Instituto Superior Técnico

Laboratory Report
Nº 3
Optimization

Group 6

Félix Saraiva – 98752

Miguel Ribeiro – 87553

Miguel Dias Gaio – 82608

2022

Table of Contents

Table of Figures:

List of Tables:

Objectives

The aim of this laboratory work is to study the use of optimization methods in the management and dimensioning of networks and systems. (Valadas, 2021)

A. Optimization Techniques and Telecommunications Problems

Exercise 1

In this exercise we wish to route four flows in the network of the following figure, the networks are:

- Flow 1, between nodes 1 and 4, with 4.5 Mb/s.
- Flow 2, between nodes 2 and 4 with 2.5 Mb/s.
- Flow 3, between nodes 2 and 3, with 4.5 Mb/s.
- Flow 4, between nodes 1 and 3, with 1.5 Mb/s.

All links have 10 Mb/s of capacity. The objective is to determine the routing solution that minimizes the maximum link load with:

a) Bifurcated routing.
b) Non-bifurcated routing.

And to compare the two solutions.

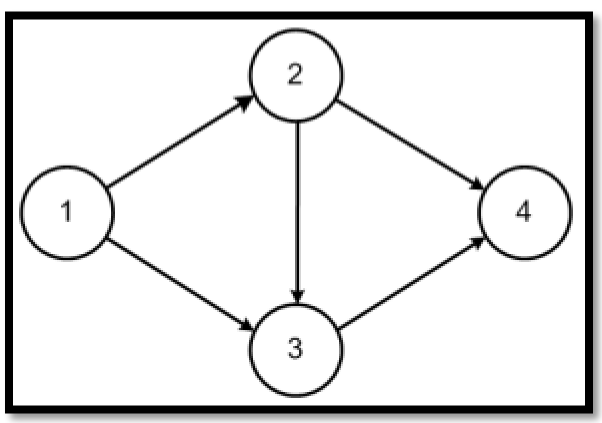

Figure 1 - Packet switched network

Part a) Bifurcated routing

In order to calculate the routing solution that minimizes the maximum link utilization with bifurcated routing, we have developed a script present in Annex A. This script finds the optimal solution of Linear Programming problems using the *lpSolve* library in R.

To obtain the desired results, we must input into the script the coefficients of variables on the objective function, the coefficients of variables on constraints, the right-hand side of the constraints, the direction of constraints and finally use these inputs in the *lpSolve* functions and print the optimal solution and the optimal value.

We will now proceed to demonstrate how to obtain such input values regarding this exercise.

Link-Path Formulation:

$b_{14} = 4.5$| $b_{24} = 2.5$| $b_{23} = 4.5$| $b_{13} = 1.5$| $C_{xy} = 10$

- Minimize$_{\{x,r\}}$ $F = r$
- Subject to:
    - $b_{23} = x_{23} = 4.5$
    - $b_{14} = x_{134} + x_{124} + x_{1234}$
    - $b_{24} = x_{234} + x_{24}$
    - $b_{13} = x_{123} + x_{13}$

      Routing Constraints

    - $x_{124} + x_{1234} + x_{123} \leq rC_{12}$
    - $x_{134} + x_{13} \leq rC_{13}$
    - $x_{234} + x_{23} + x_{123} + x_{1234} \leq rC_{23}$
    - $x_{124} + x_{24} \leq rC_{24}$
    - $x_{134} + x_{1234} + x_{234} \leq rC_{34}$

      Capacity Constraints

    - $x_{134}, x_{124}, x_{1234}, x_{234}, x_{24}, x_{23}, x_{123}, x_{13} \geq 0$
    - $0 \leq r \leq 1$

- $4.5 = x_{134} + x_{124} + x_{1234}$
- $2.5 = x_{234} + x_{24}$
- $1.5 = x_{123} + x_{13}$
- $x_{124} + x_{1234} + x_{123} - 10r \leq 0$
- $x_{134} + x_{13} - 10r \leq 0$
- $x_{234} + x_{123} + x_{1234} - 10r \leq -4.5$
- $x_{124} + x_{24} - 10r \leq 0$
- $x_{134} + x_{1234} + x_{234} - 10r \leq 0$

In the figure below we can see the representation of the above input values in our script in R.

```
# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,0,0,0,1)

# Coefficient of decision variables on restrictions
A= matrix(c(0,0,0,0,0,1,1,1,0,
            1,1,0,0,0,0,0,0,0,
            0,0,0,1,1,0,0,0,0,
            0,0,0,1,0,1,0,1,-10,
            0,0,0,0,1,0,1,0,-10,
            0,1,0,1,0,0,0,1,-10,
            1,0,0,0,0,1,0,0,-10,
            0,1,0,0,0,0,1,1,-10),
          nrow = 8, byrow = TRUE)

#Right hand side for constraints
B= c(4.5,2.5,1.5,0,0,-4.5,0,0)

cd= c("=","=","=","<=","<=","<=","<=","<=")
```

Figure 2 - Bifurcation input values in R

Table 1 - Routing Solution with bifurcated routing

Optimal Solution	Optimal Value
x_{24}=2.5; x_{234}=0; x_{12}=0; x_{123}=0; x_{13}=1.5; x_{124}=2; x_{134}=2.5; x_{1234}=0; r=0.45	0.45

Part b) Non-bifurcated routing

In order to calculate the routing solution that minimizes the maximum link utilization with non-bifurcated routing, we have developed a script present in Annex B. This script finds the optimal solution of Linear Programming problems using the lpSolve library in R. The differences between this script and the one from the previous exercise is that it takes the vector bin with the number of paths available, and it uses the binary.vec lpSolve function to

Link-Path Formulation:

Here we apply the same mathematical programming formulation as with bifurcated routing, where x's are proportions, but in this case the routing variables are binary.

$b_{14} = 4.5$| $b_{24} = 2.5$| $b_{23} = 4.5$| $b_{13} = 1.5$| $C_{xy} = 10$

- Minimize $_{\{x,\, r\}}$ $F = r$
- Subject to:
    - $x_{23} = 1$
    - $1 = x_{134} + x_{124} + x_{1234}$
    - $1 = x_{234} + x_{24}$
    - $1 = x_{123} + x_{13}$

    Routing Constraints

    - $x_{124}b_{14} + x_{1234}b_{14} + x_{123}b_{13} \leq rC_{12}$
    - $x_{134}b_{14} + x_{13}b_{13} \leq rC_{13}$
    - $x_{234}b_{24} + x_{23}b_{23} + x_{123}b_{13} + x_{1234}b_{14} \leq rC_{23}$
    - $x_{124}b_{14} + x_{24}b_{24} \leq rC_{24}$
    - $x_{134}b_{14} + x_{1234}b_{14} + x_{234}b_{24} \leq rC_{34}$

    Capacity Constraints

    - $x_{134}, x_{124}, x_{1234}, x_{234}, x_{24}, x_{23}, x_{123}, x_{13} \in [0,1]$
    - $0 \leq r \leq 1$

- $1 = x_{134} + x_{124} + x_{1234}$
- $1 = x_{234} + x_{24}$
- $1 = x_{123} + x_{13}$
- $4.5x_{124} + 4.5x_{1234} + 1.5x_{123} - 10r \leq 0$
- $4.5x_{134} + 1.5x_{13} - 10r \leq 0$
- $2.5x_{234} + 1.5x_{123} + 4.5x_{1234} - 10r \leq -4.5$
- $4.5x_{124} + 2.5x_{24} - 10r \leq 0$
- $4.5x_{134} + 4.5x_{1234} + 2.5x_{234} - 10r \leq 0$
- $x_{134}, x_{124}, x_{1234}, x_{234}, x_{24}, x_{23}, x_{123}, x_{13} \in [0,1]$
- $0 \leq r \leq 1$

In the figure below we can see the representation of the above input values in our script in R.

```
# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,0,0,0,1)
bin = c(1,2,3,4,5,6,7,8)
# Coefficient of decision variables on restrictions
A= matrix(c(0,0,0,0,0,1,1,1,0,
            1,1,0,0,0,0,0,0,0,
            0,0,0,1,1,0,0,0,0,
            0,0,0,1.5,0,4.5,0,4.5,-10,
            0,0,0,0,1.5,0,4.5,0,-10,
            0,2.5,0,1.5,0,0,0,4.5,-10,
            2.5,0,0,0,0,4.5,0,0,-10,
            0,2.5,0,0,0,0,4.5,4.5,-10),
          nrow = 8, byrow = TRUE)

#Right hand side for constraints
B= c(1,1,1,0,0,-4.5,0,0)

cd= c("=","=","=","<=","<=","<=","<=","<=")
```

Figure 3 - Non-bifurcation input values in R

Table 2 - Routing Solution with non-bifurcated routing

Optimal Solution	Optimal Value
x_{24}=1; x_{234}=0; x_{12}=0; x_{123}=1; x_{13}=0; x_{124}=0; x_{134}=1; x_{1234}=0; r=0.6	0.60

Exercise 2

Considering again the network of Figure 1, with the same offered traffic, we now wish to determine the link capacities and the flow routes that minimize the cost of the network.

The link capacities must be a multiple of **2Mb/s**, and the cost is **1000 Euros** for each 2 Mb/s module. Using the *lpSolve* package we have developed two scripts supported by the adequate mathematical programming formulations. This time the scripts represent joint optimization of the routing and the link capacity assignments with the purpose of minimizing the cost of the installed solution

Part a) Bifurcated routing

Script present in Annex C.

$a = 2_{Mb/s}$ | $E = 1000$ | $b_{14} = 4.5$ | $b_{24} = 2.5$ | $b_{23} = 4.5$ | $b_{13} = 1.5$ |

Link-Path Formulation:

- Minimize $_{\{x,\, r\}}\ Ey_{12} + Ey_{13} + Ey_{23} + Ey_{24} + Ey_{34}$
  - $x_{23} = 1$
  - $1 = x_{134} + x_{124} + x_{1234}$
  - $1 = x_{234} + x_{24}$
  - $1 = x_{123} + x_{13}$

    Routing Constraints

  - $x_{124}b_{14} + x_{1234}b_{14} + x_{123}b_{13} \le ay_{12}$
  - $x_{134}b_{14} + x_{13}b_{13} \le ay_{13}$
  - $x_{234}b_{24} + x_{23}b_{23} + x_{123}b_{13} + x_{1234}b_{14} \le ay_{23}$
  - $x_{124}b_{14} + x_{24}b_{24} \le ay_{24}$
  - $x_{134}b_{14} + x_{1234}b_{14} + x_{234}b_{24} \le ay_{34}$

    Capacity Constraints

  - $x_{134}, x_{124}, x_{1234}, x_{234}, x_{24}, x_{23}, x_{123}, x_{13} \in [0,1]$
  - $y_{12}, y_{13}, y_{23}, y_{24}, y_{34}$ non-negative integers

---

  - $1 = x_{134} + x_{124} + x_{1234}$
  - $1 = x_{234} + x_{24}$
  - $1 = x_{123} + x_{13}$
  - $4.5x_{124} + 4.5x_{1234} + 1.5x_{123} - 2y_{12} \le 0$
  - $4.5x_{134} + 1.5x_{13} - 2y_{13} \le 0$
  - $2.5x_{234} + 1.5x_{123} + 4.5x_{1234} - 2y_{23} \le -4.5$
  - $4.5x_{124} + 2.5x_{24} - 2y_{24} \le 0$
  - $4.5x_{134} + 4.5x_{1234} + 2.5x_{234} - 2y_{34} \le 0$
  - $x_{134}, x_{124}, x_{1234}, x_{234}, x_{24}, x_{23}, x_{123}, x_{13} \in [0,1]$
  - $y_{12}, y_{13}, y_{23}, y_{24}, y_{34}$ non-negative integers

In the figure below we can see the representation of the above input values in our script in R.

```r
# Order of variables:
# x24,x234,x123,x13,x124,x134,x1234,y12,y13,y23,y24,y34
b14 = 4.5
b24 = 2.5
b23 = 4.5
b13 = 1.5
a = 2
# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,0,0,1000,1000,1000,1000,1000)

# Coefficient of decision variables on restrictions
A= matrix(c(0,0,0,0,1,1,1,0,0,0,0,0,
            1,1,0,0,0,0,0,0,0,0,0,0,
            0,0,1,1,0,0,0,0,0,0,0,0,
            0,0,b13,0,b14,0,b14,-a,0,0,0,0,
            0,0,0,b13,0,b14,0,0,-a,0,0,0,
            0,b24,b13,0,0,0,b14,0,0,-a,0,0,
            b24,0,0,0,b14,0,0,0,0,0,-a,0,
            0,b24,0,0,0,b14,b14,0,0,0,0,-a),
          nrow = 8, byrow = TRUE)

#Right hand side for constraints
B= c(1,1,1,0,0,-4.5,0,0)

#Direction of Constraints
cd= c("=","=","=","<=","<=","<=","<=","<=")

#Indexes of Integer variables (y)
int = c(9,10,11,12,13)
```

Figure 4 - Network design with bifurcated routing parameters

Table 3 - Network Design solution with bifurcated routing

Optimal Solution	Optimal Value
x_{24}=0.4; x_{234}=0.6; x_{123}=0; x_{13}=1; x_{124}=0; x_{134}=1; x_{1234}=0; y_{12} = 0, y_{13}=3; y_{23}=3; y_{24}=1; y_{34}=3	10000€

Part b) Non-bifurcated routing

Script present in Annex D.

The link path formulation in this case is the same has in the previous bifurcated part a). With the difference that now x variables are binary.

In the figure below we can see the representation of the input parameters on our script.

```
# Order of variables:
# x24,x234,x123,x13,x124,x134,x1234,y12,y13,y23,y24,y34
b14 = 4.5
b24 = 2.5
b23 = 4.5
b13 = 1.5
a = 2
# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,0,0,1000,1000,1000,1000,1000)

# Coefficient of decision variables on restrictions
A= matrix(c(0,0,0,0,1,1,1,0,0,0,0,0,
            1,1,0,0,0,0,0,0,0,0,0,0,
            0,0,1,1,0,0,0,0,0,0,0,0,
            0,0,b13,0,b14,0,b14,-a,0,0,0,0,
            0,0,0,b13,0,b14,0,0,-a,0,0,0,
            0,b24,b13,0,0,0,b14,0,0,-a,0,0,
            b24,0,0,0,b14,0,0,0,0,0,-a,0,
            0,b24,0,0,0,b14,b14,0,0,0,0,-a),
            nrow = 8, byrow = TRUE)

#Right hand side for constraints
B= c(1,1,1,0,0,-4.5,0,0)

#Direction of Constraints
cd= c("=","=","=","<=","<=","<=","<=","<=")

#Indexes of Integer variables (y)
int = c(8,9,10,11,12)

bin = c(1,2,3,4,5,6,7)
```

Figure 5 - Non-bifurcated Network design input parameters

Table 4 - Network Design solution with non-bifurcated routing

Optimal Solution	Optimal Value
x_{24}=1; x_{234}=0; x_{123}=1; x_{13}=0; x_{124}=1; x_{134}=0; x_{1234}=0; y_{12} = 3, y_{13}=0; y_{23}=3; y_{24}=4; y_{34}=0	10000€

Exercise 3

Considering the same network, of figure 1, the goal of this exercise is to find a mathematical programming formulation to determine the shortest path from node 1 to node 4. The code of this exercise is present in Annex E.

Looking at the image we can establish 4 different equations that represent how the given network works:

- $x_{12} + x_{13} = 1$
- $x_{24} + x_{34} = 1$
- $x_{12} = x_{23} + x_{24}$
- $x_{13} + x_{23} = x_{34}$

We also know the cost of each static link:

- $C_{12} = C_{23} = C_{34} = 1$
- $C_{13} = C_{24} = 5$

Looking at the network and its costs, we can easily see that the best path and least expensive one is 1 -> 2 -> 4 with the respective cost of 3.

```
# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,1)
bin = c(1,2,3,4,5)
# Coefficient of decision variables on restrictions
A= matrix(c(1,1,0,0,0,0,
            0,0,0,1,1,0,
            -1,0,1,1,0,0,
            0,1,1,0,-1,0,
            1,5,1,5,1,-1),
          nrow = 5, byrow = TRUE)

#Right hand side for constraints
B= c(1,1,0,0,0)

cd= c("=","=","=","=","<=")
```

Figure 6 - Input parameters

Optimal Solution	Optimal Value
$x_{12} = 1$; $x_{13} = 0$; $x_{23} = 1$; $x_{24} = 0$; $x_{34} = 1$;	3

Exercise 4

In this exercise we are to consider a Multi-Protocol Label Switching Network of an ISP with the topology presented in the figure below.

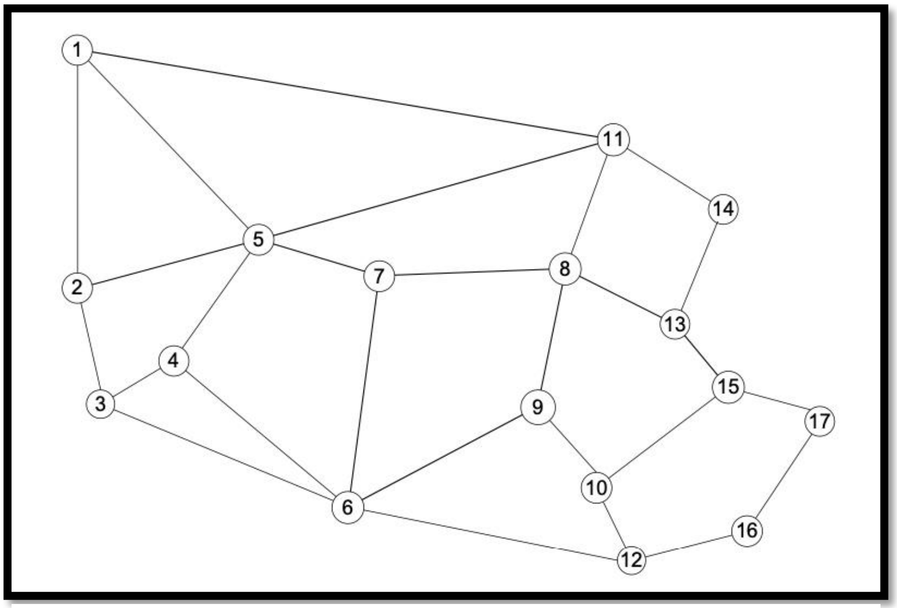

Figure 7 - ISP Network

All links have **1 Gb/s of capacity** and a **propagation delay** that occurs approximately at light speed (**3×108 meters per second**).

We assume that that, in all traffic flows, the **packet arrivals are Poisson processes**, and the packet sizes are exponentially distributed with an **average size of 1000 bytes**. The ISP requires each traffic flow to be routed through a single Label Switched Path.

The optimization goal is to define the routes of all traffic flows, such that the average packet delay is minimized.

Part A

This solution routes each flow through the path with the shortest kilometric length.

By running the script *lsroute_A.R*, which is responsible for calculating the maximum link load and the packet average delay, we get the following results:

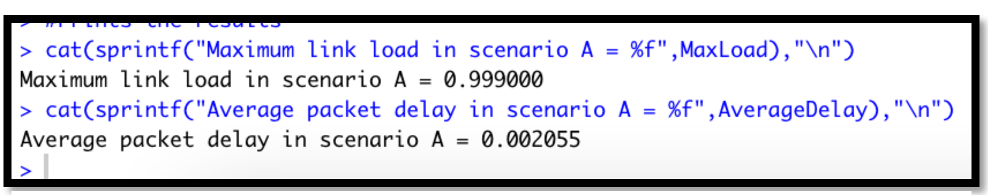

Figure 8 - lsroute_A.R script output

Exercise 5

Considering the same network of the previous exercise we are to write an R script to determine the optimal server locations for replica replacement, considering that the link delays are proportional to their kilometric length. The script must be prepared for any number of locations, and we are to find the values for optimal solution and optimal value for 1,2,3,4 and 5 server locations. Using the script *repplacement_inc.R* we have created the script in Annex H.

In this script we had to create the input parameters for the optimal function followed by the three types of constraints necessary for replica placement. These are focused on the number of server locations selected, that one server is assigned to each node, and that server assigned to each node is a server location.

In order to verify the solutions of our script we have tested our script against a smaller 4 node network represented below:

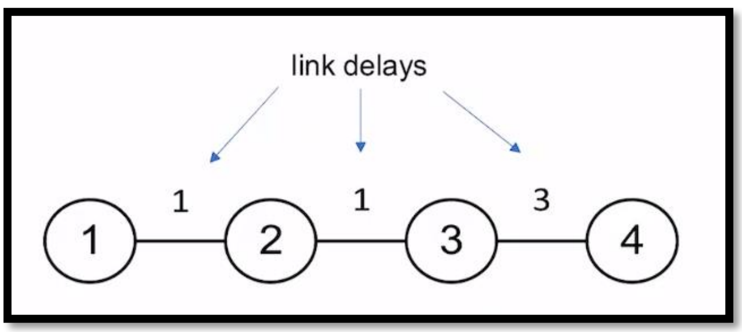

Figure 9 - Test network

In this network it is trivial to conclude the best locations for a small number of servers.

Location of 1 server:

- Node 1 → 1+2+5 = 8
- **Node 2 → 1+1+4 = 6**
- **Node 3 → 1+2+3 = 6**
- Node 4 → 5+4+3 = 12

```
> cat(sprintf("The optimal solution for %d servers locations is to put se
The optimal solution for 1 servers locations is to put servers in node: 3
> cat(sprintf("Optimal value = %f",optimum$objval))
Optimal value = 6.000000
```

Figure 10 - Location of 1 server for small network

Location of 2 servers:

- Node 1-2 → 1+4=5
- Node 1-3 → 1+3 = 4
- Node 1-4 → 1+2 = 3
- Node 2-3 → 1+3 = 4
- **Node 2-4 → 1+1 = 2**
- Node 3-4 → 1+2=3

```
> cat(sprintf("The optimal solution for %d servers locations is to put serv
The optimal solution for 2 servers locations is to put servers in node: 2
 The optimal solution for 2 servers locations is to put servers in node: 4
> cat(sprintf("Optimal value = %f",optimum$objval))
Optimal value = 2.000000
```

Figure 11 - Location of 2 servers for small network

```
#Matrix that defines the network topology; R[i,j]=1 means that there is a link
#from node i to node j
R<<-matrix(c(0,1,0,0,
             1,0,1,0,
             0,1,0,1,
             0,0,1,0),
           nrow=4,ncol=4)

#Matrix that defines the link lengths, in kilometers; L(i,j) is the length of
#the link from i to j
L<<-matrix(c(0,1,0,0,
             1,0,1,0,
             0,1,0,3,
             0,0,3,0),
           nrow=4,ncol=4)
```

Figure 12 - Matrix of the test network in R

After this verification we can conclude that our script is working properly and can now use it for the target network of this exercise.

Location of 1 server:

Table 5 - Location of 1 server

Number of Servers	Nodes	Optimal Value
1	8	3862

Location of 2 servers:

Table 6 - Location of 2 servers

Number of Servers	Nodes	Optimal Value
2	5,15	2412

Location of 3 servers:

Table 7 - Location of 3 servers

Number of Servers	Nodes	Optimal Value
3	5,12,13	1832

Location of 4 servers:

Table 8 - Location of 4 servers

Number of Servers	Nodes	Optimal Value
4	3,5,10,13	1521

Location of 5 servers:

Table 9 - Location of 5 servers

Number of Servers	Nodes	Optimal Value
5	3,5,11,12,15	1269

After analysing the several values from this experiment, we can conclude that as the number of servers increase in the network, the number of optimal value decreases.

We can also conclude that node number 5 is very powerful since it is the one with most connections, it is chosen in all solutions with exception of the first.

Exercise 6

A telephone company leases K circuits to a network operator and the cost of leasing is D euros per year and circuit. We are to create a script that determines the value of K that maximizes the net revenue per minute using the following values:

- λ=2 → incoming call rate
- $1/\mu$=5 → call duration mean
- 5 cents/minute → per call cost
- D=10000 → Euros per year and circuit

Using the script in Annex F we have discovered the following result:

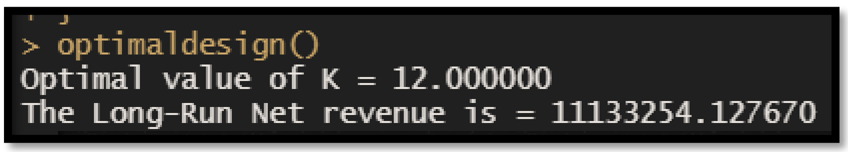

Figure 13 - Value of K of maximum net revenue per minute

Since the blocking probability is around 12% it is relatively high, in this case some customers might be tempted to change to a different telephone company because the quality of service provided by the one on this exercise is not very good, thanks to the level of the blocking probability.

So, the telephone company should decrease its net revenue by leasing more circuits, this way the blocking probability would decrease.

References

Valadas, R. (2021). *Laboratory guide nº 2 – Performance evaluation.* Lisboa: Instituto Superior Tecnico.

Annex A

```
require(lpSolve)

# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,0,0,0,1)

# Coefficient of decision variables on restrictions
A= matrix(c(0,0,0,0,0,1,1,1,0,
            1,1,0,0,0,0,0,0,0,
            0,0,0,1,1,0,0,0,0,
            0,0,0,1,0,1,0,1,-10,
            0,0,0,0,1,0,1,0,-10,
            0,1,0,1,0,0,0,1,-10,
            1,0,0,0,0,1,0,0,-10,
            0,1,0,0,0,0,1,1,-10),
          nrow = 8, byrow = TRUE)

#Right hand side for constraints
B= c(4.5,2.5,1.5,0,0,-4.5,0,0)

cd= c("=","=","=","<=","<=","<=","<=","<=")

# Find the optimal solution

optimum <- lp(direction="min",
              objective.in = C,
              const.mat = A,
              const.dir = cd,
              const.rhs = B)

print(optimum$solution)
print(optimum$objval)

```

Annex B

```
require(lpSolve)

# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,0,0,0,1)
bin = c(1,2,3,4,5,6,7,8)
# Coefficient of decision variables on restrictions
A= matrix(c(0,0,0,0,0,1,1,1,0,
            1,1,0,0,0,0,0,0,0,
            0,0,0,1,1,0,0,0,0,
            0,0,0,1.5,0,4.5,0,4.5,-10,
            0,0,0,0,1.5,0,4.5,0,-10,
            0,2.5,0,1.5,0,0,0,4.5,-10,
            2.5,0,0,0,0,4.5,0,0,-10,
            0,2.5,0,0,0,0,4.5,4.5,-10),
          nrow = 8, byrow = TRUE)

#Right hand side for constraints
B= c(1,1,1,0,0,-4.5,0,0)

cd= c("=","=","=","<=","<=","<=","<=","<=")

# Find the optimal solution

optimum <- lp(direction="min",
              objective.in = C,
              const.mat = A,
              const.dir = cd,
              const.rhs = B,
              binary.vec = bin)

print(optimum$solution)
print(optimum$objval)

```

Annex C

```
require(lpSolve)
# Order of variables:
# x24,x234,x123,x13,x124,x134,x1234,y12,y13,y23,y24,y34
b14 = 4.5
b24 = 2.5
b23 = 4.5
b13 = 1.5
a = 2
# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,0,0,1000,1000,1000,1000,1000)

# Coefficient of decision variables on restrictions
A= matrix(c(0,0,0,0,1,1,1,0,0,0,0,0,
            1,1,0,0,0,0,0,0,0,0,0,0,
            0,0,1,1,0,0,0,0,0,0,0,0,
            0,0,b13,0,b14,0,b14,-a,0,0,0,0,
            0,0,0,b13,0,b14,0,0,-a,0,0,0,
            0,b24,b13,0,0,0,b14,0,0,-a,0,0,
            b24,0,0,0,b14,0,0,0,0,0,-a,0,
            0,b24,0,0,0,b14,b14,0,0,0,0,-a),
          nrow = 8, byrow = TRUE)

#Right hand side for constraints
B= c(1,1,1,0,0,-4.5,0,0)

#Direction of Constraints
cd= c("=","=","=","<=","<=","<=","<=","<=")

#Indexes of Integer variables (y)
int = c(9,10,11,12,13)


# Find the optimal solution

optimum <- lp(direction="min",
              objective.in = C,
              const.mat = A,
              const.dir = cd,
              const.rhs = B,
              int.vec = int)

print(optimum$solution)
print(optimum$objval)
```

Annex D

```
require(lpSolve)
# Order of variables:
# x24,x234,x123,x13,x124,x134,x1234,y12,y13,y23,y24,y34
b14 = 4.5
b24 = 2.5
b23 = 4.5
b13 = 1.5
a = 2
# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,0,0,1000,1000,1000,1000,1000)

# Coefficient of decision variables on restrictions
A= matrix(c(0,0,0,0,1,1,1,0,0,0,0,0,
            1,1,0,0,0,0,0,0,0,0,0,0,
            0,0,1,1,0,0,0,0,0,0,0,0,
            0,0,b13,0,b14,0,b14,-a,0,0,0,0,
            0,0,0,b13,0,b14,0,0,-a,0,0,0,
            0,b24,b13,0,0,0,b14,0,0,-a,0,0,
            b24,0,0,0,b14,0,0,0,0,0,-a,0,
            0,b24,0,0,0,b14,b14,0,0,0,0,-a),
            nrow = 8, byrow = TRUE)

#Right hand side for constraints
B= c(1,1,1,0,0,-4.5,0,0)

#Direction of Constraints
cd= c("=","=","=","<=","<=","<=","<=","<=")

#Indexes of Integer variables (y)
int = c(8,9,10,11,12)

bin = c(1,2,3,4,5,6,7)

# Find the optimal solution

optimum <- lp(direction="min",
              objective.in = C,
              const.mat = A,
              const.dir = cd,
              const.rhs = B,
              int.vec = int,
              binary.vec = bin)

print(optimum$solution)
print(optimum$objval)

```

Annex E

```
require(lpSolve)

# Coefficients of decision variables on objective function
C= c(0,0,0,0,0,1)
bin = c(1,2,3,4,5)
# Coefficient of decision variables on restrictions
A= matrix(c(1,1,0,0,0,0,
            0,0,0,1,1,0,
            -1,0,1,1,0,0,
            0,1,1,0,-1,0,
            1,5,1,5,1,-1),
          nrow = 5, byrow = TRUE)

#Right hand side for constraints
B= c(1,1,0,0,0)

cd= c("=","=","=","=","<=")

# Find the optimal solution

optimum <- lp(direction="min",
              objective.in = C,
              const.mat = A,
              const.dir = cd,
              const.rhs = B,
              binary.vec = bin)

print(optimum$solution)
print(optimum$objval)

```

Annex F

```
# M/M/K/K System limiting probability
limprob = function(p,i){
  j=0:i
  p^i/factorial(i)/sum(p^j/factorial(j))
}

#Long-run net revenue per minute
netrev = function(K,c,p,D){
  d=D*100/(365*24*60)
  revenue_min=c*p*(1-limprob(p,K))-K*d
  return(revenue_min)
}
optimaldesign = function(){
  #Parameters:
  lambda = 2 #incoming call rate
  c=5 #per call cost
  D = 10000 #per year and circuit cost
  u = 1/c
  p=lambda/u

  Kmax=30
  K=seq(1,Kmax,1)
  rev=c(length=length(K))

  for (n in 1:length(K)){
    rev[n]=netrev(K[n],c,p,D)
  }
  # Number of K with higher net revenue
  optimalK = which.max(rev)
  # Number of revenue of optimal K
  netRev = rev[optimalK]*365*24*60

  cat(sprintf("Optimal value of K = %f \n",optimalK))
  cat(sprintf("The Long-Run Net revenue is = %f \n",netRev))

}
optimaldesign()
```