Project Nr. 2: Machine Repair Infrastructure

System Description

N identical machines that are used continuously for producing manufactured parts, may occasionally fail and need to be repaired. Failures can be of different types and the repair infrastructure is organized in three sections devoted to different aspects of the repair action: testing of the failure and certification of the repair, repair of serious failures, and provision of spare parts. The "life-cycle" of each machine can thus be described in the following way. A machine that is performing normally for a certain amount of time, may fail and is thus taken to a *repair shop* (also called *repair infrastructure*) where it is first checked by a technician to identify the type of repairs it needs. We call these machine *failed*. During this preliminary analysis, if the machine underwent a small failure, a quick fix is performed and the machine is certified as "repaired" and sent back to its original manufacturing plant and considered to be "operational" again. If instead the problem is more serious, the machine is sent to the actual "repair lab" being qualified as *faulty* if a short repair operation needs to be performed or considered *damaged* if a "long repair" is deemed to be necessary on the basis of the results of the preliminary tests. A specialized technician of the repair lab works on the faulty machine and sends it back to the manufacturing plant as "repaired". When performing a "long repair" operation on a damaged machine, the action may be interrupted because of the need of spare parts that become available after a certain amount of time. After changing all the parts that are needed (possibly repeating the repair/spare-part cycle a few times), the machine is completely repaired, identified as "serviced", and sent back to the "testing" station for a final quality control. Usually this final test is successful and the machine is sent back to the manufacturing plant as "repaired", but it is possible that the repair is not yet satisfactory, in which case the machine is returned to the lab as a damaged one for further fixing.

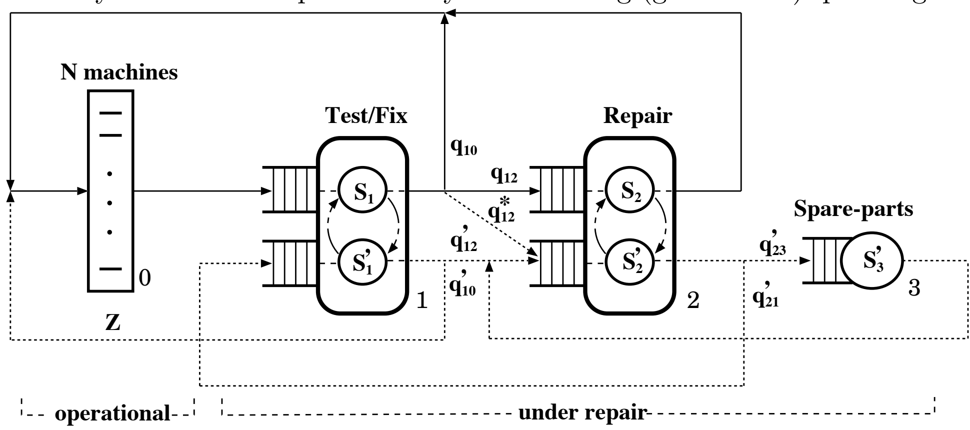
When several machines are simultaneously undergoing the repair operation some additional features of the organization of this repair infrastructure need to be specified to account for the interference that may occur among machines that are at different repair stages. The most important additional aspect that becomes relevant when several machines are simultaneously being repaired, is the priority that is given at the repair actions performed by the repair infrastructure both in the Test/Fix station and the Repair lab.

Modelling assumptions

Repair shops of this type are very complex and even the small version considered before would require quite a lot of additional specifications to make the description complete.

For the purpose of this project, we can assume (1) that the number N of machines is fixed, (2) that the repair infrastructure has enough room to accommodate all the machines that may require service, (3) that all the queues in the system are managed with FCFS policies, (4) that the intrinsic variability of all the operations performed by the different sections of the infrastructure are captured by assuming that the service times of all the components of this system are represented by random variables, and (5) that all the choices concerning the routing of the machines among the different nodes of the model are probabilistically decided. Moreover, it is assumed that both in the "test and fix" station and in the "repair" station only one technician is working in each of them and that different input queues are used to distinguish between failed and serviced machines in the test and fix station and between faulty and damaged machines in the "repair" station.

A model of this system can be represented by the following (generalized) queueing network.



The machines are assumed to work in parallel and thus the Manufacturing plant can be assumed to be a station of *Infinite Server* type. The "spare part procurement" node is a single server queueing station. The "Test and Fix" and the "Repair" stations have more complex behaviours since they have two input queues each and the technicians (the servers) have to implement the priority policy chosen by the "Management". The operation of these two stations is the element that makes the analysis of this model complex because of the necessity of considering machines with different needs. For this reason Discrete Event Simulation is needed for assessing the behavior of this system.

Given all these assumptions and that in the "Test and Fix" and "Repair" stations priority is given to serviced and damaged machines, the parameters of the model are the following:

- Average operational time (also time to next failure) $Z = 2500\,min$,
- Average "testing" time $S_1 = 5\,min$,
- Average "fix" time for faulty machines $S_2 = 50\,min$
- Average "quality control" time for serviced machines $S'_1 = 4\,min$
- Average "long-repair" time of damaged $S'_2 = 200\,min$
- Average spare part provision time $S'_3 = 500\,min$
- Testing choices for "failed" machines $q_{1,0} = 0.1$, $\quad q_{1,2} = 0.5$, and $q^*_{1,2} = 0.4$
- Quality control choices for "serviced" machines $q'_{1,0} = 0.9$, $\quad q'_{1,2} = 0.1$,
- Spare part provision choices $q'_{2,3} = 0.30$, $\quad q'_{2,1} = 0.70$.

All the stations are characterized by negative exponential distributions, except for the testing station that has a uniform distribution $U(3, 7)$ and the long-repair section of damaged machines of the repair lab which has a two stage hyper-exponential distribution

$$f_X(x) = \alpha * 1/\mu_1 * exp(-x/\mu 1) + \beta * 1/\mu_2 * exp(-x/\mu 2)$$

with parameters $\alpha = 0.95$, $\beta = 0.05$, $\mu_1 = 60\,min$, and $\mu_2 = 2860\,min$.

Analysis

Write the simulator of this simple system assuming the presence of 20 Machines (in practical situations, the number of machines involved in systems of this type is much larger, but the same is true also for the "time_between_failures", so that we can assume the load of the repair infrastructure to be reasonable). Provide interval estimates for

- the 'average total repair time of damaged machines (the time required for a damaged machine to be certified as "repaired" **after** having its failure properly diagnosed (i.e., excluding the time required for the tests to be performed).

Before starting any implementation, identify in a clear manner all the events that need to be managed as well as their effect on the state of the system (and on the statistics that you are collecting during the course of the simulation).
Use separate random number streams for the different random variables present in the model.
Use the regenerative method to compute the confidence intervals, starting the simulation from a suited regeneration point.
Provide the results with a 90% confidence level and a precision of $\mp 5\%$.

Validation

In order to check the correctness of the simulator that you have just implemented, you need to compare the results that you are obtaining with those provided with alternative methods applied to a simplified version of the model.

Three are the elements of the system studied in this project that make simulation the only method for its analysis. Specifically, the presence of technicians that have to managed two input queues each, the priority management scheme, and the general Uniform and hyper-exponential) distributions of the durations of two activities of the repair lab. Modifying these features of the system it is possible to construct a model that is suited for alternative methods of analysis.

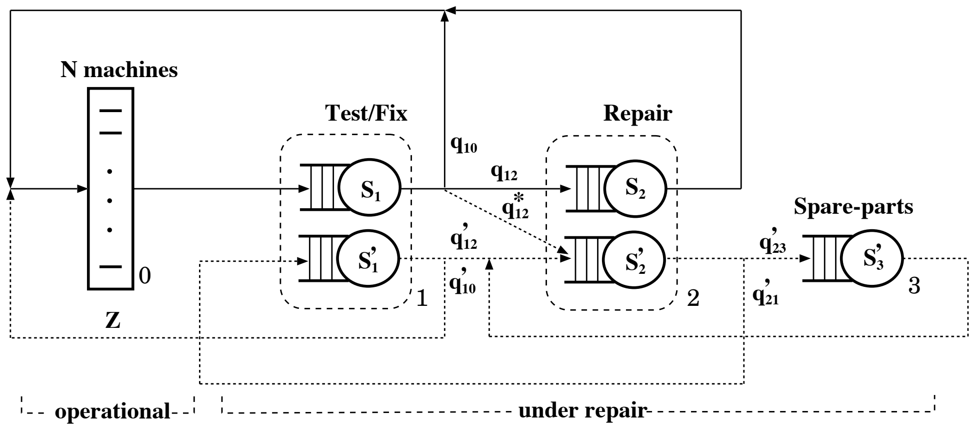
- **First Validation Step**

  The operation of the Text/Fix and Repair stations can be captured in an approximate manner by assuming that (i) each of the four queues of the Test/Fix and Repair stations have a dedicated technician so that the priority structure mentioned in the description of the project becomes irrelevant and (ii) the (individual) testing time and long-repair time of damaged machines have negative exponential distributions with averages equal to $s_1 = 5\, min$ and $s_2' = 200\, min$,respectively.

  With this simple modifications, the model becomes a "Product Form Queueing Network" (with 6 stations) for which a Bottleneck Analysis can be preliminary performed, before computing the average repair time of faulty machines and the average repair time of damaged machines using the Mean Value Analysis (MVA) algorithm and assuming that the number of machines present in the system varies from 1 up to 50.

  To get confidence on the correctness of your simulator, check first that the interval estimate(s) include the theoretical values obtained with the MVA algorithm.

  To increase your confidence on the simulator that you have developed, perform an extensive validation using the technique discussed at the end of the course.



Having implemented the detailed simulator according to a specified priority structure (for instance assuming that serviced and damaged machines have priority on failed and faulty ones, you can use it for validation purposes by simply introducing the following modifications:

- Make sure that every time a machine enters an empty queues it always find a flag that says that the technician is available and is not busy serving a machine of the other type.
- When a machine completes its service time (for whatever reason), it always checks the queue of machines of its same type and puts in service a machine of its same type if there is one.
- Replace the call to the generators of instances of the uniform and hyper-exponential distributions with that of the generator of instances of the negative exponential one.

- **Second Validation Step**

  To improve the confidence on your implementation, it is possible to release some of the restrictions introduced in the simplified model of the first validation step, focusing the attention on the representation of the Repair section of the lab.

  With the objective of obtaining a model that can be analyzed as a Markov Chain, we only need to assume that the Testing time has a negative exponential distribution with average $s_1 = 5\,min$.

  To keep the model simpler, we can also assume that two technicians are managing the test and quality control queues, while a single technician (as in the original model) works in the repair section of the lab whose long-repair time distribution is now that specified for the original model.

  With these modifications, a Markov Chain can be constructed for the analysis of the simplified version of the system. The size of the Markov Chain (the number of possible states) depends on the number of stations in the network and on the number of workstations connected to the system. Since the focus of this phase of the analysis of the system is on the performance of the repair section of the lab, we can reduce the size of the Markov Chain by assuming that the lab has enough “Spare Parts” so that their procurement time can be set to zero (non queue can form in front of the Spare-parts station).

In summary the Markov chain can be constructed using the following model specifications:

- Average operational time (also time to next failure) $Z = 2500\, min$,
- Average "testing" time $S_1 = 5\, min$,
- Average "fix" time for faulty machines $S_2 = 50\, min$
- Average "quality control" time for serviced machines $S'_1 = 4\, min$
- Average "long-repair" time of damaged $S'_2 = 200\, min$
- Average spare part provision time $S'_3 = 0\, min$
- Testing choices for "failed" machines $q_{1,0} = 0.1, \quad q_{1,2} = 0.5$, and $q^*_{1,2} = 0.4$
- Quality control choices for "serviced" machines $q'_{1,0} = 0.9, \quad q'_{1,2} = 0.1$,
- Spare part provision choices $q'_{2,3} = 0.30, \quad q'_{2,1} = 0.70$.

As already stated before, all the activities are characterized by negative exponential distributions, but that of the long-repair which has a two stage hyper-exponential distribution

$$f_X(x) = \alpha * 1/\mu_1 * exp(-x/\mu 1) + \beta * 1/\mu_2 * exp(-x/\mu 2)$$

with parameters $\alpha = 0.95$, $\beta = 0.05$, $\mu_1 = 60\, min$, and $\mu_2 = 2860\, min$.
- Priority is given to serviced and damaged machines.

If the Markov chain is constructed manually, set the number of machines $N = 3$.

Also in this second validation step, to get confidence on the correctness of your simulator, check first that the interval estimate(s) includes the theoretical value(s) derived with proper manipulations of the steady-state solution (steady state probability distribution) of the Markov Chain and then perform an extensive validation using the technique discussed at the end of the course.

Again, having implemented the detailed simulator first, you can use it for validation purposes by simply introducing the following modifications:

- Change the "Arrival" and "Departure" functions of the Test and Quality Control queues in order to account for the dedicated technicians
- Call the negative-exponential generator for producing instances of the testing times.

Notice, that if two dedicated technicians are assumed to oversee the repair section of the lab and the two stages of the hyper-exponential distribution of the long-repair times are set identical with the average of $200\, min.$, the hyper-exponential distribution reduces to a negative exponential one with expected value equal to $200\, min.$ and it becomes possible to show that the Markov Chain is isomorphic to that underlying a Product Form Queueing Network similar to that used in the first validation step with the only difference of having $s'_3 = 0 min$.
This network can be solved again with the MVA algorithm thus providing a way of checking whether the implementation of the steady-state solution of the Markov chain yields exact values.

Hints:

The validation of the simulator against the results obtained with the numerical methods (MVA algorithm and/or Markov Chain solution), is more effective if smaller are the changes introduced in the simulator to adapt it to the simplified model.

Whenever it is possible, control these modifications with properly chosen parameters that can be easily modified to make your program to simulate the detailed version of the model as well as the simplified one.

Report

Summarize the results of your work writing a brief document in which you

- describe the basic features of the simulator you have implemented, with a special emphasis on the events you have identified (list all of them) and on their effect on the state of the system (as well as on the statistics that you are collecting during the course of the simulation).
  Give specific emphasis on the description of the handler of the events that mostly characterize the project.
  Make sure that your report contains all the information needed to grade the project.
  The code that you submit must be considered as complementary material that can be used during the grading to remove possible doubts, but should not be assumed to replace the description contained in the report.
  - define in a precise manner the regeneration point you have used in your simulation;
  - provide a print-out (or a way of getting the print-out) of the state of the system after the handling of each of the first 30 events processed by the "engine" of your simulator;
  - provide ways to allow turning on/off the possibility of printing the measures performed during each regeneration cycle and used to compute the confidence intervals requested by the text of the project;
- devote a particular attention to the description of the validation process you have used;
- comment on the performance of the system you have simulated providing the results for the following four cases
  1. failed and faulty machines have priority on the serviced and damaged ones (respectively on the Test/Fix station and on the Repair station); repair times of damaged machines have hyper-exponential distribution with parameters $\alpha = 0.95$, $\beta = 0.05$, $\mu_1 = 60\,min$, and $\mu_2 = 2860\,min$.
  2. failed and faulty machines have priority on the serviced and damaged ones (respectively on the Test/Fix station and on the Repair station); repair times of damaged machines have negative exponential distribution with average $\mu = 200, min$
  3. serviced and damaged machines have priority on the failed and faulty ones (respectively on the Test/Fix station and on the Repair station); repair times of damaged machines have hyper-exponential distribution with parameters $\alpha = 0.95$, $\beta = 0.05$, $\mu_1 = 60\,min$, and $\mu_2 = 2860\,min$.
  4. serviced and damaged machines have priority on the failed and faulty ones (respectively on the Test/Fix station and on the Repair station); repair times of damaged machines have negative exponential distribution with average $\mu = 200, min$