+02				-3.220E+02
33	-3.230E+02				-3.230E+02
34					-3.230E+02
35					-3.230E+02
36					-3.230E+02
37					-3.230E+02

Tabela 4



Neste exemplo mostramos abaixo um gráfico que mostra em cada iteração principal, o número de regiões de não diferenciabilidade que atravessamos, ou seja, na busca linear por partes, quanto podemos caminhar, que vem definir o que chamamos de iterações intermediárias.



O exemplo acima foi retirado de Golstein e Youdine [87]. Nas referências pesquisadas não foram encontrados outros exemplos de problemas de programação linear por partes, sujeito a restrições lineares, que despertasse algum interesse. Afim de construirmos exemplos interessantes, o que fizemos foi tomar alguns problemas de programação linear encontrados na biblioteca NetLib. Considerando então apenas os polítopos, criamos uma função objetivo linear por partes, montando assim problemas a serem estudados.

### Exemplo 5

De acordo com o que comentamos acima, temos o Exemplo 5 com 27 equações e 59 variáveis, que foi criado a partir de AFIRO(NetLib), usando a mesma função objetivo do

Exemplo 4. A Tabela 5 mostra quatro soluções iniciais geradas aleatoriamente para este problema.

Soluções Iniciais				
x[1]=	2.020	3.492	0.932	0.708
x[2]=	4.295	3.170	4.515	0.095
x[3]=	4.614	1.542	4.560	2.172
x[4]=	3.176	6.504	6.008	6.600
x[5]=	0.018	1.702	0.698	1.062
x[6]=	3.840	16.140	6.260	10.140
x[7]=	5.362	0.980	0.028	1.974
x[8]=	2.025	1.560	0.940	1.035
x[9]=	8400.000	77400.000	74700.000	29000.000
x[10]=	29800.000	64700.000	9400.000	27000.000
x[11]=	69500.000	81900.000	57500.000	63700.000
x[12]=	70100.000	93800.000	78300.000	22500.000
x[13]=	6000.000	21900.000	60500.000	25600.000
x[14]=	84000.000	22700.000	60500.000	74800.000
x[15]=	64100.000	37600.000	31000.000	56000.000
x[16]=	27700.000	43200.000	75000.000	91000.000
x[17]=	47800.000	28200.000	82600.000	13600.000
x[18]=	19900.000	72700.000	29500.000	93400.000
x[19]=	64900.000	95500.000	87500.000	2400.000
x[20]=	81500.000	43900.000	24800.000	96900.000
x[21]=	80600.000	56200.000	12600.000	44000.000
x[22]=	75000.000	78200.000	7700.000	3600.000
x[23]=	55700.000	65800.000	30700.000	12800.000
x[24]=	5500.000	93100.000	55000.000	92600.000
x[25]=	21500.000	97000.000	6300.000	64700.000
x[26]=	44400.000	95900.000	43400.000	50500.000
x[27]=	32700.000	17100.000	21800.000	15700.000
x[28]=	61100.000	26700.000	50100.000	8000.000

x[29]=	58900.000	72600.000	50100.000	44100.000
x[30]=	53800.000	19700.000	67000.000	39500.000
x[31]=	14900.000	10000.000	76900.000	31500.000
x[32]=	71500.000	82000.000	80300.000	89400.000
x[33]=	-6.889	-1.220	-8.143	-1.559
x[34]=	-1.035	-2.803	-5.020	-5.850
x[35]=	77.980	76.508	79.068	79.292
x[36]=	-8395.700	-30657.000	-84695.000	-35840.000
x[37]=	-89989.000	-44580.000	-120990.000	-100390.000
x[38]=	-64089.000	-37579.000	-30992.000	-55985.000
x[39]=	8480.000	77478.000	74779.000	29079.000
x[40]=	29796.000	64684.000	9393.700	26990.000
x[41]=	69495.000	81899.000	57500.000	63698.000
x[42]=	70098.000	93798.000	78299.000	22499.000
x[43]=	-104900.000	-153200.000	-124600.000	-18400.000
x[44]=	-69589.000	-25324.000	7450.000	-57770.000
x[45]=	-27200.000	-42700.000	-74500.000	-90500.000
x[46]=	-34660.000	-73440.000	12460.000	-48140.000
x[47]=	75766.000	107900.000	-35848.000	28222.000
x[48]=	-283160.000	-322360.000	-203160.000	-237760.000
x[49]=	-58600.000	41300.000	-5800.000	21200.000
x[50]=	-30600.000	17700.000	35700.000	46900.000
x[51]=	-23000.000	-48700.000	-8900.000	2900.000
x[52]=	55600.000	-66400.000	-4900.000	-84600.000
x[53]=	-722110.000	-1191500.000	-713480.000	-645980.000
x[54]=	-3014.700	-4707.300	-8170.400	-9916.800
x[55]=	60511.000	-8939.500	49094.000	58325.000
x[56]=	19899.000	72699.000	29500.000	93400.000
x[57]=	53796.000	19694.000	66998.000	39496.000
x[58]=	-81193.000	-43597.000	-24496.000	-96597.000
x[59]=	-78700.000	-47300.000	-107600.000	-87200.000

Tabela 5

A Tabela 6 e o Gráfico 2 abaixo, apresentam as iterações necessárias para se obter a solução do Exemplo 5, bem como os respectivos valores da função objetivo em cada iteração.

Iterações	Valores da Função Objetivo			
1	1.35E+08	1.85E+08	1.42E+08	9.61E+07
2	1.03E+08	1.47E+08	6.17E+07	5.90E+07
3	7.70E+07	8.36E+07	4.44E+07	3.25E+07
4	5.52E+07	5.85E+07	3.19E+07	1.29E+07
5	4.03E+07	4.67E+07	2.02E+07	4.41E+06
6	1.29E+07	3.31E+07	2.97E+06	1.07E+06
7	7.94E+06	2.71E+07	5.78E+05	2.44E+05
8	2.66E+06	2.26E+07	2.18E+05	5.47E+04
9	5.64E+05	1.36E+06	6.01E+04	1.70E+04
10	1.02E+05	2.38E+05	3.65E+04	9.78E+03
11	5.75E+04	1.12E+05	2.44E+04	3.62E+03
12	4.62E+04	6.01E+04	1.77E+04	2.69E+03
13	2.50E+04	1.29E+04	1.68E+04	1.89E+03
14	2.34E+04	8.22E+03	1.57E+04	1.35E+03
15	2.11E+04	6.88E+03	1.55E+04	8.72E+02
16	1.93E+04	3.25E+03	1.53E+04	8.07E+02
17	1.89E+04	1.93E+03	1.51E+04	7.71E+02
18	9.30E+03	1.17E+03	1.47E+04	6.87E+02
19	6.94E+03	8.89E+02	1.44E+04	4.41E+02
20	5.78E+03	5.85E+02	1.41E+04	2.19E+02
21	5.14E+03	3.86E+02	1.41E+04	2.08E+02
22	4.99E+03	1.26E+01	1.37E+04	1.95E+02
23	4.97E+03	-4.70E+01	6.09E+03	1.44E+02
24	4.85E+03	-7.59E+01	1.74E+03	-4.57E+01
25	4.75E+03	-7.94E+01	9.50E+02	-6.36E+01

26	3.63E+03	-9.23E+01	2.66E+02	-6.97E+01
27	3.06E+03	-1.25E+02	2.41E+02	-8.12E+01
28	3.05E+03	-1.25E+02	2.39E+02	-8.14E+01
29	3.04E+03	-1.25E+02	1.68E+02	-8.19E+01
30	2.77E+03	-1.34E+02	-1.40E+02	-8.44E+01
31	2.31E+03	-1.40E+02	-1.45E+02	-8.44E+01
32	-8.37E+01	-1.47E+02	-1.49E+02	-8.45E+01
33	-1.20E+02	-1.48E+02	-1.55E+02	-8.62E+01
34	-1.23E+02	-1.49E+02	-1.69E+02	-1.16E+02
35	-1.24E+02	-1.49E+02	-1.90E+02	-1.18E+02
36	-1.24E+02	-1.49E+02	-1.96E+02	-1.22E+02
37	-1.26E+02	-1.49E+02	-1.98E+02	-1.22E+02
38	-1.26E+02	-1.49E+02	-2.00E+02	-1.22E+02
39	-1.26E+02	-1.49E+02	-2.00E+02	-1.22E+02
40	-1.26E+02	-1.49E+02	-2.07E+02	-1.22E+02
41	-1.26E+02	-1.49E+02	-2.07E+02	-1.22E+02
42	-1.26E+02	-1.49E+02	-2.07E+02	-1.29E+02
43	-2.01E+02	-1.49E+02	-2.07E+02	-1.41E+02
44	-2.02E+02	-1.49E+02	-2.07E+02	-1.41E+02
45	-2.02E+02	-1.50E+02	-2.07E+02	-1.41E+02
46	-2.02E+02	-1.50E+02	-2.07E+02	-1.42E+02
47	-2.02E+02	-1.50E+02	-2.07E+02	-1.46E+02
48	-2.02E+02	-1.50E+02	-2.07E+02	-1.46E+02
49	-2.02E+02	-1.50E+02	-2.07E+02	-1.46E+02
50	-2.02E+02	-1.50E+02	-2.07E+02	-1.46E+02
51	-2.07E+02	-1.50E+02	-2.07E+02	-1.46E+02
52	-2.07E+02	-1.50E+02	-2.07E+02	-1.46E+02
53	-2.07E+02	-1.53E+02	-2.07E+02	-1.46E+02
53	-2.07E+02	-1.54E+02	-2.07E+02	-1.46E+02
55	-2.07E+02	-1.54E+02	-2.07E+02	-1.46E+02
56	-2.07E+02	-1.54E+02	-2.07E+02	-1.99E+02

57	-2.07E+02	-1.54E+02	-2.07E+02	-2.00E+02
58	-2.07E+02	-1.54E+02		-2.00E+02
59	-2.07E+02	-1.54E+02		-2.00E+02
60	-2.07E+02	-1.54E+02		-2.00E+02
61	-2.07E+02	-1.94E+02		-2.00E+02
62	-2.07E+02	-2.05E+02		-2.00E+02
63	-2.07E+02	-2.07E+02		-2.00E+02
64		-2.07E+02		-2.00E+02
65		-2.07E+02		-2.00E+02
66		-2.07E+02		-2.00E+02
67		-2.07E+02		-2.00E+02
68		-2.07E+02		-2.00E+02
69		-2.07E+02		-2.00E+02
70		-2.07E+02		-2.07E+02
71		-2.07E+02		-2.07E+02
72		-2.07E+02		-2.07E+02
73		-2.07E+02		-2.07E+02
74		-2.07E+02		-2.07E+02
75		-2.07E+02		-2.07E+02
76				-2.07E+02
77				-2.07E+02

Tabela 6

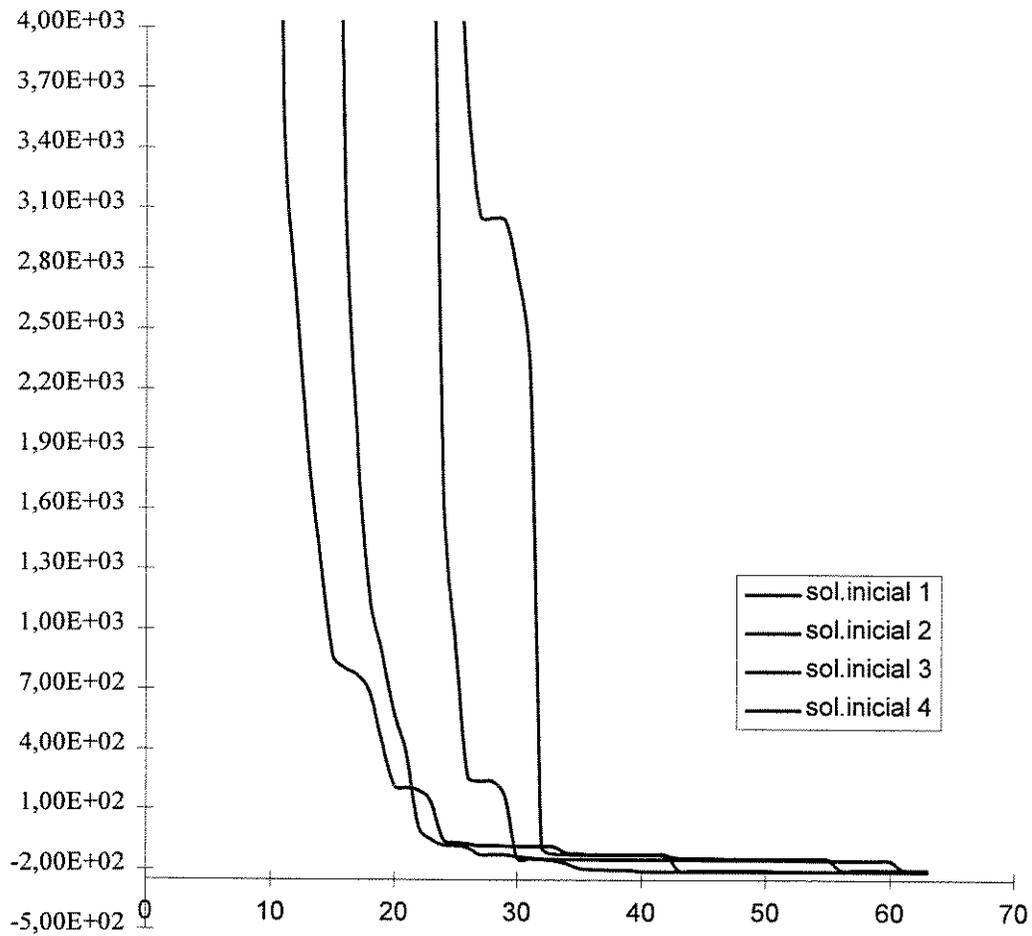


Gráfico 2

Devemos observar que no total de iterações efetuadas pelo algoritmo estão contabilizadas aquelas correspondentes ao cálculo de uma solução interior inicial e que, devido à forma utilizada em (113) ou (114), podem contribuir bastante para o aumento das mesmas.

Vejam agora, num exemplo simples, um estudo comparativo quando da aplicação do algoritmo de pontos interiores linear por partes em um problema linear por partes e da aplicação do mesmo no problema linear equivalente, a medida que vamos aumentando o número de pontos de quebra.

### Exemplo 6

Considere o seguinte problema de programação não-linear:

$$\begin{aligned} \text{Min } g(x_1, x_2) &= 4x_1^2 + 2x_2^2 - 6x_1 - 3x_2 \\ \text{s.a. } &\begin{cases} -x_1 + x_2 \leq 2 \\ 2x_1 + x_2 \leq 8 \\ 2x_1 - x_2 \leq 4 \end{cases} \\ &0 \leq x_1 \leq 3, 0 \leq x_2 \leq 4, \end{aligned}$$

cuja região de factibilidade é aquela apresentada na Figura 4.

Podemos ver que o mínimo ocorre no ponto  $(x_1, x_2) = (0.75, 0.75)$ , no interior da região de factibilidade. Observemos ainda que a função objetivo é separável com  $g(x_1, x_2) = g_1(x_1) + g_2(x_2)$ , onde  $g_1(x_1) = 4x_1^2 - 6x_1$  e  $g_2(x_2) = 2x_2^2 - 3x_2$ .

O que faremos é tomar sucessivas aproximações lineares,  $f_1$  e  $f_2$ , das funções quadráticas  $g_1$  e  $g_2$ , aumentando sucessivamente o número de pontos de quebra,  $k$ , das variáveis  $x_1$  e  $x_2$ , gerando problemas lineares por partes.

O problema não linear proposto acima é então aproximado pelo problema linear por partes:

$$\begin{aligned} \text{Min } f(x_1, x_2) &= f_1(x_1) + f_2(x_2) \\ \text{s.a. } &\begin{cases} -x_1 + x_2 \leq 2 \\ 2x_1 + x_2 \leq 8 \\ 2x_1 - x_2 \leq 4 \end{cases} \\ &0 \leq x_1 \leq 3, 0 \leq x_2 \leq 4, \end{aligned}$$

com  $f_1$  e  $f_2$  funções lineares por partes definidas nos intervalos  $[0,3]$  e  $[0,4]$ , respectivamente.

As Figuras 8 e 9, apresentadas abaixo, representam, respectivamente, os números de iterações efetuadas pelo algoritmo de pontos interiores linear por partes aplicado no problema aproximado e no problema linear equivalente, quando tomamos, respectivamente, 4, 8, 16, 32, 64 e 128 pontos de quebra na aproximação linear por partes feita acima. Em cada caso quatro diferentes soluções iniciais são utilizadas.

Podemos observar que quando resolvemos o problema linear por partes diretamente( via algoritmo interior linear por partes ) a variação do número de iterações não acompanha a variação do número de pontos de quebra, tendendo a uma estabilização(Figura 8). Já na solução via o mesmo algoritmo, mas agora aplicado ao problema linear equivalente, podemos observar um aumento no número de iterações, a medida que aumentamos o número de pontos de quebra(Figura 9).

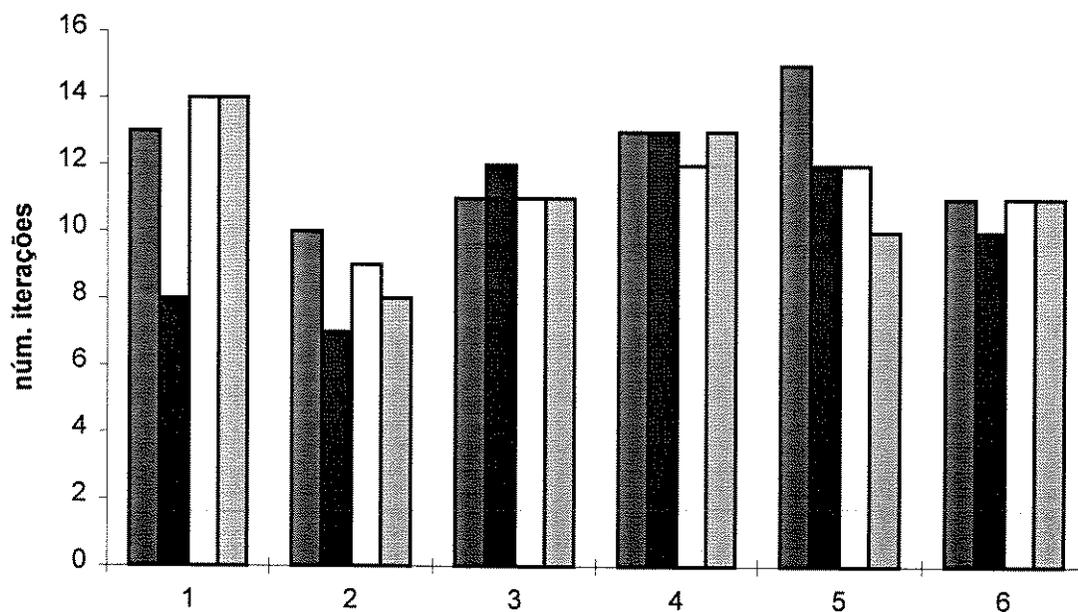


Figura 8

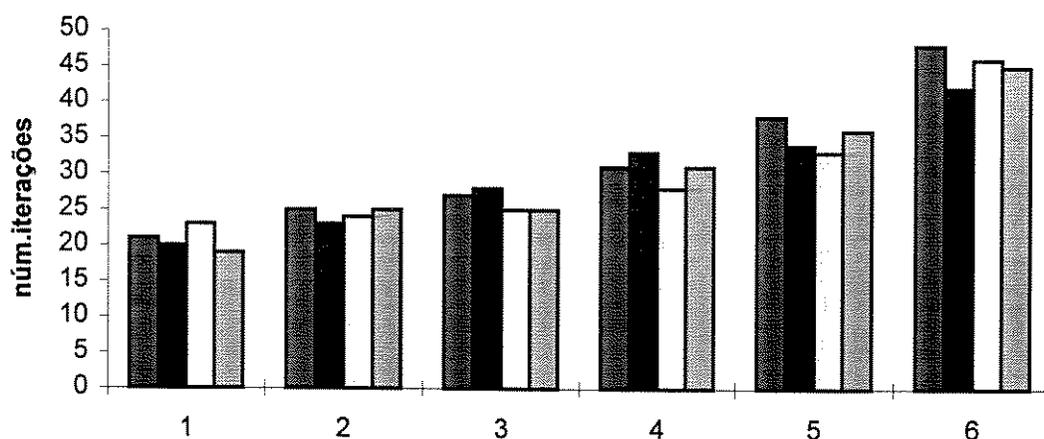


Figura 9

Já as Figuras 10 e 11 mostram o comportamento da função objetivo a medida que as iterações são efetuadas no algoritmo interior linear por partes, o mesmo ocorrendo nas Figuras 12 e 13 para o algoritmo aplicado no problema linear equivalente, ainda em

relação ao exemplo acima. Como já observamos anteriormente, no caso do algoritmo de pontos interiores linear por partes, rapidamente nos aproximamos da solução, enquanto no caso linear isto é obtido mais lentamente.

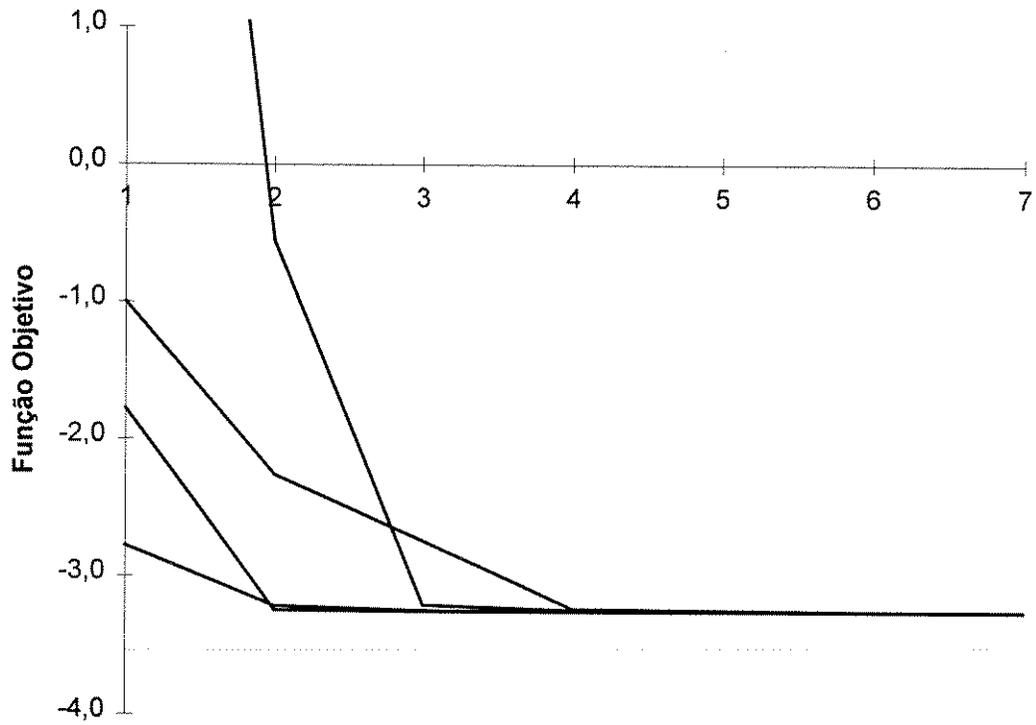


Figura 10

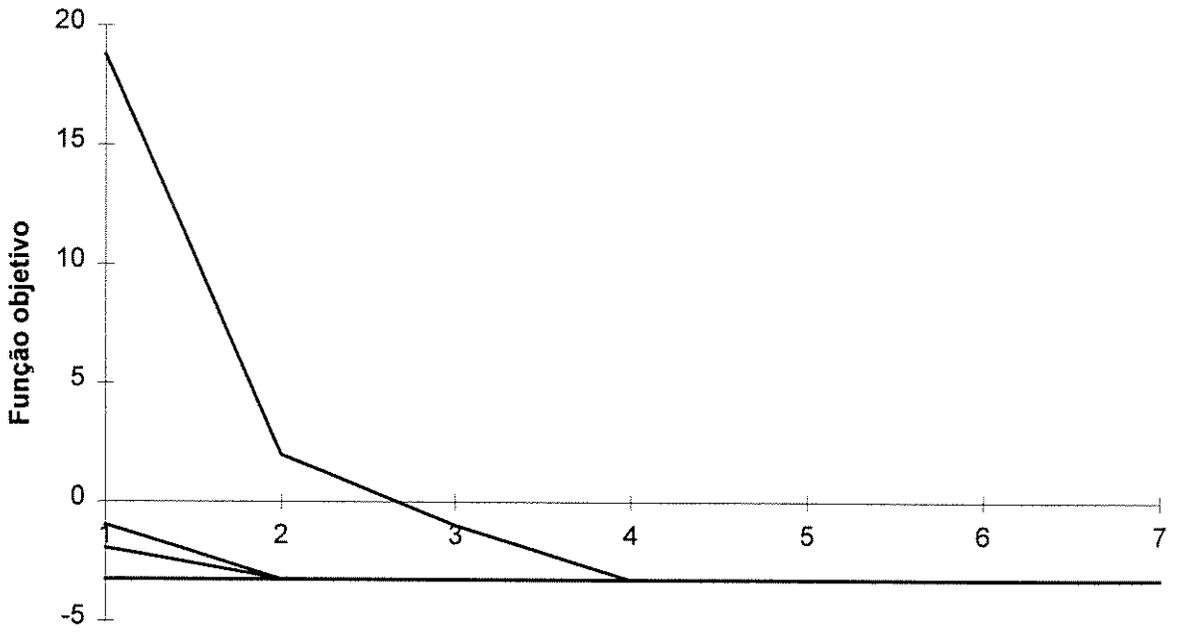


Figura 11

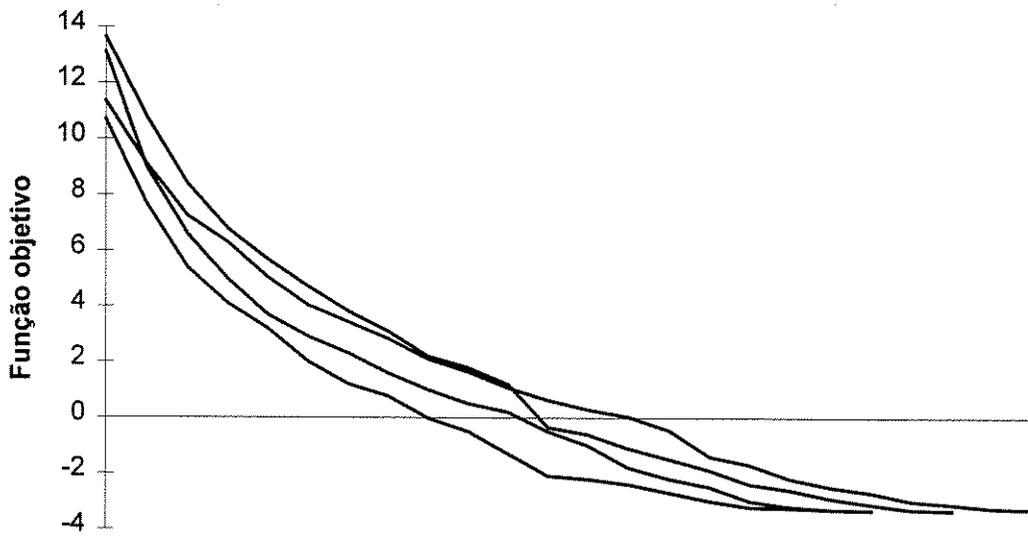


Figura 12

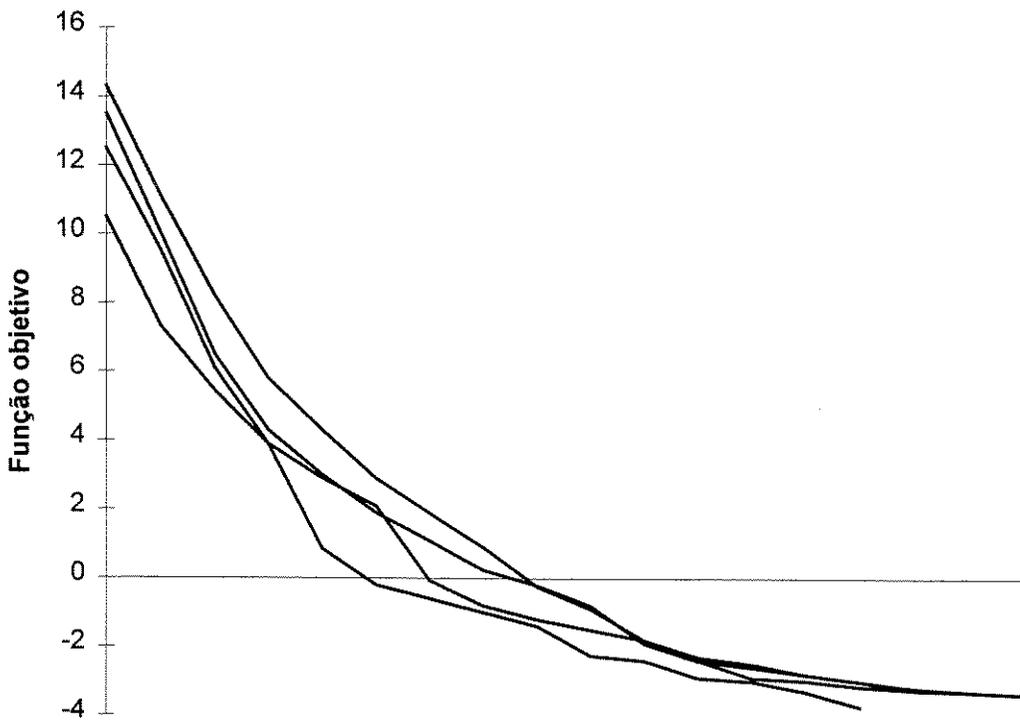


Figura 13

## **Conclusões e perspectivas**

Neste trabalho abordamos o problema de programação linear por partes, cuja função objetivo é separável e convexa e restrita a equações ou inequações lineares. Enfocamos métodos de pontos interiores, mostrando como tais métodos originalmente desenvolvidos para a programação linear podem ser estendidos para programas lineares por partes. Introduzimos o conceito de fronteira virtual, delimitando a região de linearidade do problema e permitimos que a busca unidimensional linear por partes pudesse transpor tais fronteiras. Desta forma a estrutura linear por partes é preservada no momento de desenvolver um método. O método de Sherali et al., em particular, foi detalhadamente estendido e apresentamos algumas experiências computacionais, sugerindo um campo frutífero de pesquisa, uma vez que em poucas iterações os pontos interiores gerados se aproximam da solução ótima, para então exigir um número maior de iterações até que um critério de parada fosse verificado (Outros métodos de pontos interiores podem ser empregados). Mostramos também que a técnica proposta neste trabalho, para estender métodos de pontos interiores, não pode ser obtida pela simples aplicação do método ao programa linear equivalente e explorando-se particularidades deste. Esta técnica, por sua vez, pode ser empregada pelos métodos do tipo simplex, sem perda de generalidade. O emprego de métodos de pontos interiores ao programa linear equivalente não permite que os componentes das  $n$  variáveis originais atinjam seus limites. Porém, apenas um

componente pode estar fora de seus limites na solução ótima (propriedade 1) e justamente a aproximação final à fronteira exige mais iterações de métodos de pontos interiores. Por outro lado, a técnica aqui proposta incorpora esta propriedade e a sequência obtida pode ser vista como híbrida no programa linear equivalente, onde apenas  $n$  variáveis (um componente para cada variável original) se mantêm nos interiores de seus intervalos de definição e os demais num de seus limites. Em outras palavras, o programa linear equivalente é rigoroso demais com as fronteiras virtuais, enquanto que nossa abordagem considera as fronteiras virtuais apenas no momento de definição da direção factível de descida, para em seguida abandoná-las como restrições e incorporá-las como não-diferenciabilidade à função objetivo.

Apesar de sabermos que algoritmos de pontos interiores são realmente efetivos para problemas grandes e esparsos, os exemplos explorados neste trabalho (pequenos) apontam para uma aplicabilidade do algoritmo e sugerem a importância do investimento numa pesquisa para a solução de problemas grandes.

O princípio usado para a definição da direção de busca pode ser aplicado na solução do conhecido problema *minmax*. A saber

$$\min \max \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\}, \quad (119)$$

onde

$$f_i(\mathbf{x}) = \mathbf{a}_i^t \mathbf{x} - b_i, \quad \mathbf{x} \in \mathfrak{R}^n.$$

A função  $f(x) = \max\{f_1(x), f_2(x), \dots, f_m(x)\}$ , com  $x \in \mathbb{R}^2$ , é ilustrada abaixo pela linha em destaque.

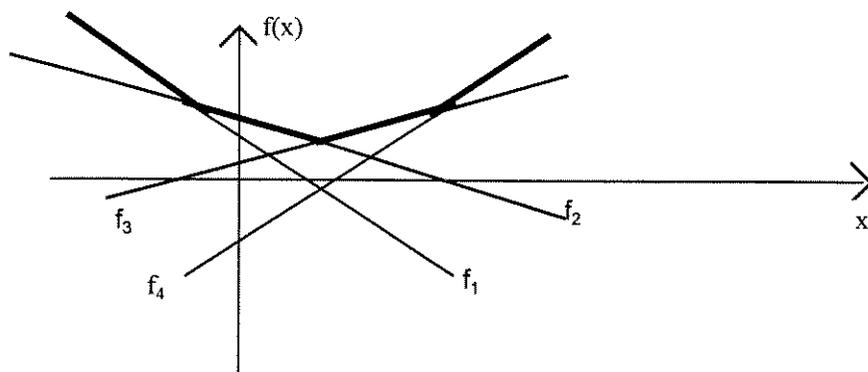


figura 8

Observe que agora os pontos de quebra não estão disponíveis como no PLP definido em (88).

Uma abordagem comum no tratamento do problema minimax é sua transformação para um programa linear e então especializar métodos de programação linear (prática esta bastante usual no tratamento da maioria dos problemas lineares por partes, como já observamos anteriormente). Para tanto, definimos uma nova variável

$$z = \max \{f_1(x), f_2(x), \dots, f_n(x)\}$$

e recaímos no seguinte programa linear

$$\begin{aligned}
& \min z \\
& \text{s. a. } z \geq f_1(\mathbf{x}) \\
& \quad z \geq f_2(\mathbf{x}) \\
& \quad \vdots \\
& \quad z \geq f_n(\mathbf{x})
\end{aligned}$$

A extensão de métodos de pontos interiores para o problema (119) pode ser obtida definindo-se a região linear de  $f(\mathbf{x})$  em torno de um ponto para a escolha da direção de descida. Em seguida fazemos uma busca unidimensional linear por partes.

Aqui uma solução interior, como antes, é aquela que não coincide com os pontos de quebra. Consideremos que  $\mathbf{x}^k$  seja uma tal solução e que

$$\max\{f_1(\mathbf{x}^k), f_2(\mathbf{x}^k), \dots, f_n(\mathbf{x}^k)\} = f_r(\mathbf{x}^k).$$

Então, para  $\mathbf{x}$  próximo de  $\mathbf{x}^k$ , teremos  $f(\mathbf{x}) = f_r(\mathbf{x}) = \mathbf{a}_r^t \mathbf{x} - b_r$ .

A região linear de  $f(\mathbf{x})$  em torno de  $\mathbf{x}^k$  é então

$$\Omega_L = \left\{ \mathbf{x} \text{ tal que } \mathbf{a}_i^t \mathbf{x} - b_i \leq \mathbf{a}_r^t \mathbf{x} - b_r \right\}$$

e para a obtenção da direção consideremos o seguinte problema:

$$\begin{aligned}
& \min f(\mathbf{x}) = \mathbf{a}_r^t \mathbf{x} - b_r \\
& \text{s. a. } (\mathbf{a}_i - \mathbf{a}_r)^t \mathbf{x} \leq b_i - b_r, \text{ com } i = 1, 2, \dots, n \text{ e } i \neq r.
\end{aligned} \tag{120}$$

Vale a pena observar que toda vez que ultrapassamos pontos-de-quebra, o problema (116) se altera, porém a direção  $\mathbf{d}$  se mantém, enquanto de descida. Para que possamos entender melhor o procedimento, vejamos:

Dada a solução  $\mathbf{x}^k$  e calculada a direção  $\mathbf{d}$ , os pontos de não-diferenciabilidade são  $\gamma_{ir} > 0$  tal que

$$f_r(\mathbf{x}^k + \gamma_{ir}\mathbf{d}^k) = f_i(\mathbf{x}^k + \gamma_{ir}\mathbf{d}^k).$$

Assim

$$f_r(\mathbf{x}^k) + \gamma_{ir}\mathbf{a}_r^t\mathbf{d}^k = f_i(\mathbf{x}^k) + \gamma_{ir}\mathbf{a}_i^t\mathbf{d}^k.$$

Ou ainda

$$\gamma_{ir} = \frac{f_i(\mathbf{x}^k) - f_r(\mathbf{x}^k)}{\mathbf{a}_r^t\mathbf{d}^k - \mathbf{a}_i^t\mathbf{d}^k}.$$

Tomamos então  $\gamma_{k1r} = \min_i \{\gamma_{ir}\}$ , e agora temos

$$f_r(\mathbf{x}^k + \gamma_{k1r}\mathbf{d}^k) = f_{k1}(\mathbf{x}^k + \gamma_{k1r}\mathbf{d}^k).$$

Se  $\mathbf{a}_{k1}^t\mathbf{d}^k < 0$ , então  $\mathbf{d}^k$  continua de descida, calculamos  $\gamma_{ik1} > 0$ ,

$$\gamma_{ik1} = \frac{f_i(\mathbf{x}^0) - f_{k1}(\mathbf{x}^0)}{\mathbf{a}_{k1}^t\mathbf{d}^k - \mathbf{a}_i^t\mathbf{d}^k}, \quad i \neq k_1$$

e tomamos  $\gamma_{k2k1} = \min_i \{\gamma_{ik1}\}$ .

Repetimos os cálculos acima até que  $\mathbf{a}_{k_j}^t \mathbf{d}^k \geq 0$ , onde então fazemos

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma \mathbf{d}^k,$$

com  $\gamma = \alpha \gamma_{k_j k_{j-1}}$ , com  $0 < \alpha < 1$ .

Um estudo mais pormenorizado do problema minmax pode vir a ser um tema de pesquisas futuras.

## Bibliografia

- [1]-Eisenhart, C., Boscovith and the combination of observations in Roger Joseph Boscovith, L.L. Whyte, eds., Fordham University Press, New York(1961).
- [2]-Rhodes, E.C., Reducing observations by the method of minimum deviations, Phil. Mag . (Seventh Series )9(1930) 974 - 992.
- [3]-Singleton, R.R., A method for minimizing the sum of absolute values of deviations, Ann. Math. Stat.11(1940)301- 310.
- [4]-Harris, T., Regression using minimum absolute deviations, Amer. Statistician,4(1950)14- 15.
- [5]-Charnes, A., Cooper, W.W. and Ferguson, R.O., Optimal estimation of executive compensation by linear programming, Management Sci.1(1955)138 - 151.
- [6]-Wagner, H.M., Linear programming techniques for regression analysis, J.Amer. Stat. Assoc.54(1959)206 - 212.
- [7]-Karst, O.J., Linear curve fitting using least deviations, J. Amer. Assoc.53(1958)118 - 132.
- [8]-Usow, K.H., On  $L_1$  approximation II: Computation for discrete functions and discretization effects , SIAM J. Anal.4(1967)233 - 244.
- [9]-Abdelmalek, N.N., Linear  $L_1$  approximation for a discrete point set and  $L_1$  solutions of overdetermined equations , J.ACM18(1971)41-47.

- [10]-Claerbaut, J. and Muir, F., Robust modelling with erratic data, *Geophysics*38(1973)826 - 844.
- [11]-Schlossmacher, E.J., An iterative technique for absolute deviations curve fitting, *J. Amer. Stat. Assoc.*68(1973)857-865.
- [12]-McCormick, G.F. and Sposito, V.A., Using the  $L_2$  estimator in  $L_1$  estimation, *SIAM J. Numer. Anal.*13(1976)337 - 343.
- [13]-Barrodale, I. and Young, A., Algorithms for best  $L_1$  and  $L_\infty$  linear approximations on a discrete set, *Numer. Math.*,8,1966.
- [14]-Barrodale, I. and Roberts, F.D.K. , An improved algorithm for discrete  $L_1$  linear approximation , *SIAM J. Numer. Anal.*10(1973)839 - 848.
- [15]-Robers, P.D. and Robers, S.S., Discrete linear  $L_1$  approximation by interval linear programming, *Communications of the ACM*, 16(1973)629-631.
- [16]-Bartels, R.H., Conn, A.R. and Sinclair, J.W., Minimization techniques for piecewise differentiable functions : The  $L_1$  solution to an overdetermined systems , *SIAM J. Numer. Anal.*15(1978)224 - 241.
- [17]-Armstrong, R.D., Frome, E.L. and Kung, D.S., A revised simplex algorithm for the absolute deviation curve fitting problem , *Commun. Stat.*, B.8(1979)175 -190.
- [18]-Abdelmalek, N.N.,  $L_1$  solution of overdetermined systems of linear equations , *ACM Trans. on Math. Soft.*6(1980)220 - 227.

- [19]-Meketon, M.S., Least absolute value regression , Tech. Report ,AT&T Bell Lab ,Murray Hill, N.J.,1987.
- [20]-Sherali, H.D., Skarpness, B.O. and Kim, B., An assumption-free convergence analysis for a perturbation of the scaling algorithm for linear programs with application to the  $L_1$  estimation problem ,Naval Res. Logist. ,35(1988)473 - 492.
- [21]-Ruzinski, S.A. and Olsen, E.T.,  $L_1$  and  $L_\infty$  minimization via a Karmarkar 's algorithm, IEEE Transactions on Acoustics Speech and Signal Processing, 37(1989)245 - 253.
- [22]-Coleman, T.F. and Li, Y , A global and quadratic affine scaling method for linear  $L_1$  problems, Tech. Report 89 - 1026 (1989), Computer Science Dept., Cornell University.
- [23]-Zhang, Y., A primal, Dual interior point Approach for computing the  $L_1$  and  $L_\infty$  solutions of overdetermined linear systems , Tech . Report, Dept. of Math. Stat., University of Maryland, Baltimore County Campus, Baltimore, 1990.
- [24]-Oliveira,A.R.L., Nascimento, M.A. and Lyra Filho, C., An interior point method specialized to the  $L_1$  regression problem -RT-DENSIS - 001/91 ,FEE-Unicamp, 1991.
- [25]-Duarte, A.M. and Vanderbei, R.J.,Tech . Report, Program in operations research and statistics, Princeton University, Princeton, N. J., 1994.
- [26]-Karmarkar, N.K., A new polynomial-time algorithm for linear programming ,  
Combinatorica4(1984)373-395

- [27]-Hertog, D D. and Roos, C., A survey of search directions in interior point methods for linear programming, *Mathematical Programming* 52(1991)481-509.
- [28]-Anstreicher, K.M., A stander form variant, and safeguarded linesearch, for the modified Karmarkar algorithm, *Mathematical Programming* 47(1990)337-351.
- [29]-Anstreicher, K.M. and Bosh, R.A., Long steps in on  $O(n^3L)$  algorithm for linear programming, *Mathematical Programming* 54(1992)251-265.
- [30]-Gay, D.M., A variant of Karmarkar's linear programming algorithm for problems in standard form, *Mathematical Programming* 37(1987)81-90.
- [31]-De Ghellinck, G., Vial, J., A polynomial Newton methods for linear programming, *Algorithmica* 1(1986)425-453.
- [32]-Gonzaga, C.C., Interior points algorithms for linear programming with inequality constrains, *Mathematical Programming (Series B)*52(1991)209-225.
- [33]-Yamashita, H., A polynomially and quadratically convergent method for linear programming, *Mathematical Systems Institute,(Tokyo,Japan,1986)*.
- [34]-Fourer, R., Notes on semi-linear programming, informal report, Department of Industrial Engineering and Management Sciences, Northwestern University(Evanston, IL,1981).
- [35]-Ye, Y. and Kojima, M., Recovering optimal basis is Karmarkar's polynomial algorithm for linear programming, *Mathematical Programming* 39(1987)305-317.

- [36]-Dikin, I.I., Iterative solution of problems of linear and quadratic programming, Doklady Akademii Nauk SSSR 174(1967)747-748.
- [37]-Adler, I., Karmarkar, N., Resende M.G.C. and Veiga, G., An implementation of Karmarkar's algorithm for linear programming, Mathematical Programming 44(1989)297-335.
- [38]-Megiddo, N. and Shub, M., Boundary behavior of interior point algorithms in linear programming, Mathematics of Operations Research 14(1)(1989)97-146.
- [39]-Barnes, E.R., A variation on Karmarkar's algorithm for solving linear programming problems, Mathematical Programming 36(1986)174-182.
- [40]-Monma, C.L., and Morton, J., Computational experience with a dual affine variant of Karmarkar's method for linear programming, Operations Research Letters 6(1987)261-267.
- [41]-Monteiro, R.D.C., Adler, I. and Resende, M.G.C., A polynomial -time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension, Mathematics os Operations Research 15(2)(1990)191-214.
- [42]-Vanderbei, R.J., Meketon, M.S. and Freedman, B.A., A modification of Karmarkar's linear programming algorithm, Algorithmica 1(1986)395-407.
- [43]-Gill, P.E., Murray, W. , Saunders, M.A., Tomlin, J.A. and Wright, M.H., On projected Newton barrier methods for linear programming and an equivalence to Karmarkar 's projective method, Mathematical Programming 36(1986)183-209.

[44]-Gonzaga, C.C., An algorithm for solving linear programming problems in  $O(n^3L)$  operations, in N.Megiddo, ed., Progress in Mathematical Programming, Interior Point and Related Methods, (Springer, New York, 1988)1-28.

[45]-Roos, C., A new trajectory following polynomial-time algorithm for the linear programming problem, Journal on Optimization Theory and its Applications 63(3)(1987) 433-458.

[46]-Roos, C., An  $O(n^3L)$  approximate center method for linear programming, in: S. Dolecki, ed. Proceedings of the Fifth French-German Conference. Lecture Notes in Mathematics 1405 (Springer, Berlin, 1989)147-158.

[47]-Renegar, J., A polynomial-time algorithm based on Newton's method for linear programming, Math.Progr.40(1988)59-93.

[48]-Roos, C., A new trajectory following polynomial-time algorithm for the linear programming problem, Journal on Optimization Theory and its Applications 63(3)(1987)433-458.

[49]-Roos, C., An  $O(n^3L)$  approximate center method for linear programming, in: S. Dolecki, ed. Proceedings of the Fifth French - German Conference. Lecture Notes in Mathematics 1405 (Springer, Berlin, 1989)147-158.

- [50]-Roos, C. and Den Hertog, D., A polynomial method of approximate weighted centers for linear programming, Report 89-13, Faculty of Mathematics and Informatica/Computer Science, Delft University of Technology ( Delft, Netherlands,1989).
- [51]-Roos, C. and Vial, J.-Ph., A polynomial method of approximate centers for linear programming, *Mathematical Programming* 54(1992)295-306.
- [52]-Todd, M.J. and Ye, Y., A centered projective algorithm for linear programming, *Mathematics of Operations Research* 15(1990)508-529.
- [53]-Vaidya, P.M., An algorithm for linear programming which requires  $O((m+n)n^2 + (m+n)^{1.5}n)L$  arithmetic operations, *Mathematical Programming* 47(1990)175-201.
- [54]-Gonzaga, C.C., Polynomial affine algorithms for linear programming, *Mathematical Programming* 49(1990)7-21.
- [55]-Hertog, D D., Roos, C. and Terlaky, T., A potential reduction variant of Renegar's short-step path-following method for linear programming, *Linear Algebra and its Applications* 152(1991)43-68.
- [56]-Roos, C., Vial, J.-Ph., Long steps with the logarithmic penalty barrier function in linear programming, in: J. Gabszewicz, J.,F. Richard and L. Wolsey, eds., *Economic Decision-making: Games, Economics and Optimization. Contributions in Honour of Jacques H. Drèze*(Elsevier Science Publisher, Amsterdam, 1990)433-441.

- [57]-Anstreicher, K.M. and Bosh, R.A., Long steps in on  $O(n^3L)$  algorithm for linear programming, *Mathematical Programming* 54(1992)251-265.
- [58]-Freund, R.M., Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function, *Mathematical Programming* 51(1991)203-222.
- [59]-Gonzaga, C.C., Large-step path-following methods for linear programming, Part 1: barrier function method, *SIAM Journal on Optimization* 1(1991)268-279.
- [60]-Kojima, M., Mizuno, S. and Yoshise, A., An  $O(\sqrt{n}L)$  iteration potential reduction algorithm for linear complementarity problems, *Mathematical Programming* 50(1991)331-342.
- [61]-Megiddo, N. , Pathways to the optimal set in linear programming , progress in mathematical programming , interior - point and related methods ,131-158, Springer-Verlag, New York,1980.
- [62]-Kojima, M. , Mizuno, S. e Yoshise, A.,A primal-dual interior point algorithm for linear programming, in N.Megiddo, ed., *Progress in Mathematical Programming, Interior Point and Related Methods*(Springer,New York,1988)29-47.
- [63]-Monteiro, R.C. and Adler, I., An  $O(n^3L)$  primal-dual interior point algorithm for linear programming, *Mathematical Programming* 44(1989)43-66.

- [64]-McShane, K.A., Monma, C.L. and Shanno, D., An implementation of a primal-dual interior point method for linear programming, *ORSA Journal on Computing*, 1(2)(1989)70-83.
- [65]-Lustig, I.J., Marsten, R.E. and Shanno, G.S., Computational experience with a primal-dual interior point method for linear programming, TR J-89-11(October 1989), Industrial and Systems Engineering Report Series, Georgia Institute of Technology, Atlanta.
- [66]-Vanderbei, R.J. and Carpenter, T.J., Symmetric indefinite systems for interior point methods, *Mathematical Programming* 58(1993)1-32.
- [67]-Fourer, R. and Mehrotra, S., Solving symmetric indefinite systems in an interior-point method for linear programming 62(1993)15-39.
- [68]-Shanno, D.F. and Bagchi, A., A unified view of interior point methods for linear programming, *Annals of Operations Research* 22(1990)55-70.
- [69]-Witzgall, C., On discrete  $L_1$  approximation, working paper.
- [70]-Monteiro, R.C. and Adler, I., Interior path - following primal - dual algorithms, part I: linear programming, *Math. Prog.* 44(1989)27-41.
- [71]-Gonzaga, C.C. and Todd, M.J., an  $O(\sqrt{nL})$ -iteration large - step primal-dual affine algorithm for linear programming, tech report 862, School of operations research and industrial engineering, Cornell University, 1989.

- [72]-Mehrotra, S., On finding a vertex solution using interior-point methods, tech. report, 89-22, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208, 1989.
- [73]-Tapia, R., Zhang, Y., Saltzman, M. and Weiser, A., The predictor-corrector interior-point method as a composite Newton method, Preliminary Report, July, 1990.
- [74]-Zhang, Y., Tapia, R. and Dennis, J., On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms, Department of Mathematics and Statistics, University of Maryland, Baltimore County Campus, Baltimore, Maryland 21228.
- [75]-Seneta, E. and Steiger, W.L., A new lad curve fitting algorithm : Slightly overdetermine equation systems in  $L_1$ , Discrete Applied Mathematics 7(1984)79-91
- [76]-Armstrong, D.D. and Kung, D.S., Algorithm As 135: Min-max estimatives for linear multiple regression problem, Appl. Statist., 28(1979)93-100.
- [77]-Barrodale, I. and Phillips, C., An improved algorithm for discrete chebyshev linear programming, Proc. 4th Manitoba Conference on Numerical Mathematics, University of Manitoba, Canada, 177-190, 1974.
- [78]-Bloomfield, P. and Steiger, W., Least absolute deviations curve-fitting, SIAM Journal on Scientific and Statistical Computing, 1(1980)290-301.

- [79]-Narula, S.C. and Wellington, J.F., An efficient algorithm for the MSAE and the MMAE regression problems, *SIAM Journal on Scientific and Statistical Computing* 9(4)(1988)717-727.
- [80]-Sklar, M.G. and Armstrong, R.D., Least absolute value and Chebyshev estimation utilizing least squares results, *Mathematical Programming* 24(1982)364-352.
- [81]-Kojima, M. ,Meggido, N. and Mizuno, S., A primal-dual exterior point algorithm for linear programming, Research Report 8500, IBM Research Division , Almaden, 1991.
- [82]-Mizuno, S., Polynomiality of the Kojima- Meggido-Mizuno infeasible interior point algorithm for linear programming, Technical Report, Department of Operations Research and Industrial Engineering, Cornell University, 1992.
- [83]-Huard, P., Resolution of mathematical programming with non-linear constraints by the method of centers, J. Abadie, *Non-Linear Programming*, 207-219, 1967-Holland, Amsterdam.
- [84]-Gonzaga, C.C., Algoritmos de pontos interiores para programação linear, 17º Colóquio Brasileiro de Matemática.
- [85]-Nakagawa, J.M. Planejamento de sistemas telefônicos: Alocação de centros de fios. Tese de Mestrado, Faculdade de Engenharia Elétrica, Unicamp, Campinas, São Paulo, Brasil, 1984.

- [86]-França, P.M., Fernandes, J.F.R. e Tavares, H.M.F., Expansão de redes telefônicas. SBA: Controle & Automação, vol.1, nº 3, julho, 1987.
- [87]-Golstein, E. and Youdine, E.T., Problèmes particuliers de la programmation linéaire, Éditions MIR, Moscou, 1966.
- [88]-Dantzig, G.B., Recent advances in linear programming, Management Science 2(1956)131-144.
- [89]-Charnes, A. and Lemke, C.E., Minimization of non-linear separable convex functionals, Naval Research Logistics Quarterly 1(1954)138-151.
- [90]-Dantzig, G.B., Johnson, S. and White, W., A linear programming approach to the chemical equilibrium problem, Management Science 5(1958)38-43.
- [91]-Rockafellar, R.T., Network flows and monotropic optimization, Wiley-Interscience, New York, 1984.
- [92]-Snyder, R.D., Programming with piecewise-linear objective functions, Department of Econometrics and Operations Research, Monash University, Clayton, Victoria, Australia, 1981.
- [93]-Conn, A.R., Linear programming via nondifferentiable penalty function, SIAM Journal on Numerical Analysis 13(1976)145-154.
- [94]-Bartels, R.H., A penalty linear programming method using reduced-gradient basis-exchange techniques, Linear Algebra and its Applications 29(1980)17-32.

- [95]-Fourer, R. A, Simplex algorithm for piecewise-linear programming I - Derivation and proof., *Mathematical Programming* 33(1985)204-233.
- [96]-Premoli, A., Piecewise-linear programming: the compact (cplp) algorithm, *Mathematical Programming* 36(1986)210-227.
- [97] Murty, K.G., *Linear and combinatorial programming*, John Wiley and Sons, INC, New York (1976).
- [98]-Bazaraa, M.S., Jarvis, J.J. e Sherali, H.D., *Linear programming and network flows*, John Wiley & Sons 2<sup>a</sup> ed., New York, 1990.
- [99]-Bazarra, M.S e Shetty, C.M., *Nonlinear programming: Theory and algorithms*. Wiley, New York, 1979.
- [100]-Todd, M.J. e Burrell, B.P., Un extension to Karmarkar's algorithm for linear programming using dual variables, *Algorithmica* 1(1985)409-424.
- [101]-Fang, S. e Puthenpura, S., *Linear optimization and extensions: Theory and algorithms*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [102]-Gonzaga, C.C., Search Directions for Interior Linear Programming Methods, *Algorithmica* 6(1991)153-181.
- [103]-Monteiro, R.D.C. and Adler, I., Interior path following primal-dual algorithms, part II- Convex quadratic programming - *Mathematical Programming* 44(1989), 43-46.

- [104]-Tapia, R.A., On the formulation and theory of the Newton interior-point method for non-linear programming, *Journal of Optimization Theory and Applications* 89(1996)507-541.
- [105]- Fourer, R. A, Simplex algorithm for piecewise-linear programming II - Finiteness, feasibility and degeneracy , *Mathematical Programming* 41(1988)281-315.
- [106]- Fourer, R. A, Simplex algorithm for piecewise-linear programming III - Computational analysis and applications , *Mathematical Programming* 53(1992)213-235.
- [107]- Cavichia, M. C., Arenales, M.N., “Uma visão dos métodos de solução de sistemas lineares inconsistentes no senso da norma  $L_1$ ”, XVIII Congresso Nacional de Matemática Aplicada e Computacional, Curitiba-Pr, 693-697,1995.
- [108]- Cavichia, M. C., Arenales, M.N, “A piecewise-linear programming problem: solution via interior points”, VIII Latin-Iberian-American Congress on Operations Research and System Engineering e XXVIII Simpósio Brasileiro de Pesquisa Operacional, Rio de Janeiro-RJ, 1069-1073,1996.
- [109]- Cavichia, M. C., Arenales, M.N, “Solução de um problema de programação linear por partes via pontos interiores”, XIX Congresso Nacional de Matemática Aplicada e Computacional, Goiânia-Go, 431-432,1996.
- [110]- Arenales, M.N., Programação linear: Novos métodos e alguns problemas particulares, tese de doutorado(1984), Unicamp-FEEC.