LSINF2275 Project
« Data mining and decision making »
Markov Decision Processes

Professor: Marco Saerens marco.saerens@uclouvain.be
Address: Université catholique de Louvain
LSM & ICTEAM
Place des Doyens 1
B-1348 Louvain-la-Neuve
Belgique
Telephone: 010 47.92.46.
Fax : 010 47.83.24.
Teaching assistants: Mathieu Zen mathieu.zen@uclouvain.be

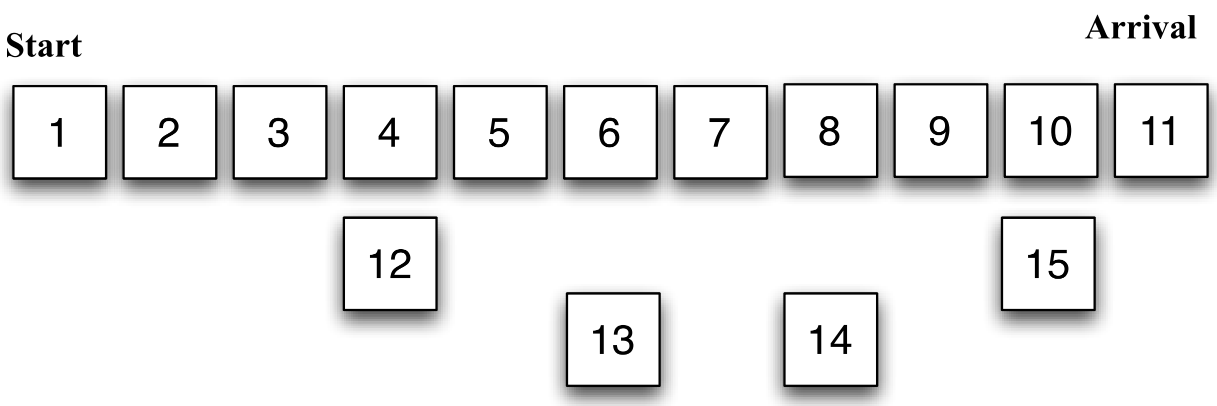
Objective

The objective of this project is to put into practice several techniques used in the analysis of quantitative data (and decision making). This will be done through the study of a practical case which requires the use of software related to the statistical processing of data (such as *Matlab*). This work aims at applying the « Markov decision processing » in the framework of a simulation of a Snakes and Ladders game (i.e. a « jeu de l'oie »).

Problem statement

The realisation of the project will be carried out in groups of 2 students. You are asked to determine the solution of the following problem :

Let us assume that you are playing a Snakes and Ladders game which is made of 15 squares. Square number 1 is the initial square (start) and square 11 is the winning square (arrival). If you reach square 11, you win. A schematic representation of this game is shown below:



To move forward, you can choose between two dice :

- The « security » dice which can only give two possible values : 1 or 0. It allows you to move forward by 0 or 1 square, with a probability of 1/2. With that dice, you are invincible, which means that you cannot fall into any trap.
- The « risky » dice which randomly allows the player to move by 0, 1 or 2 squares, each of these actions having a probability of 1/3. When you use that dice, you are exposed to the traps if they are on the square you reach.

At square 3, there is a junction with two possible paths. If the player only passes through square 3, he will continue to square 4, as if the other path did not exist. Conversely, if the player stops on square 3, he will have, on the following turn, an equal probability of going to square 4 with the normal path or to square 12 for the shortcut. Both paths ultimately reach the final square.

It should also be possible to define « trap » squares. If the player stops on one of these squares, he will have to restart from the 1st square on the following turn.

We ask you to determine the optimal strategy, i.e. for each square, which dice should be played, to reach the arrival square in a minimal number of **turns**, on average. You should determine this solution in two different scenarios :

- You should exactly stop on the arrival square to win. The game board is designed as a circle, which means that, if you pass the last square, you reach the 1st square again.
- You win as soon as you have **passed** the arrival square (i.e. if you are **on** square 11 or further – see details later).

You should implement, in *Matlab, Octave, Python* or *R*, the following function :

`[Expec, Dice] = markovDec(list)`

This function launches the Markov Decision Process algorithm to determine the optimal strategy regarding the choice of the dice in the Snakes and Ladders game, using the « Value-Iteration » method.

<u>In :</u>

List : Vector of 15 values representing the 15 squares of the Snakes and Ladders game.
The value for each square = 0 if it is an ordinary square.
= 1 if it is a « trap » square (go back to square 1).

<u>Out :</u>

Expec : Vector that contains the expectation of the cost for the 15 squares of the game.
Dice: Vector that contains the choice of the dice to use for each of the 15 squares of the game (1 for « security » dice and 2 for « risky » dice).

<u>Additional experiments</u>

You could define two other kinds of traps : (2) « prison » squares : if the player stops on this square, he will have to wait one turn before playing again. (3) « retreat » squares : if the player stops on this square, he will have to move backwards by 2 squares.

(Optional) Obviously, it is also very interesting to compare this strategy with other strategies, i.e. the use of dice 1 only, or dice 2 only, or a mixed random strategy, etc. Launch/simulate an important number of games and compare the performances of each of these strategies with the optimal strategy (or policy) obtained by value iteration. You should then report (i) the theoretical expected cost, obtained by value iteration, and compare it with the empirical average cost when playing (simulating) a large number of games with the optimal strategy, and (ii) compare the empirical average cost obtained by the optimal strategy with the other sub-optimal strategies mentioned before. You could test this with a few different configurations of traps, depending on your time left.

<u>Report</u>

You are asked to write a report of around 6 pages (not including the code). This report will be made of

- a brief explanation of the method used to determine the optimal strategy regarding the choice of the dice.
- a description of your implementation along with results/comparisons comparing the optimal strategy and the empirical results obtained for the Snakes and Ladders game with, or without trap squares at relevant places.