Assignment 1 - Linear Programming

Contents

The problem

A trading company is looking for a way to maximize profit per transportation of their goods. The company has a train available with 3 wagons.

When stocking the wagons they can choose among 4 types of cargo, each with its own specifications. How much of each cargo type should be loaded on which wagon in order to maximize profit?

More data

TRAIN WAGON j	WEIGHT CAPACITY (TONNE) w_j	VOLUME CAPACITY (m^2) s_j
(wag) 1	10	5000
(wag) 2	8	4000
(wag) 3	12	8000

CARGO TYPE i	AVAILABLE (TONNE) a_i	VOLUME (m^2/t) v_i	PROFIT (PER TONNE) p_i
(cg) 1	18	400	2000
(cg) 2	10	300	2500
(cg) 3	5	200	5000
(cg) 4	20	500	3500

The decision variables

The variables in question are the fraction of each one of the three wagons to be allocated to each of the types of cargos

- x_{i1} = amount of cargo $i, \forall i \in \{1,2,3,4\}$ allocated to wagon 1
- x_{i2} = amount of cargo $i, \forall i \in \{1,2,3,4\}$ allocated to wagon 2
- x_{i3} = amount of cargo $i, \forall i \in \{1,2,3,4\}$ allocated to wagon 3

or in brief:

x_{ij} = amount of cargo i allocated to wagon j, $\forall i \in \{1,2,3,4\}$ and $\forall j \in \{1,2,3\}$

The objective function

The *objective function* is the function that we seek to maximize, that is the profit, P:

$$P = 2000x_{11}+2500x_{21}+5000x_{31}+3500x_{41}+2000x_{12}+2500x_{22}+5000x_{32}+3500x_{42}+2000+x_{13}+2500x_{23}+5000x_{33}+3500x_{43}$$

---

or in brief:

$$P = \sum_i \sum_j p_i x_{ij}$$

The constraints

The following constraints have to be taken in consideration;

- Weight capacity per train wagon, that is: $\sum_i x_{ij} \leq w_j, \quad j = \{1, \ldots, 3\}$
- Volume capacity per train wagon, that is: $\sum_i v_i x_{ij} \leq s_j, \quad j = \{1, \ldots, 3\}$
- Limited availability per cargo type, that is: $\sum_j x_{ij} \leq a_i, \quad i = \{1, \ldots, 4\}$

Building the model

The resulting linear programming model is:

$$\begin{aligned} \text{maximize} \quad & P(x) = \sum_i \sum_j p_i x_{ij} \\ \text{s.t.} \quad & \sum_i x_{ij} \leq w_j, \quad j = \{1, \ldots, 3\} \\ & \sum_i v_i x_{ij} \leq s_j, \quad j = \{1, \ldots, 3\} \\ & \sum_j x_{ij} \leq a_i, \quad i = \{1, \ldots, 4\} \\ & x_{ij} \geq 0, \quad j = 1, \ldots, 3, i = 1, \ldots, 4 \end{aligned}$$

***

That is:

Let's load the necessary libraries and define the datasets to be used.

```
library(lpSolveAPI)
model <- make.lp(0,12)
```

Adding constraints by row

```
row.add.mode(model,"on")
```

```r
add.constraint(model,
               xt=c(1,1,1,1),
               type="<=",rhs=10,
               indices=c(1:4))   # Constraint 1
add.constraint(model,
               xt=c(1,1,1,1),
               type="<=",rhs=8,
               indices=c(5:8))   # Constraint 2
add.constraint(model,
               xt=c(1,1,1,1),
               type="<=",rhs=12,
               indices=c(9:12))   # Constraint 3
```

```r
add.constraint(model,
               xt=c(400,300,200,500),
               type="<=",rhs=5000,
               indices=c(1:4))   # Constraint 5
add.constraint(model,
               xt=c(400,300,200,500),
               type="<=",rhs=4000,
               indices=c(5:8))   # Constraint 6
add.constraint(model,
               xt=c(400,300,200,500),
               type="<=",rhs=8000,
               indices=c(9:12))   # Constraint 7
```

```r
add.constraint(model,
               xt=c(1,1,1),
               type="<=",rhs=18,
               indices=c(1,5,9))   # Constraint 8
add.constraint(model,
               xt=c(1,1,1),
               type="<=",rhs=10,
               indices=c(2,6,10))   # Constraint 9
add.constraint(model,
               xt=c(1,1,1),
               type="<=",rhs=5,
               indices=c(3,7,11))   # Constraint 10
add.constraint(model,
               xt=c(1,1,1),
               type="<=",rhs=20,
               indices=c(4,8,12))   # Constraint 11

row.add.mode(model,"off")
```

```r
#set objective coefficients
c <-rep(c(2000,2500,5000,3500),3)
print(c)
```

```
##  [1] 2000 2500 5000 3500 2000 2500 5000 3500 2000 2500 5000 3500
```

```r
set.objfn(model, c)
```

```r
#set objective direction
lp.control(model,sense='max')
```

Solving the problem

```r
#solve the model, if this return 0 an optimal solution is found
sol<-solve(model)

#this return the proposed solution
get.objective(model)
```

```
## [1] 107500
```

```r
get.variables(model)
```

```
##  [1]  0  5  5  0  0  0  0  8  0  0  0 12
```

```r
get.objective(model)
```

```
## [1] 107500
```

The basic variables are:

```r
get.basis(model, nonbasic = F)
```

```
##  [1] -12 -18 -22  -4  -5  -6  -7 -16 -13  -8
```

Sensitivity analysis

Let's use some functions seen during the exercise sessions to get a pretty-print of the sensitivity report:

```r
printSensitivityObj(model) # get pretty-printed sensitivity analysis for the obj func.
```

```
##    Objs          Sensitivity
## 1    C1   -inf <= C1 <= 2500
## 2    C2   2500 <= C2 <= 2500
## 3    C3    5000 <= C3 <= inf
## 4    C4   -inf <= C4 <= 3500
## 5    C5   -inf <= C5 <= 2500
## 6    C6   -inf <= C6 <= 2500
## 7    C7   -inf <= C7 <= 5000
## 8    C8   3500 <= C8 <= 3500
## 9    C9   -inf <= C9 <= 2500
## 10  C10 -inf <= C10 <= 2500
## 11  C11 -inf <= C11 <= 5000
## 12  C12  3500 <= C12 <= inf
```

Perform the sensitivity analysis for the model solved.

```r
 printSensitivityRHS(model)# get pretty-printed sensitivity analysis for the rhs
```

```
##    Rhs        Sensitivity
## 1   B1       5 <= B1 <= 15
## 2   B2        8 <= B2 <= 8
## 3   B3      12 <= B3 <= 16
## 4   B4   -inf <= B4 <= inf
## 5   B5   -inf <= B5 <= inf
## 6   B6   -inf <= B6 <= inf
## 7   B7  -inf <= B7 <= inf
## 8   B8  -inf <= B8 <= inf
## 9   B9      0 <= B9 <= 10
## 10 B10    15 <= B10 <= 20
## 11 B11     -5 <= B11 <= 5
## 12 B12 -inf <= B12 <= inf
## 13 B13 -inf <= B13 <= inf
## 14 B14      0 <= B14 <= 5
## 15 B15     -5 <= B15 <= 0
## 16 B16 -inf <= B16 <= inf
## 17 B17      0 <= B17 <= 0
## 18 B18 -inf <= B18 <= inf
## 19 B19     -5 <= B19 <= 0
## 20 B20     -8 <= B20 <= 0
## 21 B21     -5 <= B21 <= 0
## 22 B22 -inf <= B22 <= inf
```

while the shadow prices are given by the dual variables:

```r
get.dual.solution(model)
```

```
##  [1]    1 2500 2500 2500    0    0    0    0    0 2500 1000 -500    0    0    0
## [16] -500    0    0    0 -500    0    0    0
```

Let's check that the shadow price is for real the increment of the objective function if we increase the rhs by one unity. For instance, what happend if we update the rhs value of the first constraint?

```r
set.constr.value(model, constraints = 1, rhs = 11)
solve(model)
```

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

```r
get.objective(model)
```

```
## [1] 110000
```

```r
#the objective function should be
107500 + 2500
```

```
## [1] 110000
```

Indeed the objective function value has changed by an amount equal to the first dual variable, that is 2500.

Check that the basic solusion hasn't changed

```r
get.basis(model, nonbasic = F)
```

```
##  [1] -12 -18 -22  -4  -5  -6  -7 -16 -13  -8
```

Questions about LP

1. Can an LP model have more than one optimal solution. Is it possible for an LP model to have exactly two optimal solutions? Why or why not?

ans. An LP model can have more than one optimal solution, it can happen when the level curve of the objective function is parallel to the active constraint. In this case all the points belonging to the segments between the two optimal corners (basic solutions) are also optimal. Therefore, it is impossible for an LP problem to have exactly two optimal solutions.

Recall the fundamental theorem of Linear Programming: the maxima and minima of a linear function over a convex polygonal region occur at the region's corners. Further, if an extreme value occurs at two corners, then it must also occur everywhere on the line segment between them.

2. Are the following objective functions for an LP model equivalent? That is, if they are both used, one at a time, to solve a problem with exactly the same constraints, will the optimal values for x_1 and x_2 be the same in both cases? Why or why not?

$$\max 2x_1 + 3x_2$$

$$\min -2x_1 - 3x_2$$

ans. Yes the optimal values for X1 and X2 will be the same because max $f(x) = -min - f(x)$

3. Which of the following constraints are not linear or cannot be included as a constraint in a linear programming problem?

a.
$$2x_1 + x_2 - 3x_3 \geq 50 \rightarrow \textit{linear}$$

b.
$$2x_1 + \sqrt{x_2} \geq 60 \rightarrow \textit{not linear}$$

c.
$$4x_1 - \frac{1}{2}x_2 = 75 \rightarrow \textit{linear}$$

d.
$$\frac{3x_1 + 2x_2x_1 - 3x_3}{x_1 + x_2 + x_3} \leq 0.9 \rightarrow \textit{not linear}$$

e.
$$3x_1^2 + 7x_2 \leq 45 \rightarrow \textit{not linear}$$