第3章

随机数生成器

3.1 本章引言

随机数(RNs)是任何蒙特卡罗模拟的重要组成部分。任何蒙特卡罗模拟的质量都取决于所使用随机数的质量(或随机性)。如果随机数均匀分布,则实现了高度的随机性。因此,需要设计出一种方法,生成随机的数字序列,在重复之前有很长的周期,并且不需要耗费大量资源。

计算机上随机数生成器的早期实现可以追溯到冯·诺依曼在与曼哈顿计划(1941—1945 年)相关的蒙特卡罗模拟中使用了随机数生成器。从那时起,许多生成随机数的方法[15,40,58-59,62-64,69-70]被开发出来。L'Ecuyer 指出,一般的计算机用户通常认为计算机生成均匀随机数的问题已经得到解决。尽管在开发"好"生成器方面取得了重大进展,但仍有"坏"或"不合适"的生成器,并产生了不良结果。因此,任何蒙特卡罗用户都应该清楚这些问题及其局限性。

完整起见,本章介绍了实验和算法技术,并讨论了随机性检验的选择,通过实例证明了随机数生成器的随机性和周期高度依赖几个参数的"正确"选择。这些例子表明,参数的微小变化会对估计的随机数产生重大影响。

本章将回顾用于确定随机数的实验和算法,检查随机数生成器的行为,回顾几个随机性检验,并详细说明参数对随机数序列长度和生成随机数随机性的影响。

3.2 随机数生成方法

产生随机数的常见方法有两种:①实验;②算法。

1. 实验(查表,在线)

实验(物理过程)用于生成一系列随机数,这些随机数以表的形式保存在计算机内存中。例如:①抛硬币或掷骰子;②从瓮中抽球,即抽彩票;③旋转第 2 章中介绍的标记圆盘;④在飞镖游戏中从中心测量飞镖的位置。Frigerio 和 Clark[36]

提出了一种方法，记录放射性物质在给定时间间隔内的衰变次数，例如，计数20 ms，如果数字是奇数，则记录0位(bit)，否则记录1位，然后形成31位的数字，这个过程在1 h内产生了不到6000个数字。显然，在蒙特卡罗模拟中使用这种方法的速度是相当慢的。(阿贡国家实验室(ANL)程序中心有一盘包含2.5×10^6个随机数的磁带。)另一种技术是监控计算机部件(如计算机内存)以生成随机数。注意，这是在执行实际模拟时完成的。

2. 算法(或确定性方法)

算法用于生成随机数。这种方法由于其确定性的本质，通常被称为伪随机数生成器(PRNG)，所生成的数字被称为伪随机数(PRNs)。

每种方法都有其优缺点，这直接影响到对方法的使用。选择“正确”方法有如下6个重要因素。

(1) 随机性：随机数应该是真正随机的；也就是说，它们应该均匀分布。在实验方法中，如果过程遵循均匀分布，则实现随机性。在算法方法中，生成的序列必须满足几个统计检验。

(2) 再现性：为了测试模拟算法或进行敏感性/扰动研究，有必要能够再现随机数序列。

(3) 随机数序列的长度：为了对实际工程问题进行蒙特卡罗模拟，需要数百万个随机数；因此，生成器必须能够产生大量的随机数。

(4) 计算机内存：生成器不应占用过多的计算机内存。(请注意，这个问题可能对于下一代计算机而言并不重要。)

(5) 生成时间：生成随机数序列所需的时间(工程师/分析师)不是很重要，可以是几天或者几个月。

(6) 计算机时间：生成数字所需的计算机时间应该明显短于实际模拟的时间。

表3.1基于上述因素对实验随机数生成器和算法随机数生成器进行了比较。

表3.1　实验随机数生成器与算法随机数生成器的比较

要　素	实　验		算　法
	查表法	在线法	
随机性	好	好	待检验
再现性	可以	不可以	可以
周期*	有限	无限	有限
计算机内存	大	小	小
生成时间**	长	无	无
计算机时间	短	长	短

* 随机数序列的最大长度。

** 生成随机数所需时间。

在实践中，算法方法是首选，主要是因为它的序列是可重复的，并且只需要最小的努力（计算机资源，工程师时间）。在使用任何算法生成器时，都必须进行一系列随机性检验，并确定/测量生成器重复其序列的周期，即随机性的丧失。

下面几节将讨论不同的算法随机（伪随机）数生成器和相关的随机性检验方法。

3.3 伪随机数生成器

一个好的伪随机数生成器必须产生一个在（0，1）上均匀分布的随机数序列。此外，它应该有一个很长的周期，并且必须通过一系列的随机性检验。本节将介绍常用的生成器，包括：①同余生成器；②多重递归生成器。

3.3.1 同余生成器

同余生成器是德里克·亨利·莱默（D. H. Lehmer）提出的一种整数生成器。它采用：

$$x_{k+1}=(ax_k+b),\quad \bmod M,\quad b<M \tag{3.1}$$

其中，x_0（种子）、a、b、M 是给定的整数；M 是计算机表示的最大整数，例如，在 32 位的二进制机器上，最大的无符号整数是 $2^{32}-1$，最大的有符号整数是 $2^{31}-1$。模函数决定了（$\alpha=ax_k+b$）除以 M 的余数，例如，35 对 16 取模等于 3。（注意，同余一词源于 $\frac{\alpha}{M}$ 与 $\frac{x_{k+1}}{M}$ 具有相同余数，即 x_{k+1} 在模 M 下与 α 同余）。值得注意的是，式（3.1）被称为线性同余生成器，如果 $b=0$，则称为乘法同余生成器。

这里讨论的方法将给出一个在［0，$M-1$］上的随机整数 x，为了将生成的随机整数转换为［0，1）上的随机数，使用关系 $\eta=\frac{x}{M-1}$。

用一个简单的例子来讨论同余生成器的细节和相关问题是有指导意义的。考虑由式（3.2）给出的线性同余生成器[20]：

$$x_{k+1}=(5x_k+1),\quad \bmod 16 \tag{3.2}$$

假设种子为 1，则随机数序列可以表示为一个随机数循环（顺时针），如图 3.1 所示。

图 3.1 中的循环展示了一个生成器的预期行为。

（1）这个序列的周期是 16，即等效于模量 $M=2^4$，最大值为 2^4-1。

（2）选择任何其他数（小于 $M=16$）作为种子将导致上述序列的周期性移位。

检验乘子（a）对上述生成器周期的影响是有指导意义的。表 3.2 给出了选择不同乘子（3～15）时生成器的周期。

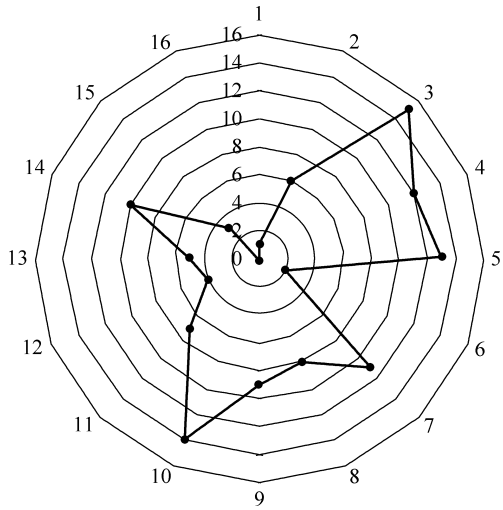

图 3.1　由式(3.2)给出的生成器随机循环示意

注意：等高线给出了随机数值，径向索引指的是序列中随机数的顺序

表 3.2　乘子(a)对伪随机数生成器周期的影响(式(3.2))

参数	值												
乘子(a)	3	4	5	6	7	8	9	10	11	12	13	14	15
周期	8	2	16	4	4	2	16	4	8	2	16	4	2

表 3.2 表明，如果乘子(a)等于 5、9 或 13，则得到完整周期为 16，否则得到部分周期为 2、4 或 8。更具体地说，图 3.2 给出了乘子 9(实线)和乘子 13(虚线)的随机数序列。

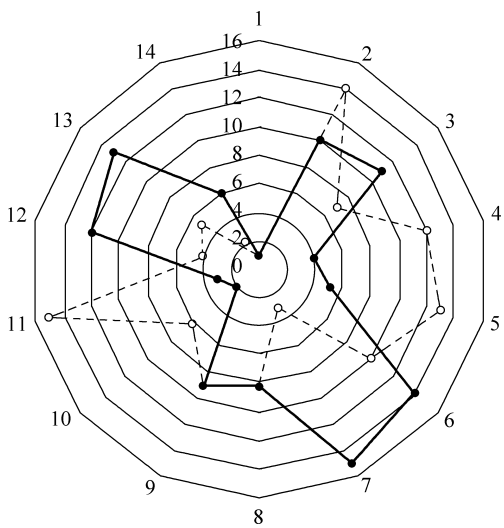

图 3.2　乘子 9(实线)和乘子 13(虚线)的随机数序列

图 3.3 给出了导致得到部分周期的乘子 8(实线)和乘子 12(虚线)的随机数序列。

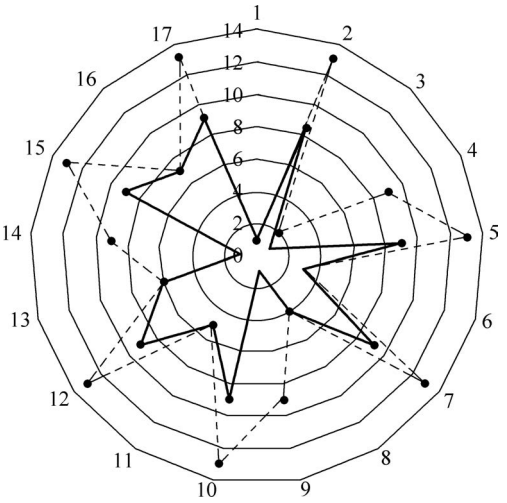

图 3.3 乘子 8(实线)和乘子 12(虚线)的随机数序列

正如预期的那样，在具有完整周期的情况下，随机数形成具有不同值的形状，而所有部分周期的情况都呈现重复的形状。接下来，检验常数(b)的影响是有指导意义的，对于乘子为 5 的情况，本节分析了常数(b)对生成器周期的影响。表 3.3 比较了不同 b 值(2～8)时的生成器的周期。

表 3.3 表明，奇数 b 能得到一个完整的周期，而偶数 b 只能得到部分周期。

表 3.3 常数(b)对 PRNG 周期的影响(见式(3.2))

参数	值						
常数(b)	2	3	4	5	6	7	8
周期	8	16	2	16	8	16	4

考虑到上述观察结果，对于任意 n，生成器 $x_{k+1}=(5x_k+1)\bmod 2^n$ 应该实现一个完整的周期。图 3.4 和图 3.5 分别给出了 $n=5$ 和 $n=6$ 时的随机数序列。

由图 3.4 和图 3.5 可知，正如期望的那样，$M=32$ 和 $M=64$ 分别对应生成 32 个和 64 个随机数，即全周期。

为了实现线性同余生成器的全周期，可以使用 Hull 和 Dobell[53] 定理的推论，该定理设置了不同参数的性质，包括乘子、常数和模。表 3.4 给出了这些属性。

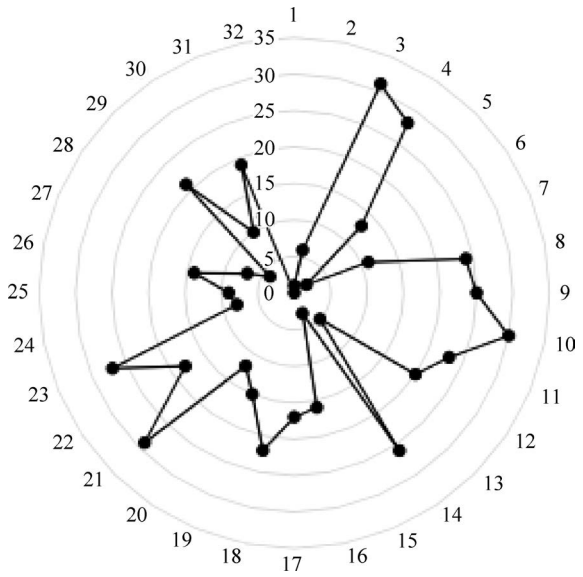

图 3.4　$M=2^5$ 时的式(3.2)PRNG 的随机数序列

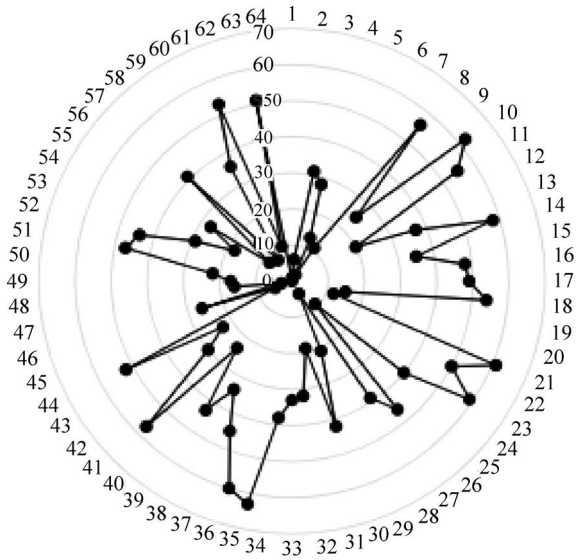

图 3.5　$M=2^6$ 时的式(3.2)PRNG 的随机数序列

表 3.4　实现全周期的线性同余生成器参数的性质

参　　数	值	注　　释
乘子(a)	$4N+1$	$N>0$
常数(b)	奇数	
模(M)	2^k	$k>1$

现在，设置常数参数(b)为 0 来考虑一个乘法同余生成器，即

$$x_{k+1}=(ax_k)\bmod 16 \tag{3.3}$$

假设种子为 1，表 3.5 给出了不同乘子(2～15)的不同随机数序列的周期。

表 3.5　乘子(a)对 PRNG 周期的影响(见式(3.3))

参数	值													
乘子(a)	2	3	4	5	6	7	8	9	10	11	12	13	14	15
周期	—	4	—	4	—	2	—	2	—	4	—	4	—	2

图 3.6 显示了乘法生成器(见式(3.3))对乘子 3 和乘子 11 生成的随机数序列。这演示了生成器的部分周期，因为每种情况都会导致多边形通过重复种子而终止。

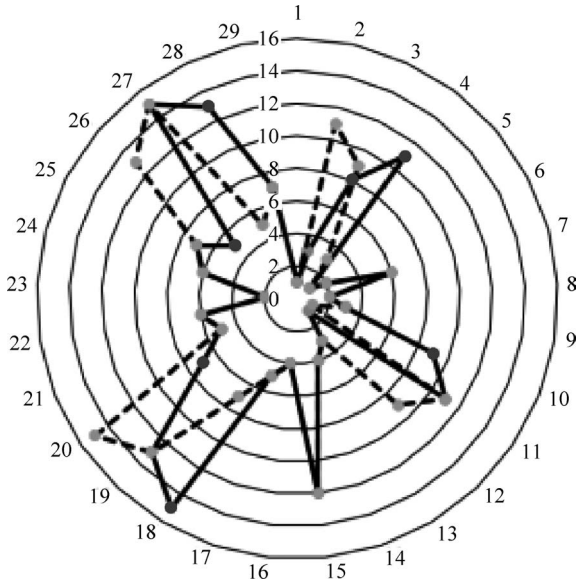

图 3.6　乘子 3(实线)和乘子 11(虚线)的随机数序列，式(3.3)PRNG

同理可以得到乘子分别为 5 和 13 的结果。这些结果表明，若 $N\geqslant 0$，则当 $a=8N+3$ 或 $a=8N+5$ 时，乘法同余生成器的周期为 2^{k-2}，其中 k 对应于模的指数，如 $16=2^4$。现在，考虑一个具有奇数乘数(16 339)和偶数模 2^k 的生成器，使用表 3.6 中给出的算法确定在任意幂 N 下的周期。

表 3.6　线性同余随机数生成器的一个算法

算　　法	注　　释
readt，*，multi，const，imod	读取种子、乘子、常量和模数
$ixx(1)=x_0$	初始化随机数数组

续表

算　　法	注　　释
do $i=2,i$mod	循环生成随机数，
iab=multi * $ixx(i-1)$+const	计算 $ax_{i-1}+b$
$ixx(i)$=mod(iab,$i$mod)	用“mod”函数得到新的随机数
if ($ixx(i)$.lt.0)then	检查随机数是否为正
$ixx(i)=ixr(i)+i$mod	
end if	
if ($ixx(i)$.eq.xo)then	确定周期
$i$period=$i-1$	
go to 10	
else	
$i$period = i	
end if	
end do	
10 continue	

注意，mod 函数可以用式(3.4)代替：

$$ixx(i)=iab-\mathrm{INT}(iab/\mathrm{mod})i\mathrm{mod} \tag{3.4}$$

注意，该算法的实现依赖特定系统的整数运算知识。(有关这方面的讨论，见附录 A。)

如果考虑 2^k 的模(mod)的不同幂 k，可以证明，每个序列的周期是模的 25%。

为了实现乘法同余生成器的大周期(大于模量的 25%)，已经证明[20]，如果考虑以下几点，可以实现 $M-1$ 的周期：

(1) M 是质数。

(2) 乘子 a 是 M 的基本因子。

Park 和 Miller[79]证明了模数为 $2^{31}-1$，乘子为 16 807 时，周期为 $M-1$。关于最佳随机数生成器更进一步的信息，读者可以参考以下文献：Bratley、Fox 和 Schrage[15]、Park 和 Miller[79]、Zeeb 和 Burns[121]、L'Ecuyer[63]、Gentle[39] 及 Marsaglia[69]。

总之，具有合理周期的同余生成器对于大多数物理模拟可以产生令人满意的结果，因为物理系统通过将相同的随机数应用于不同的物理过程来引入随机性。然而，对于解决数学问题，这种说法并不总是正确的。

3.3.2　多重递归生成器

一组伪随机数生成器称为多重递归生成器[39]，可以表示为

$$x_{k+1}=(a_0x_k+a_1x_{k-1}+\cdots+a_jx_{k-j}+b),\quad \mathrm{mod}\ M \tag{3.5}$$

其通过选择 $j+1$ 个随机数（可能来自更简单的生成器）来初始化生成器。生成器的长度和随机性（或统计属性）取决于 a_j、b 和 M 的值。这类生成器的一个特例是斐波那契生成器。

斐波那契生成器是一个浮点数生成器。它的特点是通过前两个数字的组合（差、和或积）计算出一个新数字。如式(3.6)所示[55]：

$$x_k = x_{k-17} - x_{k-5} \tag{3.6}$$

是一个 17 和 5 滞后的斐波那契生成器，所生成的序列取决于初始 x_k 的选择，如 17。由于上面的斐波那契公式 x_k+1 依赖前面的 17 个值，所以它的周期（p）可以很大，即 $p=(2^{17}-1)2^n$，其中 n 是 x_i 的小数部分的位数。例如，对于 32 位的浮点运算，$n=24$，因此，p 约为 2^{41} 或 10^{12}。由于预期周期大，因此对于一些大型问题，斐波那契生成器是一个很好的选择。超级计算机上更大的问题使用了滞后 97 和 33 的斐波那契生成器。

要启动斐波那契生成器，必须生成初始随机数，如 17 个数字。一种方法是以二进制形式表示每个初始随机数：

$$r=\frac{r_1}{2}+\frac{r_2}{2^2}+\cdots+\frac{r_m}{2^m},\quad \text{对于其中 } m \leqslant n\text{（尾数）} \tag{3.7}$$

其中，r_i 是二进制数，即 0 或 1。需要一个更简单的同余生成器来设置每个位 r_i 为 0 或 1。例如，我们可以根据整数同余生成器的输出是大于 0 还是小于 0，将 r_i 设置为 0 或 1。因此，斐波那契公式的质量确实取决于初始数字的质量，或者更简单的整数生成器的质量。表 3.7 给出了推荐的斐波那契生成器列表。

表 3.7　推荐的斐波那契生成器列表

生　成　器	期 望 周 期
$x_k=x_{k-17}-x_{k-5}$	$(2^{17}-1)\times 2^{24}=2.2\times 10^{12}$
$x_k=x_{k-31}-x_{k-13}$	$(2^{31}-1)\times 2^{24}=3.6\times 10^{16}$
$x_k=x_{k-97}-x_{k-33}$	$(2^{97}-1)\times 2^{24}=2.7\times 10^{36}$

请注意，当使用 32 位整数时，同余生成器可以拥有的最大周期是 2^{32}，或 4.3×10^9，这比表 3.7 中的任何一个建议都要小得多$\left(\text{至少是其}\frac{1}{500}\right)$。

3.4　随机性检验

如果伪随机数生成器通过了一系列随机性检验，那么它就是一个可以被接受的生成器。许多随机性检验[59,68]通过考察各种参数来检验随机数序列的独立性和一致性。本节的其余部分将讨论一组检验，这些检验根据随机数的位数和整数来检查随机数[77]。

3.4.1　χ^2-检验

χ^2-检验测量样本与假定概率分布(假设)之间的偏差。χ^2-检验的公式由式(3.8)给出：

$$\chi^2=\sum_{i=1}^{n}\frac{(N_i-Np_i)^2}{Np_i} \tag{3.8}$$

其中，$\{p_1,p_2,\cdots,p_n\}$是与 N 个事件相关的假设概率集合，这些事件属于 N 个类别，观测频率为 $N_1,N_2,\cdots,N_n$。请注意，与假设分布相比，该检验一次检查整个抽样分布，并且在这个意义上比检查样本均值、样本方差等更普遍。对于较大的 N 值，随机变量 χ^2 近似服从自由度为 $n-1$ 的 χ^2-分布密度函数。

3.4.1.1　χ^2-分布

如果随机变量 $\omega=\chi^2$ 具有 χ^2-分布[19]，且遵循如下概率密度函数：

$$f_m(w)\mathrm{d}w=w^{\frac{m}{2}-1}2^{\frac{m}{2}}\Gamma\left(\frac{m}{2}\right)\mathrm{e}^{-\frac{w}{2}}\mathrm{d}w \tag{3.9}$$

其中，m 为正整数，表示自由度；Γ 为伽马函数，$\omega>0$。一般来说，我们感兴趣的是找到 $\omega=\chi^2$ 小于给定值 χ_0^2 的概率，但可用的 χ^2 分布表通常给出 $\chi^2\geqslant\chi_0^2$ 的概率。因此，有必要使用概率的补集，即

$$P(\chi^2\leqslant\chi_0^2)=1-P(\chi^2\geqslant\chi_0^2) \tag{3.10}$$

其中，

$$P(\chi^2\geqslant\chi_0^2)=\int_{\chi_0^2}^{\infty}\mathrm{d}wf_m(w) \tag{3.11}$$

值得注意的是，随着 m 的增大，χ^2-分布趋于正态分布，分布的均值和方差分别等于 m 和 $2m$。

3.4.1.2　χ^2-检验的流程

根据式(3.8)得到卡方(χ^2)值，然后将这些值与 χ^2 分布表中给出的确定值进行比较，如表 3.8 所示。

表 3.8　一个 χ^2 分布表的例子($P(\chi^2\geqslant\chi_0^2)$)

自由度	0.990	0.950	0.050	0.010	0.001
1	0.000	0.004	3.840	6.640	10.830
2	0.020	0.103	5.990	9.210	13.820
3	0.115	0.352	7.820	11.350	16.270
4	0.297	0.711	9.490	13.280	18.470
5	0.554	1.145	11.070	15.090	20.520
6	0.872	1.635	12.590	16.810	22.460
7	1.239	2.167	14.070	18.480	24.320