Tarea 2

UNIVERSIDAD PRIVADA DE SANTA CRUZ DE LA SIERRA

Antonio Miguel Natusch Zarco
MI415 - Modelación y Simulación
2022111958
Ingeniería de Sistemas

Índice

Números pseudoaleatorios

Estos problemas se pueden encontrar en la **sección 2.5** del libro principal de la materia, de García et al. (2013, p. 52–58),

1) Notación y Conceptos

- $r_i = \{r_1, r_2, r_3, ..., r_n\}$: Secuencia de números aleatorios entre el intervalo $(0, 1)$ que contiene n números, todos ellos diferentes.
- n: *Período* o *ciclo de vida* del generador que creó la secuencia r_i.
- $$f(r) : \begin{cases} 1, & 0 \leq r \leq 1 \\ 0, & \text{en cualquier otro valor} \end{cases}$$

  ...distribución uniforme continua que debe seguir un conjunto de r_i.
- N: Período de vida *bastante grande* para un generador de números pseudoaleatorios.
- X_i: Número entero positivo que se utiliza como semilla o detonador en los algoritmos congruenciales y no congruenciales para generar números pseudoaleatorios.

(García et al., 2013, p. 22-23)

1.1) Algoritmos no congruenciales

1.1.1) Algoritmo de cuadrados medios

Este algoritmo no congruencial fue propuesto en la decada de los cuarenta del siglo xx por Von Neumann y Metropolis. Requiere un número entero detonador (llamado semilla) con D dígitos, el cual es elevado al cuadrado para seleccionar del resultado los D dígitos del centro; el primer número se determina simplemente anteponiendo el «0.» a esos dígitos. Para obtener el segundo r_i se sigue el mismo procedimiento, solo que ahora se elevan al cuadrado los D dígitos del centro que se seleccionaron para obtener el primer r_i.
Este método se repite hasta obtener n números r_i. A continuación se presentan con más detalle los pasos para generar números con el algoritmo de cuadrados medios.

1. Seleccionar una semilla (X_0) con D dígitos $(D > 3)$.
2. Sea $Y_0 =$ resultado de elevar X_0 al cuadrado; sea $X_1 =$ los D dígitos del centro, y sea r_i = 0.D dígitos del centro.
3. Sea $Y_i =$ resultado de elevar X_i al cuadrado; sea $X_{i+1} =$ los D dígitos del centro, y sea r_i = 0.D dígitos del centro para toda $i = 1, 2, 3, ..., n$
4. Repetir el paso 3 hasta obtener los n números r_i deseados.

Nota Si no es posible obtener los D dígitos del centro del número Y_i, agregue ceros a la izquierda del número Y_i. (García et al., 2013, p. 24)

Ejemplo

Generar los primeros 5 números a partir de una semilla X_0 = 5735, de donde se puede observar que D = 4 dígitos.

Solución:

$$
\begin{aligned}
&Y_0 = (5735)^2 = 32890225 && X_1 = 8902 && r_1 = 0.8902 \\
&Y_1 = (8902)^2 = 79245604 && X_2 = 2456 && r_2 = 0.2456 \\
&Y_2 = (2456)^2 = 06031936 && X_3 = 0319 && r_3 = 0.0319 \\
&Y_3 = (0319)^2 = 101761 && X_4 = 0176 && r_4 = 0.0176 \\
&Y_4 = (0176)^2 = 030976 && X_5 = 3097 && r_5 = 0.3097
\end{aligned}
$$

El algoritmo de cuadrados medios generalmente es incapaz de generar una secuencia de r_i con periodo de vida n grande. Además, en ocasiones sólo es capaz de generar un número, por ejemplo, si $X_0 = 1000$, entonces $X_1 = 0000; r_i = 0.0000$ y se dice que el algoritmo se degenera con la semilla de $X_0 = 1000$. (García et al., 2013, p. 25)

1.1.2) Algoritmo de productos medios

La mecánica de generación de números pseudoaleatorios 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 numero pseudoaleatorio r_i = 0.D dígitos. Después se elimina una semilla, y la otra se multiplica por el primer numero de D dígitos, para luego seleccionar del producto los D dígitos que conformarán un segundo numero r_i. 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 r_i. Siempre se irá eliminando el numero 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 producto medios.

1. Seleccionar una semilla (X_0) con D dígitos $(D > 3)$
2. Seleccionar una semilla (X_1) con D dígitos $(D > 3)$
3. Sea $Y_0 = X_0 * X_1$; sea $X_2 =$ los D dígitos del centro, y sea $r_i = 0.D$ dígitos del centro.
4. Sea $Y_i = X_i * X_{i+1}$; sea $X_{i+2} =$ los D dígitos del centro, y sea $r_{i+1} = 0.D$ dígitos del centro para toda $i = 1, 2, 3, ..., n$.
5. Repetir el paso 4 hasta obtener los n números r_i deseados.

Nota Si no es posible obtener los D dígitos del centro del número Y_i, agregue ceros a la izquierda del número Y_i. (García et al., 2013, p. 25)

Ejemplo

Generar los primeros 5 números r_i a partir de las semillas $X_0 = 5015$ y $X_1 = 5374$; observe que ambas semillas tienen $D = 4$ dígitos.

Solución:

$$\begin{array}{lll} Y_0 = (5015)(5734) = 28756010 & X_2 = 7560 & r_1 = 0.7560 \\ Y_1 = (5734)(7560) = 43349040 & X_3 = 3490 & r_2 = 0.3490 \\ Y_2 = (7560)(3490) = 26384400 & X_4 = 3844 & r_3 = 0.3844 \\ Y_3 = (3490)(3844) = 13415560 & X_5 = 4155 & r_4 = 0.4155 \\ Y_4 = (3844)(4155) = 15971820 & X_6 = 9718 & r_5 = 0.9718 \end{array}$$

(García et al., 2013, p. 26)

1.1.3) 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 (X_0) con D dígitos $(D > 3)$.
2. Seleccionar una constante (a) con D dígitos $(D > 3)$.
3. Sea $Y_0 = a * X_0$; sea $X_1 =$ los D dígitos del centro, y sea $r_i = 0.D$ dígitos del centro.
4. Sea $Y_i = a * X_i$; sea $X_{i+1} =$ los D dígitos del centro, y sea $r_{i+1} = 0.D$ dígitos del centro para toda $i = 1, 2, 3, ..., n$.
5. Repetir el paso 4 hasta obtener los n números r_i deseados.

Nota Si no es posible obtener los D dígitos del centro del número Y_i, agregue ceros a la izquierda del número Y_i. (García et al., 2013, p. 26)

Ejemplo Generar los primeros 5 números r_i a partir de la semilla $X_0 = 9803$ y con la constante $a = 6965$. Observe que tanto la semilla como la constante tienen $D = 4$ dígitos.

Solución:

$$\begin{array}{lll} Y_0 = (6965)(9803) = 68277895 & X_1 = 2778 & r_1 = 0.2778 \\ Y_1 = (6965)(2778) = 19348770 & X_2 = 3487 & r_2 = 0.3487 \\ Y_2 = (6965)(3487) = 24286955 & X_3 = 2869 & r_3 = 0.2869 \\ Y_3 = (6965)(2869) = 19982585 & X_4 = 9825 & r_4 = 0.9825 \\ Y_4 = (6965)(9825) = 68431125 & X_5 = 4311 & r_5 = 0.4311 \end{array}$$

(García et al., 2013, p. 26)

1.1.4) Algoritmo lineal

Este algoritmo congruencial fue propuesto por Lehmer en 1951. Según Law y Kelton, no ha sido el más usado. El algoritmo congruencial lineal genera una secuencia de números enteros por medio de la siguiente ecuación recursiva:

$$X_i + 1 = (ax_i + c) \bmod(m) \quad i = 0, 1, 2, 3, ..., n$$

donde X_0 es la semilla, a es la constante multiplicativa, c es una constante aditiva, y m es el módulo. $X_0 > 0, a > 0, c > 0,$ y $m > 0$ deben ser números enteros. La operación «mod(m)» significa multiplicar X_i por a, sumar c, y dividir el resultado entre m para obtener el residuo X_{i+1}. Es importante señalar que la ecuación recursiva del algoritmo congruencial lineal genera una secuencia de números enteros $S = (0, 1, 2, 3, ..., m - 1)$, y que para obtener números pseudoaleatorios en el intervalo $(0, 1)$ se requiere la siguiente ecuación:

$$r_i = \frac{X_i}{m - 1} \quad i = 0, 1, 2, 3, ..., n$$

(García et al., 2013, p. 27)

Ejemplo 1

Generar 4 números entre 0 y 1 con los siguientes parámetros: $X_0 = 37, a = 19, c = 33$ y $m = 100$.

Solución:

$$\begin{aligned} X_1 &= (19 * 37 + 33) \bmod 100 = 36 & r_1 &= \tfrac{36}{99} = 0.3636 \\ X_2 &= (19 * 36 + 33) \bmod 100 = 17 & r_2 &= \tfrac{17}{99} = 0.1717 \\ X_3 &= (19 * 17 + 33) \bmod 100 = 56 & r_3 &= \tfrac{56}{99} = 0.5656 \\ X_4 &= (19 * 56 + 33) \bmod 100 = 97 & r_4 &= \tfrac{97}{99} = 0.9797 \end{aligned}$$

En el ejemplo anterior se dieron de manera arbitraria cada uno de los parámetros requeridos: X_0, a, c y m. Sin embargo, para que el algoritmo sea capaz de lograr el máximo periodo de vida N, es preciso que dichos parámetros cumplan ciertas condiciones. Banks, Carson, Nelson y Nicol sugieren lo siguiente:

$m = 2^g$
$a = 1 + 4k$
k debe ser entero
c relativamente primo a m
g debe ser entero

Bajo estas condiciones se obtiene un periodo de vida máximo: $N = m = 2^g$.

Ejemplo 2 Generar suficientes números entre 0 y 1 con los parámetros $X_0 = 6, k = 3, g = 3,$ y $c = 7$,
hasta encontrar el periodo de vida máximo (N).

Como podemos ver, si se cumplen las condiciones que Banks, Carson, Nelson y Nicol sugieren, se logrará el periodo máximo $N = m = 8$. A continuación se presente el desarrollo de la generación de los números r_{i^*}

$$a = 1 + 4(3) = 13 \text{ y } m = 2^3 = 8$$

$$X_0 = 6$$

$$\begin{aligned}
X_1 &= (13 * 6 + 7) \bmod 8 = 5 & r_1 &= \tfrac{5}{7} = 0.714 \\
X_2 &= (13 * 5 + 7) \bmod 8 = 0 & r_2 &= \tfrac{0}{7} = 0.000 \\
X_3 &= (13 * 0 + 7) \bmod 8 = 7 & r_3 &= \tfrac{7}{7} = 1.000 \\
X_4 &= (13 * 7 + 7) \bmod 8 = 2 & r_4 &= \tfrac{2}{7} = 0.285 \\
X_5 &= (13 * 2 + 7) \bmod 8 = 1 & r_5 &= \tfrac{1}{7} = 0.142 \\
X_6 &= (13 * 1 + 7) \bmod 8 = 4 & r_6 &= \tfrac{4}{7} = 0.571 \\
X_7 &= (13 * 4 + 7) \bmod 8 = 3 & r_7 &= \tfrac{3}{7} = 0.428 \\
X_8 &= (13 * 3 + 7) \bmod 8 = 6 & r_8 &= \tfrac{6}{7} = 0.857
\end{aligned}$$

Es importante mencionar que el número generado en $X_8 = 6$ es exactamente igual a la semilla X_0, y si continuáramos generando más números, éstos se repetirían. Además, sabemos que el algoritmo congruencial lineal genera una secuencia de números enteros $S = (0, 1, 2, 3, ..., m - 1)$. Observe que en este caso de genera la secuencia $S = (0, 1, 2, 3, 4, 5, 6, 7)$, es decir, se generan todos los números enteros

Ejemplo 3 Consideremos de nuevo el ejemplo anterior, pero tratemos de infringir de manera arbitraria alguna de las condiciones. Supongamos que $a = 12$; se sabe que a no es el resultado de $1 + 4k$, donde k es un entero. Veamos el comportamiento del algoritmo congruencial lineal ante tal cambio.

Solución:

$$a = 1 + 4(3) = 13 \text{ y } m = 2^3 = 8$$

$$X_0 = 6$$

$$\begin{aligned}
X_1 &= (12 * 6 + 7) \bmod 8 = 7 & r_1 &= \tfrac{7}{7} = 1.000 \\
X_2 &= (12 * 7 + 7) \bmod 8 = 3 & r_2 &= \tfrac{3}{7} = 0.428 \\
X_3 &= (12 * 3 + 7) \bmod 8 = 3 & r_3 &= \tfrac{3}{7} = 0.428
\end{aligned}$$

El periodo de vida en este caso es $N = 2$, de manera que, como puede ver, el periodo de vida máximo no se logra. Como conclusión tenemos que si no se cumple alguna de las condiciones, el periodo de vida máximo $N = m$ no se garantiza, por lo que el periodo de vida será menor que m.

(García et al., 2013, p. 27–28)

1.1.5) Algoritmo congruencial multiplicativo

El algortimo congruencial multiplicativo surge del algoritmo congruencial lineal cuando $c = 0$. Entonces, la ecuación recursiva es:

$$X_i + 1 = (aX_i) \operatorname{mod}(m) \quad i = 0, 1, 2, ..., n$$

Para transformar los números X_i en el intervalo $(0, 1)$ se usa la ecuación $r_i = \frac{x_i}{m-1}$.
De acuerdo con Banks, Carson, Nelson y Nicol, para que el algoritmo congruencial multiplicativo logre el máximo periodo de vida N, es preciso que los parámetros cumplan las siguientes condiciones:

$m = 2^g$
$a = 3 + 8koa = 5 + 8k$
$k = 0, 1, 2, 3, ...$
X_0 debe ser un número impar
g debe ser entero

A partir de estas condiciones se logra un periodo de vida máximo: $N = \frac{k}{4} = 2^{g-2}$.

Ejemplo

Generar suficientes números entre 0 y 1 con los parámetros $X_0 = 17, k = 2, g = 5$, hasta encontrar el periodo o ciclo de vida.

Solución:

$$a = 5 + 8(2) = 21 \quad \text{y} \quad m = 32$$

$$X_0 = 17$$

$$\begin{aligned}
X_1 &= (21 * 17) \bmod 32 = 5 & r_1 &= \tfrac{5}{31} = 0.612 \\
X_2 &= (21 * 5) \bmod 32 = 9 & r_2 &= \tfrac{9}{31} = 0.2903 \\
X_3 &= (21 * 9) \bmod 32 = 29 & r_3 &= \tfrac{29}{31} = 1.9354 \\
X_4 &= (21 * 29) \bmod 32 = 1 & r_4 &= \tfrac{1}{31} = 0.3225 \\
X_5 &= (21 * 1) \bmod 32 = 21 & r_5 &= \tfrac{21}{31} = 0.6774 \\
X_6 &= (21 * 21) \bmod 32 = 25 & r_6 &= \tfrac{25}{31} = 0.8064 \\
X_7 &= (21 * 25) \bmod 32 = 13 & r_7 &= \tfrac{13}{31} = 0.4193 \\
X_8 &= (21 * 13) \bmod 32 = 17 & r_8 &= \tfrac{17}{31} = 0.5483
\end{aligned}$$

Si la semilla X_0 se repite, volverán a generarse los mismos números. Por lo tanto, el periodo de vida es $n = 8$, el cual corresponde a $N = \frac{m}{4} = \frac{32}{4} = 8$.

Otro ejemplo

Ahora bien, si quebrantamos la condición de que la semilla sea un número impar, digamos con $X_0 = 12$, tenemos:

Solución:

$$X_0 = 12$$

$$X_1 = (21 * 12) \bmod 32 = 28 \qquad r_1 = \frac{28}{31} = 0.9032$$
$$X_2 = (21 * 28) \bmod 32 = 12 \qquad r_2 = \frac{12}{31} = 0.3870$$

En vista de que la semilla X_0 se repite, volverán a generarse los mismos números. Por lo tanto, el periodo de vida es $N = 2$. (García et al., 2013, p. 29–30)

1.1.6) Algoritmo congruencial aditivo

Este algoritmo requiere una secuencia previa de n números enteros $X_1, X_2, X_3, X_4, ..., X_n$ para generar una nueva secuencia de números enteros que empieza en $X_{n+1}, X_{n+2}, X_{n+3}, X_{n+4}, ...$.

Su ecuación recursiva es:

$$X_i = (X_{i+1} + X_{i-n}) \operatorname{mod}(m) \quad i = n + 1, n + 2, n + 3, ...N$$

Los números r_i pueden ser generados mediante la ecuación:

$$r_i = \frac{x_i}{m - 1}$$

Ejemplo

Bibliografía

García, E., García, H., & Cárdenas, L. (2013). *Simulación y análisis de sistemas con ProModel* (Segunda Edición). PEARSON, México.