MINI PROJECT 1 - INTERIM REPORT

APMA 3100

By Winston Zhang, Luke Mathe

Resubmitted 14. March 2022

TABLE OF CONTENTS AND FIGURES

1. OBJECTIVE

The objective of this project is to develop a simple Monte-Carlo simulation, which is a technique of artificially creating very large samples of random variables that can be used to mimic the behavior of random processes. This project is intended for us students to become familiar with designing such a model of experimentation on a computer, so that we as engineers can become skilled enough to apply such a system to more sophisticated problems.

2. SIMULATION SYSTEM

As described by the assignment description, a Monte-Carlo simulation system is comprised of three components:

1. **Random Number Generator**

   An algorithm that outputs a series of real numbers between 0 and 1, also known as pseudo-random numbers. As suggested by the name, a cursory glance at these pseudo-random numbers would appear to be indistinguishable from a sample of independent realizations of a uniform random variable.

2. **Random Variable Generator**

   An algorithm that maps realizations of the uniform random variable into realizations of the random variable with a specific cumulative distribution function.

3. **Mathematical model**

   This model should describe the problem set out to be simulated. This model specifies the CDF of the random variable described previously in list item **(2)**.

3. RANDOM NUMBER GENERATOR

As instructed by the assignment description, the random number generator's implementation will follow that of a commonly used algorithm, known as the linear congruential random number generator. This algorithm is a cyclical recursive function, that generates its outputs through real number division as described below:

$$x_i = (a\, x_{i-1} + c)(\textit{modulo}\ K),$$

$$u_i = x_i / K,$$

where parameters a, c, and K are constants, and the *modulo* refers to the remainder of a number after performing integer division. For this specific implementation, parameters are described in Table 1:

Starting value (seed)	$x_0 = 1000$
Multiplier a	$a = 24\ 693$
Increment c	$c = 3517$
Modulus K	$K = 2^{17}$

Table 1: Parameters for Random Number Generator Algorithm

With these parameters, this specific implementation of our random number generator will yield a cycle of pseudo-random numbers that is 2^{17} values long. This algorithm was implemented in Java as its own class, and took a total of 44 lines to write, including test cases in the main method. The algorithm was run with a few test inputs to ensure correctness, and the corresponding pseudo-random numbers that were expected as outputs, are described in Table 2, with inputs given by the assignment instructions highlighted in boldface:

$\boldsymbol{u_1}$	**0.4195**
$\boldsymbol{u_2}$	**0.0425**
$\boldsymbol{u_3}$	**0.1274**
u_{51}	0.5157
u_{52}	0.4273
u_{53}	0.7682

Table 2: Expected Pseudo-Random Number Outputs for Random Number Generator Algorithm

4. RANDOM VARIABLE GENERATOR

Discrete Random Variable

As covered in APMA 3100 material up until this point, the cumulative distribution function F of random variable X can be defined as the probability of non-exceedance:

$$F(x) = P(X \leq x) = \sum_{y \leq x} p(y),$$

with the previously defined and implemented random number generator, a given random number u_i generates realization x_i of X via the rule:

$$x_i = \min\{x : F(x) >= u_i\},$$

and the above rule may be repeatedly executed over $x = 1, \ldots, k$ until $F(x)$ is found to be greater than u_i.

Continuous Random Variable

Let X be a continuous random variable having a continuous distribution function F whose inverse exists in a closed form such that a given random variable u_i, as input to the inverse CDF, generates realization x_i of X:

$$x_i = F^{-1}(u_i)$$

5. SIMULATION PROBLEM

Model

1. W is a continuous random variable based on X being a continuous random variable when the call is available to be answered. However, a discrete number of seconds are taken when the call cannot be answered for one reason or another, in the instance when the call cannot be answered over the course of 4 attempts, W would be a discrete random variable. The model of the call being answered is a continuous random variable X with an exponential distribution, with $\mu_X = 12$ and thus $\lambda_X = 1/12$. W is a discrete random variable with μ_W being the mean time spent on the phone by the representative.
2. The CDF of W can be modeled by cumulative probabilities from the probability tree in combination with the integrated probability from the exponential distribution of the continuous random variable of X.
3. The model can be represented as a decision tree, as shown in Figure 1:

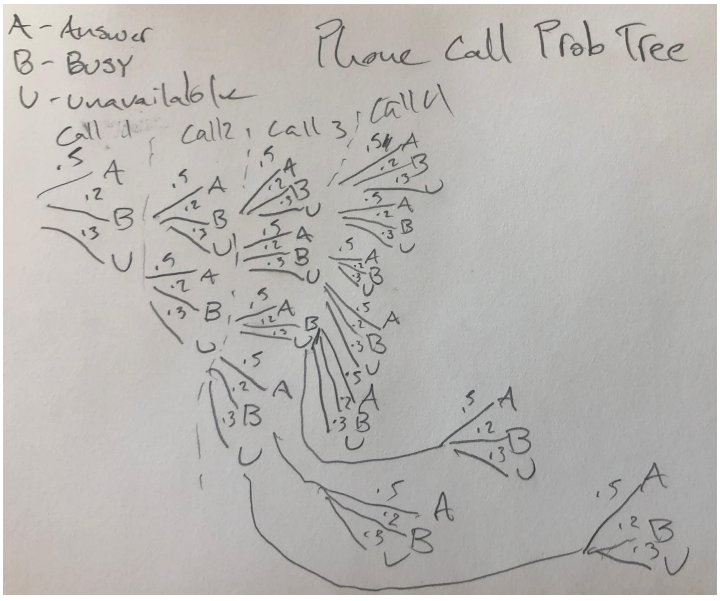

Figure 1: Simulation Problem Modeled as Decision Tree

This model can also be represented in a more condensed form as a flowchart, as shown in Figure 2:

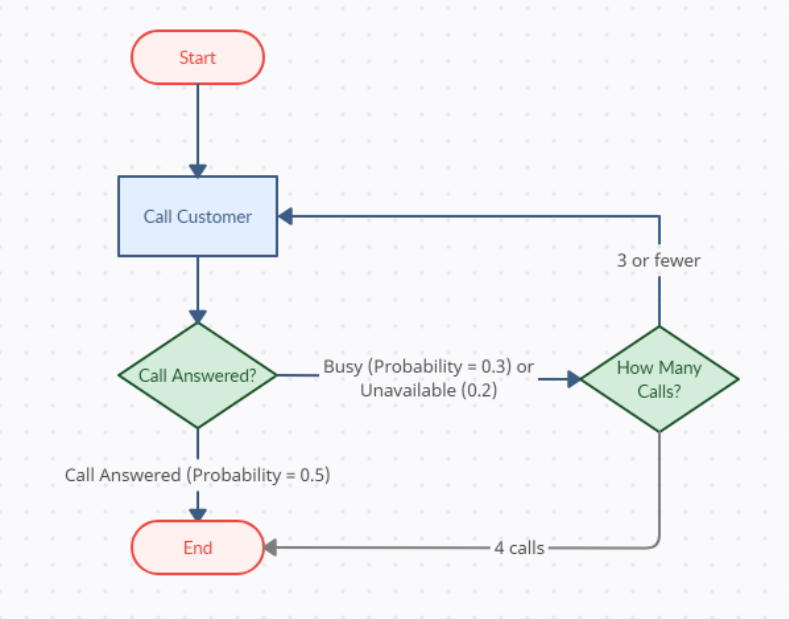

Figure 2: Simulation Problem Modeled as Flowchart

Data Collection

Data was collected through an ad-hoc experiment with actual phones. The time it took to mirror the actions taken by the telemarketer in the assignment description was recorded and repeated, and the average values were recorded. These values will replace those originally described in the assignment description, as shown in Table 3:

Activity	Described value (seconds)	Actual Value (seconds)
Turn on Phone, Dial Number	6	7.08
Detect Busy Signal	5	5.4
Wait for 5 rings	25	32.9
End Call	1	1.2

Table 3: Experimental Time Values

In order to more realistically simulate the described process, numbers being dialed were random and read from a “phone book”, as dialing memorized numbers may skew the data towards smaller time values.