ISyE 4045 – Spring 20223

Homework #6 — Due Monday, 4/24

Follow the instructions on the syllabus to submit a single archive containing all codes and a single PDF document containing your discussion and findings. Name the archive as `LastName-FirstName.zip`.

1. Problem 2.26 from the text of Law (2015): Two-piece suits are processed by a dry cleaner, illustrated in Figure 1, as follows. The shop currently employs five servers (labeled as 1–5). Each of servers 1, 2, 3 and 5 have a single FIFO queue, while server 4 serves two FIFO queues. Suits arrive with exponential interarrival times having mean 10 minutes, and are all initially served by server 1, perhaps after a wait in queue. Upon completion of service at server 1, one piece of the suit (the jacket) goes to server 2, and the other part (the pants) to server 3. During service at server 2, the jacket has a probability of 0.05 of being damaged, and while at server 3 the probability of a pair of pants being damaged is 0.10. Upon leaving server 2, the jackets go into a queue for server 4; upon leaving server 3, the pants go into a different queue for server 4. Server 4 matches and reassembles suit parts, initiating this when he is idle and two parts from the *same* suit are available. If both parts of the reassembled suit are undamaged, the suit is returned to the customer. If either (or both) of the parts is (are) damaged, the suit goes to customer relations (server 5). Assume that all travel times are negligible (0), and the service times are exponential with the following means (in minutes) and use the indicated stream assignments:

Server number	Mean service time (minutes)	Stream
1	6	1
2	4	2
3	5	3
4	5 (undamaged)	4
4	8 (damaged)	4
5	12	5

   These streams may help in the analysis later.

   Build a Simio model for this facility. Clearly, you should use a Separator object to model server 1 and a Combiner object to model server 4. Make the interarrival time and the capacities of the servers referenced properties; this will help you creating scenarios with server capacities and mean interarrival times as experimental Controls.

   Hints: Search for SimBits using the key words *Separator Make Copies* and *Combine Matching Members*. The list below contains a few additional hints:

   - To use random numbers from a stream other than the default (0), insert the stream number as the second argument of the expression. For instance, Random.Exponential(6,10) generates realizations from the exponential distribution with mean 6 using uniform random numbers from stream 10.

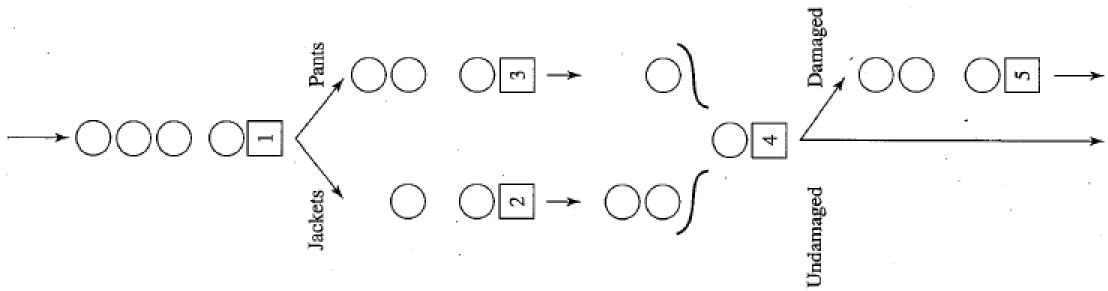

Figure 1: Dry-cleaning operation

- At the Separator object (server 1), you can make copies of the original entities (suit). The original entity (parent) leaves the "top" link and, to follow Figure 1, represents the pants. The copy leaves from the bottom link, hence it represents the jacket. Keep in mind that the copies get new entity ID; this ID is stored in the entity state variable (attribute) Entity.ID.
- Before processing at the Combiner object (server 4), you must determine the status of the suit to be assembled; this will be used to obtain the mean time for the exponential distribution (5 or 8 minutes). You can do this using either an add-on process with Decide steps or state assignment(s) based on a Boolean entity state representing the status of the suit. Feel free to use your creativity.
- To make sure that server 4 matches the appropriate jackets and pants, created an entity state variable that stores the entity ID of the original entity. You can assess the matching of the portions of the original suit if you attach an entity label to each entity instance.

Create an experiment with 50 replications, each over a 12-hour window starting with an empty system. Make sure to include the stream number in the assignments for the experimental Controls. Create (S)MORE plots for the following metrics:

(a) Average time in system for all suits that leave the system. I suggest creating a Tally Statistic whose values are recorded when suits are about to exit (e.g., when they enter input nodes at Sinks).
(b) Average length of each input queue at servers 1–5. Notice that server 4 has two queues, one for jackets and the other for pants.
(c) Total number of processed suits during the 12-hour window. You can collect totals at the input nodes of Sinks.

2. Consider the dry-cleaning model from Problem 1. The scope of this problem is the estimation of metrics of the marginal distribution of the time-in-system for "good" suits. Click on the Model in the Navigation window and change the run length to 500,000 hours. You can log the Tally observations and copy/paste the data into a text file *or* you can create a File element and use an add-on process at the respective sink with a Write step that records the data in a text file (the second is better). In the latter case, you may record fewer than 13 decimal places using the

Math.Round(Expression,x) to round off the observation to the xth decimal place; this will reduce the file size (5–6 decimal places may be enough).

(a) Make the long run (it took me nearly 20 minutes), select the Pivot Grid within the Results tab, and check if Simio was able to compute an approximate 95% CI for the expected value of the time in system for a typical "good" suit in steady state. If it succeeded, report the CI.

(b) Quit Simio and execute the Sequest app (files sequest_wx6.exe or sequest.jar) on your computer. If you have a Mac, you have only have the second option. In either case, make sure you install the Java JRE.

- First use the SKART method to compute an approximate 95% CI for the expected value of the time in system for an arbitrary good suit in steady state. Use a CI *relative* precision requirement of 0.05. If the method succeeds, report the respective CI and a snipping of the Execution Status area of the application window. If the method fails, include a capture of the final screen with the notification that more data are needed.
- Then use the Sequest method to compute approximate 95% CIs for the median, 75th, 90th, 95th, and 99th percentiles of the time in system of a typical good suit in steady state. Use a CI *relative* precision requirement of 0.1. If the method succeeds, report the respective CI and a snipping of the Execution Status area of the application window. If the method fails, include a capture of the final screen with the notification that more data are needed.

3. Reset the run length to 12 hours. Now suppose that the mean interarrival time drops to 6 minutes. It should be clear that the system will have a hard time sustaining this increase in the arrival rate. Suppose that you can increase the capacity of one "station" among 1–5. Where should that be? Let's use the expected value of the time in system in part 1(a) above as the metric for comparisons.

(a) Make 20 independent replications for each assignment, and then use Tukey's method (Subset Selection Analysis add-in) with the default confidence level. Does the tool pick a best allocation, with regard to the expected value of the average time in system, for the extra unit of capacity? Make a precise statistical statement regarding the optimality of the solution(s) selected by the add-in. Include a screen shot in your write-up.

(b) Now conduct an experiment using the Kim and Nelson (2001) (KN add-in in Simio). Change the Required Replications value to 10, set the default number of replications to 10, set the replication limit to 100, and use an indifference zone of $\delta = 2$ minutes for the expected value of the average time in system. What is the outcome of the experiment? Make a precise statistical statement regarding the optimality of the solution(s) selected by the KN add-in. How does the outcome compare with the outcome of the experiment in part 2(a) above? Did the KN experiment use fewer replications to reach a verdict? Include a screen shot in your write-up.