Metaheuristics for Optimization

Series 0 : Stochastic processes

Return no later than September 18, 2023 (23h59)

Throughout the semester, you will have to implement algorithms that require efficient event generation with given probabilities. This series is an introduction to several possible methods. The problem is always the following : *let us consider N possible events occurring with probability P_i, $i \in \{0, ..., N-1\}$*. Generate a sequence of events such that each event occurs with the appropriate probability. The problem is not so trivial, as we will see through examples.

Simulation of a balanced dice

First, we simulate rolling a N-face balanced dice. Each side of the dice has a $\frac{1}{N}$ probability of being obtained.
For each roll of the dice, generate a random number $r \in [0, 1)$ and consider that the event $i = \lfloor rN \rfloor$. Events can be visualized as "drawers" in the space of possible random numbers. This case is very simple, because it is assumed that the operations involved can be performed in $O(1)$.
Generate a long sequence of rolls of a six-sided dice, and check that the frequency of appearance of each side is $\frac{1}{6}$. Draw a histogram of the frequencies.
How would the histogram change if the sides are not equi-probable ?

Simulation of a biased coin toss

A coin has a probability p of falling on 'heads' and a probability $1-p$ of falling on 'tails'. When $p \neq 0.5$, the coin is biased. To simulate the throw of a coin, generate a random number $r \in [0, 1)$, and use the variable $\lambda = \lfloor r + p \rfloor$. What are the possible values of λ, and with what probabilities are they obtained ? Finally, use the following variable $x = \lambda a + (1 - \lambda)b$ (a for 'tails' and b for 'heads') to verify that the part you generated is biased with the p probability you defined.

Simulation of a double biased coin toss

This time, two coins have a probability p_1 and p_2 of falling on 'heads', and a probability $1-p_1$ and $1-p_2$ of falling on 'tails'. Based on the previous exercise, explain how to simulate a double coin toss, and implement it.
N.B. The order is important.

Roulette method

The last method considered in this TP is the roulette method. Given N events and their probabilities P_i, $i \in \{0, ..., N-1\}$, the initialization step consists in calculating the cumulative probabilities $P_i^{cumul} = \sum_{j=0,...,i} P_j$,

$i \in \{0, .., N-1\}$. Generate a random number $r \in [0, 1)$, and convince yourself that the event performed is the smallest i such as $P_i^{cumul} > r$.

Implement the latter method and check that for a large number of events generated, the frequency of occurrence of each event corresponds to the probability associated with it.

Work to return

Submit your code, results, and short comments, and upload them on moodle, by the end of the day (Monday, September 18, 2023 at 23h59).

NB : The use of python notebooks is highly recommended, to be able to include code, results, and comments in the same file. Graphs must include a title and a legend for the axes when needed.