Simulation Techniques

for

Queues and Queueing Networks

Terminology

System The collection of interacting objects that need to be simulated

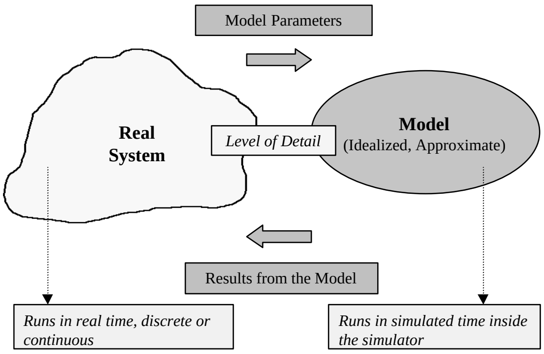
Entity A particular object of interest in the System

Attribute Some relevant property of an Entity that is sought to be studied through simulations

State The set of variables that are required to describe the system

Event The changing of the system from one state to another

Simulation Model of a Real System



System State may be Continuous or Discrete

Model should imitate the real system in as much detail as possible or in as much detail as needed

- The Real System (*Queuing Network*) will have its

  ∗ functional entities (*queues, servers, routes etc.*)

  ∗ interactions and interdependencies between the entities as a function of time

- Model simplifies the system for study but its results must be such that it correlates well with the behavior of the Real System

Simulation vs. Analysis ?

Advantages of Simulation

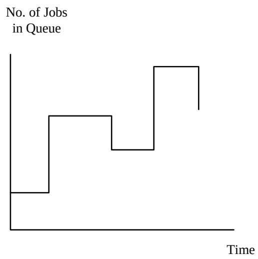
- usually closer to real system with fewer simplifying assumption
- model structure, algorithms and variables may be changed quickly to see how it affects the system
- may be able to provide performance results which are not obtainable through analytical models

Disadvantages of Simulation

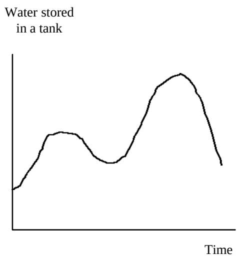
- simulators may take longer to construct and debug
- validation may take additional time and effort
- computationally expensive as a large number of long simulation runs may be needed
- relationships between model variables may be difficult to visualize
- sensitivity analysis may be difficult

 5

Simulations may be Discrete/Continuous Time and/or Discrete/Continuous State



Discrete State, Continuous Time



Continuous State, Continuous Time

 6

Simulated Times or **Event Times** are the times at which the simulation model's state undergoes a change. *Nothing changes in the model between these event times.*

In a Queue or a Queuing Network, these Event Times would be the various arrival/departure instants to/from the queue(s)

Continuous Simulation

• Uses a continuous simulation clock which is typically incremented following a *fixed-increment-time-advance* approach

• The time increments should be as fine as possible and should at least be small enough to capture all state changes in the system

• Inefficient as there will be a lot of time instants when no change would be observed in the system

Discrete Event Simulation

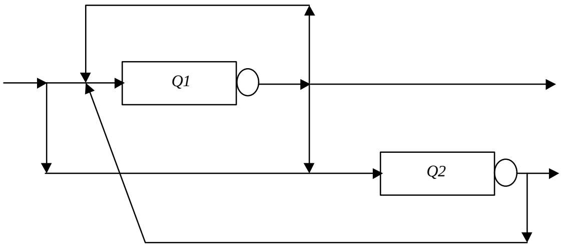
• Focus only on system changes at event times

• After processing the current event, forward system clock to the next event time (The clock jumps may vary in size.)

> *Simulation moves from the current event to the event occurring next on the event list that is generated and updated for the system.*
>
> *Processing the current event may create additional events which are placed appropriately in the event list*

• Fast and efficient as only system behaviour at event times need to be considered

An Example Queueing Network (to be simulated)



System Descriptors:

Number of Servers in Q1 | Number of Servers in Q2
Number of Buffers in Q1 | Number of Buffers in Q2

External Arrival Process (to Q1 alone in this case)
Service Process at Q1 | Service Process at Q2

Process Descriptions may be simplified approximations or may be complex stochastic descriptions

Routing Probabilities

Ext. ® Q1, Q2 *Q1 ® Ext., Q1, Q2* *Q2 ® Ext., Q1, Q2*

Blocking Mechanism if Finite Buffers at Q1, Q2

Service Discipline (*usually FIFO but may be different*)

System State: **(N_1, N_2)**

N_1 *Number in Q1* *(including the ones in service)*
N_2 *Number in Q2* *(including the ones in service)*

Events to be considered in the Simulation -

* Arrivals from External World
* Departures to External World
* Arrivals to Q1
* Arrivals to Q2
* Service Completions at Q1
* Service Completions at Q2

These events are inter-related with one event triggering a subsequent one

Performance Parameters of Interest

The simulator may provide some or all of these as output

∗ Number in Q1 (including/excluding those in service)
∗ Number in Q2 (including/excluding those in service)
First and Second Moments, Higher Moments, Distribution
∗ Waiting Time in Q1 and/or Total Time spent in Q1 by a job entering Q1
∗ Waiting Time in Q1 and/or Total Time spent in Q1 by a job entering Q1
First and Second Moments, Higher Moments, Distribution

 13

Performance Parameters of Interest *……….continued …...*

∗ Server Utilization at Q1 and Q2
∗ Sojourn Time of a job entering the network
∗ Effective Arrival Process entering Q1
∗ Effective Arrival Process entering Q2
∗ Departure Process from Q1 and Q2 to Q1 and/or Q2
∗ Departure Process from Q1 and Q2 to the External World
∗ Blocking/Loss Probability for the various flows

The simulator must incorporate additional variables (counters etc.) to record data during simulation. These will be used to provide the statistics etc. required to obtain the performance results desired.

 14

Simulation Algorithm

Main Loop

1. Process *current event* on top of the event list

   This may create additional events which will then have to be inserted in their proper place in the event list

2. Move to the *next event* on the event list

3. Repeat from Step 1 until the *termination conditions* stipulated are reached

Apart from the *Main Loop*, the simulator must also do the following -

• Have appropriate procedures for processing each type of event

> These will include all the actions that need to be taken for processing the event (including generation of other events, if required) and collection of data for statistics

• Processing and Reporting of Results

• Computation of *Confidence Levels* and *Confidence Intervals*

Copyright 2002, Sanjay K. Bose

Event Processing (for the example queueing network)

External Arrival Event -

[1] Schedule next external arrival event and place it in the right place on the event list

[2] Toss a random coin to decide whether arrival should go to Q1 or Q2. Convert the event to one of an arrival to Qi (i=1,2) based on this; however, reschedule it in the same place as before on the event list

For this system, the simulation process may be started by putting *one external arrival event* on the event list and scheduling it to occur at *t=0*

Arrival Event to Qi i=1, 2

[1] If no free server available, increment by one the number in the buffer in Qi

This makes the job wait in queue for service later

[2] If a server is available, start the job's service by -

(i) Increasing by one the number of busy servers

(ii) Scheduling a departure event corresponding to this job and placing it properly in the event list

Departure Event from Qi i=1, 2

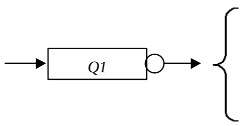
[1] Toss a random coin to decide the routing appropriately

[2] If departure is to the external world, then the job just leaves the system

[3] If departure is for Qj then convert the event to one of an arrival to Qj (j=1,2) based on this; however, reschedule it in the same place as before on the event list

Additional software will be needed to incorporate various counters etc. to gather statistical results for processing once the simulation is over.

Collecting and Processing Simulator Outputs



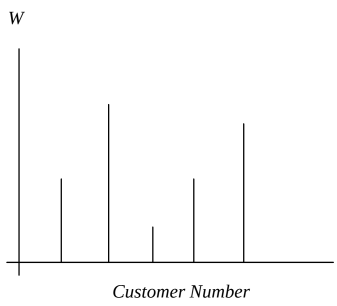
Observation Based Variables

Time Weighted Variable

Observation Based Variables are based on individual observations, e.g. Service Time of a Job or Waiting Time of a Job in the queue

$\{W_1, W_2, \ldots\ldots\ldots\ldots\ldots\ldots W_K\}$

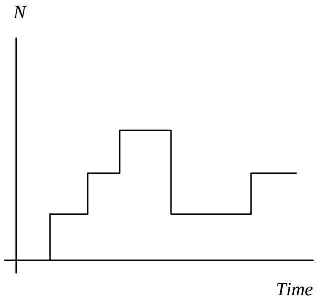
$E\{W\}$ and σ_W^2



Time-weighted Variables are based on the extent of their actual duration and have to be weighted with these duration

$\{N(t),\ 0 \le t \le T\}$

$E\{N\}$ and σ_N^2



Important to remember that it is meaningful to gather statistical results from a simulation run only when the simulated system is in steady state

Observed Means and Variances

Observation Based Variable

Observed Mean $W_O = \frac{1}{K}\sum_{j=1}^{K} W_j$

Observed Variance $s_{WO}^2 = \frac{1}{(K-1)}\sum_{j=1}^{K}(W_j - W_O)^2$

Time Weighted Variable

Observed Mean $N_O = \frac{1}{T}\int_{t=0}^{T} N(t)dt$

Observed Variance $s_{NO}^2 = \frac{1}{T}\left[\int_{t=0}^{T} N^2(t)dt\right] - N_O^2$

A simulation run provides only *observed moments* based on the results of that run

- *No guarantee that the observed values of the moments are the same as or are close to the actual moments of the random variable if its distribution were known.*
- *This is why a number of independent simulation runs are required to provide confidence estimation on the observed results*

The observed moments obtained from a simulation run are meaningful only when the moments actually exist and can be properly defined.

- Computation of moments meaningful only for a system in equilibrium

  Valid mean results cannot be obtained for overloaded systems which never reach steady state

- To get proper results, an initial *warm-up* period must be ignored to let the system reach equilibrium before results are collected

Confidence Interval Estimation

Point Estimation of the output variables of a simulation experiment is of limited use **-**

∗ if the variance is large, the actual "spread" of the observed random variable will be large

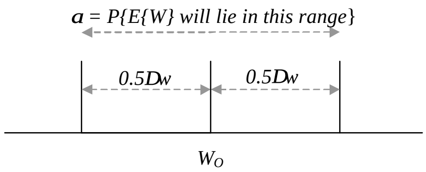
∗ there is no estimate as to the degree of confidence one can place in the point estimates that have been obtained

This is why **Confidence Interval Estimation** is required

Random Variable is W with actual mean $E\{W\}$ and observed mean W_O

Dw = confidence interval

a = confidence level



The true mean E{W} of the random variable is expected to lie between ±0.5Dw of the calculated mean W_O *with a probability of* a

or

On the average, we would be correct (a x *100*) percent of the time, that $E\{W\}$ will indeed lie in the range (W_O - $0.5Dw$, $W_O+0.5Dw$)

Procedure for Confidence Estimation

1. Observe the required variable W for n independent runs

The runs must be independent. This is ensured by choosing different seeds for the random number generators for each run.

Let W_{oj} be the sample mean of W from the j^{th} run

2. From these n observations, compute the observed mean W_O and the observed variance s_{WO}^2 as follows

$$W_O = \frac{1}{n}\sum_{j=1}^{n} W_{Oj} \qquad s_{WO}^2 = \frac{1}{(n-1)}\sum_{j=1}^{n}(W_{Oj} - W_O)^2 \qquad (7.5)$$

3. Using appropriate tables, obtain the Student's t-parameter $t_{0.5(1-a),(n-1)}$ for $(n-1)$ degrees of freedom.

For example, with $a=0.95$: $t_{0.025,5} = 2.571 \qquad t_{0.025,10} = 2.228$

4. Use the following to compute the range of the confidence interval within which we would expect the actual mean $E\{W\}$ to lie with (a x 100) percent confidence.

$$\left(W_0 - \left[t_{0.5(1-a),(n-1)} \right] \frac{s_{WO}}{\sqrt{n}}, W_0 + \left[t_{0.5(1-a),(n-1)} \right] \frac{s_{WO}}{\sqrt{n}} \right) \qquad (7.6)$$

If the confidence interval obtained is not tight enough then more simulation runs would be required until the desired tightness is obtained

Disclaimers!

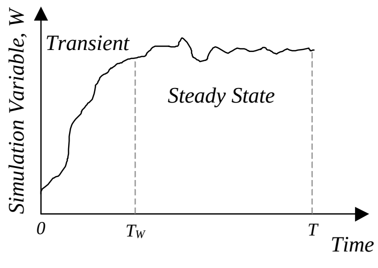
• Generally, increasing the number of runs will reduce the confidence interval for a given confidence level but this is not always guaranteed.

• It is not possible to specify *apriori* the number of runs needed to get the desired tightness in the confidence interval. One will actually have to keep doing more runs of the simulation experiment until the desired confidence level is obtained.

• If a number of variables are being observed, the runs will have to be continued until the required confidence intervals of all the variables are satisfied.

Even if it is not feasible to obtain confidence estimates for each variable observed over a range (such as in a graph), such estimates must be done for some typical points to satisfy oneself that the results are all right.

Transient Behaviour and the Warm-up Interval



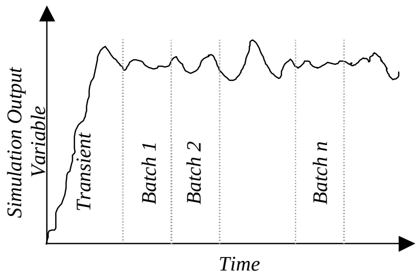
• Equilibrium behaviour after time instant T_W

• Warm-up interval should be $(0,T_W)$ or more

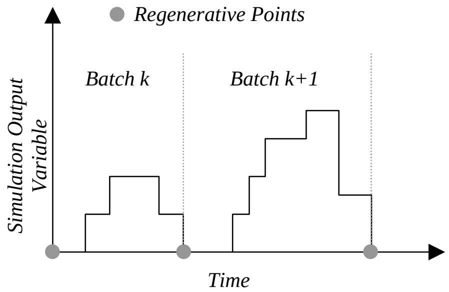
• Statistics should be collected only after the warm-up interval is over and the system has reached equilibrium

No apriori guidelines can be given about the choice of the warm-up interval. The best strategy is to actual observe a few runs to see what the transient behaviour looks like before choosing the warm-up interval to be omitted while gathering equilibrium results

Data Collection in Steady State Conditions

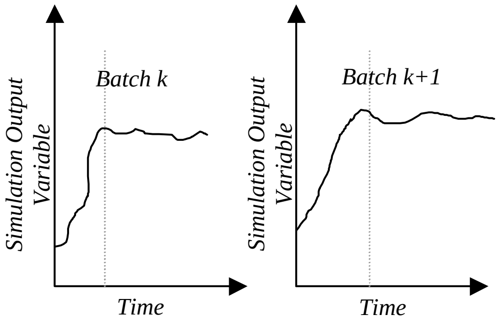


Subinterval Method or the Method of Batch Means



The Regenerative Method

Still extracts multiple runs from a single long run but reduces the correlation one may get between batches in the Method of Batch Means



Separate batches run with different initial random seed from the random number generator

The Replication Method