

SOLUÇÕES APROXIMADAS DE SISTEMAS
SOBREDETERMINADOS LINEARES NA NORMA L_1 .

VALDECIR POLIZELLI

ORIENTADOR: JOSÉ VITÓRIO ZAGO

DISSERTAÇÃO APRESENTADA AO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E CIÊNCIAS DA COMPUTAÇÃO DA UNIVERSIDADE ESTADUAL DE CAMPINAS COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE EM MATEMÁTICA APLICADA.

Julho de 1980.

UNICAMP
BIBLIOTECA CENTRAL

- 2 -

Aos meus pais

Antenor Polizelli

Maria Polizelli

e irmãos

Wanderley

Valdilene

AGRADECIMENTOS

Agradeço ao Professor Dr. JOSÉ VITÓRIO ZAGO, pela orientação de Tese.

Agradeço a todos Professores do PEXC que colaboraram na minha formação.

Agradeço a MARIA DE FÁTIMA CASTELLI, pelo incentivo, compreensão, e carinho.

Agradeço a JOSÉ ALBERTO CUMINATO, pelo auxílio na parte de computação.

Agradeço aos amigos: Dário, Kamões, Emerson, Eldon, Adilson, Heloisa, Marfiza, Chetti, Valéria, Silvan, Méricles, Olinde, Vera, Benine, Clotilde, Ângela, e a demais amigos pelos incentivos, momentos agradáveis - que passaram juntos.

Agradeço a Sra. VERA MARIA DE QUEIROZ MOUTA, pela paciência nos trabalhos de datilografia.

Agradeço a Srta. MARILDA, pela ajuda na bibliografia deste trabalho.

Agradeço ao CNPQ, pela bolsa concedida durante o mestrado.

SUMÁRIO

O estudo se refere a métodos que calculam soluções aproximadas de sistemas de equações lineares sobredeterminados resolvendo-os por: 1) técnicas para minimização de funções diferenciáveis, para solução L_1 de um sistema linear sobredeterminados; 2) método de deslocamento virtual - para o problema de aproximação discreta L_1 .

Ou transformando o sistema num problema de programação linear, como nos métodos: 3) um método para o problema de aproximação linear na norma L_1 ; 4) algoritmo melhorado para aproximação linear discreta L_1 .

Quanto as soluções obtidas todos os métodos podem ser considerados equivalentes, dada a pequena diferença existentes entre seus resíduos.

NOTAÇÕES

01. $A_{m \times n}$ \rightarrow matriz de m linhas por n colunas;
02. I_m \rightarrow matriz identidade de ordem m;
03. I_n \rightarrow matriz identidade de ordem n;
04. A^* \rightarrow matriz transposta conjugada de A;
05. A_D^{-1} \rightarrow inversa a direita de A;
06. A_E^{-1} \rightarrow inversa a esquerda de A;
07. A^T \rightarrow transposta de A;
08. A_{ij} \rightarrow submatriz de A;
09. G \rightarrow inversa generalizada;
10. $N(A^*)$ \rightarrow complemento ortogonal de $N(A)$;
11. $N(A)$ \rightarrow espaço vetorial gerado pelas colunas de A;
12. $M(A')$ \rightarrow espaço vetorial gerado pelas linhas de A;
13. P_A \rightarrow operador projetor de E^n sobre $N(A)$;
14. P_N \rightarrow operador projetor sobre o núcleo de A_j ;
15. PGFP \rightarrow posto de uma matriz;
16. t_T \rightarrow traço de uma matriz;
17. (\cdot) \rightarrow produto interno ;
18. \oplus \rightarrow soma direta ;
19. $|$ \rightarrow tal que ;

20. Sgn \rightarrow sinal;

21. Min \rightarrow mínimo;

22. Max \rightarrow máximo.

BIBLIOGRAFIA

- [1] ABDELMALEK, Nahib N. - "An Efficient method for the discrete linear L_1 Approximation Problem mathematics of computation" 29 (131) : 844 - 850, Jul. 1975.
- [2] _____ - On the discrete linear L_1 Approximation and L_1 solutions of over determined linear equations. Journal of approximation theory. 11, 38 - 53, 1974.
- [3] BARRODALE, I & ROBERTS, F. D. K. - An Improved Algorithm for discrete L_1 linear approximation. Siam J. Numer. Anal. 10 (5) : 839 - 849, 1973.
- [4] BARTELS, Richard, CONN, Andrew R, SINCLAIR, James W. - Minimization techniques for piecewise differentiable functions: the L_1 solution to an overdetermined linear system. Siam J. Numer. Anal. 15 (2) : 224 - 241, Apr. 1978.
- [5] FRASER, W & BENNETT - A method of virtual displacements for the Degenerate discrete L_1 Approximation Problem. 32 (142) : 421 - 430, Apr. 1978.

ÍNDICE

	<u>Página</u>
<u>CAPÍTULO I</u> - INVERSA GENERALIZADA	1
1. Introdução	1
1.1. Inversa da matriz quadrada	1
1.2. Inversa de uma matriz retangular	1
1.3. Definição de Inversa Generalizada	2
1.4. Inversa generalizada de MOORE - PENROSE	5
2. Projetores	6
2.1. Projetor Ortogonal	7
2.2. Projetor Ortogonal - Definição 2	8
3. Algoritmo QR	9
4. Implementação de um projetor	10
5. Condição de IVAR	12
6. Definição de função convexa	14
7. Objetivo do estudo	15
7.1. Relação entre aproximação linear e aproximação L_1 ..	16
 <u>CAPÍTULO II</u> - TÉCNICAS PARA MINIMIZAÇÃO DE FUNÇÕES DIFERENCIÁVEIS POR PEDACOS - SOLUÇÃO L_1 PARA UM SISTEMA LINEAR SO- BREDETERMINADO	 19
1. Introdução	19
2. Programação linear e a norma L_1	21
3. Mudança linear no argumento do Φ	23
4. Projeção	25

	<u>Página</u>
5. Optimalidade e um ponto morto	27
6. O gradiente restrito e escolha de α	32
7. O algoritmo simples	35
8. Elucidação do algoritmo	37
9. O algoritmo para caso degenerado	39
10. Exemplo	43
 <u>CAPÍTULO III - UM MÉTODO DE DESLOCAMENTO VIRTUAL PARA O PRO-</u>	
<u>BLEMA DE APROXIMAÇÃO DISCRETO DEGENERADO L_1</u>	48
1. Introdução	48
2. Descrição do método	49
3. Degeneração	55
4. Exemplo	63
 <u>CAPÍTULO IV - UM MÉTODO PARA RESOLUÇÃO DE PROBLEMA DE APROXI-</u>	
<u>MAÇÃO LINEAR NA NORMA L_1</u>	68
1. Introdução	68
2. O algoritmo dual simplex	73
3. Caso de degeneração	76
4. Elucidação do método	79
5. Exemplo	79
 <u>CAPÍTULO V - ALGORITMO MELHORADO PARA APROXIMAÇÃO LINEAR DIS-</u>	
<u>CRETA L_1</u>	85

	<u>Página</u>
1. Comentários preliminares	85
2. Apresentação do algoritmo	85
3. Exemplo	92

CAPÍTULO I

"INVERSA GENERALIZADA"

1. INTRODUÇÃO

1.1. Inversa da Matriz Quadrada

Seja A uma matriz quadrada de ordem m ($A_{m \times m}$) com $\text{POST}(A) = m$ então existe uma única matriz A^{-1} , denominada Inversa de A , tal que $A.A^{-1} = A^{-1}.A = I_m$, onde I_m é uma matriz identidade de ordem m .

1.2. Inversa de uma Matriz Retangular

Obs: $A^* = \bar{A}^T$

a) Inversa à direita:

Seja A matriz retangular $n \times m$ e admitamos que $\text{POST}(A) = m$, no caso em que AA^* é de ordem $n \times n$ e existe n , a inversa de $(AA^*)^{-1}$ existe

$$I_n = (AA^*)(AA^*)^{-1} = A[\bar{A}^*(AA^*)^{-1}] = AA_D^{-1}$$

onde $A_D^{-1} = A^*(AA^*)^{-1}$

temos $AA_D^{-1} = I_n$ onde A_D^{-1} é denominada inversa à direita de A .

b) Inversa à esquerda:

Seja A uma matriz retangular de ordem $n \times m$. Suponhamos que $\text{POST}(A) = n$ tal que A^*A é uma matriz de ordem n , logo existe $(A^*A)^{-1}$ e

$$I_n = (A^*A)^{-1}(A^*A) = [(A^*A)^{-1}A^*]A = A_E^{-1}A \quad \therefore \quad \text{onde}$$

$$A_E^{-1} = (A^*A)^{-1}A^*$$

Portanto, temos $A_E^{-1} A = I_n$ onde A_E^{-1} é denominada inversa à esquerda de A.

Observação: É óbvio que A_E^{-1} ou A_D^{-1} existem apenas em casos particulares, quando o posto da matriz $m \times n$ é n ou m.

c) Representação Geral da Inversa à esquerda e à direita:

Teorema 1 :

Seja A uma matriz de ordem $m \times n$. Uma solução geral de $AX = I$ quando $\text{POST}(A) = m$ é $X = VA^*(AVA^*)^{-1}$ onde

V é uma matriz arbitrária tal que $\text{POST}(AVA^*) = \text{POST}(A)$

Uma solução geral de $XA = I$ onde $\text{P}(A) = n$ é

$$X = (A^*VA)^{-1} A^*V$$

onde V é uma matriz tal que $\text{POST}(A^*VA) = \text{POST}(A)$.

1.3. Definição de Inversa Generalizada

Definição 1 - Seja A uma matriz de ordem $m \times n$ e de posto arbitrário.

Uma inversa generalizada de A é uma matriz G tal que $x = Gy$ é uma solução de $Ax = y$ para algum y que faz a equação consistente.

Notação G Inversa Generalizada

Lema 1 - G existe se, e só se $AGA = A$

Demonstração:

Temos por definição $Ax = y$ um sistema consistente se $y = AZ$ para algum vetor Z de ordem $n \times 1$. pelo fato da existência de G temos:

$$\text{De } Ax = y \xrightarrow{\exists G} x = Gy \text{ mas como } y = AZ \text{ temos}$$

$$x = GAZ \rightarrow Ax = A(GAZ) \rightarrow y = A(GAZ)AZ = A(GAZ) \rightarrow \boxed{A = AGA}$$

(+) Assumimos que o sistema $Ax = y$ é consistente $\rightarrow \exists$ um vetor v tal que $Av = y$. Por hipótese temos $AGA = A \rightarrow AGAv = Av = y \rightarrow AGAv = y$ como $Av = y$ teremos

$$\begin{array}{l} \rightarrow AGy = y \\ \quad e \\ \quad Ax = y \end{array} \quad \left\| \begin{array}{l} \\ \\ \end{array} \right. \rightarrow x = Gy \text{ é solução da equação } Ax = y.$$

Usando o Lema (1) podemos fazer a seguinte definição equivalente de uma inversa generalizada.

Definição 2 - Uma inversa generalizada de A uma matriz de ordem $m \times n$ é a matriz G de ordem $n \times m$ tal que

$$AGA = A.$$

Lema 2 -

a) G existe se, e só se $H = GA$ é idempotente e $\text{POST}(H) = \text{trH} = \text{POST}(A)$.

b) G existe se, e só se $F = AG$ é idempotente e $\text{POST}(F) = \text{tr}F = \text{POST}(A)$.

Demonstração:

(+) Por hipótese G existe $\rightarrow AGA = A \rightarrow GAGA = GA \rightarrow H \cdot H = H \rightarrow$
 $\rightarrow H^2 = H \rightarrow H$ é idempotente.

(+) Temos que $H = GA$ é idempotente e ainda $\text{POST}(H) = \text{POST}(A) \rightarrow$

$M(A^*) = M(H^*) \rightarrow G(H^*) = \mathcal{Q}(A^*)$ onde $M(A^*)$ é o espaço vetorial gerado pelas colunas de A^* .

$\mathcal{Q}(H^*) = M(H^*)^\perp$ é o complemento ortogonal de $M(H)$ logo podemos afirmar que $H(I - H) = 0$ pois $H^2 = H$ de $H(I - H) = H - H^2 = H - H = 0$ de $H(I - H) = 0 \rightarrow A(I - H) = 0 = A(I - GA)$ ou $A = AGA$.

Pois $A(I - H) = A - AH = A - AGA = A - A = 0 \rightarrow A - AGA = 0 \rightarrow A = AGA$.

Observação: A demonstração do item (b) é equivalente ao anterior.

Definição 3 - Uma inversa generalizada de A de ordem $m \times n$ é uma matriz G de ordem $n \times m$ tal que GA é idempotente e $\text{POST}(GA) = \text{POST}(A)$ ou AG é idempotente e $\text{POST}(AG) = \text{POST}(A)$.

1. 4. INVERSA GENERALIZADA DE MOORE - PENROSE

Definição - Numa matriz G de ordem $(n \times m)$ é a inversa generalizada de A de ordem $m \times n$ se

$$AG = P_A$$

$$GA = P_G$$

onde P_A é o operador (matriz) projetando vetores de E^m sobre $M(A)$ e P_G é o operador (matriz) projetando os vetores de E^m em $M(G)$.

Observações: 1) $M(A)$ espaço das colunas de A ;

2) $M(A')$ espaço das linhas de A .

Definição: (Penrose)

G é a inversa generalizada de A se:

a) $A G A = A$

b) $(A G)^* = A G$

c) $G A G = G$

d) $(G A)^* = G A$

2. PROJETORES

Definição - Consideramos um vetor arbitrário $x \in E = E_1 \oplus E_2$ e expressamos $x = x_1 + x_2$ tal que $x_1 \in E_1$ e $x_2 \in E_2$, onde x_1 e x_2 são únicos.

O operador $P: x \rightarrow x_1$ é chamado o projetor em E_1 ao longo E_2 .

Observação: O projetor P é um operador linear homogêneo.

Teorema 1.2

Um operador linear homogêneo P é um projetor se, e só se P é idempotente, ou seja, $P^2 = P$.

Demonstração:

(\Rightarrow) Seja P um projetor sobre E_1 ao longo de E_2 , ou seja:

Observemos que para todo $x_1 \in E_1$ temos que $P(x_1) = x_1$ logo,

temos para $\forall x = x_1 + x_2$ para $x_1 \in E_1$

$$x_2 \in E_2$$

$$\begin{aligned} P^2(x) &= PP(x) = P(P(x_1 + x_2)) = P(x_1) = x_1 = P(x) \therefore P^2(x) = \\ &= P(x) \Rightarrow P^2 = P. \end{aligned}$$

(\Leftarrow) Temos P um operador linear homogêneo.

Construíremos $E_1 = \{v \mid v = Pu \text{ para algum } u \in E\}$ e

$E_2 = \{u \mid Pu = 0\}$ onde u e v são vetores do espaço E .

Claramente E_1 e E_2 são subespaços de E e, ainda

$$E = E_1 \oplus E_2.$$

$$\forall x \in E \rightarrow x = v + u \text{ onde } v \in E_1$$

$$u \in E_2$$

$$P(x) = P(v + u) = P(v) + P(u) = P(v) + 0 = v. \therefore P \text{ é um projetor sobre } E_1$$

$$\text{ao longo de } E_2.$$

Lema 1. - Se P é um projetor sobre E_1 ao longo E_2 , então $I - P$ é um projetor sobre E_2 ao longo E_1 .

Demonstração:

$$P \text{ é um projetor sobre } E_1 \text{ ao longo } E_2 \rightarrow \forall x \in E \rightarrow P(x) = x_1$$

$$\text{para } x = x_1 + x_2$$

$$(I - P)(x) = I(x) - P(x) = x - x_1 = x_1 + x_2 - x_1 = x_2. \therefore$$

$$\therefore (I - P)(x) = x_2 \rightarrow I - P \text{ é um projetor em } E_2 \text{ ao longo } E_1.$$

2.1.1. Projetor Ortogonal

Seja E um espaço vetorial sobre o corpo dos complexos com um produto interno (u, v) definido para cada par de vetores u, v em E .

Definição - O projetor em E_1 ao longo de E_2 é denominado projetor ortogonal sobre E_1 , se E_2 é o complemento ortogonal de E_1 em E .

Teorema 2.2

Um operador linear homogêneo P é um projetor ortogonal se, e só se:

(a) $P^2 = P$;

(b) $P^\# = P$ onde $P^\#$ é o adjunto de P , definido pela condição

$$(x, Py) = (P^\# x, y) \quad \text{para cada } x, y \in E.$$

Demonstração:

(a) Ver Teorema (2.1);

(b) Observemos inicialmente que $Px \in E_1$ e

$$(I - P)y \in E_2 \quad \forall x, y \in E.$$

Uma vez que, se P é um projetor ortogonal, temos:

$$\begin{aligned} ((I - P)y, Px) &= 0 \quad \forall x, y \in E \rightarrow (P^\# (I - P)y, x) = \\ &= 0 \quad \forall x, y \in E \rightarrow P^\# (I - P)y = 0 \quad \forall y \in E \rightarrow P^\# (I - P) = 0 \rightarrow \\ P &= P^\# P \rightarrow P = (P^\#)^\# = (P^\# P)^\# = P^\# P = P^\# \rightarrow P = P. \end{aligned}$$

2.2. PROJETOR ORTOGONAL - Definição 2

Definição: $P : E \rightarrow H$

$$P(u) = (u, u_1) u_1 + \dots + (u, u_r) u_r$$

$\forall u \in E$, chama-se projetor ortogonal de E sobre H .

3. ALGORITMO QR

$$A_{n \times k} = Q_{n \times k} \cdot R_{k \times k}$$

$$Q \text{ unitária} \quad Q^* = Q^{-1}$$

$$R \text{ triangular superior}$$

a_i e q_i são colunas i , das matrizes A , Q respectivamente.

$$(i) \quad r_{11} = \|a_1\|_2 = \sqrt{a_{11}^2 + a_{21}^2 + \dots + a_{n1}^2}$$

$$(ii) \quad q_1 = \begin{pmatrix} q_{11} \\ q_{21} \\ \vdots \\ q_{n1} \end{pmatrix} = \frac{a_1}{\|a_1\|_2} = \frac{a_1}{r_{11}} = \frac{1}{r_{11}} \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix}$$

Para $k = 2, \dots, n$

$$(iii) \quad r_{jk} = q_j^* a_k, \quad j = 1, 2, \dots, k-1$$

$$(iv) \quad r_{kk} = \|a_k - \sum_{i=1}^{k-1} r_{ik} q_i\|_2 = \|x_k\|_2$$

$$(v) \quad q_k = \frac{a_k - \sum_{i=1}^{k-1} r_{ik} q_i}{r_{kk}} = \frac{x_k}{r_{kk}}$$

Observação 1 - $\text{Ker}(P) = H$

Observação 2 - $P^2 = P \rightarrow E = \text{Ker}(P) \oplus \mathcal{G}_m(P)$

Observação 3 - $E = H + H$

4. IMPLEMENTAÇÃO DE UM PROJETO

Seja A_Z uma matriz de ordem $n \times k$ cujas colunas são linearmente independentes, logo ela pode ser fatorada na forma:

$$A_Z = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & \vdots & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

onde $Q = \begin{bmatrix} Q_1 & \vdots & Q_2 \end{bmatrix}$ é uma matriz ortogonal $n \times n$, R é uma matriz triangular superior, e não singular - Q_1 denota as primeiras k colunas de Q e Q_2 , as restantes $n-k$ colunas.

O projetor P_N sobre o espaço nulo de A_Z , denotado por $N(A_Z)$, é dado pela matriz:

$$P_N = Q_2 Q_2^T$$

No caso em que A_Z possui colunas linearmente dependentes, procedemos da seguinte maneira:

Se a coluna j_0 é eliminada da matriz A_Z resultando a matriz \bar{A}_Z , cuja fatoração é a seguinte:

$$\bar{A}_Z = Q \begin{bmatrix} H \\ -\frac{H}{O} \\ 0 \end{bmatrix}$$

onde H é uma matriz de Hessenberg superior, a qual é obtida de R por eliminação da j_0 -ésima coluna de R .

Para obtermos esta fatorização aplicaremos uma sequência de transformadas de Givens, do seguinte modo:

$$\begin{aligned} \bar{A}_Z &= \begin{bmatrix} Q & G_{j_0} & \dots & G_{k-1} \end{bmatrix} \begin{bmatrix} G_{k-1} & \dots & G_{j_0} \end{bmatrix} \begin{bmatrix} H \\ -\frac{H}{O} \\ 0 \end{bmatrix} = \\ &= \bar{Q} \begin{bmatrix} \bar{R} \\ -\frac{\bar{R}}{O} \\ 0 \end{bmatrix} \quad \text{onde} \quad G_V = \begin{bmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & \boxed{\begin{matrix} \gamma & \rho \\ \rho & -\gamma \end{matrix}} & & \\ & 0 & & 1 & \ddots & 1 \end{bmatrix} \end{aligned}$$

onde γ e $-\gamma$ ocupam as posições v e $v+1$ na diagonal principal, e $\gamma^2 + \rho^2 = 1$.

$$\therefore \bar{A}_Z = \bar{Q} \cdot \bar{R} \rightarrow \bar{A}_Z = \begin{bmatrix} \bar{Q}_1 & \vdots & \bar{Q}_2 \end{bmatrix} \begin{bmatrix} \bar{R} \\ -\frac{\bar{R}}{O} \\ 0 \end{bmatrix} \quad \text{onde}$$

\bar{Q}_1 denota as primeiras k colunas de Q e \bar{Q}_2 as $n-k$ restantes.

Logo, o projetor é dado por $P_N = \bar{Q}_2 \bar{Q}_2^T$

Para o processo oposto, a adição de uma coluna em A_z podemos escrever:

$$\bar{A}_z = [A_z, a] = \left[Q \begin{bmatrix} R \\ - \\ 0 \end{bmatrix}, a \right] = Q \begin{bmatrix} R & | & v_1 \\ - & - & - \\ 0 & | & v_2 \end{bmatrix} =$$

$$\begin{bmatrix} Q_1 & \vdots & Q_2 \end{bmatrix} \begin{bmatrix} R & | & v \\ - & - & - \\ 0 & | & v_2 \end{bmatrix} \quad \text{onde} \quad \begin{aligned} v_1 &= Q_1^T \cdot a \\ v_2 &= Q_2^T \cdot a \end{aligned}$$

Aplicando uma sequência de transformadas de Givens, obtemos a fatoração da seguinte forma:

$$\bar{A}_z = \left[Q \begin{matrix} G_{n-1} & \dots & G_{k+1} \end{matrix} \right] \left[\begin{matrix} G_{k+1} & \dots & G_{n-1} \end{matrix} \right] \begin{bmatrix} R & | & v_1 \\ - & - & - \\ 0 & | & v_2 \end{bmatrix} =$$

$$= \bar{Q} \begin{bmatrix} R & | & v \\ - & - & - \\ 0 & | & d \end{bmatrix} = \bar{Q} \begin{bmatrix} \bar{R} \\ - \\ 0 \end{bmatrix} \quad \text{onde}$$

$$d = (\sqrt{v_2^T \cdot v_2}, 0, \dots, 0)^T$$

Observação - Este tipo de projetor será utilizado na solução de sistema linear sobredeterminado na norma L_1 .

5. CONDIÇÃO DE HAAR

Dizemos que um conjunto de $n + 1$ funções

$\{ f_0, \dots, f_n \}$ satisfaz a condição de Haar num conjunto compacto

X se:

- (i) Cada f_i é contínua em X ;
- (ii) O determinante generalizado de Vandermonde

$$D(x_0, \dots, x_n) = \begin{bmatrix} f_0(x_0) & f_1(x_0) & \dots & f_n(x_0) \\ \vdots & \vdots & & \vdots \\ f_0(x_n) & f_1(x_n) & \dots & f_n(x_n) \end{bmatrix}$$

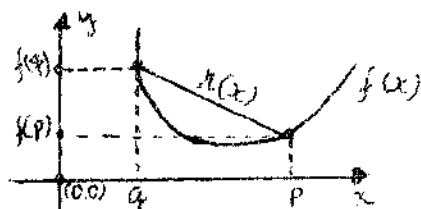
é diferente de zero para algum conjunto de $n+1$ pontos distintos $x_i \in X$.

Um sistema de funções, satisfazendo a condição de Haar, é denominado sistema de Chebyshev.

Exemplo:

$\{ 1, \cos\theta, \cos_2\theta, \dots, \cos n\theta, \sin\theta, \dots, \sin n\theta \}$ é um sistema de Chebyshev no intervalo $[0, 2\pi]$ ou $[-\pi, \pi]$.

6. DEFINIÇÃO DE FUNÇÃO CONVEXA



$$\begin{pmatrix} x \\ r(x) \end{pmatrix} = \begin{pmatrix} q \\ f(q) \end{pmatrix} + h \begin{pmatrix} p-q \\ f(p)-f(q) \end{pmatrix} \rightarrow$$

$$\rightarrow \begin{cases} x = hp + (1-h)q \\ r(x) = hf(q) + (1-h)f(p) \end{cases}$$

$$f(x) = f(hp + (1-h)q) \leq r(x) = hf(q) + (1-h)f(p) \quad .^{\circ}.$$

$$\therefore f(x) \leq hf(q) + (1-h)f(p).$$

Teorema:

A função residuo definida por:

$$\phi(x) = \| A^T x - b \|_1 = \sum_{i=1}^n | a_i^T x - b_i | \quad \text{é convexa}$$

Demonstração:

sejam p e q vetores . Fazendo $x = hp + (1-h)q$ e

$$b_i = \alpha b_i + (1-\alpha) b_i$$

$$\phi(hp + (1-h)q) = \sum_{i=1}^n | a_i^T (hp + (1-h)q) - (\alpha b_i + (1-\alpha) b_i) | \leq$$

$$\leq h \sum_{i=1}^m |a_i^T p - b_i| + (1-h) \sum_{i=1}^m |a_i^T q - b_i| =$$

$$= h \phi(p) + (1-h) \phi(q).$$

7. OBJETIVO DO ESTUDO

Nosso objetivo é criar uma função aproximação linear a uma função dada, isto é, consideremos uma função $f(x)$ definida em valores reais num subconjunto discreto.

$$p = \{p_1, p_2, \dots, p_n\}$$

do espaço Euclidiano E^n .

Dadas ainda as funções $y_j(p)$ de valores reais, definidos em p , logo construiremos uma função aproximação linear,

$$L(x, p) = \sum_{j=1}^n x_j y_j(p_j) \text{ para algum conjunto}$$

$$x = \{x_1, \dots, x_n\} \text{ de números reais.}$$

Feito isto, determinaremos a melhor aproximação $L(x^*, p)$ de modo que a

$$\sum_{i=1}^m |f(p_i) - L(x, p_i)|$$

seja mínima possível.

Podemos garantir que sempre existe uma melhor aproximação de uma função dada; vários algoritmos foram desenvolvidos para resolvê-la mas, alguns destes algoritmos requerem várias restrições a respeito do conjunto dado de funções

$$\{ \gamma_1(p), \dots, \gamma_n(p) \}$$

exigindo por exemplo que as funções

$$\gamma_i(p) \quad i = 1, \dots, n$$

sejam linearmente independentes em p , ou que satisfaçam as condições de Haar em p .

Logo, no experimento prático estas condições restringem quase que totalmente sua aplicação, uma vez que dificilmente a interpretação de um experimento científico raramente constituirá um polinômio de uma única variável independente (já que funções contínuas com mais de uma variável independente nunca constituem um conjunto de Haar)

Agora, se o problema de aproximação for apresentado de maneira equivalente a um determinado sistema de equações lineares, poderá estar totalmente dificultada a decisão de antemão sobre o posto da matriz dada. Logo, não teremos a necessária base inicial para aplicação do método.

7.1. RELAÇÃO ENTRE PROGRAMAÇÃO LINEAR E APROXIMAÇÃO L_1 .

Inicialmente, para o problema L_1 vamos utilizar a seguinte notação:

$$a_{ij} = \gamma_j(p_i) \quad , \quad b_i = f(p_i)$$

e definir as variáveis não negativas u_i, v_i, b_j, c_j colocando agora:

$$b_i - \sum_{j=1}^n a_{ij} x_j = u_i - v_i \quad i = 1, \dots, m$$

e $x_j = r_j - s_j$ para $j = 1, \dots, n$ logo, uma melhor aproximação de programação linear primal é dada por:

$$(2) \quad \begin{aligned} & \text{Minimize} \quad \sum_{i=1}^m (u_i + v_i) \\ & \text{S.a.} \quad \begin{cases} b_i = \sum_{j=1}^n (r_j - s_j) a_{ij} + u_i - v_i \\ i = 1, 2, \dots, m \\ r_j, s_j, u_i, v_i \geq 0. \end{cases} \end{aligned}$$

TEOREMA:

Se o posto de uma matriz $A = \{a_{ij}\}^T$ de ordem $m \times n$ é $k (< n)$ então existe uma melhor aproximação L_1 , a qual interpolará $f(b)$ em pelo menos k pontos de b .

Alguns autores têm sugerido que, para o caso em que m assume um valor grande (2) poderá ser resolvido utilizando o dual, que é mais conveniente, ou seja,

$$(3) \quad \begin{aligned} & \text{MÁX} \quad \sum_{i=1}^m (d_i b_i - b_i) \\ & \text{S.a.} \quad \begin{cases} \sum_{j=1}^n d_i a_{ji} = \sum_{j=1}^n a_{ij} & i = 1, \dots, m \\ e \quad 0 \leq d_i \leq 2. \end{cases} \end{aligned}$$

Obviamente há, ainda, outras maneiras para resolver o problema, como por exemplo: duas formas de algoritmo simplex revisado e o algoritmo primal dual. Mas, devido à densidade da tabela condensada correspondente a (2) e à viabilidade de uma solução básica inicial factível passar por vários vértices simplex numa iteração separada, podemos garantir que a forma padrão do método simplex é a forma viável para solucionar o problema L_1 .

CAPÍTULO II

TÉCNICAS PARA MINIMIZAÇÃO DE FUNÇÕES DIFERENCIÁVEIS POR PEDACOS SOLUÇÃO L_1 PARA UM SISTEMA LINEAR SOBRE DETERMINADO [1]

1 - INTRODUÇÃO

Seja A uma matriz real de ordem $n \times m$ ($m > n \geq 2$) com as colunas a_1, \dots, a_m , seja b um vetor de m componentes reais b_1, \dots, b_m .

Construiremos o sistema determinado:

$$1.1. \quad A^T x = b$$

de equações lineares para o qual devemos determinar um vetor x que seja a melhor aproximação do sistema na norma L_1 , ou seja:

1.2. minimize

$$\phi(x) = \|A^T x - b\|_1 = \sum_{i=1}^n |a_i^T x - b_i|.$$

Para algum ponto x (no momento não faremos nenhuma restrição sobre o ponto x) vamos considerar o conjunto

$$1.3. \quad Z = \{i \mid a_i^T x - b_i = 0\} = \{i_1, \dots, i_k\}$$

isto é, o conjunto dos índices para os quais as equações são satisfeitas, e Z^C indicará o complementar de Z .

Associado ao conjunto Z nós teríamos a matriz \therefore .

\therefore (1.4) $A_Z = \{a_{i_1} \dots a_{i_k}\} \quad i, j \in Z$, cujas colunas são obtidas de A , para os índices em que equações são satisfeitas. Consideremos o espaço nulo.

$$(1.5) \quad N = N(A_Z^\Gamma) = \{y \mid a_{i_1}^\Gamma y = 0 \text{ para } i \in Z\}.$$

Denotaremos por P_N o projetor ortogonal sobre o espaço nulo.

Consideremos, para cada ponto x , o vetor

$$(1.6) \quad r = A^\Gamma x - b = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_m \end{bmatrix}^\Gamma \quad \text{e}$$

$$(1.7) \quad S = \text{Sgn}(r) = \begin{bmatrix} \text{Sgn}(\rho_1) & \dots & \text{Sgn}(\rho_m) \end{bmatrix}^\Gamma =$$

$$\begin{bmatrix} T_1 & \dots & T_m \end{bmatrix}^\Gamma.$$

Ainda consideremos o seguinte conjunto:

$$(1.8) \quad A = \{ \alpha_\ell \mid \ell \in Z^C \text{ e } \alpha_\ell = -\rho_\ell / a_{i_1}^\Gamma P \text{ e } \alpha_\ell > 0 \}.$$

O objetivo deste estudo é um algoritmo; que solucionará o sistema (1.1), o qual deve ser considerado sob dois pontos de vista distintos: O matemático e o computacional. Consideremos também que a matriz A tem posto completo, isto é, as linhas são linearmente independentes. O caso de dependência linear, requererá considerações especiais que serão estudadas em seu devido momento.

2 - PROGRAMAÇÃO LINEAR E A NORMA L_1 .

A função (1.2) pode ser expressa como um problema de programação linear equivalente. Isto é:

$$\text{Min } \phi(x) = \|A^T x - b\|_1 = \sum_{i=1}^m |a_i^T x - b_i| \leftrightarrow$$

$$\leftrightarrow \text{Min } \ell^T W = \sum_{i=1}^m W_i$$

Sujeito a

$$W_i \geq |a_i^T x - b_i| \quad \text{para } i = 1, \dots, m$$

$$\text{onde } \ell = [1, \dots, 1]^T \quad \text{e} \quad W = [W_1, \dots, W_m]^T$$

isto, ainda pode ser reescrito na forma:

temos

$$|a_i^T x - b_i| \leq W_i \rightarrow -W_i \leq a_i^T x - b_i \leq W_i \rightarrow \begin{cases} W_i \geq a_i^T x - b_i \\ -W_i \leq a_i^T x - b_i \\ W_i \geq b_i - a_i^T x \end{cases}$$

logo, fazemos $\text{Min } \ell^T W$.

$$\text{Sujeito a } W_i \geq (a_i^T x - b_i) \quad \text{e}$$

$$W_i \geq (b_i - a_i^T x) \quad \text{para } i = 1, \dots, m.$$

Colocando na forma matricial, obtemos:

$$\left\{ \begin{array}{l} (2.1) \quad \text{Minimize} \quad [\ell^{\Gamma}, 0^{\Gamma}] \begin{bmatrix} W \\ x \end{bmatrix} \\ \\ \text{S.a} \quad \begin{bmatrix} I & -A^{\Gamma} \\ I & +A^{\Gamma} \end{bmatrix} \begin{bmatrix} W \\ x \end{bmatrix} \geq \begin{bmatrix} -b \\ +b \end{bmatrix} \end{array} \right.$$

O dual de (2.1) é dado pelo problema de programação linear.

Fazendo $y = [u, v]$

$$\left\{ \begin{array}{l} \text{Max} \quad [u^{\Gamma}, v^{\Gamma}] \begin{bmatrix} -b \\ +b \end{bmatrix} \\ \\ \text{S.a} \quad [u^{\Gamma}, v^{\Gamma}] \begin{bmatrix} I & -A^{\Gamma} \\ I & +A^{\Gamma} \end{bmatrix} \leq [\ell^{\Gamma}, 0^{\Gamma}] \\ \\ \text{onde } u, v \geq 0. \end{array} \right.$$

Desenvolvendo o resultado dual, teremos:

$$\begin{aligned} [u^{\Gamma}, v^{\Gamma}] \begin{bmatrix} I & -A^{\Gamma} \\ I & +A^{\Gamma} \end{bmatrix} &= [\ell^{\Gamma}, 0^{\Gamma}] \rightarrow \\ \rightarrow \left\{ \begin{array}{l} u^{\Gamma} + v^{\Gamma} = \ell^{\Gamma} \rightarrow u + v = \ell \cdot 0 \\ -u^{\Gamma} A^{\Gamma} + v^{\Gamma} A^{\Gamma} = 0^{\Gamma} \rightarrow Av - Au = 0 \end{array} \right. \end{aligned}$$

Portanto, temos:

$$\left\{ \begin{array}{l} \text{Max} \quad -u^T b + v^T b \\ \text{S. a.} \quad \left\{ \begin{array}{l} u + v = \ell \\ Av + Au = 0 \end{array} \right. \quad u \geq 0 \quad \text{e} \quad v \geq 0 \\ \text{Colocando-se} \quad v = \ell - u \quad \text{e} \quad y = 2u - \ell \end{array} \right.$$

Observação: A função $\phi(x)$ será minimizada num número finito de passos, aproximada por minimização de funções diferenciáveis por pedacos.

Seja x algum ponto, A_z é a matriz definida (1.4), então, a descontinuidade do gradiente ϕ e x serão manipuladas usando o projetor no espaço nulo de A_z^T .

3. MUDANÇA LINEAR NO ARGUMENTO DE ϕ .

Seja x um ponto, e P como sendo o vetor que definirá a direção de translação do ponto x .

Vamos calcular o valor da função ϕ para o ponto transladado $x + \alpha P$ para $\alpha > 0$ ou seja,

$$\begin{aligned} \phi(x + \alpha P) &= \sum_{i=1}^m [a_i^T (x + \alpha P) - b_i] \cdot \text{Sen} [a_i^T (x + \alpha P) - b_i] = \\ &= \sum_{i=1}^m [a_i^T x - b_i + a_i^T \alpha P] \cdot \text{Sen} [a_i^T x - b_i + \alpha a_i^T P] = \end{aligned}$$

$$= \sum_{i \in \mathbb{Z}^C} [\rho_i + a_i^\Gamma \alpha P] \cdot \text{Sgn}(\rho_i + a_i^\Gamma \alpha P) + \alpha \sum_{i \in \mathbb{Z}} [a_i^\Gamma P] \cdot \text{Sgn}[a_i^\Gamma P]$$

isto deve-se ao fato de que $\rho_i = a_i^\Gamma x - b_i$ é zero para $i \in \mathbb{Z}$.

Se α é um escalar pequeno, particularmente se $0 < \alpha < \text{Min } A$ para A definido anteriormente, então teremos:

$$\text{Sgn}[\rho_i + \alpha a_i^\Gamma P] = \text{Sgn}[\rho_i] = \alpha_i \quad \text{para } i \in \mathbb{Z}^C$$

este resultado é válido porque, se $\alpha = \text{Min } A +$

$$+ \quad \alpha = \frac{\rho_i}{a_i^\Gamma P}$$

Temos:

$$\text{Sgn}\left[\rho_i + \frac{\rho_i}{a_i^\Gamma P} a_i^\Gamma P\right] = \text{Sgn}[2\rho_i] = \text{Sgn}[\rho_i].$$

Como $\alpha < \frac{\rho_i}{a_i^\Gamma P}$ obviamente, o resultado acima torna-se válido.

Portanto, temos:

$$\begin{aligned} (3.1) \quad \phi(x + \alpha P) &= \sum_{i \in \mathbb{Z}^C} [\rho_i + \alpha a_i^\Gamma P] \text{Sgn}[\rho_i + \alpha a_i^\Gamma P] + \\ &+ \alpha \sum_{i \in \mathbb{Z}} [a_i^\Gamma P] \text{Sgn}[a_i^\Gamma P] = \end{aligned}$$

$$= \sum_{i \in Z} C_i |\rho_i| + \alpha \left[\sum_{i \in Z} C_i a_i^T P \sigma_i + \sum_{i \in Z} |a_i^T P| \right]$$

Se definirmos um vetor h por

$$h = \sum_{i \in Z} C_i \sigma_i a_i \rightarrow$$

$$\rightarrow h^T = \sum_{i \in Z} C_i \sigma_i a_i^T$$

logo, substituindo em (3.1), teremos:

$$\phi(x + \alpha P) = \sum_{i \in Z} C_i |\rho_i| + \alpha \left[h^T P + \sum_{i \in Z} |a_i^T P| \right] =$$

$$= \phi(x) + \alpha h^T P + \alpha \sum_{i \in Z} |a_i^T P|.$$

Finalmente, obtemos:

$$(3.2) \quad \boxed{\phi(x + \alpha P) = \phi(x) + \alpha h^T P + \alpha \sum_{i \in Z} |a_i^T P|}$$

Notemos que, solucionar o sistema (1.1) é equivalente a minimizar a função resíduo $\phi(x)$.

Daí, há a necessidade da fórmula (3.2), uma vez que partiremos de um ponto x arbitrário (seria interessante, se iniciássemos por um ponto que pelo menos fosse solução de n equações lineares do sistema). Obviamente este ponto não é solução ótima do sistema; logo, devemos transladá-lo na direção dada pelo vetor P e por α , em busca de um novo ponto que minimize a função resíduo $\phi(x)$. Este procedimento será repetido um número finito de vezes, até que a solução ótima seja atingida, isto é, o referido ponto que minimize a função resíduo $\phi(x)$.

4. PROJEÇÃO

Se nós considerarmos apenas os vetores de direção P que estão no espaço nulo de A_Z , teremos:

$$(4.1) \quad \phi(x + \alpha P) = \phi(x) + \alpha h^T P \quad \text{para todo } P \in N$$

$$\text{pois } a_i^T P = 0 \quad \text{para } i \in Z.$$

Devemos determinar algum $P \in N$ tal que $h^T P < 0$ para obter $\phi(x + \alpha P) < \phi(x)$ que nos dá uma solução mais eficiente do que a anterior.

A melhor escolha do vetor P é dada por $-P_N h$ (projeção de $-h$ no espaço nulo de A_Z), que é não nulo ao longo desta projeção.

Se a projeção é nula, o correspondente ponto x é denominado ponto morto.

Observação: O conjunto dos pontos mortos são os candidatos a ponto ótimo do sistema linear, isto é, se um ponto x é ótimo então x é um ponto morto.

5. OTIMALIDADE E PONTO MORTO.

Consideremos o vetor direção geral P definido anteriormente.

Afirmamos que $\phi(x + \alpha P) \geq \phi(x)$ para todo $\alpha > 0$ se, e só se

$$h^T P + \sum_{i \in Z} |a_i^T P| \geq 0$$

Demonstração:

$$(+) \quad \phi(x + \alpha P) \geq \phi(x) \quad \rightarrow$$

$$\rightarrow \phi(x) + \alpha h^T P + \alpha \sum_{i \in Z} |a_i^T P| \geq \phi(x) \quad \rightarrow$$

$$\rightarrow \alpha h^T P + \alpha \sum_{i \in Z} |a_i^T P| \geq 0 \quad \text{como } \alpha > 0$$

$$\text{Afirmamos } h^T P + \sum_{i \in Z} |a_i^T P| \geq 0 \quad \text{para todo } i \in Z$$

$$(+) \quad \text{temos } h^T P + \sum_{i \in Z} |a_i^T P| > 0$$

$$\text{logo } \phi(x + \alpha P) = \phi(x) + \alpha \left[h^T P + \sum_{i \in Z} |a_i^T P| \right] \quad \text{temos}$$

$\alpha > 0$ e usando a hipótese implica que

$$\Phi(x + \alpha P) \geq \Phi(x).$$

O resultado anterior nos dá um ponto x mínimo local de $\Phi(x)$. Mas, como a função $\Phi(x) = \|A^\Gamma x - b\|_1$ é convexa logo, x é um mínimo global.

Suponhamos que $P_N h$ não é nula, isto implica que

$$h^\Gamma P + \sum_{i \in Z} |a_{i1}^\Gamma P| \leq 0$$

$$\text{logo } \Phi(x + \alpha P) \leq \Phi(x).$$

Portanto, o correspondente ponto x não pode ser ótimo; e isto nos leva a concluir que se $P_N h$ é nula o correspondente ponto x é um ponto morto, candidato à solução ótima.

Então, h pertence ao espaço A_Z .

Logo, podemos escrever:

$$h = A_Z W = \sum_{j=1}^k W_j a_{ij}$$

para algum $W = [W_1, \dots, W_k]$ e $i, j \in Z$

logo, (*) $h^\Gamma P + \sum_{i \in Z} |a_{i1}^\Gamma P| \geq 0$ poderá ser reescrito na forma:

$$\sum_{j=1}^k [1 + \text{Sgn}(a_{ij} P) W_j] [a_{i1}^{\Gamma} P] \geq 0$$

Para $W = [W_1 \dots W_k]^{\Gamma} \rightarrow$

$$\rightarrow h^{\Gamma} = \sum_{j=1}^k W_j a_{ij}^{\Gamma} \quad \text{substituindo em (*)}$$

$$\sum_{j=1}^k W_j a_{ij}^{\Gamma} P + \sum [a_{i1}^{\Gamma} P] =$$

$$= \sum_{j=1}^k [1 + \text{Sgn}(a_{ij} P) (W_j)] |a_{ij} P|.$$

Neste caso, W é único se, e só se as colunas de A_Z são linearmente independentes.

(+) Temos $A_Z = [a_{i1}, \dots, a_{ik}]$ com a_{ik} LI.

Suponhamos $\exists W, u$ |

$$\left\{ \begin{array}{l} h = \sum_{j=1}^k W_j a_{ij} \\ h = \sum_{j=1}^k u_j a_{ij} \end{array} \right. \quad \left| \begin{array}{l} \\ \end{array} \right. \rightarrow \sum_{j=1}^k (W_j - u_j) a_{ij} = 0$$

$$\begin{array}{l} \text{hip} \\ \rightarrow \quad W_j - u_j = 0 \rightarrow W_j = u_j \quad \forall j \end{array}$$

logo $W = u$

(\rightarrow) Hip $\vdash W$ é um vetor único.

Tese: $\left\{ \begin{array}{l} \text{As colunas de } A_Z = [a_{i_1}, a_{i_2}, \dots, a_{i_k}] \text{ são linearmente} \\ \text{independente.} \end{array} \right.$

Demonstração:

$$\begin{array}{l} \text{Temos por } (*, *) \text{ que } h = \sum_{j=1}^k W_j a_{i_j} \text{ e por hipótese } W \text{ é único } \rightarrow \\ \rightarrow h = W_1 a_{i_1} + W_2 a_{i_2} + \dots + W_k a_{i_k} \rightarrow W_j \neq 0 \text{ para algum } j, \\ 1 \leq j \leq k. \end{array}$$

Suponhamos que as colunas de A_Z são linearmente dependente, logo existem escalares α_i $i = 1, \dots, k$ tal que $\alpha_1 a_{i_1} + \alpha_2 a_{i_2} + \dots + \alpha_k a_{i_k} = 0$ onde $\alpha_i \neq 0$ para alguns $1 \leq i \leq k$.

$$\begin{array}{l} \text{Consideremos } \alpha_1 \neq 0 \text{ e } a_{i_1} = b_2 a_{i_2} + b_3 a_{i_3} + \dots + b_k a_{i_k} \text{ onde} \\ b_i = \frac{-\alpha_1}{\alpha_i} \text{ para } i = 2, 3, \dots, k. \text{ Como } h = W_1 a_{i_1} + W_2 a_{i_2} + \dots + \\ + W_k a_{i_k} \rightarrow h = W_1 (b_2 a_{i_2} + b_3 a_{i_3} + \dots + b_k a_{i_k}) + W_2 a_{i_2} + \dots + \\ + W_k a_{i_k} \rightarrow h = (W_1 b_2 + W_2) a_{i_2} + (W_1 b_3 + W_3) a_{i_3} + \dots + (W_1 b_k + W_k) a_{i_k}. \end{array}$$

Finalmente temos

$$\left\{ \begin{array}{l} h = W_1 a_{i_1} + W_2 a_{i_2} + \dots + W_k a_{i_k} \\ h = (W_1 b_2 + W_2) a_{i_2} + (W_1 b_3 + W_3) a_{i_3} + \\ \quad + \dots + (W_1 b_k + W_k) a_{i_k}, \end{array} \right.$$

como W é único, concluímos que $W_1 = 0$. Consideremos $a_2 \neq 0$ e por analogia ao caso anterior concluímos que $W = 0$. Assim sucessivamente, concluímos que $W_3 = W_4 = \dots = W_k = 0$. $\therefore W$ é um vetor nulo (absurdo).

Um ponto x para o qual o resultado acima é verdadeiro, denomina-se um ponto morto não degenerado.

Notemos que:

$$\sum_{j=1}^k \{ 1 + \operatorname{Sen} (a_{ij}^T P) W_j \} [a_{ij}^T P]$$

pode ser feito negativo se $|W_{j_0}| > 1$ para algum j_0 .

Suponhamos que o j_0 acima existe e fazemos

$$P = -\operatorname{Sen} (W_{j_0}) P_{N_{j_0}} a_{i_{j_0}}$$

onde N_{j_0} representa o espaço nulo de A_2 , cuja coluna $a_{i_{j_0}}$ foi eliminada.

Portanto $P^T a_i = 0$ para $i \in Z - j_0$

Ainda temos

$$\text{Sgn}(P^T a_{i_{j_0}}) W_{j_0} = - |W_{j_0}|$$

pois, se $W_{j_0} < -1$ temos:

$$P = -\text{Sgn}(W_{j_0}) P_{N_{j_0}} a_{i_{j_0}} \rightarrow$$

$$P = P_{N_{j_0}} a_{i_{j_0}} \rightarrow P^T a_{i_{j_0}} > 0$$

$$\text{Sgn} | P_{N_{j_0}}^T a_{i_{j_0}} a_{i_{j_0}} | W_{j_0} = - |W_{j_0}|$$

Logo, concluímos que um ponto morto é ótimo se, só se $|W_j| \leq 1$ para todo $j = 1, \dots, k$.

6. O GRADIENTE RESTRITO À ESCOLHA DE α .

Para atingir um ponto x ótimo, podemos utilizar:

$$(6.1) \quad \phi(x + \alpha P) = \phi(x) + \alpha P^T q \quad \text{com}$$

$$P^T q < 0 \quad \text{e} \quad \alpha \text{ satisfazendo } 0 < \alpha < \text{Min } A.$$

A fórmula (6.1) pode ser justificada da seguinte maneira:

$$\text{Temos} \quad \phi(x + \alpha P) = \phi(x) + \alpha h^T P + \sum_{i \in Z} |a_i^T P| = \quad \textcircled{*}$$

Como $P \in N \rightarrow a_i^\Gamma P = 0$

ainda $q = \phi'(x) = \sum \sigma_i a_i = h^\Gamma \rightarrow h^\Gamma = q.$

$$(*) \quad \phi(x) + \alpha P^\Gamma q.$$

Observação: $\phi(x) = \sum_{i=1}^m |a_i^\Gamma x - b_i| = \sum_{i=1}^m \text{Sgn}(a_i x - b_i) (a_i^\Gamma x - b_i) =$

$$= \sum_{i=1}^m \sigma_i (a_i^\Gamma x - b_i)$$

logo, podemos verificar que:

$$\phi'(x) = \sum_{i=1}^n \sigma_i a_i^\Gamma.$$

No caso em que x não é um ponto morto, podemos fazer: $q = h$ e

$P = P_N h$ sem suposição sobre o posto de A .

Quando x é um ponto morto não degenerado, podemos tomar

$$q = h + \text{Sgn}(P_{i_{jo}}^\Gamma) a_{i_{jo}} \quad e$$

$$P = -\text{Sgn}(W_{jo}) P_{N_{jo}} a_{i_{jo}}$$

onde j_o é índice de uma componente de $W = [W_1 \dots W_j]$ para a qual

$$|W_{j_o}| > 1.$$

Quando a restrição gradiente existe nas regras acima, isto é, q existe; é possível obter um ponto $x + \alpha P$ para o qual o valor de

$$\phi(x + \alpha P) \leq \phi(x).$$

Pelas suposições, não podemos fazer α maior que o limite dado por $0 < \alpha < \text{Min } A$.

Sempre que α é feito maior que o limite superior, então definimos gradiente restrito para o ponto $x + \alpha P$ por:

$$q' = q - 2 \sum_{\ell \in L} \sigma_{\ell} a_{\ell}.$$

onde o $L \subset \mathbb{Z}^C$ é o conjunto dos índices para os quais $\alpha = \text{Min } A$.

No caso de problema não degenerado, L consiste em um único índice ℓ' , de modo que o gradiente reduz-se a

$$q' = q - 2\sigma_{\ell'} a_{\ell'}, \quad \text{onde } \ell' \text{ é o mínimo do raio positivo de } A.$$

Contudo, estamos certos de que $P^T q' < 0$. Logo, podemos afirmar que:

$$(6.6) \quad P^T q < 2 P^T \left(\sum_{\ell \in L} \sigma_{\ell} a_{\ell} \right).$$

Demonstração:

Temos

$$P^T q < 0$$

$$P^T q' < 0 \quad \text{logo}$$

$$P^T q < P^T q - P^T q' = P^T (q - q') \quad \textcircled{*}$$

$$\text{mas } g' = g - 2 \sum_{\ell \in L} \sigma_{\ell} a_{\ell} \rightarrow \boxed{g - g' = 2 \sum_{\ell \in L} \sigma_{\ell} a_{\ell}}$$

$$\textcircled{*} \quad 2 P^{\Gamma} \sum_{\ell \in L} \sigma_{\ell} a_{\ell}$$

$$\therefore P^{\Gamma} g < 2 P^{\Gamma} \sum_{\ell \in L} \sigma_{\ell} a_{\ell}$$

Observemos que quando α assume o seguinte maior valor do conjunto A , a função $\Phi(x)$ decresce e nós podemos continuar percorrendo os próximos valores de A , ajustando-os conforme $g' = g - 2 P^{\Gamma} \left(\sum_{\ell \in L} \sigma_{\ell} a_{\ell} \right)$.

Quando um maximal $\alpha \in A$ tenha sido encontrado, aplicaremos a translação do ponto x para $x + \alpha P$, e reconsideremos a questão para este novo ponto, com o objetivo de obter uma nova direção em que a função $\Phi(x)$ seja decrescente.

7. O ALGORITMO SIMPLES.

a) Caso não degenerado:

(0) selecionar algum ponto x ;

(1)(i) Identifique $Z = \{ i_1, \dots, i_k \}$;

Seja A_Z e N definido por (1.4) e (1.5) respectivamente.

(ii) Compute h de acordo com (3.4) ;

(iii) Compute $P = P_N h$;

Se $P \neq 0$, faça $g = h$, e vá para (2).

(iv) Compute W conforme (5.2) ;

(iv) Se $|w_j| \leq 1$ para todo $j = 1 \dots k$ pare ;

neste caso x é ótimo

(vi) Encontre $i_{j0} \in Z$ tal que $|w_{j0}| > j$;

(vii) Troque Z por $Z - \{ i_{j0} \}$ e faça a correspondente mudança em Λ_Z e N .

Calcule:

$$P = -\text{Sgn}(w_{j0}) P_N a_{i_{j0}}$$

e

$$q = h - \text{Sgn}(w_{j0}) a_{i_{j0}}$$

(2) Determine os elementos de Λ dados em (1.8) e ordene-os:

$$0 < \alpha_{\ell 1} < \alpha_{\ell 2} < \dots < \alpha_{\ell v}$$

onde $\alpha_{\ell v} = -\rho_{\ell j} / a_{\ell i}^\Gamma P$. Seja $v = 1$.

(3) Se $P^\Gamma q \geq 2_{\sigma_{\ell v}} P^\Gamma a_{\ell v}$ então vá para (5);

(4) Troque q por $q - 2_{\sigma_{\ell v}} a_{\ell v}$

Troque v por v_1 ;

(5) Substitua x por $x + \alpha_{\ell v} P$, e vá para (1).

8. ELUCIDAÇÃO DO ALGORITMO.

Iniciaremos a aplicação do algoritmo por um ponto qualquer; este produz uma sequência de pontos x' , os quais são obtidos a partir de seu antecessor x dado pelo passo (5), isto é, translada x por $x + \alpha_{\ell v} P$ e vai para (1).

Neste caso $\alpha_{\ell v}$ é escolhido no passo (3), com a característica de que este novo ponto satisfaz dentre outras a equação correspondente ao índice ℓ_v .

Se o ponto x' não é ponto morto, o vetor direção P que determinará o ponto x'' será dado por $P = P_N h$; isto implica que o ponto x'' satisfará as equações do ponto anterior e no mínimo mais uma, ou seja, a dele próprio. Deste modo, obteremos:

$$\phi(x'') < \phi(x').$$

Dai, podemos concluir que a menos de um ponto morto atingido; o conjunto de equações satisfeitas por cada novo ponto contém estritamente o conjunto de equações de seu antecessor, obtendo deste modo uma sequência estritamente de crescente, determinada pela função $\phi(x)$, ou seja,

$$\phi(x') > \phi(x'') > \dots > \phi(x) \quad \text{onde o ponto } x \text{ acima é morto.}$$

Para o caso de ponto morto, o qual denotaremos por \hat{x} , a determinação do vetor direção P , caso exista, será dada pelos passos (i) (iv) e (1) (VII). Obviamente se este vetor P não existir, \hat{x} será a solução ótima procurada.

Denotaremos por \hat{A}_2 a submatriz de A associada do ponto morto \hat{x} .

Observemos que $\phi(x)$ é constante em todo o espaço nulo $N(\tilde{A}_Z)$ pois

$$\text{Teremos: } \phi(x + \alpha P) = \phi(x) + \alpha h^\Gamma P$$

e

$$h = \sum_{j=1}^k w_j a_{ij} \rightarrow h^\Gamma = \sum_{j=1}^k w_j a_{ij}^\Gamma$$

logo

$$\phi(x + \alpha P) = \phi(x) + \alpha \sum_{j=1}^k w_j a_{ij}^\Gamma P$$

$$\text{Como } P \in N \text{ temos } a_{ij}^\Gamma P = 0$$

$$\text{Finalmente: } \phi(x + \alpha P) = \phi(x).$$

Vamos considerar estes valores constantes $\hat{\phi} = \phi(\hat{x})$ característicos de um ponto morto, associado a submatriz \tilde{A}_Z no subespaço

$$N(\tilde{A}_Z^\Gamma) \neq \emptyset.$$

Suponhamos que um vetor P tenha sido encontrado no passo (1) em que (5.1) é violado, então

$$\phi(x + \alpha P) < \hat{\phi}.$$

Demonstração:

$$\phi(\hat{x} + \alpha P) = \phi(\hat{x}) + \alpha h^\Gamma P$$

em (5.1) temos

$$h^T P + \sum_{i \in Z} |a^T P| \geq 0 \quad \text{para todo } P.$$

Como (5.1) foi violado, podemos afirmar que $h^T P < 0$

$$\text{Logo} \quad \phi(\tilde{x} + \alpha P) < \hat{\phi}.$$

Isto significa que, não pode ser encontrado um novo ponto morto associado a $\tilde{\Lambda}_2$ depois que \tilde{x} tenha sido deixado para trás.

Logo, podemos concluir que existe um número finito de execuções dos passos (1) a (5) entre dois pontos mortos, e que apenas um número finito destes pontos será gerado; desde que cada um seja associado a uma submatriz de Λ .

Por outro lado, como pelo menos um dos pontos mortos será solução ótima, o algoritmo não terminará até que a mesma seja encontrada.

9. O ALGORITMO PARA CASO DEGENERADO.

O algoritmo anterior torna-se inválido no caso de degeneração para os passos (1) (iv) e (1) (vii) e igualmente inválida a ordenação dos α 's, isto porque $W = \Lambda_Z^{-1} h$. Mas, com degeneração Λ_Z^{-1} não existe, porque trata-se de uma matriz não quadrada.

Logo, devemos aplicar algumas modificações no algoritmo anterior para acomodar estes tipos de dificuldades. Estas, serão resumidas no seguinte:

(2) Determine A dado em (1.8) e ordene-os:

$$0 \leq \alpha_{\ell_1} \leq \alpha_{\ell_2} \leq \dots \leq \alpha_{\ell_v}$$

seja $v = 1$.

(3) Seja $L_v = \{ \ell \mid \alpha_\ell = \alpha_{\ell_v} \}$

Se $p^\Gamma q \geq 2 p^\Gamma \sum_{\ell \in L_v} \sigma_\ell \alpha_\ell$, então vá para (5)

(4) Troque q por $q - 2 \sum_{\ell \in L_v} \sigma_\ell \alpha_\ell$

Troque v por $v + \lambda_v$ onde λ_v é o número de elementos de L_v .

Quanto aos passos (1) (iv) e (1) (vii) nós devemos retornar para (5.2), isto é, $h = A_Z W$.

Facilmente percebemos que o vetor W expressa h como combinação linear das colunas da matriz A_Z e esta não é única, uma vez que as colunas de A_Z são linearmente dependentes.

Logo, podemos escrever:

(9.1) $W = A_Z^\Gamma h + v = \hat{W} + v$ onde v é um vetor qualquer de $N(A_Z^\Gamma)$ e A_Z^Γ é a inversa generalizada de A_Z .

A expressão (5.3) pode ser modificada para:

Temos

$$\begin{aligned}
 & \sum_{j=1}^k [1 + \text{Sgn} (a_{ij}^T P) w_j] [a_{ij}^T P] = \\
 & = \sum_{j=1}^k (w_j a_{ij}^T P + |a_{ij}^T P|) = \\
 & = \sum_{j=1}^k [(\hat{w}_j + v_j) a_{ij}^T P + |a_{ij}^T P|] = \\
 & = \sum_{j=1}^k [(a_{ij}^T P + \hat{w}_j a_{ij}^T P) + \sum v_i P^T a_{ij}] = \\
 & = \sum_{j=1}^k [1 + \hat{w}_j \text{Sgn} (a_{ij}^T P)] |a_{ij}^T P| + P^T A_Z v.
 \end{aligned}$$

Observemos que, quando $P^T A_Z v = 0$ e substituindo \hat{w} por w retornaremos à formula anterior.

Com a dependência linear das colunas de A_Z não podemos garantir que para um índice j o vetor P possa ser escolhido, de modo que $a_{j0}^T P \neq 0$ enquanto que, $a_{ij}^T P = 0$ para $j \neq j_0$. Isto será verdadeiro apenas quando as colunas de A_Z forem linearmente independentes.

Sempre podemos expressar A_Z como QRV^T onde V e Q possuem colunas ortogonais e R é uma matriz não singular. Isto não é possível; apenas quando A_Z for a matriz nula. Este é um caso fora de recra, então podemos expressar

$$a_{ij} = Q(RV^T \ell_j) \quad (j = 1 - - - k) \quad \text{onde } \ell_j = (\ell_{11}, - - -, \ell_{ij}, - - - \ell_{in}).$$

$$\ell_{ij} = 0 \quad i \neq j$$

$$\ell_{ij} = 1 \quad i = j$$

Por outro lado temos:

$$A_Z^F = VR^{-1} Q^F \quad (9.4)$$

Combinando as fórmulas (9.1) e (9.4), as quais renderão um ponto ótimo para o passo (1) do algoritmo anterior se, e só se não existir vetor P para o qual

$$\sum_{j=1}^k [1 + \theta_j \sin(S_j^F Q^F P)] + S_j^F Q^F P < 0$$

Como podemos observar, do ponto de vista teórico, o problema de degeneração está resolvido; uma vez que a solução ótima foi atingida. Mas, da - das as dificuldades matemáticas, e a demora para passar de um ponto morto a outro, devido à infinita possibilidade de escolha do vetor P , nós não utili - zaremos este processo.

Logo, optamos pela inclusão de uma ou mais colunas linearmente indepen- dentes à matriz A_Z ou pela eliminação de colunas inativas, ou seja, linear- mente dependentes da mesma, de modo que torna-o possível a implementação da matriz projetora (ver item 2.3 da introdução). Consequentemente o vetor P pode ser determinado, e o algoritmo será normalmente aplicado como veremos no exemplo a seguir.

10. EXEMPLO

$$1) \text{ Seja } A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}_{2 \times 5} \quad \text{e} \quad b = [1 \ 1 \ 2 \ 3 \ 2]_{1 \times 5}$$

Calcular a melhor solução usando a norma L_1 para o sistema

$$A^T x = b \rightarrow \begin{cases} x_1 + x_2 = 1 \\ x_1 + 2x_2 = 1 \\ x_1 + 3x_2 = 2 \\ x_1 + 4x_2 = 3 \\ x_1 + 5x_2 = 2 \end{cases}$$

ou seja

$$\min \phi(x) = \|A^T x - b\|_1 = \sum_{i=1}^m |a_i^T x - b_i|$$

APLICAÇÃO DO ALGORITMO

(o) Selecionar um ponto x_0

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$(1) \text{ i. } z = \{ 1, 2 \} \rightarrow z^C = \{ 3, 4, 5 \}$$

$$A_Z = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \quad N = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}$$

$$\text{ii.} \quad h = \begin{pmatrix} -3 \\ -12 \end{pmatrix}$$

$$\text{iii.} \quad A_Z = QR \rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} \sqrt{2/2} & -\sqrt{2/2} \\ \sqrt{2/2} & \sqrt{2/2} \end{bmatrix} \begin{bmatrix} 2 & 3\sqrt{2/2} \\ 0 & \sqrt{2/2} \end{bmatrix}$$

$$P = -P_N h \quad \text{e} \quad P_N = Q_2^T \cdot Q_2$$

como $Q = [Q_1 \vdots Q_2]$ onde Q_1 é formada pelas primeiras K colunas de Q , e Q_2 pelas $n-K$ colunas de Q onde K é o posto de Q , logo

$$Q_1 = Q \quad Q_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\text{portanto} \quad P_N = (0 \ 0) \rightarrow P = 0$$

se $P \neq 0$, fazemos $g = h$, e vá para (2). (não é o caso).

$$\text{iv.} \quad h = A_Z W \rightarrow \begin{pmatrix} -3 \\ -12 \end{pmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \rightarrow$$

$$\begin{cases} w_1 + w_2 = -3 \\ w_1 + 2w_2 = -12 \end{cases} \parallel \rightarrow \begin{cases} w_1 = 6 \\ w_2 = -9 \end{cases}$$

v. $|w_j| \leq 1$ para $j = 1 \dots k \rightarrow x$ é ótimo

$$|w_1| = 6 > 1 \quad |w_2| = 9 \geq 1$$

vi. $j_0 = 2$

vii. $Z = \{1\}$ $\bar{A}_Z = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$N = \{j / j_1 = -j_2\}$$

$$P = -\text{Sgn}(w_{j_0}) P_N a_{i_{j_0}} = P_N(\bar{A}_Z) a_{i_{j_0}}$$

$$P = P_N(\bar{A}_Z) \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\bar{A}_Z = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2/2} & -\sqrt{2/2} \\ \sqrt{2/2} & \sqrt{2/2} \end{bmatrix} \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} H \\ 0 \end{bmatrix}$$

$$P_N(\bar{A}_Z) = \begin{pmatrix} -\sqrt{2/2} \\ \sqrt{2/2} \end{pmatrix} \begin{pmatrix} -\sqrt{2/2} & \sqrt{2/2} \end{pmatrix} = \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix}$$

$$P = \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}$$

$$\rightarrow P = (-1/2, 1/2)^T$$

$$g = h - \text{Sgn}(w_{jo}) a_{i_{jo}} \quad g = \begin{pmatrix} -3 \\ -12 \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} -2 \\ -10 \end{pmatrix}$$

$$\therefore g = \begin{pmatrix} -2 \\ -10 \end{pmatrix}$$

$$(2) \quad A = \{ \alpha_\ell \mid \ell \in Z^C \text{ e } \alpha_\ell = -\gamma_\ell / a_\ell^\Gamma P \text{ e } \alpha_\ell > 0 \}$$

$$Z^C = \{ 3, 4, 5 \}$$

$$\alpha_3 = -\gamma_3 / a_3^\Gamma P = 1 / \left[1, 3 \mid \begin{bmatrix} -1/2 \\ 1/2 \end{bmatrix} \right] = 1 \quad \rightarrow \quad \boxed{\alpha_3 = 1}$$

$$\alpha_4 = 4/3$$

$$\alpha_5 = 1/2 .$$

$$\alpha_5 \leq \alpha_3 \leq \alpha_4 .$$

$$(3) \quad P^\Gamma g \geq 2\sigma_5 P^\Gamma a_5$$

$$P^\Gamma g = (-1/2 \quad 1/2) \begin{pmatrix} -2 \\ -10 \end{pmatrix} = 1 - 5 = -4$$

$$2\sigma_5 P^\Gamma a_5 = -2(-1/2 \quad 1/2) \begin{pmatrix} 1 \\ 5 \end{pmatrix} = -2(2) = -4$$

$$P^\Gamma g = 2\sigma_5 P^\Gamma a_5 \quad \text{onde} \quad \boxed{\sigma_5 = -}$$

(4) O passo (4) não será utilizado.

$$\begin{aligned}
 (5) \quad x &= x + \alpha p \quad \rightarrow \quad x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix} = \\
 &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -1/4 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix} \\
 x &= \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}
 \end{aligned}$$

$$(1) \text{ i. } \quad Z = \{ 1, 5 \} \quad Z^C = \{ 2, 3, 4 \}$$

$$A_Z = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \quad N = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}$$

$$\text{ii. } \quad h = \begin{pmatrix} -1 \\ -5 \end{pmatrix}$$

$$\text{iii. } \quad P = 0$$

$$\text{iv. } \quad w_1 = 0 \quad w_2 = -1$$

$$\text{v. } \quad |w_1| = 0 \leq 1 \quad |w_2| = 1 \leq 1$$

$$\text{logo } \quad x = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix} \quad \text{é ponto ótimo.}$$

CAPÍTULO III

UM MÉTODO DE DESLOCAMENTO VIRTUAL PARA O PROBLEMA DE APROXIMAÇÃO DISCRETO DEGENERADO L_1 . [5]

1. INTRODUÇÃO

Consideremos o sistema de equações

$$(1.1) \quad \sum_{j=1}^n a_{ij} x_j = b_i \quad i = 1, \dots, m,$$

logo temos: Λ_{mxn}

Se $\text{POST}(\Lambda) = k \leq n$, podemos afirmar que existe um ponto x , o qual minimiza a função.

$$\phi(x) = \sum_{i=1}^m | \phi_i(x) | = \sum_{i=1}^m | (\Lambda_i x) - b_i |$$

onde

$\Lambda_i = [a_{i1}, \dots, a_{in}]$ é um vetor linha.

Quanto à degeneração, se $\phi_i(x) = 0$

para $i = 1, \dots, k+P$ com $P \geq 1$,

então o ponto x é denominado vértice degenerado. Se $\phi_i(x) = 0$ para $i = 1, \dots, k$ então x é um vértice ordinário.

Para obtenção de uma extremidade, ou seja, de um vértice, pegaremos

$\phi_i(x)$ = para $(k - 1)$ valores, permitindo ao restante dos $\phi_i(x)$ variar.

Quanto ao ponto mínimo para $\phi(x)$, este será um vértice ordinário, e $\phi(x)$ será não decrescente para o deslocamento ao longo de cada extremidade a partir de x . Agora, se o resultado acima é válido, mas x é um ponto degenerado, não podemos garantir que o ponto x é mínimo. Este fato será ilustrado posteriormente.

A degeneração de um vértice pode ser removida aplicando perturbações nos b_i para os índices i , correspondendo a qualquer $\phi_i(x)$ não usado para determinar x ; deste modo x passa a ser um vértice ordinário de um problema modificado.

Podemos observar que; o método será aplicado diretamente no sistema, sem transformá-lo em um problema de programação Linear equivalente; diferenciando da maioria dos métodos de nossa literatura.

2. DESCRIÇÃO DO MÉTODO.

Seja e_k a notação de um vetor unitário com 1 na posição k e zero nas demais, consideremos a matriz \bar{A} que consiste dos coeficientes das n equações do sistema

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad (i = 1, \dots, m)$$

de modo que $\text{POST}(\bar{A}) = n$.

Definiremos, ainda, o vetor \bar{b} constituído pelos escalares b_i correspondentes às equações que constituem \bar{A} .

Logo temos o sistema

$$(2.1) \quad \bar{A}X = \bar{b}.$$

A extremidade emanada do vértice x que satisfaz (2.1) tem a direção do vetor E_k que satisfaz

$$\bar{A} E_k = e_k \quad k = 1, \dots, n.$$

Logo

$$E_k = \bar{A}^{-1} e_k$$

Se $\phi_i(x) = 0$ para exatamente n valores de i , a função

$$\phi(x) = \sum_{i=1}^n |\phi_i(x)| \quad \text{tem um mínimo } x \text{ se, e só se } \phi(x) \text{ é não}$$

decrecente em cada direção $E_k \quad k = 1, \dots, n.$

Suponhamos x , um vértice ordinário, não necessariamente um mínimo, sendo a solução de

$$\bar{A}X = \bar{b}, \quad \text{isto é,} \quad \phi_i(x) = 0 \quad i = 1, \dots, n$$

$$\phi_i(x) \neq 0 \quad i = n+1 \dots m.$$

Observemos que $\text{POST}(\bar{A}) = n$

logo, existe \bar{A}^{-1} inversa de \bar{A} e ainda denotaremos a coluna j da inversa por C_j .

Como não é um ponto de mínimo para o sistema, devemos transladá-lo em uma direção conveniente; isto será feito da seguinte maneira:

$$(2.2) \quad Y = X + \lambda C_j \quad \text{com } \lambda \text{ escalar.}$$

O vértice Y está sobre a extremidade j , e a mesma é determinada pelo vetor direção C_j de \bar{A}^{-1} .

Para o vértice Y acima teremos

$$(2.3) \quad \Phi(Y) = |\lambda| + \sum_{i=n+1}^m |\phi_i(x)| + \lambda (A_i, C_j) \quad '$$

$$\text{Temos} \quad \Phi(x) = \sum_{i=1}^m |\phi_i(x)| =$$

$$= \sum_{i=1}^m |(A_i, x) - b_i| \quad \text{logo}$$

$$\Phi(Y) = \sum_{i=1}^m |(A_i, x + \lambda C_j) - b_i| =$$

$$= \sum_{i=1}^m |(A_i, x) - b_i + \lambda (A_i, C_j)| =$$

$$= \sum_{i=1}^m | \phi_i(x) + \lambda (A_i, C_j) | =$$

$$= \sum_{i=1}^n | \phi_i(x) + \lambda (A_i, C_j) | + \sum_{i=n+1}^m | \phi_i(x) + \lambda (A_i, C_j) |$$

como $\phi_i(x) = 0$ para $i = 1, \dots, n$

e

$$(A_i, C_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad \text{para } i = 1, \dots, n$$

Temos

$$\phi(Y) = |\lambda| + \sum_{i=n+1}^m | \phi_i(x) + \lambda (A_i, C_j) |.$$

Para todo λ suficientemente pequeno, podemos escrever:

$$\phi(Y) = |\lambda| + \sum_{i=n+1}^m [\phi_i(x) + \lambda (A_i, C_j)]$$

Observemos que $\phi(Y) = |\lambda|$ para $i = 1, \dots, n$,

isto é, o novo vértice Y satisfaz quase que totalmente as equações do vértice anterior, o pequeno erro $|\lambda|$ aparece exatamente na j -ésima equação que é eliminada pela translação do ponto x para $Y = X + \lambda \cdot C_j$.

Portanto, podemos concluir que cada novo vértice, obtido pela translação do vértice anterior, satisfaz $n - 1$ equações deste, e no mínimo mais uma nova equação das $m - n$ restantes.

Se $(A_i, C_j) = 0$ para $i = n + 1, \dots, m$

então $\phi_i(Y)$ são restos constantes sobre a extremidade j ,

pois:

$$\text{temos } \Phi(Y) = |\lambda| + \sum_{i=n+1}^m [\phi_i(x) + \lambda (A_i, C_j)]$$

como $(A_i, C_j) = 0$ $i = n + 1, \dots, m$

teremos

$$\Phi(Y) = |\lambda| + \sum_{i=n+1}^m \phi_i(x) \quad \text{onde}$$

$\phi_i(x) \neq 0$ para todo i , logo $\Phi(Y)$ é uma constante.

Por outro lado os zeros de $\phi_i(Y)$ são dados por:

temos

$$\phi_i(Y) = \phi_i(x) + \lambda (A_i, C_j) \quad \text{para } i = n + 1 \dots m$$

$$\phi_i(Y) = 0 \rightarrow \phi_i(x) + \lambda_{ij} (A_i, C_j) = 0 \rightarrow$$

$$\rightarrow \lambda_{ij} = \frac{-\phi_i(x)}{(A_i, C_j)}$$

Observemos que ϕ é uma função diferenciável em λ ao longo de todas as extremidades, exceto nos zeros de $\phi_i(Y)$ onde a derivada de ϕ vale

$$2 + (A_i, C_j) +$$

portanto teremos: $\phi'(0-) = -1 + \sum_{i=n+1}^m (A_i, C_j)$

$$\phi'(0+) = 1 + \sum_{i=n+1}^m (A_i, C_j)$$

Demonstração:

$$\text{temos: } \phi = \lambda + \sum_{i=n+1}^m [\phi_i(x) + \lambda (A_i, C_j)] +$$

$$+ \phi' = \pm 1 + \sum_{i=n+1}^m (A_i, C_j) +$$

$$+ \phi(0-) = -1 + \sum_{i=n+1}^m (A_i, C_j)$$

$$\phi'(0+) = 1 + \sum_{i=n+1}^m (A_i, C_j).$$

Só atingiremos um ponto x mínimo da função ϕ , ao longo da extremidade j , quando um dos $\phi'(0-)$ ou $\phi'(0+)$ é zero ou se ϕ' troca de sinal para $\lambda = 0$.

Se o referido ponto é mínimo para todo j , então x é um mínimo da função Φ ; logo, será solução ótima para o sistema (1.1).

A solução ótima será única, se as derivadas de Φ forem diferentes de zero no ponto de mínimo; caso contrário teremos várias soluções ótimas, isto é, se uma ou ambas derivadas forem nulas então o ponto de mínimo não será único.

Para atingirmos uma solução ótima do sistema de equações faremos várias iterações, que consistem no aproveitamento ou abandono de equações de forma conveniente, será descrita a seguir, de modo que cada novo vértice obtido minimize a função resíduo.

A aplicação do método consiste no seguinte:

Devemos começar o método por um ponto arbitrário, e suponhamos que seja solução do sistema (2.1) (sugerimos um ponto que seja solução, de pelo menos n das equações do sistema) e o aplicamos nas derivadas da função resíduo.

Se ambos $\Phi'(0+)$ e $\Phi'(0-)$ são negativos, e os λ_{ij} positivos calculados por (2.3) e ordenados pela relação de ordem menor ou igual (\leq), para $\Phi'(0+)$ nós adicionaremos sucessivamente os termos

$$2 \mid (A_i, C_j) \mid$$

até que a soma torne-se zero ou positiva.

Feito isto, o índice i para o qual ocorre o fato acima, determina o vértice Y que produz um mínimo de Φ sobre a extremidade.

A equação j é substituída pela equação i e o processo é repetido um núme-

ro finito de vezes até que ponto mínimo de Φ seja atingido.

Se ocorrer que, ambos $\Phi'(0-)$ e $\Phi'(0+)$ são positivos, ordenaremos os λ_{ij} negativos pela mesma relação anterior e,

para $\Phi'(0-)$ adicionaremos $-2 \mid (A_i, C_j) \mid$

até que a soma torne-se nula ou negativa, e o restante do processo é análogo ao anterior.

Notemos que o processo apresentado pode ser aplicado somente para sistemas cujos vértices são ordinários.

Agora, estudaremos as modificações aplicadas no método anterior; de modo que o novo método solucionará sistemas cujos vértices são degenerados.

3. DEGENERACÃO:

Suponhamos que x seja um vértice degenerado para o sistema dado por (1.1).

Logo, temos:

$$\phi_i(x) = 0 \quad (i = 1, \dots, n)$$

e ainda para um inteiro positivo P , $0 < P < m-n$

temos $\phi_i(x) = 0$ para $i = n+1, \dots, n+P$

e

$$\phi_i(x) > 0 \quad \text{para} \quad n+P < i \leq m.$$

Aplicando a seguinte perturbação teremos um problema modificado, para o qual x é um vértice ordinário.

As perturbações são feitas da seguinte maneira: para algum pequeno parâmetro positivo u e n vetor cujos componentes $n_i > 0$, faremos

$$b_i = \begin{cases} b_i - u n_i & (i = n+1, \dots, n+P) \\ b_i & (i = n+P+1, \dots, m) \end{cases}$$

Aplicando as modificações acima na função resíduo teremos:

$$\begin{aligned} \Phi_u(x) &= \sum_{i=1}^m |(\Lambda_i, x) - b_i| = \\ &= \sum_{i=1}^m |(\Lambda_i, x) - b_i| + \sum_{i=n+1}^{n+P} |(\Lambda_i, x) - b'_i| + \\ &+ \sum_{i=n+P+1}^m |(\Lambda_i, x) - b_i| \end{aligned}$$

Fazendo a translação $y = x + \lambda C_j$ obtemos

$$\begin{aligned} \Phi_u(y) &= \sum_{i=1}^n |(\Lambda_i, x + \lambda C_j) - b_i| + \\ &+ \sum_{i=n+1}^{n+P} |(\Lambda_i, x + \lambda C_j) - b'_i| + \\ &+ \sum_{i=n+P+1}^m |(\Lambda_i, x + \lambda C_j) - b_i| = \end{aligned}$$

$$\begin{aligned}
 &= \sum_{i=1}^n \left[(A_i, x) - b_i + \lambda (A_i, C_j) \right] + \\
 &+ \sum_{i=n+1}^{n+P} \left[(A_i, x) - b_i + u n_i + \lambda (A_i, C_j) \right] + \\
 &+ \sum_{i=n+P+1}^m \left[(A_i, x) - b_i + \lambda (A_i, C_j) \right] = \\
 &= \sum_{i=1}^n \left[\phi_i(x) - \lambda (A_i, C_j) \right] + \\
 &+ \sum_{i=n+1}^{n+P} \left[\phi_i(x) + u n_i + \lambda (A_i, C_j) \right] + \\
 &+ \sum_{i=n+P+1}^m \left[\phi_i(x) + \lambda (A_i, C_j) \right]
 \end{aligned}$$

Como $\phi_i(x) = 0$ para $i = 1, \dots, n+P$

$$(A_i, C_j) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

Finalmente, teremos:

$$\Phi_u(Y) = |\lambda| + \sum_{i=n+1}^{n+P} \left[u n_i + \lambda (A_i, C_j) \right] +$$

$$+ \sum_{i=n+p+1}^m \phi_i(x) + (A_i, C_j)$$

Notemos que se $(A_i, C_j) = 0$ então $\phi_u(Y)$ é uma função constante sobre a extremidade j , logo podemos concluir que neste caso a solução ótima não é única.

Observemos que ϕ_u é diferenciável em para todas as direções j , menos nos zeros de $\phi_i(Y)$.

No caso em que $(A_i, C_j) \neq 0$ os zeros são dados por:

Temos que

$$\phi_i(Y) = \begin{cases} (A_i, Y) - b'_i & \text{para } i = n+1, \dots, n+p \\ (A_i, Y) - b_i & \text{para } i = n+p+1, \dots, m \end{cases}$$

onde $Y = x + \lambda C_j$.

a) Cálculo dos zeros de $\phi_i(Y)$ para $i = n+1, \dots, n+p$

Sabemos que $b'_i = b_i - u n_i$

então:

$$\begin{aligned} \phi_i(Y) &= (A_i, x + \lambda C_j) - b_i + u n_i = \\ &= \phi_i(x) + \lambda (A_i, C_j) + u n_i \end{aligned}$$

Como $\phi_i(x) = 0$ para $i = n+1, \dots, n+P$

logo, os zeros de $\phi_i(Y)$ serão dados por:

$$(3.3) \quad \lambda_{ij} = \frac{-u_i n_i}{(A_i, C_j)} \quad \text{para } i = n+1, \dots, n+P$$

b) Cálculo dos zeros de $\phi_i(Y)$ para $i = n+P+1, \dots, m$

$$\phi_i(Y) = (A_i, x + \lambda C_j) - b_i = \phi_i(X) + \lambda (A_i, C_j)$$

Finalmente, os zeros de $\phi_i(Y)$ serão dados por:

$$(3.4) \quad \lambda_{ij} = \frac{-\phi_i(X)}{(A_i, C_j)} \quad \text{para } i = n+P+1, \dots, m.$$

Verifiquemos que todos λ_{ij} dados em (3.3) poderão assumir pequenos valores em módulo, para isso bastando escolher u suficientemente pequeno, enquanto que os demais λ_{ij} dados em (3.4) assumirão pequenos valores quando x for um ponto ótimo, isto porque $\phi_i(x)$ será o mínimo possível.

Quando estamos procurando um mínimo sobre uma determinada extremidade j , nós podemos dizer que os hiperplanos correspondentes aos índices $i = n+1, \dots, n+P$, são encontrados primeiro, e ainda podemos ajustar os valores de u_i , de modo que estes hiperplanos sejam colocados numa ordem específica.

Observemos que são os valores dos λ_{ij} que determinam o passo da translação do ponto x ; logo, se a direção de procura do mínimo for caracterizada por λ positivo, devemos apenas utilizar os índices i , para os quais $(A_i, C_j) < 0$, e se λ for negativo procederemos de maneira similar.

Quanto à forma de procura de um mínimo ser realizada de acordo com o crescimento do índice i , visa facilitar a programação, e ainda por ser desconhecida uma técnica melhor.

Suponhamos que a perturbação u_{n_i} foi aplicada para $i = n+1 - \dots - n+P$, o teste para um mínimo é aplicado sucessivamente ao longo de cada extremidade, a partir do ponto x .

Se o teste é satisfeito para todos os casos, x será declarado ponto mínimo de $R(x)$.

Mas, se a procura for desenvolvida por uma extremidade j , em que x não é ponto de mínimo, e cujo λ_{ij} é positivo, não é feita a troca de sinal para a equação j . Por outro lado, se algum hiperplano i tenha sido percorrido, o sinal de b_i e das componentes de A_i são trocados.

Agora, se o teste para um mínimo for satisfeito para algum membro do conjunto de equações correspondentes a $(n+1) \leq i \leq n+P$, não é feita a troca de sinal da equação junto a qual o mínimo for encontrado, e a permutação da equação é feita da maneira usual.

Mas, a procura de um mínimo é feita na direção em que λ é negativa, procederemos de maneira similar, com a diferença de que neste caso os sinais de b_j e da componente A_j são trocados.

Se porventura todas as equações correspondentes aos índices $n+1 \leq i \leq n+P$ tenham sido utilizadas, e o mínimo não for encontrado, o ponto x , correspondente à extremidade j não será um mínimo do sistema original, e uma permutação de equações é feita de modo usual; sendo aplicado o teste para verificar se o referido é degenerado ou ordinário.

Para o método acima citado, o ajustamento da perturbação pode ser realizado durante o processamento da computação; isto abre a possibilidade para que o método termine com um número finito de ciclos, cada um com a decisão de que um deles é um mínimo, ou pedindo por uma permutação de equações e consequentemente um novo vértice.

4. EXEMPLO

Dado o sistema

$$\begin{cases} x_1 + x_2 = 1 \\ x_1 + 2x_2 = 1 \\ x_1 + 3x_2 = 2 \\ x_1 + 4x_2 = 3 \\ x_1 + 5x_2 = 2 \end{cases}$$

Calcular a melhor solução usando a norma L_1

$$\min \phi(x) = \sum_{i=1}^m |\phi_i(x)| = \sum_{i=1}^m |(A_i, x) - b_i| \quad \text{onde}$$

A_i é i -ésima linha de A .

$$(1) \quad x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \bar{A} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \quad \bar{b} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

(2) Vetor direção E_k

$$\bar{A} E_k = \ell_k \rightarrow E_k = \bar{A}^{-1} \ell_k \quad k = 1, 2 \quad \text{onde } \ell_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \ell_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\bar{A}^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \rightarrow E_1 = \begin{pmatrix} 2 \\ -1 \end{pmatrix} = C_1 \quad E_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} = C_2$$

(3) $Y = x + \lambda C_j$

$$\lambda = \lambda_{ij} = \frac{-\phi_i(x)}{(A_i, C_j)} \rightarrow \lambda_{ij} = 0 \quad \text{para } i = 1, 2 \quad \text{pois } \phi_i(x) = 0$$

$$\lambda_{31} = \frac{-\phi_3(x)}{(A_3, C_1)} = 1 / (1 \ 3) \begin{pmatrix} 2 \\ -1 \end{pmatrix} = 1/1 = 1$$

$$\lambda_{41} = \frac{-\phi_4(x)}{(A_4, C_1)} = 2 / (1 \ 4) \begin{pmatrix} 2 \\ -1 \end{pmatrix} = 2 / -2 = -1$$

$$\lambda_{51} = \frac{-\phi_5(x)}{(A_5, C_1)} = 1 / (1 \ 5) \begin{pmatrix} 2 \\ -1 \end{pmatrix} = 1 / -3 = -1/3$$

$$\lambda_{32} = 1/2 \quad \lambda_{42} = 2/3 \quad \lambda_{52} = 1/4$$

$$1/4 \leq 1/2 \leq 2/3 \leq 1 \quad \text{para} \quad \lambda_{ij} \geq 0$$

$$-1 \leq -1/3 \quad \text{para} \quad \lambda_{ij} \leq 0$$

(4) Teste de otimalidade

$$\text{se } \phi'(0-) < 0 \quad \text{e} \quad \phi'(0+) > 0$$

então x é ponto ótimo, isto para todas extremidades j

$$\phi'(0-) = -1 + \sum_{i=k+1}^m (A_i, C_j) \quad \begin{matrix} j = 1, 2 \\ k = 2 \end{matrix}$$

$$\phi'(0-) = -1 + (A_3, C_1) + (A_4, C_1) + (A_5, C_1) = -1 + 1 - 1 - 3 = -4$$

$$\rightarrow \phi'(0-) < 0$$

$$\phi'(0+) = 1 + \sum_{i=k+1}^m (A_i, C_j) \quad , \quad \begin{matrix} j = 1, 2 \\ k = 2 \end{matrix}$$

$$\Phi(0+) = 1 + (A_3, C_1) + (A_4, C_1) + (A_5, C_1) = 1 + 1 - 1 - 3 = -2$$

$$\rightarrow \Phi(0+) = -2 < 0$$

$$\text{logo } x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ não é ótimo.}$$

(5) Adicionaremos sucessivamente os termos $2|(A_i, C_j)|$ onde os índices i, j são determinados pelos $\lambda_{ij} \geq 0$ começando do menor até que $\Phi'(0+)$ torne-se zero ou positivo.

Feito isto, o índice i para o qual ocorre o fato acima, determina o vértice Y sobre a extremidade j .

$$\Phi(0+) = -4 + 2|(A_5, C_2)| = -4 + 2.4 = 4 > 0$$

$$Y = X + \lambda_{52} C_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 1/4 \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}$$

$$\rightarrow Y = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}$$

(b) A equação j é substituída pela i no sistema, vã para (1)

$$\left\{ \begin{array}{l} x_1 + x_2 = 1 \\ x_1 + 5x_2 = 2 \\ x_1 + 3x_2 = 2 \\ x_1 + 4x_2 = 3 \\ x_1 + 2x_2 = 1 \end{array} \right.$$

$$Y = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix} - 1 \begin{pmatrix} -1/4 \\ 1/4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

portanto $Y = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ que é a solução inicial .

Portanto o método entrou em ciclagem

$$\text{mas } \phi \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 4 \quad \text{e} \quad \phi \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix} = 7/4 = 1,75.$$

∴ Portanto $Y = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}$ é a melhor solução.

Observação - Apesar do trabalho 5 afirmar que a ciclagem não ocorre, observamos pelo exemplo acima que ela ocorreu. Portanto, o algoritmo deve ser modificado para prever essa possibilidade.

$$\bar{A} = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \rightarrow \bar{A}^{-1} = \begin{bmatrix} 5/4 & -1/4 \\ 1/4 & 1/4 \end{bmatrix}$$

$$C_1 = \begin{pmatrix} 5/4 \\ -1/4 \end{pmatrix} \begin{pmatrix} -1/4 \\ 1/4 \end{pmatrix} = C_2$$

$$\lambda_{31} = \frac{-\phi_3(x)}{(A_3, C_1)} = \frac{1/2}{1/2} = 1$$

$$\lambda_{41} = \frac{-\phi_4(x)}{(A_4, C_1)} = \frac{5/4}{1/4} = 5$$

$$\lambda_{51} = \frac{-\phi_5(x)}{(A_5, C_1)} = \frac{-1/4}{3/4} = -1/3$$

$$\lambda_{32} = \frac{-\phi_3(x)}{(A_3, C_2)} = \frac{1/2}{1/2} = 1$$

$$\lambda_{42} = \frac{-\phi_4(x)}{(A_4, C_2)} = \frac{5/4}{3/4} = 5/3$$

$$\lambda_{52} = \frac{-\phi_5(x)}{(A_5, C_2)} = \frac{-1/4}{1/4} = -1$$

$$1 \leq 5/3 \leq 5, \quad -1 \leq -1/3$$

$$\Phi'(0-) = -1 + (A_3, C_1) + (A_4, C_1) + (A_5, C_1) =$$

$$= -1 + 1/2 + 1/4 + 3/4 = 1/2 > 0$$

$$\Phi(0+) = 1 + 1/2 + 1/4 + 3/4 > 0$$

$$\Phi'(0-) = 1/2 - 2|(A_5, C_2)| = 1/2 - 1/2 = 0$$

CAPÍTULO IV

UM MÉTODO PARA RESOLUÇÃO DE PROBLEMA DE APROXIMAÇÃO LINEAR NA NORMA L_1 .

[1] [2]

1. INTRODUÇÃO

Consideremos o sistema de equações lineares

$$(1) \quad Ax = b$$

onde A é uma matriz real de ordem $n \times m$, de posto $k \leq m < n$ e b é um n -vetor.

Nosso objetivo é determinar um m vetor x que minimize a norma L_1 dada por:

$$\text{Min } \Phi(x) = \sum_{i=1}^n |r_i(x)| = \sum_{i=1}^n |a_i x - b_i|$$

Esta fórmula pode ser reduzida para um problema de programação linear equivalente dada por:

Dual

$$\text{MAX } Z = \sum_{i=1}^n b_i (d_i - 1)$$

$$\text{S.a} \quad A^T d = \sum_{i=1}^n a_i^T \quad (I)$$

$$\text{com } 0 \leq d_i \leq 2 \quad i = 1, \dots, m \quad (II)$$

Antes de começarmos a descrição do método faz-se necessária a apresentação de algumas definições e resultados, para um melhor entendimento do processo.

Definiremos a base pelo conjunto d e consideremos

$I(d) \subset \{1, \dots, n\}$ conjunto dos índices para o qual o conjunto

$\{d_i \mid i \in I(d)\}$ são denominados variáveis básicas.

Indicaremos por $L(d) = \{i \mid d_i = 0\}$ e $U(d) = \{i \mid d_i = 2\}$,

isto é, são os conjuntos dos índices das variáveis não básicas onde estas assumem, respectivamente, os seus limites inferiores e superiores: D denotará a matriz básica, e as variáveis básicas serão indicadas por

$$d_D = \{d_{Di}\} \quad i = 1, \dots, m,$$

as quais serão obtidas a partir de (I) onde A_i indica a i -ésima coluna da matriz A .

$$e \quad d_D = D^{-1} \left[\sum_{j=1}^n A_j^T - 2 \sum_{i \in U(d)} A_i \right] = d_{B_0} - 2 \sum_{i \in U(d)} Y_i \quad (II)$$

Observemos que, este resultado é válido desde que algumas das suas variáveis não básicas possam assumir seus limites superiores ($=2$).

Demonstração de (II):

$$\text{Temos} \quad A^T d = \sum_{i=1}^n A_i^T$$

fazendo

$$A^{\Gamma} = \left| \begin{array}{c} A^{I(d)} : A^{U(d)} \cup L(d) \end{array} \right|$$

e

$$d = \left| \begin{array}{c} d_j : d_i \end{array} \right| \quad \text{para} \quad \begin{array}{l} j \in I(d) \\ i \in L(d) \cup u(d) \end{array}$$

Aplicando em (I) temos:

$$(A^{\Gamma})^{I(d)} d_j + (A^{\Gamma})^{u(d)} \cup L(d) d_i = \sum_{j=1}^n c_j^{\Gamma}$$

como

$$d_j = d_D \quad \text{para} \quad j \in I(d)$$

e

$$(A^{\Gamma})^{I(d)} = D, \text{ ou seja, matriz b\'asica. Ent\~ao } \exists D^{-1} \text{ inversa de } D.$$

Substituindo:

$$D d_D + \sum_{i \in L(d)} A_{i1}^{\Gamma} d_i + \sum_{i \in u(d)} A_{i1}^{\Gamma} d_i = \sum_{j=1}^n A_{j1}^{\Gamma}$$

como

$$d_1 = 0 \quad \text{para} \quad i \in L(d)$$

$$d_1 = 2 \quad \text{para} \quad i \in U(d)$$

$$m_D + 2 \sum_{i \in U(d)} A_i^r = \sum_{j=1}^n A_j^r \quad \rightarrow$$

$$\rightarrow d_D = D^{-1} \left[\sum_{j=1}^n A_j^r - 2 \sum_{i \in U(d)} A_i^r \right] = d_{D_0} - 2 \sum_{i \in U(d)} y_i$$

Introduziremos agora:

TEOREMA 1:

Uma condição necessária e suficiente para que um programa não nulo seja valente ao sistema $Ax = b$ seja ótimo; é que $(n - m)$ elementos de d_D cada qual tendo o valor zero (limite inferior) ou 2 (limite superior) e que os outros m elementos sejam variáveis básicas.

TEOREMA 2:

Uma solução básica é máxima se os parâmetros

$\{ z_i - b_i \} \quad i \notin I(d)$ satisfazem as relações

$$z_i - b_i \geq 0 \quad i \in L(d)$$

$$z_i - b_i \leq 0 \quad i \in u(d)$$

onde $z_i = b_D^r \cdot y_i$

Observação 1: Temos $z = \sum_{i=1}^n b_i (d_i - 1)$

que também pode ser expresso por:

$$z = \sum_{i \in I(d)} b_i (d_i - 1) - \sum_{i \in L(d)} b_i + \sum_{i \in u(d)} b_i$$

que é decorrente do Teorema 1 pois,

$$\begin{aligned} z &= \sum_{i=1}^n b_i (d_i - 1) = \sum_{i \in I(d)} b_i (d_i - 1) + \\ &+ \sum_{i \in L(d)} b_i (d_i - 1) + \sum_{i \in u(d)} b_i (d_i - 1) \end{aligned}$$

Mas, como

$$d_i = 0 \quad \text{para} \quad i \in L(d)$$

$$d_i = 2 \quad \text{para} \quad i \in u(d)$$

Finalmente:

$$z = \sum_{i \in I(d)} b_i (d_i - 1) + \sum_{i \in L(d)} b_i + \sum_{i \in u(d)} b_i$$

TEOREMA 3:

Consideremos uma certa iteração a partir de uma solução básica, e seja

$\{ z_i - b_i \}$ para $i = 1, 2, \dots, n$ o custo marginal.

Então, a sequência de colunas não básicas, as quais poderão entrar na base e conseqüentemente poderão sair da mesma, é dado por:

1) Se $d_{D_i} > 2$ a sequência é dada pelas colunas correspondentes aos parâmetros

$$\pi_r = (z_r - b_r) / y_{ir} > 0 \quad r \notin I(d),$$

começando a partir do menor, onde y_{ir} são pivôs (ver item 4).

2) Se $d_{D_i} < 0$, então a sequência é dada pelas colunas correspondentes aos parâmetros

$$\theta_r = (z_r - b_r) / y_{ir} < 0 \quad r \notin I(d),$$

começando a partir do algebricamente maior.

2. O ALGORITMO DUAL SIMPLEX.

(1) Para todo $z_i - b_i < 0 \quad i \in L(d)$ fazemos $b_i = 2$, e colocaremos um sinal sobre a coluna correspondente.

Calcule $d_D = d_{D_0} - 2 \sum_{i \in U(d)} y_i$ e vá para (2).

(2) Se todo d_{D_ℓ} satisfaz $0 \leq d_{D_\ell} \leq 2$

uma solução ótima foi atingida, caso contrário vá para (3).

(3) Examinar d_{D_ℓ} para $\ell = 1, 2, \dots, m$ e considerar

$$d_{r_1} = \text{Min} \{ d_1, d_2 \} \quad \text{onde}$$

$$d_1 = \text{Min}_\ell \{ d_{D_\ell} ; d_{D_\ell} < 0 \} \quad \text{e}$$

$$d_2 = \text{Min}_\ell \{ 2 - d_{D_\ell} , d_{D_\ell} > 2 \}$$

Caso 1' -

$\bar{d}_{D_i}^r < 0$ e A_r^Γ é determinado pelo Teorema 3.

(3.1') Se $\bar{y}_{ir} < 0$ não trocamos d'_D e vá para (3.5).

(3.2') Se $\bar{y}_{ir} > 0$, adicionar $2y_r$ em d'_D e vá para (3.5). Remova a marca para coluna A_r^Γ ; isto indica que d_r não é mais limite superior. Vá para (3.5).

Caso 2' -

$\bar{d}_{D_i}^r > 2$ e A_r^Γ é determinado pelo Teorema 3.

(3.3'). Se $\bar{y}_{ir} > 0$, subtrair $2y_j$ de d'_D e coloque uma marca sobre a coluna A_j^Γ para indicar que ela é, agora, seu limite superior. Vá para (3.5).

(3.4'). Se $\bar{y}_{ir} < 0$, adicionamos $2y_r$ e subtraímos $2y_j$ de d'_D .

Renova a marca de A_r^Γ e coloque a marca na coluna A_j^Γ . Vá para (3.5).

(3.5) Calcule \bar{d}'_{Di} a partir de:

$$(a) \quad \bar{d}'_{D_i} = d_{D_i} \quad e \quad \bar{y}_{it} = y_{it} \quad t = 1$$

$$(b) \quad \bar{d}'_{D_i} = d'_{D_i} / y_{i,t-1} \quad e \quad \bar{y}_{it} = y_{it} / y_{i,t-1}$$

$$t = 2, 3, \dots, k.$$

Aqui \bar{d}'_{D_i} e y_{it} representam a i -ésima componente das variáveis básicas e o pivô na tabela $t = 1, 2, \dots, k$. O vetor d'_D é o vetor d_D adicionando algum $2y_r$ ou subtraindo algum $2y_j$; o elemento \bar{d}'_{D_i} é a i -ésima componente.

Se a solução não satisfaz a condição $0 \leq d_i \leq 2$, substituímos A_j^Γ por A_r^Γ e vá para o Caso 1' ou Caso 2', dependendo do novo \bar{d}'_{D_i} se é < 0 ou > 2 , respectivamente.

(2) A mudança da tabela pode ser adiada até que o posterior $2Y_i$ tenha sido adicionado ou $2Y_j$ tenha sido subtraído de d_{D_i} , ou seja, até que algumas colunas não básicas, as quais poderão entrar e posteriormente deixar a base correspondente ao d_{D_i} que não satisfaz a condição (II).

Este procedimento continuará até que a última coluna não básica para a qual d_{D_i} satisfaz (II); com isto garantimos o máximo decresce da função z .

Assumimos que o processo será iniciado a partir de uma solução básica inicial, dada pela tabela simplex.

Como veremos, no exemplo, esta corresponde à tabela (1).

As etapas acima sugerem que, não necessitamos calcular $(k - 1)$ tabelas intermediárias para $t = 2, 3, \dots, k$, que correspondem $(k - 1)$ colunas não básicas que entram e posteriormente deixam a base; necessitando apenas da tabela $t = k + 1$.

Mas, para obtenção da tabela $t = k + 1$ necessitamos conhecer o valor dos parâmetros $\{d_{D_i}\}$ e dos elementos pivôs $\{y_{it}\}$ para as tabelas intermediárias que são dadas, respectivamente, por

$$(d_{D_i} / y_{i,t-1}) \quad \text{e} \quad (y_{it} / y_{i,t-1}) \quad \text{onde}$$

$y_{i,t-1}$ é o pivô da tabela anterior.

Isto pode ser resumido no seguinte:

Se a solução satisfaz a condição $0 < d_i < 2$, a mudança da tabela é feita da maneira usual. Vá para (2).

3. CASO DE DEGENERACÃO

Não há provisões iniciais da ocorrência de degeneração, mas se surgir dificuldade do tipo, uma ou mais colunas não básicas A_r^r para a qual $(z_r - b_r) / y_{ir} = 0$, isto implicará que $\pi_r = 0$ ou $\theta = 0$. Logo, estamos diante de uma degeneração que dificultará a escolha de uma coluna não básica A_r^r que entrará na base, uma vez que ela depende exclusivamente de $\pi_r > 0$ ou $\theta_r < 0$.

Esta dificuldade poderá ser resolvida da seguinte maneira:

Se $d_r = 0$, substituiremos $(z_r - b_r)$ por um pequeno número δ e se $d_r = 2$ substituiremos $(z_r - b_r)$ por $-\delta$.

Feito isto, nós recalculamos o raio $(z_r - b_r) / y_{ir}$ que obviamente será > 0 ou < 0 e aplicamos o algoritmo normalmente, já que o impasse mencionado está resolvido.

4. ELUCIDAÇÃO DO MÉTODO

O processo consiste em etapas:

(1) A aplicação da eliminação de Gauss Jordan poderá ser aplicada após adição de $2y_r$ ou subtração de y_j .

$$\bar{d}'_{D_i} = d_{D_i} \quad \text{e} \quad \bar{y}_{it} = y_{it} \quad \text{para} \quad t = 1$$

$$\bar{d}_{D_i} = d'_{D_i} / y_{i,t-1} \quad \text{e} \quad \bar{y}_{it} = y_{it} / y_{i,t-1}$$

$$\text{para} \quad t = 2, 3, \dots, k$$

onde \bar{d}'_{D_i} e \bar{y}_{it} representam a i -ésima componente da variável básica;
é o pivô na tabela $t = 1, 2, \dots, k$.

Nós também, necessitamos conhecer a sequência das k -colunas, que entrarão ou eventualmente poderão sair da base em uma iteração; estas serão das pelo Teorema 3.

Com isto, estamos aptos à aplicação do método que nos renderá um resultado satisfatório e com poucas dificuldades.

5. EXEMPLO

Dado o sistema:

$$x_1 + x_2 = 1$$

$$x_1 + 2x_2 = 1$$

$$x_1 + 3x_2 = 2$$

$$x_1 + 4x_2 = 3$$

$$x_1 + 5x_2 = 2$$

passamos o sistema para um problema de programação linear equivalente.

$$\text{MÁX } z = d_1 + d_2 + 2d_3 + 3d_4 + 2d_5 - 7$$

$$\text{s.t.} \quad d_1 + d_2 + d_3 + d_4 + d_5 = 5$$

$$d_1 + 2d_2 + 3d_3 + 4d_4 + 5d_5 = 15$$

$$0 \leq d_i \leq 2 \quad i = 1, \dots, 5.$$

APLICAÇÃO DO ALGORITMO

TABELA SIMPLEX

(0)

d_i	1	1	2	3	2	-7
$\sum_{j=1}^5 a_{ij}$	a_1	a_2	a_3	a_4	a_5	
5	1	1	1	1	1	
15	1	2	3	4	5	

Continuação da TABELA SIMPLEX

-5	1	0	-1	-2	-3
10	0	1	2	3	4

$$d_{D_0} = \begin{pmatrix} -5 \\ 10 \end{pmatrix} \quad Y_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad Y_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$Y_3 = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \quad Y_4 = \begin{pmatrix} -2 \\ 3 \end{pmatrix} \quad Y_5 = \begin{pmatrix} -3 \\ 4 \end{pmatrix}$$

$$(1) \quad Z_1 = b_D^T Y_1 \rightarrow Z_1 = b_D^T Y_1 \rightarrow Z_1 = (1, 1) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1$$

$$Z_1 = 1 \quad Z_2 = 1 \quad Z_3 = 1 \quad Z_4 = 1 \quad Z_5 = 1$$

$$d_D = d_{D_0} - 2 \sum_{i \in U(d)} Y_i$$

$$\text{se } Z_1 - b_1 < 0 \rightarrow i \in U(d)$$

$$Z_3 - b_3 = 1 - 3 = -2 < 0 \rightarrow 3 \in U(d)$$

$$Z_4 - b_4 = 1 - 3 = -2 < 0 \rightarrow 4 \in U(d)$$

$$Z_5 - b_5 = 1 - 2 = -1 < 0 \rightarrow 5 \in U(d)$$

$$U(d) = \{ 3, 4, 5 \}.$$

$$d_D = \begin{pmatrix} -5 \\ 10 \end{pmatrix} - 2 \sum_{i=3}^5 Y_i = \begin{pmatrix} -5 \\ 10 \end{pmatrix} - 2 \begin{pmatrix} 6 \\ 9 \end{pmatrix} =$$

$$= \begin{pmatrix} -5 \\ 10 \end{pmatrix} + \begin{pmatrix} 12 \\ -18 \end{pmatrix} = \begin{pmatrix} 7 \\ -8 \end{pmatrix}$$

$$d_D = \begin{pmatrix} 7 \\ -8 \end{pmatrix}$$

$$(2) \quad d_D = \begin{pmatrix} 7 \\ -8 \end{pmatrix} \rightarrow d_{D_1} = 7 \quad d_{D_2} = -8$$

$d_{D_1} \notin [0, 2]$ e $d_{D_2} \notin [0, 2]$ logo esta solução não é ótima.

$$(3) \quad d_1 = -8 \quad d_2 = -5$$

$$d_{D_2} = -8$$

Caso 1'

$$d_D = -8 < 0 \quad \text{usando o Teo } \textcircled{3}$$

$$\theta_r = (z_r - b_r) / Y_{ir} < 0 \quad r \notin I(d).$$

$$\theta_3 = (z_3 - b_3) / Y_{23} = -2/2 = -1 < 0$$

$$\theta_4 = (z_4 - b_4) / Y_{24} = -2/3 = -2/3 < 0$$

$$\theta_5 = (Z_5 - b_5) / Y_{25} = -1/4 < 0$$

(3.2') Tomando o θ_r algebricamente menor

$$Y_{23} > 0$$

$$d'_D = d_D + 2Y_3 = \begin{pmatrix} 7 \\ -8 \end{pmatrix} + 2 \begin{pmatrix} -1 \\ 2 \end{pmatrix} = \begin{pmatrix} 5 \\ -4 \end{pmatrix}$$

(3.5')

$$(a) \quad \bar{d}'_{D_1} = d'_{D_1} \quad e \quad Y_{it} = Y_{it} \quad t = 1$$

$$(b) \quad \bar{d}'_{D_1} = d'_{D_1} / Y_{i,t-1} \quad e \quad \bar{Y}_{it} = Y_{it} / Y_{i,t-1} \quad t = 2, 3, 4$$

$$\bar{d}'_{D_1} = d'_{D_1} = 5$$

$$\bar{Y}_{11} = Y_{11} = 1$$

$$\bar{d}'_{D_1} = 5/1 = 5$$

$$\bar{d}'_{D_2} = d'_{D_2} = -4$$

$$Y_{21} = 2$$

$$\bar{d}'_{D_2} = -4/2 = -2$$

$$d_D = \begin{pmatrix} 5 \\ -2 \end{pmatrix}$$

$$(2) \quad d_1 = -2 \qquad d_2 = 2 - 5 = -3$$

$$d_{D_1} = -3 < 0$$

$$\theta_4 = -2/-2 = 1 > 0 \qquad (\text{não entra nenhuma coluna}).$$

$$\theta_5 = -1/-3 = 1/3 > 0$$

$$d_{D_2} = -2$$

$$\theta_4 = -2/3 < 0$$

$$\theta_5 = -1/4 < 0$$

$$\boxed{r = 4} \qquad \boxed{i = 2}$$

$$(3.2') \quad Y_{34} = 3 > 0$$

$$d_D = \begin{pmatrix} 5 \\ -2 \end{pmatrix} + 2Y_4 = \begin{pmatrix} 5 \\ -2 \end{pmatrix} + 2 \begin{pmatrix} -2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

(3.5)

$$d_{D_1} = 1 \qquad d_{D_2} = 2$$

$$0 \leq d_{D_1} \leq 2 \qquad 0 \leq d_{D_2} \leq 2$$

Solução ótima para o problema dual foi atingida e é dada por:

$$d_D = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Para obtenção da melhor solução do sistema original basta resolver o sistema formado pela 2a. e 4a. equação

$$\begin{cases} x_1 + x_2 = 1 \\ x_1 + 4x_2 = 3 \end{cases} \rightarrow \begin{cases} x_1 = 1/3 \\ x_2 = 2/3 \end{cases}$$

Solução ótima $x = (1/3, 2/3)$.

$$\phi(x) = \sum_{i=1}^n |\phi_i(x)| = \sum_{i=1}^n |a_i x - b_i|$$

$$\phi(x) = 5/3 = 2,6.$$

CAPÍTULO V

ALGORITMO MELHORADO PARA APROXIMAÇÃO LINEAR DISCRETA L_1 . [3]

1. COMENTÁRIOS PRELIMINARES:

O algoritmo a ser estudado trata-se de uma modificação do método simplex; de modo que irá impor um mínimo de restrições, e com a vantagem de escolher as funções aplicações lineares; ainda foi descoberto o modo de passar totalmente várias vizinhanças do vértice simplex numa iteração se parada para problemas do tipo

$$(2) \quad \begin{array}{ll} \text{MIN} & \sum_{i=1}^m (u_i + v_i) \\ \text{S.a.} & \left\{ \begin{array}{l} b_i = \sum_{j=1}^n (r_j - s_j) a_{ij} + u_i - v_i \\ i = 1, 2, \dots, m \\ r_j, s_j, u_i, v_i \geq 0. \end{array} \right. \end{array}$$

Isto, naturalmente, nos permite concluir que este método é um dos mais eficientes da literatura de Matemática Aplicada na resolução de problemas de aproximação linear.

2. APRESENTAÇÃO DO ALGORITMO:

O algoritmo trata-se de uma modificação do método simplex, o qual solucionará problemas de aproximação por interpolação a um conjunto de pontos dados, através de uma função linear.

Observando o problema de programação linear (2), este revela que:

- (1) Uma solução básica inicial factível é imediatamente disponível;
- (2) Apenas n colunas são necessárias para conter as informações do lado direito da igualdade construída.

Denotaremos as colunas da tabela simplex correspondente ao problema (2) por R, r_j, s_j, u_i, v_i , com a base inicial sendo formada por u_1, \dots, u_m sempre que cada b_i é não negativo.

Caso isto aconteça trocaremos o sinal da correspondente linha e substituiremos na base u_i por v_i .

Claramente notemos que $r_j = -s_j$ e $u_i = -v_i$ e que a soma do custo marginal de r_j e s_j é zero e de u_i e v_i é -2 .

O algoritmo passa por dois estágios que consistem no seguinte:

Estágio 1 - Restrito à escolha das colunas pivôs durante as n primeiras iterações dos vetores r_j e s_j .

O vetor escolhido para entrar na base é aquele que possui maior custo marginal não negativo. E o vetor candidato a deixar a base será escolhido dentre os vetores u_i e v_i que causam redução máxima na função objetivo.

O posto k da matriz é conhecido no final do estágio 1; este é dado pelo número de vetores r_j e s_j pertencentes à base.

Uma vez que k dos vetores u_i ou v_i tenham sido removidos da base isto significa que; a tabela correspondente representa uma aproximação que interpola um conjunto de k pontos dados.

Se a aproximação interpola mais do que k pontos, então estamos diante de uma degeneração que não causará problema algum na prática.

Estágio 2 - Envolve a permutação dos vetores não básicos u_i ou v_i com as básicos u_i ou v_i ; os vetores básicos r_j e s_j não poderão deixar a base durante o estágio 2. Quanto à entrada e saída do vetor da base aplica-se a mesma regra do estágio 1.

O algoritmo termina quando todos os custos marginais são não negativos.

Observemos o seguinte fato: Que a principal modificação do método simplex é na escolha dos vetores u_i ou v_i para sair da base. Em ambos os estágios 1 e 2 o vetor escolhido é aquele que causa maior redução na função objetivo.

Faremos agora uma apresentação detalhada do algoritmo para seu melhor entendimento, isto é:

- (a) Estágio 1 refere-se às primeiras n interações e Estágio 2 refere-se a todas subsequentes iterações.

- (b) A iniciação da tabela é executada pela entrada dos dados corresponden-
do à tabela 1, cuja dimensão é de $(m + 2) \times (n + 2)$.

TABELA 1

Tabela condensada simplex inicial assumindo que cada b_i é não
negativo.

BASE	R	r_1	r_2	-----	r_n
u_1	b_1	a_{11}	a_{12}		a_{1n}
u_2	b_2	a_{21}	a_{22}		a_{2n}
\vdots	\vdots	\vdots	\vdots		\vdots
u_m	b_m	a_{m1}	a_{m2}		a_{mn}
Custo	$\sum_{i=1}^m f_i$	$\sum_{i=1}^m a_{i1}$	$\sum_{i=1}^m a_{i2}$	-----	$\sum_{i=1}^m a_{in}$
Marginal					

- (c) Durante o estágio 1 o vetor escolhido entre os r_j e s_j para entrar na
base é aquele que possui maior custo marginal não negativo. E, não mais
exigimos que a soma do custo marginal de cada par r_j e s_j seja zero
e ainda, o custo marginal de todos os $2n$ vetores são imediatamente dispo-
níveis.

Durante o estágio 2 o vetor escolhido para entrar na base entre os vetores não básicos u_i e v_i que possuem maior custo marginal. Não mais exigimos que a soma custo marginal de cada u_i e v_i seja 2.

O algoritmo quase sempre termina durante o estágio 2, quando todos os vetores não básicos possuem custo marginal positivo; obviamente uma solução ótima foi atingida.

Se porventura a tabela final contém vetores básicos r_j ou s_j , associados com valores negativos, devemos multiplicar as respectivas linhas por -1, e permutar o vetor básico r_j (ou s_j) pelo correspondente vetor não básico s_j (ou r_j).

(d) Em ambos estágios 1 e 2 o vetor escolhido para sair da base, dentre os u_i ou v_i , é aquele que causa maior redução na função objetivo.

A regra normal do método simplex para determinar o vetor que deve sair da base, é modificada para o seguinte:

Primeiramente, aplicamos a regra normal para determinar a linha pivotamento entre os vetores u_i e v_i . Localizado o vetor, o pivô é o elemento positivo que está na interseção da linha pivotamento com a coluna pivotamento.

Feito isto, subtraindo-se duas vezes o valor do pivô do custo marginal da coluna pivotamento obtemos um resultado não positivo, então aplicamos a transformação simplex neste pivô.

Por outro lado, se aplicando o processo acima e o resultado obtido for positivo devemos proceder da seguinte maneira: se subtraindo duas vezes a linha o valor do pivô do custo marginal da coluna pivotamento, obtemos um resultado não positivo a transformação simplex é aplicada com este pivô. Caso contrário, subtraímos duas vezes a linha pivotamento da linha custo marginal, multiplica a linha por (-1) , e substitua na base o vetor u_i (ou v_i) correspondente a linha pivotamento dada por $v_i(u_i)$.

Esta operação decresce o valor da função objetivo e troca o sinal do pivô.

A regra normal para determinação da linha pivotamento é novamente aplicada, e o novo pivô é localizado.

Este procedimento é repetido até que um pivô não possa ser rejeitado, e a transformação simplex é então aplicada com este pivô.

Mas, pode suceder que um vetor não factível seja encontrado para sair da base, ou melhor, uma coluna pivotamento que não contém elementos positivos opostos aos vetores básicos u_i 's e v_i 's. No estágio 1 isto ocorre se, o posto da matriz

$$A = \{ a_{ij} \}^T \text{ é menor que } n.$$

Neste caso a coluna pivotamento correspondente aos vetores r_j (ou s_j) pode ser ignorada nos futuros cálculos, e a transformação simplex não é efetuada nesta iteração.

No estágio 2, desde que a existência de uma solução do problema L_1 é garantida, uma linha pivotamento viável é sempre computada para alguma coluna pivotamento dada pela regra (c) acima.

- (e) As regras para transformação da tabela simplex são nas mesmas aplicadas na forma . padrão . do método simplex.
- (f) A iteração calculada no algoritmo é aumentada devido às transformações simplex, ou ainda, pelo fato destas transformações serem repassadas durante o estágio 1, porque um vetor factível pode não ser encontrado para sair da base.

3. EXEMPLO

Dado o sistema:

$$\left\{ \begin{array}{l} x_1 + x_2 = 1 \\ x_1 + 2x_2 = 1 \\ x_1 + 3x_2 = 2 \\ x_1 + 4x_2 = 3 \\ x_1 + 5x_2 = 2 \end{array} \right.$$

passamos o sistema para um problema de programação linear equivalente.

Fazemos $x_j = r_j - s_j$ $j = 1, 2$. obtemos

$$\text{Minimize } \sum_{i=1}^5 (u_i + v_i)$$

$$\text{S.a. } \left\{ \begin{array}{l} 1 = r_1 + r_2 - s_1 - s_2 + u_1 - v_1 \\ 1 = r_1 + 2r_2 - s_1 - 2s_2 + u_2 - v_2 \\ 2 = r_1 + 3r_2 - s_1 - 3s_2 + u_3 - v_3 \\ 3 = r_1 + 4r_2 - s_1 - 4s_2 + u_4 - v_4 \\ 2 = r_1 + 5r_2 - s_1 - 5s_2 + u_5 - v_5 \end{array} \right.$$

(1) CÁLCULO DA SOLUÇÃO:

TABELA 1

CUSTO ↓	→ BASE	R	θ_1	θ_2	s_1	s_2	u_1	u_2	u_3	u_4	u_5	v_1	v_2	v_3	v_4	v_5
1	u_1	1	1	1	-1	-1	1	0	0	0	0	-1	0	0	0	0
1	u_2	1	1	2	-1	-2	0	1	0	0	0	0	-1	0	0	0
1	u_3	2	1	3	-1	-3	0	0	1	0	0	0	0	-1	0	0
1	u_4	3	1	4	-1	-4	0	0	0	1	0	0	0	0	-1	0
1	u_5	2	1	5	-1	-5	0	0	0	0	1	0	0	0	0	-1
CUSTO MARGINAL		9	5	15	-5	-15	0	0	0	0	0	-2	-2	-2	-2	-2

(2) CÁLCULO DO CUSTO MARGINAL

Consideremos:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = I_5 \quad \text{matriz básica}$$

$$T = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix} \quad \text{matriz básica.}$$

$CB = (1 \ 1 \ 1 \ 1 \ 1)$ custos das variáveis básicas

$CT = (0 \ 0)$ custos das variáveis não básicas

$CM = C_B \cdot B^{-1} T = CT$ (custo marginal)

$$CM_T = (1 \ 1 \ 1 \ 1 \ 1) I_5 \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix} = (1 \ 1 \ 1 \ 1 \ 1) \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix} - (0 \ 0)$$

(c) b_2 entra na base pois possui o maior custo marginal não negativo.

(c') Escolha do Pivô:

Aplicando a regra simplex temos os possíveis candidatos como sendo (5) (2) (3).

(d) Consideremos $\text{Pivô} = 5$

$$(5 \ 15) - 2(1 \ 5) = (5 \ 15) - (2 \ 10) = (3 \ 5)$$

temos $1 \geq 0$.

$$\text{Pivô} = 3$$

$$(1 \ 1) - 2(1 \ 3) = (1 \ 1) - (2 \ 6) = (-1 \ -5)$$

temos $-5 \geq 0$.

Aplicaremos o pivotamento em 3.

$$= (5 \quad 15) - (0 \quad 0) = (5 \quad 15) \rightarrow CM_T = (5 \quad 15).$$

Consideremos:

$$B' = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{matriz pseudobásica.}$$

$$CM_{B'} = C_B B'^{-1} \cdot B' = C_{B'} \xrightarrow{-} \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \xrightarrow{-(1 \ 1 \ 1 \ 1 \ 1)} \rightarrow CM_{B'} = (1 \ 1 \ 1 \ 1 \ 1) I$$

$$= (1 \ 1 \ 1 \ 1 \ 1) \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} - (1 \ 1 \ 1 \ 1 \ 1) =$$

$$= (-1 \ -1 \ -1 \ -1 \ -1) - (1 \ 1 \ 1 \ 1 \ 1) = (-2 \ -2 \ -2 \ -2 \ -2)$$

$$CM_{B'} = (-2 \ -2 \ -2 \ -2 \ -2)$$

TABLE 1.1

BASE	R	x_1	x_2	s_1	s_2	u_1	u_2	u_3	u_4	u_5	v_1	v_2	v_3	v_4	v_5
u_1	1	1	1	-1	-1	1	0	0	0	0	-1	0	0	0	0
v_2	-1	-1	-2	1	2	0	-1	0	0	0	0	1	0	0	0
u_3	2	1	(3)	-1	-3	0	0	1	0	0	0	0	-1	0	0
u_4	3	1	4	-1	-4	0	0	0	1	0	0	0	0	-1	0
v_5	-2	-1	-5	1	5	0	0	0	0	-1	0	0	0	0	1
u_1	1/3	2/3	0	-2/3	0	1	0	-1/3	0	0	-1	0	1/3	0	0
v_2	1/3	-1/3	0	1/3	0	0	-1	2/3	0	0	0	1	-2/3	0	0
b_2	2/3	1/3	1	-1/3	-1	0	0	1/3	0	0	0	0	-1/3	0	0
u_4	1/3	-1/3	0	1/3	0	0	0	-4/3	1	0	0	0	4/3	-1	0
v_5	4/3	2/3	0	-7/3	0	0	0	5/3	0	-1	0	0	5/3	0	-1

Temos:

$$T = \begin{bmatrix} 3/2 & -1/2 \\ 1/2 & 1/2 \\ -1/2 & 1/2 \\ 1/2 & -3/2 \\ -1 & 2 \end{bmatrix}$$

$$CM_T = (0 \ 1 \ 0 \ 1 \ 1) \begin{bmatrix} 3/2 & -1/2 \\ 1/2 & 1/2 \\ -1/2 & 1/2 \\ 1/2 & -3/2 \\ -1 & 2 \end{bmatrix} - (1 \ 1) =$$

$$= (0 \ 1) - (1 \ 1) = (-1 \ 0).$$

$$CM_R = (0 \ 1 \ 0 \ 1 \ 1) \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \\ 1 \end{pmatrix} - 0 =$$

$$= 1/2 + 1/2 + 1 = 2.$$

Obtemos $CM_T = (-1 \ 0)$ logo terminamos a aplicação do algoritmo neste passo, pois o custo marginal é não positivo.

Portanto obtemos:

$$b_1 = 1/2 \quad v_2 = 1/2 \quad b_2 = 1/2 \quad u_4 = 1/2 \quad v_5 = 1$$

Temos então $b_1 = 1/2 = x_1$ e $b_2 = 1/2 = x_2$ fazendo $x = (x_1 \ x_2) +$
 $\rightarrow x = (1/2 \ 1/2)$

Obtemos que \tilde{e} solução da 1a. e da 3a. equação do sistema inicial.

(2) Cálculo do Resíduo:

Usando a função resíduo

$$\phi(x) = \sum_{i=1}^5 |\phi_i(x)|$$

$$\phi_i(x) = b_i - \sum_{j=1}^2 a_{ij} x_j$$

$$i = 1$$

$$\phi_1(x) = 1 - 1/2 - 1/2 = 0$$

$$i = 2$$

$$\phi_2(x) = 1 - 1/2 - 1 = -1/2$$

$$i = 3$$

$$\phi_3(x) = 2 - 1/2 - 3/2 = 0$$

$$i = 4$$

$$\phi_4(x) = 3 - 1/2 - 2 = 1/2$$

$$i = 5$$

$$\phi_5(x) = 2 - 1/2 - 5/2 = -1$$

$$\phi(x) = 1/2 + 1/2 + 1 = 2.$$