Simulatie (FEB22013X)

Tutorial 5: A queueing model

In this tutorial you are asked to develop a simulation model for a queueing system. This model shows many similarities to the $M/M/c$ queuing model and to the simulation model created in Tutorial 2. However, whereas we provided a lot of hints in Tutorial 2, we now expect you to build the model by yourself. Make sure that your simulation model is generally applicable for different parameter values.

The answers of your group must be handed in as one PDF file of at most 4 pages. The file can be handed in via Blackboard (in the dropbox for Assignment 5). The deadline is Tuesday April 4, 2017 at 23:59h. Formulate your answers in a concise way and put the relevant Matlab-code in the same document. If necessary, you can add an appendix in the same document with the other Matlab-code. Mention your names and student numbers on the first page. The PDF file must have the following name:
`studentnr1_studentnr2_assignment5.pdf`

Opgave 1 The city council of Gorinchem in The Netherlands has started an analysis of the Grote Merwedesluis ('Great Merwede Lock/Sluice') in Gorinchem. Captains of both the shipping industry and pleasure cruising have been complaining about the long waiting times for this lock. The lock has a capacity of ten ships and is currently operated only if there are at least six ships inside. By operating the lock only when there are sufficient ships, the number of operations (i.e., the number of times the water level has to be changed) can be reduced. Unfortunately, this policy also increases the waiting times. The city council is now considering to operate the lock already when there are fewer ships in order to reduce the waiting times. You have been asked to construct a simulation model for the lock and to optimise the decision. Figuur 1 presents an illustration of how a lock works.

Ships use the lock to go from the river Merwede (south) to the Merwede canal (north) and vice versa. Assume that ships arrive from the south with rate $\lambda_1 = 27$ per day. From the north ships arrive with rate $\lambda_2 = 20$ per day. The Merwede Lock has a capacity of maximal 10 ships. The duration of a lock operation (i.e.,

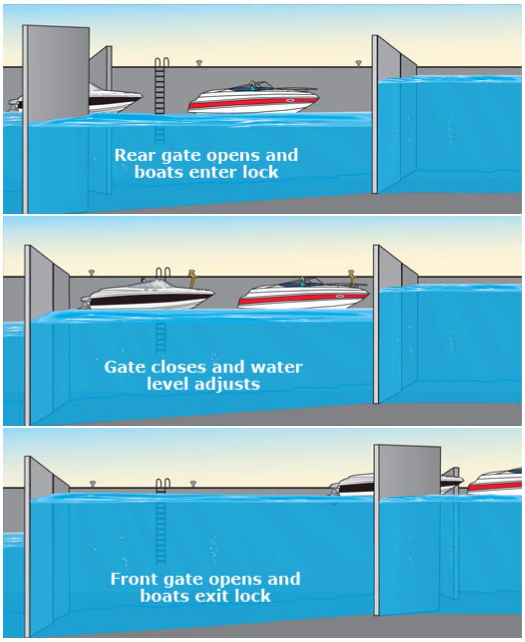

Figure 1: Illustration of a lock operation.

changing the water level), including the entering and departing of the ships, can be characterised with the following pdf, with x being the duration in hours:

$$f(x) = 12x^2(1-x), \quad x \in [0, 1]$$

The city council has decided to minimise the yearly cost and use that as a measure to determine a good strategy for the lock. Operating the lock costs on average 200 € each time. There are also costs involved with waiting time for the ships. It is expected that these costs are 15 € per ship per hour. In the simulation model we will determine the yearly costs. You can assume that the lock operates 365 days a year and 24 hours a day. Replicate your simulation 10 times to get a reliable estimate of the yearly costs. We assume that there are no ships waiting at the beginning of the year and that the lock is opened on the south. The following variables and events are used in the simulation model:

State variables:

x	the state of the lock (0 = idle, 1 = busy)
y	the side at which the lock is open or at which is was open the last time (1 = south, 2 = north)
Q_1	the number of waiting ships on the south side
Q_2	the number of waiting ships on the north side.

Counter variables:

O	the number of lock operations
S	the number of ships that went through the lock
C	the total time used for lock operations
W	the total waiting time

Events:

- arrival of a ship at side 1
- arrival of a ship at side 2
- completion of a lock operation (the doors of the lock are open again)

a. Implement this simulation model in Matlab.

Hints:

- Take the simulation model for the $M/M/c$ queuing system you created in Tutorial 2 as a starting point. Use the general structure of a DES model to structure your thoughts.
- Express the time t in days.
- Do not try to write the entire program at once. Instead, build the program step by step, and test adaptations of your program frequently. For instance, try to write the library routines first, which generate inter-arrival times for ships and the duration of a lock operation. These routines can be tested by making a histogram of 1.000 draws.
- Transform the function `service_realisation(mu)` into a function that generates draws from the distribution of the duration of the lock operations, which is introduced above. Use one of the methods explained in Lectures 8 - 10. Describe the function/algorithm you use to generate draws from the distribution of the duration of the lock operations and provide the Matlab code corresponding to this function/algorithm in your PDF document.
- Transform the event routine `procedure_arrival()` into the event routine `procedure_ship_arrival()`, which handles the arrival of a ship and in which you determine whether or not a lock operation should be started. Make sure that this procedure can handle ships arriving from both sides and take as input of the procedure the side from which the ship is arriving. Transform the event routine `procedure_departure()` into the event routine `procedure_lock_completion()`, which handles the completion of a lock operation. Also check in this procedure whether or not a new lock operation should be started directly. It is useful

to start with making flowcharts for the routines `procedure_ship_arrival()` and `procedure_lock_completion()`.

- Determine which counter variables you need, and use these consistently in all components of your program.
- Use the debugging mode to check your simulation model. Analyse for example if the system state and counter variables are updated correctly by debugging the program on the following line:
  `[t,i] = schedule_next_event(eventlist);`. This helps you to test whether your program functions appropriately and to identify errors and their causes.

b. Analyse the current strategy of the lock, when the water level is only adjusted if there are at least six ships in de lock. What are the yearly costs? What is the percentage of time that the lock is busy with operations? What is the average waiting time per ship?

c. As said the city council considers operating the lock with fewer ships to reduce the waiting time. Define the parameters k_1 and k_2 as follows: when the lock is open on the south side, a lock operation is started when there are at least k_1 ships. Similarly, a lock operation is started when there are at least k_2 ships from the north side. Determine the optimal values $k_1 \in \{1, 2, ..., 8\}$ and $k_2 \in \{1, 2, ..., 8\}$ that minimise the yearly costs.

So far, we implicitly assumed that the expected number of ships arriving at the lock is equal in every part of the year. However, in the summer it is a lot busier. The arrival process of ships over the year can be characterised with the following functions:

$$\lambda_1(t) = 27 + 2\sin\left(\frac{t}{60} + 5\right) \quad \text{and} \quad \lambda_2(t) = 20 + 5\sin\left(\frac{t}{60} + 5\right),$$

with t measured in days $t \in \{1, 2, ..., 365\}$.

d. Model the arrivals of ships as a non-homogeneous Poisson-process with arrival rates $\lambda_1(t)$ and $\lambda_2(t)$. Does this adaptation affect the optimal values for k_1 and k_2?

Hint: make use of the algorithm in the slides (or in Section 5.5 of the book Simulation, Fourth Edition by Sheldon M. Ross).

As a consequence of the long waiting times, many ships are currently waiting in the area around the lock. After observing this, the city council has decided to allow at most 10 ships waiting at each side of the lock. Ships that arrive when there are already 10 ships waiting have to choose another route and are no longer considered in the system.

e. Evaluate the effect of this measure on the average waiting time per ship and the yearly costs of the original strategy of the lock $k_1 = k_2 = 6$.

f. What is the optimal strategy $(k_1, k_2) \in \{1, ..., 8\}^2$ after the introduction of the measure? What percentage of arriving ships gets rejected in the optimal strategy?