

Anexo

Código en R para resolver una red de Jackson cerrada

```
1 library(igraph)
2 library(ggplot2)
3 library(cowplot)
4
5 rendimiento <- function(nodos, salidas, servidores, enrutados, m){
6
7
8   # resolver el sistema de las ecuaciones de trafico para
9   # calcular las tasas de llegadas relativas
10  # usando la funcion solve
11  llegadas_relativas <- c(1, solve(diag(1, nodos-1) - t(enrutados[-1, -1]), enrutados[1, -1]))
12  print(llegadas_relativas)
13  # crear las matrices con las sucesiones de L y W (L empieza en
14  # 0 y W en 1)
15  L <- matrix(0, nrow=m+1, ncol=nodos)
16  W <- matrix(0, nrow=m, ncol=nodos)
17
18  # iterar
19  for(i in 1:m) {
20
21    # calcular W_i a partir de L_{i-1}
22    W[i,] <- 1/salidas + L[i,]/(servidores*salidas)
23
24    # calcular L_i a partir de W_i
25    L[i+1,] <- i*llegadas_relativas*W[i,]/(sum(llegadas_relativas*W[i,]))
26  }
27
28  # quitar la fila 0 de L
29  L <- L[-1,]
30
31  # calcular las tasas de llegadas reales
32  llegadas_reales <- L/W
33
34  # calcular los porcentajes de utilizacion (U)
35  U <- matrix(0, nrow=m, ncol=nodos)
```

```

34  for (i in 1:m) {
35    U[i,]<-100*llegadas_reales[i,]/(servidores*salidas)
36  }
37  # poner etiquetas a las filas y columnas de las tablas de las
    cuatro medidas de rendimiento (L, W, U y las tasas de
    llegadas reales)
38  colnames(L)<-unlist(lapply(1:nodos, function(i){paste("L_",i,sep
    =""))}))
39  colnames(W)<-unlist(lapply(1:nodos, function(i){paste("W_",i,sep
    =""))}))
40  colnames(U)<-unlist(lapply(1:nodos, function(i){paste("U_",i,sep
    =""))}))
41  colnames(llegadas_reales)<-unlist(lapply(1:nodos, function(i){
    paste("lambda_",i,sep=""))}))
42  rownames(L)<-rownames(W)<-rownames(U)<-rownames(llegadas_reales
    )<-unlist(lapply(1:m, function(i){paste("m=",i,sep=""))}))
43
44  # sacar las tablas de las cuatro medidas de rendimiento en la
    consola
45  print(L)
46  print(W)
47  print(U)
48  print(llegadas_reales)
49
50  # sacar el grafo en la consola
51  # usando la funcion graph.adjacency
52  grafo<-graph.adjacency(enrutados,mode="directed", weighted =
    T)
53  window(x)
54  plot.igraph(grafo,edge.curve=0.5, edge.label = E(grafo)$weight)
55  # crear los graficos de las cuatro medidas de rendimiento
56  l<-lapply(list(L,W,U,llegadas_reales), function(i){return(data.
    frame("m"=rep(1:m,times=nodos),"i"=as.vector(i),
57  "nodo"=factor(rep(1:nodos,each=m))))})
58  labs<-c("L (clientes)", "W (tiempo)", "U (%)", "llegadas (clientes
    /u.t.)")
59  plot_l<-lapply(1:4, function(j){return(ggplot(l[[j]], aes(x=m,y=i
    ,group=nodo, color=nodo))+geom_line()+labs(y=labs[j]))})
60
61  # sacar los cuatro graficos en la consola
62  plot_grid(plot_l[[1]], plot_l[[2]], plot_l[[3]], plot_l[[4]], ncol
    = 2, byrow = T)
63 }

```

Ejemplo de ejecución del programa

```
1 rendimiento(nodos=5,  
2     salidas=c(1,4,5,8,10),  
3     servidores=c(4,1,1,1,1),  
4     enrutados=matrix(c(0,0.25,0.25,0.25,0.25,0.75,0.25,0,  
5         0,0,0.8,0,0.2,0,0,0.875,0,0,0.125,0,0.9,0,0,0,0.1)  
6         ,nrow = 5,byrow = T),  
7     m=200)
```