# Operations Research

## Laboratory Session 4: Solving linear models with R (part II)

### by Josu Ceberio and Ana Zelaia

In the previous lab session, the aim was to implement the functions to calculate all the feasible basic solutions of any given linear model. To that end, the definition of the system of linear equations was sufficient, and we did not care about the objective function to accomplish our tasks.

Conversely, in this lab-session, it is our aim to implement the needed functions to solve any given linear model optimally (when it is possible). When the model is feasible, the optimal solution among the basic feasible solutions will be returned. If the model has multiple optimal solutions, then, the function will return all of them. As in the previous lab session, the unbounded problems are out of the scope of the practice.

$$\begin{aligned} \max \quad z = \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \\ \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

## The `solveProblem` functions

Firstly, given a linear model written in the maximization standar form, the function will verify that the problem is feasible (at least one solution exists for the model). If it is so, then, it will implement the routines that calculate the set of all basic solutions, and using the objective function $z$, return the optimal solution. Specifically, the function must return a list that containts: (1) a vector (if the problem has only one solution) or a matrix (if it has multiple solutions), and (2) the objective value $z^*$ of the optimal solution.

The function will print a message in the standard output providing information related to the set of feasible solutions (their number). If the problem is not feasible, it should print a zero. The `message` function can be very helpful to do this task.

Provide two implementations of the functions: using `for` loops, and `apply` functions.

To check the correctness of the functions implemented, let us consider the following linear model:

$$\begin{aligned} \max \ z = 3x_1 + 4x_2 + 5x_3 + 6x_4 \\ \text{subject to} \qquad\qquad\qquad \\ 2x_1 + x_2 + x_3 + 8x_4 = 6 \\ x_1 + x_2 + 2x_3 + x_4 = 4 \\ x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

```r
A <- matrix(c(2, 1, 1, 8, 1, 1, 2, 1), nrow=2, byrow=TRUE)
b <- c(6, 4)
c <- c(3, 4, 5, 6)
```

The optimal solution: $x^* = (0.0000000, 3.7142857, 0.0000000, 0.2857143)$, $z^* = 16.57143$

```r
# Auxiliary function for evaluating the objective function with the
# solution provided (x) and the coefficients of the objective function (funCoeff).
calculateObjectiveFun <- function(x, funCoeff) {
  val <- 0
  for (i in 1:length(funCoeff)) {
    val <- val + (x[i] * funCoeff[i])
  }
  return(val)
}
```

**Exercise 1.** Implementation of `solveProblem` using `for` loops. To ease your task, employ the function `basic.feasible.solutions_for` implemented in the previous laboratory session. Return the result in a list.

```r
solveProblem_for <- function(A, b, c){
  # Get the feasible solutions with the solitions_for() method
  lst <- basic.feasible.solutions_for(A, b)

  # Print a message with the number of feasible solutions
  message("Number of feasible solutions: ")
  message(length(lst))

  # If the length of lst is 0, no feasible solutions, print alert message
  # and return NULL.
  if (length(lst) == 0) {
    message("There is no feasible solution!")
    return(NULL)
  }

  # Init the variables:
  # - zValues for saving z values of each solution.
  # - optimalLst for returning the optimal solutions.
  # maxVal for saving the optimal z value.
  zValues <- list()
  optimalLst <- list()
  maxVal <- -Inf

  # For each solution:
  # - Calculate the Z value evaluating the objective function.
  # - Save the zValue in zValues list.
  # - If the Z value is greater than the previous optimal value, change maxVal.
  i <- 1
  for (l in lst) {
    v <- calculateObjectiveFun(l, c)
    zValues[i] <- v
    i <- i + 1
    if (v > maxVal) {
      maxVal <- v
    }
  }

  # Append to the optimalLst list the solutions with the Z value equal to the
  # optimal value saved in maxVal.
  i <- 1
  lt <- list()
  for (l in 1:length(lst)) {
    if (zValues[l] == maxVal) {
      optimalLst[i] <- lst[l]
      i <- i + 1
    }
  }
  # Insert the optimal value as another element of the list.
  optimalLst[[i]] <- maxVal

  # Return the list with the optimal solutions and the optimal value.
  return(optimalLst)
}
solveProblem_for(A,b,c)
# The problem has only one optimal solution.
# $solutions
# $solutions[[1]]
# [1] 0.0000000 3.7142857 0.0000000 0.2857143
#
#
# $optimal.val
# [1] 16.57143
```

**Exercise 2.** Implementation of `solveProblem` using `apply` fuction. To ease your task, employ the function `basic.feasible.solutions_apply` implemented in the previous laboratory session. Return the result in a matrix.

```
solveProblem_apply <- function(A, b, c){
  # Get the solutions matrix using solutios_apply() method.
  x <- basic.feasible.solutions_apply(A, b)

  # Print a message with the number of feasible solutions
  message("Number of feasible solutions: ")
  message(ncol(x))

  # If the number of columns of x is 0, no feasible solutions, print alert
  # message and return NULL.
  if (ncol(x) == 0) {
    message("There is no feasible solution!")
    return(NULL)
  }

  # Get the values for each solution evaluating with the auxiliary function.
  vals <- apply(x, 2, calculateObjectiveFun, funCoeff=c)

  # Get the maximal value of the list of Z values.
  maxVal <- max(vals)

  # Vector/matrix with the optimal solutions, only add the solutions with Z
  # value equal to maxVal.
  optimalM <- x[, vals == maxVal]

  # Set a list for returning the result.
  resLst <- list()

  # Add the vector/matrix of optimal solutions as a first element of the list.
  resLst[[1]] <- optimalM

  # Add the optimal value as a second element of the list.
  resLst[[2]] <- maxVal
  # Return the list with the optimal solutions and the optimal value.
  return(resLst)
}

solveProblem_apply(A,b,c)
# The problem has only one optimal solution.
# $solutions
# [1] 0.0000000 3.7142857 0.0000000 0.2857143
#
# $optimal.val
# [1] 16.57143
```

## Linear Models

Use the implemented functions to solve the following problems, and check the correctness of the solutions.

**Problem 1.**

$$\begin{aligned} \max\ z = -2x_1 - 4x_2 - 3x_3 \\ \text{subject to} \qquad\qquad \\ 2x_1 + x_2 + 2x_3 \geq 8 \\ 4x_1 + 2x_2 + 2x_3 \geq 10 \\ 6x_1 + x_2 + 4x_3 \geq 12 \\ x_1, x_2, x_3 \geq 0 \end{aligned}$$

```
# Implement here.
A <- matrix(c(2, 1, 2, -1, 0, 0, 4, 2, 2, 0, -1, 0, 6, 1, 4, 0, 0, -1), nrow=3, byrow=TRUE)
b <- c(8, 10, 12)
c <- c(-2, -4, -3, 0, 0, 0)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

There is a unique optimal basic feasible solution for the problem.

$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*) = (4, 0, 0, 0, 6, 12)$, $z^* = -8$

**Problem 2.**

$$\begin{aligned} \min\ z = 2x_1 + x_2 + 3x_3 + 2x_4 \\ \text{subject to} \qquad\qquad\qquad \\ 2x_1 + 2x_2 + 2x_3 + 2x_4 \geq 22 \\ 4x_1 + 4x_2 + x_3 + 4x_4 \leq 20 \\ 2x_1 + 8x_2 + 2x_3 + x_4 \geq 15 \\ x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

```
# Implement here.
A <- matrix(c(2, 2, 2, 2, -1, 0, 0, 4, 4, 1, 4, 0, 1, 0, 2, 8, 2, 1, 0, 0, -1), nrow=3, byrow=TRUE)
b <- c(22, 20, 15)
c <- c(-2, -1, -3, -2)
# Change the sign of the Z optimal value.
```

```
solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

There is a unique optimal basic feasible solution for the problem

$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*) = (0, 3, 8, 0, 0, 0, 25)$, $z^* = 27$

**Problem 3.**

$$\begin{aligned} \max \;\; z = x_1 + 2x_2 \\ \text{subject to} \quad\quad \\ x_1 + 2x_2 \leq 5 \\ x_1 + x_2 \geq 2 \\ x_1 - x_2 \leq 4 \\ x_1, x_2 \geq 0 \end{aligned}$$

There are 2 optimal basic feasible solutions for the problem.

$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*) = (4.3,\ 0.33,\ 0,\ 2.66,\ 0)$, $z^* = 5$

$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*) = (0.0,\ 2.5,\ 0.0,\ 0.5,\ 6.5)$, $z^* = 5$

```
# Implement here.
A <- matrix(c(1, 2, 1, 0, 0, 1, 1, 0, -1, 0, 1, -1, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(5, 2, 4)
c <- c(1, 2)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**Problem 4.**

$$\begin{aligned} \max \;\; z = 3x_1 + 4x_2 \\ \text{subject to} \quad\quad \\ x_1 - 2x_2 \leq 4 \\ x_1 + x_2 \geq 6 \\ 2x_1 + 3x_2 \leq 2 \\ x_1, x_2 \geq 0 \end{aligned}$$

There is no feasible solution for the problem

```
# Implement here.
A <- matrix(c(1, -2, 1, 0, 0, 1, 1, 0, -1, 0, 2, 3, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(4, 6, 2)
c <- c(3, 4)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

## Solving the linear models in the list of Simplex exercises

Solve the problems in the Simplex list of exercises using the functions `solveProblem_for` and `solveProblem_apply`, and verify the correctness of the obtained solutions.

**6.1** There is a unique optimal basic feasible solution for the problem.

$x^* = (3.6\ 0.8\ 0.0\ 0.0)$, $z^* = 2.8$

```r
# Implement here.
A <- matrix(c(1, -2, 1, 0, 4, -3, 0, 1), nrow=2, byrow=TRUE)
b <- c(2, 12)
c <- c(1, -1)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**6.2** There are three basic feasible solutions that correspond to the same extreme point $(x_1^* = 6, x_2^* = 0)$.

$x^* = (6\ 0\ 0\ 18\ 0)$ $x^* = (6\ 0\ 0\ 18\ 0)$ $x^* = (6\ 0\ 0\ 18\ 0)$, $z^* = 6$

```r
# Implement here.
A <- matrix(c(1, 6, -1, 0, 0, -2, 3, 0, 1, 0, 1, 2, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(6, 6, 6)
c <- c(1, 1)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**6.3** There is a unique optimal basic feasible solution for the problem

$x^* = (3.33\ 0.33\ 10.00\ 0.00\ 0.00)$, $z^* = 12$

```r
# Implement here.
A <- matrix(c(-2, 2, 1, 0, 0, 2, -2, 0, 1, 0, 1, -4, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(4, 6, 2)
c <- c(4, -4)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**6.4** There are 2 optimal basic feasible solutions for the problem.

$x^* = (4.3\ 0.33\ 0\ 2.66\ 0)$, $z^* = 5$

$x^* = (0.0\ 2.5\ 0.0\ 0.5\ 6.5)$, $z^* = 5$

```r
# Implement here.
A <- matrix(c(1, 2, 1, 0, 0, 1, 1, 0, -1, 0, 1, -1, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(5, 2, 4)
c <- c(1, 2)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**6.5** There are 2 optimal basic feasible solutions for the problem.

$x^* = (1\ 2\ 3\ 0\ 0)$, $z^* = 6$

$x^* = (2.5\ 0.5\ 0.0\ 0.0\ 1.5)$, $z^* = 6$

```r
# Implement here.
A <- matrix(c(1, -1, 1, 0, 0, 2, 2, 0, 1, 0, 1, 2, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(2, 6, 5)
c <- c(2, 2)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**6.6** The problem is unbounded. Do not do.

**6.7** The problem is unbounded. Do not do.

**6.8** There is no feasible solution for the problem

```r
# Implement here.
A <- matrix(c(1, -2, 1, 0, 0, 1, 1, 0, -1, 0, 2, 3, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(4, 6, 2)
c <- c(3, 4)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**7.1** There is a unique optimal basic feasible solution for the problem

$x^* = (0\ 8\ 0\ 12\ 0)$, $z^* = 16$

```r
# Implement here.
A <- matrix(c(1, -1, 1, 1, 0, 2, 1, 4, 0, 1), nrow=2, byrow=TRUE)
b <- c(4, 8)
c <- c(3, 2, 1)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**7.2** There is a unique optimal basic feasible solution for the problem

$x^* = (1\ 7\ 0\ 9\ 0)$, $z^* = 12$. Sign change. . .

$x^* = (-1\ 7\ 0\ 9\ 0)$, $z^* = -12$.

```r
A <- matrix(c(1, 1, -1, 0, 0, 1, 2, -1, -1, 0, -1, 1, 2, 0, 1), nrow=3, byrow=TRUE)
b <- c(6, 4, 8)
c <- c(-5, 1, 2)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**7.3** There is a unique optimal basic feasible solution for the problem

$x^* = (5\ 6\ 0\ 0\ 0\ 6)$, $z^* = 13$. Sign change. . . $z^* = -13$

```r
A <- matrix(c(3, -1, 2, 1, 0, 0, -2, 4, 1, 0, 1, 0, -4, 4, 8, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(9, 14, 10)
c <- c(1, -3, 2)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**7.4** There are 2 optimal basic feasible solutions for the problem

$x^* = (0\ 2\ 0\ 2\ 0\ 0)$, $x^* = (0\ 3\ 0\ 0\ 2\ 0)$, $z^* = -24$.

Sign change... $z^* = 24$

```
A <- matrix(c(2, 4, 2, 1, -1, 0, -4, 4, -1, 2, 0, -1), nrow=2, byrow=TRUE)
b <- c(10, 12)
c <- c(10, 8, 6, 4)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**7.5** There are 2 optimal basic feasible solutions for the problem

$x^* = (4.5\ 0.0\ 0.0\ 0.5\ 0.0\ 0.0\ 0.0)$,

$x^* = (0\ 0\ 0\ 23\ 9\ 0\ 0)$, $z^* = 73$.

Sign change... $z^* = -73$

```
A <- matrix(c(2, 1, 1, 2, 1, 0, 0, 8, 4, -2, -1, 0, -1, 0, 4, 7, 2, 1, 0, 0, 1), nrow=3, byrow=TRUE)
b <- c(2, 10, 4)
c <- c(9, 5, 4, 1)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**7.6** There is no feasible solution for the problem.

```
A <- matrix(c(3, 2, 2, 1, 1, 1, 0, 2, 1, 3, 2, 2, 0, -1), nrow=2, byrow=TRUE)
b <- c(9, 10)
c <- c(16, -2, -1, 2, 3)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```

**7.7** Unbounded problem. Do not do this exercise.

**7.8** There is no feasible solution for the problem.

```
A <- matrix(c(1, 2, 2, 1, 1, 1, 0, 2, 1, 3, 2, 2, 0, -1), nrow=2, byrow=TRUE)
b <- c(2, 12)
c <- c(-3, -1, 2, 2, -1)

solveProblem_for(A, b, c)
solveProblem_apply(A, b, c)
```