Simulação: geração de números pseudo-aleatórios

A ideia de gerar sequências aleatórias através dum *algoritmo numérico* pode parecer inteiramente inadequada [Ecker & Kupferschmid, 1988]. Os algoritmos do género que se podem implementar num programa de computador são, por definição, completamente determinísticos e nunca podem, portanto, produzir um efeito aleatório de qualquer tipo. Se um computador estiver a funcionar normalmente, oferece precisamente o mesmo resultado de cada vez que executa uma operação sobre os mesmos dados. Qualquer sequência de números gerados por um algoritmo terá, pois, a propriedade de que u_{k+1} pode ser determinado com certeza a partir do conhecimento de u_1, u_2, …, u_k, e não será, portanto, aleatória, no sentido pretendido. Como pode então tal comportamento, perfeitamente previsível, ser usado para produzir números utilizáveis numa simulação ?

A resposta está em que, na simulação, estamos interessados apenas em estatísticas de longo prazo. Sob condições adequadas, uma sequência de números gerados por um algoritmo pode ter as mesmas propriedades estatísticas que uma sequência gerada por um processo aleatório natural.

Uma sequência de números que seja gerada por um algoritmo mas seja intermutável com outra sequência verdadeiramente aleatória é chamada *pseudo-aleatória*.

> Uma sequência de números é **pseudo-aleatória** se qualquer subsequência suficientemente curta for intermutável com uma sequência comparável realmente aleatória.

A única diferença inevitável entre uma sequência aleatória e uma pseudo--aleatória é que esta última se pode repetir — e isto justifica a expressão "suficientemente curta" na definição. O período de repetição do algoritmo deve, assim, ser mais longo que a sequência de números necessária à simulação.

Geradores de números aleatórios

Feita a distinção entre sequências produzidas por processos aleatórios naturais e sequências pseudo-aleatórias produzidas por algoritmos, é convencional usar uma terminologia que ignora essa distinção. Assim, um algoritmo para geração duma sequência de números pseudo-aleatórios é habitualmente chamado simplesmente **gerador de números aleatórios**, sendo comum chamar **números aleatórios** aos números das sequências pseudo-aleatórias, assim como **número aleatório** a *um* elemento da sequência, embora o conceito de aleatoriedade se aplique verdadeiramente à sequência como um todo.

Há muitos algoritmos para gerar números aleatórios, diferindo na medida em que exibem as seguintes propriedades desejáveis:

- Comprimento do período de repetição
- Independência estatística aparente de números sucessivos
- Distribuição uniforme

- Rapidez de obtenção
- Repetibilidade (reprodutibilidade)

As propriedades citadas são dificilmente conciliáveis, pelo que a selecção dum gerador conveniente para uma dada simulação pode ser uma tarefa não-trivial.

1. O algoritmo congruencial multiplicativo

É o mais simples gerador de números aleatórios.

> **0**. Fixe os parâmetros p e m
> **1**. Seja u_0 o número inicial, não-nulo e ímpar; $k = 0$
> **2**. $u_{k+1} = (m\, u_k) \bmod p$
> **3**. $k \leftarrow k + 1$; goto **2**

Todos os números intervenientes no algoritmo são inteiros não-negativos. O parâmetro m é o **multiplicador**, p é o **módulo** e o número u_0 é a **semente**. O passo principal do algoritmo é a **fórmula de recorrência** do *Passo 2*, sendo, como se sabe,

$x \bmod p$ = resto da divisão (inteira) de x por p

Se nenhum u_k for zero, os inteiros que se podem gerar vão de 1 a $p-1$, logo $(u_k - 1)$ vai de 0 a $p-2$ e os correspondentes números reais r_k do intervalo [0, 1] obter-se-ão por $r_k = \dfrac{u_k - 1}{p-2}$. Para ver como o algoritmo funciona, considere-se o seguinte exemplo, arbitrando-se $p = 17$, $m = 5$ e $u_0 = 7$:

$u_1 = (5 \times 7) \bmod 17$ = resto da divisão inteira de $35/17$ = **1**
$u_2 = (5 \times \mathbf{1}) \bmod 17 =$ **5**
$u_3 = (5 \times \mathbf{5}) \bmod 17 =$ **8**

Continuando o processo, obtém-se:

7, 1, 5, 8, 6, 13, 14, 2, 10, 16, 12, 9, 11, 4, 3, 15, $^{(R)}$ 7, 1, 5, 8, …

Após R, verifica-se repetição, pelo que o período foi de 16 números. (Dir-se-á adiante que $p = 17$ foi uma escolha feliz.) O período é sempre menor que o parâmetro p, logo p deve ser escolhido tão grande quanto possível. Devido a considerações de eficiência computacional, se o comprimento de "palavra" for w bits e fizermos $p = 2^w$, a multiplicação na fórmula de recorrência dá um resultado com comprimento de "palavra dupla", sendo a palavra de mais baixa ordem o produto módulo 2^w. Outra escolha corrente para p é 2^{w-1}. Em qualquer dos casos, o facto de nenhuma divisão (propriamente dita) ser necessária para efectuar a operação de "módulo" dá vantagem sobre outros valores de p.

É possível que o período seja muito menor que p, se o multiplicador m for uma escolha "infeliz" (veja-se $p = 10$, $m = 5$, $u_0 = 5$). Para obter o período máximo, que é $p-1$, m deve ser escolhido de modo a que $m-1$ seja múltiplo de todos os números primos submúltiplos de p. Por outro lado, para que a sequência gerada tenha boas propriedades estatísticas, m deve ser próximo de $\sqrt{p}$ e **a sequência usada numa simulação *não deve ser mais longa* que a raiz quadrada do período de repetição**.

O gerador congruencial multiplicativo mais vulgar, disponível em Fortran como sub-rotina RANDU, tem $p = 2^{31} = 2\,147\,483\,648$ (note-se que este inteiro excede em 1 o máximo habitualmente representável) e $m = 2^{16} + 3 = 65\,539$. Estas escolhas dão um período de 2^{29} (*ca.* 10^9) números e o algoritmo corre rapidamente em computadores com um comprimento de palavra de 32 bits. No entanto, as sequências produzidas têm propriedades estatísticas ***inconvenientes*** para muitas situações.

2. Outros geradores de números aleatórios uniformes

Obtêm-se propriedades estatísticas melhoradas adicionando uma constante à fórmula de recorrência congruencial multiplicativa:

$$u_{k+1} = (a + m\,u_k) \bmod p$$

O número *a* é o **incremento** e um algoritmo com esta fórmula da recorrência chama-se um gerador **congruencial linear** ou **congruencial misto**. Um destes algoritmos que apresenta *boas propriedades estatísticas* usa:

$p = 2\,147\,483\,648$ $= 2^{31}$

$m = 843\,314\,861$ [☺ $p\,(p / 8) + 5 =^{?} 843\,314\,861{,}5$]

$a = 453\,816\,693$ [☺ $p\left(3-\sqrt{3}\right)/6 = 453\,816\,692{,}9$]

Outro método com boas propriedades estatísticas é o **algoritmo GPSS**. Ilustrar-se-á com um exemplo de inteiros decimais, embora seja realmente efectuado em aritmética binária em computador. Suponhamos que se pretendem gerar números r_k entre 0,0000 e 0,9999. Começamos com um multiplicador *m*, fixo, e uma semente u_0:

$$m = 5167$$

$$u_0 = 3729$$

O produto terá oito dígitos.

$$m\,u_0 = 19\,267\,743 \qquad (19\;\;2677\;\;43)$$

Os quatro dígitos *centrais* formarão o primeiro número aleatório r_1 e os quatro dígitos *da direita* serão usados como u_1. Assim,

$$r_1 = 0{,}2677$$
$$u_1 = 7743$$

Em seguida, é formado o produto $m\,u_1$, extraídos os seus quatro dígitos centrais como dígitos de r_2 e os quatro dígitos da direita como u_2 e assim por diante.

Um algoritmo aparentado, o do **quadrado do meio** (“mid-square algorithm”), muito simples mas de propriedades insatisfatórias, é o que se exemplifica adiante. Seja $r_1 = 9876$; r_2 será formado pelos quatro algarismos centrais do número de oito algarismos que é r_1^2 (eventualmente completado à esquerda com zeros).

r_2 = {4 algarismos centrais de r_1^2 } = {4 alg. c. de 97 5353 76} = 5353

r_3 = {4 algarismos centrais de r_2^2 } = {4 alg. c. de 28 6546 09} = 6546

Além da questão da repetição de período, poderá chegar-se a uma situação como a seguinte (se fosse $r_k = 2001$)

r_{k+1} = {4 algarismos centrais de r_k^2 } = {4 alg. c. de 04. 004.0 01} = 40

r_{k+2} = {4 alg. c. de 00 001.6 00} = 16

r_{k+3} = {4 alg. c. de 00 0002 56} = 2

r_{k+4} = {4 alg. c. de 00 0000 04} = 0

e daí em diante sempre 0.

3. Aplicação a uma distribuição especificada

Na situação mais frequente de a variável aleatória a simular não seguir a distribuição uniforme, a técnica a utilizar é a seguinte, aqui ilustrada com uma distribuição exponencial de parâmetro *c*.

$$f(x) = c\,\mathrm{e}^{-cx} \qquad \text{com} \quad 0 \le x \le 1$$

$$F(x) = \int_0^x f(t)\,\mathrm{d}t = 1 - \mathrm{e}^{-cx}$$

pois que, obviamente, $F(0) = 0$. Invertendo a função, vem

$$x = \frac{\ln(1-F)}{-c}$$

Fazendo $F = u$, com u número aleatório uniforme, será

$$x = \frac{\ln(1-u)}{-c}$$

ou —visto que, se u é uniforme em [0, 1], $1-u$ também o é—, mais simplesmente,

$$x = \frac{\ln u}{-c}$$

Assim, conhecido *c*, obtida uma série de números aleatórios *u* (uniformes), dispor-se-á duma série de números aleatórios *x*, com uma distribuição exponencial especificada.

▪ □ ▪