**A** board (Ising!) [Le Monde mathematical puzzle](https://xianblog.wordpress.com/2011/09/03/le-monde-puzzle-website/) in the optimisation mode, again:

*On a 7×7 board, what is the maximal number of locations that one can occupy when imposing at least two empty neighbours ?*

Which I tried to solve by brute force and simulated annealing (what else?!), first defining a target

targ=function(tabz){

sum(tabz[-c(1,9),-c(1,9)]-1.2\*(tabz[-c(1,9),-c(1,9)]\*tabz[-c(8,9),-c(1,9)]

+tabz[-c(1,9),-c(1,9)]\*tabz[-c(1,2),-c(1,9)]

+tabz[-c(1,9),-c(1,9)]\*tabz[-c(1,9),-c(8,9)]

+tabz[-c(1,9),-c(1,9)]\*tabz[-c(1,9),-c(1,2)]>2))}

on a 9×9 board where I penalise prohibited configuration by a factor 1.2 (a wee bit more than empty nodes). The perimeter of the 9×9 board is filled with ones and never actualised. (In the above convoluted products, the goal is to count how many neighbours of the entries equal to one are also equal to one. More than 2 is penalised.) The simulated annealing move is then updating the 9×9 grid gridz:

temp=1

maxarg=curarg=targ(gridz)

for (t in 1:1e3){

for (v in 1:1e4){

i=sample(2:8,1);j=sample(2:8,1)

newgrid=gridz;newgrid[i,j]=1-gridz[i,j]

newarg=targ(newgrid)

if (log(runif(1))

and calls to the procedure always return 28 entries as the optimum, as in

[,1] [,2] [,3] [,4] [,5] [,6] [,7]

[1,] 1 0 1 0 1 0 1

[2,] 0 1 1 0 1 1 0

[3,] 1 1 0 1 0 1 1

[4,] 0 0 1 0 1 0 0

[5,] 1 1 0 1 0 1 1

[6,] 0 1 1 0 1 1 0

[7,] 1 0 1 0 1 0 1

As it happens, I had misread the wording of the original puzzle, which considered a dynamic placement of the units on the board, one at a time with two free neighbours imposed.