Practice Exam: Advanced Simulation for Finance, Business and Economics

Bernd Heidergott & Ad Ridder

May 22, 2020

Exercise 1 [40 Credits]

Consider a stochastic process $\{X(t), t \geq 0\}$ given by its differential equation

$$dX(t) = rX(t)dt + \sigma\sqrt{X(t)}\,dW(t), \tag{1}$$

where $r, b, \sigma > 0$ are positive parameters, and where $\{W(t), t \geq 0\}$ is the standard Wiener process. Let $T > 0$ be some finite horizon, and suppose that we are interested in the quantity

$$V = \mathbb{E}\Big[e^{-rT}\max\big(X(T) - Y, 0\big)\Big].$$

for some random variable Y independent from everything else. Assume that Y has cumulative distribution funtion $F(y), y \in \mathbb{R}$.

(a). [**10 Credits**] Write down the Euler scheme of the stochastic differential equation (1) using constant small time increments $h > 0$. Then, give the algorithm of simulating a single (approximated) sample path $\{X_t, 0 \leq t \leq T\}$ using your Euler scheme. Given this sample path, how do you compute $e^{-rT}\max\big(X(T) - Y, 0\big)$?

(b). [**10 Credits**] Give the expression of the sample average estimator of V using sample size n. How do you estimate the standard error of this estimator?

(c). [**10 Credits**] Now write down the complete Monte Carlo simulation algorithm for estimating V, including reporting standard error and 95% confidence interval.

(d). [**5 Credits**] Which properties do you have to check before you can "trust" the confidence interval obtained in (c)?

(e). [**5 Credits**] How would you verify your computer program (i.e, check that your code is correct)?

Abbreviated solution.

(a). Replacing dt by h, we arrive at

$$X(t+h) - X(t) = rX(t)h + \sigma\sqrt{hX(t)}\, Z$$

with Z standard normal. Hence, given $X(t)$ and an independent sample of Z we arrive at

$$X(t+h) = (r+1)X(t)h + \sigma\sqrt{hX(t)}\, Z.$$

(b). DIY

(c). DIY

(d). Make a histogram of your estimator and check for asymptotic normality.

(e). Run your program for instances of the model you know the answer of. Also print a trace and make a step by step check of the program.

Exercise 2 [20 Credits]

(a). [**10 Credits**] Consider the adjacency matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Consider the graph associated with the above adjacency matrix. Give the maximally strongly connected sub-graphs of this graph.

(b). [**10 Credits**] Given is a strongly connected graph with N nodes, where each node has a self-loop. Consider a multi-agent system on this graph. Each agent updates her beliefs according to a Poisson-λ-clock, with $\lambda = 1$. When agent i updates her belief at time t, she follows the minimal belief of her neighbors, i.e., her new belief is

$$c_i(t^+) = \min\{c_j(t^-) : (i,j) \in \mathcal{V}\}.$$

What is the the limiting belief vector of the agents?

Solution (a) The graph has two maximally strongly connected sub-graphs: the cyle (1,2), (2,1), and the cycle (5,4),(4,3),(3,5)

(b) As each agent has a self-loop, she will only minimize her belief. As the network is strongly connected, an agent will eventually reach the minimal value of c as belief. The limiting belief of each agent is $\min\{c_j : 1 \leq j \leq N\}$.

Exercise 3 [40 Credits]

Starcity is a small amusement park with three rides (attractions): 1 = Froghopper, 2 = Startower, and 3 = Rollercoaster. On a typical day during high season, people arrive at the park in groups of size k with probability a_k for $k = 2, 3, \ldots, 6$ ($a_k > 0$, $\sum_{k=2}^{6} a_k = 1$). The groups arrive according to a Poisson process with parameter λ. From each group one person takes care of paying the entrance fee at the ticket office which consists of two windows. These persons queue up in two seperate lines (one for each window), where a newly arriving person chooses the shortest queue (and arbitrary one when both queues have the same length). Handling the payments and getting some information about the park take together a random time X_0 with distribution function $F_0(\cdot)$ and expectation $1/\mu$. After entering the park each member of the group acts individually, independent from what the others of his group are going to do. Each person will visit only one attraction which he chooses according to the probabilities p_1, p_2, p_3 ($p_j > 0$, $\sum_{j=1}^{3} p_j = 1$). Then he walks to his chosen attraction which takes a random time X_j with distribution function $F_j(\cdot)$ (in case of the j-th attraction, where $j = 1, 2, 3$). At the scene of the attraction he joins the queue (when there is one) and waits for his ride. Attraction j ($j = 1, 2, 3$) operates as follows:

- it can accomodate at most c_j people per ride;
- a ride takes a random time Y_j with distribution function $G_j(\cdot)$;
- a new ride starts whenever there are at least m_j people waiting for a ride, where $m_j < c_j$;
- for simplicity assume that loading and unloading take no time.

Finally, walking back from the attraction to the exit takes a random time Z_j with distribution function $H_j(\cdot)$ ($j = 1, 2, 3$). All the random variables involved in the model are independent of each other.

The first visitors to the park arrive already from 7.30 hrs onwards, although the park opens its ticket office and its entrance gates at 8.00 hrs. Most probably, by this time there will be already two lines in front of the ticket office. Closing time is at 19.00 hrs, meaning that no new arrivals are admitted from then on. The people in the park will always get their ride. Possibly, the last ride at attraction j is with less than m_j people ($j = 1, 2, 3$).

The management of Starcity receives many complaints of long waiting lines in front of the attractions, and foresees that the popularity of the park will decline. In order to improve the logistics in the park, the management sets up a simulation study from which such performance measures are estimated. Having the simulation model, it is then possible to consider several solutions of making the waiting lines smaller.

The performance measures include

- w_0: expected average waiting time at the ticket office per person (per day);
- ℓ_j: expected average waiting line length at attraction j per unit time (per day), $j = 1, 2, 3$;
- r_j: expected fraction of time (per day) that a ride at attraction j is going on, $j = 1, 2, 3$.

(a). [**10 Credits**] In a mathematical model of Starcity, which random variables do you need to define to get an estimator of the performance measure w_0? How do you get an estimate and a confidence interval of the estimate?

(b). [**10 Credits**] Suppose that you construct a discrete-event simulation model (DES) for estimating the performance measures. Specify the event list and the system state of your DES that enable you to observe the system dynamically in time, and define the counter (or statistical) variables to be used for calculating daily observations of the variables whose expectations are the performance measures.

(c). [**10 Credits**] The (pseudo-)code of simulating a single day looks like

```
t = 0;
initialize;

while t<30
    [t,event] = schedule_next_event;
    if event = arrival_at_park : ???;
end;

while t<690
    [t,event] = schedule_next_event;
    if event = arrival_at_park : ???;
    if event = ticket_window_departure : ???;
    if event = ???;
    ..............
    if event = ???;
end;

while park_not_empty
    [t,event] = schedule_next_event;
    ..............
end;

collect_statistics;
```

Time is measured in minutes and t represents the current simulation clock time. Complete this code with the other events that you defined in (b). Give the details of how you need to update state, event list and counter variables in at least two of the procedures in this pseudo-code listed after `if event =` .

(d). [**5 Credits**] How would you verify your program?

(e). [**5 Credits**] Can you suggest solutions to the management that will (probably) make the waiting lines smaller?

Solution.

(a). For the expected average waiting time at the ticket office per person per day, we define the random variables

$$N \doteq \text{the number of groups arriving during a day;}$$
$$W_k \doteq \text{the time spent in the queue at the ticket office by the person of group } k \text{ who is in charge for the payments, } k = 1, \ldots, N.$$

The average waiting time a random day is

$$Y \doteq \frac{1}{N}\sum_{k=1}^{N} W_k.$$

The performance measure of interest is

$$w_0 \doteq \mathbb{E}[Y] = \mathbb{E}\Big[\frac{1}{N}\sum_{k=1}^{N} W_k\Big].$$

We obtain an estimate $\widehat{w_0}$ by simulating independently n days, say $i = 1, \ldots, n$. The i-th day has $N^{(i)}$ groups arriving, from which the paying person of group k spends $W_k^{(i)}$ in the queue, $k = 1, \ldots, N^{(i)}$. The average waiting time spent on the i-th day is

$$Y_i \doteq \frac{1}{N^{(i)}}\sum_{k=1}^{N^{(i)}} W_k^{(i)}, \quad i = 1, \ldots, n.$$

These averages $Y_1, \ldots, Y_n$ are i.i.d. as Y. Thus, an unbiased estimator of w_0 is

$$\overline{Y}_n \doteq \frac{1}{n}\sum_{i=1}^{n} Y_i,$$

and any observation of $\overline{Y}_n$ from a simulation experiment may be considered to be an estimate $\widehat{w_0}$. For constructing a confidence interval of an estimate, consider the standard error of estimator $\overline{Y}_n$:

$$\texttt{SE} \doteq \sqrt{\mathbb{V}ar(\overline{Y}_n)} = \sqrt{\mathbb{V}ar(Y)/n}.$$

An unbiased estimator of $\mathbb{V}ar(Y)$ is the sample variance:

$$S^2 \doteq \frac{1}{n-1}\sum_{i=1}^{n}\big(Y_i - \overline{Y}_n\big)^2.$$

Our simulation produces a realisation s^2 of S^2. Hence, $\widehat{\texttt{SE}} \doteq \sqrt{s^2/n}$ is an estimate of the standard error. A $100(1-\alpha)\%$ confidence interval is

$$\big(\widehat{w_0} - t_{n-1,1-\alpha/2}\,\widehat{\texttt{SE}}, \widehat{w_0} + t_{n-1,1-\alpha/2}\,\widehat{\texttt{SE}}\big),$$

where $t_{n-1,p}$ is the p-th quantile of the Student t-distribution with $n-1$ degree of freedom.

(b).

- System state (at time t) is a data sructure with information on
(i) the number of persons queueing at the ticket office in line 1, the arrival times of the group they belong to, and the group size;
(ii) idem line 2;
(iii) the status of the two ticket windows (busy or idle);
(iv) the number of persons walking from the entrance to attraction 1, to attraction 2, and to attraction 3;
(v) the number of persons queueing at attraction 1, and their arrival times at the attraction;
(vi) idem attraction 2, and attraction 3;
(vii) the status of the three attractions (riding or waiting);
(viii) when an attraction rides, its occupation;
(ix) the number of persons walking from attraction 1 to the exit, from attraction 2, and from attraction 3;

- Events deal with occurences that change the system state:
(i) a group arriving at the park;
(ii) a person completing the payments;
(iii) a person arriving at attraction 1;
(iv) a person arriving at attraction 2;
(v) a person arriving at attraction 3;
(vi) end of a ride of attraction 1;
(vii) end of a ride of attraction 2;
(viii) end of a ride of attraction 3;
(ix) a person leaving the park from attraction 1;
(x) a person leaving the park from attraction 2;
(xi) a person leaving the park from attraction 3.

- Event list at time t is a data structure consisting of clock times that show for all events when they are scheduled to occur. In order of the events, and suppose that the current time is t =9:20:37 hr (20 minutes and 37 seconds after 9 o'clock).
(i) There is a single clock for event (i), for instance it says 9:22:18 hr. This means, the first group to arrive at the park after time t =9:20:37 is scheduled at 9:22:18.
(ii) Any busy server at the ticket window has a clock ticking until payments are completed. For instance, both are busy, and the clocks could be 9:23:29, 9:24:06.
(iii) Any person walking from the ticket office to attraction 1 at time t (it is known how many because of state component(iv)) has a clock that says when he will arrive at attraction 1. All the clock times are keeping track

of (chronologically). For instance, suppose there are 4 persons, then the clocks might be 9:24:19, 9:28:38, 9:29:02, 9:33:51 hr.
(iv) idem for person heading to attraction 2;
(v) idem for person heading to attraction 3;
(vi) There is a single clock for event (vi), for instance it says ∞. This means that, actually, attraction 1 is waiting for enough persons for the next ride;
(vii) idem for the ride of attraction 2; the clock time might be 9:30:49 meaning that the ride is in progress and will finish at 9:30:49;
(viii) idem for the ride of attraction 3;
(ix) Any person walking from attraction 1 to the exit at time t (it is known how many because of state component(ix)) has a clock that says when he will arrive at the exit gate. All the clock times are keeping track of (chronologically). For instance, suppose there are 2 persons, then the clocks might be 9:25:38, 9:30:27 hr.
(x) idem from attraction 2;
(xi) idem from attraction 3.

- Counter variables are statistics that are being recorded during a simulation run; their values at the end of the run are used to compute performance measures.
  (i) Waiting time at the ticket office: you need a variable, say W_1, that accumulates the waiting times of all the persons that queued in line 1 (in front of ticket window 1); similarly a variable W_2 accumulating the waiting times in line 2; and a variable N that keeps track of the number of arriving groups at the park that were permitted to enter. At the end of the run, $(W_1 + W_2)/N$ is the average waiting time at the ticket office.
  (iii) Average queue length at attraction 1: consider all event times of a simulation run, chronologically, $0 = T_0 < T_1 < \cdots < T_{N_e}$, with N_e the number of events that occurs during a simulation day. Let M_j be the number of persons queueing at time T_j+ (just after the event time). Then, at the end of the day the average queue length is computed by

$$\frac{1}{T_{N_e}} \sum_{j=1}^{N_e} \big(T_j - T_{j-1}\big) M_{j-1}.$$

  (iv) Similarly for the other attractions.
  (v) fraction of time that ride 1 is in progress: now multiply the inter-event times $T_j - T_{j-1}$ with the binary variable B_{j-1} which is 1 when the ride is

in progress at event time T_{j-1}, and otherwise 0. Then

$$\frac{1}{T_{N_e}}\sum_{j=1}^{N_e}\big(T_j - T_{j-1}\big)B_{j-1},$$

is the fraction of time that attraction 1 is in progress. (vi) Similarly for the other attractions.

(c). The first ??? deals with the event of an arrival at the park between 7:30 and 8:00 hr, when the park is not yet open. There is only queueing. Note that t is the event time. The program would handle the following actions.
(i) Simulate the group size from the (a_k) probabilities, say K.
(ii) Check the queue lengths N_1, N_2 in front of the two ticket windows, and choose the shortest, say N_1. Then increase N_1 by 1.
(iii) Add the person at the end of the queue with information on arrival time t, and group size K.
(iv) Simulate from the exponential distribution the time until the next arrival, say A. Then the event list becomes just the single clock $t + A$.

The second ??? deals with the event of an arrival at the park after 8:00 hr. Note that t is the event time. Let t_{previous} be the previous event time. The actions are.
(i) Update all the counter variables for performance measures ℓ_j (queue length at attraction j, $j = 1, 2, 3$), and r_j (fraction of time ride j is in progress, $j = 1, 2, 3$). See (b).
(ii) Simulate the group size from the (a_k) probabilities, say K.
(iii) Check whether a ticket window is free, say window 1 is free. This server becomes busy. Simulate a handling time X_0 from distribution function F_0, and add time $t + X_0$ on the event list. (iv) If both ticket windows are busy, then check the queue lengths N_1, N_2 in front of the two ticket windows, and choose the shortest, say N_1. Then increase N_1 by 1, and add the person at the end of the queue with information on arrival time t, and group size K.
(iv) Simulate from the exponential distribution the time until the next arrival, say A. Add time $t + A$ on the event list.

(d). Verification is the process of checking that the computer program is a correct implementation of the mathematical model. Several techniques could be applied.

- Check whether the random variables $X_0, X_1, X_2, X_3, Y_1, Y_2, Y_3, Z_1, Z_2, Z_3$ are correctly simulated.
- Print a trace.

- Compute the fraction of persons that enter attraction 1 in your simulations. It should be (close to) p_1. Similarly for the fractions entering the other attractions.
- Set group size $K = 1$ (fixed, deterministic), and give the handling time X_0 of the ticket office an exponential distribution. Then the ticket office is a Poisson-exponential join-the-shortest-queue model for which you can determine exact (numerically) expressions of queue lengths.
- Set capacity of attraction 1 to $c_1 = 1$. Then attraction 1 becomes a single server queue.