Lab2 - Simulating a Hospital Emergency Room

Martina Martini s306163
Politecnico di Torino

Assignment Write a simple simulator of of the queueing process within a Hospital Emergency Room. Arriving customers can be classified into tree categories:

- red code (very urgent)
- yellow code (moderately urgent)
- green code (not urgent)

Customers arrive upon time according to some unknown stationary arrival process: 1/6 of arriving customers are red code; 1/3 are yellow code and the remaining fraction (1/2) are green code. K different medical teams operate in parallel. Customers are served according to a strict-priority service discipline (a yellow-code customer enters service only when there are no red-code customers waiting, etc.) Upon the arrival of a red-code customer, if a yellow or green code customer 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. Make reasonable assumptions (and justify them) on the arrival process(es) of customers and on the service times of customers (choose suitable distribution(s)).

For $K = 1$, simulate a few scenarios, by varying the rate of arrival of customers and/or service times distributions. Write a brief comment on the assumptions you did and results you got.

1 System settings and assumptions

The management of a Hospital Emergency Room is similar to simulating a multi-servers FIFO queue. Differently by that easier case, here you need to take care of the priority codes of the arriving patients. Also, you can assume that the time to serve a patient with a red code is higher than the time to serve a patient with a yellow code, which is higher than the time to serve a green code. For this reason, you can assume different service rates based on the color code. In the code, you find:

- $SERVICE\_RED = 2.0$,
- $SERVICE\_YELLOW = 1.0$,
- $SERVICE\_GREEN = 0.6$.

However, the service times are exponentially distributed based on these rates and depending on the overcited probabilities.

Another assumption can be made in thinking of the waiting line as a list that has its own capacity (not infinite).

Also, you can consider different lambdas as arrival rates for the inter-arrival process, which are intended as exponentially distributed. In the provided code they are chosen among $arrival\_lambdas = [5, 7, 9, 10, 12, 15]$.

Finally, you can consider a finite time for the simulation (tunable for more comparisons), which in the provided code is set to $SIM\_TIME = 200$.

2 Working process

To better understand the processes taken under consideration, it follows a brief explanation about how the Emergency Room system works.

Step 1 At time 0 the parameters are set as presented in teh previous section:

- The number of free studios is equal to the total number of studios;
- The number of patients in the Hospital system is set to 0, yet.

Step 2 Firstly, the system assigns the color code to each patient just arrived based on the problem specifications. Also, once a patient arrives, he is put in a list containing the other patients with the same color code. At this time, the event of type arrival is created and inserted in the PriorityQueue FES (Future Event Set). Clearly, the first created event is an arrival type at time 0. From this to the following ones, each arrival is determined by the inter-arrivals (computed as explained in the previous section). At this point, depending on the associated color code, the system computes the service time and different paths can be followed.

Step 3a If the color code of the patient does not indicate high risk ($GREEN$ or $YELLOW$), then you can explore the following scenarios:

case 1 If the waiting line is full and there are no free studios, then the patient is dropped and the related variable is incremented.

case 2 If the waiting line is not empty but neither full and if there are some free studios, then the system checks the patients waiting in the waiting line. For each patient in the waiting line, if he has a green code, then the relative counter is incremented. Now, you can face up two cases:

In the first case the system checks if the just-arrived patient is green: if the counter is equal to the length of the waiting line, then all the people in line are green, so the just-arrived client is at the end of the row. Instead, if there's a yellow code in the waiting line, then the green patient is put in the last seat of the waiting line to leave priority to the yellow code.

In the second case if the the just arrived patient is yellow code: if the counter is equal to the length of the waiting line, then all the waiting people are green so the yellow one is served. Instead, if there is another yellow patient in the line then the just-arrived patient is put after the already-present yellow patient.

The process, in this case, continues with the call of the function *departure*(), which takes the first patient in the row under consideration to be served. For him, the service time and the departure time, as well as the event in the FES, are set.

case 3 If the waiting line is empty but there are some free studios (possibly all of them are free), then the just-arrived patient can be served immediately so, the departure time is scheduled, the free studio is occupied and the number of patients in the system is incremented.

Step 3b If the color code of the patient indicates high risk (RED), then you can explore the following scenarios:

case 1 If there are some free studios available, then one of them is occupied by the just-arrived patient, the departure time is set and the event is created. Notice that each time a patient is served (for all of the color codes) the systems stores him in a PriorityQueue called *clients_served* based on his time of arrival.

case 2 If there is no free studio available, one of the clients present in one of the studios is stopped in order to leave the priority to the red code. Then, the client stopped is put in the FES to take trace of the time of the event, which is exactly the time in which the red patient arrives. Now, the red code patient is served, the FES is fulfilled and the number of users in the system increases.

Step 4a In the $GREEN$ or $YELLOW$ cases, after the patient's departure, the patient stops being served and the number of patients in the system decreases.

Step 4b In the RED case, after the departure of the red code, the stopped patient can continue to be served, so the new departure time is set based on how much time the stopped patient needs to conclude the service time. Now, another event is created but the counter of the patients present in the system is not increasing since the stopped patient was already counted in the system before.

Step 5 After the event of departure, the studio is set back to free for the next service.

3 Final considerations

The simulation results are clear to notice: when the number of available studios k increases, the process can be parallelized, so the number of dropped patients decreases as this quantity increases. When the number of available studios becomes equal to 1, the time spent waiting and the probability of being dropped are surely higher. Also, if the lambda becomes higher, the number of dropped patients reaches very high percentages. To conclude, more considerations can be made by modifying the setup parameters in order to make more comparisons.