Linear Programming Assignment

Cesareo Giacomo 805716

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 i	WEIGHT CAPACITY (TONNE) w_i	VOLUME CAPACITY (m^2) s_i
(wag) 1	10	5000
(wag) 2	8	4000
(wag) 3	12	8000

CARGO TYPE j	AVAILABLE (TONNE) a_j	VOLUME (m^2) v_j	PROFIT (PER TONNE) p_j
(cg) 1	18	400	2000
(cg) 2	10	300	2500
(cg) 3	5	200	5000
(cg) 4	20	500	3500

The decision variables

The first step about solving a linear programming optimization problem is defining the decision variables.
In this exercise the decision variables are defined as $C_{i,j}$ where i representing the train wagon number ($i = 1, 2, 3$) and j depicting the cargo type ($j = 1, 2, 3, 4$).
For example:

- $C_{1,1}$ Indicates the quantity in tonnes of the first cargo type on the first wagon.
- $C_{1,2}$ Indicates the quantity in tonnes of the first cargo type on the second wagon .
- And so on up to the fourth cargo type quantity loaded on the third wagon represented as $C_{3,4}$

The objective function

The goal of this assignment is all about maximazing the total profit of the company by carefully choosing how each wagon should be loaded with a given quantity of each type of cargo. The objective function can be computed by multiplying each value of the decision variables $C_{i,j}$ by the PROFIT (PER TONNE) value.
The objective function is presented below:

$$\begin{aligned} Max(2000C_{1,1} + 2500C_{1,2} + 5000C_{1,3} + 3500C_{1,4} + \\ +2000C_{2,1} + 2500C_{2,2} + 5000C_{2,3} + 3500C_{2,4} + \\ +2000C_{3,1} + 2500C_{3,2} + 5000C_{3,3} + 3500C_{3,4}) \end{aligned}$$

The constraints

The first constraints depicted below are those about the total volume capacity of each wagon, indeed these can't carry more than their maximum capacity: the volume of each cargo type (for example $400m^2$ for the first one) is multiplied by the decision variables $C_{i,j}$ and can't exceed the maximum amount of each wagon (for example $5000m^2$ for the first one).

$$\begin{aligned} 400c_{1,1} + 300c_{1,2} + 200c_{1,3} + 500c_{1,4} \leq 5000 \\ 400c_{2,1} + 300c_{2,2} + 200c_{2,3} + 500c_{2,4} \leq 4000 \\ 400c_{3,1} + 300c_{3,2} + 200c_{3,3} + 500c_{3,4} \leq 8000 \end{aligned}$$

Same thing, as seen above, for the maximum weight capacity for each wagon: the sum of every quantity of $C_{i,j}$ can not exceed the maximum amount for each wagon.
For example the first constrain of those depicted below represent the total amount of tonnes, choosen between each cargo type, selected for the first wagon.

$$\begin{aligned} C_{1,1} + c_{1,2} + C_{1,3} + C_{1,4} \leq 10 \\ C_{2,1} + c_{2,2} + C_{2,3} + C_{2,4} \leq 8 \\ C_{3,1} + c_{3,2} + C_{3,3} + C_{3,4} \leq 12 \end{aligned}$$

The third constrains set represent the maximum number of available cargo type expressed in tonnes. For example there is a maximum of 18 tonnes available for the first cargo type.

$$\begin{aligned} C_{1,1} + C_{2,1} + C_{3,1} \leq 18 \\ C_{1,2} + C_{2,2} + C_{3,2} \leq 10 \\ C_{1,3} + C_{2,3} + C_{3,3} \leq 5 \\ C_{1,4} + C_{2,4} + C_{3,4} \leq 20 \end{aligned}$$

Last constrain set is all about non negativity of the decision variables.

$$\begin{aligned} C_{i,j} \geq 0 \\ i = 1, 2, 3 \\ j = 1, 2, 3, 4 \end{aligned}$$

Building the model

```r
library(tidyr)
library(dplyr)
library(lpSolve)
library(lpSolveAPI)
```

```r
model <- make.lp(0,12) # Initialize the model with 0 constrains and 12 decision variables.
lp.control(model, sense = 'max')
```

```r
# Objective function's coefficients (profit per tonne)
set.objfn(model, c(2000,2500,5000,3500,2000,2500,5000,3500,2000,2500,5000,3500))
```

```r
# Volume Constrains.
add.constraint(model, c(400, 300, 200, 500), "<=", 5000 ,1:4)
add.constraint(model, c(400, 300, 200, 500), "<=", 4000 ,5:8)
add.constraint(model, c(400, 300, 200, 500), "<=", 8000 ,9:12)

# Weight constrains.
add.constraint(model, c(1, 1, 1, 1), "<=", 10, 1:4)
add.constraint(model, c(1, 1, 1, 1), "<=", 8, 5:8)
add.constraint(model, c(1, 1, 1, 1), "<=", 12, 9:12)

# Available tonne for each cargo type constrains.
add.constraint(model, c(1,1,1), "<=", 18, c(1,5,9))
add.constraint(model, c(1,1,1), "<=", 10, c(2,6,10))
add.constraint(model, c(1,1,1), "<=", 5, c(3,7,11))
add.constraint(model, c(1,1,1), "<=", 20, c(4,8,12))

# Non negativity constrains.
set.bounds(model, lower = c(0, 0, 0, 0,0, 0, 0, 0,0, 0, 0, 0))

solve(model)
```

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

Solve(model) returned 0 meaning that the model has been solved correctly.

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

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

The results about maximization can be interpreted as follow: the first four digits show the quantity of each cargo type choosen for the first wagon, then the amount of cargo type selected for the second wagon are shown in the next four digits. To make things more clear:

- First wagon: 5 units of the second and 5 units of the third cargo type.
- Second wagon: 8 units of the fourth cargo type.
- Third wagon: 12 units of the fourth cargo type.

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

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

This results represent the maximazed profit.

Sensitivity analysis

The following code return a vector of 23 elements: the first one is the result of the optimization achieved by the model, the next 10 digits, one for every constrain, represent the value of the left hand side, for example: 2500 depict that over $5000m^2$ about the first constrain the model selected $2500m^2$, so in this case it will be a slack variable $S_1 = 2500$ representing the difference between the right hand side of the first constrain and this value. The values [10, 8, 12] are those about the second set of constrain and they are completely selected by the model, so the slack variables [S_4, S_5, S_6] will be equal to zero. The last 12 digits of the array represent the solution seen above.

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

```
##  [1] 107500   2500   4000   6000     10      8     12      0      5      5
## [11]     20      0      5      5      0      0      0      0      8      0
## [21]      0      0     12
```

The following function return an array depicting the indices about the variables of the optimal solution, it is important to notice that the first ten elements (from -1 to -10) are the slack variables. So the optimal solution for the model is the one with those variables: [S_1, S_2, S_3, S_7, S_8, $C_{1,2}$, $C_{1,3}$, $C_{2,2}$, $C_{2,4}$, $C_{3,4}$]

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

```
##  [1]  -1  -2  -3 -12 -18 -22  -7 -16 -13  -8
```

It is important now to analyze the shadow prices: with the following command is returned an array representing the shadow prices for each constrain. For example the last element shows that an increment of one unit in the right hand side of the last constrain build up the optimazed profit of 1000. It's important to notice that a shadow price equal to zero means that the constrain associated with this one is not binding for the optimization problem.

```r
get.dual.solution(model)[2:11]
```

```
##  [1]    0    0    0 2500 2500 2500    0    0 2500 1000
```

Finally it is usefull to analyze the sensitivity for both the right hand side and the objective function. Thanks to the folowing two functions presented during class exercises we can inspect the sensitivity coefficients.

Here are presented the values for the right hand side: only the first ten elements are shown because only those explain the values for the ten constrains of the model. For example the RHS of the first constrain can take values between $-inf$ and $+inf$ meaning that even a really big change into the RHS will not change the optimization's result. It is important to mention that, as presented before, if the shadow price is equal to zero, then the constrain is not binding, it is reasonable to think that every change made in the RHS will not produce any change to the optimal solution, indeed if the shadow price is equal to zero the sensitivity is expected to be, for the i_{th} constrain, $-inf \leq B_i \leq +inf$

```r
printSensitivityRHS(model)
```

```
##    RHS   Sensitivity
## 1   B1 -inf<=B1<=inf
## 2   B2 -inf<=B2<=inf
## 3   B3 -inf<=B3<=inf
## 4   B4      5<=B4<=15
## 5   B5       8<=B5<=8
## 6   B6     12<=B6<=16
## 7   B7 -inf<=B7<=inf
## 8   B8 -inf<=B8<=inf
## 9   B9      0<=B9<=10
## 10 B10   15<=B10<=20
```

The same conclusion can be made about the coefficients of the objective function: for example the first element can take value $-inf \leq C_{1,1} \leq 2500$ so that the optimal solution does not change. The second coefficient must be equal to 2500 and even the smallest change made in his coefficient will bring to another optimal solution.

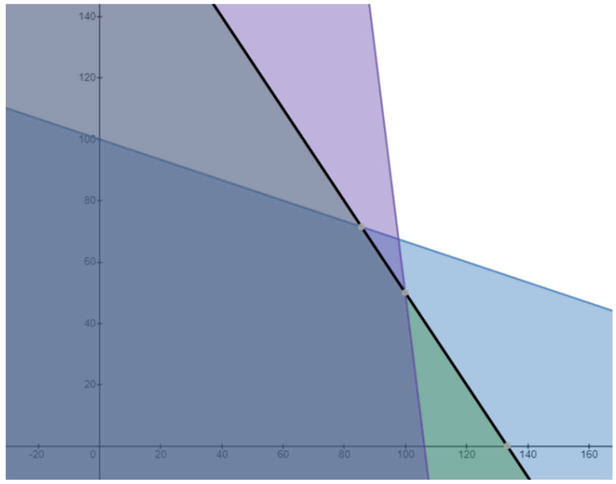
```r
printSensitivityOBJ(model)
```

```
##    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
```

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?

Linear Programming model can have just one or more than two optimal solution, for example: considering a maximization problem (presented graphically in the immage below) with two $\leq$ constraints, if the objective function (the black line) has the same slope of the immaginary line that intersect the two vertices (two possible optimal solutions), then not only the two points are considered optimal but also all the points between those two are considered valid optimal solutions.



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?

$$max(2x_1 + 3x_2)$$
$$min(-2x_1 - 3x_2)$$

The above objective functions represent the same line in a bi-dimensional space, then values for the variables x_1 and x_2 remain the same for both models. The only thing that will take different value is the output of the objective function: the maximization problem will have a positive value as solution and the minimization problem a negative value, that's because the result must balance the signs of the objective function coefficients. The optimized values will be one opposite of the other, for example 100 and -100.

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

a. $2x_1 + x_2 - 3x_3 \geq 50$ it represent a plane so can be included in a linear programming problem.

b. $2x_1 + \sqrt{x_2} \geq 60$ Square root is non linear then we can not use it in a linear programming model.

c. $4x_1 - \frac{1}{2}x_2 = 75$ The function represent a line, it is valid in LP model and will be a binding constrain because of the equality.

d. $\frac{3x_1+2x_2x-3x_3}{x_1+x_2+x_3} \leq 0.9$ If x multiplied by x_2 is a number and not a variable a fraction between two planes is linear, else, if x in an unknown variable, the function is not linear.

e. $3x_1^2 + 7x_2 \leq 45$ Squared variables are not linear, then we can not use this function.