

离散事件系统与连续系统仿真的C语言实现

吴小钧 (西安西北工业学院计算机系 710032)

摘要:现行的计算机仿真语言分为通用仿真程序语言和专用仿真程序语言,前者如 GPSS、DYNAMD,后者如 IDIMS、GCMS 等。前者对用户是一个黑盒系统,而后者又存在专用性太强、应用范围小的局限性。本文通过两个实例的剖析和实现,力图在广大程序员都十分熟悉的 C 语言与系统仿真的实现之间架起一座桥梁。

关键词:系统仿真 离散事件系统仿真 连续系统仿真 仿真语言

一、引言

仿真(Simulation)是通过对系统模型的实验来研究一个存在的或设计中的系统。计算机仿真则是指借助计算机,用系统的模型对真实系统或设计中的系统进行试验,以达到分析、研究与设计该系统的目的。计算机仿真过程如图 1 所示:

其中仿真模型的建立是整个仿真过程的重点,其关键在于仿真算法和仿真语言的选择。现行的仿真语言一般分为两种即:通用仿真语言和专用仿真语言。通用仿真语言通常是在其他的通用高级语言(如 FORTRAN)的基础上产生的。它实质上是一个用该高级语言编写的一

个通用主程序,用户根据自己所研究的实际问题的具体情况填入参数,经执行得到仿真结果。整个仿真的实现过程对于用户来讲是一个黑盒(Black Box),用户无法了

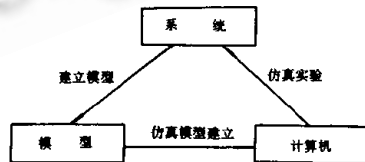


图 1 计算机仿真

解具体的仿真实现过程。此外,现实中的仿真实例无疑是千变万化的,而这些各具特色的仿真实例势必会受到通用仿真语言预先搭起的框架的束缚,即通用仿真语言缺乏灵活性。专用仿真语言针对专门领域的管理系统或专门问题的仿真,专用性过强,应用面窄。

众多程序员均得心应手的C语言,它具有表达能力强、代码质量高、可移植性好等特点,并兼备高级语言及低级语言的许多优点,已成为目前国际上应用非常广泛的主流程序设计语言。“用C语言来实现系统仿真如何?”这正是本文作者在学习系统仿真学中不断思考的一个问题,同时也是本文讨论的核心所在。

本文将结合实例,分别讨论在离散事件系统与连续系统仿真中如何用C语言来实现。

二、离散事件系统仿真实例

计算机仿真分为两大类:离散事件系统仿真和连续系统仿真。离散事件系统是指系统状态只是在离散时间点上发生变化的系统,而且这些离散时间点一般是不确定的。

例1:单服务台排队系统仿真。顾客的到达时间一般是随机的,为每个顾客服务的时间长度也是随机的。描述该系统的状态是服务台的状态(忙或闲)、顾客排队等待的队长。显然,这些状态量的变化也只能在离散随机时间点上发生。这是一个典型的离散事件系统。

1. 系统参数说明

(1)实体到达模式。设顾客到达时间间隔 A_i 服从均值为 $\beta A = 5\text{min}$ 的指数分布即:

$$f(A) = \frac{1}{\beta A} e^{-A/\beta A} \quad (A \geq 0)$$

(2)服务模式。设每个顾客接受服务的时间长度 S_i 也服从指数分布,均值为 $\beta_s = 4\text{min}$ 即:

$$f(s) = \frac{1}{\beta_s} e^{-s/\beta_s} \quad (s \geq 0)$$

(3)服务规则。鉴于单服务台系统,故采用单队列并按 FIFO 方式服务。

2. 系统模型分析

(1)用反变换法确定随机变量 A_i 与 S_i 。设 U_1 为取值为 $[0, 1]$ 范围内服从均匀分布的随机变量,由反变换法原理,令

$$U_1 = F(A) = 1 - e^{-A/\beta A}$$

$$\text{则} \quad A = -\beta A \ln(1 - U_1)$$

$$\text{令} \quad U_2 = 1 - U_1$$

$$\text{则} \quad A = -\beta A \ln U_2$$

$$\text{同理可得} \quad S = -\beta_s \ln U_3$$

其中 U_2 与 U_3 均为取值在 $[0, 1]$ 范围内的服从均匀分布的随机变量。

(2)确定系统状态变量

顾客队长:NIQ(Number In Quene)

服务员忙闲状态:STATUS(0:闲;1:忙)

(3)定义系统事件

事件类型	事件描述	属性
#1	顾客到达系统	到达时间
#2	顾客离开系统	离去时间

(4)仿真目的。对于本例,假设仿真目的是要求估计服务 n 个顾客后顾客平均队长 $\hat{Q}(n)$ 及平均排队等待时间 $\hat{d}(n)$:

$$\hat{Q}(n) = \bar{Q}(n) = 1/T \int_0^T Q(t) dt \quad (2-1)$$

$$\hat{d}(n) = \bar{D}(N) = \sum_{i=1}^n D_i/n \quad (2-2)$$

其中 D_i 为第 i 个顾客等待时间, $Q(t)$ 为 t 时刻排队等待的顾客数, T 为完成 n 个顾客服务所耗时间, $\hat{d}(n)$ 、 $\hat{Q}(n)$ 表示估计值, $\bar{D}(n)$ 、 $\bar{Q}(n)$ 表示平均值。计算机实现 (2-1) 是有困难的,鉴于本系统为离散事件系统,采用事件调度法时仿真钟的推进具有随机性,故可以用下述 (2-3) 式的实际累加操作来代替 (2-1) 式的理论估计值:

$$\hat{Q}(n) = 1/T \sum_{i=1}^m R_i \quad (2-3)$$

$$R_i = \eta_i (b_i - b_{i-1}) \quad (2-4)$$

在 (2-4) 中, η_i 为在时间区间 $[b_{i-1}, b_i]$ 上排队人数; b_i 为第 i 个事件发生的时间(任何一类事件均可); m 为 $[t_0, t_0 + T]$ 区间上发生事件总数, t_0 为仿真初始时间, T 为仿真运行总时间。

3. 仿真程序说明

(1)主程序流程图(图2)

(2)初始化子程序流程图(图3)

(3)报告子程序 report。该子程序打印如下数据:

①模型及实验参数:

平均顾客到达时间间隔: MARRVT

平均顾客接受服务时间: MSERVT

仿真长度(即接受服务的顾客总数): TOTCUS

②统计数据:

顾客排队平均队长:AVGNIQ

平均每个顾客排队等待时间:AVGDEL

(4) 随机变量产生子程序 expon。该子程序调用 TC2.0 数学库中的 rand() 库函数, 产生一个在 0~32767 之间的随机数, 并由此得到在 [0, 1] 上均匀分布的随机变量 u, 最后返回 $-r \ln u$, r 为均值 MARRVT 或 MSERVT, 视情况而定。

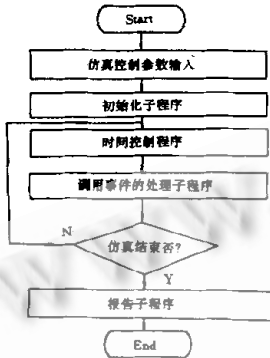


图2 主程序流程图

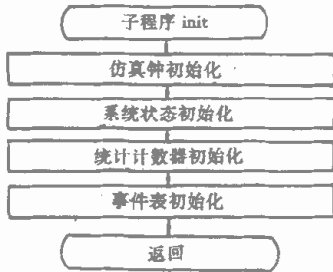


图3 初始化子程序流程图

4. 用 C 语言编写的仿真程序(略)

5. 仿真结果分析

本程序用简单易用的 TC2.0 编写, 编译顺利通过, 在笔者的 Pentium 133MHz + 16M 内存微机上运行结果如下:

仿真长度	1000	2000	3000	5000	10000
\bar{Q}	2.446661	4.553268	4.295066	3.957506	3.523758
\bar{d}	11.647000	20.733500	19.791000	18.136801	16.040501

由解析计算得到的该单服务台排队系统在 $\beta A = 5.0, \beta s = 4.0$ 时的性能稳态理论值为:

$Q = 3.2$

$d = 16.0$

将仿真结果与稳态理论值相对照, 可见 $\bar{Q}(10000) = 3.523758$ 与 $\bar{d}(10000) = 16.040501$ 非常接近稳态理论值, 因而仿真结果是可信的, 本次仿真过程是成功的。

6. 意义

本实例属于离散事件系统仿真中的典型例子, 它的适用范围很广, 如订票系统、库存系统、加工制造系统、交通控制系统、计算机系统等等, 都可以通过本例中的实现方法, 举一反三, 得到解决。

三、连续系统仿真实例

系统状态随时间连续变化的系统称为连续系统。这类系统的动态特性可用微分方程或一组状态方程来描述, 也可以用差分方程或一组离散状态方程来描述。在一些社会经济系统中, 虽然系统实际状态变化是连续的, 但系统的统计数据却往往只是按月、季度、甚至按年来得到, 因此也只能用差分方程或离散状态方程来描述。下面所举的林业系统的例子便是一个典型的用欧拉法差分方程来求解的一个连续系统仿真实例。

例 2: 某地区林业系统。该地区林区分为三部分: 宜林荒地、已造林但尚未成森林的林地以及森林地。系统的目标是改善生态环境, 达到期望的森林覆盖率。

1. 林业系统因果关系图(图 4)

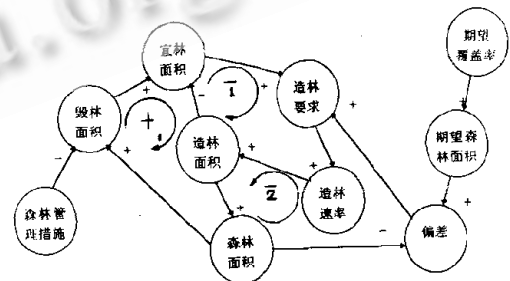


图4 林业系统因果关系图

注释: \pm 正反馈 $+$ 正反馈环
 \mp 负反馈 $-$ 负反馈环

2. 林业系统系统流图(图 5)

注释:

L_1 : 宜林荒地面积

L_2 : 造林但尚未成森林面积

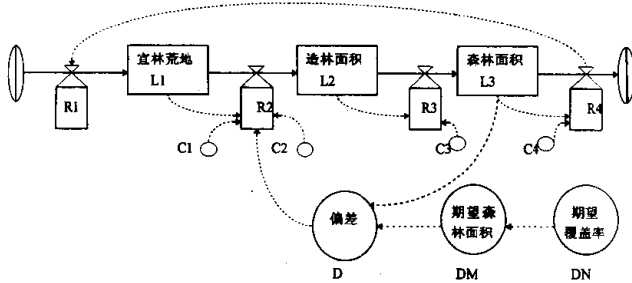


图5 林业系统系统流图

L_3 : 森林面积

R_1 : L_1 的增加率

R_2 : L_2 的增加率, 同时也是 L_1 的减少率

R_3 : L_3 的增加率

R_4 : L_3 的减少率

D: 期望森林面积与实际森林面积的偏差

DM: 期望森林面积

DN: 期望覆盖率

C_1, C_2 : 造林方案常数

C_3 : 从造林到森林郁闭成林的时间延滞常数

C_4 : 毁林率

3. 数学模型及其数值解法(略)

4. 仿真程序流程图(图6)

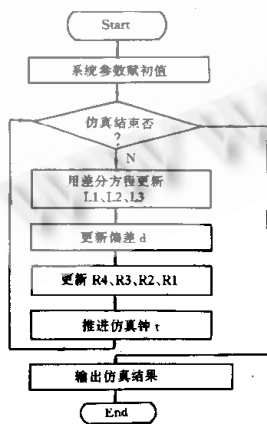


图6 仿真程序流程图

5. 用C语言编写的仿真程序(略)

6. 仿真结果分析

本程序在 TC2.0 中编译顺利通过, 在 Pentium 133MHz+16M 内存微机上运行得到的仿真结果如下:

仿真次数	C_1	C_3	C_4	Dmin	L_1	L_2	L_3
1	10	6	0.005	0.5	2.916672	1.750012	58.333392
2	10	6	0.05	0.5	17.783182	10.434235	34.782574
3	5	6	0.05	5	10.161289	12.193550	40.645153
4	2	6	0.005	0.5	0.605769	1.817308	60.576916
5	10	2	0.005	0.5	2.971729	0.594347	59.433922

把第2次仿真与第1次仿真的结果作比较, 可以看出当其他条件不变, 将毁林率增至10倍(由0.005到0.05)导致森林面积由 $L_3 = 58.333302$ 降至 $L_3 = 34.782574$ 。把第4次和第1次作比较, 当其他条件不变时, 将宜林荒地改造成造林地的时间延滞常数减少5倍(由 $C_1 = 10$ 降至 $C_1 = 2$) 导致森林面积从 $L_3 = 58.33302$ 增至 $L_3 = 60.576916$, 可见减少 C_1 为增加 L_3 的一个途径。同样, 观察第5次仿真的结果, 也不难看出, 减小从造林到森林郁闭的时间延滞常数 C_3 也是增加森林面积 L_3 的一个方法。总之, 通过计算机的多次模拟仿真得到一系列的仿真结果, 提供给用户, 以帮助用户生成满意的决策, 这也正是本仿真系统的目标和最终价值所在。

四、结论

本文所举的实例内容都比较简单, 类型也很典型, 建模时有些地方也做了适当的简化, 但这并不妨碍对这种实现途径的可行性与有效性的探索。通过以上两个实例的实现过程和运行结果可以看出: 在系统仿真学的两大分支——离散事件系统仿真和连续系统仿真——中, 用大家十分熟悉且简便易行的C语言所编写的仿真程序均达到了仿真活动所预期的目的, 并且仿真结果与理论值的吻合程度也是十分令人满意的。与此同时, 用C语言编写仿真程序使程序员始终处于主控地位, 不受制于类似通用仿真语言或专用仿真语言所强加给仿真人员的种种限制, 自由度大, 灵活性强, 并且具有很强的可移植性。希望本文能起到抛砖引玉的作用, 促使更多的C语言仿真程序出现, 推动系统仿真研究的发展。

(来稿时间: 1997年10月)