Unidad 2 - Números Pseudo Aleatorios

En esta unidad, exploraremos los **números pseudo aleatorios**, sus algoritmos y las pruebas estadísticas que se utilizan para validar su calidad. Los números pseudo aleatorios se utilizan en simulaciones, criptografía, juegos de azar y más, donde la aleatoriedad es crucial. A pesar de su nombre, los números pseudo aleatorios no son verdaderamente aleatorios, ya que son generados a partir de una secuencia determinística, pero parecen seguir un comportamiento aleatorio en su distribución.

1. Los Números Pseudo Aleatorios

Definición

Los **números pseudo aleatorios** son secuencias de números que se generan de manera determinista, pero tienen propiedades que los hacen parecer aleatorios. Es decir, no son aleatorios en el sentido clásico, pero su distribución en un rango determinado parece seguir una distribución uniforme.

Importancia de los Números Pseudo Aleatorios

- **Simulación**
  En simulaciones, especialmente en simulación Monte Carlo, es esencial contar con números que se distribuyan uniformemente en un rango determinado.
- **Criptografía**
  En sistemas de encriptación, los números pseudo aleatorios son fundamentales para generar claves seguras.
- **Juegos de azar y loterías**
  Se usan para generar números impredecibles en juegos como el póker o la lotería.

---

2. Algoritmos No Congruenciales

Los **algoritmos no congruenciales** para generar números pseudo aleatorios no utilizan la fórmula congruencial comúnmente asociada con los generadores de números aleatorios. A continuación, se explican algunos de estos algoritmos.

A. Cuadrados Medios (Middle Square Method)

El **método de los cuadrados medios** es uno de los primeros métodos conocidos para generar números pseudo aleatorios. Este algoritmo toma el cuadrado de un número, y luego se extraen los dígitos del medio del resultado como el siguiente número en la secuencia.

Ejemplo práctico:

Supongamos que el número inicial es 1234.

1. Se toma el cuadrado de 1234: 12342=15227561234^2 = 152275612342=1522756.
2. Se extraen los dígitos del medio (por ejemplo, los 4 dígitos centrales): 2275.

Este número es el siguiente en la secuencia. Luego, repites el proceso con 2275 para obtener el siguiente número.

Desventaja

Este algoritmo tiene el inconveniente de que, en algunos casos, la secuencia de números se puede estabilizar o entrar en ciclos repetitivos.

B. Productos Medios (Middle Product Method)

El **método de productos medios** funciona de manera similar al de los cuadrados medios, pero en este caso, toma dos números y calcula el producto de ellos, luego extrae los dígitos del medio.

Ejemplo práctico:

Supongamos que tenemos los números 1234 y 5678.

1. Multiplicamos 1234 por 5678: 1234×5678=70066521234 \times 5678 = 70066521234×5678=7006652.
2. Extraemos los dígitos centrales: 0066.

Este número es el siguiente en la secuencia pseudo aleatoria.

C. Multiplicador Constante (Constant Multiplier Method)

En el **método del multiplicador constante**, el número inicial se multiplica por una constante kkk, y el resultado se toma como el siguiente número.

Ejemplo práctico:

Supongamos que el número inicial es 123 y el multiplicador es 5.

1. Multiplicamos: 123×5=615123 \times 5 = 615123×5=615.
2. El siguiente número de la secuencia es 615.

Este proceso se repite para generar más números.

3. Algoritmos Congruenciales

Los **algoritmos congruenciales** son muy populares debido a su simplicidad y eficiencia. Estos algoritmos generan números pseudo aleatorios utilizando una fórmula congruencial.

A. Algoritmo Congruencial Lineal (Linear Congruential Generator - LCG)

El **generador congruencial lineal** utiliza la siguiente fórmula para generar números pseudo aleatorios:

$$X_{n+1} = (aX_n + c) \mod m$$

Donde:

- XnX_nXn es el número anterior,
- aaa es un multiplicador constante,
- ccc es un incremento constante,
- mmm es el módulo (el rango de números generados).

Ejemplo práctico:

Supongamos que tenemos los siguientes valores:

- $X_0 = 1$ (valor inicial),
- $a = 5$,
- $c = 1$,
- $m = 16$.

Aplicamos la fórmula:

$$X_1 = (5 \times 1 + 1) \mod 16 = 6$$

$$X_2 = (5 \times 6 + 1) \mod 16 = 31 \mod 16 = 15$$

Repetimos el proceso para obtener más números pseudo aleatorios.

B. Algoritmo Congruencial Multiplicativo (Multiplicative Congruential Generator)

Este algoritmo utiliza una fórmula similar, pero sin el término de incremento ccc. La fórmula es:

$$X_{n+1} = (aX_n) \mod m$$

Ejemplo práctico:

Si tomamos:

- $X_0 = 1$,
- $a = 5$,
- $m = 16$.

Entonces, aplicamos la fórmula:

$$X_1 = (5 \times 1) \mod 16 = 5$$

$$X_2 = (5 \times 5) \mod 16 = 25 \mod 16 = 9$$

Y así sucesivamente.

C. Algoritmo Congruencial Aditivo (Additive Congruential Generator)

Este algoritmo se basa en la adición de los números generados previamente:

$$X_{n+1} = (X_n + X_{n-1}) \mod m$$

Ejemplo práctico:

Supongamos que:

- $X_0 = 1$,
- $X_1 = 2$,
- $m = 16$.

Entonces, el siguiente número sería:

$$X_2 = (X_1 + X_0) \mod 16 = (2 + 1) \mod 16 = 3$$

Y así sucesivamente.

D. Algoritmo Congruencial No Lineal (Non-Linear Congruential Generator)

En este tipo de algoritmos, la relación entre los números generados no es lineal. Pueden involucrar productos no lineales o exponentes, lo que aumenta la complejidad y puede mejorar la calidad de los números generados.

Ejemplo:

$$X_{n+1} = (X_n^2 + X_n + 1) \mod m$$

4. Propiedades de los Números Pseudo Aleatorios

Para que los números generados por los algoritmos sean considerados de calidad, deben cumplir con varias propiedades importantes:

A. Uniformidad

Los números deben estar distribuidos de manera uniforme sobre el rango de valores posibles. Esto significa que cada número en el intervalo debe ser igualmente probable.

B. Periodicidad

Aunque los números pseudo aleatorios eventualmente se repiten, el periodo debe ser lo suficientemente largo como para que las repeticiones no afecten los resultados de las simulaciones.

C. Independencia

Los números deben ser independientes entre sí, es decir, el valor de un número no debe influir en el siguiente número generado.

5. Pruebas Estadísticas para los Números Pseudo Aleatorios

A. Pruebas de Independencia

Las **pruebas de independencia** verifican si los números generados por un algoritmo no tienen correlación entre ellos. Una forma común de hacer esto es mediante el cálculo de la autocorrelación entre los números generados.

Ejemplo:

Se generan 100 números pseudo aleatorios y se calcula la correlación entre los valores consecutivos. Si la correlación es baja, esto sugiere que los números son independientes.

B. Pruebas de Uniformidad

Las **pruebas de uniformidad** aseguran que los números están distribuidos uniformemente en el rango especificado. Una de las pruebas más comunes es la **prueba de chi-cuadrado**, que compara las frecuencias observadas de los números generados con las frecuencias esperadas.

Ejemplo práctico

Generamos una secuencia de 1000 números y dividimos estos números en 10 grupos. Comprobamos si cada grupo contiene aproximadamente 100 números. Si no es así, el generador podría no ser lo suficientemente uniforme.

Bibliografía:

1. **Knuth, D. E.** (1998). *The Art of Computer Programming, Volume 2: Seminumerical Algorithms* (3rd ed.). Addison-Wesley.
2. **Gentle, J. E.** (2003). *Random Number Generation and Monte Carlo Methods*. Springer.
3. **L'Ecuyer, P.** (2014). *An Introduction to the Mathematics of Financial Derivatives*. Academic Press.
4. **Marsaglia, G., & Zaman, A.** (1991). *Towards a Universal Random Number Generator*. *ACM Transactions on Mathematical Software*, 3(3), 439-445.