Simulation 11 - Excercises

Half a point bonus for the final exam (graded 0-30) for each excercise. The problems are discussed/solutions presented in three sessions that will be announced separately.

Problem 1 *Consider event based approach/model for the wash machine (lecture 2). Formulate two different ways to determine the utilization rate of the machine.*

Problem 2 *What changes are needed to the event based implementation of the wash machine example if we add another machine that serves the same queue (are new events needed, how the events must be modified).*

Problem 3 *Modify the basic example (Examples/Basic) from JavaSim - package to conform to the wash machine example of the lectures (i.e. define maximum capacity for the queue and ensure that it is not violated).*

Problem 4 *What changes are needed to the previous implementation if we want to model two machines serving the same queue.*

Problem 5 *Sketch an event based model for a situation (cf. harbour case) where clients arrive to two service points, each having its own queue and after being served they are routed forward to a third service point (after a randomly varying journey time). The third service point has its own queue. How does it show in the model if we want to follow the clients individually or if we are interested only in utilization of the services.*

Problem 6 *Consider the previous case as a process/object based model (in practice as a modification or extension of the JavaSim -example). Which processes have to be modified. Are new processes needed.*

Problem 7 *Formulate a process based model for a situation where a queue is served by two (identical) services that require operator assistance for the start up period of each client. Only one operator is available (and can not assist two machines/clients simultaneously).*

Problem 8 *Test the original Lehmer generator ($m = 10^8 + 1$, $a = 23$) for randomness (in particular for serial correlation). A convenient way for this is χ^2 test. The procedure is as follows: construct k-tuples of consecutive random numbers using the generator (pairs, triplets). Evaluate the maximum of each k-tuple. Divide the interval $(0, 1)$ to, say 10, classes and determine the* *expected probability for the maxima to belong to the classes (in practice you can set the class boundaries to* $(i/10)^{1/k}$ *which will lead to equal probability* $(1/10)$ *for each class. Count the number of maxima* N_i *belonging to each class* i.

The χ^2 *statistics can be computed as*

$$\sum_{i=1}^{n}(N_i - E_i)^2/(E_i)$$

where (for equally probable classes $E_i = N/n$ *and* N *is the total amount of samples (k-tuples) generated.*

Run 5 replications of the test, each with at least 100 samples and report the maximal and minimal values of the test variable (that should stay between 3.3 and 16.9 if the random stream is really random).

Problem 9 *Consider elimination method for producing random numbers. How would you produce normally distributed random numbers* ($N(0,1)$) *if you can use uniformly and exponentially distributed streams. Try to estimate how many pairs of random numbers you need for each* $N(0,1)$ *output value.*

Problem 10 *Is it true that* $\pi = 3$. *Simulate the Buffon needle experiment of the lectures (using appropriate random numbers to replace throwing of the needle). Report the experiment (generators used, number of trials) and the results (estimate for* π *and its confidence interval with some significance level).*

Problem 11 *Modify the Basic example of the JavaSim package so that it will produce, instead of one result for one run, the results using renewal technique from each independent subinterval. As renewal points you can use the situations where the system becomes empty. (The Basic example is initialized to such a situation, sceduling a future arrival to an empty system).*

Consider the wash machine example from the lectures (one service serving one queue). Assume that the queue will hold at most two waiting clients (in addition to the one being served). New clients arrive to the system on arerage every 8 minutes (interarrival times exponentally distributed). We consider two different service options with service times either $Unif(4,8)$ or $Unif(6,10)$ distributed.

Simulate both variants (for example ten independent runs of 1000 minutes). Be careful to not use same random numbers to generate both service and interarrival times. Pay also attention to the possible initial transient at start up.

Problem 12 *Evaluate the simulated utilization rate for both variants and its sample variance/standard deviation. Determine a suitable confidence interval (corresponding to 95% significance level, for example).*

Problem 13 *Theoretically the utilization rate of the faster service is* $3/4$*:th of that of the slower one when same amount of clients are served. Can you infer from the simulation that results* $U_{fast} > 0.75 * U_{slow}$ *if you consider the simulations as fully independent.*

Problem 14 *Can you infer that* $U_{fast} > 0.75 * U_{slow}$ *if you compare the simulations as pairs (using same simulation seed values for both variants).*

Problem 15 *Monitor the amount of lost clients. How can you derive an estimate for the utilization rates and their difference based on lost clients. Is this estimate more accurate than the estimate obtained directly from the service times.*

Problem 16 *How would you estimate the utilization rate using the total amount of clients generated as a concommittant control variable (i.e. the total of lost clients and clients entering the queue). What is the expectation for this and how does the difference of forecasted and observed utilization rate behave (does it have smaller variation/confidence interval than the actual utilization rate).*

Problem 17 *How does the utilization rate depend on the interarrival times. Simulate the faster variant with interarrival times of 7 and 9 minutes. Formulate a simple regression model for utilization rate as a function of interarrival time.*

Problem 18 *Is the above regression model valid. How does it predict the results of the original situation (interarrival times of 8 minutes). Could/should something be made differently.*

In case you can not manage to modify/compile/run the JavaSim cases, samples of results will be made available on the course web pages to enable the analysis.