Generadores de números aleatorios.

La clave en la generación discreta de eventos aleatorios es la habilidad para generar números "*aleatorios*". Por lo general, se requiere una gran cantidad de ellos para un estudio de simulación típico. Es esencial que sean generados tan rápido y eficientemente como sea posible.

No se deberían llamar realmente "*aleatorios*" sino más bien *pseudoaleatorios*[1], es decir, aquellos generados por técnicas determinísticas en sucesiones reproducibles. Tales números se utilizan para introducir el comportamiento estocástico del sistema bajo estudio.

La generación de tales números debe cumplir con ciertas pruebas estadísticas de "*aleatoriedad*". La mayoría de los métodos para generar números pseudoaleatorios suelen ser de naturaleza iterativa.

Propiedades deseables de los generadores de números pseudoaleatorios.[2]

1. ***"Aleatoriedad".*** Que "no exista correlación" entre los números generados. Uniformemente distribuidos entre (0,1). Aplicación de test estadísticos de aleatoriedad.

2. ***Período largo.*** Los generadores están basados en el uso de fórmulas determinísticas precisas y la secuencia obtenida se repetirá. Se llama *periodo* al tamaño de la secuencia generada. Es deseable que el periodo sea lo más largo posible. Desde un punto de vista práctico se desea que sea lo suficientemente largo de manera que la secuencia no se repita ella misma durante una misma simulación (ejecución del modelo en la computadora) sencilla.

3. ***Reproducibilidad.*** Capacidad para reproducir una misma secuencia dada (verificación más fácil del programa, utilización de números pseudoaleatorios "idénticos" en la simulación de diferentes sistemas de manera de obtener comparaciones más precisas).

4. ***Eficiencia computacional.*** Se desea reducir los requerimientos de *tiempo* y *memoria* en la medida de lo posible (velocidad).

Entre los primeros generadores se encuentra el "método de los cuadrados centrales" propuesto por John von Neumann en 1946. El número "aleatorio" se obtiene exponenciando al cuadrado el viejo número y quedándose con los dígitos centrales.

$$n_{i+1} = \{ \text{ dígitos medios de } n_i^2 \}$$
$$n_1 = 25073$$

[1] *Pseudo*: Prefijo de origen griego que significa *falso*.
[2] *Byron S. Gottfried,* "Elements of Stochastic Process Simulation", Prentice Hall Inc., Englewood Cliffs, New Jersey, 1984.

$$n_1^2 = 628655329$$

$$n_2 = 86553$$

$$n_2^2 = 7491421809$$

$$n_3 = 14218$$

$$n_3^2 = 202151524$$

$$n_4 = 21515 \ldots$$

La desventaja de este generador es que tiende a degenerar en presencia de cifras bajas.

n_i	$n_i / 10000 \in (0,1)$	n_i^2
1002		01004004
0040	0.0040	00001600
0016	0.0016	00000256
0002	0.0002	00000004

... se estanca en 0.

Variante del método: $n_{i+1} = \{$ *dígitos centrales de* $n_i \times n_{i-1}\}$

Algunos autores consideran que las dos propiedades más importantes que se deben pedir a una secuencia de números pseudoaleatorios son la *uniformidad* y la *independencia*[3]. En la literatura sobre simulación mediante eventos discretos se parte de un generador de números pseudoaleatorios uniformemente distribuidos entre 0 y 1, pues este será posteriormente la base para la generación de variables aleatorias, discretas o continuas.

Métodos congruenciales.[4]

Se denomina de esta manera a uno de los métodos más comunes para generar secuencias de números pseudoaleatorios. Se trata de un método iterativo que genera la secuencia de números a través de una *relación de recurrencia*, comenzando con un número inicial llamado *semilla*. La relación de recurrencia utilizada no es más que una *relación de congruencia*[5].

Definición: $a \equiv b (\text{mod} \quad m)$ significa que $m | (a-b)$[6].

[3] *J. Banks, John S. Carson, Barry L. Nelson,* "Discrete-Event System Simulation", Prentice-Hall, Inc., 1999.
[4] Introducidos por Lehmer (1951).
[5] Se trata de una relación de equivalencia (se cumplen las propiedades reflexiva, simétrica y transitiva).
[6] *m* divide a (*a*-*b*) o *m* es divisor de (*a*-*b*).

Equivalentemente, *a* y *b* son congruentes módulo *m*, si (*a*-*b*) es algún múltiplo de *m*, es decir, (*a*-*b*) = *km* ó (*a*-*b*) pertenece al conjunto de múltiplos de *m*. Si $r \equiv a(\text{mod} \quad m)$ con $0 \le r < m$, entonces *r* se denomina "residuo" de la división de *a* entre *m*. Como la relación de congruencia es una relación de equivalencia, se cumple que $r \equiv a$ y $a \equiv r$ (simetría). Luego *a*-*r* = *km* y *a* = *km*+*r*. Ejemplo: $5 \equiv 13(\text{mod} \quad 8)$.

El *método congruencial mixto*[7] genera una sucesión de números enteros en el rango comprendido entre 0 y *m*-1. En general, se denota como sigue,

$$x_{n+1} \equiv (ax_n + b)(\text{mod} \quad m)$$

donde *a*, *b* y *m* son enteros positivos (*a*<*m*, *b*<*m*). Si *b* = 0 se tiene un *método congruencial multiplicativo*.

$$x_{n+1} \equiv ax_n(\text{mod} \quad m)$$

"Entre los posibles generadores basados en el método congruencial multiplicativo, uno de los más utilizados es el *generador de Learmouth-Lewis*,

$$x_{n+1} \equiv 7^5 x_n(\text{mod} \quad 2^{31} - 1)$$

El mismo ha sido probado ampliamente y los resultados de las pruebas estadísticas indican que es muy satisfactorio. Algunas versiones de este generador son usadas por el paquete de la International Mathematics and Statistical Library (IMSL)"[8]. Si *a* = 1 y se sustituye *b* por algún número pseudoaleatorio anterior a x_n en la sucesión como por ejemplo x_{n-1}, se tiene entonces un *método congruencial aditivo*. Este método requiere más de una semilla para iniciar el cálculo.

$$x_{n+1} \equiv (x_n + x_{n-1})(\text{mod} \quad m)$$

Los detalles teóricos relacionados con la construcción de buenos generadores son objeto de la literatura especializada. Por lo general, caemos en el terreno del *análisis numérico*. Se deben escoger *a*, *b* y *m* de manera tal que los números generados tengan una "conducta aleatoria aceptable". El número *m* representa el número deseado de valores diferentes que se pueden generar como números pseudoaleatorios. Nos interesa que la longitud del periodo de la secuencia pseudoaleatoria sea máximo (periodo completo). Además, la escogencia de los parámetros antes mencionados debe ser tal que se garantice que cada entero comprendido entre 0 y *m*-1 ocurra exactamente una vez en cada ciclo para contribuir a la uniformidad de los números pseudoaleatorios generados.

[7] Se dice *mixto* pues la relación de recurrencia posee un término aditivo y otro multiplicativo.
[8] *Frederick S. Hillier, Gerald J. Lieberman,* "Introducción a la Investigación de Operaciones", McGraw-Hill, 1997.

Algunas de las reglas que se pueden poner en práctica para la escogencia de los parámetros *a*, *b* y *m* son las siguientes[9].

Generador Congruencial Multiplicativo	***Generador Congruencial Mixto***
$m = 2^{w-1}$ $a \cong 2^{w/2}$ $a \equiv \pm 3 (\text{mod} \quad 8)$ *Semilla:* Cualquier entero impar positivo y cuyo valor sea menor que *m*. *Se obtiene una secuencia de números con un periodo es m/4.*	$m = 2^{w-1}$ $a \cong 2^{w/2}$ $a \equiv 1 (\text{mod} \quad 4)$ *Semilla y b:* Cualquier par de enteros impares positivos y cuyos valores sean menores que *m*. *Se obtiene una secuencia de números con un periodo es m (periodo completo).*

w = número de bits por palabra de la computadora.

$x_{n+1} \equiv a x_n (\text{mod} \quad m)$	$x_{n+1} \equiv (a x_n + b) (\text{mod} \quad m)$
$x_1 \equiv a x_0 (\text{mod} \quad m)$ $x_2 \equiv a^2 x_0 (\text{mod} \quad m)$ ... $x_n \equiv a^n x_0 (\text{mod} \quad m)$	$x_1 \equiv (a x_0 + b) (\text{mod} \quad m)$ $x_2 \equiv \left(a^2 x_0 + \frac{b(a^2 - 1)}{(a-1)} \right) (\text{mod} \quad m)$ ... $x_n \equiv \left(a^n x_0 + \frac{b(a^n - 1)}{(a-1)} \right) (\text{mod} \quad m)$

En ambos casos, para obtener los números pseudoaleatorios dividimos cada número generado entre *m*: $u_i = \frac{x_i}{m}, \quad 0 < u_i < 1$.

Ejemplo: Computadora a 12 bits, *w* = 12.

$m = 2^{11} = 2048$ $a = 67 \cong 2^6 = 64, \quad 67 \equiv 3 (\text{mod} \quad 3)$ $x_0 = 129$ $x_1 = 67 * 129 (\text{mod} \quad 2048) = 451$	$m = 2^{11} = 2048$ $a = 65 \cong 2^6 = 64, \quad 65 \equiv 1 (\text{mod} \quad 4)$ $x_0 = 129, \quad b = 1$ $x_1 = 65 * 129 + 1 (\text{mod} \quad 2048) = 194$

[9] *Byron S. Gottfried,* "Elements of Stochastic Process Simulation", Prentice Hall Inc., Englewood Cliffs, New Jersey, 1984.

$u_1 = \frac{451}{2048} = 0.220215$	$u_1 = \frac{194}{2048} = 0.094727$
$x_2 = 67 * 451 (\text{mod} \quad 2048) = 1545$	$x_2 = 65 * 194 + 1 (\text{mod} \quad 2048) = 323$
$u_2 = \frac{1545}{2048} = 0.754395$	$u_2 = \frac{323}{2048} = 0.157715$
...	...
Se tendrá un total de 2048/4 = 512 números distintos. La secuencia se repetirá a partir de $x_{513} = x_1 = 451$.	*Se tendrá un total de 2048 números distintos (periodo completo). La secuencia se repetirá a partir de* $x_{2049} = x_1 = 194$.

"Por lo general, el método congruencial mixto se utiliza con menos frecuencia que el método multiplicativo pues la secuencia obtenida presenta un comportamiento estadísticamente menos "aleatorio" que el método multiplicativo y el método mixto es más lento que el multiplicativo."

La mayoría de los lenguajes de programación vienen dotados con un generador de números pseudoaleatorios. Para efectos del presente curso nos "bastará" con el generador suministrado por el lenguaje de programación, por el paquete general o por el paquete especializado seleccionado para formular el programa de computadora en el proceso de simulación.

Pruebas para generadores de números aleatorios.

Como se dijo anteriormente, la secuencia de números pseudoaleatorios generada debe satisfacer la propiedad de *uniformidad* (los números deben ser seleccionados de una distribución uniforme en el intervalo [0,1]) e *independencia* (el orden de la secuencia debe ser "aleatorio"). Existen pruebas (tests) para evaluar las mencionadas propiedades[10]. En general, a la mayor parte del software comercial de simulación se le realizan las pruebas adecuadas para garantizar "uniformidad" e "independencia".

Para probar la *uniformidad* existen pruebas estadísticas (*Test de Frecuencia*) como:

1. *Kolmogorov-Smirnov.*
2. *Ji-cuadrada.*

Para probar la *independencia* se pueden utilizar pruebas estadísticas como:

1. *La prueba de las corridas (run test).* Examina la manera como los números de una misma secuencia se encuentran dispuestos (conjuntos de números ordenados de manera creciente o decreciente, *corridas*) para evaluar independencia.

[10] *J. Banks, John S. Carson, Barry L. Nelson,* "Discrete-Event System Simulation", Prentice-Hall, Inc., 1999.

2. *La prueba del intervalo (gap test)*. Es utilizada para determinar el significado (tamaño, distribución, etc.) del intervalo de recurrencia de un mismo dígito, es decir, la longitud del intervalo entre dos ocurrencias de un mismo dígito.

3. *La prueba de correlación (autocorrelation test)*. Evalúa la correlación entre números de una misma secuencia y compara la correlación de una muestra con el valor de correlación esperado (0).

4. *La prueba del póquer (poker test)*. Se basa en la frecuencia de repetición (alta o baja) de ciertos dígitos en una misma secuencia de números.