

Markov Chains

Mercan Horia, Ariton Alexandru

June 2023

1 Introduction

We aim to find a solution for any given labyrinth

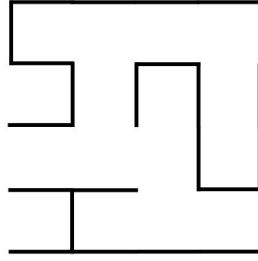


Figure 1: Sample Labyrinth

For the entire labyrinth we make an adjacency matrix, whose element a_{ij} is 1 if one can reach from cell i to cell j in a single movement.

From that we make a Stochastic matrix which we name *the link matrix*. For each cell we define the probability of reaching one of its' neighbours by the value $\frac{1}{N}$ where N is the number of neighbours.

We will consider the *winner* state represented as a vertex in our graph. In the following matrices that we will build, the penultimate column and the penultimate line in the adjacency matrix represents the linkages between *winner* states and positions in labyrinth. Also, the last line and the last column will be associated to the *loser* state.

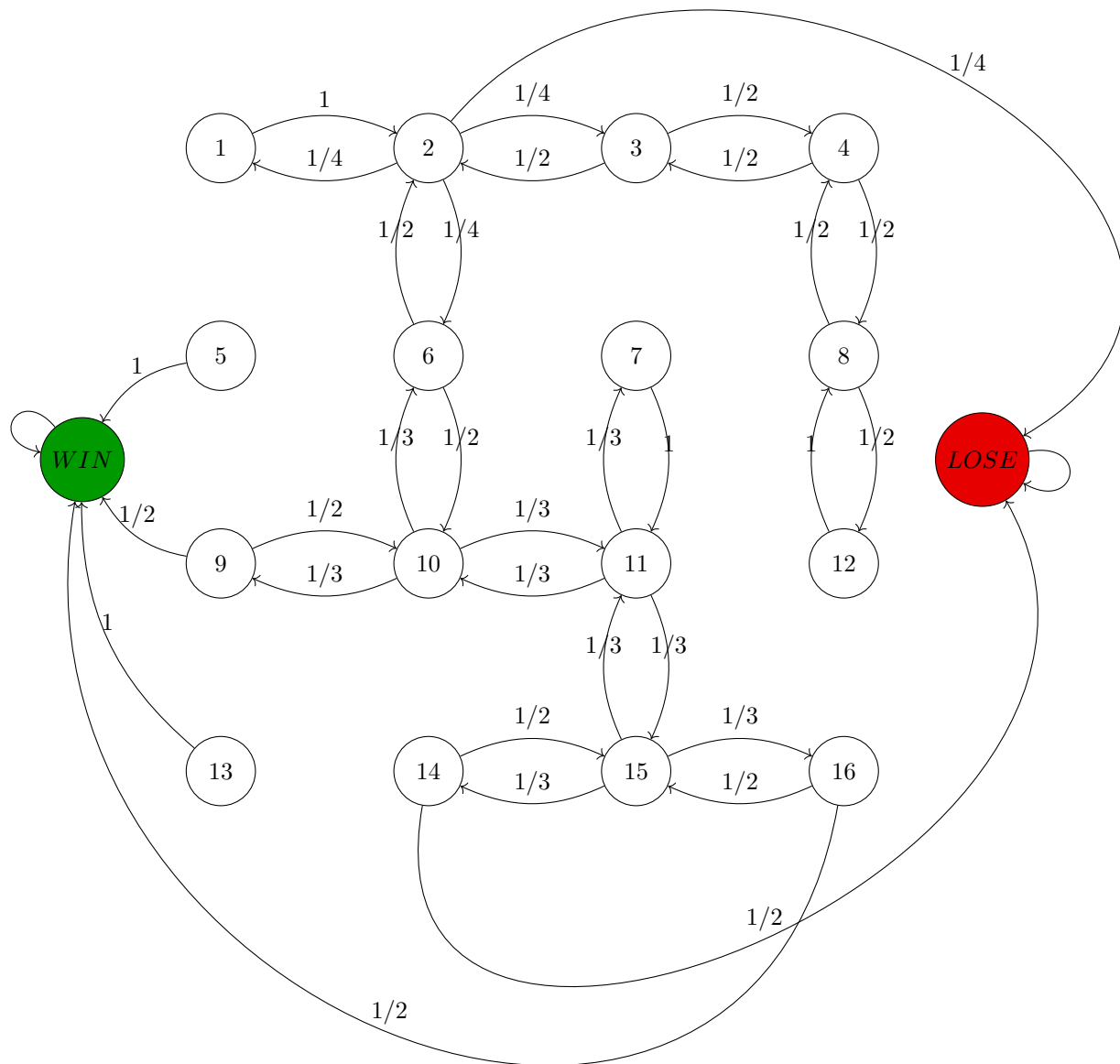
For the given labyrinth they are as following.

[illegible]

We will consider the following line-stochastic matrix, associated to our labyrinth.

[illegible]

Using this matrix, we can represent the following Markov Chain:



Based on this graph, we will construct a linear system to be solved. For every position in our labyrinth, based on the all possible transitions, our desire is to find out the probability to win this puzzle. We will make some notations: p_k as the probability to win when you are placed on the position associated with the state k . We will make the following observations: $p_{WIN} = 1$ and $p_{LOSE} = 0$.

We will consider the column-vector $\tilde{p} \in \mathbb{R}^{16 \times 1}$ with its entries representing the probabilities of the 16 states.

We will make one more essential observation: The probability of winning for a given state is directly influenced by the one-grade neighbour states, and every probability can be expressed using the law of total probabilities.

So, by considering the lines and columns only associated with the partial graph of the states, we can write the following matrix:

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

Also, by considering the column of the directed linkages between states and the win state as:

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1/2 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1/2 \\ 1 \\ 0 \end{bmatrix}$$

Using the information that our probabilities have to respect the law of total probabilities, we can write the following linear system:

$$\tilde{p} = G\tilde{p} + b,$$

with the unknown vector \tilde{p} . Now we will make use of the Jacobi Method for solving this linear equation.

Let us make some observation about solving the linear systems using Jacobi Method. Given a linear system $Ax = b$, we would like to solve it by using an iterative method. Jacobi Method uses the simple idea of writing the matrix A as a difference of matrices D and P , where D represents the matrix that only contains the entries placed on the principal diagonal of A .

$$A = D - P$$

So, by denoting x_p the values of the solution at every iterative step p , we can iterate following the rule:

$$x_{p+1} = D^{-1}Px_p + D^{-1}b$$

The convergence of this method is assured by $\rho(D^{-1}P) < 1$

In our case, we have to solve the system

$$(I_n - G)\tilde{p} = b$$

We will present below the code used for solving this problem, based on the Jacobi method:

jacobi.m

```

function [x, iter] = jacobi(A, b, max_iter, tol)
[n] = size(A, 1);
D = diag(diag(A));
P = D - A;

disp("The spectral radius of the iteration matrix")
max(abs(eig(inv(D) * P)))
G = inv(D) * P;
b = inv(D) * b;
x0 = zeros(n, 1);

for iteration = 1 : max_iter
    iter = iteration;

    x = G * x0 + b;
    if (norm(x - x0) < tol)
        break;
    endif

    x0 = x;
endfor

decimals = round(-log(tol)/log(10));
x = round(x * 10^decimals);
x = x / 10^decimals;
endfunction

```

solveMarkov.m

```

clear; clc; close all;
load Matrix;

[n] = size(Matrix, 1);
G = Matrix(1 : n - 2, 1 : n - 2);
b = Matrix(1:n - 2, n - 1);

[x, iter] = jacobi(eye(n - 2) - G, b, 10000, 1e-6);

disp("Probabilitatile sunt:")
disp(x);

disp("Numarul necesar de iteratii:")
disp(iter)

```

By running solving the linear system with the proposed methods, we will obtain the following solution for our column-vector:

$$\tilde{p} = \begin{bmatrix} 0.190474 \\ 0.190474 \\ 0.190472 \\ 0.19047 \\ 1 \\ 0.380951 \\ 0.547618 \\ 0.190469 \\ 0.785714 \\ 0.571427 \\ 0.547618 \\ 0.190468 \\ 1 \\ 0.261904 \\ 0.523809 \\ 0.761904 \end{bmatrix}$$

So, for the given puzzle, the probability of escaping the labyrinth by entering in the state 4 is 0.19047.