Bonus

library("Matrix")  
library("igraph")

##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

library("rdist")  
  
P = 3  
N = 10  
W = matrix(sample(1:5,N^2,replace = T), nrow=N, ncol=N, byrow=T)  
for (i in 1:N){  
 W[i,i] = 0  
}  
Amat = matrix(0, nrow=(1 + N + (N^2) + (N^2)), ncol=((N^2)+(N^3)))  
  
Amat[1,seq(1, by=(N+1), length.out=(N))] = 1  
  
column = 1  
for(i in 2:(N+1)){  
 Amat[i,seq(column, by=(1), length.out=(N))] = 1  
 column = column +N  
}  
  
count = 1  
column = 1  
for(i in (N + 2):(1 + N + (N^2))){  
 Amat[i,column] = -1  
 if (column != count)  
 {  
 Amat[i,count] = 1  
 }  
 if (count %% N == 0){  
 column = column + N + 1  
 }  
 count = count + 1  
}  
  
I = 1  
K = 1  
for(i in (N + (N^2) + 2):(1 + N + (N^2) + (N^2))){  
   
 Xijk = array(0,dim=c(N,N,N))  
 for(j in 1:N){  
 Xijk[I,K,j] = Xijk[I,K,j] + 1  
 }  
 for(j in 1:N){  
 Xijk[I,j,K] = Xijk[I,j,K] - 1  
 }  
   
 Xijk.vector = integer(N^3)  
 vector.position = 1  
 for (x in 1:N){  
 for (y in 1:N){  
 for (z in 1:N){  
 Xijk.vector[vector.position] = Xijk[x,y,z]  
 vector.position = vector.position + 1  
 }  
 }  
 }  
   
 Yij = array(0,dim=c(N,N))  
 for(j in 1:N){  
 Yij[I,K] = Yij[I,K] - W[I,j]  
 }  
 for(j in 1:N){  
 Yij[j,K] = Yij[j,K] + W[I,j]  
 }  
   
 Yij.vector = integer(N^2)  
 vector.position = 1  
 for (x in 1:N){  
 for (y in 1:N){  
 Yij.vector[vector.position] = Yij[x,y]  
 vector.position = vector.position + 1  
 }  
 }  
   
 Amat[i,] = c(Yij.vector, Xijk.vector)  
   
 if (K == N){  
 I = I + 1  
 K = 1  
 }  
 else {  
 K = K + 1  
 }  
}  
  
image(Matrix(Amat))