

Exemplo em R do método simplex para programação linear

Elias Teixeira Krainski

22 de dezembro de 2005

1 Exemplo típico

O exemplo típico é a otimização da produção de dois produtos a lucros diferentes que utilizam várias máquinas de forma diferente.

1.1 Apresentação do problema

Uma indústria fabrica dois produtos, I e II, e para fabricá-los, utiliza-se de três máquinas: A, B e C. Na tabela 1.1 pode-se observar o tempo de utilização em horas de cada uma das máquinas para a fabricação de ambos os produtos.

Produto	Tempo Máquina A	Tempo Máquina B	Tempo Máquina C
I	2	4	2
II	1	2	2

Tabela 1: Tempo de utilização das três máquinas para a fabricação dos produtos I e II

As três máquinas apresentam restrição quanto ao tempo de operação. Essas restrições estão na tabela 2.

Máquina	Tempo de utilização semanal (em horas)
A	160
B	120
C	280

Tabela 2: Tempo de utilização semanal em horas de cada uma das máquinas

O lucro fornecido pelo produto I é de um dólar enquanto que o lucro fornecido pelo produto II é de um dólar e meio. Desejamos fabricar ambos os produtos de forma que se obtenha o maior lucro possível.

1.2 Modelagem do problema

Na solução, vamos considerar x_1 e x_2 o número de unidades produzidas dos produtos I e II, respectivamente.

Maximizar a função lucro dada por:

$$L = x_1 + 1.5 * x_2 ,$$

Sujeito às restrições:

$$2x_1 + 2x_2 \leq 160$$

$$x_1 + 2x_2 \leq 120$$

$$4x_1 + 2x_2 \leq 280$$

$$x_1, x_2 \geq 0$$

Em notação matricial:

Maximizar:

$$L = cX \Rightarrow L = [1 \ 1.5] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Sujeito à:

$$Ax \leq b \Rightarrow \begin{bmatrix} 2 & 2 \\ 1 & 2 \\ 4 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 160 \\ 120 \\ 280 \end{bmatrix}$$

1.3 Solução do problema

A solução do problema pode ser feita utilizando o algoritmo simplex. Na linguagem R a função `simplex()`, disponível no pacote **boot**, implementa esse algoritmo.

```
> require(boot)
```

```
Loading required package: boot
```

```
[1] TRUE
```

```
> args(simplex)
```

```
function (a, A1 = NULL, b1 = NULL, A2 = NULL, b2 = NULL, A3 = NULL,
```

```
        b3 = NULL, maxi = FALSE, n.iter = n + 2 * m, eps = 1e-10)
```

```
NULL
```

A função `simplex()` possui vários argumentos, contendo todas as restrições $\{\leq, =, \geq\}$. Na tabela 3 temos o significado de cada argumento:

A solução para o problema proposto, pode então ser obtida pelo simples comando:

```
> simplex(c(1, 1.5), rbind(c(2, 2), c(1, 2), c(4, 2)), c(160, 120,
+      280), maxi = TRUE)
```

Argumento	Significado
a:	vetor de coeficientes da função objetivo
A1:	vetor ou matriz de coeficientes das restrições \geq (lado esquerdo)
b1:	número ou vetor de coeficientes das restrições \geq (lado direito)
A2:	vetor ou matriz de coeficientes das restrições \leq (lado esquerdo)
b2:	número ou vetor de coeficientes das restrições \leq (lado direito)
A3:	vetor ou matriz de coeficientes das restrições $=$ (lado esquerdo)
b3:	número ou vetor de coeficientes das restrições $=$ (lado direito)
maxi:	lógico indicador de minimização ou maximização
n.iter:	número máximo de iterações
eps:	tolerância ou precisão

Tabela 3: Argumentos da função `simplex()`

Linear Programming Results

```
Call : simplex(a = c(1, 1.5), A1 = rbind(c(2, 2), c(1, 2), c(4, 2)),
             b1 = c(160, 120, 280), maxi = TRUE)
```

Maximization Problem with Objective Function Coefficients

```
  x1  x2
1.0 1.5
```

Optimal solution has the following values

```
x1 x2
40 40
```

The optimal value of the objective function is 100.

O lucro máximo é obtido ao se produzir 40 unidades de cada produto. O lucro será de 100 dólares.