



**UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH**

**Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa**

Estudi i modelització d'una xarxa d'aigua potable per a la seva millor gestió

ANNEXOS

AUTOR:

VICTOR LOBATO TORRES

DIRECTORS:

RAMON PEREZ MAGRANE

SERGI GRAU TORRENT

Grau en Enginyeria en Tecnologies Industrials

TREBALL FINAL DE GRAU

28 de Setembre de 2021



Índex d'Annex

Annex I: Caracterització de la xarxa	4
Annex II: Ponderació de coeficients emissors equirepartits	10
Annex III: Ponderació de coeficients emissors per longituds.....	19
Annex IV: Ponderació coeficients emissors per probabilitats de trencament.....	31
Annex V: Optimització coeficients emissors equirepartits.	38
Annex VI: Optimització coeficients emissors per longituds.	44
Annex VII: Optimització coeficients emissors per probabilitats de trencament.	52

Annex I: Caracterització de la xarxa

```
#####
```

```
#Autor: Víctor Lobato Torres
```

```
#Data: 18/03/2021
```

```
#Descripció3: Caracteritzacio_Xarxa
```

```
#####
```

```
#####
```

```
#Netejem el workspace
```

```
#i anomenem el fitxer d'entrada i el de sortida
```

```
#####
```

```
rm(list=ls())
```

```
camí<-getwd()
```

```
entrada<-paste(camí,"/data/proba1.inp",sep = "")
```

```
sortida<-paste(camí,"/data/proba1.rpt",sep = "")
```

```
sortida2<-paste(camí,"/data/proba1.bin",sep = "")
```

```
#####
```

```
#Carreguem les llibreries
```

```
#####
```

```
library(epanet2toolkit) #eina per a utilitzar epanet desde entorn de R
```

```
library(epanetReader) # serveix per a llegir resultats de epanet
```

```
library(ggplot2) #crear grafics
```

```
library(reshape2) #crear titols als grafics
```

```
library(hrbrthemes) #posa colors als grafics
```

```
library(tidyverse)
```

```
library(fitdistrplus)
```

```
#####
```

```
#Es hora d'obrir l'epanet
```

```
#####
```

```
ENopen(entrada,sortida,sortida2) #obre l'epanet

numnodes<-ENgetcount("EN_NODECOUNT") #conta quants nodes tenim al model

numlinks<-ENgetcount("EN_LINKCOUNT") #conta quantes tuberies tenim

indexnode<-seq(1,numnodes) #fa una conta començant per 1 dels nodes per a fer un bucle

indexlink<-seq(1,numlinks) #fa una conta començant per 1 de les tuberies per a fer un bucle

#####

# Inicialitzar valors de les canonades

#####

idlink<-NULL #per determinar el bucle

diamlink<-NULL #el primer valor de diámetro sigui nul

ruglink<-NULL #rugositat primer valor 0

lonlink<-NULL #longitud primer valor 0

nodilink<-NULL #node inicial de la tuberia

nodflink<-NULL #node final de la tuberia

loninode<-NULL #longitud node inicial

lonfnode<-NULL #Longitud final del node

pressio_estatica<-NULL #Pressio estatica als nodes

perduescarrega<-NULL #Perdues de carrega totals assumides

#####

#una vegada inicialitzades les dades podem fer un bucle per tal de llegirles

#####

for (i in 1:numlinks)
{
  idlink[i]=ENgetlinkid(i)

  diamlink[i]=ENgetlinkvalue(i,"EN_DIAMETER") #Diametres de les canonades

  ruglink[i]=ENgetlinkvalue(i,"EN_ROUGHNESS") #Rugositat de les canonades

  lonlink[i]=ENgetlinkvalue(i,"EN_LENGTH") #Longitud de les canonades

  nodes<-ENgetlinknodes(i)

  nodilink[i]<-nodes[1]

  nodflink[i]<-nodes[2]
```

```
loninode[i]=(lonlink[i]/2)
lonfnode[i]=(lonlink[i]/2)
}

#####

#Inicialitzem les dades dels nodes

#####

idnode<-NULL
demnode<-NULL
basedemnode<-NULL
pnode<-NULL
elevnode<-NULL
patnode<-NULL
coefemis<-NULL
consumnode<-NULL
longnode<-NULL
total<-NULL

#####

#Realitzem el bucle dels nodes per llegir les dades

#####

for (i in 1:numnodes)
{
  idnode[i]=ENgetnodeid(i)
  demnode[i]=ENgetnodevalue(i,"EN_DEMAND")
  basedemnode[i]=ENgetnodevalue(i,"EN_BASEDEMAND")
  coefemis[i]=ENgetnodevalue(i,"EN_EMITTER")
  elevnode[i]=ENgetnodevalue(i,"EN_ELEVATION")
  patnode[i]=ENgetnodevalue(i,"EN_PATTERN")
}

#####
```

```
#Realitzem la simulació per despres llegir pressio

#y demanda dels nodes

#####

ENsolveH()

for (i in 1:numnodes)

{

  pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")

  consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")

}

#####

#Creem els data.frames per poder llegir les dades

#obtingudes per les canonades i nodes

#####

DadesLinks<-data.frame(indexlink,idlink,lonlink,diamlink,ruglink,nodilink,nodflink)

DadesNodes<-

data.frame(indexnode,idnode,elevnode,basedemnode,patnode,demnode,coefemis,pnode,consumnode)

DadesLinks2=DadesLinks[DadesLinks$lonlink<100,] #Dades de longituds mes petites que 100

DadesLinks3=DadesLinks[DadesLinks$lonlink>100,] #Dades de longituds mes grans que 100

#####

#Llegim el per saber la quantitat total de km de xarxa

#####

tuberias=read.csv("edades_nodes.csv", header=TRUE, sep=";")

names(tuberias)

pipes=read.csv("materials_xarxa.csv", header=TRUE, sep=";")

names(pipes)

DadesLinksreals=DadesLinks[DadesLinks$lonlink>0.15,]

DatosTotales = cbind(DadesLinksreals,tuberias)
```

```
pipes$longitudmaterial=0

for (i in 1:9)
{
  for (j in 1:5281)
  {
    if (pipes$material[i]==DatosTotales$material[j])
    {
      pipes$longitudmaterial[i]=pipes$longitudmaterial[i]+DatosTotales$lonlink[j]
    }
  }
}

#####

#Llegits la quantitat de KM i de materials passem a

#La creacio de grafics

#####

#####

#Distribucio de Longituds

#####

ggplot(data = DadesLinks, aes(x=lonlink))+geom_histogram(bins=70,fill="orange",col="black")+
labs(title = "Distribució de Longituds",x="Longituds (m)")+
theme(plot.title = element_text(hjust=0.5,face = "bold",colour = "cadetblue"))

#####

#Distribucio de Longituds < 100 m

#####

ggplot(data = DadesLinks2, aes(x=lonlink))+geom_histogram(bins=70,fill="orange",col="black")+
labs(title = "Distribució de Longituds menors de 100m",x="Longituds (m)")+
theme(plot.title = element_text(hjust=0.5,face = "bold",colour = "cadetblue"))
```



```
#####  
  
#Distribucio de Longituds > 100 m  
  
#####  
  
ggplot(data = DadesLinks3, aes(x=lonlink))+geom_histogram(bins=70,fill="orange",col="black")+  
labs(title = "Distribució de Longituds majors de 100m",x="Longituds (m)")+  
theme(plot.title = element_text(hjust=0.5,face = "bold",colour = "cadetblue"))  
  
#####  
  
#Caracteritzacio dels materials de la xarxa  
  
#####  
  
ggplot(data=DatosTotales,aes(x=fct_infreq(material),y=lonlink))+geom_col(fill="palegreen4")+  
coord_flip()+labs(title = "Materials Xarxa", x="Materials", y="Longituds (m)")+  
theme(plot.title = element_text(hjust = 0.5, face = "bold", colour = "cadetblue"))  
  
#####  
  
#Caracteritzacio de les edats de la xarxa  
  
#####  
  
ggplot(data=DatosTotales, aes(x=fct_infreq(edat),y=lonlink))+geom_col(fill="palegreen4")+  
coord_flip()+labs(title = "Edats xarxa", x= "Edat (anys)",y="longituds (m)")+  
theme (plot.title = element_text(hjust = 0.5, face = "bold", colour = "cadetblue"))  
  
ENclose() #Tanquem el programa
```

Annex II: Ponderació de coeficients emissors equirepartits.

```
#####
```

```
#Autor: Víctor Lobato Torres
```

```
#Data: 18/03/2021
```

```
#Descripció: Ponderacio_Equirepartits
```

```
#####
```

```
#####
```

```
#Netejem el workspace
```

```
#i anomenem el fitxer d'entrada i el de sortida
```

```
#####
```

```
rm(list=ls())
```

```
camí<-getwd()
```

```
entrada<-paste(camí,"/data/proba1.inp",sep = "")
```

```
sortida<-paste(camí,"/data/proba1.rpt",sep = "")
```

```
sortida2<-paste(camí,"/data/proba1.bin",sep = "")
```

```
#####
```

```
#Carreguem les llibreries
```

```
#####
```

```
library(epanet2toolkit) #eina per a utilitzar epanet desde entorn de R
```

```
library(epanetReader) # serveix per a llegir resultats de epanet
```

```
library(ggplot2) #crear grafics
```

```
library(reshape2) #crear titols als grafics
```

```
library(hrbrthemes) #posa colors als grafics
```

```
library(GA) #algoritmes genetics
```

```
library(Rcpp) #llibreria adscrita als algoritmes genetics
```

```
library(MASS) # 2 llibreria adscrita als algoritmes genetics
```

```
library(survival)
```

```
library(tictoc) #llibreria que calcula el temps de simulacio
```

```
#####  
  
#Es hora d'obrir l'epanet  
  
#####  
  
ENopen(entrada,sortida,sortida2) #obre l'epanet  
  
numnodes<-ENgetcount("EN_NODECOUNT") #conta quants nodes tenim al model  
  
numlinks<-ENgetcount("EN_LINKCOUNT") #conta quantes tuberies tenim  
  
indexnode<-seq(1,numnodes) #fa una conta començant per 1 dels nodes per a fer un bucle  
  
indexlink<-seq(1,numlinks) #fa una conta començant per 1 de les tuberies per a fer un bucle  
  
#####  
  
#Inicialitzem les dades dels nodes  
  
#####  
  
idnode<-NULL #Identitat dels nodes  
  
coefemis<-NULL #Coeficients emissors dels nodes  
  
consumnode<-NULL #Consum dels nodes  
  
pnode<-NULL #Pressio dinamica dels nodes  
  
patnode<-NULL #Pattern dels nodes  
  
elevnode<-NULL #Elevacio dels nodes corresponents  
  
pressio_estatica<-NULL #Pressio estatica dels nodes  
  
perduescarrega<-NULL #Perdues de carrega assolides a la ponderacio  
  
#####  
  
#Realitzem el bucle dels nodes per llegir les dades  
  
#####  
  
for (i in 1:numnodes)  
{  
  
  idnode[i]=ENgetnodeid(i) #Identificacio del node  
  
  coefemis[i]=ENgetnodevalue(i,"EN_EMITTER") #Coeficient Emissors  
  
  patnode[i]=ENgetnodevalue(i,"EN_PATTERN") #Pattern del node  
  
  elevnode[i]=ENgetnodevalue(i,"EN_ELEVATION") #Elevacio del node  
  
}
```

```
#####  
  
#Realitzem la simulació per despres llegir pressio  
  
#y demanda dels nodes  
  
#####  
  
ENSolveH() #Resolucio hidraulica  
  
for (i in 1:numnodes)  
{  
  pnode[i]=ENgetnodevalue(i,"EN_PRESSURE") #Llegim la pressio dinamica del node  
  consumnode[i]=ENgetnodevalue(i,"EN_DEMAND") #Llegim el consum del node  
}  
  
#####  
  
#Implementem el bucle per augmentar les fuites  
  
#Coeficients emissors fins el valor de consum del  
  
#Diposit estipulat  
  
#####  
  
consum_objectiu=-84.5 #Consum del diposit que volem assolir  
  
while (consumnode[5123]>consum_objectiu) #Bucle while per establir dita condicio de consum  
{  
  for (i in seq(1,numnodes-1,1))  
  {  
    Llegir_primer_coef<-ENgetnodevalue(1,"EN_EMITTER")  
    ENsetnodevalue(i,"EN_EMITTER",Llegir_primer_coef+0.000007)  
    coefemis[i]=ENgetnodevalue(i,"EN_EMITTER")  
  }  
  ENSolveH()  
  for (i in 1:numnodes)  
  {  
    pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")  
    consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")  
  }  
}
```

```
print(Llegir_primer_coef)
print(consumnode[5123])
}
for(i in 1:numnodes)
{
  pressio_estatica[i]=310.33-elevnode[i] #Llegim la pressio estatica una vegada adjudicats els coeficients
  perduescarrega[i]=pressio_estatica[i]-pnode[i] #Llegim les perdues de carrega una vegada adjudicats els nodes
}
```

DadesNodes<-data.frame(perduescarrega) #Data frame de les perdues de carrega per crear grafiques mes endavant

```
#####
```

```
#Una vegada resolt el bucle pasem a la simulacio
```

```
#####
```

```
#El primer pas es definir els vectors dels patterns#
```

```
diposit=read.csv("pattern_diposit_elevacio.csv", header=TRUE, sep=";") #CSV proporcionat per l'empresa Patern  
d'elevacio del diposit
```

```
demanda=read.csv("pattern_demanda_nodes.csv", header=TRUE, sep=";") #CSV proporcionat per l'empresa Patern  
de demanda del diposit
```

```
Vector_pattern_elevacio<- diposit$DIPOSIT[1:288] #Vector del pattern d'elevacio per un 1 dia
```

```
Vector_pattern_demanda<- demanda$DEMANDA
```

```
#####
```

```
#Plantegem la simulacio d'1 semana amb reports de 5 min
```

```
#####
```

```
ENsettimeparam("EN_DURATION",604800) #Simulacio d'1 setmana
```

```
ENsettimeparam("EN_REPORTSTEP",300) #Reports de 5 min
```

```
#####
```

```
#Introduim el vector d'elevacio del diposit
```

```
#####
```

```
ENsetpattern(1,Vector_pattern_elevacio)
ENsetnodevalue(5123,"EN_PATTERN",1)
ENsetnodevalue(5123,"EN_ELEVATION",1)

#####

#Introduim el vector de la demanda dels nodes
#####

ENsetpattern(2,Vector_pattern_demanda)

for (i in seq(1,numnodes-1,1))
{
  ENsetnodevalue(i,"EN_PATTERN",2)
}

#####

#Resolem hidraulicament, amb qualitat i creem el report
#####

ENSolveH() #Resolucio hidraulica
ENSolveQ() #Resolucio de qualitat
ENreport() #Report

#Tanquem el programa per llegir els resultats de la simulacio

ENclose()

#####

#Guardem els resultats obtinguts dintre del data.frame
#NodeSimulats
#####

resul<-read.rpt(sortida)
NodeSimulats<-resul$nodeResults
```

```
#####  
  
#Creem un data.frame dels tres nodes d'estudi en qüestio  
  
#####  
  
Node188008=NodeSimulats[NodeSimulats$ID=="188008",]  
Node235846=NodeSimulats[NodeSimulats$ID=="235846",]  
Node518=NodeSimulats[NodeSimulats$ID=="518",]  
Nodeestudi=read.csv("simulacio_empresa.CSV",header = TRUE,sep=";")  
  
#####  
  
#Crea els data.frames podem pasar a calcular els MSE i MAE dels  
  
#Tres nodes en qüestio  
  
#####  
  
Num=2017 #Numero total de variables  
  
#MSE per els tres nodes en qüestio  
  
MSE_188008=(sum((Node188008$Pressure-Nodeestudi$Mig)^2))/(Num) #Llegim el MSE del primer node  
MSE_235846=(sum((Node235846$Pressure-Nodeestudi$Mig.1)^2))/(Num) #Llegim el MSE del segon node  
MSE_518=(sum((Node518$Pressure-Nodeestudi$Mig.2)^2))/(Num) #Llegim el MSE del tercer node  
  
#MAE per els tres nodes en qüestio  
  
MAE_188008=(sum((abs(Node188008$Pressure-Nodeestudi$Mig))^2))/(Num) #Llegim el MAE del primer node  
MAE_235846=(sum((abs(Node235846$Pressure-Nodeestudi$Mig.1))^2))/(Num) #Llegim el MAE del segon node  
MAE_518=(sum((abs(Node518$Pressure-Nodeestudi$Mig.2))^2))/(Num) #Llegim el MAE del tercer node  
  
#####  
  
#Es hora de llegir resultats mitjançant grafiques  
  
#####
```

```
#####  
  
#Primer mirarem la comparacio entre les pressions simulades i  
  
#Les aportades per l'empresa que son el cas d'estudi  
  
#####  
  
Pressio_188008<-NULL  
  
Pressio_235846<-NULL  
  
Pressio_518<-NULL  
  
Pressio_188008_estudi<-NULL  
  
Pressio_235846_estudi<-NULL  
  
Pressio_518_estudi<-NULL  
  
Llegenda_Node_1<-NULL  
  
Llegenda_Node_2<-NULL  
  
Llegenda_Node_3<-NULL  
  
tiempo<-NULL  
  
  
#Llegim les pressions dels tres nodes per els dos casos en qüestio  
  
for (i in 1:Num)  
{  
  Pressio_188008[i]=Node188008$Pressure[i]  
  Pressio_188008_estudi[i]=Nodeestudi$Mig[i]  
  Pressio_235846[i]=Node235846$Pressure[i]  
  Pressio_235846_estudi[i]=Nodeestudi$Mig.1[i]  
  Pressio_518[i]=Node518$Pressure[i]  
  Pressio_518_estudi[i]=Nodeestudi$Mig.2[i]  
  tiempo[i]=Node518$timeInSeconds[i]  
}  
  
#Creem la llegenda de les grafiques següents  
  
for (i in 1:(Num*2))  
{  
  if (i<=Num)
```



```
{
  Llegenda_Node_1[i]="simulacio"
  Llegenda_Node_2[i]="simulacio"
  Llegenda_Node_3[i]="simulacio"
}
if(i>Num)
{
  Llegenda_Node_1[i]="estudi"
  Llegenda_Node_2[i]="estudi"
  Llegenda_Node_3[i]="estudi"
}
}

#Creem els data.frames per crear un sol data frame dels nodes

combinacio_188008 <- data.frame (cbind(Pressio_188008,Pressio_188008_estudi,tiempo))

combinacio_chetada_188008 <- melt(combinacio_188008, measure.vars =
c("Pressio_188008","Pressio_188008_estudi"), variable.name = "variable188008", value.name = "valores188008")

combinacio_235846 <- data.frame (cbind(Pressio_235846,Pressio_235846_estudi,tiempo))

combinacio_chetada_235846 <- melt(combinacio_235846, measure.vars = c("Pressio_235846",
"Pressio_235846_estudi"),variable.name = "variable235846", value.name = "valores235846")

combinacio_518 <- data.frame(cbind(Pressio_518,Pressio_518_estudi,tiempo))

combinacio_chetada_518 <- melt(combinacio_518, measure.vars = c("Pressio_518", "Pressio_518_estudi"),
variable.name = "variable518", value.name = "valores518")

#Creem les grafiques per els tres nodes en questio.

ggplot (combinacio_chetada_188008,
aes(x=tiempo,y=valores188008,fill=variable188008,group=Llegenda_Node_1,colour=Llegenda_Node_1))+
geom_line()+
labs(title = "Diferencia Node 1", x="temps (s)", y= "Pressio (m)")+
theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))

ggplot (combinacio_chetada_235846,
aes(x=tiempo,y=valores235846,fill=variable235846,group=Llegenda_Node_2,colour=Llegenda_Node_2))+
geom_line()+
```

```
labs(title = "Diferencia Node 2", x="temps (s)", y= "Pressio (m)")+  
  
theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))  
  
ggplot                                                                    (combinacio_chetada_518,  
aes(x=tiempo,y=valores518,fill=variable518,group=Llegenda_Node_3,colour=Llegenda_Node_3))+  
  
geom_line()+  
  
labs(title = "Diferencia Node 3", x="temps (s)", y= "Pressio (m)")+  
  
theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))  
  
#Graficas de las perduas de carrega  
  
ggplot(data=DadesNodes,aes(x=perduescarrega))+geom_histogram(bins=70,fill="orange",col="black")+  
  
labs(title = "Perdues de Carrega Equirepartits",x="Perdues de Carrega (m)")+  
  
theme(plot.title = element_text(hjust=0.5,face = "bold",colour = "cadetblue"))  
  
#####  
  
#Comparacio amb grafica normal  
  
#####  
  
distribucio_gamma <- rnorm(5123, 0.86, 0.12)  
  
combinacion <- data.frame(cbind(distribucio_gamma,perduescarrega))  
  
combinacion_chetada <- melt(combinacion, measure.vars = c("distribucio_gamma","perduescarrega"),  
variable.name = "variable", value.name = "valores")  
  
ggplot(combinacion_chetada, aes(x=valores,fill=variable))+  
  
geom_histogram(color="#e9ecf", position = "identity", bins=50, alpha=0.5)+  
  
scale_fill_manual(values = c("#ff6e00","#000000"))+  
  
theme_ipsum()+  
  
labs(title = "Normal - Perdues de carrega", x="Mesures")+  
  
theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))
```

Annex III: Ponderació de coeficients emissors per longituds

```
#####
```

```
#Autor: Víctor Lobato Torres
```

```
#Data: 18/03/2021
```

```
#Descripció: Ponderacio de Longituds
```

```
#####
```

```
#####
```

```
#Netejem el workspace
```

```
#i anomenem el fitxer d'entrada i el de sortida
```

```
#####
```

```
rm(list=ls())
```

```
camí<-getwd()
```

```
entrada<-paste(camí,"/data/proba1.inp",sep = "")
```

```
sortida<-paste(camí,"/data/proba1.rpt",sep = "")
```

```
sortida2<-paste(camí,"/data/proba1.bin",sep = "")
```

```
#####
```

```
#Carreguem les llibreries
```

```
#####
```

```
library(epanet2toolkit) #eina per a utilitzar epanet desde entorn de R
```

```
library(epanetReader) # serveix per a llegir resultats de epanet
```

```
library(ggplot2) #crear grafics
```

```
library(reshape2) #crear titols als grafics
```

```
library(hrbrthemes) #posa colors als grafics
```

```
library(GA) #algoritmes genetics
```

```
library(Rcpp) #llibreria adscrita als algoritmes genetics
```

```
library(MASS) # 2 llibreria adscrita als algoritmes genetics
```

```
library(survival)
```

```
library(tictoc) #llibreria que calcula el temps de simulacio
```

```
#####  
  
#Es hora d'obrir l'epanet  
  
#####  
  
ENopen(entrada,sortida,sortida2) #obre l'epanet  
  
numnodes<-ENgetcount("EN_NODECOUNT") #conta quants nodes tenim al model  
  
numlinks<-ENgetcount("EN_LINKCOUNT") #conta quantes tuberies tenim  
  
indexnode<-seq(1,numnodes) #fa una conta començant per 1 dels nodes per a fer un bucle  
  
indexlink<-seq(1,numlinks) #fa una conta començant per 1 de les tuberies per a fer un bucle  
  
#####  
  
# Inicialitzar valors de les canonades  
  
#####  
  
idlink<-NULL #per determinar el bucle  
  
lonlink<-NULL #longitud primer valor 0  
  
nodilink<-NULL #node inicial de la tuberia  
  
nodflink<-NULL #node final de la tuberia  
  
loninode<-NULL #longitud node inicial  
  
lonfnode<-NULL #Longitud final del node  
  
pressio_estatica<-NULL #pressio estatica, coef=0  
  
perduescarrega<-NULL #Perdues de carrega del model  
  
#####  
  
#una vegada inicialitzades les dades podem fer un bucle per tal de llegirles  
  
#####  
  
for (i in 1:numlinks)  
{  
  
  idlink[i]=ENgetlinkid(i) #adjudiquem una identitat a les canonades  
  
  lonlink[i]=ENgetlinkvalue(i,"EN_LENGTH") #Llegim longituds  
  
  nodes<-ENgetlinknodes(i) #identitat pels nodes  
  
  nodilink[i]<-nodes[1] #Identifiquem el node inicial de la canonada  
  
  nodflink[i]<-nodes[2] #Identifiquem el node final de la canonada
```

```
lonnode[i]=(lonlink[i]/2) #Adjudiquem longitud inicial al node
lonfnode[i]=(lonlink[i]/2) #Adjudiquem longitud final al node
}

#####

#Inicialitzem les dades dels nodes

#####

idnode<-NULL #identitat del node
demnode<-NULL #demanda del node
basedemnode<-NULL #demanda base del node
pnode<-NULL #pressio dinamica al node
elevnode<-NULL #Elevacio del node
coefemis<-NULL #coeficient emissor del node
consumnode<-NULL #consum del node
total<-NULL #Total de metres de la xarxa

#####

#Realitzem el bucle dels nodes per llegir les dades

#####

for (i in 1:numnodes)
{
idnode[i]=ENgetnodeid(i) #Identifiquem el node
demnode[i]=ENgetnodevalue(i,"EN_DEMAND") #Llegim la seva demanda
basedemnode[i]=ENgetnodevalue(i,"EN_BASEDEMAND") #Llegim la demanda base
coefemis[i]=ENgetnodevalue(i,"EN_EMITTER") #Llegim el coeficient emissor
elevnode[i]=ENgetnodevalue(i,"EN_ELEVATION") #Elevacio del node corresponent
}

#####

#Realitzem la simulació per despres llegir pressio
#y demanda dels nodes

#####
```

```
ENsolveH() #Resolucio hidraulica

for (i in 1:numnodes)
{
  pnode[i]=ENgetnodevalue(i,"EN_PRESSURE") #Llegim la pressio dinamica una vegada simulat la xarxa
  consumnode[i]=ENgetnodevalue(i,"EN_DEMAND") #Llegim el consum del node una vegada simulat la xarxa
}

#####

#Creem els data.frame per tal de poder adjudicar longituds als nodes mitjançant la cerca de dos bucles
#####

DadesLinks<-data.frame(indexlink,idlink,lonlink,nodilink,nodflink) #Data.frame de les canonades
DadesNodes<-data.frame(indexnode,idnode,elevnode,basedemnode,demnode,coefemis,pnode,consumnode)
#Data.Frame dels nodes
DadesNodes$lonnode=0 #Adjudiquem el valor total de la xarxa

for (i in 1:numnodes) #Doble bucle per saber el valor de longitud total de la xarxa.
{
  for (j in 1:numlinks)
  {
    if (DadesNodes$indexnode[i]==DadesLinks$nodilink[j])
    {
      DadesNodes$lonnode[i]=DadesNodes$lonnode[i]+DadesLinks$lonlink[j]/2
    }
    if (DadesNodes$indexnode[i]==DadesLinks$nodflink[j])
    {
      DadesNodes$lonnode[i]=DadesNodes$lonnode[i]+DadesLinks$lonlink[j]/2
    }
  }
}

#####

#Creem una variable de coeficient i adjudiquem el valor corresponent amb la longitud atribuïda
#####
```

```
coefemis2<-0 #Nova variable coeficient emissor per poder llegir els primers
total<-sum(DadesNodes$lonnode) #Total longitudinal de la xarxa
for (i in 1:numnodes)
{
  coefemis2[i]=(DadesNodes$lonnode[i]/total)*10 #Adjudiquem el coeficient emissor en relacio a les longituds
}

#####

#Creem un bucle per a ponderar les fuites fins el valor objectiu d'aigua
#lliurada
#####

consum_objectiu=84.5 #Consum del diposit que volem assolir

while (abs(consumnode[5123]+consum_objectiu)>=1) #Bucle while per imposar que es compleixi el consum
modificant coefemis2
{
  if (abs(consumnode[5123]+consum_objectiu)>1)
  {

    for (i in seq(1,numnodes-1,1))
    {
      Vector_Coeficients<-coefemis2[i]
      ENsetnodevalue(i,"EN_EMITTER",Vector_Coeficients*0.98)
      coefemis2[i]=ENgetnodevalue(i,"EN_EMITTER")
    }
  }

  if (abs(consumnode[5123]+consum_objectiu)<1)
  {
    for (i in seq(1,numnodes-1,1))
    {
      Vector_Coeficients<-coefemis2[i]
      ENsetnodevalue(i,"EN_EMITTER",Vector_Coeficients*1.0005)
      coefemis2[i]=ENgetnodevalue(i,"EN_EMITTER")
    }
  }
}
```

```
    }  
  }  
  ENSolveH()  
  for(i in 1:numnodes)  
  {  
    pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")  
    consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")  
  }  
  print(Vector_Coeficients)  
  print(consumnode[5123])  
}  
  
for(i in 1:numnodes) #Llegim la pressio estatica i les perdues de carrega  
{  
  pressio_estatica[i]=310.33-elevnode[i]  
  perduescarrega[i]=pressio_estatica[i]-pnode[i]  
}  
  
DadesNodes2<-data.frame(perduescarrega,coefemis2) #Nou data.frame per representar aquestes dades  
  
#####  
#Una vegada resolt el bucle pasem a la simulacio  
#####  
  
#El primer pas es definir els vectors dels patterns#  
  
diposit=read.csv("pattern_diposit_elevacio.csv", header=TRUE, sep=";") #CSV proporcionat per l'empresa Pattern  
elevacio  
  
demanda=read.csv("pattern_demanda_nodes.csv", header=TRUE, sep=";") #CSV proporcionat per l'empresa Pattern  
demanda  
  
Vector_pattern_elevacio<- diposit$DIPOSIT[1:288] #Vector d'elevacio del pattern 1 dia  
Vector_pattern_demanda<- demanda$DEMANDA #Vector de demanda del pattern 1 dia
```



```
#####  
#Plantegem la simulacio d'1 semana amb reports de 5 min  
#####  
  
ENsettimeparam("EN_DURATION",604800) #Simulacio d'1 setmana  
ENsettimeparam("EN_REPORTSTEP",300) #Reports cada 5 minuts  
  
#####  
#Introduim el vector d'elevacio del diposit  
#####  
  
ENsetpattern(1,Vector_pattern_elevacio)  
ENsetnodevalue(5123,"EN_PATTERN",1)  
ENsetnodevalue(5123,"EN_ELEVATION",1)  
  
#####  
#Introduim el vector de la demanda dels nodes  
#####  
ENsetpattern(2,Vector_pattern_demanda)  
  
for (i in seq(1,numnodes-1,1))  
{  
  ENsetnodevalue(i,"EN_PATTERN",2)  
}  
  
#####  
#Resolem hidraulicament, amb qualitat i creem el report  
#####  
  
ENSolveH() #Hidraulica  
ENSolveQ() #Qualitat  
ENreport() #Report
```

```
#Tanquem el programa per llegir els resultats de la simulacio

ENclose()

#####

#Guardem els resultats obtinguts dintre del data.frame

#NodeSimulats

#####

resul<-read.rpt(sortida)

NodeSimulats<-resul$nodeResults

#####

#Creem un data.frame dels tres nodes d'estudi en qüestio

#####

Node188008=NodeSimulats[NodeSimulats$ID=="188008",] #Primer node proporcionat
Node235846=NodeSimulats[NodeSimulats$ID=="235846",] #Segon node proporcionat
Node518=NodeSimulats[NodeSimulats$ID=="518",] #Tercer node proporcionat
Nodeestudi=read.csv("simulacio_empresa.csv",header = TRUE,sep=";") #Llegim el csv de les dades per els tres nodes

#####

#Creats els data.frames podem pasar a calcular els MSE i MAE dels
#Tres nodes en qüestio

#####

Num=2017 #Numero de mostres

#MSE per els tres nodes en qüestio

MSE_188008=(sum((Node188008$Pressure-Nodeestudi$Mig)^2))/(Num) #Formula del MSE pel primer node
MSE_235846=(sum((Node235846$Pressure-Nodeestudi$Mig.1)^2))/(Num) #Formula del MSE pel segon node
MSE_518=(sum((Node518$Pressure-Nodeestudi$Mig.2)^2))/(Num) #Formula del MSE pel tercer node
```

```
#MAE per els tres nodes en qüestio
```

```
MAE_188008=(sum((abs(Node188008$Pressure-Nodeestudi$Mig))^2))/(Num) #Formula del MAE pel primer node
```

```
MAE_235846=(sum((abs(Node235846$Pressure-Nodeestudi$Mig.1))^2))/(Num) #Formula del MAE pel segon node
```

```
MAE_518=(sum((abs(Node518$Pressure-Nodeestudi$Mig.2))^2))/(Num) #Formula del MAE pel tercer node
```

```
#####
```

```
#Es hora de llegir resultats mitjançant grafiques
```

```
#####
```

```
#####
```

```
#Primer mirarem la comparacio entre les pressions simulades i
```

```
#Les aportades per l'empresa que son el cas d'estudi
```

```
#####
```

```
Pressio_188008<-NULL
```

```
Pressio_235846<-NULL
```

```
Pressio_518<-NULL
```

```
Pressio_188008_estudi<-NULL
```

```
Pressio_235846_estudi<-NULL
```

```
Pressio_518_estudi<-NULL
```

```
Llegenda_Node_1<-NULL
```

```
Llegenda_Node_2<-NULL
```

```
Llegenda_Node_3<-NULL
```

```
tiempo<-NULL
```

```
#Llegim les pressions dels tres nodes per els dos casos en qüestio
```

```
for (i in 1:Num)
```

```
{
```

```
  Pressio_188008[i]=Node188008$Pressure[i]
```

```
  Pressio_188008_estudi[i]=Nodeestudi$Mig[i]
```

```
  Pressio_235846[i]=Node235846$Pressure[i]
```

```
  Pressio_235846_estudi[i]=Nodeestudi$Mig.1[i]
```

```
Pressio_518[i]=Node518$Pressure[i]
Pressio_518_estudi[i]=Nodeestudi$Mig.2[i]
tiempo[i]=Node518$timeInSeconds[i]
}

#Creem la llegenda de les grafiques següents

for (i in 1:(Num*2))
{
  if (i<=Num)
  {
    Llegenda_Node_1[i]="simulacio"
    Llegenda_Node_2[i]="simulacio"
    Llegenda_Node_3[i]="simulacio"
  }
  if(i>Num)
  {
    Llegenda_Node_1[i]="estudi"
    Llegenda_Node_2[i]="estudi"
    Llegenda_Node_3[i]="estudi"
  }
}

#Creem els data.frames per crear un sol data frame dels nodes

combinacio_188008 <- data.frame (cbind(Pressio_188008,Pressio_188008_estudi,tiempo))
combinacio_chetada_188008 <- melt(combinacio_188008, measure.vars =
c("Pressio_188008","Pressio_188008_estudi"), variable.name = "variable188008", value.name = "valores188008")
combinacio_235846 <- data.frame (cbind(Pressio_235846,Pressio_235846_estudi,tiempo))
combinacio_chetada_235846 <- melt(combinacio_235846, measure.vars = c("Pressio_235846",
"Pressio_235846_estudi"),variable.name = "variable235846", value.name = "valores235846")
combinacio_518 <- data.frame(cbind(Pressio_518,Pressio_518_estudi,tiempo))
combinacio_chetada_518 <- melt(combinacio_518, measure.vars = c("Pressio_518", "Pressio_518_estudi"),
variable.name = "variable518", value.name = "valores518")
```

#Creem les grafiques per els tres nodes en questio.

```
ggplot                                                                    (combinacio_chetada_188008,  
aes(x=tiempo,y=valores188008,fill=variable188008,group=Llegenda_Node_1,colour=Llegenda_Node_1))+  
  
geom_line()+  
  
labs(title = "Diferencia Node 1", x="temps (s)", y= "Pressio (m)")+  
  
theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))
```

```
ggplot                                                                    (combinacio_chetada_235846,  
aes(x=tiempo,y=valores235846,fill=variable235846,group=Llegenda_Node_2,colour=Llegenda_Node_2))+  
  
geom_line()+  
  
labs(title = "Diferencia Node 2", x="temps (s)", y= "Pressio (m)")+  
  
theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))
```

```
ggplot                                                                    (combinacio_chetada_518,  
aes(x=tiempo,y=valores518,fill=variable518,group=Llegenda_Node_3,colour=Llegenda_Node_3))+  
  
geom_line()+  
  
labs(title = "Diferencia Node 3", x="temps (s)", y= "Pressio (m)")+  
  
theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))
```

#####

#Grafica Perdues de carrega

#####

```
ggplot(data=DadesNodes2, aes(x=perduescarrega))+geom_histogram(bins=70,fill="orange",col="black")+  
  
labs(title = "Perdues de Carrega Longituds", x="Perdua de Carrega (m)")+  
  
theme(plot.title = element_text(hjust = 0.5, face = "bold", colour = "cadetblue"))
```

#Ajustacio amb grafica de distribucio normal

```
distribucio_normal<- rnorm(5123, 1, 0.23)  
  
combinacion <- data.frame(cbind(distribucio_normal,perduescarrega))  
  
combinacion_chetada <- melt(combinacion, measure.vars = c("distribucio_normal","perduescarrega"),  
variable.name = "variable", value.name = "valores")  
  
ggplot(combinacion_chetada, aes(x=valores,fill=variable))+
```

```
geom_histogram(color="#e9ecef", position = "identity", bins=50, alpha=0.5)+  
scale_fill_manual(values = c("#ff6e00", "#000000"))+  
theme_ipsum()+  
labs(title = "Normal - Perdues de carrega", x="Mesures")+  
theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))
```

Annex IV: Ponderació coeficients emissors per probabilitats de trencament.

```
#####  
  
#Autor: Víctor Lobato Torres  
  
#Data: 18/03/2021  
  
#Descripció: Ponderacio_Equirepartits  
  
#####  
  
#####  
  
#Netejem el workspace  
  
#i anomenem el fitxer d'entrada i el de sortida  
  
#####  
  
rm(list=ls())  
  
cami<-getwd()  
  
entrada<-paste(cami,"/data/proba1.inp",sep = "")  
sortida<-paste(cami,"/data/proba1.rpt",sep = "")  
sortida2<-paste(cami,"/data/proba1.bin",sep = "")  
  
#####  
  
#Carreguem les llibreries  
  
#####  
  
library(epanet2toolkit) #eina per a utilitzar epanet desde entorn de R  
library(epanetReader) # serveix per a llegir resultats de epanet  
library(ggplot2) #crear grafics  
library(reshape2) #crear titols als grafics  
library(hrbrthemes) #posa colors als grafics  
library(GA) #algoritmes genetics  
library(Rcpp) #llibreria adscrita als algoritmes genetics  
library(MASS) # 2 llibreria adscrita als algoritmes genetics  
library(survival)  
library(tictoc) #llibreria que calcula el temps de simulacio
```

```
#####  
  
#Es hora d'obrir l'epanet  
  
#####  
  
ENopen(entrada,sortida,sortida2) #obre l'epanet  
  
numnodes<-ENgetcount("EN_NODECOUNT") #conta quants nodes tenim al model  
  
numlinks<-ENgetcount("EN_LINKCOUNT") #conta quantes tuberies tenim  
  
indexnode<-seq(1,numnodes) #fa una conta començant per 1 dels nodes per a fer un bucle  
  
indexlink<-seq(1,numlinks) #fa una conta començant per 1 de les tuberies per a fer un bucle  
  
  
  
#####  
  
# Inicialitzar valors de les canonades  
  
#####  
  
idlink<-NULL #per determinar el bucle  
  
diamlink<-NULL #el primer valor de diámetro sigui nul  
  
ruglink<-NULL #rugositat primer valor 0  
  
lonlink<-NULL #longitud primer valor 0  
  
nodilink<-NULL #node inicial de la tuberia  
  
nodflink<-NULL #node final de la tuberia  
  
loninode<-NULL #longitud node inicial  
  
lonfnode<-NULL #Longitud final del node  
  
pressio_estatica<-NULL  
  
perduescarrega<-NULL  
  
longitud_probabilitat<-NULL  
  
  
  
#####  
  
#una vegada inicialitzades les dades podem fer un bucle per tal de llegirles  
  
#####  
  
for (i in 1:numlinks)  
{  
  
  idlink[i]=ENgetlinkid(i)  
  
  diamlink[i]=ENgetlinkvalue(i,"EN_DIAMETER")
```



```
ruglink[i]=ENgetlinkvalue(i,"EN_ROUGHNESS")
lonlink[i]=ENgetlinkvalue(i,"EN_LENGTH")
nodes<-ENgetlinknodes(i)
nodilink[i]<-nodes[1]
nodflink[i]<-nodes[2]
loninode[i]=(lonlink[i]/2)
lonfnode[i]=(lonlink[i]/2)
}

#####
#Inicialitzem les dades dels nodes
#####

idnode<-NULL
demnode<-NULL
basedemnode<-NULL
pnode<-NULL
elevnode<-NULL
patnode<-NULL
coefemis<-NULL
consumnode<-NULL
longnode<-NULL
total<-NULL

#####
#Realitzem el bucle dels nodes per llegir les dades
#####

for (i in 1:numnodes)
{
idnode[i]=ENgetnodeid(i)
demnode[i]=ENgetnodevalue(i,"EN_DEMAND")
basedemnode[i]=ENgetnodevalue(i,"EN_BASEDEMAND")
coefemis[i]=ENgetnodevalue(i,"EN_EMITTER")
elevnode[i]=ENgetnodevalue(i,"EN_ELEVATION")
```

```
    patnode[i]=ENgetnodevalue(i,"EN_PATTERN")
  }

#####

#Realitzem la simulació per despres llegir pressio
#y demanda dels nodes
#####

ENsolveH()

for (i in 1:numnodes)
{
  pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")
  consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")
}

#####

#Creem els data.frame per tal de poder adjudicar longituds als nodes mitjançant la cerca de dos bucles
#####

DadesLinks<-data.frame(indexlink,idlink,lonlink,diamlink,ruglink,nodilink,nodflink)

DadesNodes<-
data.frame(indexnode,idnode,elevnode,basedemnode,patnode,demnode,coefemis,pnode,consumnode)

#####

#Llegim els csv de probabilitats per tal de saber el trencament associat a les canonades
#####

tuberias=read.csv("probabilitat.csv", header=TRUE, sep=";")

#####

#Dintre del .inp existeixen unes quantes tuberies que son ficticies per això fem aquest data
#####

DadesLinksreals=DadesLinks[DadesLinks$lonlink>0.15,]

#####
```

```
#Creem un data on junta les dades i despres adjudiquem les probabilitats
```

```
#####
```

```
DatosTotales = cbind(DadesLinksreals,tuberias)
```

```
for (i in 1:5281)
```

```
{
```

```
  longitud_probabilitat[i]=lonlink[i]*DatosTotales$probabilitat[i]
```

```
  loninode[i]=(longitud_probabilitat[i]/2)
```

```
  lonfnode[i]=(longitud_probabilitat[i]/2)
```

```
}
```

```
#####
```

```
#Creem una variable de coeficient i adjudiquem el valor corresponent amb la longitud atribuïda
```

```
#####
```

```
DadesNodes$lonnode=0
```

```
for (i in 1:numnodes)
```

```
{
```

```
  for (j in 1:5281)
```

```
  {
```

```
    if (DadesNodes$indexnode[i]==DatosTotales$nodilink[j])
```

```
    {
```

```
      DadesNodes$lonnode[i]=DadesNodes$lonnode[i]+DatosTotales$lonlink[j]/2
```

```
    }
```

```
    if (DadesNodes$indexnode[i]==DatosTotales$nodflink[j])
```

```
    {
```

```
      DadesNodes$lonnode[i]=DadesNodes$lonnode[i]+DatosTotales$lonlink[j]/2
```

```
    }
```

```
  }
```

```
}
```

```
coefemis2<-0
```

```
total<-sum(DadesNodes$lonnode)
```

```
for (i in 1:numnodes)
```

```
{
```

```
coefemis2[i]=(DadesNodes$lonnode[i]/total)*10
}

#####

#Creem un bucle per a ponderar les fuites fins el valor objectiu d'aigua
#lliurada
#####

while (abs(consumnode[5123]+84.5)>=1)
{
  if (abs(consumnode[5123]+84.5)>1)
  {

    for (i in seq(1,numnodes-1,1))
    {
      x<-coefemis2[i]
      ENsetnodevalue(i,"EN_EMITTER",x*0.98)
      coefemis2[i]=ENgetnodevalue(i,"EN_EMITTER")
    }
  }

  if (abs(consumnode[5123]+84.5)<1)
  {
    for (i in seq(1,numnodes-1,1))
    {
      x<-coefemis2[i]
      ENsetnodevalue(i,"EN_EMITTER",x*1.0005)
      coefemis2[i]=ENgetnodevalue(i,"EN_EMITTER")
    }
  }

  ENSolveH()

  for(i in 1:numnodes)
  {
    pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")
    consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")
```

```
}  
  
print(x)  
print(consumnode[5123])  
}  
  
for(i in 1:numnodes)  
{  
  pressio_estatica[i]=310.33-elevnode[i]  
  perduescarrega[i]=pressio_estatica[i]-pnode[i]  
}  
  
DadesNodes2<-data.frame(perduescarrega)  
  
ggplot(data=DadesNodes2, aes(x=perduescarrega))+geom_histogram(bins=70,fill="orange",col="black")+  
  labs(title = "Perdues de Carrega trencament", x="Perdua de Carrega (m)")+  
  theme(plot.title = element_text(hjust = 0.5, face = "bold", colour = "cadetblue"))  
  
#Ajustacio amb grafica de distribucio normal  
  
distribucio_normal<- rnorm(5123, 1, 0.23)  
combinacion <- data.frame(cbind(distribucio_normal,perduescarrega))  
combinacion_chetada <- melt(combinacion, measure.vars = c("distribucio_normal","perduescarrega"),  
variable.name = "variable", value.name = "valores")  
ggplot(combinacion_chetada, aes(x=valores,fill=variable))+  
  geom_histogram(color="#e9ecf", position = "identity", bins=50, alpha=0.5)+  
  scale_fill_manual(values = c("#ff6e00", "#000000"))+  
  theme_ipsum()+  
  labs(title = "Normal - Perdues de carrega", x="Mesures")+  
  theme(plot.title = element_text(hjust = 0.5,face = "bold",colour = "cadetblue"))  
  
ENclose()
```

Annex V: Optimització coeficients emissors equirepartits.

```
#####
```

```
#Autor: Víctor Lobato Torres
```

```
#Data: 18/03/2021
```

```
#Descripció: Ponderacio_Equirepartits
```

```
#####
```

```
#####
```

```
#Netejem el workspace
```

```
#i anomenem el fitxer d'entrada i el de sortida
```

```
#####
```

```
rm(list=ls())
```

```
camí<-getwd()
```

```
entrada<-paste(camí,"/data/proba1.inp",sep = "")
```

```
sortida<-paste(camí,"/data/proba1.rpt",sep = "")
```

```
sortida2<-paste(camí,"/data/proba1.bin",sep = "")
```

```
#####
```

```
#Carreguem les llibreries
```

```
#####
```

```
library(epanet2toolkit) #eina per a utilitzar epanet desde entorn de R
```

```
library(epanetReader) # serveix per a llegir resultats de epanet
```

```
library(ggplot2) #crear grafics
```

```
library(reshape2) #crear titols als grafics
```

```
library(hrbrthemes) #posa colors als grafics
```

```
library(GA) #algoritmes genetics
```

```
library(Rcpp) #llibreria adscrita als algoritmes genetics
```

```
library(MASS) # 2 llibreria adscrita als algoritmes genetics
```

```
library(survival)
```

```
library(tictoc) #llibreria que calcula el temps de simulacio
```

```
#####
```

```
#Es hora d'obrir l'epanet
```

```
#####
```

```
ENopen(entrada,sortida,sortida2) #obre l'epanet  
numnodes<-ENgetcount("EN_NODECOUNT") #conta quants nodes tenim al model  
numlinks<-ENgetcount("EN_LINKCOUNT") #conta quantes tuberies tenim  
indexnode<-seq(1,numnodes) #fa una conta començant per 1 dels nodes per a fer un bucle  
indexlink<-seq(1,numlinks) #fa una conta començant per 1 de les tuberies per a fer un bucle
```

```
#####
```

```
#Inicialitzem les dades dels nodes
```

```
#####
```

```
idnode<-NULL #Identitat dels nodes  
coefemis<-NULL #Coeficients emissors dels nodes  
consumnode<-NULL #Consum dels nodes  
pnode<-NULL #Pressio dinamica dels nodes  
patnode<-NULL #Pattern dels nodes  
elevnode<-NULL #Elevacio dels nodes corresponents  
pressio_estatica<-NULL #Pressio estatica dels nodes  
perduescarrega<-NULL #Perdues de carrega assolides a la ponderacio
```

```
#####
```

```
#Realitzem el bucle dels nodes per llegir les dades
```

```
#####
```

```
for (i in 1:numnodes)  
{  
  idnode[i]=ENgetnodeid(i) #Identificacio del node  
  coefemis[i]=ENgetnodevalue(i,"EN_EMITTER") #Coeficient Emissors  
  patnode[i]=ENgetnodevalue(i,"EN_PATTERN") #Pattern del node  
  elevnode[i]=ENgetnodevalue(i,"EN_ELEVATION") #Elevacio del node  
}
```

```
#####
```

```
#Realitzem la simulació per despres llegir pressio
```

```
#y demanda dels nodes

#####

ENsolveH() #Resolucio hidraulica

for (i in 1:numnodes)

{

  pnode[i]=ENgetnodevalue(i,"EN_PRESSURE") #Llegim la pressio dinamica del node

  consumnode[i]=ENgetnodevalue(i,"EN_DEMAND") #Llegim el consum del node

}

#####

#Implementem el bucle per augmentar les fuites

#Coeficients emissors fins el valor de consum del

#Diposit estipulat

#####

consum_objectiu=-84.5 #Consum del diposit que volem assolir

while (consumnode[5123]>consum_objectiu) #Bucle while per establir dita condicio de consum

{

  for (i in seq(1,numnodes-1,1))

  {

    Llegir_primer_coef<-ENgetnodevalue(1,"EN_EMITTER")

    ENsetnodevalue(i,"EN_EMITTER",Llegir_primer_coef+0.000007)

    coefemis[i]=ENgetnodevalue(i,"EN_EMITTER")

  }

  ENsolveH()

  for (i in 1:numnodes)

  {

    pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")

    consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")

  }

  print(Llegir_primer_coef)

  print(consumnode[5123])

}
```



```
for(i in 1:numnodes)
{
  pressio_estatica[i]=310.33-elevnode[i] #Llegim la pressio estatica una vegada adjudicats els coeficients
  perduescarrega[i]=pressio_estatica[i]-pnode[i] #Llegim les perdues de carrega una vegada adjudicats els nodes
}
```

```
ENclose()
```

```
Num=2017
```

```
#####
```

```
#ALGORITMES GENETICS
```

```
#####
```

```
f<-function(z){ #Funcio objectiu en funcio de z (variable optimitzacio)
```

```
diposit=read.csv("pattern_diposit_elevacio.csv", header=TRUE, sep=";")
```

```
demanda=read.csv("pattern_demanda_nodes.csv", header=TRUE, sep=";")
```

```
Vector_pattern_elevacio<- diposit$DIPOSIT[1:288]
```

```
Vector_pattern_demanda<- demanda$DEMANDA
```

```
Nodeestudi=read.csv("simulacio_empresa.csv",header = TRUE,sep=";")
```

```
ENopen(entrada,sortida,sortida2) #obre l'epanet
```

```
#####
```

```
#Posem l'elevacio del diposit
```

```
#####
```

```
ENsetpattern(1,Vector_pattern_elevacio)
```

```
ENsetnodevalue(5123,"EN_PATTERN",1)
```

```
ENsetnodevalue(5123,"EN_ELEVATION",1)
```

```
#####
```

```
#Duracio d'una semana amb 5 min de report
```

```
#####
```

```
ENsettimeparam("EN_DURATION",604800)
```

```
ENsettimeparam("EN_REPORTSTEP",300)
```

```
ENgettimeparam("EN_REPORTSTEP")

#####

#Posem la demanda dels nodes corresponents

#####

ENsetpattern(2,Vector_pattern_demanda)

for (i in seq(1,numnodes-1,1))

{

  ENsetnodevalue(i,"EN_PATTERN",2)

}

#####

#Posem el coeficient de la iteració

#####

for (i in seq(1,numnodes-1,1))

{

  ENsetnodevalue(i,"EN_EMITTER",z)

}

#####

#Simulem

#####

ENSolveH()

ENSolveQ()

ENreport()

ENclose()

resul<-read.rpt(sortida)

NodeSimulats<-resul$nodeResults

Node188008=NodeSimulats[NodeSimulats$ID=="188008",]

Node235846=NodeSimulats[NodeSimulats$ID=="235846",]

Node518=NodeSimulats[NodeSimulats$ID=="518",]
```

```
Funcio_Objje<-      (sum((Node188008$Pressure-Nodeestudi$Mig)^2))/(Num)+(sum((Node235846$Pressure-  
Nodeestudi$Mig.1)^2))/(Num)+(sum((Node518$Pressure-Nodeestudi$Mig.2)^2))/(Num)  
  
print(Funcio_Objje)  
  
return(Funcio_Objje)  
  
}
```

```
fitness<- function(z)-f(z) #Funcio Fitness del problema d'optimitzacio
```

```
GA=ga(type = c("real-valued"), fitness = fitness, lower = 0.0004, upper = 0.0005,
```

```
  maxiter = 2) #Funcio dels algoritmes genetics
```

```
summary(GA) #Resum dels algoritmes genetics
```

```
plot(GA) #Plot dels algoritmes genètics
```

Annex VI: Optimització coeficients emissors per longituds.

```
#####  
  
#Autor: Víctor Lobato Torres  
  
#Data: 18/03/2021  
  
#Descripció: Ponderacio de Longituds  
  
#####  
  
#####  
  
#Netejem el workspace  
  
#i anomenem el fitxer d'entrada i el de sortida  
  
#####  
  
rm(list=ls())  
  
cami<-getwd()  
  
entrada<-paste(cami,"/data/proba1.inp",sep = "")  
  
sortida<-paste(cami,"/data/proba1.rpt",sep = "")  
  
sortida2<-paste(cami,"/data/proba1.bin",sep = "")  
  
#####  
  
#Carreguem les llibreries  
  
#####  
  
library(epanet2toolkit) #eina per a utilitzar epanet desde entorn de R  
  
library(epanetReader) # serveix per a llegir resultats de epanet  
  
library(ggplot2) #crear grafics  
  
library(reshape2) #crear titols als grafics  
  
library(hrbrthemes) #posa colors als grafics  
  
library(GA) #algoritmes genetics  
  
library(Rcpp) #llibreria adscrita als algoritmes genetics  
  
library(MASS) # 2 llibreria adscrita als algoritmes genetics  
  
library(survival)  
  
library(tictoc) #llibreria que calcula el temps de simulacio
```

```
#####  
  
#Es hora d'obrir l'epanet  
  
#####  
  
ENopen(entrada,sortida,sortida2) #obre l'epanet  
  
numnodes<-ENgetcount("EN_NODECOUNT") #conta quants nodes tenim al model  
  
numlinks<-ENgetcount("EN_LINKCOUNT") #conta quantes tuberies tenim  
  
indexnode<-seq(1,numnodes) #fa una conta començant per 1 dels nodes per a fer un bucle  
  
indexlink<-seq(1,numlinks) #fa una conta començant per 1 de les tuberies per a fer un bucle  
  
#####  
  
# Inicialitzar valors de les canonades  
  
#####  
  
idlink<-NULL #per determinar el bucle  
  
lonlink<-NULL #longitud primer valor 0  
  
nodilink<-NULL #node inicial de la tuberia  
  
nodflink<-NULL #node final de la tuberia  
  
loninode<-NULL #longitud node inicial  
  
lonfnode<-NULL #Longitud final del node  
  
pressio_estatica<-NULL #pressio estatica, coef=0  
  
perduescarrega<-NULL #Perdues de carrega del model  
  
#####  
  
#una vegada inicialitzades les dades podem fer un bucle per tal de llegirles  
  
#####  
  
for (i in 1:numlinks)  
{  
  idlink[i]=ENgetlinkid(i) #adjudiquem una identitat a les canonades  
  lonlink[i]=ENgetlinkvalue(i,"EN_LENGTH") #Llegim longituds  
  nodes<-ENgetlinknodes(i) #identitat pels nodes  
  nodilink[i]<-nodes[1] #Identifiquem el node inicial de la canonada  
  nodflink[i]<-nodes[2] #Identifiquem el node final de la canonada  
  loninode[i]=(lonlink[i]/2) #Adjudiquem longitud inicial al node
```

```
lonfnode[i]=(lonlink[i]/2) #Adjudiquem longitud final al node
}

#####

#Inicialitzem les dades dels nodes

#####

idnode<-NULL #identitat del node
demnode<-NULL #demanda del node
basedemnode<-NULL #demanda base del node
pnode<-NULL #pressio dinamica al node
elevnode<-NULL #Elevacio del node
coefemis<-NULL #coeficient emissor del node
consumnode<-NULL #consum del node
total<-NULL #Total de metres de la xarxa

#####

#Realitzem el bucle dels nodes per llegir les dades

#####

for (i in 1:numnodes)
{
  idnode[i]=ENgetnodeid(i) #Identifiquem el node
  demnode[i]=ENgetnodevalue(i,"EN_DEMAND") #Llegim la seva demanda
  basedemnode[i]=ENgetnodevalue(i,"EN_BASEDEMAND") #Llegim la demanda base
  coefemis[i]=ENgetnodevalue(i,"EN_EMITTER") #Llegim el coeficient emissor
  elevnode[i]=ENgetnodevalue(i,"EN_ELEVATION") #Elevacio del node corresponent
}

#####

#Realitzem la simulació per despres llegir pressio
#y demanda dels nodes

#####

ENSolveH() #Resolucio hidraulica
```

```
for (i in 1:numnodes)
{
  pnode[i]=ENgetnodevalue(i,"EN_PRESSURE") #Llegim la pressio dinamica una vegada simulat la xarxa
  consumnode[i]=ENgetnodevalue(i,"EN_DEMAND") #Llegim el consum del node una vegada simulat la xarxa
}

#####

#Creem els data.frame per tal de poder adjudicar longituds als nodes mitjançant la cerca de dos bucles

#####

DadesLinks<-data.frame(indexlink,idlink,lonlink,nodilink,nodflink) #Data.frame de les canonades
DadesNodes<-data.frame(indexnode,idnode,elevnode,basedemnode,demnode,coefemis,pnode,consumnode)
#Data.Frame dels nodes
DadesNodes$lonnode=0 #Adjudiquem el valor total de la xarxa
for (i in 1:numnodes) #Doble bucle per saber el valor de longitud total de la xarxa.
{
  for (j in 1:numlinks)
  {
    if (DadesNodes$indexnode[i]==DadesLinks$nodilink[j])
    {
      DadesNodes$lonnode[i]=DadesNodes$lonnode[i]+DadesLinks$lonlink[j]/2
    }
    if (DadesNodes$indexnode[i]==DadesLinks$nodflink[j])
    {
      DadesNodes$lonnode[i]=DadesNodes$lonnode[i]+DadesLinks$lonlink[j]/2
    }
  }
}

#####

#Creem una variable de coeficient i adjudiquem el valor corresponent amb la longitud atribuïda

#####

coefemis2<-0 #Nova variable coeficient emissor per poder llegir els primers
total<-sum(DadesNodes$lonnode) #Total longitudinal de la xarxa
for (i in 1:numnodes)
```

```

{
  coefemis2[i]=(DadesNodes$lonnode[i]/total)*10 #Adjudiquem el coeficient emissor en relacio a les longituds
}

#####

#Creem un bucle per a ponderar les fuites fins el valor objectiu d'aigua

#lliurada

#####

consum_objectiu=84.5 #Consum del diposit que volem assolir

while (abs(consumnode[5123]+consum_objectiu)>=1) #Bucle while per imposar que es compleixi el consum
modificant coefemis2
{
  if (abs(consumnode[5123]+consum_objectiu)>1)
  {
    for (i in seq(1,numnodes-1,1))
    {
      Vector_Coeficients<-coefemis2[i]
      ENsetnodevalue(i,"EN_EMITTER",Vector_Coeficients*0.98)
      coefemis2[i]=ENgetnodevalue(i,"EN_EMITTER")
    }
  }
  if (abs(consumnode[5123]+consum_objectiu)<1)
  {
    for (i in seq(1,numnodes-1,1))
    {
      Vector_Coeficients<-coefemis2[i]
      ENsetnodevalue(i,"EN_EMITTER",Vector_Coeficients*1.0005)
      coefemis2[i]=ENgetnodevalue(i,"EN_EMITTER")
    }
  }
  ENSolveH()
  for(i in 1:numnodes)
  {

```



```
    pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")
    consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")
}
print(Vector_Coeficients)
print(consumnode[5123])
}

for(i in 1:numnodes) #Llegim la pressio estatica i les perdues de carrega
{
    pressio_estatica[i]=310.33-elevnode[i]
    perduescarrega[i]=pressio_estatica[i]-pnode[i]
}

ENclose()

Num=2017

#####
#ALGORITMES GENETICS
#####

f<-function(z){ #Funcio objectiu en funcio de z (variable optimitzacio)

    diposit=read.csv("pattern_diposit_elevacio.csv", header=TRUE, sep=";")
    demanda=read.csv("pattern_demanda_nodes.csv", header=TRUE, sep=";")
    Vector_pattern_elevacio<- diposit$DIPOSIT[1:288]
    Vector_pattern_demanda<- demanda$DEMANDA
    Nodeestudi=read.csv("simulacio_empresa.csv",header = TRUE,sep=";")
    ENopen(entrada,sortida,sortida2) #obre l'epanet

    #####
    #Posem l'elevacio del diposit
    #####

    ENsetpattern(1,Vector_pattern_elevacio)
    ENsetnodevalue(5123,"EN_PATTERN",1)
```

```
ENsetnodevalue(5123,"EN_ELEVATION",1)
#####
#Duracio d'una semana amb 5 min de report
#####
ENsettimeparam("EN_DURATION",604800)
ENsettimeparam("EN_REPORTSTEP",300)
ENgettimeparam("EN_REPORTSTEP")

#####
#Posem la demanda dels nodes corresponents
#####
ENsetpattern(2,Vector_pattern_demanda)
for (i in seq(1,numnodes-1,1))
{
  ENsetnodevalue(i,"EN_PATTERN",2)
}
#####
#Posem el coeficient de la iteració
#####
coefe_long=z*coefemis2
for (i in seq(1,numnodes-1,1))
{
  ENsetnodevalue(i,"EN_EMITTER",coefe_long[i])
}
#####
#Simulem
#####
ENSolveH()
ENSolveQ()
ENreport()
ENclose()

resul<-read.rpt(sortida)
NodeSimulats<-resul$nodeResults
Node188008=NodeSimulats[NodeSimulats$ID=="188008",]
```

```
Node235846=NodeSimulats[NodeSimulats$ID=="235846",]
```

```
Node518=NodeSimulats[NodeSimulats$ID=="518",]
```

```
Funcio_Objje<-      (sum((Node188008$Pressure-Nodeestudi$Mig)^2))/(Num)+(sum((Node235846$Pressure-  
Nodeestudi$Mig.1)^2))/(Num)+(sum((Node518$Pressure-Nodeestudi$Mig.2)^2))/(Num)
```

```
print(Funcio_Objje)
```

```
return(Funcio_Objje)
```

```
}
```

```
fitness<- function(z)-f(z) #Funcio Fitness
```

```
GA=ga(type = c("real-valued"), fitness = fitness, lower = 0.85, upper = 1.15 ,
```

```
maxiter = 2) #Algoritme genetic
```

```
summary(GA) #resum del algoritme
```

```
plot(GA) #Grafica del algoritme
```

Annex VII: Optimització coeficients emissors per probabilitats de trencament.

```
#####  
  
#Autor: Víctor Lobato Torres  
  
#Data: 18/03/2021  
  
#Descripció: Ponderacio_Equirepartits  
  
#####  
  
#####  
  
#Netejem el workspace  
  
#i anomenem el fitxer d'entrada i el de sortida  
  
#####  
  
rm(list=ls())  
  
cami<-getwd()  
  
entrada<-paste(cami,"/data/proba1.inp",sep = "")  
sortida<-paste(cami,"/data/proba1.rpt",sep = "")  
sortida2<-paste(cami,"/data/proba1.bin",sep = "")  
  
#####  
  
#Carreguem les llibreries  
  
#####  
  
library(epanet2toolkit) #eina per a utilitzar epanet desde entorn de R  
library(epanetReader) # serveix per a llegir resultats de epanet  
library(ggplot2) #crear grafics  
library(reshape2) #crear titols als grafics  
library(hrbrthemes) #posa colors als grafics  
library(GA) #algoritmes genetics  
library(Rcpp) #llibreria adscrita als algoritmes genetics  
library(MASS) # 2 llibreria adscrita als algoritmes genetics  
library(survival)  
library(tictoc) #llibreria que calcula el temps de simulacio
```

```
#####  
  
#Es hora d'obrir l'epanet  
  
#####  
  
ENopen(entrada,sortida,sortida2) #obre l'epanet  
  
numnodes<-ENgetcount("EN_NODECOUNT") #conta quants nodes tenim al model  
  
numlinks<-ENgetcount("EN_LINKCOUNT") #conta quantes tuberies tenim  
  
indexnode<-seq(1,numnodes) #fa una conta començant per 1 dels nodes per a fer un bucle  
  
indexlink<-seq(1,numlinks) #fa una conta començant per 1 de les tuberies per a fer un bucle  
  
  
  
#####  
  
# Inicialitzar valors de les canonades  
  
#####  
  
idlink<-NULL #per determinar el bucle  
  
diamlink<-NULL #el primer valor de diametre sigui nul  
  
ruglink<-NULL #rugositat primer valor 0  
  
lonlink<-NULL #longitud primer valor 0  
  
nodilink<-NULL #node inicial de la tuberia  
  
nodflink<-NULL #node final de la tuberia  
  
loninode<-NULL #longitud node inicial  
  
lonfnode<-NULL #Longitud final del node  
  
pressio_estatica<-NULL  
  
perduescarrega<-NULL  
  
longitud_probabilitat<-NULL  
  
  
  
#####  
  
#una vegada inicialitzades les dades podem fer un bucle per tal de llegirles  
  
#####  
  
for (i in 1:numlinks)  
{  
  idlink[i]=ENgetlinkid(i)  
  diamlink[i]=ENgetlinkvalue(i,"EN_DIAMETER")
```

```
ruglink[i]=ENgetlinkvalue(i,"EN_ROUGHNESS")
lonlink[i]=ENgetlinkvalue(i,"EN_LENGTH")
nodes<-ENgetlinknodes(i)
nodilink[i]<-nodes[1]
nodflink[i]<-nodes[2]
loninode[i]=(lonlink[i]/2)
lonfnode[i]=(lonlink[i]/2)
}

#####
#Inicialitzem les dades dels nodes
#####

idnode<-NULL
demnode<-NULL
basedemnode<-NULL
pnode<-NULL
elevnode<-NULL
patnode<-NULL
coefemis<-NULL
consumnode<-NULL
longnode<-NULL
total<-NULL

#####
#Realitzem el bucle dels nodes per llegir les dades
#####

for (i in 1:numnodes)
{
idnode[i]=ENgetnodeid(i)
demnode[i]=ENgetnodevalue(i,"EN_DEMAND")
basedemnode[i]=ENgetnodevalue(i,"EN_BASEDEMAND")
coefemis[i]=ENgetnodevalue(i,"EN_EMITTER")
elevnode[i]=ENgetnodevalue(i,"EN_ELEVATION")
```

```
    patnode[i]=ENgetnodevalue(i,"EN_PATTERN")
  }

#####

#Realitzem la simulació per despres llegir pressio
#y demanda dels nodes
#####

ENsolveH()

for (i in 1:numnodes)
{
  pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")
  consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")
}

#####

#Creem els data.frame per tal de poder adjudicar longituds als nodes mitjançant la cerca de dos bucles
#####

DadesLinks<-data.frame(indexlink,idlink,lonlink,diamlink,ruglink,nodilink,nodflink)

DadesNodes<-
data.frame(indexnode,idnode,elevnode,basedemnode,patnode,demnode,coefemis,pnode,consumnode)

#####

#Llegim els csv de probabilitats per tal de saber el trencament associat a les canonades
#####

tuberias=read.csv("probabilitat.csv", header=TRUE, sep=";")

#####

#Dintre del .inp existeixen unes quantes tuberies que son ficticies per això fem aquest data
#####

DadesLinksreals=DadesLinks[DadesLinks$lonlink>0.15,]

#####
```

```
#Creem un data on junta les dades i despres adjudiquem les probabilitats
```

```
#####
```

```
DatosTotales = cbind(DadesLinksreals,tuberias)
```

```
for (i in 1:5281)
```

```
{
```

```
  longitud_probabilitat[i]=lonlink[i]*DatosTotales$probabilitat[i]
```

```
  loninode[i]=(longitud_probabilitat[i]/2)
```

```
  lonfnode[i]=(longitud_probabilitat[i]/2)
```

```
}
```

```
#####
```

```
#Creem una variable de coeficient i adjudiquem el valor corresponent amb la longitud atribuïda
```

```
#####
```

```
DadesNodes$lonnode=0
```

```
for (i in 1:numnodes)
```

```
{
```

```
  for (j in 1:5281)
```

```
  {
```

```
    if (DadesNodes$indexnode[i]==DatosTotales$nodilink[j])
```

```
    {
```

```
      DadesNodes$lonnode[i]=DadesNodes$lonnode[i]+DatosTotales$lonlink[j]/2
```

```
    }
```

```
    if (DadesNodes$indexnode[i]==DatosTotales$nodflink[j])
```

```
    {
```

```
      DadesNodes$lonnode[i]=DadesNodes$lonnode[i]+DatosTotales$lonlink[j]/2
```

```
    }
```

```
  }
```

```
}
```

```
coefemis2<-0
```

```
total<-sum(DadesNodes$lonnode)
```

```
for (i in 1:numnodes)
```

```
{
```



```
coefemis2[i]=(DadesNodes$lonnode[i]/total)*10
}

#####

#Creem un bucle per a ponderar les fuites fins el valor objectiu d'aigua
#lliurada
#####

while (abs(consumnode[5123]+84.5)>=1)
{
  if (abs(consumnode[5123]+84.5)>1)
  {

    for (i in seq(1,numnodes-1,1))
    {
      x<-coefemis2[i]
      ENsetnodevalue(i,"EN_EMITTER",x*0.98)
      coefemis2[i]=ENgetnodevalue(i,"EN_EMITTER")
    }
  }

  if (abs(consumnode[5123]+84.5)<1)
  {
    for (i in seq(1,numnodes-1,1))
    {
      x<-coefemis2[i]
      ENsetnodevalue(i,"EN_EMITTER",x*1.0005)
      coefemis2[i]=ENgetnodevalue(i,"EN_EMITTER")
    }
  }

  ENSolveH()

  for(i in 1:numnodes)
  {
    pnode[i]=ENgetnodevalue(i,"EN_PRESSURE")
    consumnode[i]=ENgetnodevalue(i,"EN_DEMAND")
```

```
}  
print(x)  
print(consumnode[5123])  
}  
  
for(i in 1:numnodes)  
{  
  pressio_estatica[i]=310.33-elevnode[i]  
  perduescarrega[i]=pressio_estatica[i]-pnode[i]  
}  
  
ENclose()  
  
Num=2017  
  
#####  
#ALGORITMES GENETICS  
#####  
  
f<-function(z){ #Funcio objectiu en funcio de z (variable optimitzacio)  
  
  diposit=read.csv("pattern_diposit_elevacio.csv", header=TRUE, sep=";")  
  demanda=read.csv("pattern_demanda_nodes.csv", header=TRUE, sep=";")  
  Vector_pattern_elevacio<- diposit$DIPOSIT[1:288]  
  Vector_pattern_demanda<- demanda$DEMANDA  
  Nodeestudi=read.csv("simulacio_empresa.csv",header = TRUE,sep=";")  
  ENopen(entrada,sortida,sortida2) #obre l'epanet  
  
  #####  
  #Posem l'elevacio del diposit  
  #####  
  ENsetpattern(1,Vector_pattern_elevacio)  
  ENsetnodevalue(5123,"EN_PATTERN",1)  
  ENsetnodevalue(5123,"EN_ELEVATION",1)  
  #####
```

```
#Duracio d'una semana amb 5 min de report
#####

ENsettimeparam("EN_DURATION",604800)
ENsettimeparam("EN_REPORTSTEP",300)
ENgettimeparam("EN_REPORTSTEP")

#####

#Posem la demanda dels nodes corresponents
#####

ENsetpattern(2,Vector_pattern_demanda)
for (i in seq(1,numnodes-1,1))
{
  ENsetnodevalue(i,"EN_PATTERN",2)
}
#####

#Posem el coeficient de la iteració
#####

for (i in seq(1,numnodes-1,1))
{
  ENsetnodevalue(i,"EN_EMITTER",z)
}
#####

#Simulem
#####

ENSolveH()
ENSolveQ()
ENreport()
ENClose()

resul<-read.rpt(sortida)
NodeSimulats<-resul$nodeResults
Node188008=NodeSimulats[NodeSimulats$ID=="188008",]
Node235846=NodeSimulats[NodeSimulats$ID=="235846",]
Node518=NodeSimulats[NodeSimulats$ID=="518",]
```

```
Funcio_Objje<- (sum((Node188008$Pressure-Nodeestudi$Mig)^2))/(Num)+(sum((Node235846$Pressure-  
Nodeestudi$Mig.1)^2))/(Num)+(sum((Node518$Pressure-Nodeestudi$Mig.2)^2))/(Num)  
  
print(Funcio_Objje)  
  
return(Funcio_Objje)  
  
}
```

```
fitness<- function(z)-f(z) #Funcio Fitness del problema d'optimitzacio
```

```
GA=ga(type = c("real-valued"), fitness = fitness, lower = 0.0004, upper = 0.0005,
```

```
  maxiter = 2) #Funcio dels algoritmes genetics
```

```
summary(GA) #Resum dels algoritmes genetics
```

```
plot(GA) #Plot dels algoritmes genetics
```