

EFFICIENT METHODS IN SIMULATION MODELING AND PROGRAMMING

Jin Wang and Chunlei Liu, Valdosta State University, Valdosta, GA 31698

ABSTRACT

In the real world related Monte Carlo simulation modeling and programming, application of efficient methods plays a very important role. We discuss some efficient methods, such as Bootstrap resampling, Variance Reduction, and random number reusing techniques. Our study shows that the reused pseudo random numbers are less random and less uniform. We provide a feasible method to improve the quality of reusing pseudo random numbers with detailed procedure of implementation.

Keywords

Bootstrap, Variance Reduction, pseudo random number, Linear Congruential Generator, Reusing random number

INTRODUCTION

Due to the complexity of real world problems, Monte Carlo simulation may be the only feasible method of solving these problems. Simulation itself is efficient, accurate, and easy to implement. In Monte Carlo simulation modeling and programming, application of efficient methods plays a very important role.

We review and discuss some efficient methods in simulation modeling and programming. In general, we use all available information to fit system variables into existing distributions. Therefore, we generate system variables based on these fitted distributions. Sometimes, the fitted distributions are not reliable and not robust due to the limited data information. Bootstrapping is resampling technique without using a theoretical or simulation model. No distributions are needed to simulate the system. Under the limited situation of information, Bootstrapping is more accurate and easier to implement. This method generates samples from the data set directly.

The accuracy of simulation outputs are evaluated by the common standard: sample variance or standard error. We discuss some variance reduction techniques in simulation modeling, programming, and output analysis. Simulation efficiency means increasing the output accuracy without any additional simulation cost. We do know that if we want to improve accuracy of one more digit, we need to increase the iteration number of simulation by 100 times. The new cost of simulation is 100 times.

Monte Carlo simulation requires generating lots of random numbers. Reusing the generated random numbers is an efficient technique in simulation programming. In Monte Carlo simulation, all random numbers are pseudo random. They are deterministic numbers on a cycle with a period. They act like random numbers but are not true random numbers. There is a quality issue of reusing pseudo random numbers here. These reused random numbers are less uniform and independent. The period number is reduced. We provide a feasible method to pull the period back to the original level. Two different situations are discussed with detailed implementation procedures.

EFFICIENT METHODS IN SIMULATION

In modeling and solving real world problems, sometimes the closed-form models do not exist due to the difficulty and complexity of these problems. In other situations, even if the closed-form models or solutions exist, the computational results cannot be derived in the time manner. For example, a fund manager has to decide what stocks should be dropped, kept, or added into the current portfolio. All analysis based on today's and previous historic data has to be done before the market open tomorrow. Monte Carlo simulation does provide an easy and efficient way to solve such complex problems [1] [4] [6] [7]. There are so many real world applications for computer simulation methods. For practitioners, simulation may be the only feasible resource to manage realities.

How to efficiently model and simulate a complex system is an important issue. In a general simulation procedure, all available information (data) is used to fit into an existing distribution. We generate samples from this distribution through simulation methods. Therefore we can evaluate the system performance measures from the simulation outputs. This method is referred as a parametric simulation [3] [6].

The parametric simulation has its limitations if the information is scarce. For example, we are interested in the average life time for a particular model of aircraft. The historic life time data is limited to a very small sample size. Small data set can be fitted into many distributions in statistics. However these fitted distributions (models) are not reliable. Bootstrapping is a nonparametric simulation method. It is a great method when only limited information is available. No theoretical and simulation models are needed. In addition to the efficiency, Bootstrapping is more accurate and easy to implement. We generate a sample directly from the available data set without using any distribution. This simulation method is also referred to as a resampling technique. It has become very popular in recent years for estimating such things as standard errors, confidence intervals, biases, and prediction errors. Its automatic nature and applicability to complicated problems have contributed to its popularity [3] [5].

A powerful tool, computer simulation is widely used in practice [1] [4] [6] [7]. However practitioners have paid less attention to the statistical analysis of computer simulation output results. How good are the simulation results? What is the common measurement for evaluating the accuracy of the simulation results? The variance is the common standard used to determine the accuracy for the significant and the swing digits [10].

In theory, the variance can be reduced if the sample size is increased. However, there is an extra cost of increasing the sample size. In general, if we want to improve the accuracy to one more digit, we need to run the simulation experiment 100 times more. That means the cost is 100 times greater [10]. Making the sample size bigger is not what we usually mean by variance reduction. There are many ways to reduce the variance in simulation implementations without any extra costs. Variance Reduction is a technique that reduces the variance of the point estimator to make the simulation experiment more efficient.

In system design (policy) evaluation, we should compare the alternative system under the similar conditions. Common Random Numbers are used in generating such systems for comparison [1] [6] [7]. The variance of the overall system performance (difference) is reduced due to the positive correlation between systems. Implementation of the Common Random Numbers technique can be difficult for some complex systems.

Antithetic Variates is a Variance Reduction technique in simulating a single system [1] [6] [7]. We generate pairs instead of a single variate. The pair itself is negatively correlated. We use the average of the pair as a single sample or observation point for simulation analysis. In theory, the variance will be reduced due to the negative correlation.

The main idea of Control Variates method is to take advantage of positive correlation between certain random variables to gain a variance reduction [1] [6] [7]. For example, in portfolio risk control, Coca Cola and Pepsi are in the same market sector. Their stock prices bounce up and down at the same time with positive correlation. We may use the stock price of Pepsi as a control variable to correct (adjust) the stock price of Coca Cola to obtain a variance reduction.

There are some other Variance Reduction methods, which involve high level implementation techniques, such as Importance Sampling, Stratified Sampling, and Conditioning methods [1] [6] [7].

In Monte Carlo simulation modeling and programming, random number generation is key. In the next two sections, we study the random number reusing method.

RANDOM NUMBER GENERATION

A great amount of random numbers is needed in Monte Carlo simulation implementation. Random numbers were originally generated either manually or mechanically using techniques such as spinning of wheels, dice rolling, or card shuffling [7]. The modern approach is to use computers to generate random numbers. Such random numbers are called Pseudo Random Numbers. By definition, random variates generated from the Uniform (0, 1) distribution are called random numbers.

In theory, pseudo random numbers are not random [1] [6] [7]. They are from a deterministic sequence of numbers in (0, 1). We assume that pseudo random numbers act like true random numbers.

The most commonly used random number generator is the Linear Congruential Generator (LCG), which was proposed by Lehmer [1] [6] [7]. This algorithm was used to generate pseudo random numbers in many software, such as Borland C/C++, IBM VisualAge C/C++, Microsoft Visual/Quick C/C++, IMSL, Random class in Java API [9].

The LCG is defined by the recurrence relation:

$$W_{i+1} = (a W_i + c) \text{ modulo } m,$$

where a , c , n and W_0 are non-negative integer constants. W_0 is called the initial seed. This generates pseudo random integers,

$$W_i \in \{0, 1, \dots, n-1\}.$$

The pseudo random numbers are,

$$U_i = \frac{W_i}{n}, i = 1, 2, \dots$$

At most, this algorithm only produces n possible different numbers. If the generator has a full cycle, then its period is n . This means that all n numbers are different. Let U_n be any pseudo random number with period n , then it is a discrete uniform random variable in $[0, 1]$ and

$$U_n \in \left\{ \frac{0}{n}, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n} \right\}.$$

This discrete Uniform (0, 1) random variable converges to the continuous Uniform (0, 1) random variable in distribution [9].

In a parametric simulation setting, there are mainly four different ways to generate system variates. The four methods are Inverse Transformation, Composition, Acceptance/Rejection, and Special Properties [1] [6] [7].

ANALYSIS OF REUSING RANDOM NUMBER TECHNIQUE

For the simulation of a complex system, we are required to generate lots of random numbers. For example, the Acceptance/Rejection method is very costly in terms of using random numbers. In order to improve the simulation efficiency, can random numbers be reused as part of the simulation? Is there any quality issue on these reusing random numbers?

The reusing random number technique was introduced in [8]. For example, we want to generate the Double Exponential random variable with the following probability density function:

$$f(x) = \frac{1}{2} f_1(x) + \frac{1}{2} f_2(x),$$

here

$$f_1(x) = \frac{1}{\beta} \text{Exp}\left(\frac{x}{\beta}\right) I\{x < 0\}$$

$$f_2(x) = \frac{1}{\beta} \text{Exp}\left(-\frac{x}{\beta}\right) I\{x \geq 0\}.$$

We use the Inverse Transformation method to generate this variable. The simulation codes are listed as following,

- General Code:
 $u = \text{rand}()$ *call random number generator*
 if ($u < .5$) then
 $u_1 = \text{rand}()$
 $x = \beta \ln(1 - u_1)$
 else
 $u_2 = \text{rand}()$
 $x = -\beta \ln(1 - u_2)$
 End if
- Efficient Code:
 $u = \text{rand}()$
 if ($u < .5$) then
 $x = \beta \ln(2u)$
 else
 $x = -\beta \ln(2(1 - u))$
 End if

Comparing both codes, the efficient code only uses and reuses one random number. This code has the efficient advantage of reusing random numbers. What is the theoretical background of reusing random numbers? The theory is stated as follows.

Theorem 1: Let u be a Uniform (0, 1) random variable and $C = \{u: 0 < a \leq u \leq b < 1\}$ be a condition. We define the conditional random variable u_c as follows

$$u_c = u|C,$$

and then transfer u_c into the following random variable,

$$u_r = \frac{u_c - a}{b - a}.$$

Therefore u_r is Uniform (0, 1) random variable.

Proof: By definition, u_c is a Uniform (a, b) random variable. Then we enlarge (transfer) it from the sub interval (a, b) into the whole interval (0, 1). This completes our proof: u_r is Uniform (0, 1) random variable.

Here u_r is the random number for reusing. Theoretically speaking, this result is correct only for true random numbers. Due to the special property and structure of pseudo random numbers, we have a quality issue of reusing pseudo random numbers.

If a LCG has a full period n , there are only n different pseudo random numbers on the cycle. For example, if $(a, b) = (0, .5)$, this implies that there are only $n/2$ different ready-to-reuse pseudo random numbers. Interval (0, .5) only contains 50% of the total different pseudo random numbers. Under the transformation, on interval (0, 1), at most, we have $n/2$ possible different values of pseudo random numbers. In the sense of uniformity and randomness, the quality of reusing pseudo random number is diminished. This means that the period was reduced.

How do we solve this problem? In order to maintain the same quality, we need to increase the period and pull it back to the original level. The technique of adding random numbers is one feasible way to produce a longer period [2]. The extra cost is negligible, since we are reusing random numbers. Here is the main result from [2]. If u and v are two independent Uniform (0, 1) variates, then $u + v \text{ (modulo 1)}$ is “closer” to Uniform (0, 1) than either u or v in the sense of uniform and independent.

Technically, how many random numbers do we need to add them up? The total length of each sub interval should not be less than 1, which is the length of the interval (0, 1). This number can be determined based on the values of a and b . For example, if $(a, b) = (0, .5)$, we need at least two random numbers for addition. This method works while a and b are not fixed for every reusing pseudo random number. For example, pseudo random numbers are from the Acceptance/Rejection implementation.

For a single random number generator, if both values of a and b are fixed, adding up random numbers would not increase the period. We should use different random number generators to implement this idea. Specifically, we need equal numbers of generators and pseudo random numbers for this technique. In the sum combination of pseudo random numbers, only one pseudo random number should be selected from each generator. By doing this, the period quality problem will be resolved completely.

CONCLUSIONS

We have discussed some important efficient methods in Monte Carlo simulation modeling, programing, and output analysis. The main result of this study is the technique of reusing pseudo random numbers. We derive the theoretical result to support this idea. The quality of reused random number has diminished in the sense of independent and uniform properties. The period of reused random number has reduced. We provide a feasible method of improving the length of the period. Adding random numbers is a way to create a longer random number generator period. We provide detailed steps of implementation in different situations.

ACKNOWLEDGEMENT

We would like to thank Professor Bruce Schmeiser of Purdue University for providing this topic to us.

REFERENCES

- [1] Bratley, P., Fox, B.L., and Schrage, L.E. *A Guide to Simulation*, 2nd Edition, New York, NY: Springer-Verlag, 1987.
- [2] Brown, M., and H. Solomon, 1979, "On Combining Pseudorandom Number Generators," *Annals of Statistics*, 1979, 7: 691-695.
- [3] Efron, B., and Tibshirani, R.J. *An Introduction to the Bootstrap*. New York, NY: Chapman & Hall, 1993.
- [4] Fishman, G.S. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer Series in Operations Research, New York, NY: Springer-Verlag, 1996.
- [5] Hall, P. *The Bootstrap and Edgeworth Expansion*. New York, NY, Springer-Verlag, 1992.
- [6] Law, A.M. and Kelton, W.D. *Simulation Modeling and Analysis*, 3rd Edition, New York, NY: McGraw Hill, 2000.
- [7] Ross, S.M. *A Course in Simulation*, New York, NY: Macmillan Publishing Company, 1990.
- [8] Schmeiser, B.W. *IE 581 Lecture Notes*, The School of Industrial Engineering, Purdue University, 1992.
- [9] Wang, A.L., Xu, Z., and Kicey, C.J. "Asymptotic Statistical Analysis of Pseudo Random Numbers," *Proceedings of the 2011 IEEE International Conference on Computer Science and Automation Engineering*, Piscataway, NJ: IEEE Press, 2011, 542–545.
- [10] Wang, A.L. and Kicey, C.J. "On the Accuracy of Buffon's Needle: A Simulation Output Analysis," *Proceedings of the 49th Annual ACM SE Conference*, New York, NY: ACM Press, 2011, 233–236.