Foundations and Methods of Stochastic Simulation

随机仿真原理与方法

【美】巴里·L·尼尔森 著
曹军海 杜海东 申莹 译

国防工业出版社
National Defense Industry Press
Springer

随机仿真原理与方法

Foundations and Methods of Stochastic Simulation

[美] 巴里·L·尼尔森 著
曹军海 杜海东 申 莹 译

国防工业出版社
·北京·

著作权合同登记　图字：军-2018-025 号

图书在版编目（CIP）数据

随机仿真原理与方法/（美）巴里·L·尼尔森（Barry L. Nelson）著；曹军海，杜海东，申莹译. —北京：国防工业出版社，2019.6

书名原文：Foundations and Methods of Stochastic Simulation

ISBN 978-7-118-11700-4

Ⅰ. ①随… Ⅱ. ①巴… ②曹… ③杜… ④申… Ⅲ. ①系统仿真 Ⅳ. ①TP391.9

中国版本图书馆 CIP 数据核字（2019）第 056022 号

Translation from the English language edition:

Foundations and Methods of Stochastic Simulation: A First Course

by Barry Nelson

※

国防工业出版社出版发行

（北京市海淀区紫竹院南路 23 号　邮政编码 100048）

三河市腾飞印务有限公司印刷

新华书店经售

*

开本 710×1000　1/16　印张 16¼　字数 303 千字

2019 年 6 月第 1 版第 1 次印刷　印数 1—1500 册　定价 109.00 元

（本书如有印装错误，我社负责调换）

国防书店：（010）88540777　　发行邮购：（010）88540776

发行传真：（010）88540755　　发行业务：（010）88540717

译者序

随着现代工业的进步，科学研究的深入，以及信息技术、计算机技术的蓬勃发展，计算机仿真技术已经成为分析、研究和设计各类系统，特别是复杂大系统的一种有效而灵活的方法论工具。作为仿真技术原理与方法的入门级教程，本书是计算机仿真领域的国际顶尖专家——美国西北大学工业工程与管理科学系巴里·L·尼尔森教授的一部经典著作，它凝聚了尼尔森教授多年从事计算机仿真技术及其应用领域教学和研究工作的成果，在计算机仿真领域具有非常重要的影响，非常适合计算机仿真技术的入门者学习。

本书的主要特点在于其思想性、基础性和实践性。本书首先从“为什么要进行仿真”这个基本问题出发，通过一个简单的示例，为读者揭示系统仿真的本质、应用的需求和基本思想，并解答了仿真技术应用的根本性问题；其次分别从仿真的观点、仿真输入、仿真输出、实验的设计等多个方面为读者进一步揭示了仿真技术实现的关键性问题，具有很好的理论基础性。尼尔森教授这部著作的另一大闪光点就是其实践性。全书从第2章开始就引入了一个具体的示例，之后又通过一个附有全部源代码的、基于VBA的仿真程序的详细设计，为读者从根本上解释了仿真开发的基本原理，直接缩小了读者与仿真应用开发的距离，使学习者大大克服了对仿真技术开发的畏惧心理。

应该说，得益于长年在仿真技术领域的研究与教学经验，尼尔森教授在这部著作中充分考虑了每一个仿真技术初学者所面临的种种困惑，力求以通俗易懂且不失理论性和学术性的方式，为读者揭示了仿真技术的本源，这对于学习仿真基础知识的初学者来说是非常合适的。

本书在版权引进和出版过程中，得到了“十二五”国防预先研究项目（项目号：51319050302）的资助。全书共分为9章，其中第1～4章由曹军海翻译，第5～7章由杜海东翻译，第8、9章及附录由申莹翻译，全书由李羚玮、郑竣兮、梁精睿审校。曹军海负责全书的翻译策划、统稿等工作。

计算机仿真技术日新月异，书中涉及大量数学、建模与仿真、计算机软件领域的专有概念和术语，且有一些概念和术语目前尚无公认的中文译法。若有不符合学习者习惯或者不准确的术语出现，敬请批评和指正！

译　者

2019年1月于北京

前　言

像经常出现的情况那样，本书是从一门课程逐渐发展而来的。有意思的是，这门课程的发展出自这样一个故事：当时我正在指导我们的一位研究生 Viji Krish－namurthy，她的研究涉及为在一个修理和维护环境下使用灵活工人设计规则（Iravani 和 Krishnamurthy，2007），采用马尔科夫链分析已经得到了一些用于简单系统的优化的策略，现在她想测试这些策略的鲁棒性以便用于更现实的问题，这就是仿真的由来。Viji 已经学习了具有代表性的初级仿真建模课程，该课程使用了一个商业仿真软件产品，以及一门只关注于设计和分析但不包括模型构建的高级课程。她（包括我）花费了大量的时间尝试用一个商业软件产品来仿真她想测试的复杂的工人分配规则。商业仿真软件环境通过引入那种用户期望的系统特性，使得建模工作非常容易。不幸的是建模的典型特性容易表达，但都很难代表不同的事物，而科学研究总是针对不同的事物的。

最后，在沮丧中，我手写了 3 页伪代码来模拟 Viji 想要的东西，并将笔记交给了她，然后告诉她用 C 语言（很幸运她会）完成代码。第二天她回来时激动地对我说："现在我可以测试任何我想要的东西了，而且运行只需要几秒钟！"自从这次经历之后，IEMS 435 "随机仿真引论"这门课程诞生了，它是博士研究生的必修课。然而，IEMS 435 不仅仅是关于建模和编程的课程，Viji 还需要在她的模型上运行精心设计的实验，这些实验可以提供令人信服的证据支持或反对她通过分析得出的规则。所以，实验设计和分析也是 IEMS 435 课程的一部分，写作本书也正是想对此有所帮助。

本书的目标如下：

（1）为那些从未学习过离散事件、随机仿真课程的学生奠定基础，使他们可以用一种低级编程语言建立仿真。我确信，如果他们能够做到这一点，就可以在需要的时候很容易地掌握高级仿真建模环境（也可能为教职人员讲授一门课程）提供支撑。

（2）为学生打好基础，使他们能够将仿真应用到非仿真研究项目中。这是为什么强调实实在在地编写仿真程序（这可以提供最大的灵活性、控制和理解），开展实验设计和分析。

（3）为学生学习仿真技术的高级课程奠定基础，包括在他们的导师指导下的自学。一般仿真方面的初级课程强调建模和商业软件，打破了建模与分析之间的重要联系，对面向研究的高级课程来说不能起到很好的基础作用，因为高级课程可能将仿真几乎整个作为一个数学对象来对待。

（4）提供一个牢固的仿真方面的数学/统计学基本知识基础以及一些（不是全部）解决实际问题的工具。

本书的观念非常类似于 Law（2007）的著作，同时包含了仿真的建模和分析两个方面，但不同之处在于，本书试图不达到全面或者纵览这一领域。它的目标是简明、准确以及完整，为教师留下大量空间，以便扩充他们感兴趣或者对他们非常重要的领域。我希望教师能要求学生在开始阅读其他参考文献或期刊论文之前，阅读全书得到一个完整的、条理清楚的图像。最后，我提供了相关文献的线索。总之，尽管不够综合性，但本书是完整的。所以，它对不能上课需要靠自己学习仿真基础知识的学生或者研究人员来说是非常合适的。Asmussen 和 Glynn（2007）的著作与本书的一些目标是一致的，它是一本非常优秀的介绍先进仿真分析的著作，涵盖了比我更多的论题，但它在解决建模或者编程问题上没有达到与本书相同的程度。

对应于本书中的两章，有关仿真建模与编程的材料使用了 Excel 中的 VBA（Visual Basic for Applications）编程技术。这一选择来自于这样的动机，那就是越来越少的到我这里学习的研究生具备编程经验。VBA/Excel 非常容易获取，也很易于快速掌握，对学生日后学习 Java、C 语言或者其他编程语言是一个很好的准备。书中的这部分内容对于已经掌握如何仿真编程的学生来说，可以直接跳过而不必在意这些提示。本书第 4 章的 Java 和 Matlab 版本代码、书中描述的所有软件，以及练习所需的所有数据集，都可以从本书的网站下载：users. iems. northwestern. edu/nelsonb/IEMS435/。

本书适用于高年级本科生和研究生。在概率论与统计方面的一门必修课是很有必要的，仅仅学习了统计学是不够的。虽然本书使用了 Tractable 随机过程模型（如马尔科夫队列）作为示例，但并不要求读者在这些专题方面具备任何背景（实际上，学习完本书后学生们可能会发现，“随机过程”这门课更有指导性和实际意义）。如果学生们没有编写计算机算法程序的经验，例如，用 Matlab 或者某些其他编程语言，那么教师将不得不在本书的教学中补充更多的编程实践。但是对于一个过来人甚至优秀的程序员来说，就不必如此了。

第 1 章通过一个简单的可靠性问题对本书进行了精要的总结（除了编程）。第 2 章使用 VBA，通过一个快速的仿真编程起步指导来弥补编程经验方面的不足。第 3 章引入了大量的 Tractable 例子，说明在通过仿真方法分析随机系统时会遇到的关键问题，这些例子贯穿全书，所以这一章是必读的。VBASim 是开发出来用于支撑本书的一个 VBA 子程序和类模块的集合，在第 4 章中对它进行了说明，如果使用其他编程语言或者程序包，可以跳过该章。第 5 章强化了对随机过程进行仿真和数学性/数值型分析之间的联系，该章还具有双重任务，就是为后续有关设计和分析的章节做准备，以及对学生学习更高级的仿真方法课程做好准备。第 6 ~ 8 章分别介绍输入建模、输出分析和实验

设计，都极大地独立于具体用于构建仿真的编程方法。本书以第9章作为结尾，针对使用仿真方法来解决系统分析问题，对在研究中使用仿真方法给出了指南。

缺少什么吗？本书没有触及计算环境问题，如果有计算环境，你可能想做一些完全不同的事，如你具有一个含500个可利用的CPU的云计算集群，我预测到本书写第2版的时候，将更容易促成这样一个离散事件随机仿真的环境，到时会加入一些一般性的指导和建议。虽然本书中有大量关于仿真优化的材料，但缺少具体算法，这反映了一个事实，即：针对所有类型问题的基本方法，当前还没有一致的观点，这也终将改变。最后，除了讨论其含义，没有对等地讨论仿真模型的校验问题。

很多学生和我的同事对本书中使用的编程方法的设计做出了贡献。IEMS 435这门课最初使用Law和Kelton（2000）的著作作为教材，Dingxi Qiu花费了一个暑假将该书中的C代码转换为VBA代码。Christine Nguyen帮助开发了VBASim——本书中描述的仿真支撑库。Feng Yang与我一起开展了一个研究项目，我们使用VBA进行仿真分析。Lu Yu帮助编写了解题手册。Luis de la Torre和Weitao Duan分别完成了Java和Matlab版本的VBASim。

我已经得到了大量的反馈。IEMS 435课程的学生们带着欢乐接受了本书的未完成版本，以及印刷错误和错别字问题。Larry Leemis提供了一个全标注的早期草稿，Jason Merrick用它来教学。我的很多朋友阅读并标注了接近完成的书稿中的章节，包括Christos Alexopoulos、Bahar Biller、John Carson、Xi Chen、Ira Gerhardt、Jeff Hong、Sheldon Jacobson、Seong-Hee Kim、Jack Kleijnen、Jeremy Staum、Laurel Travis、Feng Yang、Wei Xie、Jie Xu和Enlu Zhou。Michael Fu在如何编写梯度估计方面给我提供了思路，同样，Bruce Schmeiser在误差估计方面也做了贡献。Seyed Iravani、Chuck Reilly和Ward Whitt确保我在第9章中无误地使用他们论文中的信息。除上述人员之外，还有一些人员为本书创作提供了思路，包括Sigriin Andradottir、Russell Cheng、Dave Goldsman、Shane Henderson、David Kelton、Pierre L'Ecuyer、Lee Schruben和Jim Wilson。感谢你们。

最后很重要的一点，本书的编写工作部分地得到了国家自然科学基金（授予号：CMMI-1068473）的支持。据我所知，NSF比所有联邦政府机构以更少的投入完成更多的事情。

美国伊利诺伊州埃文斯顿
巴里·L·尼尔森

目　录

第 1 章　为什么要仿真 …… 1

1.1　示例 …… 1

1.2　形式化分析 …… 3

1.3　问题与扩展 …… 5

练习 …… 5

第 2 章　仿真编程：快速入门 …… 7

2.1　一个 TTF 仿真程序 …… 7

2.2　一些重要的仿真概念 …… 10

2.2.1　随机变量生成 …… 11

2.2.2　随机数生成 …… 13

2.2.3　仿真重复 …… 14

2.3　VBA 入门知识 …… 16

2.3.1　VBA：基础 …… 16

2.3.2　函数、子程序和范围 …… 19

2.3.3　与 Excel 交互 …… 21

2.3.4　类模块 …… 22

练习 …… 24

第 3 章　示例 …… 26

3.1　$M(t)/M/\infty$队列 …… 26

3.2　$M/G/1$ 队列 …… 28

3.3　AR（1）替代模型 …… 30

3.4　一个随机活动网络 …… 32

3.5　亚洲式期权 …… 33

练习 …… 35

第 4 章　用 VBASim 进行仿真程序设计 …… 37

4.1　VBASim 简介 …… 37

4.2　$M(t)/M/\infty$ 队列仿真 …… 38

4.2.1　问题及扩展 …… 43

4.3　$M/G/1$ 队列仿真 …… 44

4.3.1 *M/G/1* 队列中的 Lindley 仿真 …… 45
4.3.2 基于事件的 *M/G/1* 队列仿真 …… 47
4.3.3 问题和扩展 …… 52
4.4 随机活动网络的模拟 …… 53
4.4.1 SAN 的最大路径仿真 …… 53
4.4.2 SAN 离散事件仿真 …… 54
4.4.3 问题和扩展 …… 58
4.5 亚式期权仿真 …… 59
4.6 案例分析：服务中心仿真 …… 60
4.6.1 问题和扩展 …… 67
练习 …… 68
第 5 章 两种仿真观点 …… 72
5.1 仿真建模和分析框架 …… 72
5.2 随机过程仿真 …… 75
5.2.1 模拟渐近行为 …… 76
5.2.2 基于仿真重复的渐近解 …… 79
5.2.3 仿真重复中的渐近解 …… 81
5.2.4 超过样本均值 …… 84
附录：迭代过程和稳定状态 …… 85
练习 …… 87
第 6 章 仿真输入 …… 89
6.1 输入建模概述 …… 89
6.1.1 输入建模故事 …… 89
6.1.2 输入过程特性 …… 92
6.2 单变量输入模型 …… 93
6.2.1 单变量分布推理 …… 94
6.2.2 估计和检验 …… 98
6.2.3 单变量分布的属性匹配 …… 100
6.2.4 经验分布 …… 104
6.2.5 直接使用输入数据 …… 106
6.2.6 无数据输入模型 …… 107
6.3 多变量输入过程 …… 108
6.3.1 非稳态到达过程 …… 109
6.3.2 随机矢量 …… 118
6.4 随机变量生成 …… 120

6.4.1 拒绝法 …… 121
6.4.2 特殊属性 …… 123
6.5 伪随机数生成 …… 124
6.5.1 多重递归生成器 …… 126
6.5.2 组合生成器 …… 126
6.5.3 伪随机数生成器的合理使用 …… 128
附录1 GLD 属性 …… 129
附录2 NORTA 法的实现 …… 130
练习 …… 134
第7章 仿真输出 …… 140
7.1 性能指标、风险和误差 …… 140
7.1.1 点估计量和误差测量 …… 142
7.1.2 风险和误差指标 …… 144
7.2 输入不确定性和输出分析 …… 146
7.2.1 输入不确定性：是什么 …… 146
7.2.2 输入不确定性：怎么办 …… 147
附录 $M/M/\infty$队列的输入不确定性 …… 150
练习 …… 151
第8章 实验设计和分析 …… 156
8.1 通过仿真重复控制误差 …… 158
8.2 稳态仿真设计和分析 …… 161
8.2.1 固定精度问题 …… 163
8.2.2 固定样本量问题 …… 166
8.2.3 批次统计 …… 171
8.2.4 稳态仿真：附言 …… 172
8.3 仿真优化 …… 173
8.3.1 收敛 …… 175
8.3.2 正确选择 …… 176
8.4 仿真优化实验设计 …… 178
8.4.1 随机数赋值 …… 178
8.4.2 为正确选择进行的设计和分析 …… 183
8.4.3 自适应随机搜索 …… 190
8.4.4 改进方向中的搜索 …… 192
8.4.5 最优化样本均值问题 …… 198
8.4.6 随机约束条件 …… 200

8.5 利用控制变量法进行方差缩减 …… 204
8.5.1 *M/G/1* 队列控制变量 …… 207
8.5.2 随机活动网络控制变量 …… 208
8.5.3 亚洲式期权控制变量 …… 208
附录 控制变量估计值的性质 …… 209
练习 …… 210
第9章 研究中的仿真 …… 216
9.1 随机试验问题 …… 217
9.2 鲁棒性研究 …… 218
9.3 关联实验影响因子 …… 220
9.4 控制研究仿真中的误差 …… 221
附录A VBASim …… 225
A.1 核心子程序 …… 225
A.2 随机数生成 …… 228
A.3 随机变量生成 …… 232
A.4 类模块 …… 236
A.4.1 CTStat …… 236
A.4.2 DTStat …… 237
A.4.3 Entity …… 238
A.4.4 EventCalendar …… 238
A.4.5 EventNotice …… 240
A.4.6 FIFOQueue …… 240
A.4.7 Resource …… 241
参考文献 …… 244

第 1 章　为什么要仿真

随机仿真是一种分析系统性能的方法，这些系统的行为取决于随机过程和那些可以完全用概率模型表征的过程之间的交互。随机仿真是一种与随机模型的数学性和数值性分析相并列的方法（如 Nelson 1995），它常用于期望的性能度量难以用数学方法解决或者没有误差有界的数值估计的情况。计算机使得随机仿真切实可行，但不使用任何计算机就可以对该方法进行描述，这正是在这里我们要做的事情。

1.1　示　　例

下面以一个简单、传统的但能够说明本书中要解决的关键问题的例子开始。我们将使用这样的例子贯穿本书，它们是经过精心选择来说明现实问题复杂性的例子，没有为了分析或者仿真而复杂化。在教学和研究活动中，能够说明复杂行为，但没有为了解释或分析而复杂化的示例，这是非常重要的：它们建立人们的直觉认识，并提供了一个全面考虑各种新观点而不被大量细节所迷惑的途径。实际的模型可能确实被复杂化了，具有大量的输入和输出，以及需要成千上万行的计算机代码实现，所以在第 2 ~ 4 章中也要解决建模与仿真编程的问题。

例 1.1（系统故障前时间 TTF）。在这里考虑的 TTF 系统由两个部件组成：一个作为工作部件；另一个作为冷储备部件。当（当前）工作部件发生故障的时候，储备部件就变为工作部件，而故障的部件立即开始修理。故障的部件在修理完成后，会变为储备部件。在同一时间只有一个部件可以被修理，所以如果两个部件都发生故障则系统就会故障，直到至少一个部件开始工作为止。每个部件的工作时间（故障前时间）等概率地可能为 1 ~ 6 天，而修理工作确切地要花费 2.5 天。另外，一个修好的部件等同于一个新的部件。

如果我们对这样一个系统感兴趣，那么可能想知道该系统的故障特征。例如系统首次故障前的平均时间，或者系统长期运行的系统可用度。请注意，虽然各部件的故障特征是完全指定的（它们的工作时间等概率地取 1 ~ 6 天），但系统整体的故障特征——那取决于故障与修理活动之间的相互转换——不是

立即就能清楚的。实际上，像这样简单的系统，从数学上获取这些性能特征是非常困难的，而进行仿真却是很容易的（正如在下面所介绍的那样），这就是为什么要仿真。

如果一个系统的行为可以完全通过一个概率模型描述，则将它的那些特性称为输入，如部件的故障前时间等。而那些由此引申出的定量特性，如系统首次故障前时间或者一定时间范围内的系统可用度等，称为输出。

随机仿真方法由生成实现输入、执行系统逻辑产生输出和根据输出评估系统性能特征三部分工作组成。

第6章解决表达和生成输入的问题；第7章说明输出分析的问题；第8章是关于实验设计的内容。

将TTF系统在任何时间点的状态用可工作部件的数量表达，其值可以是2、1或者0。导致系统状态发生变化的事件包括部件故障和修理完成。假设有一个随机性发生源用于生成输入（在本例中是部件的故障前时间），当前状态、一个未来事件列表（称为事件日历），以及一个时钟就足够用来模拟出系统行为的抽样路径。这里将运用一个物理机制，一个合理的六面体骰子作为随机性发生源。

表1.1显示了如果第一组4次掷骰子得到的数是5、3、6、1，在首次系统故障时停止（当状态变为0时，意味着没有可工作的部件）的情况下的抽样路径；每次掷骰子的结果放在一个盒子里，这样来代表六面体骰子的一次滚动。该路径是通过执行图1.1中所示的通用仿真算法生成的，需要注意的是，当一个故障发生时，状态值减1，而一个修复完成时状态值加1。

表1.1　TTF系统仿真直至首次系统故障

时钟/天	系统状态	时间日历		备　注
		下次故障	下次维修	
0	2	$0+\boxed{5}=5$	∞	初始化为完全可工作
5	1	$5+\boxed{3}=8$	$5+2.5=7.5$	一个工作，另一个维修中
7.5	2	8	∞	下次故障停留在日历中
8	1	$8+\boxed{6}=14$	$8+2.5=10.5$	一个工作，另一个维修中
10.5	2	14	∞	再次完全可工作
14	1	$14+\boxed{1}=15$	$14+2.5=16.5$	马上将激活故障
15	0	∞	16.5	系统故障

表1.1中说明的方法一开始时经常让人觉得不自然。例如，在很容易计算出每个部件将在何时发生故障的情况下，为什么只跟踪记录下一次故障？以及为什么不在时间为0：00时将下一次修理也放入日历中？因为可以很容易地看

到它会在7.5天时发生。原因如下：

初始化：设置时钟和系统状态的初始值，将至少一个故障事件写入事件日历。
时间推进：更新时钟至下一个待发生事件的时间点（并且将该事件从日历中移除）。
更新：将状态更新到与当前事件相适应的值，并安排任何新的未来事件在当前事件时钟加一个时间增量后发生。
结束：如果一个终止条件已经满足，则停止仿真；否则转至时间推进。

图 1.1　通用仿真算法

缓慢的仿真器规则：只有当不这样做会导致事件不按照时间顺序发生时，才对未来事件进行预先计划。

例如，虽然可以计算出何时（两个部件同时故障，但不强行将备用部件的故障计划到工作部件）发生故障时。同样，没必要将一次维修安排到正好某个部件发生故障时，即使可以那样做。另外，在时钟为8天时，必须同时安排好下次故障和下次维修这两个事件，因为从该时刻起的系统演化取决于它们中哪一个首先发生。这一规则被证明在编写复杂的仿真程序时是非常重要的，复杂的仿真程序中可以包含很多种在日历上同步发生的事件。特别是，这一规则可以避免不得不取消那些因为另一个事件首先发生而变得无关紧要的事件。

这里有一些TTF系统示例的特性，它们对本书中的随机仿真程序（几乎所有）都是普遍适用的：

（1）仿真时间（仿真时钟）是从一个事件时间跳到另一个事件时间的，而不是连续更新。因为这个原因，称这种仿真为离散事件仿真。

（2）时钟、系统的当前状态、未来事件列表以及事件逻辑就是我们将仿真推进到下一次状态变化所需的全部条件。

（3）当达到特定的系统状态，或者在一个固定的仿真时间点，或者当一个特定事件发生时，仿真结束。

1.2　形式化分析

表1.1显示了TTF系统的抽样路径。用 Y 表示首次系统故障时间，用 $S(t)$ 表示到时刻 t 时系统可以工作的部件的数量。一个抽样路径提供了对 Y 的一次观测数据，但是 $S(t)$ 是一个函数，它的值在整个仿真过程中是不断演化的。如果对可工作部件数量在某个仿真时间范围 T 上的平均值感兴趣，把它表示为 $\bar{S}(T)$，那么它是一个时间平均值，因为 $S(t)$ 在所有时间点上都有一个值。因此，有

$$\bar{S}(T) = \frac{1}{T}\int_0^T S(t)\,\mathrm{d}t = \frac{1}{e_N - e_0}\sum_{i=1}^{N} S(e_{i-1}) \times (e_i - e_{i-1})$$

式中：$0 = e_0 \leqslant e_1 \leqslant \cdots \leqslant e_N = T$ 是抽样路径上的事件时间。

像 $S(t)$ 这样的连续时间输出，它们的一个特征就是在离散事件仿真系统中它们是分段连续的，因为它们只能在事件时间点改变值。

对于表 1.1 中的仿真程序，Y 的观测值是 15，而 $\bar{S}(T)$ 在 $T = 15$ 天时的观测值为

$$\frac{1}{15-0}[2 \times (5-0) + 1 \times (7.5-5) + 2 \times (8-7.5) + 1 \times (10.5-8) +$$

$$2 \times (14-10.5) + 1 \times (15-14)] = \frac{24}{15}$$

当然，这只是一种可能的输出组合 $(Y, S(Y))$。仿真重复是同一个模型在统计学上相互独立的多次仿真重复。我们会经常生成多个仿真重复改进对系统性能的估计。一个重要的区别就在于输出数据是重复内还是重复间。Y 和 $S(t)$ 是重复内的输出。系统故障次数 Y_1，Y_2，…，Y_n 和平均可工作部件数量 $\bar{S}_1(T), \bar{S}_2(T), \cdots, \bar{S}_n(T)$ 来自于 n 个不同的重复，它们是重复间输出。来自不同重复的结果本质上是独立同分布的。它们是相互独立的，因为在每次重复时重新掷骰子，它们也是同分布的，因为每次掷骰子时都使用了相同的初始条件和模型逻辑。

运行仿真程序估计系统性能，通常以此来对比各种备选的系统设计方案。当仿真满足某些版本的强大数据定理（SLLN）时，我们可以证明使用基于仿真的评估器的有效性，这一论题将在第 5 章进行更正式地论述。两个版本的 SLLN 是与随机仿真相关的：

（1）当仿真重复的数量 n 增加时，有：

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} Y_i = \mu$$

式中：μ 可以理解为 TTF 系统示例的首次系统故障平均时间。

（2）当一次仿真重复的 $T \to \infty$ 时（如果在首次系统故障时不停止仿真，这是有道理的），则：

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T S(t)\,\mathrm{d}t = \theta$$

（以概率 1），式中：θ 可以理解为 TTF 系统示例中平均可工作部件数的长运行均值。

给我们的信息是这样的，当仿真的投入增加时（仿真重复的数量、一次重复的长度或者两者都有），仿真评估器会非常明确地收敛于某些有用的系统性能度量。这是令人安慰的，但在现实中，在远短于无限的时间停止，这不能带来完全的收敛。因此，一个重要的论题就是测量停止仿真时的残留误差，随

机仿真的一个优势就是可以用统计推断来做到这一点。

1.3 问题与扩展

TTF 系统的例子说明了随机仿真的基本问题，但不是所有的问题都能发生。考虑如下因素：

（1）假设修理时间不是固定的 2.5 天，而是 1/2 的可能性为 1 天和 1/2 的可能性为 3 天。我们生成修理时间所要做的就是投硬币，但如果一个故障和一个修理事件被计划在同一时间点发生时会怎样（这种情况现在可以明确是会发生的）？这算一次系统故障吗？这与我们执行事件逻辑顺序有关吗？可否定义一个合理的关系解除规则？

（2）假设有三个部件而不是两个。我们对系统状态的定义仍然适用吗？如果可以同时修理的部件超过一个会怎样？需要增加额外的事件吗？

（3）平均可工作部件数的长运行均值不是真正的“系统可用度”。相反，我们想得到至少一个部件可工作的长运行时间比例。那么如何从表 1.1 中提取这一性能度量值呢？T 要多长就可以得到一个长运行可用性的良好估计值？

练　习

（1）独立运行 TTF 系统仿真示例，直到首次系统故障时间。估计首次系统故障时间的期望值 $E(Y)$ 以及该估计的标准差和置信区间。

（2）用三个（一个工作，两个储备）部件而不是两个运行 TTF 系统仿真示例，直至首次系统故障时间。

（3）运行 TTF 系统示例仿真直至 $T=30$ 天时并计算可工作部件平均值和系统可用度平均值。除非你很幸运，否则在时间为 30 天时没有相应的故障或修理事件发生，那么如何能在正好 $T=30$ 时停止仿真呢？提示：如果 $S(t)>0$，则系统可用。平均可用度是系统在时间 $t=0$ 到 $t=T$ 的时间段内可用时间的百分比。

（4）针对 TTF 系统的示例，假设修理时间不是确定的 2.5 天，而是 1/2 概率为 1 天、1/2 概率为 3 天。用投硬币的方法生成修理时间。确定一个合理的关系解除规则，并运行仿真到首次系统故障时间。

（5）如果在 TTF 系统的示例中，部件故障的时间和部件修理时间服从指数分布，那么 TTF 系统是一个时间连续的马尔科夫链，它的平均首次故障时间和长运行系统可用度可以从数学上推导出来。如果知道如何做，则可以照做。

（6）用文字描述 TTF 系统示例中的两个系统事件的工作逻辑。要确保包

含对状态值的更新和对所有未来事件的计划。

（7）在表1.1中增加一列，列中更新当第j个事件执行时面积$\sum_{i=1}^{j} S(e_{i-1}) \cdot (e_i - e_{i-1})$的值，需要说明的是，通过记录这一运行中的汇总数据，可以在任何事件时间e_j立即计算出$\bar{S}(e_j)$。

（8）在表1.1中增加一列，列中更新当第j个事件执行时面积$\sum_{i=1}^{j} I\{S(e_{i-1}) = 2\} \cdot (e_i - e_{i-1})$的值。这里$I$是标志函数。用它计算系统处于全功能状态（没有部件处于故障状态）的时间比例。

（9）考虑一个改进的TTF系统，它按下列方式工作。手动（使用骰子）仿真这一系统直到发生首次系统故障的时间。

- 系统有3个部件（一个工作，两个备用，但仍然是一次只能修一个部件）；修理时间是3.5天。
- 此外，每个部件是由两个子部件组成，每个子部件的故障前时间等概率地取1~6天。
- 当第一个子部件故障时部件会故障。换句话说，要仿真一个部件的故障前时间，可以掷两次骰子，然后取较小的那个数。

针对练习（9）中的仿真，增加一列更新当第j个事件执行时面积$\sum_{i=1}^{j} S(e_{i-1}) \times (e_i - e_{i-1})$的值，这里$S(t)$是在时间$t$时可工作部件的数量。用它计算可工作部件的平均数。