Solver do R

Profa. Luciane Alcoforado/AFA

05 de abril de 2022

RESOLVENDO PROBLEMAS DE PROGRAMAÇÃO LINEAR NO R

- lp_solve que é disponibilizado no R por meio dos pacotes lpSolve e lpSolveAPI;
- GLPK que no R pode ser usado via pacote Rglpk;
- SYMPHONY que está disponível no R via pacote Rsymphony

Alguns destes pacotes têm funções que permitem a leitura de arquivos contendo problemas de programação linear, programação linear inteira e programação linear inteira mista escritos no formato CPLEX, por exemplo.

O problema

Uma marcenaria deseja estabelecer uma programação diária de produção. Na modelagem serão considerados apenas dois recursos: madeira e mão-de-obra. A marcenaria produz apenas mesas e armários.

A tabela abaixo resume a relação entre recursos e produtos. Para produzir uma mesa utiliza-se 2 m^2 de madeira e 2 H.h de mão-de-obra, cada mesa vendida retorna um lucro de $4; para produzir armário utiliza-se 3 m^2 de madeira e 1 H.h de mão-de-obra, cada armário vendido retorna um lucro de $1. A disponibilidade diária de madeira e mão-de-obra é de 12 m^2 e 8 H.h respectivamente.

Recurso	Mesa	Armario	Disponibilidade
Madeira	2 m2	3 m2	12 m2
Mão-de-obra	2 H.h	1 H.h	8 H.h
Lucro	$4	$1	-

Montagem do Modelo

Definir as variáveis de decisão do problema, ou seja, quanto deve ser produzido por dia de mesas e armários?

- x_1 = quantidade a produzir de mesas
- x_2 = quantidade a produzir de armários

O modelo matemático:

$maxz = 4x1 + 1x2$

sujeito a:

$2x1 + 3x2 \leq 12$ (R1)

$2x1 + 1x2 \leq 8$ (R2)

$x1, x2 \geq 0$

Resolvendo problemas de programaçaõ linear usando o pacote `lpSolve`:

```
##########################
####     PACOTES     #####
##########################

# Instalar pacotes necessários
#install.packages(c("lpSolve","lpSolveAPI"))

# Carregar pacotes lpSolve
suppressMessages(require(lpSolve))

##########################
####     PROBLEMA    #####
##########################

# coeficientes na função objetivo
func.objetivo <- c(4 , 1)

# coeficientes nas restrições.
R1=c(2, 3)
R2=c(2, 1)


coeficientes.restricoes <- rbind(R1, R2 )

# sinal das restrições. Deve obedecer a ordem da matriz de coeficientes
direcao.restricoes <- c("<=","<=")

# limite das restrições. Deve obedecer a ordem da matriz de coeficientes
limites.restricoes <- c(12,8)

##########################
####     SOLUÇÃO     #####
##########################

# Basicamente, usamos a função lp do pacote lpSolve com os seguintes parâmetros:
# direction: que recebe max ou min dependendo se o problema é de maximização ou minização, respectivame
# objective.in: que recebe o nome do vetor com parâmetros da função objetivo
# const.mat: que recebe o nome da matriz com coeficientes das restrições
# const.rhs: que recebe o nome do vetor com os limites das restrições
# mais opções da função podem ser obtidas por meio do help(lp) como por exemplo all.int

solucao.problema <- lpSolve::lp(direction = "max",
                               objective.in = func.objetivo,
                               const.mat = coeficientes.restricoes,
                               const.dir = direcao.restricoes,
                               const.rhs = limites.restricoes,

  all.int=F,
  compute.sens = 1)

##########################
```

```
####    RESULTADO    #####
##########################

# valor da função objetivo na solução
solucao.problema$objval
```

```
## [1] 16
```

```
# Valores para as variáveis de escolha que geram máximo ou mínimo dependendo do problema
solucao.problema$solution
```

```
## [1] 4 0
```

Conclusão: A solução indica que para obter o máximo lucro deve ser produzido $x_1 = 4$ mesas e $x_2 = 0$ armários, obtendo um lucro de $16.

Perguntas Adicionais para você pensar e resolver

Há sobra de recursos?

Se forçarmos a produção de pelo menos um armário por dia o que acontece com o lucro?

E com a sobra de recursos?

Se aumentarmos o recurso mais utilizado em 1 unidade, o que acontece com o lucro?

Incluindo nova restrição

Vamos forçar a produção diária de pelo menos 1 armário, ou seja, vamos acrescentar uma terceira restrição ao problema:

$1x2 \geq 1$ (R3)

O que acontece com a solução ótima?

Tente resolver esse problema e chegar a uma conclusão.

Colocando o problema na forma padrão

Devemos adicionar variáveis de folga/excesso às restrições do tipo "<=" ou ">=" para torná-las igualdade.

As variáveis de decisão são:

- x_1 = quantidade a produzir de mesas
- x_2 = quantidade a produzir de armários

O modelo matemático:

$maxz = 4x1 + 1x2$

sujeito a:

$2x1 + 3x2 \leq 12$ (R1)

$2x1 + 1x2 \leq 8$ (R2)

$x1, x2 \geq 0$

As variáveis de folga são:

- x_3 = folga do recurso madeira
- x_4 = folga do recurso mão-de-obra

O modelo matemático na forma padrão

$maxz = 4x1 + 1x2 + 0x3 + 0x4$

sujeito a:

$2x1 + 3x2 + 1x3 + 0x4 = 12$ (R1)

$2x1 + 1x2 + 0x3 + 1x4 = 8$ (R2)

$x1, x2, x3, x4 \geq 0$

Resolvendo o problema no R

```r
# coeficientes na função objetivo
func.objetivo <- c(4 , 1, 0, 0)

# coeficientes nas restrições.
R1=c(2, 3, 1, 0)
R2=c(2, 1, 0, 1)
coeficientes.restricoes <- rbind(R1, R2 )

# sinal das restrições.
direcao.restricoes <- c("=","=")

# limite das restrições.
limites.restricoes <- c(12,8)


solucao.problema <- lpSolve::lp(direction = "max",      objective.in = func.objetivo,
                                const.mat = coeficientes.restricoes,
                                const.dir = direcao.restricoes,
                                const.rhs = limites.restricoes,
all.int=F,  compute.sens = 1)

# valor da função objetivo na solução
solucao.problema$objval
```

```
## [1] 16
```

```r
# Valores para as variáveis de escolha que geram máximo ou mínimo dependendo do problema
solucao.problema$solution
```

```
## [1] 4 0 4 0
```

Ao introduzir as variáveis de folga ao problema, temos nelas armazenado as folgas das restrições de recursos. Por exemplo, vemos que há uma folga de 4 m^2 de madeira pois $x3 = 4$ refere-se a folga da restrição 1 (quantidade diária de madeira disponível).

Resposta das perguntas adicionais