INSTITUTO TECNOLÓGICO DE TEPIC

Métodos de generación de números pseudoaleatorios

Materia: Simulación

"RELACIÓN ENTRE EL BIENESTAR EMOCIONAL DEL ALUMNO DE INGENIERÍA EN SISTEMAS Y SU DESEMPEÑO ACADÉMICO"

Maestro/a: Soto Castro Luis Obed

Alumno:

- Haro Silva Andrés Mitchel - 16400926

Números aleatorios:

Los números aleatorios son aquellos números que se generan, presentando estos la misma probabilidad de ser elegidos o seleccionados.

Un número aleatorio es, básicamente, algo que produce un dispositivo y del cual no podemos saber nada del valor que va a tener, antes de que suceda.

Es aquel obtenido al azar, es decir, que todo número tenga la misma probabilidad de ser elegido y que la elección de uno no dependa de la elección del otro.

Números pseudoaleatorios:

Los números pseudoaleatorios se generan de manera secuencial con un algoritmo determinístico, formalmente se definen por:

Función de inicialización. Recibe un número (la semilla) y pone al generador en su estado inicial.
Función de transición. Transforma el estado del generador.
Función de salidas. Transforma el estado para producir un número fijo de bits (0 ó 1).

Una sucesión de bits pseudoaleatorios se obtiene definiendo la semilla y llamando repetidamente la función de transición y la función de salidas.

Métodos de generación de números pseudoaleatorios

Congruenciales

Lineales

Generador congruencial lineal

El algoritmo congruencial lineal genera una secuencia de números enteros por medio de la siguiente ecuación recursiva:

Fórmula
$X_{i+1} = (aX_i+C) \bmod m$

Donde X0 es la semilla
a es la constante multiplicativa.
C es una constante aditiva
m es el módulo
X0 > 0 , a > 0 , C > 0 y m > 0 deben ser números enteros.
La operación “mod m ” significa multiplicar Xi por a , sumar C y dividir el resultado entre m para obtener el residuo Xi+1 .

Para que el método alcance su máximo periodo de se debe cumplir las siguientes condiciones:

$a = 3+8k$ Donde k es un número entero positivo

X_0 debe ser un número impar

$m = 2^g$ Donde g es un número entero positivo

C es un número impar

Ejemplo:

	A	B	D	E	F	G	H	I
1	Xi+1 =	((aXi+C) mod m)						
2			k=	4			g=	3
3	X0 =	3	a=	35	C=	9	m=	8
4	x1	2						
5	x2	7						
6	x3	6						
7	x4	3						
8	x5	2						
9	x6	7						
10	x7	6						
11	x8	3						
12	x9	2						
13	x10	7						
14	x11	6						
15	x12	3						
16	x13	2						
17	x14	7						
18	x15	6						
19	x16	3						
20	x17	2						
21	x18	7						
22	x19	6						
23	x20	3						
24								

Algoritmo congruencial multiplicativo

Surge del algoritmo congruencial lineal cuando C = 0, con lo que la ecuación resulta en:

$X_{i+1} = (aX_i) \bmod m$

La ventaja resulta en el necesitar una operación menos, requiriendo así menos potencia de cómputo.

Fórmula

$X_{i+1} = (aX_i) \bmod m$

Donde X0 es la semilla.

a es la constante multiplicativa.

m es el módulo.

X0 > 0 , a > 0 , C > 0 y m > 0 deben ser números enteros.

La operación “mod m ” significa multiplicar Xi por a , sumar C y dividir el resultado entre m para obtener el residuo Xi+1 .

Para que el método alcance su máximo periodo de se debe cumplir las siguientes condiciones:

$a = 3+8k$ Donde k es un número entero positivo
X_0 debe ser un número impar
$m = 2^g$ Donde g es un número entero positivo

Ejemplo:

	A	B	D	E	H	I
1	Xi+1 =	((aXi+C) mod m)				
2			k=	2	g=	4
3	X0 =	5	a=	19	m=	16
4	x1	15				
5	x2	13				
6	x3	7				
7	x4	5				
8	x5	15				
9	x6	13				
10	x7	7				
11	x8	5				
12	x9	15				
13	x10	13				
14	x11	7				
15	x12	5				
16	x13	15				
17	x14	13				
18	x15	7				
19	x16	5				
20	x17	15				
21	x18	13				
22	x19	7				
23	x20	5				
24						

Metodo congruencial aditivo

calcula una sucesión de números pseudoaleatorios mediante la relación Xn+1= Xn +Xn-k (mod M). Para usar este método se necesitan k valores iniciales, siendo k entero. Las propiedades estadísticas de la secuencia tienden a mejorarse a medida que k se incrementa.
Este es el único método que produce periodos mayores que M.

Fórmula
$X_{i+1} = X_i + X_{i-k}$ Mod M

Ejemplo

	A	B	C	D
1	Xi+1 =	(Xi+Xi-k) mod m		
2			g=	4
3	X0 =	35	m=	16
4	x1	24		
5	x2	87		
6	x3	19		
7	x4	34		
8	x5	19		
9	x6	4		
10	x7	12		
11	x8	78		
12	x9	1		
13	x10	9		
14	x11	0		
15	x12	3		
16	x13	5		
17	x14	8		
18	x15	12		
19	x16	8		
20	x17	6		
21	x18	7		
22	x19	0		
23	x20	0		
24	x21	3		
25	x22	8		
26	x23	0		
27	x24	12		
28	x25	4		

No lineales

Algoritmo congruencial cuadrático

Este algoritmo tiene la ecuación recursiva:

$X_{i+1} = (aX^2_i + bX_i + C) \bmod m$

Para que el método alcance su máximo periodo de se debe cumplir las siguientes condiciones:

$m = 2^g$ Donde g es un número entero positivo
a Debe ser un número par
(b-1) mod 4 = 1
C es un número impar

Ejemplo:

	A	B	C	D	E	F	G	H	I	J
1			g =	3						
2	X0=	18	m =	8	b =	5	c =	7	a =	8
3	X1 =	1								
4	X2 =	4								
5	X3 =	3								
6	X4 =	6								
7	X5 =	5								
8	X6 =	0								
9	X7=	7								
10	X8 =	2								
11	X9	1								
12	X10	4								
13	X11	3								
14	X12	6								
15	X13	5								
16	X14	0								
17	X15	7								
18	X16	2								
19	X17	1								
20	X18	4								
21	X19	3								
22	X20	6								
23	X21	5								
24	X22	0								
25	X23	7								

Algoritmo de Blum Blum y Shub

Si en el algoritmo congruencial cuadrático a = 1, b = 0 y c = 0 , entonces se construye una nueva ecuación recursiva:

$X_{i+1} = (aX^2_i) \bmod m$

La anterior ecuación fue propuesta por Blum, Blum y Shub como Nuevo método para generar números que no tienen un comportamiento predecible.

Para que el método alcance su máximo periodo de se debe cumplir las siguientes condiciones:

m = 2g Donde g es un número entero positivo

Ejemplo

	A	B	C	D
1			g =	3
2	X0=	17	m =	37
3	X1	30		
4	X2	12		
5	X3	33		
6	X4	16		
7	X5	34		
8	X6	9		
9	X7	7		
10	X8	12		
11	X9	33		
12	X10	16		
13	X11	34		
14	X12	9		
15	X13	7		
16	X14	12		
17	X15	33		
18	X16	16		
19	X17	34		
20	X18	9		
21	X19	7		
22	X20	12		
23	X21	33		
24	X22	16		
25	X23	34		

No congruenciales

Método de los cuadrados medios

El método comienza tomando un número al azar, x_0, de 2^n cifras que al elevarlo al cuadrado resulta un número de hasta 4^n cifras. Si es necesario se a˜naden ceros a la izquierda para que el número resultante tenga exactamente 4^n cifras. Sea x_1 el número resultante de seleccionar las 2^n cifras centrales de x_0^2 el primer número aleatorio u_1 se obtiene poniendo un punto decimal delante las 2_n cifras de x_1. A continuación x_2 y u_2 se generan a partir de x_1 del mismo modo. Así sucesivamente.

Ejemplos:

x0 = 3708 ⇒ x_0^2 = 13|7492|64 ⇒ x1 = 7492 ⇒ u1 = 0.7492
x1 = 7492 ⇒ x_1^2 = 56|1300|64 ⇒ x2 = 1300 ⇒ u2 = 0.1300
x2 = 1300 ⇒ x_2^2 = 1|6900|00 ⇒ x3 = 6900 ⇒ u3 = 0.6900
x3 = 6900 ⇒ x_3^2 = 47|6100|00 ⇒ x4 = 6100 ⇒ u4 = 0.6100
x4 = 6100 ⇒ x_4^2 = 47|2100|00 ⇒ x5 = 2100 ⇒ u5 = 0.2100

$x5 = 2100 \Rightarrow x_5^2 = 4|4100|00 \Rightarrow x6 = 4100 \Rightarrow u6 = 0.4100$
$x6 = 4100 \Rightarrow x_6^2 = 16|8100|00 \Rightarrow x7 = 8100 \Rightarrow u7 = 0.8100$
$x7 = 8100 \Rightarrow x_7^2 = 65|6100|00 \Rightarrow x8 = 6100 \Rightarrow u8 = 0.6100$

Algoritmo de productos medios

La mecánica de generación de números pseudo aleatorios de este algoritmo no congruencial es similar a la del algoritmo de cuadrados medios. La diferencia entre ambos radica en que el algoritmo de productos medios requiere dos semillas, ambas con D dígitos; además, en lugar de elevarlas al cuadrado, las semillas se multiplican y del producto se seleccionan los D dígitos del centro, los cuales formarán el primer número pseudo aleatorio ri = 0.D . Después se elimina una semilla y la otra se multiplica por el primer número de D dígitos, para luego seleccionar del producto los D dígitos que conformarán un segundo número. Entonces se elimina la segunda semilla y se multiplican el primer número de D dígitos por el segundo número de D dígitos; del producto se obtiene el tercer número ir . Siempre se irá eliminando el número más antiguo, y el procedimiento se repetirá hasta generar los n números pseudoaleatorios. A continuación se presentan con más detalle los pasos del método para generar números con el algoritmo de productos medios

Ejemplo:

Semillas= $X_0 = 4587$, $X_1 = 2174$

$Y_1 = 4587 * 2174 = 9|9721|38$	$X_2 = 9721$ $r_1=0.9721$
$Y_2 = 2174 * 9721 = 21|1334|54$	$X_3 = 1334$ $r_2=0.1334$
$Y_3 = 9721 * 1334 = 12|9678|148$	$X_4 = 9678$ $r_3=0.9678$
$Y_4 = 1334 * 9678 = 12|9104|52$	$X_5 = 9104$ $r_4=0.9104$
$Y_5 = 9678 * 9104 = 88|1085|12$	$X_6 = 1085$ $r_5=0.1085$
$Y_6 = 9104 * 1085 = 9|8778|40$	$X_7 = 8778$ $r_6=0.8778$
$Y_7 = 1085 * 8778= 9|5241|30$	$X_8 = 5241$ $r_7=0.5241$
$Y_8 = 8778 * 5241= 46|0054|98$	$X_9 = 0054$ $r_8=0.0054$

Algoritmo de multiplicador constante

Este algoritmo no congruencial es similar al algoritmo de productos medios. Los siguientes son los pasos necesarios para generar números pseudoaleatorios con el algoritmo de multiplicador constante.
1. Seleccionar una semilla (X0) con D dígitos (D > 3).
2. Seleccionar una constante (a) con D dígitos (D > 3).
3. Sea Y0= a * X0, sea X1= los D dígitos del centro, y sea ri= 0.D dígitos del centro. 4. Sea Yi= a*X¡; sea X i + 1= los D dígitos del centro, y sea ri= 0.D dígitos del centro para toda i = 1,2,3,..., n. 5. Repetir el paso 4 hasta obtener los n números deseados.
Si no es posible obtener los D dígitos del centro del número Yi agregue ceros a la izquierda del número Y.

Ejemplo=

Semilla X_0= 5412 Constante C=8475

Y_1 = 8475 * 5412 = 45|8667|00 X_2 = 8667 r_1=0.8667
Y_2 = 8475 * 8667 = 73|4528|25 X_3 = 4528 r_2=0.4528
Y_3 = 8475 * 4528 = 38|3748|00 X_4 = 3784 r_3=0.3784
Y_4 = 8475 * 3784 = 23|0694|00 X_5 = 0694 r_4=0.0694
Y_5 = 8475 * 0694 = 5|8816|50 X_6 = 8816 r_5=0.8816
Y_6 = 8475 * 8816 = 75|7156|00 X_7 = 7156 r_6=0.7156
Y_7 = 8475 * 7156 = 60|6471|00 X_8 = 6471 r_7=0.6471
Y_8 = 8475 * 6471 = 57|8417|25 X_9 = 8417 r_8=0.8417
Y_9 = 8475 * 8417 = 71|3340|75 X_{10} = 3340 r_9=0.3340

Referencias

Becerra, G. (2016, 28 abril). *UN SISTEMA GENERADOR DE NÚMEROS PSEUDO ALEATORIOS*. https://www.researchgate.net https://www.researchgate.net/profile/Guillermo-Becerra-2/publication/228356658_UN_SISTEMA_GENERADOR_DE_NUMEROS_PSEUDO_ALEATORIOS/links/5722411608aef9c00b7c7db7/UN-SISTEMA-GENERADOR-DE-NUMEROS-PSEUDO-ALEATORIOS.pdf

Ortiz, M. T. (2014). *Estadística Computacional*. https://tereom.github.io/. https://tereom.github.io/est-computacional-2018/index.html

Haahr, M. (s. f.). *RANDOM.ORG - Introduction to Randomness and Random Numbers*. RANDOM.ORG. Recuperado 22 de septiembre de 2021, de https://www.random.org/randomness/