Introdução à Programação Linear (2001)
Algoritmo Simplex - versão 2.5

Leônidas de Oliveira Brandão
http://www.ime.usp.br/~leo
http://www.matematica.br
30 de outubro de 2001 17:36

Sumário

Índice Remissivo

Capítulo 4

Introdução ao Algoritmo Simplex

Neste capítulo sistematizaremos os resultados anteriores visando a obtenção de um algoritmo mais eficaz na resolução de problemas lineares. Nosso objetivo é desenvolvermos o primeiro algoritmo genérico de programação linear, introduzido por George B. Dantzig em 1951[1]. Este algoritmo tem pior caso exponencial, porém se comportando muito bem na prática.

Fixados uma matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ e $\boldsymbol{c} \in \mathbb{R}^n$, para simplificar a notação adoremos as seguintes convenções:

1. Denotaremos o problema canônico por (PLC): $\begin{array}{ll} \max & \boldsymbol{c}'\boldsymbol{x} \\ & \mathbf{A}\,\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geqq \mathbf{0}. \end{array}$

2. Admitiremos sempre que $m \leqq n$ e fixados m elementos de $\{1, 2, \ldots, n\}$, representaremos estes pela letra B, muitas vezes como uma lista $B = (j_1, j_2, \ldots, j_m)$, noutras como um conjunto $B = \{j_1, j_2, \ldots, j_m\}$. Sempre que estiver implicito a existência do conjunto/lista B, definiremos por $N := \{1, 2, \ldots, n\} \setminus B$ (usando a forma de conjunto).

3. A razão da convenção acima está associada com a matriz $\mathbf{A}_B$ formada pelas colunas indexadas por B, neste caso utilizada como uma lista, pois, por exemplo, a inversa (se existir) de $[\boldsymbol{a}^{j_1}|\boldsymbol{a}^{j_2}|\boldsymbol{a}^{j_3}]$ pode ser diferente da inversa de $[\boldsymbol{a}^{j_2}|\boldsymbol{a}^{j_1}|\boldsymbol{a}^{j_3}]$.

   Já a forma de conjunto é útil quando desejarmos, por exemplo, apenas analisarmos se $\{\boldsymbol{a}^i\}_{i \in B}$ é l.i.

   Analogamente definiremos $\boldsymbol{x}_B$ como o vetor do $\mathbb{R}^m$, cuja componente i é x_{j_i}. Da mesma forma, se $N := (l_1, l_2, \ldots, l_{n-m})$, então a i-ésima componente de $\boldsymbol{x}_N$ é x_{l_i}.

4. Fixada a lista B, $B := (j_1, j_2, \ldots, j_m)$, diremos que:
   - ∘ B é **base** $\iff$ $\{\boldsymbol{a}^j\}_{j \in B}$ é l.i.
   - ∘ B é **base viável** $\iff$ B é base e $\mathbf{A}_B^{-1}\boldsymbol{b} \geqq \mathbf{0}$.

5. Se B é uma base viável, diremos que cada componente de $\boldsymbol{x}_B$ é uma **variável básica** e cada linha de $\boldsymbol{x}_N$ é uma **variável não básica** ou **variável fora da base**.

   Além disso, da caracterização de vértice por colunas l.i. é fácil ver que, se B é base viável e $\overline{\boldsymbol{x}}$ o vértice associado, então

$$\overline{\boldsymbol{x}}_B := \mathbf{A}_B^{-1}\boldsymbol{b} \geqq \mathbf{0} \tag{4.1}$$

$$\overline{\boldsymbol{x}}_N := \mathbf{0}, \tag{4.2}$$

[1] *Maximization of a Linear Function of Variables Subjedct to Linear Inequalities*, em Activity Analysis of Production and Allocation, pp. 339-347, John Wiley & Sons, 1951.

e o vértice $\overline{\boldsymbol{x}}$ definido pela equação acima será dito **vértice gerado por** B.

Gostaríamos de resolver o (PLC) e já sabemos que vértices desempenham um papel chave nesta resolução. Além disso:

(1) Para um dado $\boldsymbol{x} \in X := \{\boldsymbol{x} \in \mathbb{R}^n / \ \boldsymbol{A x} = \boldsymbol{b},\ \boldsymbol{x} \geqq \mathbf{0}\}$, podemos reescrever a identidade $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ separando as colunas de $\mathbf{A}$ segundo $I(\boldsymbol{x})$,

$$\boldsymbol{b} = \mathbf{A}\boldsymbol{x} = \sum_{i \in I(\boldsymbol{x})} x_i \mathbf{A}^i + \sum_{i \notin I(\boldsymbol{x})} x_i \mathbf{A}^i.$$

Do capítulo anterior, sabemos que,

$$\text{se } \boldsymbol{x} \text{ é vértice de } X, \ \text{ então } \{\mathbf{A}^i\}_{i \in I(\boldsymbol{x})} \text{ é l. i.}$$

(2) Por outro lado, também sabemos gerar vértices a partir de bases do $\mathbb{R}^n$ formadas por colunas de $\mathbf{A}$. Dado um conjunto B, para o qual $\{\mathbf{A}^i\}_{i \in B}$ é base, podemos verificar se existe um $\boldsymbol{x} \in V(\mathrm{X})$ associado a esta base.

Para isso reordenaremos as colunas de $\mathbf{A}$ e os correspondentes multiplicadores x_i, colocando nas primeiras posições as colunas de B. Assim, seja $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_N)$ definido por $\boldsymbol{x}_B = \mathbf{A}_B^{-1}\boldsymbol{b}$ e $\boldsymbol{x}_N = \mathbf{0}$. Deste modo teremos

Lema 4.1 *Se* $\{\boldsymbol{A}_i\}_{i \in B}$ *é uma base, então qualquer que seja* $\boldsymbol{x} \in \mathbb{R}^n$*, vale a equivalência*

$$\boldsymbol{b} = \mathbf{A}\boldsymbol{x} = \mathbf{A}_B\boldsymbol{x}_B + \mathbf{A}_N\boldsymbol{x}_N \iff \boldsymbol{x}_B = \mathbf{A}_B^{-1}(\boldsymbol{b} - \mathbf{A}_N\boldsymbol{x}_N) = \mathbf{A}_B^{-1}\boldsymbol{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\boldsymbol{x}_N. \tag{4.3}$$

Lembre que, se colocarmos $\boldsymbol{x}_N := \mathbf{0}$ e obtivermos $\boldsymbol{x}_B := \mathbf{A}_B^{-1}\boldsymbol{b} \geqq \mathbf{0}$, então teremos $\boldsymbol{x} \in V(X)$ (a base será viável).

A partir da caracterização de vértice por colunas l.i. e das definições (4.1) e (4.2), podemos deduzir um primeiro algoritmo (combinatório) para resolver o (PLC).

Algoritmo 4.1 Algoritmo combinatório para seleção do melhor vértice do (PLC)

// Para decidir o melhor vértice e se $X \neq \emptyset$ *rode com* $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geqq \mathbf{0}$
// Para decidir se problema ilimitado rode com C^1
Seja P_B *uma fila definida como*
$P_B := \{\ B,$ *subconjunto das combinações de* $\{1, 2, \ldots, n\}$*, tal que* $\#B = m\ \}$
$c_{ot} \leftarrow -\infty$
enquanto $P_B \neq \emptyset$
  $B \leftarrow$ *primeiro* (P_B)
  $P_B \leftarrow P_B \backslash B$
  Seja $\overline{\mathbf{A}}$ *tal que* $\overline{\mathbf{A}}^i = \mathbf{A}^{j_i}$, $j_i \in B$ $(i = 1, 2, \ldots, m)$.
  se $\det(\overline{\mathbf{A}}) \neq 0$ *então* *//* $\overline{\mathbf{A}}$ *forma base de* $\mathbb{R}^m$
    $\boldsymbol{x}_B \leftarrow \overline{\mathbf{A}}^{-1}\boldsymbol{b}$
    $\boldsymbol{x}_N \leftarrow \mathbf{0}$
    se $\boldsymbol{x}_B \geqq \mathbf{0}$ *então* *//* $\boldsymbol{x}$ *é vértice de* X
      se $\boldsymbol{c}_B'\boldsymbol{x}_B > c_{ot}$ *então*
        $c_{ot} \leftarrow \boldsymbol{c}_B'\boldsymbol{x}_B$
        $\boldsymbol{x}_{ot_B} \leftarrow \boldsymbol{x}_B$
        $\boldsymbol{x}_{ot_N} \leftarrow \mathbf{0}$
// fim enquanto

Entretanto os resultados expostos nos itens (1) e (2), acima, podem ser aproveitados para derivar um algoritmo mais eficiente que esse.

Usando a equação (4.3) na função objetivo $\boldsymbol{c}'\boldsymbol{x}$, conseguiremos uma identidade envolvendo o valor objetivo de $\boldsymbol{x}$ e de $\overline{\boldsymbol{x}}$ (associado à B),

$$\boldsymbol{c}'\boldsymbol{x} = \boldsymbol{c}_B'\boldsymbol{x}_B + \boldsymbol{c}_N'\boldsymbol{x}_N \overset{(4.3)}{=} \boldsymbol{c}_B'(\mathbf{A}_B^{-1}\boldsymbol{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\boldsymbol{x}_N) + \boldsymbol{c}_N'\boldsymbol{x}_N = \boldsymbol{c}_B'\mathbf{A}_B^{-1}\boldsymbol{b} + (\boldsymbol{c}_N' - \boldsymbol{c}_B'\mathbf{A}_B^{-1}\mathbf{A}_N)\boldsymbol{x}_N. \tag{4.4}$$

Se definirmos $\boldsymbol{\gamma}_N := \boldsymbol{c}_N - \mathbf{A}_N(\mathbf{A}_B^{-1})'\boldsymbol{c}_B$ e usarmos as equações (4.1) e (4.2), da última identidade vem que

$$\boldsymbol{c}'\boldsymbol{x} \overset{(4.4)}{=} \boldsymbol{c}_B'\mathbf{A}_B^{-1}\boldsymbol{b} + \boldsymbol{\gamma}_N'\boldsymbol{x}_N \overset{(4.1)}{=} \boldsymbol{c}_B'\overline{\boldsymbol{x}}_B + \boldsymbol{\gamma}_N'\boldsymbol{x}_N \overset{(4.2)}{=} \boldsymbol{c}'\overline{\boldsymbol{x}} + \boldsymbol{\gamma}_N'\boldsymbol{x}_N.$$

No restante deste texto, sempre que usarmos $\boldsymbol{\gamma}_N$ deverá estar implicito a existência de uma base viável B. Note que podemos calcular $\gamma_i := c_i - \boldsymbol{c}_B'\mathbf{A}_B^{-1}\boldsymbol{a}^i = c_i - (\boldsymbol{a}^i)'(\mathbf{A}_B^{-1})'\boldsymbol{c}_B'$ também para $i \in B$, e isso resulta sempre em: $\boldsymbol{\gamma}_B = \mathbf{0}$.

Deste modo, fixada uma base viável B, tomando $\boldsymbol{\gamma}_N'$ definido por $\gamma_i := c_i - \boldsymbol{c}_B'\mathbf{A}_B^{-1}\boldsymbol{a}^i = c_i - (\boldsymbol{a}^i)'(\mathbf{A}_B^{-1})'\boldsymbol{c}_B'$ **(para** $i \in N$**)**, então qualquer que seja $\boldsymbol{x} \in X$, podemos calcular a função objetivo em $\boldsymbol{x}$ pela equação abaixo

$$\boldsymbol{c}'\boldsymbol{x} = \boldsymbol{c}'\overline{\boldsymbol{x}} + \boldsymbol{\gamma}_N'\boldsymbol{x}_N. \tag{4.5}$$

Usando esta equação podemos decidir se $\boldsymbol{x}$ é "melhor" ou "pior" que $\overline{\boldsymbol{x}}$:

- $\boldsymbol{x}$ é melhor: $\boldsymbol{c}'\boldsymbol{x} > \boldsymbol{c}'\overline{\boldsymbol{x}}$, se $\boldsymbol{\gamma}_N'\ \boldsymbol{x}_N > \mathbf{0}$;
- $\overline{\boldsymbol{x}}$ é melhor: $\boldsymbol{c}'\boldsymbol{x} \leqq \boldsymbol{c}'\overline{\boldsymbol{x}}$, se $\boldsymbol{\gamma}_N'\ \boldsymbol{x}_N \leqq \mathbf{0}$;

Mais ainda, a relação (4.5) nos fornece uma condição suficiente para otimalidade de um vértice qualquer.

Se fixarmos uma base B e denotarmos por $\overline{\boldsymbol{x}}$ o vértice associado, devemos notar que $\boldsymbol{c}'\overline{\boldsymbol{x}} = \boldsymbol{c}_B'\overline{\boldsymbol{x}}_B = \boldsymbol{c}_B'\mathbf{A}_B^{-1}\boldsymbol{b}$ e $\boldsymbol{\gamma}_N$ são constantes. Portanto, acabamos de produzir um **critério de parada** para qualquer algoritmo baseado em vértices: para um dado vértice $\boldsymbol{v} \in V(X)$, com $B := I_{\boldsymbol{v}} = \{i : v_i \neq 0\}$

$$\boldsymbol{\gamma} \leqq \mathbf{0} \implies \boldsymbol{v} \text{ é ótimo.}$$

Lema 4.2 *Se* $\overline{\boldsymbol{x}}$ *é o vértice gerado por* B *e* $\boldsymbol{\gamma}_N \leqq \mathbf{0}$, *então* $\overline{\boldsymbol{x}}$ *é uma solução ótima do (PLC).*

Demonstração Trivial a partir dos resultados anteriores. ■

Deste modo, sobra a questão: o que fazer quando $\boldsymbol{\gamma}_N \nleqq \mathbf{0}$?
Iremos analisar separadamente $\boldsymbol{\gamma}_N \nleqq \mathbf{0}$ em dois casos:

1. $\gamma_k > 0$ e $\mathbf{A}_B^{-1}\boldsymbol{a}^k \leqq \mathbf{0}$; e
2. $\gamma_k > 0$ e $\mathbf{A}_B^{-1}\boldsymbol{a}^k \nleqq \mathbf{0}$.

O primeiro caso produz um nova condição de parada, desta vez detectando ilimitação, enquanto o segundo terá desdobramentos a serem explorados nas próxima subseção.

Para algum $k \in N$, definiremos $\boldsymbol{\pi} := \mathbf{A}_B^{-1}\boldsymbol{a}^k$.

Lema 4.3 *Se* $B = (j_1, j_2, \ldots, j_m)$ *é uma base viável, para a qual* $\langle \gamma_k > 0$ *e* $\boldsymbol{\pi} = \mathbf{A}_B^{-1}\boldsymbol{a}^k \leqq \mathbf{0}\rangle$, *então o (PLC) é ilimitado e* $\boldsymbol{h}$, *definido por*[2] $h_i := \begin{cases} -\pi_{j_l}, & i = j_l \in B \\ 1, & i = k \\ 0, & i \notin B \cup \{k\} \ (\equiv i \in N \setminus \{k\}) \end{cases}$, *é uma direção de ilimitação.*

Demonstração Seja $\boldsymbol{v}$ o vértice gerado por $B = (j_1, j_2, \ldots, j_m)$ e notemos que $\boldsymbol{h} = \sum_{i \in B} h_i \boldsymbol{e}^i + \sum_{i \in N} h_i \boldsymbol{e}^i = -\sum_{i=1}^{m} \pi_i \boldsymbol{e}^{j_i} + \boldsymbol{e}^k$. Mostraremos que o vetor $\boldsymbol{h}$, acima definido é uma direção de ilimitação, provando que: $\boldsymbol{x}_\lambda := \boldsymbol{v} + \lambda\boldsymbol{h} \in X$ $(\forall \lambda \geqq 0)$ e que $\lim_{\lambda \longrightarrow \infty} \boldsymbol{c}'\boldsymbol{x}_\lambda = +\infty$. De fato:

$$\boldsymbol{x}_\lambda \geqq \mathbf{0} \text{ e } \mathbf{A}\boldsymbol{x}_\lambda = \mathbf{A}\boldsymbol{v} + \lambda\mathbf{A}\boldsymbol{h} = \boldsymbol{b} + \lambda(\mathbf{A}_B\boldsymbol{h}_B + \mathbf{A}_N\boldsymbol{h}_N) = \boldsymbol{b} + \lambda(-\mathbf{A}_B\boldsymbol{\pi} + \mathbf{A}_N\boldsymbol{e}^k) = \boldsymbol{b} + \lambda(-\mathbf{A}_B^{-1}\mathbf{A}_B\boldsymbol{a}^k + \boldsymbol{a}^k) = \boldsymbol{b}.$$

$$\boldsymbol{c}'\boldsymbol{x}_\lambda = \boldsymbol{c}'\boldsymbol{v} + \lambda\boldsymbol{c}'\boldsymbol{h} = \boldsymbol{c}'\boldsymbol{v} + \lambda(\boldsymbol{c}'_B\mathbf{A}_B^{-1}\boldsymbol{a}^k + \boldsymbol{c}'_N\boldsymbol{h}_N) = \boldsymbol{c}'\boldsymbol{v} + \lambda(\boldsymbol{c}'_B\mathbf{A}_B^{-1}\boldsymbol{a}^k + c_k) = \boldsymbol{c}'\boldsymbol{v} + \lambda\gamma_k \underset{\lambda \longrightarrow \infty}{\longrightarrow} +\infty \qquad \blacksquare$$

Quanto ao segundo caso, $\gamma_k > 0$ e $\boldsymbol{\pi} = \mathbf{A}_B^{-1}\boldsymbol{a}^k \nleqq \mathbf{0}$, podem ocorrer dois casos: existir um vértice "melhor" que $\boldsymbol{v}$ ou existir outra base para o mesmo vértice (neste caso é um vértice degenerado). A ocorrência deste último caso não implica necessariamente $\boldsymbol{v}$ não ser ótimo, conforme pode ser observado no exercício abaixo.

Exercício 4.1 *Sejam* $\mathbf{A} := \begin{pmatrix} 1 & 1 & -1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$ *e* $\boldsymbol{b} := \begin{pmatrix} 2 \\ 2 \end{pmatrix}$.

1. *prove (pela definição!) que* $\overline{\boldsymbol{x}} := (0, 2, 0, 0)$ *é vértice de* X *(logo* $B = (2,3)$ *ou* $B = (2,4)$*);*
2. *construa* $\boldsymbol{c}$, *de modo que:* $\overline{\boldsymbol{x}}$ *é vértice ótimo com* $\boldsymbol{\gamma}_N \leqq \mathbf{0}$;
3. *construa* $\boldsymbol{c}$, *de modo que:* $\overline{\boldsymbol{x}}$ *é vértice ótimo com* $\boldsymbol{\gamma}_N \nleqq \mathbf{0}$;
4. *(PLC) ilimitado, com* $\boldsymbol{\gamma}_N \geqq \mathbf{0}$.

Tente justificar os itens 2 a 4 (quem é $\boldsymbol{\gamma}_N$ *e por que a solução é ótima ou o problema é ilimitado); note que* $\boldsymbol{\gamma}_N \nleqq \mathbf{0}$ *ou* $\boldsymbol{\gamma}_N \ngeqq \mathbf{0}$ *implica que* $\boldsymbol{\gamma}_N \neq \mathbf{0}$.

No caso 2, $\langle\ k \in N,\ \gamma_k > 0$ e $\mathbf{A}_B^{-1}\boldsymbol{a}^k \nleqq \mathbf{0}\ \rangle$, a equação (4.5) dá uma pista sobre como melhorar a função objetivo: se encontrarmos um novo ponto viável para o qual $\overline{x}_k > 0$ e $\overline{x}_i = 0$, para $i \in N \setminus \{k\}$, então o valor objetivo deste será maior que o do vértice gerado por B. Isso será examinado na próxima subseção.

4.1 Troca de bases viáveis

Nesta seção analisaremos o caso 2 apontado no quadro anterior. Sendo $\boldsymbol{x}$ o vértice gerado pela base viável $B = (j_1, j_2, \ldots, j_m)$, se

- $0 < \gamma_k = c_k - \boldsymbol{c}'_B\mathbf{A}_B^{-1}\boldsymbol{a}^k$, $k \neq j_i$ $(i \in \{1, 2, \ldots, m\})$, e
- $\boldsymbol{\pi} := \mathbf{A}_B^{-1}\boldsymbol{a}^k \nleqq \mathbf{0}$,

gostaríamos de obter um novo ponto viável $\overline{\boldsymbol{x}}$, preferivelmente vértice, de tal forma que $\overline{x}_k > 0$ e $\overline{x}_i = 0$, para $i \in N \setminus \{k\}$. Com um tal procedimento poderemos melhorar "substancialmente" o algoritmo 4.1, nunca enumerando um vértice "pior" (menor valor objetivo) que o anterior.

Lema 4.4 *Sejam* $\overline{\boldsymbol{x}} \in X$, $B = (j_1, j_2, \ldots, j_m)$ *uma base viável e* $\boldsymbol{x}$ *o vértice gerado por* B.

Se $\langle \gamma_k > 0$ *e* $\overline{\boldsymbol{x}}_N = \overline{x}_k\boldsymbol{e}^k \geqq \mathbf{0}\rangle$[3] *então* $\boldsymbol{c}'\overline{\boldsymbol{x}} \geqq \boldsymbol{c}'\boldsymbol{x}$.

[2]Observe que $\boldsymbol{a}^{j_i}$ é a i-ésima coluna da lista B.

[3]Note que a dimensão de $\overline{\boldsymbol{x}}_N = \overline{x}_k\boldsymbol{e}^k$ é $n - m$ e deste modo $\boldsymbol{e}^k \in \mathbb{R}^{n-m}$. Isso implica que $\mathbf{A}_N\overline{\boldsymbol{x}}_N = \sum_{i \in N} \overline{x}_i\boldsymbol{a}^i = \overline{x}_k\boldsymbol{a}^k$.

Demonstração Como B gera $\boldsymbol{x} \in V(X)$, segue que $\boldsymbol{x}_B = \mathbf{A}_B^{-1}\boldsymbol{b} \geqq \mathbf{0}$, $\boldsymbol{x}_N = \mathbf{0}$ e assim

$$\boldsymbol{c}'\boldsymbol{x} = \boldsymbol{c}'_B\boldsymbol{x}_B + \boldsymbol{c}'_N\boldsymbol{x}_N = \boldsymbol{c}'_B\mathbf{A}_B^{-1}\boldsymbol{b}. \tag{$\star$}$$

Além disso, da definição de $\overline{\boldsymbol{x}}$,

$$\begin{aligned} &\overline{x}_k \geqq 0, \\ &\overline{x}_j = 0, \quad j \in N \setminus \{k\}, \end{aligned} \tag{$\star\star$}$$

portanto, usando a equação (4.4) para separar $\overline{\boldsymbol{x}}$ segundo os índices em B, segue que

$$\boldsymbol{c}'\overline{\boldsymbol{x}} = \boldsymbol{c}'_B\mathbf{A}_B^{-1}\boldsymbol{b} + \boldsymbol{\gamma}'_N\overline{\boldsymbol{x}}_N \overset{(\star)}{=} \boldsymbol{c}'\boldsymbol{x} + \boldsymbol{\gamma}'_N\overline{\boldsymbol{x}}_N \overset{(\star\star)}{=} \boldsymbol{c}'\boldsymbol{x} + \gamma_k\overline{x}_k \geqq \boldsymbol{c}'\boldsymbol{x},$$

concluindo a demonstração. ∎

Devemos notar que um mesmo vértice degenerado[4] pode ter mais de uma base garadora, e como a condição suficiente de otimalidade depende apenas da base ($\boldsymbol{\gamma}_N = \boldsymbol{c}_N - \mathbf{A}'_N(\mathbf{A}_B^{-1})'\boldsymbol{c}_B \leqq \mathbf{0}$). Ou seja, é possível existir um vértice ótimo degerado que não satisfaça esta condição sob determinada base enquanto sob outra base a condição se verifique. O resultado abaixo mostra em que circunstância ocorre esta múltipla representação: quando o vértice for degenerado (componentes nulas na base).

Lema 4.5 *Se existe um* $I \subseteq \{1, 2, \ldots, n\}$, *para o qual,* $\boldsymbol{b} \in C(\,[\{\boldsymbol{a}^i\}_{i \in I}]\,)$, *então para toda base* B *contendo* I, *tem-se que* $\langle x_i = 0, \forall i \notin I,\ \ \boldsymbol{x}$ *gerado por* $B\rangle$

Demonstração Como $\boldsymbol{b} \in C(\,[\{\boldsymbol{a}^i\}_{i \in I}]\,)$, existe $\overline{\boldsymbol{\alpha}} \in \mathbb{R}^n$ para o qual

$$\boldsymbol{b} = \sum_{i=1}^{n} \overline{\alpha}_i\boldsymbol{a}^i = \sum_{i \in I} \overline{\alpha}_i\boldsymbol{a}^i + \sum_{i \notin I} \overline{\alpha}_i\boldsymbol{a}^i = \sum_{i \in I} \overline{\alpha}_i\boldsymbol{a}^i, \tag{$\star$}$$

notando que, para $i \notin I$, $\overline{\alpha}_i$ não influe na soma, ou seja, $\overline{\alpha}_i = 0, \forall i \notin I$.

Tome agora uma base qualquer B (não necessariamente viável), contendo I (vem daqui a necessidade da hipótese $\{\boldsymbol{a}^i\}_{i \in I}$ ser l.i.). Então qualquer que seja $\boldsymbol{\alpha} \in \mathbb{R}^n$, para o qual $\boldsymbol{b} = \mathbf{A}\boldsymbol{\alpha}$, teremos

$$\boldsymbol{b} = \sum_{i=1}^{n} \alpha_i\boldsymbol{a}^i = \sum_{i \in I} \alpha_i\boldsymbol{a}^i + \sum_{i \in B \setminus I} \alpha_i\boldsymbol{a}^i + \sum_{i \notin B} \alpha_i\boldsymbol{a}^i \overset{\alpha_i = 0, i \notin B}{=} \sum_{i \in I} \alpha_i\boldsymbol{a}^i + \sum_{i \in B \setminus I} \alpha_i\boldsymbol{a}^i = \mathbf{A}_B\boldsymbol{\alpha}_B.$$

Como $\mathbf{A}_B$ é não singular, o sistema $\mathbf{A}_B\boldsymbol{\alpha}_B = \boldsymbol{b}$ tem solução única $\boldsymbol{\alpha}_B = \mathbf{A}_B^{-1}\boldsymbol{b}$. Entretanto $(\star)$ é uma solução deste sistema, daí segue a tese: $\alpha_i = \overline{\alpha}_i$, $\forall i \in \{1, 2, \ldots, n\}$ e em particular, $\alpha_i = \overline{\alpha}_i \overset{(\star)}{=} 0$, $\forall i \notin i$. ∎

Devemos notar que no algoritmo 4.1, o invariante mais importante do laço (*enquanto*) é: B ser base viável. Deste modo, para conseguirmos aprimorar o referido algoritmo de modo a nunca gerar um vértice "pior" (usando a direção $\boldsymbol{h} := \sum_{i \in B} \pi_{j_i}\boldsymbol{e}^i + \boldsymbol{e}^k$ obtida no lema 4.3), devemos responder às seguintes questões:

1. como gerar $\overline{\boldsymbol{x}}$ que seja viável (com as propriedades desejadas: viável e com $\overline{\boldsymbol{x}}_N = \overline{x}_k\boldsymbol{e}^k \geqq \mathbf{0}$) ?
2. como garantir que $\overline{\boldsymbol{x}}$ seja vértice ?
3. como gerar a primeira base viável ?

Nas duas próximas subseções trabalharemos com as seguintes premissas:

[4] $x_i = 0$, para algum $i \in B$, B base geradora de $\boldsymbol{x}$.

Fixado uma base viável B e denotando por $\boldsymbol{x}$ o vértice gerado por B, lembrando que $\boldsymbol{\gamma}_N := \boldsymbol{c}_N - \mathbf{A}_N(\mathbf{A}_B^{-1})'\boldsymbol{c}_B$, suporemos a existência de um índice $k \in N$ tal que

$$\langle 0 < \gamma_k = c_k - \boldsymbol{c}'_B\mathbf{A}_B^{-1}\boldsymbol{A}_k\rangle \text{ e } \langle \boldsymbol{\pi} := \mathbf{A}_B^{-1}\boldsymbol{a}^k \nleqq \mathbf{0}\rangle. \tag{4.6}$$

4.1.1 Como garantir viabilidade

Para garantir a viabilidade do novo vetor, $\overline{\boldsymbol{x}}$, precisamos assegurar que: $\overline{\boldsymbol{x}} \geqq \mathbf{0}$ e $\mathbf{A}\overline{\boldsymbol{x}} = \boldsymbol{b}$. Entretanto, considerando o lema 4.4, desejamos que $\overline{x}_k \geqq 0$ e $\mathbf{A}_N\overline{\boldsymbol{x}}_N = \overline{x}_k\boldsymbol{a}^k$.

Lema 4.6 *Sejam B uma base qualquer*[5]*, $\lambda \geqq \mathbf{0}$ e $k \notin B$.*

Se $\langle\overline{\boldsymbol{x}}_B := \overline{\boldsymbol{b}} - \lambda\boldsymbol{\pi} := \mathbf{A}_B^{-1}\boldsymbol{b} - \lambda\mathbf{A}_B^{-1}\boldsymbol{a}^k \geqq \mathbf{0}$ e $\overline{\boldsymbol{x}}_N = \lambda\boldsymbol{e}^k \geqq \mathbf{0}\rangle$, então $\overline{\boldsymbol{x}} \in X := \{\boldsymbol{x} \in \mathbb{R}^n / \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \ \boldsymbol{x} \geqq \mathbf{0}\}$.

Demonstração Por construção $\overline{\boldsymbol{x}} \geqq \mathbf{0}$, resta examinarmos se $\mathbf{A}\overline{\boldsymbol{x}} = \boldsymbol{b}$:

$$\mathbf{A}\overline{\boldsymbol{x}} = \mathbf{A}_B\overline{\boldsymbol{x}}_B + \mathbf{A}_N\overline{\boldsymbol{x}}_N = \mathbf{A}_B(\mathbf{A}_B^{-1}\boldsymbol{b} - \lambda\mathbf{A}_B^{-1}\boldsymbol{a}^k) + \lambda\mathbf{A}_N\boldsymbol{e}^k = \mathbf{A}_B\mathbf{A}_B^{-1}\boldsymbol{b} - \lambda\mathbf{A}_B\mathbf{A}_B^{-1}\boldsymbol{a}^k + \lambda\boldsymbol{a}^k = \boldsymbol{b}.$$

Logo, $\overline{\boldsymbol{x}} \in X$. ∎

Em função deste lema, para conseguirmos $\overline{\boldsymbol{x}}$ viável basta escolhermos um $\lambda \geqq 0$, tal que, $\lambda\boldsymbol{\pi} \leqq \overline{\boldsymbol{b}}$.

Considerando uma base viável B e sendo $\boldsymbol{x}$ o vértice gerado por esta base, teremos $\boldsymbol{x}_B = \overline{\boldsymbol{b}} := \mathbf{A}_B^{-1}\boldsymbol{b} \geqq \mathbf{0}$. Deste modo, para escolher um tal $\lambda \geqq 0$, devemos nos preocupar apenas com os índices dentre $\{i \in B : \pi_i > 0\}$, uma vez que, para todo $i \in B$ tal que $\pi_i \leqq 0$, segue que $\overline{x}_i = \overline{b}_i - \lambda\pi_i \geqq \overline{b}_i \geqq 0, \ \forall\lambda \geqq 0$.

Lema 4.7 *Sejam B uma base viável qualquer, $k \notin B$ tal que $\boldsymbol{\pi} := \mathbf{A}_B^{-1}\boldsymbol{a}^k \nleqq \mathbf{0}$, $s \in \arg\min_{i\in B:\pi_i>0}\left\{\frac{\overline{b}_i}{\pi_i}\right\}$ e $\lambda := \frac{\overline{b}_s}{\pi_s}$ $(\geqq \mathbf{0})$.*

Se $\langle$ $\overline{\boldsymbol{x}}$ é definido por: $\overline{\boldsymbol{x}}_B := \overline{\boldsymbol{b}} - \lambda\boldsymbol{\pi}$ e $\overline{\boldsymbol{x}}_N := \lambda\boldsymbol{e}^k$ $\rangle$ então $\overline{\boldsymbol{x}} \in X$.

Demonstração Inicialmente note que existe $s \in B$ com a propriedade desejada, pois estamos supondo $\boldsymbol{\pi} \nleqq \mathbf{0}$. Usando os mesmos argumentos do lema 4.6, podemos mostrar que $\mathbf{A}\overline{\boldsymbol{x}} = \boldsymbol{b}$. Assim, resta mostrarmos que $\overline{\boldsymbol{x}} \geqq \mathbf{0}$. Analisaremos separadamente os conjuntos: $B_p := \{i \in B : \pi_i \leqq 0\}$ e $B_n := \{i \in B : \pi_i > 0\}$.

Como $\lambda \geqq 0$, para os índices $i \in B_p$, teremos $\overline{x}_i = \overline{b}_i - \lambda\pi_i \geqq \overline{b}_i \geqq 0$, e para $i \in B_n$, lembrando que $\frac{\overline{b}_s}{\pi_s} \leqq \frac{\overline{b}_j}{\pi_j}$ $(\overline{b}_j \geqq 0, \forall j \in B_p \Rightarrow \pi_j > 0)$, segue que

$$\overline{x}_i = \overline{b}_i - \lambda\pi_i = \pi_i(\frac{\overline{b}_i}{\pi_i} - \lambda) = \pi_i(\frac{\overline{b}_i}{\pi_i} - \frac{\overline{b}_s}{\pi_s}) \geqq 0, \quad \text{para todo } j \in B_p \ (\pi_j > 0).$$

Portanto, $\overline{\boldsymbol{x}} \in X$. ∎

Assim, quando tivermos uma base viável B para a qual $\gamma_k > 0$ e $\boldsymbol{\pi} := \mathbf{A}_B^{-1}\boldsymbol{a}^k \nleqq \mathbf{0}$, se tomarmos $s \in B$ como no lema acima, obteremos um ponto viável com valor objetivo nunca menor que o vértice gerado pela base B.

4.1.2 Como garantir base viável

Com a teoria até aqui examinada, sabemos gerar um novo ponto viável a partir de uma base B (que nem precisa ser viável). Entretanto, estamos interessado em construir um algoritmo a partir destes resultados,

[5]Note que B não precisa gerar vértice, por tanto, pode-se considerar um base não viável.

assim, é necessário que $\overline{\boldsymbol{x}}$ seja não apenas viável, mas também um vértice. Ou seja, associando o vértice $\boldsymbol{x}$ com $\boldsymbol{x}^i$ e $\overline{\boldsymbol{x}}$ com $\boldsymbol{x}^{i+1}$, é necessário que:

B_i base viável	$\boldsymbol{x}^i_{B_i} = \mathbf{A}^{-1}_{B_i}\boldsymbol{b} - \mathbf{A}^{-1}_{B_i}\mathbf{A}_{N_i}\boldsymbol{x}^i_{N_i} = \mathbf{A}^{-1}_{B_i}\boldsymbol{b} \geqq \mathbf{0}$	$\boldsymbol{x}^i_N = \mathbf{0}$
B_{i+1} base viável	$\boldsymbol{x}^{i+1}_{B_{i+1}} = \mathbf{A}^{-1}_{B_{i+1}}\boldsymbol{b} - \mathbf{A}^{-1}_{B_{i+1}}\mathbf{A}_{N_{i+1}}\boldsymbol{x}^{i+1}_{N_{i+1}} \geqq \mathbf{0}$	$\boldsymbol{x}^{i+1}_{N_{i+1}} = \mathbf{0}$

Portanto, se numa interação i tivermos uma base $B = (j_1, j_2, \ldots, j_m)$ para a qual $\langle 0 < \gamma_k = c_k - \boldsymbol{c}'_B\mathbf{A}^{-1}_B\boldsymbol{A}_k\rangle$ e $\langle \boldsymbol{\pi} := \mathbf{A}^{-1}_B\boldsymbol{a}^k \not\leqq \mathbf{0}\rangle$, para aproveitarmos o ponto viável obtido pela atualização derivada no lema 4.6, precisaríamos que o $\overline{\boldsymbol{x}}$ obtido fosse vértice (e consequentemente fosse gerado por uma base $\overline{B}$).

Observando o lema 4.6, notamos que a componente $s \in B$ de $\boldsymbol{x}$ foi "zerada" em $\overline{\boldsymbol{x}}$ e a componente x_k deixou de ser zero em $\overline{x}_k$ (na verdade $\overline{x}_k > 0 \Leftrightarrow \lambda := \frac{\overline{b}_s}{\pi_s} > 0$). Deste modo, é natural pensar que $\boldsymbol{a}^{j_s}$ deixa a base, sendo substituida pela coluna $\boldsymbol{a}^k$, e assim $\overline{B}$ pode ser definida por

$$\overline{B} := (j'_1, j'_2, \ldots, j'_m), \quad \text{sendo } j'_i := \begin{cases} j_i, & i \neq s \\ k, & i = s \end{cases} \tag{4.7}$$

faça o papel de B_{i+1} na iteração $i+1$.

Daí surge nova questão: precisamos impor condição adicionais sobre $B_p = \{i \in B : \pi_i > 0\}$ para selecionar um $s \in B_p$ de tal forma que $\overline{B}$ acima definida seja base viável ?

Felizmente, como veremos no lema abaixo, não é necessário qualquer hipótese adicional para garantir que $\overline{B}$, definido pela equação (4.7), seja base viável. Na verdade, o resultado que assegura esta feliz conclusão para nosso futuro algoritmo é um resultado típico de Álgebra Linear, notando que no caso em estudo: $\boldsymbol{\pi} = \mathbf{A}^{-1}_B\boldsymbol{a}^k \Leftrightarrow \mathbf{A}_B\boldsymbol{\pi} = \boldsymbol{a}^k$, com $0 \neq \pi_s > 0$.

Lema 4.8 *Sejam $B = (j_1, j_2, \ldots, j_m)$ uma base (i.e., $\{\boldsymbol{a}^{j_i}\}_{j_i \in B}$ é l.i.) e $k \notin B$.*

Se $\boldsymbol{a}^k = \sum_{j_i \in B} \alpha_i \boldsymbol{a}^{j_i} = \mathbf{A}_B\boldsymbol{\alpha}$, com $\alpha_s \neq 0$ $(s \in B)$, então $\overline{B}$ definido pela equação (4.7) é l.i.

Demonstração Seja $\boldsymbol{\beta} \in \mathbb{R}^m$, tal que,

$$\begin{aligned} \mathbf{0} &= \mathbf{A}_{\overline{B}}\boldsymbol{\beta} = \sum_{j'_i \in \overline{B}} \beta_i \boldsymbol{a}^{j'_i} = \beta_s \boldsymbol{a}^k + \sum_{j_i \in B \wedge i \neq s} \beta_i \boldsymbol{a}^{j_i} = \beta_s \sum_{j_i \in B} \alpha_i \boldsymbol{a}^{j_i} + \sum_{j_i \in B \wedge i \neq s} \beta_i \boldsymbol{a}^{j_i} \\ &= \beta_s \alpha_s \boldsymbol{a}^{j_s} + \sum_{j_i \in B \wedge i \neq s} (\beta_i + \beta_s \alpha_i)\boldsymbol{a}^{j_i} = \sum_{j_i \in B} \overline{\beta}_i \boldsymbol{a}^{j_i}, \end{aligned}$$

sendo $\overline{\boldsymbol{\beta}}$ definido por:

$$\overline{\beta}_i := \begin{cases} \beta_i + \beta_s \alpha_i, & i \neq s \\ \beta_s \alpha_s, & i = s \end{cases}. \tag{$\star$}$$

Como $\{\boldsymbol{a}^{j_i}\}_{j_i \in B}$ é l.i., segue que $\overline{\boldsymbol{\beta}} = \mathbf{0}$. Além disso, $\alpha_s \neq 0$, por hipótese, e da segunda linha de $(\star)$, teremos $\beta_s = 0$. Assim, da primeira linha, $0 = \overline{\beta}_i + \beta_s\alpha_i = \beta_i$.

Portanto $\overline{\boldsymbol{\beta}} = \mathbf{0}$, concluindo daí que $\overline{B}$ é base. ■

Deste lema e da subseção anterior, $\overline{B}$ gerado pela equação (4.7) é uma base viável, o que nos fornece ferramentas para construir um algoritmo baseado na geração de vértice $\boldsymbol{x}^i$ nunca "piores", ou seja: $\boldsymbol{c}'\boldsymbol{x}^i \leqq \boldsymbol{c}'\boldsymbol{x}^{i+1}$.

4.1.3 Como gerar a primeira base viável

Os resultados das duas subseções anteriores conseguimos detectar vértices ótimos, semi-retas de ilimitação e, se necessário, atualizar uma base viável para outra nunca pior (i.e., com valor objetivo nunca menor que a base anterior - no caso de degenerescência de vértice pode empatar, mas quando o vértice é não degenerado a valor objetivo é estritamente maior).

Deste modo, o problema ainda não resolvido é:

1. detectar conjunto inviável ($X = \emptyset$);
2. se conjunto viável, encontrar uma primeira base viável.

Para isso usaremos variáveis artificiais para conseguirmos $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$. Se $\boldsymbol{b} \geqq \mathbf{0}$, bastaria tomarmos $\boldsymbol{y} := \boldsymbol{b}$ e considerarmos o sistema $[\mathbf{AI}]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} = \boldsymbol{b}$. Entretanto, se para algum i tivessemos $b_i < 0$, ficaríamos com $y_i < 0$!

Deste modo, podemos resolver este problema de dois modos: multiplicar -1 as linhas i, do sistema $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, que tiverem $b_i < 0$; ou usar como coluna de i o vetor $-\boldsymbol{e}^i$. No restante do texto adotaremos a segunda solução.

Assim, dado um problema $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{c}) \in \mathbb{R}^{m\times n} \times \mathbb{R}^m \times \mathbb{R}^n$ na forma canônica, montaremos um problema auxiliar (fase 0), que resolverá os itens 1 e 2 acima apontados.

O problema **fase 0** é

$$\text{seja } \mathbf{B} \in \mathbb{R}^{m\times m} \text{ a matriz cujas colunas sejam definidas por } \boldsymbol{b}^i := \begin{cases} \boldsymbol{e}^i, & b_i \geqq 0 \\ -\boldsymbol{e}^i, & b_i < 0 \end{cases} \tag{4.8}$$

$$(P) \quad \left\{ \begin{array}{ll} \max \mathbf{1}'\boldsymbol{y} & \mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{y} = \boldsymbol{b} \\ & (\boldsymbol{x}, \boldsymbol{y}) \geqq \mathbf{0} \end{array} \right\} \equiv \left\{ \begin{array}{ll} \max(\mathbf{0}'; \mathbf{1}')\boldsymbol{z} & \overline{\mathbf{A}}\boldsymbol{z} = \boldsymbol{b} \\ & \boldsymbol{z} \geqq \mathbf{0} \end{array} \right. \tag{4.9}$$

que é resolvido através do esquema de Simplex deduzido anteriormente: pois para este problema a base inicial são as últimas colunas de $\overline{\mathbf{A}}$: como $\mathbf{B}$ é uma matriz identidade com linhas multiplicadas por -1, $\mathbf{B}^{-1} = \mathbf{B}$, e assim $\overline{\mathbf{A}}_B = \mathbf{B}$ e $\overline{\mathbf{A}}_B^{-1} = \mathbf{B}$.

Podemos mostrar que este problema inicial retorna um valor negativo, $-\|\boldsymbol{b}\|_1$, quando $X = \emptyset$ e 0 quando $X \neq \emptyset$.

Lema 4.9 *Sendo v o valor ótimo de (P),* $X = \emptyset \iff v = -\|\boldsymbol{b}\|_1 < 0$.

Lema 4.10 *Sendo v o valor ótimo de (P),* $X \neq \emptyset \iff v = 0$.

Se o problema fase 0 retornar $v = 0$, podemos construir uma base viável inicial, tomando

$$B := (i : (\overline{\mathbf{A}}^{-1}\boldsymbol{b})_i > 0, i \leqq n).$$

Note que:

1. as componentes de $\boldsymbol{z} = (\boldsymbol{x}; \boldsymbol{y})$ são nulas para $i > n$, i.e., $z_{j+n} = y_j = 0$, $j \in \{1, \ldots, m\}$);
2. se $\#B < m$, será necessário completar a base com colunas de $\mathbf{A}$, neste caso o vértice inicial será degenerado.

Exercício 4.2 *Prove a afirmação 1 acima.*

Exercício 4.3 *Mostre o porquê da validade da afirmação 2 acima.*

4.2 Algoritmo Simplex

Da discussão anterior poderíamos estabelecer alguma melhoria no algoritmo 4.1 (na página 2):

Algoritmo 4.2 Método Simplex - atualização de base viáveis, enumerando vértices "melhores"

1. *Seja* $B = (j_1, j_2, \ldots, j_m)$ *uma base viável*
   $i \leftarrow 1$
2. *Calcule* $\mathbf{A}_B^{-1}$ *e* $\overline{\boldsymbol{b}} \leftarrow \mathbf{A}_B^{-1}\boldsymbol{b}$
3. *Defina* $\boldsymbol{x}^i$ *por:* $\boldsymbol{x}^i_B \leftarrow \overline{\boldsymbol{b}}$ *e* $\boldsymbol{x}^i_N \leftarrow \mathbf{0}$
   Defina $\boldsymbol{\gamma}'_N := \boldsymbol{c}'_N - \boldsymbol{c}'_B\mathbf{A}_B^{-1}\mathbf{A}_N$
4. *Se* $\boldsymbol{\gamma}_N \leqq \mathbf{0}$, *então*
   Pare: $\boldsymbol{x}^i$ *é vértice ótimo*
5. *Senão Sejam:* $i \in N$ *para o qual* $\gamma_k > 0$ *e* $\boldsymbol{\pi} \leftarrow \mathbf{A}_B^{-1}\boldsymbol{a}^k$
   Se $\boldsymbol{\pi} \leqq \mathbf{0}$, *então*
   Seja $\boldsymbol{h}$ *definida por* $\boldsymbol{h}_B \leftarrow -\boldsymbol{\pi}$ *e* $\boldsymbol{h}_N \leftarrow \boldsymbol{e}^k$
   Pare: $\boldsymbol{x}^i + \lambda\boldsymbol{h}$ $(\lambda \geqq 0)$ *semi-reta de ilimitação*
   Senão Sejam $s \in \arg\min\left\{\frac{\overline{b}_i}{\pi_i} : i \in B \text{ e } \pi_i > 0\right\}$ *e* $\lambda \leftarrow \frac{\overline{b}_s}{\pi_s}$
   Defina $\boldsymbol{x}^{i+1}$ *por:* $\boldsymbol{x}^{i+1}_B \leftarrow \overline{\boldsymbol{b}} - \lambda\boldsymbol{\pi}$ *e* $\boldsymbol{x}^{i+1}_N \leftarrow \lambda\boldsymbol{e}^k$
   Sejam $B \leftarrow B - \{s\} + \{k\} \equiv (j_1, \ldots, j_{s-1}, k, j_{s+1}, \ldots, j_m)$ *e* $i \leftarrow i + 1$
   Volte ao passo 2

Algumas observações importantes:

1. Note que o passo 2 é o mais caro e por isso merece maior atenção.
2. Note também que $\boldsymbol{a}^k$ na base B é: $\boldsymbol{a}^k = \mathbf{A}_B\boldsymbol{\pi}$, pois $\boldsymbol{\pi} := \mathbf{A}_B^{-1}\boldsymbol{a}^k$
3. Se $\mathbf{P}$ for o produto de matrizes elementares (de pivotações) tal que $\mathbf{P}\mathbf{A}_B = \mathbf{I}$, então
$$[\mathbf{A}_B|\mathbf{A}_N|\mathbf{I}] \xrightarrow{\mathbf{P}} [\mathbf{I}|\mathbf{A}_B^{-1}\mathbf{A}_N|\mathbf{A}_B^{-1}] = [\mathbf{I}|\cdots\boldsymbol{\pi}\cdots|\mathbf{A}_B^{-1}]$$
4. Se notarmos que na definição de $\boldsymbol{\gamma}_N := \boldsymbol{c}_N - \mathbf{A}'_N(\mathbf{A}_B^{-1})'\boldsymbol{c}_B$, podemos também definir $\boldsymbol{\gamma}_B := \boldsymbol{c}_B - \mathbf{A}'_B(\mathbf{A}_B^{-1})'\boldsymbol{c}_B = \mathbf{0}$. Deste modo, podemos pensar em atualizar $\boldsymbol{\gamma}_N$ também através da pivotação. Para isso definiremos $\boldsymbol{\lambda} := (\mathbf{A}_B^{-1})'\boldsymbol{c}_B$, e estenderemos a matriz $\mathbf{A}_{m\times n}$ para uma $\overline{\mathbf{A}}_{m+1\times n+1}$ e o vetor $\boldsymbol{b}$ para $(0;\boldsymbol{b})'$,
$$\overline{\mathbf{A}} := \left[\begin{array}{cc|c} 1 & -\boldsymbol{c}' & 0 \\ \hline \mathbf{0} & \mathbf{A} & \boldsymbol{b} \end{array}\right], \text{ pois, } \overline{\mathbf{A}}_B^{-1} = \begin{bmatrix} 1 & \boldsymbol{\lambda}' \\ \mathbf{0} & \mathbf{A}_B^{-1} \end{bmatrix} \Longleftrightarrow \overline{\mathbf{A}}_B\overline{\mathbf{A}}_B^{-1} = \begin{bmatrix} 1 & \boldsymbol{\lambda}' - \boldsymbol{c}'_B\mathbf{A}_B^{-1} \\ \mathbf{0} & \mathbf{I}_{m\times m} \end{bmatrix} = \mathbf{I}_{m+1\times m+1}. \quad (4.10)$$
Com a matriz estendida podemos obter todas as variáveis do sistema através do produto abaixo,
$$\begin{bmatrix} 1 & \boldsymbol{\lambda}' \\ \mathbf{0} & \mathbf{A}_B^{-1} \end{bmatrix}\left[\begin{array}{c|c|c|c} 1 & -\boldsymbol{c}'_B & -\boldsymbol{c}'_N & 0 \\ \hline \mathbf{0} & \mathbf{A}_B & \mathbf{A}_N & \boldsymbol{b} \end{array}\right] = \left[\begin{array}{c|c|c|c} 1 & -\boldsymbol{c}'_B + \boldsymbol{\lambda}'\mathbf{A}_B & -\boldsymbol{c}'_N + \boldsymbol{\lambda}'\mathbf{A}_N & \boldsymbol{\lambda}'\boldsymbol{b} \\ \hline \mathbf{0} & \mathbf{A}_B^{-1}\mathbf{A}_B & \mathbf{A}_B^{-1}\mathbf{A}_N & \mathbf{A}_B^{-1}\boldsymbol{b} \end{array}\right] = \left[\begin{array}{c|c|c|c} 1 & 0 & \boldsymbol{\gamma}' & \boldsymbol{c}'_B\mathbf{A}_B^{-1}\boldsymbol{b} \\ \hline \mathbf{0} & \mathbf{I} & \mathbf{A}_B^{-1}\mathbf{A}_N & \mathbf{A}_B^{-1}\boldsymbol{b} \end{array}\right].$$
5. Será que o procedimento acima, que denominamos apresadamente de algoritmo, efetivamente pára ?

Nosso objetivo atual é conseguir transformar $\mathbf{P}$ em uma matriz $\mathbf{P}'$ para a qual: $\mathbf{P}'\mathbf{A}_{\overline{B}} = \mathbf{I}$. Para isso basta construir uma matriz elementar que pivote o vetor coluna $\boldsymbol{\pi}$, usando como pivô $\pi_s > 0$. A razão disso é esquematizada abaixo:

se $\overline{\mathbf{P}}[e^1|\cdots|e^{s-1}|\boldsymbol{\pi}|e^{s+1}|\cdots|e^m] = [e^1|\cdots|e^{s-1}|e^s|e^{s+1}|\cdots|e^m] = \mathbf{I}$, então $[\mathbf{A}_{\overline{B}}|\mathbf{A}_{\overline{N}}|\mathbf{I}] \xrightarrow{\overline{\mathbf{P}}} [\mathbf{I}|\mathbf{A}_{\overline{B}}^{-1}\mathbf{A}_{\overline{N}}|\mathbf{A}_{\overline{B}}^{-1}]$

Deste modo, para obter $\mathbf{A}_{\overline{B}}^{-1}$ a partir de $\mathbf{A}_B^{-1}$ basta fazer uma pivotação na sua coluna k, segundo o vetor $\boldsymbol{\pi}$ como pivô π_s. Devemos "pivotar" a coluna $\boldsymbol{\pi}$ de modo a produzir a e^s: $P_s(\boldsymbol{\pi}) = e^s \implies P_s(\mathbf{A}_B^{-1}) = \mathbf{A}_{\overline{B}}^{-1}$

Uma vez conhecido um bom método de atualização das inversas de bases, agora em ordem $o(n^2)$, devemos nos preocupar com a questão 4 acima: será que o procedimento anterior é finito ?

Esta questão será respondida com o exemplo 4.4. No próximo exemplo examinaremos uma atualização de inversa a partir de uma pivotação.

Exemplo 4.1 *Para a matriz* $\mathbf{A}$ *abaixo definida e a inversa da base* $B = (1,2,4)$, *obter a inversa de* $\overline{B} = (1,3,4)$, *isto é, entra a coluna* 3 *e sai a coluna* 2. *Logo o pivô deve ser o elemento* $a_{3,2}$.

$$\mathbf{A} := \begin{bmatrix} \frac{1}{2} & -\frac{11}{2} & -\frac{5}{2} & 9 \\ \frac{1}{2} & -\frac{3}{2} & -\frac{1}{2} & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad \textit{então:} \; \mathbf{A}_B = \begin{bmatrix} \frac{1}{2} & -\frac{11}{2} & 9 \\ \frac{1}{2} & -\frac{3}{2} & 1 \\ 0 & 0 & 2 \end{bmatrix}, \; \mathbf{A}_B^{-1} = \begin{bmatrix} -\frac{3}{4} & \frac{11}{4} & 2 \\ -\frac{1}{4} & \frac{1}{4} & 1 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}, \; \mathbf{A}_B^{-1}a^3 = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix}.$$

Pivotando a inversa segundo o elemento $a_{3,2}$, *obteremos*

$$\mathbf{A}_B^{-1} = \left[\begin{array}{ccc|c} -\frac{3}{4} & \frac{11}{4} & 2 & \frac{1}{2} \\ -\frac{1}{4} & \frac{1}{4} & 1 & \boxed{\frac{1}{2}} \\ 0 & 0 & \frac{1}{2} & 0 \end{array}\right] \xrightarrow{l_1 - l_2} \left[\begin{array}{ccc|c} -\frac{1}{2} & \frac{5}{2} & 1 & 0 \\ -\frac{1}{4} & \frac{1}{4} & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 \end{array}\right] \xrightarrow{2l_2} \left[\begin{array}{ccc|c} -\frac{1}{2} & \frac{5}{2} & 1 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 2 & 1 \\ 0 & 0 & \frac{1}{2} & 0 \end{array}\right] = \mathbf{A}_{\overline{B}}^{-1}.$$

Apenas para conferir o resultado da pivotação, podemos testar o produto

$$\mathbf{A}_{\overline{B}}^{-1}\mathbf{A}_{\overline{B}} = \begin{bmatrix} -\frac{1}{2} & \frac{5}{2} & 1 \\ -\frac{1}{2} & \frac{1}{2} & 2 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & -\frac{5}{2} & 9 \\ \frac{1}{2} & -\frac{1}{2} & 1 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Exercício 4.4 *Considere o seguinte (PLC)*

$$\begin{array}{llllllllll} \max & x_1 - 7x_2 - x_3 - 2x_4 - 25x_5 & & & & & & & & \\ & \frac{1}{2}x_1 - \frac{11}{2}x_2 - \frac{5}{2}x_3 + 9x_4 + x_5 + x_6 & & & = & 0 \\ & \frac{1}{2}x_1 - \frac{3}{2}x_2 - \frac{1}{2}x_3 + x_4 + x_5 + \quad + x_7 & & & = & 0 \\ & \qquad\qquad 2x_4 & + & x_8 & = & 1 \\ & & & x_i \geqq 0 & , & i \in \{1,2,\ldots,8\} \end{array}$$

Para este problema, implemente o procedimento[6] *4.2, usando como base inicial* $B_0 := (6,7,8)$, *usando como sequência de entrada/saída na base a tabela abaixo. Se as contas estiverem corretas, deverão obter: todos os* $\gamma_k > 0$ *e* $\frac{\bar{b}_s}{\pi_s} = \min_{i \in B \wedge \pi_i > 0}\left\{\frac{\bar{b}_i}{\pi_i}\right\}$.

Iteração i	*Entra na base (k)*	*Sai da base (s)*	*Base* B_i
0			$B_0 = (6,7,8)$
1	x_1	x_6	$B_1 = (1,7,8)$
2	x_2	x_7	$B_2 = (1,2,8)$
3	x_3	x_1	$B_3 = (3,2,8)$
4	x_4	x_2	$B_4 = (3,4,8)$
5	x_6	x_3	$B_5 = (6,4,8)$
6	x_7	x_4	$B_6 = (6,7,8)$

Para evitar o problema ilustrado no exercício acima (**ciclagem**), uma regra bastante simples é selecionar sempre os primeiros índices:

[6] Sugiro usar o Scilab a partir do exemplo http://www.ime.usp.br/~leo/mac315/00/exemplos/pivoteamento1.sci.

Regra de Bland[7] - entrada: selecione o menor índice k, tal que $\gamma_k > 0$
saída: selecione o menor índice s, tal que $\min\limits_{i \in B \wedge \pi_i > 0} \left\{ \frac{\overline{b}_i}{\pi_i} \right\}$.

4.2.1 Simplex Tabular

Da discussão anterior podemos enunciar um algoritmo "manual" para resolver problemas canônicos, algoritmo este denominado **Simplex Tabular**.

$$\left.\begin{array}{ll} \max & \boldsymbol{c}'\boldsymbol{x} \\ & \mathbf{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geqq \mathbf{0} \end{array}\right\} \equiv \left\{ \begin{array}{l} \max \quad z \\ \begin{bmatrix} 1 & -\boldsymbol{c}' \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} z \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{b} \end{bmatrix} \\ \boldsymbol{x} \geqq \mathbf{0} \end{array} \right.$$

Dada uma base viável inicial[8] B, deve-se montar o primeiro **tableau**

$$\left[\begin{array}{cccc|c} -c_1 & -c_2 & \cdots & -c_n & z \\ \hline a_{1,1} & a_{1,2} & \cdots & a_{1,n} & b_1 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & b_m \end{array}\right] \quad \begin{array}{c} \text{multiplicada (pela esquerda) por } \overline{\mathbf{A}}, \\ \longrightarrow \\ \text{dada pela equação (4.10), resulta em} \end{array} \quad \left[\begin{array}{cccc|c} -\gamma_1 & -\gamma_2 & \cdots & -\gamma_n & \boldsymbol{\lambda}'\boldsymbol{b} = \boldsymbol{c}_B'\overline{\boldsymbol{b}} \\ \hline \overline{a}_{1,1} & \overline{a}_{1,2} & \cdots & \overline{a}_{1,n} & \overline{b}_1 \\ \overline{a}_{2,1} & \overline{a}_{2,2} & \cdots & \overline{a}_{2,n} & \overline{b}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \overline{a}_{m,1} & \overline{a}_{m,2} & \cdots & \overline{a}_{m,n} & \overline{b}_m \end{array}\right]$$

sendo $\overline{a}_{ij} := (\mathbf{A}_B^{-1}\mathbf{A})_{ij}$, e lembrando que $\overline{\boldsymbol{b}} := \mathbf{A}_B^{-1}\boldsymbol{b}$ e $\boldsymbol{\lambda}' := \boldsymbol{c}_B'\mathbf{A}_B^{-1}$.

Para ilustrarmos o Simplex Tabular, consideraremos um caso simples (que pode ser resolvido graficamente).

Exemplo 4.2 *Considere o problema linear* $\max\{x_1 + x_2 : -x_1 + x_2 \geqq 2 \wedge x_2 \leqq 2 \wedge \boldsymbol{x} \geqq \mathbf{0}\}$. *É fácil ver que a solução ótima deste problema é dada pelo vértice* $\overline{\boldsymbol{x}} = (0, 2)'$.

Colocando na forma canônica e usando $B = (1, 4)$ *como base viável inicial (por ser mais simples) obteremos,*

$$\left.\begin{array}{l} \max -x_1 + x_2 \\ \quad x_1 \;\; + x_2 - x_3 \qquad = 2 \\ \qquad\quad x_2 \qquad + x_4 = 2 \\ \quad x_1 \;\; , \; x_2 \; , \; x_3 \; , \; x_4 \geqq 0 \end{array}\right\} \Longrightarrow \mathbf{A} = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \mathbf{A}_B = \mathbf{I} = \mathbf{A}_B^{-1} \text{ e } \boldsymbol{\lambda}' = \boldsymbol{c}_B'\mathbf{A}_B^{-1} = (c_1, c_4) = (-1, 0).$$

Deste modo $\boldsymbol{\gamma}_N' = \boldsymbol{c}_N' - \boldsymbol{\lambda}'\mathbf{A}_N = (c_2, c_3) - (\lambda_1 + \lambda_2, -\lambda_1) = (1, 0) - (-1, 1) = (2, -1)$, *e o tableau inicial fica na forma*

$$\begin{array}{ccccc} \boldsymbol{x}^1 = (\underline{2}, 0, 0, \underline{2}) & & \boldsymbol{x}^2 = (0, \underline{2}, 0, \underline{0}) & & \boldsymbol{x}^3 = (0, \underline{2}, \underline{0}, 0) \\ \left[\begin{array}{cccc|c} 0 & \boxed{-2} & 1 & 0 & -2 \\ \hline 1 & \boxed{1} & -1 & 0 & 2 \\ 0 & 1 & 0 & 1 & 2 \end{array}\right] & \begin{array}{l} \textit{entra } x_2 \\ \textit{sai } x_1 \\ \end{array} & \left[\begin{array}{cccc|c} 2 & 0 & \boxed{-1} & 0 & 2 \\ \hline 1 & 1 & -1 & 0 & 2 \\ -1 & 0 & \boxed{1} & 1 & 0 \end{array}\right] & \begin{array}{l} \textit{entra } x_3 \\ \textit{sai } x_4 \\ \end{array} & \left[\begin{array}{cccc|c} 1 & 0 & 0 & 1 & 2 \\ \hline 1 & 1 & -1 & 0 & 2 \\ -1 & 0 & 1 & 1 & 0 \end{array}\right]. \end{array}$$

O algoritmo parou na última configuração em razão obtermos $\boldsymbol{\gamma}_N = (-1, -1)' \leqq \mathbf{0}$. *Note que a segunda iteração trocou a base de um mesmo vértice e finalmente detectando otimalidade.*

Mais ainda, como consideramos o mesmo (PLC) do exercício 4.1, esta é a resposta ao seu item 3*: na iteração* 2*, apesar do vértice* $\boldsymbol{x} = (0, 2, 0, 0)$ *ser ótimo, em função de usarmos a base* $B = (2, 4)$ *resulta* $\boldsymbol{\gamma}_N = (-2, 1)' \nleqq \mathbf{0}$, *enquanto que para a base* $B = (2, 3)$ *temos* $\boldsymbol{\gamma}_N = (-1, -1)' \leqq \mathbf{0}$.

---

[7] Para ver uma demonstração de que esta regra evita ciclagem, consulte a seção 5.3 de *Programação Linear: um primeiro curso*, de Carlos Humes Jr. e Ana Flora P. de Castro Humes, apresentado como mini-curso no *IX Congresso Nacional de Matemática Aplicada e Computacional*, realizado em Brasília em agosto de 1986.

[8] É sempre possível introduzir m variáveis artificiais para obter esta base.