Programação linear | **Lista numero 04**
Problema linear | tarcisio@member.ams.org
T. Praciano-Pereira | **Dep. de Computação**

alun@:

12 de novembro de 2012	**Univ. Estadual Vale do Acaraú**
Escrita com LaTeX	sis. op. Debian/Gnu/Linux
www.otimizacao.sobralmatematica.org	

Se entregar em papel, por favor, prenda esta *folha de rosto* na sua solução desta lista, deixando-a em branco. Ela será usada na correção.

As questões desta lista foram feitas a partir do `help(lp)` dentro do `R`.

Exercícios 1 *Problema linear* *objetivo: Resolver um problema linear usando* `R`*.*

palavras chave: *pacote* `lpSolve` *Problema linear* `R` *uma linguagem de expressões.*

1. *Desigualdades lineares*

   Para definir o domínio dum problema linear *precisamos de desigualdades lineares, alguns exemplos são tratados nesta questão.*

   Na figura (1), página 2,

   podemos identificar um intervalo $[a, b]$ sobre o qual está definida a função $y = f(x)$, uma função do primeiro.

   (a) *(V)[ ](F)[ ] O intervalo $[a, b]$ é a solução do sistema*

   $$\begin{cases} x & < 0; \\ x & > -3; \end{cases} \tag{1}$$

   (b) *(V)[ ](F)[ ] O intervalo $[a, b]$ é a solução do sistema*

   $$\begin{cases} x & < b; \\ x & > a; \end{cases} \tag{2}$$

   (c) *(V)[ ](F)[ ] O valor mínimo de $y = f(x)$ em $[a, b]$ é $f(b)$.*

   (d) *(V)[ ](F)[ ] O valor mínimo de $y = f(x)$ em $[a, b]$ é $f(a)$.*

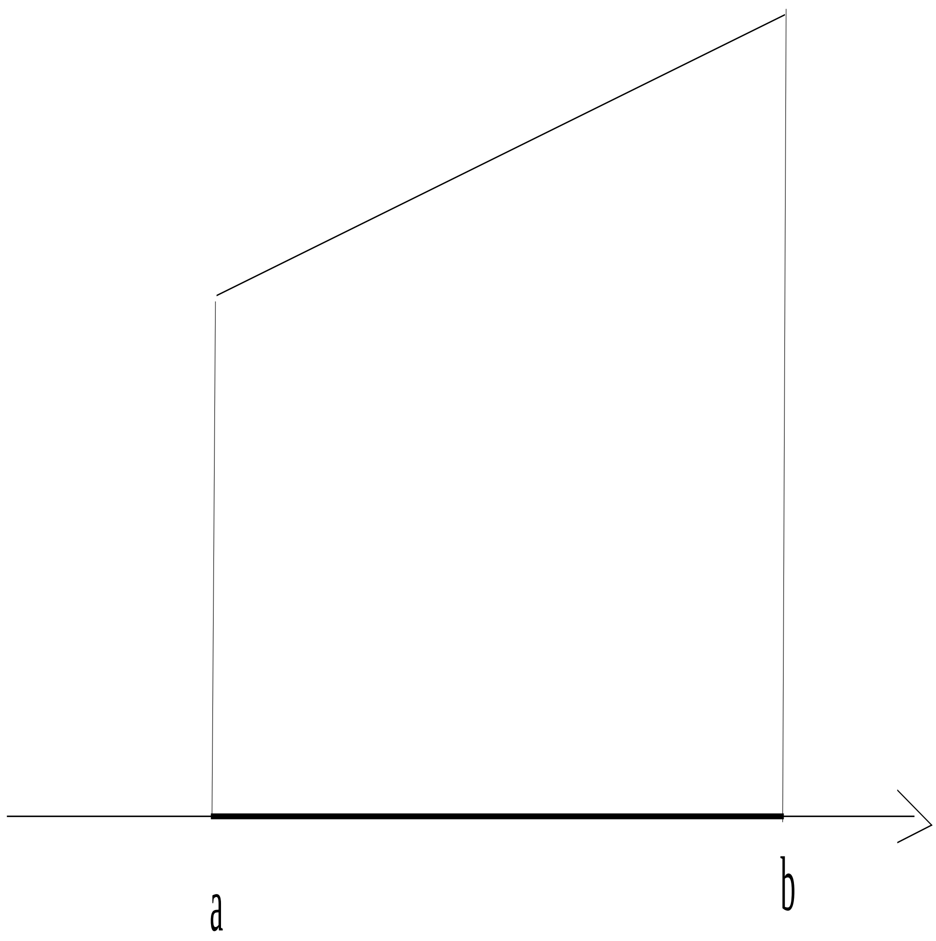

Figura 1: Uma desigualdade linear definindo $[a, b]$

(e) <u>(V)[ ](F)[ ]</u> *Como o mínimo de uma função linear sobre um conjunto convexo ocorre sobre a fronteira, a solução da desigualdade na equação (2) com a restrição $y = f(x)$ resulta na análise de apenas dois pontos: $f(a), f(b)$.*

2. <u>*restrições e função custo*</u>

   Nomenclatura: use como material básico algum dos textos sugeridos na lista 02.

   (a) <u>(V)[ ](F)[ ]</u> *Num problema de* programação linear *há um* poliedro *que é o* domínio *e uma* função custo *definida no domínio que é linear. Na questão* **??** *a função custo é*

$$y = f(x) = f(a) + m(x - a); m = \frac{f(b) - f(a)}{b - a};$$

(b) (V)[ ](F)[ ] *Num problema de* programação linear *há um* poliedro *que é o* domínio *cuja definição é feita com desigualdades lineares, por exemplo, na questão* **??** *podemos identificar o domínio* $[a,b]$ *definidos por duas desigualdades na equação (2). Estas desigualdades são chamadas* restrições.

(c) (V)[ ](F)[ ] *O sistema de desigualdades, (restrições),*

$$\left\{\begin{array}{cl} y & \leq 0; \\ x & \leq 0; \\ x+2y & \leq 2; \\ 4x+2y & \leq 5; \end{array}\right. \tag{3}$$

define um domínio Ω *que é um polígono convexo no plano, situado no terceiro quadrante.*

(d) (V)[ ](F)[ ] *O sistema de desigualdades, (restrições),*

$$\left\{\begin{array}{cl} y & \leq 0; \\ x & \leq 0; \\ x+2y & \leq 2; \\ 4x+2y & \leq 5; \end{array}\right. \tag{4}$$

define um domínio Ω *que é um polígono convexo no plano, situado no primeiro quadrante.*

(e) (V)[ ](F)[ ] *Uma função custo definida no domínio* Ω *referido no item 3 seria*

$$f(x) = 3x^2 + 4x + 9 \tag{5}$$

3. *Álgebra Linear*

(a) (V)[ ](F)[ ] *O domínio* Ω *do problema linear definido pelas restrições na equação (4) corresponde, aproximadamete, á figura (2), página 4,*

(b) (V)[ ](F)[ ] *O domínio* Ω *do problema linear definido pelas restrições na equação (4) corresponde, aproximadamente, á figura (3), página 5,*

(c) (V)[ ](F)[ ] *A função*

$$F(x,y) = \begin{pmatrix} 2 & 6 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{6}$$

é um candidado a função objetivo *definida no domínio* Ω *determinado pelas restrições das equações (4).*

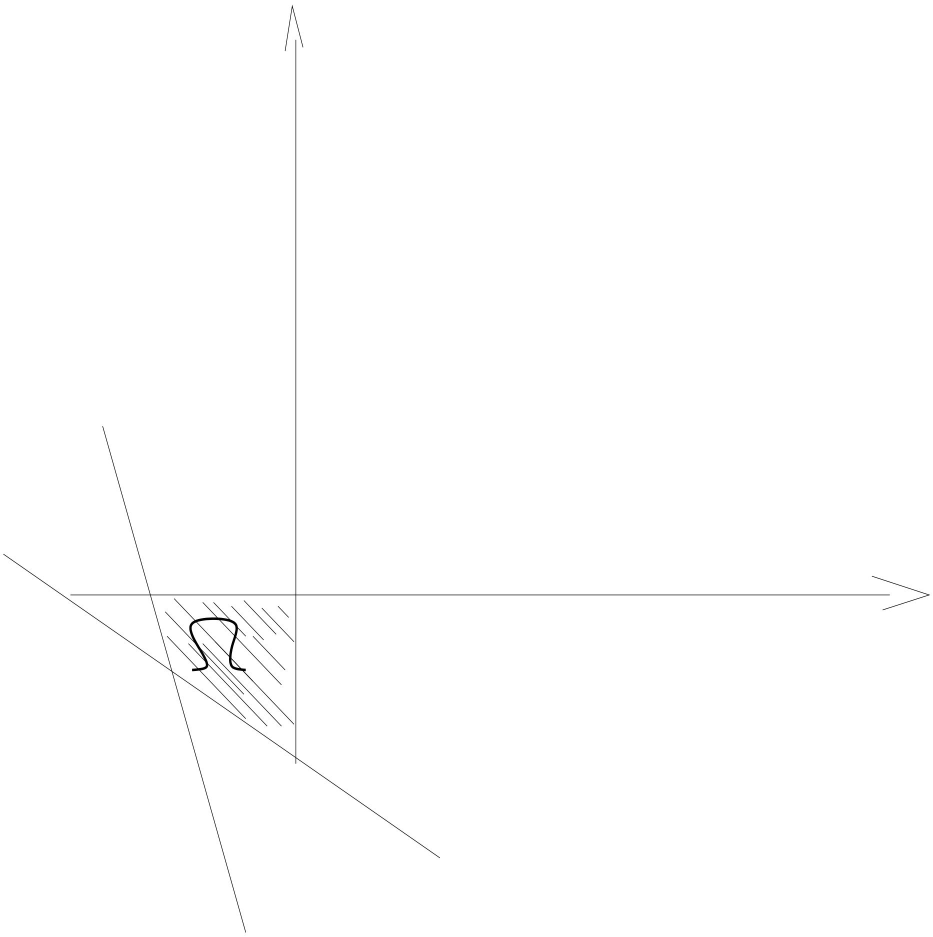

Figura 2: O domínio Ω

(d) *(V)[ ](F)[ ]* *A função*

$$F(x, y) = \begin{pmatrix} 2 & 6 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{7}$$

é um candidado a função objetivo *definida no domínio* Ω *determinado pelas restrições das equações (4).*

(e) *(V)[ ](F)[ ]* *O seguinte valor pode ser obtido por* F *sobre* Ω*:*

$$F(1, 3) = \begin{pmatrix} 7 \\ 9 \end{pmatrix} \tag{8}$$

4. *Álgebra Linear*

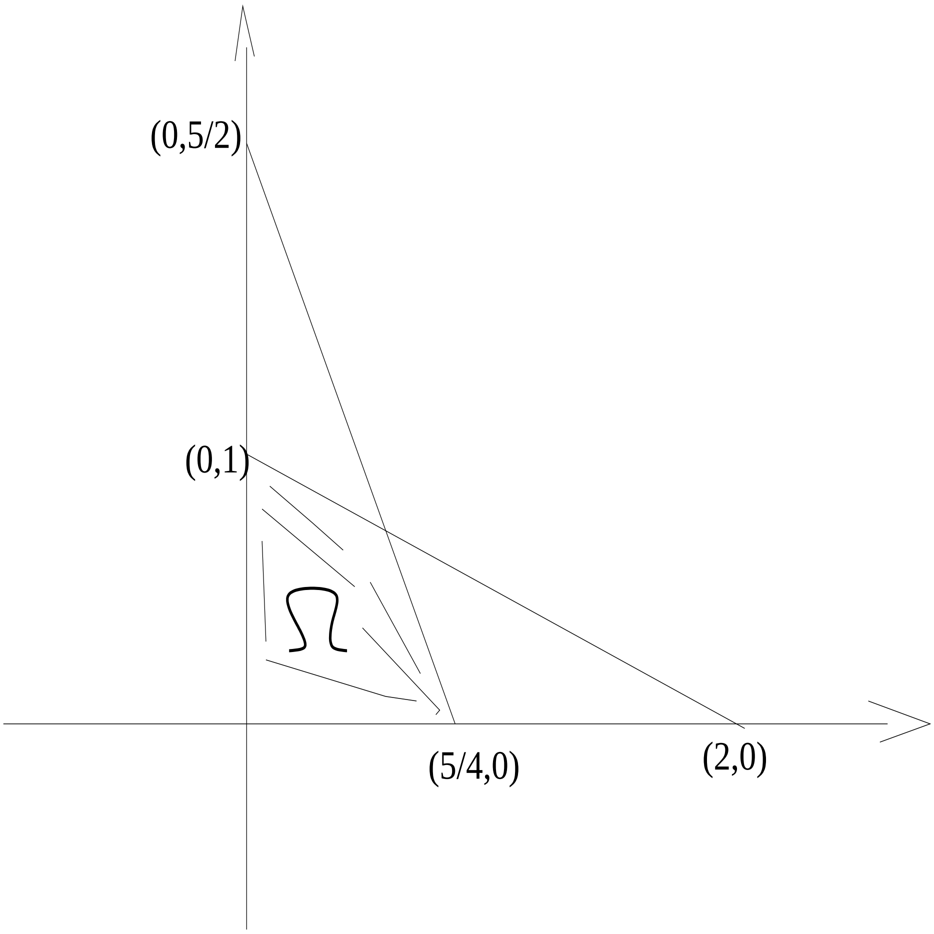

Figura 3: O domínio Ω

As restrições na equação (4) e a função de custo na equação (7) definem um problema **P**. *Use a* lista 03 *como fonte de consulta para as funções definidas nesta lista (as funções da linguagem* `R`*).*

(a) *(V)[ ](F)[ ] Os comandos seguintes do* `R` *traduzem para o pacote* `lp_solve` *o problema linear* **P**

```
g.rhs <- c(2, 6);
g.con <- matrix (c(1, 2, 4, 2), nrow=2, byrow=TRUE)
g.dir <- c("<=", "<=")
g.obj <- c(2,6)
```

(b) *(V)[ ](F)[ ] Os comandos seguintes do* `R` *traduzem para o pacote* `lp_solve` *o problema linear* **P**

```
g.rhs <- c(-2, 6);
```

```
g.con <- matrix (c(1, 2, 4, 2), nrow=2, byrow=TRUE)
g.dir <- c("<=", "<=")
g.obj <- c(2,6)
```

(c) *(V)[ ](F)[ ] Os comandos seguintes do* `R` *traduzem para o pacote* `lp_solve` *o problema linear* **P**

```
g.rhs <- c(2, 6);
g.con <- matrix (c(1, 2, 4, 2), nrow=2, byrow=TRUE)
g.dir <- c("<=", "<=")
g.obj <- c(-2,-6)
```

(d) *(V)[ ](F)[ ] Os comandos seguintes do* `R` *traduzem para o pacote* `lp_solve` *o problema linear* **P**

```
g.rhs <- c(-2, -6);
g.con <- matrix (c(1, 2, 4, 2), nrow=2, byrow=TRUE)
g.dir <- c("<=", "<=")
g.obj <- c(-2,-6)
```

(e) *(V)[ ](F)[ ] Os comandos seguintes do* `R` *traduzem para o pacote* `lp_solve` *o problema linear* **P**

```
g.rhs <- c(2, 6);
g.con <- matrix (c(1, 2, 4, 2), nrow=4, byrow=TRUE)
g.dir <- c("<=", "<=")
g.obj <- c(2,6)
```

5. *Resolvendo um problema com* `R`

É preciso incluir o pacote `lpSolve` *com o comando*

```
library(lpSolve)
```

do contrário `R` *não irá conhecer as funções de* programação linear*.*

Observação: *este era um erro que havia em edição anterior desta lista. Mas quem tivesse tentado a solução já deveria ter-se apercebido do erro. Corrigido 19:20 de 12 de Novembro...*

Considere o problema **P** *definido pela função custo da equação (7) e pelas restrições na equação (4).*

(a) *(V)[ ](F)[ ] O comando*
`lp ("min", f.obj, f.con, f.dir, f.rhs)` *resolve o problema* **P** *procurando um mínimo para a função custo que ocorre no ponto* $(2, 0) \in \Omega$.

(b) *(V)[ ](F)[ ]O comando*
`lp ("min", f.obj, f.con, f.dir, f.rhs)` *resolve o problema* **P** *procurando um máximo para a função custo que ocorre no ponto* $(0, 0) \in \Omega$.

(c) <u>(V)[ ](F)[ ]</u> *O comando*
`lp ("max", f.obj, f.con, f.dir, f.rhs)` *resolve o problema* **P** *procurando um máximo para a função custo que ocorre no ponto* $(0,0) \in \Omega$.

(d) <u>(V)[ ](F)[ ]</u> *O comando*
`lp ("max", f.obj, f.con, f.dir, f.rhs)` *resolve o problema* **P** *procurando um máximo para a função custo que ocorre no ponto* $(0, \frac{5}{2}) \in \Omega$.

(e) <u>(V)[ ](F)[ ]</u> *O comando*
`lp ("min", f.obj, f.con, f.dir, f.rhs)` *resolve o problema* **P** *procurando um máximo para a função custo que ocorre no ponto* $(0,0) \in \Omega$.

6. <u>*Dimensão três*</u>

Considere o problema linear **R** *definido pela função de custo* F *e o sistema de restrições:*

$$F(x,y,z) = x + 9y + z \tag{9}$$

$$\begin{cases} -x - 2y - 3z & > -9 \\ 3x + 2y + 2z \leq 15 \end{cases} \tag{10}$$

(a) <u>(V)[ ](F)[ ]</u> *O sistema de restrições sempre pode obtido de forma equivalente com dados positivos (normatização).*

(b) <u>(V)[ ](F)[ ]</u> *Normatizando os dados na equação (0) obtemos a forma equivalente do problema linear* **R**

$$F(x,y,z) = x + 9y + z \tag{11}$$

$$\begin{cases} x + 2y + 3z & \leq 9 \\ 3x + 2y + 2z \leq 15 \end{cases} \tag{12}$$

(c) <u>(V)[ ](F)[ ]</u> *O problema* **R** *subentende um conjunto convexo* Ω limitado, *mas que é um conjunto convexo do* $\mathbf{R}^3$

(d) <u>(V)[ ](F)[ ]</u> *O problema* **R** *subentende um conjunto convexo* Ω ilimitado *que tem dois planos como fronteira. Um dos planos tem por vetor perpendicular,* $(1,2,3)$ *e outro tem por vetor perpendicular* $(3,2,2)$.

(e) <u>(V)[ ](F)[ ]</u> *Uma projeção do conjunto convexo* Ω *sobre o plano* $\mathbf{R}^2$ *se obtém considerando* $z = 0$ *no sistema de restrições na equação (11).*

7. <u>*Álgebra Linear*</u>

Considere o problema **R** *definido no sistema de equações (11).*

(a) <u>(V)[ ](F)[ ]</u> *O sistema de comandos do* `R` *são a forma de traduzir problema* **R** *para a sintaxe de* `R`*:*

```
F.obj <- c(1, 9, 3)
F.con <- matrix (c(1, 2, 3, 3, 2, 2), nrow=2, byrow=TRUE)
F.dir <- c("<=", "<=")
f.rhs <- c(9, 15)
```

(b) *(V)[ ](F)[ ] O sistema de comandos do* `R` *são a forma de traduzir problema* **R** *para a sintaxe de* `R`*:*

```
F.obj <- c(1, 9, 3)
F.con <- matrix (c(-1, -2, -3, 3, 2, 2), nrow=2, byrow=TRUE)
F.dir <- c(">", "<=")
f.rhs <- c(-9, 15)
```

mas não é a forma normatizada *e deve ser evitado.*

(c) *(V)[ ](F)[ ] Os comandos*

```
print(F.obj)
print(F.con)
print(F.rhs)
```

fazem com que `R` *apresente as matrizes dos dados do problema.*

(d) *(V)[ ](F)[ ] Usando o comando* `lp` *do pacote* `lp_solve`[1]*, descubra como usando* `help(lp)` *depois de* `library(lpSolve)`*, é possível encontrar uma solução* máximal *para o problema* **R**.

(e) *(V)[ ](F)[ ] Usando o comando* `lp` *do pacote* `lp_solve`[2]*, descubra como usando* `help(lp)` *depois de* `library(lpSolve)`*, é possível encontrar uma solução* minimal *para o problema* **R**.

[1] é preciso incluir este pacote com o comando library(lpSolve).
[2] é preciso incluir este pacote com o comando library(lpSolve).