Generadores de números aleatorios

Uno de los principales resultados que buscaremos en simulación, es la generación por computadora de secuencias de número aleatorios, usualmente distribuidos uniformemente entre 0 y 1. Haciendo un recuento histórico, podríamos generar una secuencia mediante los siguientes métodos:

- Buscar en tablas de números aleatorios
- Observación de un proceso físico (ruido blanco, desintegración radiactiva,etc.)
- Experimentos con juegos de azar
- Algoritmos de computadora

Por obviedad, la preferencia será hacia el uso de herramientas computacionales, las cuales tendrán entre sus principales ventajas la reproducibilidad y rapidez. Una desventaja fundamental que debemos considerar es que dichas secuencias no son realmente aleatorias debido a la naturaleza determinística de su obtención, sin embargo, en términos prácticos serán de suma utilidad y podremos probar que poseen características estadísticas satisfactorias. Dichas secuencias son conocidas como pseudo-aleatorias.

Para generar las secuencias, se requiere un valor inicial (semilla) a partir del cual se derivarán todos los cálculos, generalmente, la secuencia se encontrará mediante la relación de recurrencia:

$$N_i = f(N_{i-1}, ..., N_{i-k})$$

Ahora, revisaremos los tipos más comunes de generadores de números aleatorios.

Método de Von Neumann

John Von Neumann fue un matemático húngaro-estadounidense considerado como uno de los más importantes de la historia moderna, sus contribuciones fueron de vital importancia para la computación actual. Von Neumann propuso uno de los primeros métodos para la generación de números aleatorios conocido como el método de los cuadrados medios. El algoritmo es el siguiente:

1. Elegir un número arbitrario con n dígitos
2. Elevar el número al cuadrado
3. Usar los n dígitos medios como siguiente número

Sin embargo, el generador presenta patrones muy cortos y repetitivos careciendo completamente de aleatoriedad. El mismo von Neumann dijo:

"Cualquiera que considere métodos aritméticos para producir dígitos aleatorios está, por supuesto, en pecado mortal.

Generadores de congruencia lineal (GCL)

Se basan en la relación de recurrencia:

$$N_{i+1} = (aN_i + b) \, mod \, m$$

Donde los parámetros a y m son llamados multiplicador y módulo respectivamente. El generador arrojará m valores posibles acotados entre 0 y $m-1$, la secuencia generada será periódica. El periodo dependerá de los parámetros elegidos y el valor semilla. Algunas propiedades deseables de un generador deben ser:

- Eficiente computacionalmente
- Periodo largo
- Valores sucesivos deben ser independientes y uniformemente distribuidos

En cuanto a la selección de los parámetros para un GCL, destacamos:

- a, b y m afectan el periodo y autocorrelación
- El módulo m debe ser largo
- El periodo nunca podrá exceder a m
- Para eficientar los cómputos m debe ser una potencia de 2

Si b es no nulo, es posible obtener el máximo periodo m sí y solo sí:

- m y b son primos relativos (sin factores comunes más que 1)
- Todo factor primo de m es también factor primo de $a-1$
- Si m es múltiplo de 4, también debe serlo $a-1$

Algunas configuraciones populares en GCL son:

Fuente	m	a	b
Unix GCC	2^{31} - 1	1103515245	12345

Borland C/C++	2^{32}	22695477	1
Microsoft VB 6	2^{24}	1140671485	12820163
Java	2^{48} - 1	25214903917	11

Generador Mersenne Twister (MT)

Propuesto por Makoto Matsumoto y Takuji Nishimura en 1997. Proporciona secuencias de números pseudoaleatorios de alta calidad con periodo largo (elegido como primo de Mersenne), es veloz y confiable. La implementación estándar es la llamada MT19937 que usa una longitud de palabra de 32 bits. Es el algoritmo de elección de múltiples sistemas de software como: Excel, Octave, Julia, MATLAB, PHP, Python, R, Ruby, SPSS y SAS. La definición del algoritmo no es trivial, se puede encontrar el artículo original en la liga http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/ARTICLES/mt.pdf

Los generadores vistos con anterioridad nos permiten generar secuencias de números pseudoaleatorios distribuidas uniformemente en el intervalo $[0, 1]$, sin embargo, es de interés en simulación poder generar números pertenecientes a otras distribuciones, revisemos algunas técnicas para lograrlo: