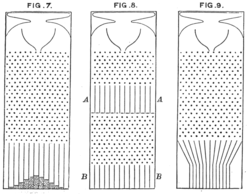
.



**S**ince [Galton’s quincunx](https://xianblog.wordpress.com/2017/12/03/a-quincunx-on-nbc/) has fascinated me since the (early) days when I saw [a model of it](http://www.aakkozzll.com/docs/order.htm) as a teenager in an industry museum near [Birmingham](https://xianblog.wordpress.com/2018/03/23/back-to-wales-54th-gregynog-statistical-conference/), I jumped on [the challenge](https://www.quantamagazine.org/how-randomness-can-arise-from-determinism-20191014/) to build an uneven nail version where the probabilities to end up in one of the boxes were not the Binomial ones. For instance,  producing a uniform distribution with the maximum number of nails with probability ½ to turn right. And I obviously chose to try simulated annealing to figure out the probabilities, facing as usual the unpleasant task of setting the objective function, calibrating the moves and the temperature schedule. Plus, less usually, a choice of the space where the optimisation takes place, i.e., deciding on a common denominator for the (rational) probabilities. Should it be 2⁸?! Or more (since the solution with two levels also involves 1/3)? Using the functions

evol<-function(P){

Q=matrix(0,7,8)

Q[1,1]=P[1,1];Q[1,2]=1-P[1,1]

for (i in 2:7){

Q[i,1]=Q[i-1,1]\*P[i,1]

for (j in 2:i)

Q[i,j]=Q[i-1,j-1]\*(1-P[i,j-1])+Q[i-1,j]\*P[i,j]

Q[i,i+1]=Q[i-1,i]\*(1-P[i,i])

Q[i,]=Q[i,]/sum(Q[i,])}

return(Q)}

and

temper<-function(T=1e3){

bestar=tarP=targ(P<-matrix(1/2,7,7))

temp=.01

while (sum(abs(8\*evol(R<-P)[7,]-1))>.01){

for (i in 2:7)

R[i,sample(rep(1:i,2),1)]=sample(0:deno,1)/deno

if (log(runif(1))/temp

I first tried running my simulated annealing code with a target function like

targ<-function(P)(1+.1\*sum(!(2\*P==1)))\*sum(abs(8\*evol(P)[7,]-1))

where P is the 7×7 lower triangular matrix of nail probabilities, all with a 2⁸ denominator, reaching

**60**

126 35

107 81 20

104 71 22 0

126 44 26 **69** 14

**61** 123 113 92 91 38

109 **60** 7 19 44 74 50

for 128P. With  four entries close to 64, i.e. ½’s. Reducing the denominator to 16 produced once

**8**

12 1

13 11 3

16  7  6   2

14 13 16 15 0

15  15  2  7   7  4

**8**   0    **8**   9   **8**  16  **8**

as 16P, with five ½’s (8). But none of the solutions had exactly a uniform probability of 1/8 to reach all endpoints. Success (with exact 1/8’s and a denominator of 4) was met with the new target

(1+,1\*sum(!(2\*P==1)))\*(.01+sum(!(8\*evol(P)[7,]==1)))

imposing precisely 1/8 on the final line. With a solution with 11 ½’s

**0.5**

1.0 0.0

1.0 0.0 0.0

1.0 **0.5** 1.0 **0.5**

**0.5** **0.5** 1.0 0.0 0.0

1.0 0.0 **0.5** 0.0 **0.5** 0.0

**0.5 0.5 0.5** 1.0 1.0 1.0 **0.5**

and another one with 12 ½’s:

**0.5**

1.0 0.0

1.0 .375 0.0

1.0 1.0 .625 **0.5**

**0.5**  **0.5  0.5  0.5**  0.0

1.0  0.0  **0.5  0.5**  0.0  **0.5**

**0.5** 1.0  **0.5**  0.0  1.0  **0.5**  0.0

Incidentally, Michael Proschan and Jeff Rosenthal have an [2009 American Statistician paper](https://amstat.tandfonline.com/doi/abs/10.1198/tast.2010.09184#.Xbng9NF7nb0) on another modification of the quincunx they call the uncunx! Playing a wee bit further with the annealing, and using a denominator of 840 let to a 2P  with 14 ½’s out of 28

**.5**

60 0

60 1 0

**30 30 30** 0

**30 30 30 30 30**

60  60  60  0  60  0

60  **30**  0  **30** **30** 60 **30**