Some Stochastic Process Models for Discrete-Event Stochastic Systems

1. Discrete-Event Stochastic Systems

Recall our previous definition of a discrete-event stochastic system: the system makes stochastic state transitions only at an increasing sequence of random times. If $X(t)$ is the state of the system at time $t$, then a typical sample path of the underlying stochastic process of the simulation, i.e. $(X(t): t \geq 0)$, looks like this (for a finite or countably infinite state space):

For a given simulation model, we need to precisely specify the process $(X(t): t \geq 0)$, so that we can generate sample paths of the process and obtain meaningful point and interval estimates of system characteristics based on well-defined properties of the process. We will describe several types of stochastic processes that can serve as the underlying process of a simulation.

2. Discrete-Time Markov Chains (DTMC’s)

Sometimes we are interested in the discrete-time process $(X_n: n \geq 0)$, where $X_n$ is the state of the system just after the $n^{th}$ state transition. (Sometimes we might only look at the system state at a specified subsequence of the state transitions.) If the state space is finite or countably infinite and the Markov property

$$P\{X_{n+1} = x | X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_0 = x_0\} = P\{X_{n+1} = x | X_n = x\}$$

holds, then $(X_n: n \geq 0)$ is a discrete-time Markov chain. A (time-homogeneous) DTMC can be specified by giving:

1. the transition matrix $P$, where

$$P = (P(x, y): x, y \in S).$$

Here $P(x, y) = P\{X_{n+1} = y | X_n = x\}$ and $S$ is the state space of $(X_n: n \geq 0)$.

2. the initial distribution $\mu$, where

$$\mu = (\mu(x): x \in S).$$

Here $\mu(x) = P\{X_0 = x\}$ for $x \in S$. 

Page 1 of 14
Example (The Markovian Jumping Frog of Calaveras County):

The frog starts in states 1 and 2 with equal probability.

Let \( X_n \) be the lily pad occupied by the frog at jump \( n \). Then \( (X_n; n \geq 0) \) is a DTMC, with

\[
P = \begin{bmatrix}
0 & 1/2 & 1/2 \\
1/3 & 0 & 2/3 \\
3/4 & 1/4 & 0
\end{bmatrix}
\quad \text{and} \quad
\mu = \begin{bmatrix}
1/2 \\
1/2 \\
0
\end{bmatrix}
\quad \text{(we’ll assume that all vectors are column vectors)}.
\]

Suppose that we are interested in the probability that the frog will be on lily pad 2 after the \( k \)th jump, where \( k \) is fixed. We can express this quantity in the form\( E[f(X_k)] \), where \( f(x) = 1 \) if \( x = 2 \) and \( f(x) = 0 \) if \( x = 1 \) or 3---the function \( f \) is called an indicator function and we sometimes write \( I(X_k = 2) \) instead of \( f(X_k) \). We can compute \( E[f(X_k)] \) in one of two ways:

1. **Numerically**, letting \( v_n(i) = P(\text{frog on pad } i \text{ after } n \text{th jump}) \) and writing
   \[
   v_n = \begin{bmatrix}
v_n(1) \\
v_n(2) \\
v_n(3)
\end{bmatrix}
   \quad \text{for } n \geq 0:
   \]

   Set
   \[
   v_0 = \mu
   \quad \text{and} \quad
   v_{m+1}^i = v_m^i P \quad \text{for } m \geq 0.
   \]
   (Here \( x^t \) denotes transpose of \( x \). The 2nd equation is equivalent to
   \[
   v_{m+1}(j) = \sum_{i=1}^{3} v_m(i) P(i,j)
   \]
   If we write \( f = (f(x): x \in S) \) as a column vector, then
   \[
   E[f(X_k)] = v_k^t f.
   \]

2. **Via simulation** (simulation is especially needed for more estimating more complicated probabilities)

   Then we need a means of simulating DTMC’s.
3. Simulation of DTMC’s

We start by generating $X_0$ according to distribution $\mu$. To do this, we need a means of generating discrete random variables from uniformly distributed random variables.

**Naive Method for Generating a Discrete Random Variable:**

To generate $Y$ having mass function $P(Y = y_i) = p_i$ for $1 \leq i \leq m$, split the unit interval into $m$ subintervals, the $i^{th}$ subinterval having length $p_i$. Generate a uniform$(0,1)$ random variable $U$. If $U$ falls in the $j^{th}$ subinterval, return $Y = y_j$.

**Example:**

$m = 3$

$P(Y = y_1) = 3/12$

$P(Y = y_2) = 8/12$

$P(Y = y_3) = 1/12$

<table>
<thead>
<tr>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/12</td>
<td>11/12</td>
<td>12/12</td>
</tr>
</tbody>
</table>

If $U = 0.27$, then return $Y = y_2$.

Disadvantage: If $m$ is large, it could take a while to find the subinterval containing $U$. (This process can be speeded up however, using the “coding vector” method---see L&K p. 492---or, even better, binary search.)

We will discuss a much faster technique, called the alias method, shortly.

**Generating the DTMC:**

To simulate the DTMC, generate $X_0$ from $\mu$ as above. To generate $X_{m+1}$, generate $Y$ according to the distribution $P(X_m, \cdot)$, and set $X_{m+1} = Y$. If, in the jumping frog example, $X_m = 3$, then $X_{m+1} = 1$ with probability $3/4$ and $X_{m+1} = 2$ with probability $1/4$.

To estimate $\theta = E[f(X_k)]$, generate $X_0, X_1, \ldots, X_k$ and compute $Z = f(X_k)$. Repeat this process $n$ times, thereby obtaining $Z_1, Z_2, \ldots, Z_n$. Then
\[ \theta_n = \frac{1}{n} \sum_{i=1}^{n} Z_i \]

is an estimator of \( \theta \). Confidence intervals can be constructed as usual.

We can generalize this procedure to estimate more complicated quantities, e.g., \( \theta = \mathbb{E}[Z] \), where \( Z = f(X_0, X_1, \ldots, X_k) \) for some fixed \( k \). For example, \( Z \) might be the number of times in the first \( k \) jumps that the frog goes from pad 2 to pad 3.

Sometimes it is best to use a simulation method specially tailored to the model at hand.

**Example (An \((s, S)\) Inventory System):**

Let \( X_n = \) inventory level at the end of period \( n \)
\( D_n = \) demand in period \( n \)

If an \((s, S)\) inventory policy is followed, then
\[
X_{n+1} = \begin{cases} 
X_n - D_{n+1} & \text{if } X_n - D_{n+1} \geq s \\
S & \text{if } X_n - D_{n+1} < s
\end{cases}
\]

If \((D_n: n \geq 1)\) is i.i.d., then \((X_n: n \geq 0)\) is a DTMC with state space \( \{s, \ldots, S\} \).

To simulate this model, it is best to simulate the chain directly from the above recursion (rather than to compute \( P \) and use the ideas previously discussed).

**Advantages:**

- only need to write a single routine to generate the non-uniform \( D_n \)'s (rather than a separate non-uniform procedure for each row of \( P \)).
- avoids the (often substantial) time required to generate all the entries of \( P \).

**Model Critique:**

- assumes no lag in delivery of orders
- assumes \( D_n \)'s are independent
- assumes no seasonal trends in demands

**Stationary distribution of a DTMC**

A distribution \( \pi \) is a stationary (or steady-state) distribution of \((X_n: n \geq 0)\) if \( X_n \) distributed according to \( \pi \) implies \( X_{n+1} \) is distributed according to \( \pi \). This implies
\[ \pi(j) = P(X_{n+1} = j) = \sum_i P(X_{n+1} = j \mid X_n = i) P(X_n = i) = \sum_i P(i, j) \pi(i) \]

or, in matrix form, \( \pi' = \pi' P \).

If the initial distribution is \( \pi \), then \( X_n \) is distributed according to \( \pi \) for \( n \geq 0 \). Under appropriate conditions,

\[ \lim_{n \to \infty} P(X_n = i) = \pi(i) \]

for any initial distribution of the chain. How might we estimate, say a steady-state expectation of the form \( \theta = E[f(X)] \) where \( X \) is distributed according to \( \pi \)? I.e., \( \theta = \sum_i f(i) \pi(i) \). We’ll talk about this problem later on.

4. General State Space Markov Chains (GSSMC’s)

These are similar to DTMC’s, but instead of a transition matrix, we have a transition kernel: \( P(x, A) \) is the probability, starting in state \( x \), that the next state is an element of the set \( A \). (We do things this way to handle, for example, the case where the state space \( S = [0, \infty) \), and the probability of hitting a specific state \( x \in S \) is zero.) To completely specify \( P(x, A) \), it usually suffices to give the value of \( P(x, A) \) for a “sufficiently large” class of sets \( A \). For example, when \( S = [0, \infty) \), it suffices to specify \( P(x, A) \) for each set of the form \( A = [0, a] \), where \( a > 0 \). I.e., it suffices to specify \( P\{X_n \leq a \mid X_{n-1} = x\} \).

To simulate a GSSMC, we usually use a method tailored to the model at hand.

**Example (A Continuous (s, S) Inventory System):**

Consider an inventory system as before, but now suppose that each \( D_n \) is a continuous random variable. (I.e., the product that is being inventoried is more like gasoline or maple syrup than like cars or toasters.) Then \( (X_n: n \geq 0) \) obeys the same recursion as in the discrete case, and hence is a GSSMC. As before, simulation of \( (X_n: n \geq 0) \) can be based on the recursion.

**Example (Waiting Times in the GI/G/1 Queue):**

Consider a service center with a single server and an infinite-capacity waiting room. Jobs arrive one at a time and are served according to a first-come, first served (FCFS) service discipline. Successive interarrival times are i.i.d., as are successive service times.
\[ w_n = \text{the waiting time (exclusive of service) of the } n\text{th customer to arrive at the queue.} \]
\[ a_n = \text{arrival time of the } n\text{th customer} \]
\[ d_n = \text{departure time of the } n\text{th customer} \]
\[ v_n = \text{processing time of the } n\text{th customer} \]

Then \[ d_n = a_n + w_n + v_n. \]

It is easy to see that
\[
\begin{align*}
W_{n+1} &= [D_n - A_{n+1}]^+ = [A_n + W_n + V_n - A_{n+1}]^+ = [W_n + V_n - I_{n+1}]^+,
\end{align*}
\]
where \( I_{n+1} = A_{n+1} - A_n \) is the \((n+1)\)th inter-arrival time and \([x]^+ = \max(x,0)\). Thus, \((W_n: n \geq 0)\) is a GSSMC.

If we have a means of generating the \( V_n \)'s and \( I_n \)'s, the above expression provides a way of recursively computing the \( W_n \)'s.

Note that we are cheating a little: the process \((W_n: n \geq 0)\) is not obtained by viewing the underlying process of a DESS at state-transition times. Such sequences of delays are so important in practice, however, that they merit our scrutiny. The sequence of delays in the GI/G/1 queues in particular has been studied exhaustively.

5. **Continuous-Time Markov Chains (CTMC’s)**

See pp. 465-473 in L&P, or Ch. 6 in Ross. If the underlying process of the simulation \((X(t): t \geq 0)\) has a finite or countably infinite state space and satisfies the Markov property:

\[
P\{X(t+u) = x \mid X(s): 0 \leq s \leq t\} = P\{X(t+u) = x \mid X(t)\}
\]

then \((X(t): t \geq 0)\) is a CTMC.

A (time-homogeneous) CTMC can be specified by giving:

1. the rate matrix (or generator matrix) \( Q \), where

\[
Q = (Q(x, y): x, y \in S) \quad \text{and} \quad Q(x, y) = \text{rate at which } X \text{ jumps from } x \text{ to } y, \quad x \neq y.
\]

Informally,

\[
P\{X(t+h) = y \mid X(t) = x\} \approx Q(x, y) h
\]

for small \( h \). We also set \( Q(x,x) = -\sum_{y \neq x} Q(x,y) \) for each \( x \).

Set \( q(x) = -Q(x,x) = \sum_{y \neq x} Q(x,y) = \text{rate at which the process jumps out of } x. \)

Informally,

\[
P\{\text{jump in } [t,t+h) \mid X(t) = x\} \approx q(x) h
\]
for small $h$. We also call $q(x)$ the intensity of state $x$. The $Q$ matrix has the following properties:

- $Q(x, y) \geq 0$ for $x \neq y$
- $\sum_{y \in S} Q(x, y) = 0$ for $x \in S$

Sometimes it is convenient to represent the rate matrix $Q$ graphically using a rate diagram. In such a diagram, each state is represented as a node. If $Q(x, y)>0$, then there is a directed arc from state $x$ to state $y$ labeled with the rate $Q(x, y)$.

(2) the initial distribution $\mu$, where

\[ \mu = (\mu(x) : x \in S). \]

Here $\mu(x) = P\{X(0) = x\}$ for $x \in S$.

Sample Path Properties:

Given that a CTMC has been in a given state for $u$ time units, what is the probability that the CTMC will remain in the state for $v$ more time units? By the Markov property, this probability cannot depend explicitly on the past history, and hence on $u$. In particular, the probability must be the same as for the case in which we have just jumped into the state. That is, if $T$ is the holding time in the state, then

\[ P\{T > u + v | T > u\} = P\{T > v\} \]

There is only one family of distributions that have this property: the exponential distributions.

Digression on the Exponential Distribution:

The exponential distribution with rate (or intensity) $\lambda$ has density function $f$ and distribution function $F$ given by

\[ f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}, \quad F(x; \lambda) = \begin{cases} 1 - e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}. \]

The mean of the distribution is $1/\lambda$. Moreover, the exponential distribution has the following three key properties: if $X \overset{D}{=} \exp(\lambda)$ and $Y \overset{D}{=} \exp(\mu)$, then

\[ \min(X, Y) \overset{D}{=} \exp(\lambda + \mu) \quad (\text{independent of whether } X < Y \text{ or } Y < X), \]

\[ P\{X < Y\} = \frac{\lambda}{\lambda + \mu}, \]

and
P\{X > a+b | X > a\} = e^{-\lambda b}.

The first two properties generalize to an arbitrary number of exponentially distributed random variables. The third property is called the “memoryless” property of the exponential distribution.

Back to Sample Path Properties

Using the Markov property along with properties of the exponential distribution, it can be shown that \((X_n: n \geq 0)\), the sequence of states visited by the chain, is a DTMC with transition matrix \(R\), where

\[
R(x, y) = \begin{cases} 
\frac{Q(x, y)}{q(x)} & \text{if } x \neq y \\
0 & \text{if } x = y.
\end{cases}
\]

Moreover, given \((X_n: n \geq 0)\), the successive holding times \((T_n: n \geq 0)\) form an independent sequence with

\[
P\{T_n > t | X_n = x\} = e^{-q(x)t}.
\]

That is, \(T_n\) is exponentially distributed with rate \(q(x)\).

Example (Poisson Process):

Consider a CTMC, denoted by \((N(t): t \geq 0)\), such that \(N(0) = 0\) with probability 1 and the rate diagram looks as follows:

In this case, \(q(x) = \lambda\) for \(x \geq 0\). The rate matrix \(Q\) and transition matrix \(R\) are

\[
Q = \begin{bmatrix}
0 & -\lambda & \lambda & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -\lambda & \lambda & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & -\lambda & \lambda & . & . & . \\
3 & 0 & 0 & 0 & -\lambda & . & 0 & . \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}, \quad R = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & . & . & . & . \\
2 & 0 & 0 & 0 & 0 & . & 0 & . \\
3 & 0 & 0 & 0 & 0 & . & 0 & . \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}
\]
From our results on sample path properties, we know that the process \( (N(t): t \geq 0) \) starts in state 0. At every state transition, the state of the process increases by 1. The holding times in successive states are i.i.d. according to an \( \exp(\lambda) \) distribution. It can be shown that

\[
P(N(t+s) = m+n \mid N(t) = m) = \frac{e^{-\lambda s} \left(\frac{\lambda s}{n!}\right)^n}{n!} \quad \text{for all } t \text{ and } m.
\]

That is, the number of upward jumps that occur within a specified time interval follows a Poisson distribution. Consequently, \( (N(t): t \geq 0) \) is called a Poisson process with rate \( \lambda \).

A common example of a Poisson process occurs when the successive interarrival times to a queue are i.i.d. according to an \( \exp(\lambda) \) distribution. Then \( N(t) \) is the number of arrivals to the queue in the time interval \([0, t]\).

Poisson processes arise in other settings also. For example, if one has a component with an exponentially distributed lifetime that is immediately replaced with an identical component, then the total number of replaced components in \([0, t]\) is given by a Poisson process \( (N(t): t \geq 0) \).

Remark:

A Poisson process is a special type of renewal (counting) process. Specifically, suppose that \( T_n \), the time at which the \( n \)th event occurs, is given by \( T_n = \tau_1 + \tau_2 + \cdots + \tau_n \), where the \( \tau_i \)'s are i.i.d. positive random variables. Then \( N(t) = \max \{n \geq 0: T_n \leq t\} \) counts the number of events that occur in \([0, t]\), and is called a renewal counting process. A Poisson process with rate \( \lambda \) is the special case in which \( \tau_i \) is an exponential random variable with mean \( \lambda^{-1} \).

**Modeling a System as a CTMC**

In the above discussion, we were given input data \( Q \) and then we defined a process with exponential holding times and DTMC jumps. We can also reverse this process. Suppose that we are told that a discrete-event process \( (X(t): t \geq 0) \) jumps from state \( x \) to state \( y \) with probability \( R(x, y) \) for \( x \neq y \) and the process never jumps from a state back to itself, so that \( R(x, x) = 0 \) for every state \( x \). Moreover, given the sequence of states visited by the process, the holding times are mutually independent, with the holding time in state \( x \) being exponential with rate \( q(x) \). Then it can be shown that \( (X(t): t \geq 0) \) is a CTMC with rate matrix given by \( Q(x, y) = q(x)R(x, y) \) for \( x \neq y \).

In particular, suppose that we have a system in which, after the state has changed to \( x \), the occurrence of one of events \( e_1, e_2, \ldots, e_m \) will cause a state transition. The occurrence of event \( e_i \) causes a transition to state \( y_i \), where \( y_1, y_2, \ldots, y_m \) need not be distinct. If the time \( T_i \) until event \( e_i \) occurs is \( \exp(\lambda_i) \), then the holding time in state \( x \), which is given by \( T = \min(T_1, T_2, \ldots, T_m) \), is exponentially distributed with rate \( q(x) = \lambda_1 + \lambda_2 + \cdots + \lambda_m \) by the basic properties of the exponential distribution. The probability that the occurrence of event \( e_j \) triggers the state transition is \( P(T_j < T_i \text{ for } j \neq i) = \lambda_j / q(x) \), independent of the holding time, again by basic properties of the exponential distribution. It follows that the probability that the new state of the system is \( y \) is given by \( R(x, y) = \sum_{j \neq i} \lambda_j / q(x) \), where \( I_y = \{i: y_i = y\} \). Let \( X(t) \)
denote the state of the system at time t. Then it follows from our discussion that \((X(t); t \geq 0)\) is a CTMC with rate matrix \(Q\) given by \(Q(x, y) = q(x)R(x, y) = \sum_{j=1}^{\infty} \lambda_j\). We will give a slightly more general version of this result shortly. The rule of thumb is that if all of the probability distributions in sight are exponential, then the system can probably be modeled as a CTMC.

**Example (M/M/1 Queue):**

Consider a GI/G/1 queue in which the interarrival time distribution is \(\exp(\lambda)\) and the service time distribution is \(\exp(\mu)\). (Thus the arrival process is a Poisson process.) This queue is denoted by M/M/1 because of the Markovian (i.e., exponential) distributions. Let \(X(t)\) be the number of jobs in the system at time \(t\). Suppose that at time \(t\) the state changes to \(x\) (where \(x > 0\)). Then the events that can cause a state transition are \(e_1 = \text{“arrival of job”}\) and \(e_2 = \text{“completion of service”}\). By the memoryless property of the exponential distribution, the time until these events occur are distributed according to an \(\exp(\lambda)\) and an \(\exp(\mu)\) distribution, respectively. Similarly, in state 0, the only event that can occur is \(e_1 = \text{“arrival of job”}\); the time until this event is \(\exp(\lambda)\). Thus we can specify \((X(t); t \geq 0)\) as a CTMC with rate diagram

```
0 1 2 3 ....
\lambda \lambda \lambda \lambda ....
\mu \mu \mu \mu ....
```

Here, \(q(0) = \lambda\) and \(q(n) = \lambda + \mu\) for \(n \geq 1\). The corresponding rate matrix \(Q\) and transition matrix \(R\) are given as follows:

\[
Q = 
\begin{bmatrix}
0 & -\lambda & 0 & 0 & \cdots & \cdots \\
1 & -\lambda - (\lambda + \mu) & \lambda & 0 & \cdots & \cdots \\
2 & 0 & -\lambda - (\lambda + \mu) & \lambda & 0 & \cdots \\
3 & 0 & 0 & -\lambda - (\lambda + \mu) & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \\
\end{bmatrix}
\]
$$R = \begin{bmatrix}
0 & 1 & 2 & 3 & \ldots & \ldots & \ldots \\
0 & 1 & 0 & 0 & \ldots & \ldots & \ldots \\
\mu/(\lambda + \mu) & 0 & \lambda/(\lambda + \mu) & 0 & \ldots & \ldots & \ldots \\
0 & \mu/(\lambda + \mu) & 0 & \lambda/(\lambda + \mu) & \ldots & \ldots & \ldots \\
0 & 0 & \mu/(\lambda + \mu) & 0 & \ldots & 0 & \ldots \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\
\vdots & \vdots & \vdots & \vdots & \vdots & 0 & \ddots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \mu/(\lambda + \mu) & \ddots
\end{bmatrix}$$

Model Critique:

1. The arrival rate is assumed constant; this precludes modeling "time of day" or seasonality effects.
2. Inter-arrival times are assumed independent, as are processing times.
3. Processing times are assumed to be exponentially distributed.

6. Simulation of CTMC’s

To compute, say, $E[f(X(t))]$ analytically, for some function $f$, we need to solve the system of differential equations

$$\dot{w} = w'Q$$
$$w(0) = \mu$$

and then set $E[f(X(t))] = f'w$. Here $f = \{f(x): x \in S\}$, $Q$ is the rate matrix, $w(t) = \{w(t; x): x \in S$ and $t \geq 0\}$, where $w(t; x) = P\{X(t) = x\}$, and $\dot{w}$ refers to the derivative of $w$ with respect to time.

This calculation can be highly nontrivial (especially if the state space $S$ is infinite!). We’ll focus on simulation methods for CTMC’s.

From the foregoing discussion, we see that a CTMC can be simulated by generating successive states according to our DTMC generation procedure and generating holding times in each state according to an exponential distribution:

Algorithm for Simulating a CTMC:

1. (Initialization) Set $n = 0$ and generate the initial state $X_0$ according to $\mu$.
2. Generate an exponential random variable $V$ with rate $q(X_n)$ and set $T_n = V$.
3. Generate $X_{n+1}$ according to $R(X_n, \cdot)$.
4. Set $n = n + 1$ and go to Step 2.
This algorithm generates the sequence \((X_n, T_n): n \geq 0\) of states and holding times. For an arbitrary time \(t\) we can then compute \(X(t) = X_{N(t)}\), where \(N(t) = \min\{n: T_0 + T_1 + \ldots + T_n > t\}\) is the number of transitions in \((0, t]\). We can then use our usual techniques to estimate quantities of the form \(E[f(X(t))]\) or even

\[
\alpha = E\left[\frac{1}{t} \int_0^t f(X(u)) \, du\right] = E\left[\frac{1}{t} \sum_{n=0}^{N(t)-1} f(X_n)T_n + f(X_{N(t)})(t - U_{N(t)})\right],
\]

where \(U_n = \sum_{i=0}^{n-1} T_i\) is the time of the \(n\)th state transition:

\[
\begin{array}{c|c|c|c}
X_2 & N(t) = 2 & \quad & \\
X_3 & N(t) = 0 & \quad & \\
X_0 & N(t) = 1 & \quad & \\
X_1 & T_0 & T_1 & T_2 & T_3 & t & U_4
\end{array}
\]

E.g., if \(f(x) = x\) for the M/M/1 queue, then \(\alpha\) is the average queue length during \([0, t]\) and \(E[f(X(t))]\) is the expected queue length at time \(t\).

To do all of this, we need a way of generating exponentially-distributed random numbers.

**Generation of Exponential Random Variables:**

Suppose that we have a uniform random variable \(U\), and set \(V = \frac{-\ln U}{\lambda}\). We claim that \(V \overset{D}{=} \exp(\lambda)\).

Proof:

\[
P(V > x) = P\left(\frac{-\ln U}{\lambda} > x\right) = P(\ln U < -\lambda x) = P(U < e^{-\lambda x}) = e^{-\lambda x}
\]

This is an instance of a more general method for generating non-uniform RV’s from uniform(0,1) RV’s called the **inversion method**.

**The Inversion Method:**

The idea is to generate a random variable \(V\) having continuous increasing distribution function \(F(x) = P(V \leq x)\) by setting

\[
V = F^{-1}(U),
\]
where $F^{-1}$ is the inverse of $F$ and $U$ is uniform(0,1). Since $F$ is non-decreasing, we have

$$P\{V \leq x\} = P\{ F^{-1}(U) \leq x\} = P\{ F(F^{-1}(U)) \leq F(x)\} = P\{ U \leq F(x)\} = F(x)$$

For example, in the case of an $\exp(\lambda)$ random variable, we have $F(x) = 1 - e^{-\lambda x}$, so that

$$F^{-1}(u) = -\frac{\ln(1 - u)}{\lambda},$$

and we can set $V = -\frac{\ln(1 - U')}{\lambda}$, where $U' \overset{d}{=} \text{uniform}(0,1)$. Observing that $U = 1 - U'$ also is a uniform(0,1) RV, we obtain our previous formula.

The method can be extended to any distribution function $F$ (not necessarily continuous) if we define

$$F^{-1}(u) = \min\{x: F(x) \geq u\}.$$

The proof is almost the same as above, and hinges on the fact that $F^{-1}(u) \leq x$ iff $u \leq F(x)$ by definition of the function $F^{-1}(u)$.

**Exercise:** Show that the inversion method, when applied to generate a discrete random variable, coincides with the naive method given earlier.

### 7. Semi-Markov Processes

A semi-Markov process $(X(t) : t \geq 0)$ with discrete state space $S$ is similar to a CTMC in that the sequence $(X_n : n \geq 0)$ of states visited by the process is a DTMC with transition matrix, say, $R$. The holding time in state $x \in S$, however, is distributed according to an arbitrary distribution function $F(\cdot; x)$ that can depend on $x$. (This definition is a little narrower than some, in which the distribution of the holding time in state $x$ can also depend on $x'$, the next state hit by the process.)

**Example (Renewal Counting Process):**

When

- $S = \{0, 1, 2, \ldots\}$
- $R(x, x+1) = 1$ for all $x \in S$
• \( F(\cdot; x) \equiv G(\cdot) \) for some function \( G(\cdot) \)

then \((X(t): t \geq 0)\) coincides with a renewal counting process.

Simulation of a semi-Markov process:

The algorithm for simulation of a CTMC applies almost unchanged, except that the holding time \( T_n \) is generated according to \( F(\cdot; X_n) \) rather than \( \exp(q(X_n)) \).

8. The Next Step

Although the stochastic processes discussed above can be useful for modeling relatively simple discrete-event stochastic systems, many (perhaps most) real-world simulations give rise to an underlying stochastic process that cannot be represented as a Markov or semi-Markov process. We need to look at a collection of more complex processes that can capture the corresponding complexity of real systems. A very useful stochastic process in this connection is the generalized semi-Markov process (GSMP), which we will discuss next.