EP2 de Programação Linear

Gustavo Chicato Wandeur - 7557797
Vinícius Bitencourt Matos - 8536221

18 de maio de 2015

1 Definições

Definição 1. Um **poliedro** é um conjunto $S \subseteq \mathbb{R}^n$ limitado por um número finito de restrições lineares de igualdade ($\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i$) ou desigualdade ($\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i$ ou $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i$).

Definição 2. Um **problema de programação linear** consiste em minimizar, sobre todos os vetores $\boldsymbol{x} \in \mathbb{R}^n$ que satisfazem dado conjunto de restrições lineares, uma função linear $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$, sendo $\boldsymbol{c} \in \mathbb{R}^n$ dado. Cada restrição linear é da forma $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i$, $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i$ ou $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i$.

O vetor c é chamado *vetor de custos*, e a função $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ é chamada *função objetivo* ou *função de custos*.

Dizemos que $\boldsymbol{x} \in \mathbb{R}^n$ é um *ponto viável* ou *solução viável* se $\boldsymbol{x}$ satisfaz a todas as restrições do problema de programação linear. O conjunto de todos os pontos viáveis é chamado *conjunto viável*. Segue das definições que o conjunto viável de um problema de programação linear é um poliedro. Um problema é *inviável* se seu conjunto viável é vazio.

Uma solução viável $\boldsymbol{x}^*$ que minimiza a função objetivo é chamada *solução ótima*, e o custo correspondente $\boldsymbol{c}^T\boldsymbol{x}$ é chamado *custo ótimo*.

Definição 3. Uma restrição é dita *ativa em* $\boldsymbol{x}^*$ se é satisfeita no ponto $\boldsymbol{x}^*$ por igualdade.

Definição 4. Um conjunto de restrições $\{\boldsymbol{a}_i^T\boldsymbol{x} \leq b_i\}_{i\in I_1} \cup \{\boldsymbol{a}_i^T\boldsymbol{x} \geq b_i\}_{i\in I_2} \cup \{\boldsymbol{a}_i^T\boldsymbol{x} = b_i\}_{i\in I_3}$ (onde I_1, I_2, I_3 são conjuntos de índices disjuntos) é **linearmente independente** se $\{\boldsymbol{a}_i\}_{i\in I_1\cup I_2\cup I_3}$ for um conjunto de vetores de $\mathbb{R}^n$ linearmente independentes, isto é, se $\sum_{k=1}^n \lambda_k \boldsymbol{a}_{ik} = \boldsymbol{0}$ apenas se $\lambda_1 = \cdots = \lambda_n = 0$.

Definição 5. Um ponto $\boldsymbol{x}^* \in \mathbb{R}^n$ é uma *solução básica* se todas as restrições de igualdade são ativas em $\boldsymbol{x}^*$ e, além disso, dentre todas as restrições ativas em $\boldsymbol{x}^*$, há no mínimo n linearmente independentes.

Definição 6. Diz-se que um problema de programação linear está no **formato padrão** se todas as variáveis são obrigatoriamente não negativas (ou seja, há uma restrição $x_i \geq 0$ para todo $i = 1, 2, \ldots, n$) e todas as demais restrições são de igualdade (isto é, da forma $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} = b_i$).

De modo compacto, podemos descrever um problema de programação linear no formato padrão como segue:

$$\begin{array}{rc} \text{minimizar} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{sujeito a} & \boldsymbol{A}\boldsymbol{x} = b \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array},$$

onde $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ é uma matriz cujas linhas são os vetores $\boldsymbol{a}_i$, $i = 1, \ldots, m$, e $\boldsymbol{x} \geq \boldsymbol{0}$ é interpretado componente a componente.

Qualquer problema de programação linear pode ser reescrito no formato padrão por meio do seguinte procedimento:

- Cada variável *livre* (cujo sinal não é restrito originalmente) x_j é substituída por $x_j^+ - x_j^-$, em que x_j^+ e x_j^- são variáveis novas e com restrição de serem ambas não negativas.

- Cada restrição da forma $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \leq b_i$ é substituída por $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} + s_i = b_i$, em que a nova variável s_i (variável de *folga*) é não negativa. Analogamente, cada restrição da forma $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} \geq b_i$ é trocada por $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} - s_i = b_i$, em que a nova variável s_i (variável de *sobra*) é não negativa.

Ademais, não há perda em impor que as linhas de $\boldsymbol{A}$ sejam linearmente independentes, uma vez que a dependência linear entre restrições nesse formato sempre leva a um dos seguintes casos:

- Há restrições redundantes, que podem ser simplesmente removidas sem alteração do problema;

- Há restrições incompatíveis, que tornam o problema inviável.

Uma vez que todo problema de programação linear pode ser expresso no formato padrão e com a matriz $\boldsymbol{A}$ com linhas linearmente independentes, é suficiente ter um método que resolve problemas desse tipo.

Consideremos o poliedro definido pelas restrições $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ e $\boldsymbol{x} \geq \boldsymbol{0}$, em que $A \in \mathbb{R}^{m\times n}$ tem posto completo. Então, existem índices $B(1), \ldots, B(m)$ tais que as colunas $\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(m)}$ são linearmente independentes e $x_i = 0$ para todo $i \notin B(1), \ldots, B(m)$.

Se $\boldsymbol{x}$ é uma solução básica, então as variáveis $x_{B(1)}, \ldots, x_{B(m)}$ são chamadas *variáveis básicas*, as demais são chamadas *variáveis não básicas*.

2 Método simplex revisado

1. Em cada iteração, começamos com uma base correspondente às colunas $\boldsymbol{A}_{B(1)}, \ldots, \boldsymbol{A}_{B(m)}$, uma solução viável básica $\boldsymbol{x}$ e a inversa $\boldsymbol{B}^{-1}$ da matriz básica.

2. Calculamos primeiramente o vetor $\boldsymbol{p}^{\mathrm{T}} = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{B}^{-1}$. Em seguida, obtemos os custos reduzidos $\bar{c}_j = c_j - \boldsymbol{p}^{\mathrm{T}}\boldsymbol{A}_j$. Se todos forem positivos, então a solução viável básica $\boldsymbol{x}$ é ótima, e o algoritmo é encerrado. Caso contrário, escolhemos o menor valor de j que satisfaça $\bar{c}_j < 0$.

3. Calculamos $\boldsymbol{u} = \boldsymbol{B}^{-1}\boldsymbol{A}_j$. Se nenhum componente de $\boldsymbol{u}$ for positivo, então a direção de redução de custos $-\boldsymbol{u}$ é viável para todo θ positivo. Logo o custo ótimo é $-\infty$ e o algoritmo termina.

4. Caso contrário, se temos ao menos uma componente positiva de $\boldsymbol{u}$, então

$$\theta^* = \min_{\{i=1,\ldots,m \mid u_i > 0\}} \frac{x_{B(i)}}{u_i}.$$

5. Seja l tal que $\theta^* = \dfrac{x_{B(l)}}{u_l}$. Formamos uma nova base trocando[1] a coluna $\boldsymbol{A}_{B(l)}$ por $\boldsymbol{A}_j$. Teremos então uma nova solução viável básica $\boldsymbol{y}$ com componentes $y_j = \theta^*$ e $y_{B(i)} = x_{B(i)} - \theta^* u_i$, $i \neq l$.

6. Montamos uma matriz na forma $\left[\boldsymbol{B}^{-1} \mid u\right]$. São realizadas então operações elementares de linha, adicionando a cada uma um múltiplo da l-ésima, a fim de que a última coluna termine como o vetor canônico $\boldsymbol{e}_l$. As m primeiras colunas resultantes correspondem à matriz $\boldsymbol{B}^{-1}$ atualizada. Retorna-se ao passo 1 e continuamos até encontrar uma condição de parada (2 ou 3).

3 O programa

3.1 Formato da entrada

O programa em Octave recebe um argumento na linha de comando, que corresponde ao nome de um arquivo de texto descrevendo o problema na seguinte ordem:

```
m n
A
b
c
x
```

O programa supõe que m e n são inteiros positivos, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ é uma matriz com *linhas linearmente independentes*, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{x} \in \mathbb{R}^n$. Além disso, supõe que $\boldsymbol{x}$ é uma solução viável básica do seguinte problema, que por hipótese *não possui soluções viáveis básicas degeneradas* e será resolvido pelo programa:

$$\begin{array}{rc} \text{minimizar} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{sujeito a} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

3.2 Exemplos de execução

3.2.1 Exemplo 1.8a do livro (adaptado)

A adaptação consistiu em transformar para o formato padrão. Solução viável básica inicial: $\boldsymbol{x} = (0,1,0)^{\mathrm{T}}$.

$$\begin{array}{rc} \text{minimizar} & x_1 + x_2 \\ \text{sujeito a} & -x_1 + x_2 + x_3 = 1 \\ & \boldsymbol{x} \geq 0 \end{array}$$

[1] O algoritmo descrito no livro para a atualização de B^{-1} depende que a nova coluna da matriz básica fique no lugar da anterior, consequentemente os índices $B(1),\dots,B(m)$ podem ficar desordenados. Optamos por manter o algoritmo dessa forma, evitando gasto desnecessário de tempo para reordenar os índices e as filas da matriz. Contudo, conforme pede o enunciado, a impressão final da solução ótima (ou da direção que leva o custo a $-\infty$) respeita a ordem inicial das variáveis.

O arquivo de entrada `exemplo-1.8a-pag-23.txt`, correspondente a este problema, é mostrado abaixo, conforme o formato descrito em 3.1:

```
1 3
-1 1 1
1
1 1 0
0 1 0
```

A variável básica inicialmente é x_2.

Calculando os custos reduzidos das variáveis não básicas, obtêm-se $\bar{c}_1 = 2$ e $\bar{c}_3 = -1$. A única escolha para entrar na base, portanto, é x_3, pois c_3 é o único custo reduzido negativo.

Agora, calcula-se o oposto do vetor de direções básicas $\boldsymbol{u} = \boldsymbol{B}^{-1} A_j$, que é $[1]$, e para cada componente positiva u_i (a única), calcula-se o quociente $\frac{x_{B(i)}}{u_i}$, escolhendo o menor possível para ser θ^*, que é o maior múltiplo da direção que ainda leva a um ponto dentro do poliedro. Seja l esse índice. Temos, então, $l = 1$, indicando $x_{B(1)}$ (isto é, x_2) sairá da base.

Após atualizarmos a matriz B^{-1} e a solução $x = (0, 0, 1)^{\mathrm{T}}$, o programa agora vai para a próxima iteração.

Ao calcular os custos reduzidos, obtém-se $\bar{c}_N = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, e como todas as coordenadas são positivas, o algoritmo para, pois a solução $\boldsymbol{x}$ atual é ótima.

3.2.2 Exemplo 1.8d do livro (adaptado)

O arquivo de entrada a ser usado neste problema é `exemplo-1.8d-pag-23.txt`.

Voltemos ao problema do exemplo anterior, com $x^{\mathrm{T}} = (0, 1, 0)$ inicial, mas agora considerando o vetor de custos $c^{\mathrm{T}} = (-1, -1)$. Calculando os respectivos custos reduzidos, temos $\bar{c}_1 = -2$ e $\bar{c}_3 = 1$. Isso significa que 1 é índice de redução de custos, e o escolhemos para entrar na base.

Calculando pelo método usual, $\boldsymbol{u} = -1 = -\boldsymbol{d}_B$. Como $\boldsymbol{u} < \boldsymbol{0}$ e, portanto, $\boldsymbol{d}_B > \boldsymbol{0}$, a variável básica única x_2 crescerá indefinidamente e, portanto, a direção viável correspondente $\boldsymbol{d} = (1, 1, 0)^{\mathrm{T}}$ será válida para qualquer valor de $\theta > 0$, com a função de custos tendo seu valor reduzido ad infinitum ao longo dela.

Assim, o algoritmo para, pois encontramos uma direção ao longo da qual o custo ótimo diverge para $-\infty$.