CS 351: Simulation Project

Due: Friday, May 5, 2023 at 11 PM (Central)

Overview

This project involves modeling a system, implementing it in software, and analyzing it using simulation. A description of the system is provided starting on page 3. The questions to be addressed via simulation can be found on page 5.

You may complete the project either individually or in groups of two people; if you intend to complete the project as a group, one member should send an email to the professor by 5 PM on April 21 indicating the members of the group (be sure to carbon copy all group members on the email).

Deliverables for the project include:

1. An implementation of the model in Java using DESMO-J with either an event-oriented perspective or a process-oriented one, submitted as source code in a **single compressed archive** (either `.zip` or `.tgz` format). This may be either an exported Eclipse project or just an archive of the relevant `.java` files (**do not** include the DESMO-J `.jar` file in your archive, as this can cause problems when uploading to Canvas).

2. A brief written report that includes answers to the study questions on page 5 *along with* explanations of how you arrived at those answers (e.g., number of runs, replication results, calculations of confidence intervals, etc.). The report should:

   - clearly identify the group members;
   - adhere to professional writing standards (e.g., written in complete sentences, reasonable formatting, no spelling or grammatical mistakes), and
   - be in `.pdf` format (**do not submit** in other formats such as `.doc`, `.docx`, or `.pages`).

One member of your group should submit the above items to the project dropbox on Canvas by the project due date.

Additionally, while you **do not need** to develop a design model for the system, it may be helpful to do so, particularly **prior** to starting your implementation. (An hour or two of planning can potentially save several hours or more of refactoring and debugging!) I am happy to discuss your design model and offer feedback during office hours. (Just make sure that you start the project early enough to leave time for doing this!)

Additional Requirements

- You *may* use example code on Canvas and/or from the labs as a starting point for your own work. **However**, if you do this, then **you must ensure** that all comments as well as names for variables, methods, and classes are updated appropriately **and** that any extraneous components are removed. Failure to adequately update any example code that you use will **negatively impact your project grade**.

- You **must** use DESMO-J's distributions for generating all random values used in your model. All distributions should be stored as member variables in your model class and instantiated in your model's `init` method. **Do not** instantiate new distributions within any event or process logic. **Do not** use `Math.random` or the `Random` class in Java. (These requirements are important for reproducibility and replication).

- All output measures of interest must be tracked using DESMO-J's statistical trackers (e.g., `Count`, `Tally`, `Accumulate`) and `Queue` structures, and these outputs must be included in the report(s) that gets generated after running the simulation.
- Your program **must** include a mechanism for performing **multiple replications** of the simulation model, using **different** random values in each replication. (One such mechanism for doing this will be introduced in class at least two weeks before the project deadline.)
- Your program should not generate any extraneous print statements or messages to standard out (other than what is produced by DESMO-J's classes already). You may however include additional debugging statements in the debug and trace files produced by DESMO-J.

Grading

At a high level, your project grade will depend on the following:

1. Implementation of current system configuration, including:
   (a) Model components
   (b) Event and/or process logic
   (c) Statistical trackers and outputs
2. Support for multiple replications
   (a) Automated mechanism for performing multiple replications
   (b) Analysis of outputs gathered across multiple replications
3. Answers to study questions involving the **Current Configuration**
4. Support for exploring alternate system configurations:
   (a) Implementation details for alternate configurations
   (b) Testing an alternate configuration should require minimal modifications to the code (e.g., by changing arguments passed into a model constructor, or changing configuration options stored as class constants)
5. Answers to study questions involving **System Comparison**
6. Answers to study questions involving **System Optimization**

In order to earn points in one of these categories, your program needs to be correct or nearly correct across all preceding categories. This means that you should focus your efforts on these categories in order.

System Description

A walk-in medical clinic is open from 8:00 AM until 8:00 PM seven days a week. Interarrival times for patients are exponentially distributed but vary throughout the day according to the following table:

Time Frame	Mean Interarrival Time (minutes)
8 AM – 10 AM	$\lambda^{-1} = 15$
10 AM – 4 PM	$\lambda^{-1} = 6$
4 PM – 8 PM	$\lambda^{-1} = 9$

When a patient arrives, they stay in a waiting room until they are called to see the nurse practitioner. If a patient arrives when there are already k other patients in the waiting room, the arriving patient immediately leaves (i.e., balks) and instead seeks treatment at a nearby emergency care center with probability $k/8$ (for $k = 0, 1, 2, 3, \ldots, 8$).

The nurse practitioner treats one patient at a time according to a "first come, first served" policy. Treatment times are exponentially distributed with a mean of 8 minutes. After treating a patient, the nurse practitioner must decide whether to refer that patient to the medical specialist employed by the clinic to resolve more urgent medical needs. Each patient has a 40% chance of being referred to the specialist. Patients who need care from a specialist cannot wait too long to receive it. If the nurse practitioner wants to refer a patient to the specialist for more urgent treatment, but the patient has already been in the clinic for more than 30 minutes (including time in the waiting room and time being seen by the nurse practitioner), the patient must instead be diverted to the nearby emergency care center to be treated (note that this occurs only if the nurse practitioner wants to refer this patient to the specialist).

The specialist has four examination rooms, each of which can hold one patient. The specialist can treat one patient at a time; if the specialist is currently treating another patient when the nurse practitioner refers a patient to the specialist, the patient waits in one of the other examination rooms until they are seen by the specialist. Because there are only four examination rooms, only three patients can wait to see the specialist at any time (not counting the patient being treated, who uses the fourth examination room). Any patient referred to the specialist while all examination rooms are full is immediately diverted to the nearby emergency care center. The specialist's treatment times are distributed exponentially with a mean of 25 minutes.

The clinic closes at 8:00 PM, after which no new patients arrive. Any patients already in the clinic after it closes will remain in the clinic until their treatment is complete (which will include being seen by the nurse and possibly the specialist, assuming they are not diverted to the emergency care center after finishing treatment by the nurse practitioner). The nurse and specialist will both remain in the clinic until all patients have left.

The clinic's current operating expenses include:

- \$1200 per day for the nurse's salary, plus \$100 for each patient treated by the nurse;
- \$1500 per day for the specialist's salary, plus \$200 for each patient treated by the specialist;
- \$300 per day for each exam room (this includes cleaning, maintenance, and utilities).
- An *additional* \$500 for each patient that gets diverted to the emergency care center for any reason (e.g., a patient treated by the nurse practitioner and then diverted to the emergency care center incurs a cost of \$600, while a patient who leaves the clinic before being seen by the nurse practitioner incurs a cost of \$500).

The clinic manager wishes to understand the expected daily operating costs, and also identify ways in which these costs can be minimized. To achieve this second goal, the clinic can:

- hire additional nurse practitioners at a cost of $1200 per practitioner per day;
- hire additional specialists for a cost of $1500 per specialist per day; and
- add additional exam rooms for a cost of $300 each per day.

However, such hires or examination rooms must be justified by a savings in overall expected costs incurred by the clinic. Note that hiring additional specialists does not increase the number of examination rooms unless new examination rooms are also purchased; for example, if you increase the number of specialists to three, but do not increase the number of examination rooms, then three examination rooms contain specialists, with only one examination room left to contain a patient who is waiting to be seen by a specialist. You may assume that any new hires are indistinguishable from the current nurse practitioner and specialist, from the perspective of the distributions of treatment times.

Study Questions

Current Configuration

For the current configuration of the clinic, provide estimates of the mean and standard deviation of each of the following quantities:

1. the clinic's daily operating costs (this includes the costs for the existing staff and exam rooms!)
2. the number of patients per day that:
   - arrive at the clinic for treatment;
   - balk due to an overcrowded waiting room;
   - get diverted to the ER after being seen by the nurse;
   - get fully treated at the clinic;
3. the average response time of patients that get fully treated at the clinic (i.e., they don't balk or get diverted to the emergency room);
4. the utilization rates for the nurse and specialist, computed across the time period from 8 AM until the nurse and specialist leave the clinic (which is *not necessarily* 8 PM!); and
5. the average number of patients in the waiting room, computed across the time period from 8 AM until the nurse and specialist leave the clinic.

System Comparison

Compare the daily operating costs of the current system with the operating costs of two alternative configurations, one in which an additional nurse practitioner is hired and one in which an additional specialist is hired.

1. Is there a statistically significant decrease in daily costs when adding an additional nurse?
2. Is there a statistically significant decrease in daily costs when adding an additional specialist?

System Optimization

Explore other alternative configurations for the clinic based on acquiring additional nurses, specialists, and/or exam rooms.

1. Find a configuration that minimizes the clinic's expected daily operating costs. Explain the process that you used to identify alternative configurations to explore and how you analyzed each one.

Hints

- The following code can be used to change the seed of the random number generator in DESMO-J (be sure to replace `<ARGS>` and `<SEED>` with appropriate values):

```
Experiment exp = new Experiment(<ARGS>);
// Set the seed for the random number generator
// (NOTE: Do this *before* connecting the experiment to the model)
exp.setSeedGenerator(<SEED>);
// Connect model and experiment
model.connectToExperiment(exp);
```

  When the model is connected to the experiment, any distributions within the model will get initialized appropriately using the experiment's random number generator. Setting the seed used by the experiment *after* connecting it to the model will **not** update the model's distributions, which is why the seed needs to be set prior to making the connection.

- When patients are waiting to see the specialist, they will most likely be kept in a queue. For the initial configuration with one specialist, the spots in the queue could be used to represent the examination rooms: if patients stay in the queue until the completion of their treatment by the specialist, then a check for available exam rooms upon the next patient's referral to the specialist could be done by simply checking the queue length.

  However, this arrangement will probably **not** work when multiple specialists are present, as each specialist would typically check the queue for the next patient to treat, and a situation could arise in which two specialists are both treating the first patient in the queue.

  Instead, the occupied exam rooms should be tracked separately from the queue of patients waiting to be seen by the specialist, and patients should be removed from the queue once their treatment by the specialist begins. You can use a state variable to track the number of available/occupied exam rooms, or DESMO-J's Res class to model the exam rooms as resources within the system.

- To ensure proper termination of the simulation model, you will need to have the patient generator stop producing new arrivals after 12 hours of simulation time. Additionally, you will need to ensure that the simulation remains running for enough time after the clinic doors close so that all patients in the clinic at 8 PM can be processed fully (which entails seeing the nurse and possibly specialist). While setting the stopping time to a large number, e.g., 16 hours, will probably ensure that all patients get processed, this will have the side effect of biasing the time-weighted averages (e.g., queue lengths) reported by DESMO-J (e.g., if the last patient leaves at 8:30 PM but the simulation runs until 12:00 AM (16 hours), then the average queue length will be computed using a 16 hour period instead of a 12.5 hour period). DESMO-J's `ModelCondition` class can be used to set additional stopping conditions for a simulation model.