Hospital Emergency Room

Queuing System L2 - Computer-Aided Simulations Lab

Henrique Pedro Roschel
Politecnico di Torino
Student id: S306718

I. Problem overview

Consider a Hospital Emergency Room where the arriving patients can be classified into 3 categories:

- **Red code** - very urgent;
- **Yellow code** - moderately urgent;
- **Green code** - not urgent.

Patients arrive upon time according to some unknown stationary arrival process, and the ratio between arrivals of patients with red, yellow and green code is 1:2:3 respectively.

In this hospital, there are K different medical units that can work in parallel. Patients are served in order of priority, i.e., a yellow code patient will only receive treatment if there are no queued red code patients, while a green code patient must wait until there are neither yellow nor red code patients in queue.

Upon the arrival of a red code customer, if a yellow or green code patient is being served, the service of the latter customer is interrupted and the service of the just arrived red code customers is immediately started. The interrupted service will be resumed later on.

We assume a set of hypothesis regarding the arriving process and the service time of patients, perform the simulations and analyze the delay and the queue size evolution along the simulation;

II. Proposed approach

A. Arriving process

The time inter arrival of patients is considered to be exponentially distributed with rate λ_A for the 3 different codes. For each arriving patient, its code is randomly generated, based on the ratio between red, yellow and green code patients.

On arrival, if one or more medical units are available, the treatment begins, independent of the patient's code. If there are no medical units available but it is a red code patient and there is a yellow or green code patient receiving treatment, the red code patient treatment starts and the yellow/green is interrupted. Otherwise, the patient is inserted on the queue. The described logical process is shown in Figure 1

B. Service process

The duration of the treatment of each patient is also considered to be exponentially distributed with rate λ_S, but in this case, the rate of service can be individualized, based on the patient's code, e.g. the treatment of red code patients may, on average, take longer than that of yellow code patients, which may take longer than green code ones.

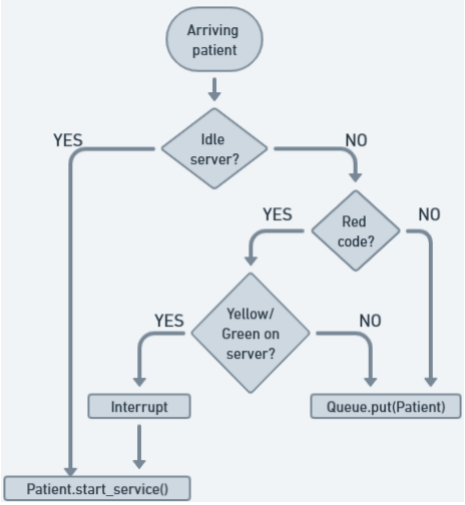

Fig. 1: Arrival process flowchart.

C. Interruption

The interruption routine happens when a red code patient arrives and a green or yellow code patient is receiving treatment. First, we check if there is a green patient on one of the servers, to be interrupted, i.e. a yellow patient can only be interrupted if there are no green patients on any server.

The patient to be interrupted will be the last one to start the service. The time of interruption and the remaining time of treatment are stored, and the patient is inserted in the queue, but in this case, on the front of it. We check if the first patients on the queue of the respective code were also interrupted, and the patient is inserted between the last interrupted patient and the first non-interrupted one.

D. Stop conditions

The chosen stop condition for the simulation combines maximum time of simulation (`MaxTimeCondition`) and the state of the queue (`PatientsQueue.isEmpty()`), i.e., the simulation stops after all clients who arrived before the specified maximum time receive treatment.

If we consider a eight-hour shift for the Hospital Emergency Room, we set `MaxTimeCondition` to 480 minutes, and all further time variables will be specified in minutes.

E. Measures

The proposed measures for the simulation are:

- **Average delay**: the average time a patient has to wait in the queue;
- **Average queue length**: average number of patients in queue along the simulation;

III. Experiments and Results

We consider an emergency room with only one medical unit per shift ($K = 1$). For specific values of rate of arrival λ_A and rate of service λ_S, $N = 1500$ simulations are performed and the confidence interval for the average delay and average queue length are computed with a confidence level 98% and plot the evolution of the queue along time for one or more simulations.

A. Unique service rate

First, we run the simulation with only one service rate for the three different urgency codes. The results are shown in Figure 2.

Regarding the average delay, we see that it reduces exponentially as the rate of service increases. Moreover, if we draw horizontal lines on the first graph, it is possible to see that the average time spent on queue by the patients is related to ratio between λ_A and λ_S. For example, the delay when $\lambda_S/\lambda_A = 0.8$ is approximately 90 minutes, which wee can see for the pairs (λ_A, λ_S): $(1.0, 0.8)$, $(1.5, 1.2)$ and $(2.0, 1.6)$.

As for the average length of the queue, it also reduces as λ_S increases, but this time on a linear fashion, until the rate of service becomes greater than the rate of arrivals. After that, the average queue size tends to zero, as expected.

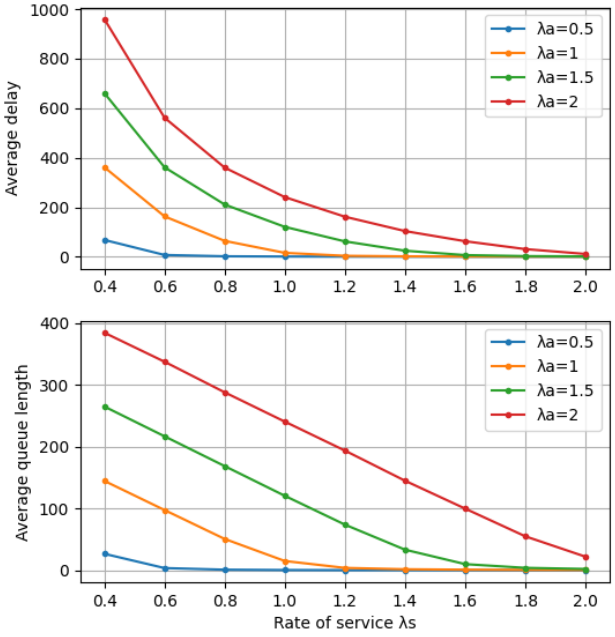

Fig. 2: Comparison between rates of arrival and service

B. Experimenting values for λ_A and λ_S

As a second experiment, we try to simulate with different values for the rates of arrival and service, while keeping $K = 1$. Each time, we can plot the evolution of the queue for one or more of the simulations performed, to compare these simulations and their results.

1) First scenario: First, we consider a scenario where one patient arrives every 2 minutes ($\lambda_A = 0.5$) and each treatment takes on average 3 minutes, independent on the urgency code ($\lambda_S = 0.333$). The obtained results are:

- Average delay: 25.646 ± 1.007 minutes;
- Average queue size: 7.924 ± 0.307.

When we look at the queue graphs, we can see that there were cases where the average queue size was greater or less than the global result, as shown in Figures 3 and 4, respectively.

In those simulations, it is possible to check the behavior of the simulation: due to priority policy, the queue contains mostly green code patients, as we expected. Not only because they are only removed from the queue where there are no yellow or red code patients on the system, but also because they are the majority of patients and usually the ones to have their treatment interrupted.

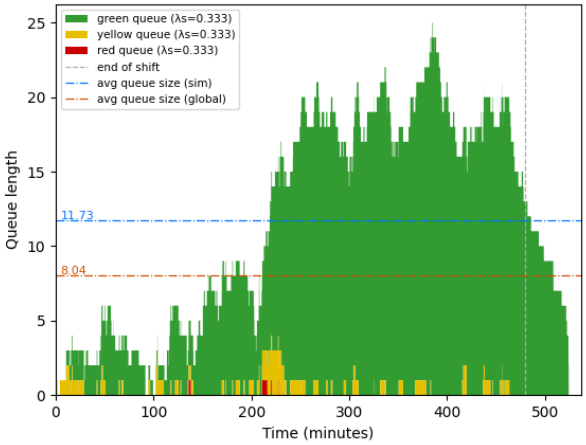

Fig. 3: Queue evolution for $\lambda_A = 0.333$ (simulation 1)

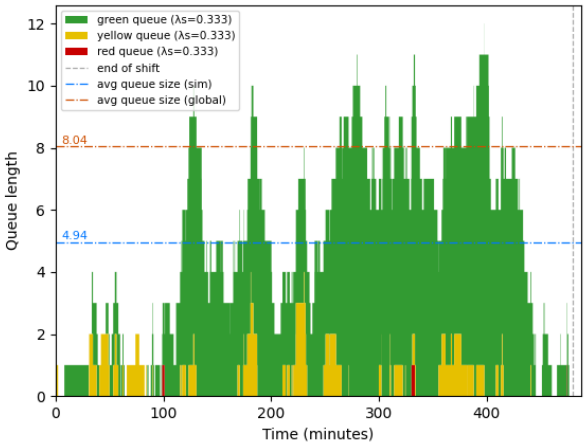

Fig. 4: Queue evolution for $\lambda_A = 0.333$ (simulation 2)

2) Second scenario: For the next scenario, consider that one patient arrives every 2 minutes approximately ($\lambda_A = 0.5$) and the service rate is different for each urgency code. Red code patients have an average treatment time of 10 minutes ($\lambda_{S,red} = 0.1$); yellow code, 4 minutes ($\lambda_{S,yellow} = 0.25$) and green code, 2 minutes ($\lambda_{S,green} = 0.5$)

Although these values seem reasonable, and may be even considered low with respect to the real ones, the results below do not correspond to what we would expect from a hospital emergency room, as we got:

- Average delay: 382.254 ± 5.267 minutes;
- Average queue size: 94.539 ± 0.947.

The expected time spent on queue by the patients is greater than 6 hours. If we look at the graph of the queue evolution for one of the simulations, as presented by Figure 5, we see that the queue length only increases from the start of the simulation until the arrival of the last patient. After that, it takes yet more than 10 hours for all patients on queue to receive treatment.

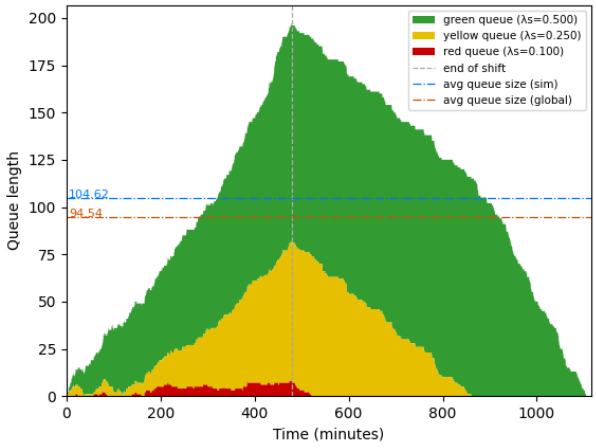

Fig. 5: Queue evolution for $\lambda_A = 0.5$ and individualized λ_S.

3) Third scenario: The results of the previous scenario are far beyond the ideal case. This is due to the fact that the number of servers was kept as $K = 1$. In this scenario we increase K to 2 medical units working in parallel, and obtain the following results:

- Average delay: 42.066 ± 1.941 minutes;
- Average queue size: 19.240 ± 0.849.

This time, we obtain more reasonable values for both measures. In a *bad case* (where the simulation average queue size was greater than the global one), as the simulation displayed by Figure 6, we can see that the time between the last arrival and the last departure dropped from 10 to 2 hours.

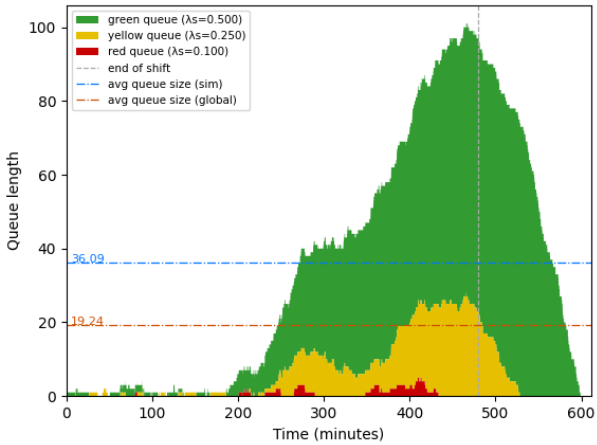

Fig. 6: Queue evolution for $\lambda_A = 0.5$; $K = 2$ (*bad case*).

However, there are also *good cases* where the service finishes before the maximum time of simulation, as the one shown in Figure 7.

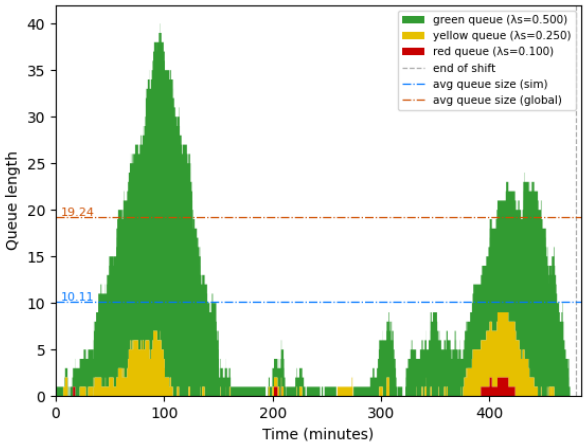

Fig. 7: Queue evolution for $\lambda_A = 0.5$; $K = 2$ (*good case*).

4) Fourth scenario: Finally, we can take it further to a higher number of parallel servers. Here, we consider 4 medical units; every 10 minutes, on average 7 patients arrive and; the average service time for red, yellow and green code patients are respectively 10, 6 and 3 minutes, i.e.:

$$K = 4; \qquad \lambda_A = 0.7;$$
$$\lambda_{S,red} = 0.100; \quad \lambda_{S,yellow} = 0.167; \quad \lambda_{s,green} = 0.333$$

For this setup, the results obtained, as exemplified on Figure 8, were:

- Average delay: 13.212 ± 0.730 minutes;
- Average queue size: 9.099 ± 0.498.

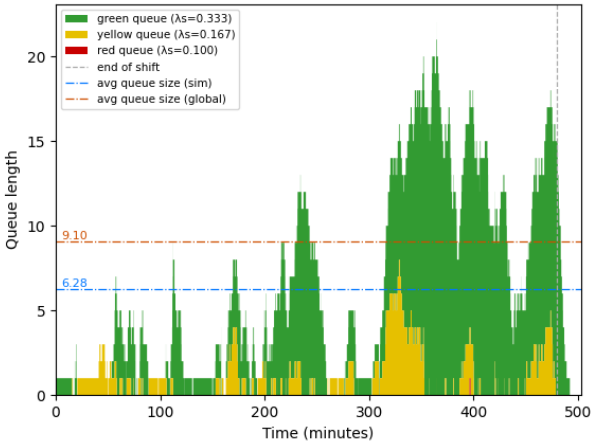

Fig. 8: Queue evolution for $K = 4$ and $\lambda_A = 0.7$.

Therefore, we are are able to simulate a more realistic scenario and obtain a results that fit in the considered context, where patients, on average, wait in queue for no longer than 15 minutes, the average queue length is also no longer than 10 patients and very urgent treatments usually are started immediately.