Assignment 1

Cavenati Laura matricola: 864000

27 marzo 2020

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?

The decision variables

The variables in question are the number of tonnes of each cargo type loaded on each of 3 wagons, giving us a total of 12 variables to track.

x_1= # tonnes of cargo type 1 loaded on the wagon 1
x_2= # tonnes of cargo type 2 loaded on the wagon 1
x_3= # tonnes of cargo type 3 loaded on the wagon 1
x_4= # tonnes of cargo type 4 loaded on the wagon 1
x_5= # tonnes of cargo type 1 loaded on the wagon 2
x_6= # tonnes of cargo type 2 loaded on the wagon 2
x_7= # tonnes of cargo type 3 loaded on the wagon 2
x_8= # tonnes of cargo type 4 loaded on the wagon 2
x_9= # tonnes of cargo type 1 loaded on the wagon 3
x_{10}= # tonnes of cargo type 2 loaded on the wagon 3
x_{11}= # tonnes of cargo type 3 loaded on the wagon 3
x_{12}= # tonnes of cargo type 4 loaded on the wagon 3

The objective function

In this case, we want to maximize the profit, P, which is in turn a function of the twelve defined variables, and their per tonne profits:

$$P = (x_1 + x_5 + x_9) * 2000 + (x_2 + x_6 + x_{10}) * 2500 + (x_3 + x_7 + x_{11}) * 5000 + (x_4 + x_8 + x_{12}) * 3500$$

The constraints

- The sum of volumes of the four cargo types of each train wagon must be less than or equal the volume capacity of the train wagon:

  $400 * x_1 + 300 * x_2 + 200 * x_3 + 500 * x_4 \leq 5000$
  $400 * x_5 + 300 * x_6 + 200 * x_7 + 500 * x_8 \leq 4000$
  $400 * x_9 + 300 * x_{10} + 200 * x_{11} + 500 * x_{12} \leq 8000$

- The sum of tonnes of each cargo type must be less than or equal the available tonnes of each cargo type:

  $x_1 + x_5 + x_9 \leq 18$
  $x_2 + x_6 + x_{10} \leq 10$

$x_3 + x_7 + x_{11} \leq 5$
$x_4 + x_8 + x_{12} \leq 20$

- The sum of tonnes of the four cargo type of each train wagon must be less than or equal the weight capacity of the train wagon:

  $x_1 + x_2 + x_3 + x_4 \leq 10$
  $x_5 + x_6 + x_7 + x_8 \leq 8$
  $x_9 + x_{10} + x_{11} + x_{12} \leq 12$

The Boundaries

x_i, with $i = 1, ..., 12$ must be greater than or equal to zero.

So, now that we know all the relevant functions, we can implement this using R and solving it by means of a linear programming solver.

Building the model using lpSolveAPI

Initializing the Linear Program

The libraries that we will use are:

```
library(lpSolveAPI)
library(dplyr)
library(tidyr)
```

Creating a model with 12 variables and 0 constraints for the original maximization problem:

```
model = make.lp(0,12)
lp.control(model, sense="max")
```

Defining the Objective Function:

```
set.objfn(model,obj=c(2000,2500,5000,3500,2000,2500,5000,3500,2000,2500,5000,3500))
```

Defining the Constraints:

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

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

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

Defining the Boundaries:

```
set.bounds(model,lower=c(0,0,0,0,0,0,0,0,0,0,0,0))
```

Solving and Viewing Results

```
solve(model)
```

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

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

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

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

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

The optimal value for the decision variables is $[0, 5, 5, 0, 0, 0, 0, 8, 0, 0, 0, 12]$.
The optimal value of the objective function, that is, the total profit is 107500.
Sensitivity analysis

```r
get.constraints(model)
```

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

Notice that the rhs values are $b^T = [5000, 4000, 8000, 18, 10, 5, 20, 10, 8, 12]$. So, only six constraints are met to equality. This means, for example, that there is still space on wagon 1 ($2500m^2$ occupied out of $5000m^2$ available), but the maximum weight has been reached (10 tonnes).

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

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

The array $[0, 0, 0, 0, 0, 2500, 1000, 2500, 2500, 2500]$ represents the shadow prices for the dual variables (that are ten because ten are the constraints in the primal problem). Notice that the first five shadow prices are zero meaning that a small change in the volumes capacity of each train wagon does not affect the solution. Also a small change in the tonnes available of cargo type 1 and 2 does not affect the solution.

Definining sensitivity bounds for the rhs and the objective function, using the functions seen in class:

```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
```

```r
printSensitivityRHS(model)
```

```
##    Rhs                                 Sensitivity
## 1   B1                          -inf <= B1 <= inf
## 2   B2                          -inf <= B2 <= inf
## 3   B3                          -inf <= B3 <= inf
## 4   B4                          -inf <= B4 <= inf
## 5   B5                          -inf <= B5 <= inf
## 6   B6 0.00000000000000177635683940025 <= B6 <= 10
## 7   B7                             15 <= B7 <= 20
## 8   B8                              5 <= B8 <= 15
```

```
## 9   B9                                8 <= B9 <= 8
## 10 B10                              12 <= B10 <= 16
## 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
```

Summary of sensitivity analysis for the changes in constraint coefficients:

```r
a<-get.sensitivity.rhs(model)
b<-get.sensitivity.obj(model)
library(knitr)
name<-c("volume wagon 1","volume wagon 2","volume wagon 3",
       "tonnes cargo 1","tonnes cargo 2","tonnes cargo 3","tonnes cargo 4",
       "weight wagon 1","weight wagon 2","weight wagon 3")
final.value<-get.constraints(model)
constraintRHS<-c(5000,4000,8000, 18,10,5,20,10,8,12)
bound.down<-round(a$dualsfrom[1:10])
bound.up<-a$dualstill[1:10]
shadow.price<-a$duals[1:10]

bound.up<-replace(bound.up, bound.up > 1.0e7, "inf")
bound.down<-replace(bound.down, bound.down < -1.0e7, "-inf")

d1 <- data.frame(name,final.value,shadow.price,constraintRHS,bound.down,bound.up)
kable(d1)
```

name	final.value	shadow.price	constraintRHS	bound.down	bound.up
volume wagon 1	2500	0	5000	-inf	inf
volume wagon 2	4000	0	4000	-inf	inf
volume wagon 3	6000	0	8000	-inf	inf
tonnes cargo 1	0	0	18	-inf	inf
tonnes cargo 2	5	0	10	-inf	inf
tonnes cargo 3	5	2500	5	0	10
tonnes cargo 4	20	1000	20	15	20
weight wagon 1	10	2500	10	5	15
weight wagon 2	8	2500	8	8	8
weight wagon 3	12	2500	12	12	16

As we can see from the table, shadow prices for nonbinding constraints are zero, according to the theory. The shadow price of a constraint indicates the amount by which the objective function value changes given a unit increase in the RHS value of the constraint, assuming all other coefficients remain constant. For example if we increase by a unit the tonnes avaible of cargo type 4, the objective function increase of 1000.

Summary of sensitivity analysis for the changes in objective function coefficients:

```r
name<-c("C1","C2","C3","C4","C5","C6","C7","C8","C9","C10","C11","C12")
value.of.variable<-get.variables(model)
objective.coefficient<-c(2000,2500,5000,3500,2000,2500,5000,3500,2000,2500,5000,3500)
bound.down<-b$objfrom
bound.up<-b$objtill
bound.down=replace(bound.down, bound.down < -1.0e4, "-inf")
bound.up=replace(bound.up, bound.up > 1.0e4, "inf")

d2 <- data.frame(name,value.of.variable,objective.coefficient,bound.down,bound.up)
kable(d2)
```

name	value.of.variable	objective.coefficient	bound.down	bound.up
C1	0	2000	-inf	2500
C2	5	2500	2500	2500
C3	5	5000	5000	inf
C4	0	3500	-inf	3500
C5	0	2000	-inf	2500
C6	0	2500	-inf	2500
C7	0	5000	-inf	5000
C8	8	3500	3500	3500
C9	0	2000	-inf	2500
C10	0	2500	-inf	2500
C11	0	5000	-inf	5000
C12	12	3500	3500	inf

The objective coefficient C1 was 2000 therefore it can increase up to 2500 or decrease up to $-inf$ without changing the optimal solution assuming all other coefficients remain constant. The range of variability of C2 and C8 are 0 then we may are in the special case of a degenerate solution.

Questions about LP

1. No, it isn't possible for an LP model to have exactly two optimal solutions. An LP model have one optimal solution when the level curve intersects the feasible region at a single point and have more than one optimal solution when the level curve is parallel to the edge of the feasible region. In such case, all the points of the edge become optimal solutions. So, if a linear program model has more than one solution, it has infinitely many.
   An Integer Linear Programming model insted, can have exactly two optimal solutions.

2. max $2x_1 + 3x_2$
   min $-2x_1 - 3x_2$
   These objective functions for an LP model are equivalent. This because maximize the function $2x_1 + 3x_2$ means find the value M of the objective function for which:
   $2x_1 + 3x_2 \leq M$ with $x_1, x_2 \in$ faesible region
   $2x_1 + 3x_2 \leq M \iff -2x_1 - 3x_2 \geq -M$
   $-2x_1 - 3x_2 \geq -M$ is the inequality to minimize the objective function $-2x_1 - 3x_2$ (where the minimum is $-M$).
   So the point at which the first function attains its maximum and the point at which the second function attains its minimumum are the same.
   IN an LP model the only difference will be in the objective functions, whose value will be opposite.

3. In a linear programming problem the constraint is of the form:
   $a_{k1}x_1 + a_{k2}x_2 + ... + a_{kn}x_n \leq b_k$ or

$a_{k1}x_1 + a_{k2}x_2 + ... + a_{kn}x_n = b_k$ or
$a_{k1}x_1 + a_{k2}x_2 + ... + a_{kn}x_n \geq b_k$

That is, the constrain is a linear function of x_i with $i = 1, ..., n$.
So the constrains b, d and e cannot be included as a constraint in a linear programming problem. Infact:

a. $2x_1 + x_2 - 3x_3$ is linear
b. $2x_1 + \sqrt{x_2}$ $\sqrt{x_2}$ is not linear
c. $4x_1 - \frac{1}{2}x_2$ is linear
d. $\frac{3x_1+2x_2x_1-3x_3}{x_1+x_2+x_3}$ $2x_2x_1$ is not linear
e. $3x_1^2 + 7x_2$ $3x_1^2$ is not linear