IE523: Financial Computing
Fall, 2018
Programming Assignment 7: Simulation Exercises
Due Date: 2 November 2018
©Prof. R.S. Sreenivas

There are two parts to this programming assignment.

1   Part 1: How to Loose as Little as Possible

Skim through the first few pages of reference [1], which is available on *Compass*. We have two players Alice and Bob, each have a coin that they can toss. The probability of heads for Alice's coin is q, and that for Bob's coin is p, where $q < p$. They agree to to the following game

> Each will toss their respective coins n times, Alice wins if and only if she gets more heads than Bob at the end of the n-many tosses.

Obviously, the game favors Bob over Alice, but reference [1] studies the following problem

> What is the optimal value of n that will maximize Alice's chance of winning?

You should take some time to understand equation 1.1 of the paper. You should also understand what Theorem 2.2 of the paper is trying to say about the shape of the plot shown in figure 1 in a general setting (i.e. for any value of $q < p$; not just $q = 0.18$ and $p = 0.2$). This will give you an idea of how you might want to write the optimization-code for this problem described in subsection 1.1.

1.1   The Optimization Problem of Reference [1]

In this part of the programming assignment you are going to write a C++ program that verifies the results reported in [1]. For this programming assignment I want you to define an appropriate Class structure and member functions that picks the optimal value of n for values of p and q that are read at the command line. A sample output is shown in figure 1. Depending on your implementation, it might take longer on your computer, be patient!

2   Part 2: The Experimental Verification of Reference [1]

In the previous part of this programming assignment you wrote C++ code that computed the (combinatorial) expression for the probability of Alice's winning the "n-coin-toss" game. In this programming assignment you are going to write

Figure 1: Sample output. Depending on your implementation, it might take some time to run on your computer. Be patient.).

a simulation that "experimentally" verifies the (combinatorial) expression used in the previous programming assignment.

For a given p and q, your simulation should try a range of possible values for n (say, from 1 to 25), and repeat the "n-coin-toss" game a large number of times no_of_trials (say, no_of_trials $= 10^6$ times) – you then count the number of times Alice wins, and divide it by the number of repetitions (i.e. 10^6) and present that as the probability of Alice's winning. You should plot this probability as a function of n, and compare it with the (combinatorial/theoretical) expression in [1]. I am looking for a plot that looks like what is shown in figure 2. Depending on your implementation and the number of repetitions of the game, it might take longer on your computer, be patient!
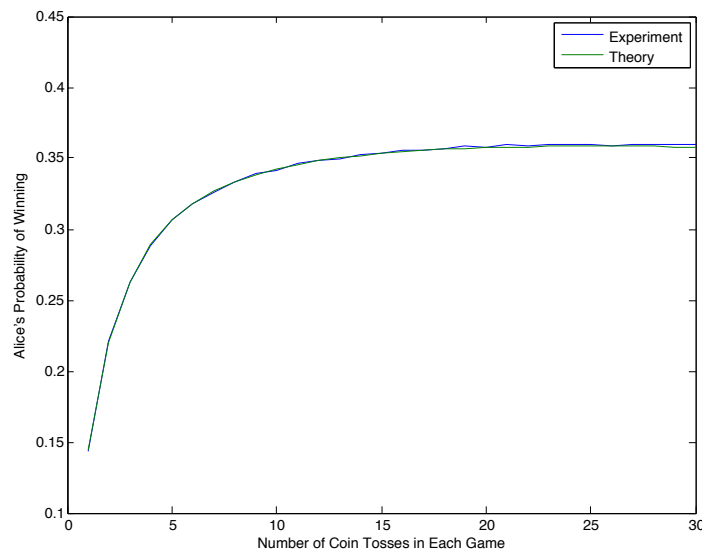

Figure 2: Sample output $q = 0.18$, $p = 0.2$, no_of_trials $= 500,000$. Depending on your implementation, it might take some time to run on your computer. Be patient. This run took a little over 6 minutes on my computer.

References

[1] V. Addona, S. Wagon, and H. Wilf. How to lose as little as possible. *ARS MATHEMATICA CONTEMPORANEA*, 4:29–62, 2011.