Ch 2 Modeling complex systems

- Introduction
- List Processing in Simulation
- A Simple Simulation Language: simlib
- Single-Server Queueing System with simlib
- Time-Shared Computer Model
- Multi-teller Bank with Jockeying
- Job-Shop Model
- Efficient Event-List Manipulation
2.1 Introduction

- Most real-world systems
  - Quite complex
  - A difficult and time-consuming task (without supporting software)
2.2 List processing in simulation

- Complex simulations require many lists
  - Attribute
- Use FIFO manner to process the list
- Two approaches to storing lists
  - Sequential allocation
    - E.g. list of arrival times for customers in the queue
  - Linked allocation
Advantages of linked allocation

- Time can be significantly reduced
- We can speed up event-list processing considerably
- Memory can be reduced
- It is a general framework that allows one to store and manipulate many lists simultaneously with ease.
LAS: list of available space

Arrival time: 10, 15, 25

FIGURE 2.1
State of the lists for the queueing simulation at time 25.
2.3 A Simple Simulation Language: simlib

- An easy-to-understand C-based simulation “language”
- Symbols, declarations, and definitions see Chap. 1
2.4 Single-server queueing system with simlib

- M/M/1 queueing system
2.5 Time-shared computer model

FIGURE 2.13
Time-shared computer model.
FIGURE 2.14
Event graph, computer model.
FIGURE 2.17
Flowchart for arrival function, computer model.
Function start_CPU_run

Remove job from queue and compute CPU time

Decrement this job’s remaining service time

Place job in CPU

Schedule an end-CPU-run event for this job on this pass

Return

FIGURE 2.19
Flowchart for function start_CPU_run, computer model.
FIGURE 2.21
Flowchart for function end_CPU_run, computer model.
2.6 Multiteller bank with jockeying

FIGURE 2.25
The customer being served by teller \( i = 3 \) completes service, causing the customer from the tail of queue \( j = 2 \) to jockey.
FIGURE 2.26
Event graph, bank model.
Function arrive

Schedule the next arrival event

Is a teller idle?

Yes

Tally a delay of 0 for this customer

Make the teller busy

Schedule a departure event for this customer

No

Find the number, shortest_queue, of the leftmost shortest queue

Place the customer at the end of queue number shortest_queue

Return

FIGURE 2.29
Flowchart for arrival function, bank model.
FIGURE 2.31
Flowchart for departure function, bank model.
FIGURE 2.33
Flowchart for function jockey, bank model.
2.7 Job-shop model

- A manufacturing shop consists of 5 groups of machines
- Groups 1, 2, ..., 5 consist of 3, 2, 4, 3, and 1 machines
- There are 3 types of jobs
- The routing for the different job types

<table>
<thead>
<tr>
<th>Job type</th>
<th>Machine groups in routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3, 1, 2, 5</td>
</tr>
<tr>
<td>2</td>
<td>4, 1, 3</td>
</tr>
<tr>
<td>3</td>
<td>2, 5, 1, 4, 3</td>
</tr>
</tbody>
</table>
FIGURE 2.37
Manufacturing system with five work stations, showing the route of type 1 jobs.
FIGURE 2.38
Event graph, job-shop model.
FIGURE 2.41
Flowchart for arrival function, job-shop model.
FIGURE 2.43
Flowchart for departure function, job-shop model.
2.8 Efficient event-list manipulation

- Efficient searching method
- Data structures etc.