SUMA 20
noviembre 1995, pp. 91-98

Generación de números aleatorios

Vicente Trigo Aranda

Este artículo se centra en el estudio de los métodos más usuales implementados en el software informático para generar números aleatorios. El análisis de dichos algoritmos se acompaña de una panorámic de su evolución histórica y de tres programas en Turbo Pascal que permiten al lector comprobar determinados aspectos de la exposición teórica.

MISCELÁNEA

La gran mayoría de los programas de aplicación científica: lenguajes de programación, hojas de cálculo, paquetes estadísticos, etc., llevan incorporadas una o varias funciones (del tipo RND, RANDOM, ALETU, etc.) que permiten al usuario obtener números aleatorios. La principal utilidad práctica de estos números radica en la simulación de sistemas, aunque no es despreciable su importancia en juegos, sorteos, etc., ni en temas mucho menos prosaicos, como la mecánica cuántica o la teoría de la evolución.

Debe quedar claro que el tema a tratar en este artículo se centra en un aspecto muy concreto y específico: el estudio de los métodos más usuales implementados en el software informático para generar números aleatorios. Es decir, mi objetivo es mostrar al lector una panorámica de cómo los programas de ordenador más comunes «eligen al azar» sin pretender, ni mucho menos, que el aficionado a la informática deba generar sus propios números aleatorios. Por tanto, en el enfoque prima más lo divulgativo que lo práctico.

Otro tema, sin duda mucho más apasionante, es el de utilizar estos números generados aleatoriamente para resolver problemas prácticos, mediante simulación. Por razones de espacio no puedo tratarlo ahora pero, si resulta de interés para los lectores de SUMA, en posteriores números presentaré algún ejemplo de las múltiples ventajas que ofrece la simulación para resolver problemas de toda índole.

¿Qué se entiende por números aleatorios?

No es fácil decidir cuándo una secuencia de números es aleatoria y, de hecho, es un problema que entra de lleno en el terreno filosófico. Así por ejemplo, en palabras de Martin Gardner, «una sucesión de dígitos absolutamente desordenada es un concepto lógicamente contradictorio». Más concreta pero, paradójicamente, a la vez más abstracta, es la definición de aleatoriedad de Kolmogorov: «la longitud del programa más corto que diga a una máquina de Turing como escribir la sucesión de dígitos».

Si nos movemos en un terreno más práctico, se suele admitir que una sucesión de números es aleatoria cuando cumple las dos siguientes condiciones, propuestas por D. H. Lehmer:

1.ª Una persona que desconozca la norma de generación no puede predecir el siguiente término de la sucesión.

2.ª Si analizamos la sucesión mediante cualquier tests de aleatoriedad (serial, rachas, gaps, máximo, CSCS, etc.) lo supera sin problemas.

Lamentablemente, y por razones de espacio, tengo que soslayar ahora el estudio y análisis de los diversos tests de aleatoriedad y uniformidad que analizan la calidad de los números aleatorios generados.

Un último detalle a tener en cuenta. Nuestra meta será encontrar una sucesión de variables aleatorias independientes distribuidas uniformemente en el intervalo [0,1). El motivo por el cual centramos nuestro estudio en este intervalo es porque a partir de una uniforme [0,1) disponemos de algoritmos matemáticos para generar otras variables aleatorias, ya sean discretas (binomial, Poisson, etc.) o continuas (exponencial, normal, gamma, etc.).

Tengamos presente, además, que aunque en teoría los números aleatorios [0,1) pueden tener infinitas cifras decimales, en la práctica nos tenemos que conformar con un número finito. Una posible forma de generarlo (utilizada en simulación manual) será elegir una serie finita de dígitos aleatorios y considerarla como la parte decimal del número aleatorio; otra posibilidad (utilizada en simulación por ordenador) será elegir un número entero del rango 0..m-1 y dividirlo después por m.

Tablas de números aleatorios

Cuando a comienzos de siglo se descubre la importancia del azar se comienza a estudiar la obtención de números aleatorios. Enseguida se desecharon los métodos primarios, como lanzar dardos a una diana o anotar dígitos uno tras otro según nos parece, porque presentan claros sesgos que los invalidan, y se vio la necesidad de construir tablas de dígitos realmente aleatorios.

...se desecharon los métodos primarios, como lanzar dardos a una diana o anotar dígitos uno tras otro según nos parece, porque presentan claros sesgos que los invalidan

En 1927 aparece la primera publicación de este tipo. L. H. C. Tippett cogió el censo de parroquias inglesas y tomó la cifra central de su superficie; su tabla constaba de 41.600 dígitos. Doce años después M. G. Kendall y B. Babington construyeron, con ayuda de una ruleta dividida en diez partes, cien mil dígitos. En 1954 Royo y Ferrer editaron su *Tablas de números aleatorios y estadísticos* con un cuarto de millón de dígitos.

La cumbre de este tipo de tablas se alcanzó con el libro *A million random digits* de Rand Corporation, publicado en 1955. Los dígitos se obtuvieron mediante dispositivos electrónicos que producían cifras binarias aleatorias; tras eliminar los sesgos detectados se transformaban en números decimales, se sumaban por parejas y se seleccionaba el último dígito.

Este tipo de libros, como señala Alfred Bork, son propios de nuestro siglo. Es evidente que nunca antes en la historia del pensamiento científico había tenido ningún sentido su elaboración. ¿Qué habrían pensado Newton o Gauss de unos libros cuyas páginas sólo contenían dígitos y más dígitos?

Aparecen los ordenadores

Con la llegada de los ordenadores estas tablas, que servían para un uso manual, perdieron su importancia. Por un lado su almacenamiento en la memoria del ordenador sería ciertamente costoso; baste pensar en el tiempo que se emplearía en cargarlos y en el espacio que ocuparían. Por otro lado, y aunque resulte sorprendente para mucha gente, un millón de dígitos es un número insuficiente para bastantes simulaciones. Por cierto, y para terminar con este tema de las tablas de números aleatorios, Warren Weaver descubrió que en cualquiera de ellas los dígitos más pequeños aparecen con mayor frecuencia.

Los matemáticos se pusieron a buscar algoritmos matemáticos para generar números aleatorios de forma que, mediante una simple fórmula matemática, pudiesen obtenerse millones y millones de estos números. De esta forma el costo computacional de generación sería ciertamente bajo; además se tendría la ventaja adicional de poder reproducir la sucesión estudiada en cualquier momento, lo que facilita enormemente la tarea de depuración de los programas de simulación.

Los métodos más prácticos resultan ser los aritméticos; es decir, aquellos en que cada término de la sucesión se obtiene mediante una expresión aritmética que depende del término anterior:

$$X_{n+1} = f(X_n)$$

y el que da origen a la sucesión, X_0, se denomina *semilla*.

En realidad estos números no serán aleatorios y sería más exacto calificarlos como «pseudoaleatorios», pero si la sucesión satisface las dos condiciones anteriores fijadas por Lehmer, en la práctica puede ser considerada como aleatoria.

Históricamente el primer método fue propuesto por John von Neumann en 1946 y se conoce como «Método del centro del cuadrado». En él se toma como semilla un número de 2n dígitos y se eleva al cuadrado; como éste tendrá 4n dígitos (si es preciso se añaden ceros a la izquierda) se toma como siguiente término el formado por las 2n cifras centrales.

Por ejemplo, si $2n = 4$ y $X_0 = 1996$ se tendrá la sucesión:

$X_0 = 1996$	$X_0^2 = 03\ 9840\ 16$
$X_1 = 9840$	$X_1^2 = 96\ 8256\ 00$
$X_2 = 8256$	$X_2^2 = 68\ 1615\ 36$
$X_3 = 1615$	$X_3^2 = 02\ 6082\ 25$

El programa CENTROCUA adjunto codifica este método con números de cuatro cifras y, ejecutándolo, puede comprobarse que resulta poco eficiente. Basta considerar semillas como, por ejemplo, 3317 o 3792.

Vista la poca utilidad de este método Lehmer propuso otro del mismo estilo. Se parte de una semilla X_0 de n cifras y un multiplicador k de m cifras y se calcula el producto, que tendrá n + m cifras; se tomará como siguiente número de la sucesión el resultante de restar a las n últimas cifras del producto las m primeras.

...el primer método fue propuesto por John von Neumann en 1946 y se conoce como «Método del centro del cuadrado».

PROGRAMA CENTROCUA

```
PROGRAM Centro_del_cuadrado;

USES CRT;

VAR  num,r,n:LONGINT;
     ch:CHAR;

BEGIN
   REPEAT
      CLRSCR;
      GOTOXY(22,12);
      WRITE('Introduce la semilla (4 cifras):  ');
      READ(num);
      REPEAT
         n:=0;
         CLRSCR;
         GOTOXY(23,1);
         WRITE('Números generados a partir de  ',num:4);
         GOTOXY(1,3);
         REPEAT
            num:=num*num;
            r:=num MOD 100;
             num:=((num-r) DIV 100) MOD 10000;
             IF (n MOD 10)<>0 THEN
               WRITE(num:6,'  ')
            ELSE
               BEGIN
                  WRITELN;
                  WRITE(num:6,'  ');
               END;
            n:=n+1;
         UNTIL (n=100);
         GOTOXY(20,25);
         WRITE('Repetir (R), Continuar (C), Terminar (T) ?');
         REPEAT
            ch:=READKEY;
         UNTIL (ch IN ['c','C','t','T','r','R']);
      UNTIL NOT(ch IN ['c','C']);
   UNTIL (ch IN ['t','T']);
   CLRSCR;
END.
```

PROGRAMA LEHMER

```
PROGRAM Primer_metodo_de_Lehmer;

USES CRT;

VAR  ch:CHAR;
  num,r,n,k:LONGINT;

BEGIN
   REPEAT
       CLRSCR;
       GOTOXY(22,12);
       WRITE('Introduce la semilla:  ');
       READ(num);
       GOTOXY(22,15);
       WRITE('Introduce el multiplicador:  ');
       READ(k);
       REPEAT
           n:=0;
           CLRSCR;
           GOTOXY(12,1);
           WRITE('Números generados a partir de  ',num:4);
           WRITE('Multiplicador ':20,k);
           GOTOXY(1,3);
           REPEAT
               num:=k*num;
               num:=(num MOD 10000) - (num DIV 10000);
               IF (n MOD 10)<>0 THEN
                      WRITE(num:6,'  ')
                   ELSE
                       BEGIN
                           WRITELN;
                           WRITE(num:6,'  ');
                       END;
               n:=n+1;
           UNTIL (n=100);
           GOTOXY(20,25);
           WRITE('Repetir (R), Continuar (C), Terminar (T) ?');
           REPEAT
               ch:=READKEY;
           UNTIL (ch IN ['c','C','t','T','r','R']);
       UNTIL NOT(ch IN ['c','C']);
   UNTIL (ch IN ['t','T']);
   CLRSCR;
END.
```

Por ejemplo, si k = 57 y X_0 = 2345 se tendrá la sucesión:

$kX_0 = 13\ 3665 \Rightarrow X_1 = 3665 - 13 = 3652$

$kX_1 = 20\ 8164 \Rightarrow X_2 = 8164 - 20 = 8144$

$kX_2 = 46\ 4208 \Rightarrow X_3 = 4208 - 46 = 4162$

$kX_3 = 23\ 7234 \Rightarrow X_4 = 7234 - 23 = 7211$

Como este algoritmo tampoco es muy útil para obtener números aleatorios (puede comprobarse con el programa LEHMER adjunto), Lehmer continuó su búsqueda y en 1948 propuso el método que vemos seguidamente.

Método congruencial lineal

Dado un número entero positivo m (módulo), la sucesión de números aleatorios se generará mediante la fórmula:

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

donde a (multiplicador), c (incremento) y X_0 son naturales inferiores a m.

Por ejemplo, si partimos de los valores:

$m = 17 \quad a = 7 \quad c = 1 \quad X_0 = 3$

la sucesión obtenida será: 5, 2, 15, 4, 12, 0, 1, 8, 6, 9,....

Es evidente que tarde o temprano alguno de los números de la sucesión tendrá que repetirse, entrándose así en un bucle cíclico, cuya longitud se denomina *período*. Así, en el ejemplo anterior puede comprobarse que el período es 16; en cambio si la semilla hubiese sido X_0 = 14 el período sería 1.

Como es lógico, los cuatro parámetros de partida determinan la calidad de la sucesión y es casi un arte elegirlos adecuadamente. Por una parte es conveniente que el período de la sucesión sea lo mayor posible y por otra que sus términos tengan una cierta calidad aleatoria; es decir, que superen una serie de tests empíricos.

Si nos centramos en lo referente al máximo período, el valor del módulo parece sensato fijarlo en función del ordenador; así, si éste trabaja con palabras de 2^{32} bits parece aconsejable tomar precisamente m = 2^{32}. En cuanto a los parámetros, *a* y *c*, se eligen de acuerdo con el siguiente resultado:

Teorema: La sucesión congruencial lineal tiene período máximo, e igual a m, si y sólo si se cumple:

i) c y m son primos entre sí.
ii) a–1 es múltiplo de todos los primos que dividen a m.
iii) Si m es múltiplo de 4, a–1 también lo es.

Tan útil y cómodo de codificar es este método ideado por Lehmer que resulta numerosísimo el software basado en él para simular el azar. Así, por ejemplo, en Turbo Pascal el generador de números aleatorios de tipo REAL sigue un algoritmo de congruencia lineal con los siguientes parámetros:

$m = 2^{32} \quad c = 1 \quad a = 134.775.813$

También se emplea este método en la rutina MTH$RANDOM de la librería VAX/VMS, con los valores:

$m = 2^{32} \quad c = 1 \quad a = 69.069$

El tercer programa adjunto, CONGRUEN, codifica este algoritmo de congruencia lineal y permite trabajar con datos de tipo LONGINT. Además, si el usuario así lo decide, estudia si los datos introducidos satisfacen las tres condiciones del teorema anterior, generándose de esta forma una secuencia de período máximo.

Generadores multiplicativos puros

El método multiplicativo puro es un caso particular del algoritmo congruencial, en donde el incremento se toma nulo. Por tanto, en estos generadores la expresión recurrente podrá escribirse en la forma:

$$X_{n+1} = a^n X_0 \mod m$$

Es evidente que, de acuerdo con el teorema anterior, el período no podrá ser nunca *m* pero, a cambio, la rapidez de generación de estos números será mayor. Además, con un poco de cuidado puede obtenerse un período igual a m–1, lo que en la práctica es lo mismo que si fuese *m*. Para ello hay que basarse en el siguiente resultado:

Tan útil y cómodo de codificar es este método ideado por Lehmer que resulta numerosísimo el software basado en él para simular el azar.

PROGRAMA CONGRUEN

```
PROGRAM Algoritmo_de_congruencia_lineal;
{$N+,E+}

USES CRT;

VAR a,c,m,x,semilla:LONGINT;
   decimales:EXTENDED;
   contador:1..80;
   tecla,pulsar:CHAR;

FUNCTION primo(x:LONGINT):BOOLEAN;
VAR z:LONGINT;
BEGIN
   primo:=TRUE;
   IF (x MOD 2 = 0) AND (x>2) THEN primo:=FALSE
     ELSE
       FOR z:=1 TO TRUNC(SQRT(x)/2) DO
         IF x MOD (2*z+1)=0 THEN primo:=FALSE;
END;

FUNCTION primos_entre_si(u,v:LONGINT):BOOLEAN;
VAR a1,a2,r:LONGINT;
BEGIN
   IF (u=1) OR (v=1) THEN primos_entre_si:=TRUE
     ELSE
       BEGIN
         IF u>=v THEN BEGIN a1:=u;a2:=v; END
           ELSE BEGIN a1:=v;a2:=u; END;
         REPEAT
           r:= a1 MOD a2;
           a1:=a2;
           a2:=r;
         UNTIL r=0;
         IF a1=1 THEN primos_entre_si:=TRUE
           ELSE primos_entre_si:=FALSE;
       END;
END;

PROCEDURE periodo;
VAR p:LONGINT;
BEGIN
   IF primo(m) THEN WRITE('PERIODO NO MAXIMO') ELSE
   IF (m MOD 4=0) AND ((a-1) MOD 4 <>0) THEN
                  WRITE('PERIODO NO MAXIMO')
   ELSE IF primos_entre_si(m,c)=FALSE THEN
                  WRITE('PERIODO NO MAXIMO')
     ELSE
       BEGIN
         FOR p:=2 TO TRUNC(SQRT(m)) DO
           BEGIN
             IF (m MOD p=0) AND (primo(p)) THEN
               BEGIN
                 IF ((a-1) MOD p)<>0 THEN
                   BEGIN
                     WRITE('PERIODO NO MAXIMO');
                     EXIT;
                   END;
```

```
                END;
            IF (m MOD p=0) AND (primo(m DIV p)) THEN
                BEGIN
                    IF ((a-1) MOD (m DIV p))<>0 THEN
                        BEGIN
                            WRITE('PERIODO NO MAXIMO');
                            EXIT;
                        END;
                END;
            END;
        WRITE('PERIODO MAXIMO');
    END;
END;

BEGIN
    REPEAT
        {Introducción de datos}
        CLRSCR;
        GOTOXY(10,8);WRITE('¿Valor de a? ');READLN(a);
        GOTOXY(10,11);WRITE('¿Valor de c? ');READLN(c);
        GOTOXY(10,14);WRITE('¿Valor de m? ');READLN(m);
        GOTOXY(10,17);WRITE('¿Semilla? ');READLN(semilla);
        REPEAT
            GOTOXY(10,20);
            WRITE('¿Deseas saber si alcanza periodo máximo?
(S/N) ');CLREOL;
            GOTOXY(57,20);pulsar:=READKEY;
        UNTIL pulsar IN ['s','S','n','N'];

        IF UPCASE(pulsar)='S' THEN
            BEGIN
                periodo;
            REPEAT UNTIL KEYPRESSED;
        END;

        {Cálculo de la secuencia aleatoria}
        x:=semilla;
        REPEAT
            CLRSCR;
            GOTOXY(10,2);WRITE('a = ',a);
            GOTOXY(35,2);WRITE('c = ',c);
            GOTOXY(60,2);WRITE('m = ',m);
            GOTOXY(1,4);
            FOR contador:=1 TO 80 DO
                BEGIN
                    WRITE(x:9,' ');
                    IF contador MOD 8=0 THEN WRITELN;
                    decimales:=FRAC((1.0*a*x+c)/m);
                    x:=ROUND(decimales*m);
                END;
            GOTOXY(20,24);
            WRITE('Repetir (R), Continuar (C), Terminar (T) ? ');
            REPEAT
                tecla:=READKEY;
            UNTIL (tecla='R') OR (tecla='C') OR (tecla='T')
                OR (tecla='r') OR (tecla='c') OR (tecla='t');
        UNTIL (tecla<>'C') AND (tecla<>'c');
    UNTIL (tecla='T') OR (tecla='t') ;
    CLRSCR;
END.
```

Teorema: Consideremos una sucesión congruencial lineal de módulo m, número primo mayor que 2, e incremento c nulo. Tendrá período máximo, e igual a m–1, si para todos los divisores primos q de m–1 se satisface que $a^{(m-1)/q}$ no es congruente con 1 módulo m.

Por ejemplo, si tomamos los siguientes parámetros:

$$m = 2^{16} + 1 = 65.537 \qquad a = 75$$

podemos comprobar que se satisfacen las condiciones del teorema anterior y, por tanto, su período será 65.536:

Es sencillo ver que m es primo y como el único divisor primo de m–1 es 2, basta con calcular el valor de ($75^{32.768}$ mod 65.537). Ya se haga manualmente o con ayuda del ordenador, es fácil obtener que el resto de esa potencia no es 1, sino 65.536.

Curiosamente éste era el generador de números aleatorios que llevaba implementado el BASIC del popular y querido ZX-Spectrum.

Pero los generadores multiplicativos puros no sólo se empleaban en aquellos entusiastas años del nacimiento de la microinformática. En la actualidad también hay muchos programas que los utilizan; así, por ejemplo, lo podemos encontrar en la librería de rutinas matemáticas y estadísticas IMSL, con los parámetros:

$$m = 2^{31} - 1 = 2.147.483.647$$
$$a = 950.706.376$$

Por último, indicar que cuando el módulo m no es primo se acostumbra trabajar con valores que sean potencias de 2 o de 10. En el primer caso ($m = 2^z$, con $z > 3$) el período máximo posible es m/4 y exige que a sea 3 o 5 módulo 8. Si $m = 10^z$, con $z > 4$, y la semilla no es múltiplo de 2 o 5 el período máximo es $5 \cdot 10^{z-2}$ si y sólo si a módulo 200 es alguno de los valores siguientes: 3, 11, 13, 19, 21, 27, 29, 37, 53, 59, 61, 67, 69, 77, 83, 91, 109, 117, 123, 131, 133, 139, 141, 147, 163, 171, 173, 179, 181, 187, 189 y 197.

Epílogo

Existen muchos otros métodos en la literatura científica para generar números aleatorios (congruencias aditivas, lineales generales, mezcla de varias, etc.) pero, hoy por hoy, los congruenciales lineales continúan siendo los predominantes en la industria informática ya que, con parámetros apropiados (y todos los comentados anteriormente lo son), se pueden generar en poco tiempo sucesiones formadas por miles de millones de números aleatorios que superan sin problemas los test empíricos de aleatoriedad y uniformidad.

A estas alturas, y para concluir este breve paseo por el mundo de la implementación del azar en los ordenadores, quizás lo más apropiado sea citar la célebre frase del matemático R. Coveyou: «La generación de números aleatorios es demasiado importante para dejarla en manos del azar».

Vicente Trigo
IB Félix de Azara
Zaragoza

Bibliografía

AGUARON, J. y otros (1993): *Simulación*, Servicio de Publicaciones de la Universidad de Zaragoza, Zaragoza.

GARDNER, M. (1975): *Carnaval matemático*, Alianza Editorial, Madrid.

KNUTH, D. E. (1969): *The art of computer programming, II*, Addison Wesley, Massachusetts.

TRIGO, V. y R. AGUSTÍN (1994): *Prácticas de Estadística por Ordenador*, Librería Belén, Zaragoza.

MATERIAL DIDÁCTICO DEL PROFESORADO EN SUS AULAS

La FESPM ha acordado presentar una exposición de materiales educativos realizados por profesores y profesoras de la Federación en el ICME-8 que se celebrará en Sevilla en julio de 1996. Ha sido nombrado coordinador de esta actividad Arturo Mandly Manso de la Sociedad Extremeña.

Todos los socios de las sociedades federadas que lo deseen pueden participar de acuerdo con las siguientes condiciones:

- No se considerará como material didáctico el que se limite a una mera descrición de actividades.
- Los programas informáticos que desarrollen actividades didácticas para el aula también podrán incluirse, para lo cual deberán acompañar las condiciones mínimas que debe cumplir el ordenador sobre el que debe rodar la aplicación.
- Se especificarán los autores y autoras y la sociedad a la que pertenecen.
- Se incluirá un valor aproximado de los materiales, dado que se realizará un seguro de la exposición.
- Junto con cada material debe presentarse una guía de utilización y descripción del material, así como el nivel educativo al que va dirigido y tres descriptores. Se utilizará formato DIN A4. es obligatorio presentarlo en castellano y, si es posible, también en inglés.
- Si el material necesitase para su funcionamiento el estar conectado a la red eléctrica, u otro tipo de necesidades, deberá especificarse claramente a efectos de tenerlo previsto a la hora de su ubicación en la exposición.
- Los materiales y la documentación serán enviados a la dirección particular infraescrita antes del 29 de febrero de 1996.
- Una vez finalizado el ICME-8 los materiales serán devueltos.

Dirección particular:
Arturo Mandly Manso
Plaza de las Albercas, 8, 2.° C
06400-DON BENITO (Badajoz)
Tno.: (924) 81 05 47

Dirección del centro de trabajo:
IES Cuatro Caminos
Calle Torres Isunza, s/n
06400-DON BENITO (Badajoz)
Tno.: (924) 81 07 02
Fax: (924) 81 14 74

NUEVA DIRECCIÓN

Revista SUMA
ICE Universidad de Zaragoza
C. Pedro Cerbuna, 12
50009-ZARAGOZA

Tno.: (976) 76 13 49
Fax: (976) 76 13 45