Modeling, Testing and Evaluating the Performance of an Exam Session

Francesco Bruno

Marco Imbelli Cai

Nicolò Picchi

Università di Pisa, DII

Master degree in Computer Engineering, course of Performance Evaluation of Computer Systems and Networks

Table of contents

Introduction

The goal of this analysis is to model and then study the performance of an oral exam session.

The latter is managed by a certain number of teachers N and is attended by students. Each student who starts the exam needs to answer N questions to consider his/her exam completed, and there is always one that needs to be examined. Each student will take t seconds to answer one single question, where t is a random variable that can have a:

- *Uniform* distribution
- *Lognormal* distribution

Each teacher can examine one student at a time.

The teachers can manage the exam using one of the following methods:

- *One student per teacher:* each teacher asks a sequence of N questions to a student. The student completes his/her exam after answering the N-th question, and is allowed to leave the system. The teacher will then be able to examine another student.
  From now on, this method will be called *Parallel* method.
- *Pipelined:* teachers are sorted and ask one question each. After a student answers the i-th question (which has been asked by the i-th teacher), he/she will move on to the $(i + 1)$-th teacher, all the way up to N. The first teacher examines a new student as soon as the latter is not asking a question. The student completes his/her exam after answering the N-th question, and is allowed to leave the system.

Modeling

Assumptions

Some assumptions have been made during the modeling and development phase of the system:

- The communication between a student and a professor is instantaneous. This means that the time required by a student to "move" inside the room to get to the professor(s) and the time required for the professor to tell the question to the student are both negligible quantities, hence they are considered equal to zero in this analysis.
- When a student has finished answering a question, the professor doesn't need any additional time to think about a new one.

Implementation

The simulator used to analyze project has been developed using the *OMNeT++* framework, version 6.0.1.

The analysis is carried out on the *Exam* network, inside of which we can find two types of modules:

- ***Committee* module:** used as a container for the N "Prof" modules.
- ***StudentGenerator* module:** used as a generator for the different students that will be examined.

Each student is implemented as a *cMessage* instance, created by the *StudentGenerator* module, and handled differently by the latter and by the professors depending on the examination mode in use.

In particular:

- **In the parallel case:** at the start, the generator creates N different students, and sends one to each professor. The professor, upon receiving a student, starts asking a series of N questions. The student is then sent back to the generator: this notifies the latter that the student has been examined, and that a new one can be sent to the professor (Figure 1).
  Note that there is no need to create new student objects: since N of them were allocated at the start, they can be re-used until the simulation finishes.
- **In the pipelined case:** at the start, just one student is generated, and is sent to the first professor. He will be asked a question, and will be sent back to the generator. The student will then be sent to the next professor in the pipeline only if the latter is not busy. If this is not the case, it will be placed in a queue, and it will be examined when its turn will come (Figure 2).

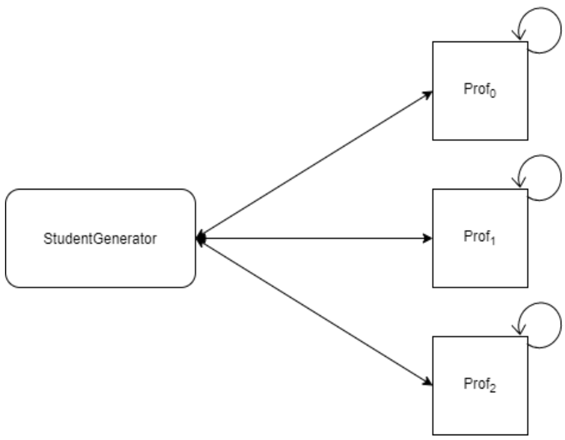

Figure 1: Parallel method, N=3

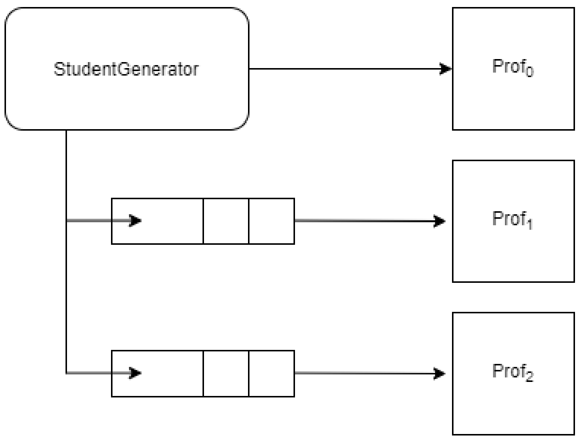

Figure 2: Pipeline method, N=3

The complete implementation of the system's actors can be found in the following files:

- Student generator: *StudentGenerator.ned, StudentGenerator.cc, StudentGenerator.h*
- Professor: *prof.ned, prof.h, prof.cc*
- Student: *Student.msg, Student_m.h, Student_m.cc*
- Exam committee: *committee.ned*
- Exam network: *exam.ned*

Performance indices

In the *Parallel* examination mode, the studied index is:

- *ExaminationTime* : computed for each examined student. It measures the total time needed from a student to answer N questions (summation of N answer times).

In the *Pipelined* examination mode, the studied indexes are:

- *ExaminationTime* : computed for each examined student. Measures the interval of time that passes between the instant in which the student starts answering the first question and the instant in which he finishes answering the last one (the N-th one).
  It is computed by summing up the time needed to answer N questions and the *WaitingTime*, which is defined below .
- *WaitingTime* : measures the time spent by each student inside the pipeline without being examined, so it's the sum of the intervals of time in which the student is in a queue.
- *IdleTime* : it's computed for each teacher, and consists in the sum of the intervals of time in which the professor is not examining a student (hence, in which he's being idle).
  In particular, the following intervals of time are considered:
  - The interval of time between the beginning of the simulation and the instant in which the teacher receives its first student.

- The interval of time between the moment in which a teacher sends back a student to the generator (notifying it that that student has been examined by the professor) and the moment in which the same professor receives a new one.

Calibration

Uniform distribution

In the uniform scenario the time (t) that a student takes to answer a question is a random variable $t \sim U(300, 600)$ s. The variable's lower and upper bounds correspond to 5 and 10 minutes respectively. The chosen values yield a mean value for t of 450 seconds (between 7 and 8 minutes): this is a fair approximation of the answer time during a real oral exam session.

Lognormal distribution

The two scenarios need to be comparable, hence the mean value for the lognormal distribution is set to 450s (the same of the uniform distribution one).
The value for the standard deviation is estimated using the PDF and the CDF of the lognormal distribution (plotted by using Matlab). This value is estimated through visual inspection, and is set to 150s. The value is chosen in such a way so that the distribution can fit the real behavior of a group of students during a real oral exam: according to the CDF, 10% of the students will answer questions in less than 5 minutes and only 15% of them will need more than 10 minutes to do so.

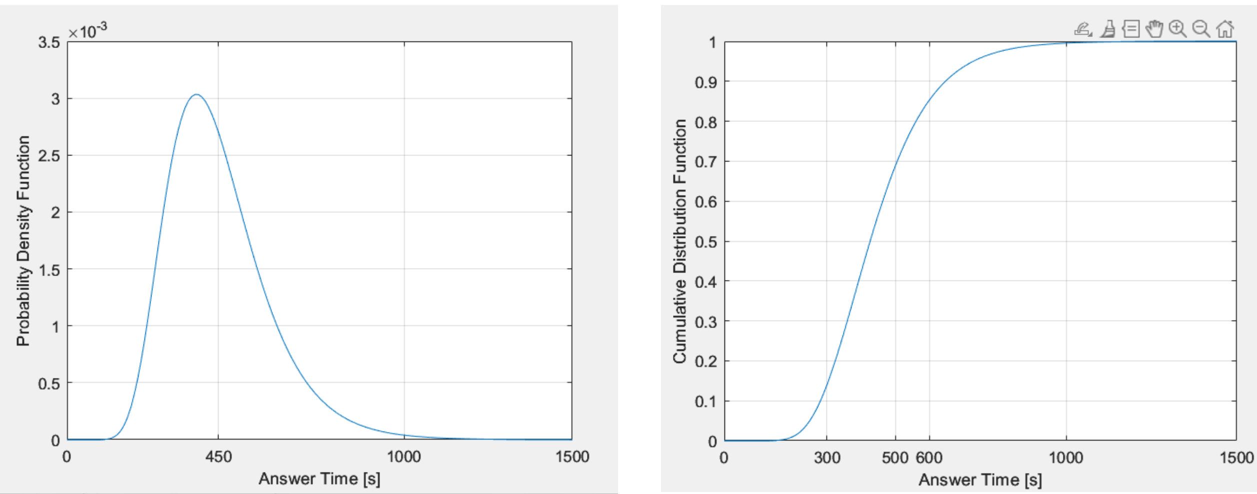

Figure 3: Probability density function and cumulative distribution function of a lognormal random variable

Lognormal and normal distributions are related to each other according to the following relationship*: X is a random variable with a Lognormal distribution if Y = ln(X) is Normally distributed.*
In *OMNeT++*, lognormal parameters are the mean and the standard deviation of the corresponding normal distribution.
These parameters can be computed by employing the following formulas:

$$\begin{cases} \mu_{Normal} = 2 \cdot \ln(\mu_{Lognormal}) - \frac{1}{2} \cdot \ln(\mu_{Lognormal}^2 + \sigma_{Lognormal}^2) \\ \sigma_{Normal} = \sqrt{-2 \cdot \ln(\mu_{Lognormal}) + \ln(\mu_{Lognormal}^2 + \sigma_{Lognormal}^2)} \end{cases}$$

The computations yield a mean value of 6,0566 and a standard deviation od 0,3246.

The RV t will then be distributed according to $t \sim LN(6.0566, 0.3246)$.

Model verification

The verification phase is useful to understand if the model correctly implements the system under analysis. To do so, the number of students that have completed the oral exam in a given *Simulation Time* is monitored (i.e., the network's *Throughput* over an entire *Simulation Time* window). Recall that a student is considered examined when he/she has answered all the N questions, both in the *Parallel* and *Pipelined* methods.

The verification phase of the model has been structured in two phases:

- **Deterministic answer time:** the time needed by a student to answer a question is fixed and is the same for all the students
  - Analysis in the *Parallel* case
  - Analysis in the *Pipelined* case
- **Random answer time:** the time needed by a student to answer a question is randomized according to the uniform and the lognormal distributions
  - Analysis in the *Parallel* case

The tests have been performed with the following choice of parameters:

Simulation Time	21.600s = 6 hours
N	{1, 3, 5, 7}

Table 1: verification tests parameters

Deterministic answer time

As explained in the above paragraph, the first test consists in employing a deterministic answer time. Its value has been set to the mean value of Uniform distribution, so equal to 450s.

In this way, the results computed by math and the ones provided by the simulator itself are comparable: the following paragraphs explain the used formulas and the gathered results.

Parallel method

Table 2.1 contains the math results computed by using the below equations, while Table 2.2 shows the simulator results:

N	Simulation Time	#ExaminedStudents
1	21.600	48
3	21.600	48
5	21.600	45
7	21.600	42

Table 2.1: math results

N	Simulation Time	#ExaminedStudents
1	21.600	48
3	21.600	48
5	21.600	45
7	21.600	42

Table 2.2: simulator results

The formulas used to compute the values are the following, where t has been set to 450s:

$$\#ExaminedStudents = N \cdot \left\lfloor \frac{Simulation\ Time}{ExaminationTime} \right\rfloor$$

$$ExaminationTime = N \cdot t$$

Pipelined method

In this case, *#ExaminedStudents* is computed as follows, where t has been set to 450s:

$$\#ExaminedStudents = N \cdot \left\lfloor \frac{Simulation\ Time}{ExaminationTime} \right\rfloor \qquad if\ N = 1$$

$$\#ExaminedStudents = 1 + \left\lfloor \frac{SimulationTime - ExaminationTime}{t} \right\rfloor \qquad if\ N > 1$$

ExaminationTime is instead computed in the same way as before.

The results are shown in Table 3.1 and Table 3.2:

N	Simulation Time	#ExaminedStudents
1	21.600	48
3	21.600	46
5	21.600	44
7	21.600	42

Table 3.1: math results

N	Simulation Time	#ExaminedStudents
1	21.600	48
3	21.600	46
5	21.600	44
7	21.600	42

Table 3.2: simulator results

It's quite straightforward that, if $N = 1$, the number of examined students can be computed in the same way as it's done in the *Parallel* case. Instead, if $N > 1$, the second formula has to be applied: $(SimulationTime - ExaminationTime)$ is, as a matter of fact, the time left to the students, after the first one, to complete the exam. Then, after t seconds have passed, the second student has finished his exam too, and the third one is still missing the last question to be considered examined (so he/she will still take a time t to finish being examined). In general, after the passing of t seconds, a student will finish his/her exam, and the next one will still require to be asked the last question to be considered examined too. Hence, $(SimulationTime - ExaminationTime)/t$ will yield the number of examined students after the moment the first one has finished his/her exam. The latter is then added to the result.

Note that, by employing a deterministic t (i.e., the answer time), both the *WaitingTime* and *IdleTime* quantities are null. There is no waiting time simply because there is no enqueueing in each professor's queue: in the moment in which a student has finished answering a question of a specific professor, he/she moves directly to the next one (or he/she has finished the exam), and a new one will arrive to said professor. The idle time of a professor is null too, since the time that a student would need to move from the generator to the latter has been assumed to be zero. In this way, each professor always has a student under examination.

Introduction of randomness

Once the randomness was introduced, the following tests were carried out to verify the correctness of the results obtained by the simulation with those computed by employing formulas. In all cases, only one repetition of a week of duration (21.600 s) was performed. Note that it's very difficult to carry out a comparison of the results computed by math with those obtained from the simulator when using a lognormal distribution, because of the high value of its standard deviation. For this reason, only the uniform scenario is considered, and the results are reported below.

Parallel method

The number of examined students is computed with the following formula, and is compared with the results obtained with the simulation. The data can be found in Table 4.1:

$$\#ExaminedStudents = N \cdot \left\lfloor \frac{Simulation\ Time}{E[Examination\ Time]} \right\rfloor$$

	E[Examination Time]		***#ExaminedStudents***	
N	***Simulator results***	***Math results***	***Simulator results***	***Math results***
1	$468s$	$450s$	46	48
3	$1.368s$	$450 \cdot 3 = 1.350s$	45	48
5	$2.270s$	$450 \cdot 5 = 2.250s$	45	45
7	$3.200s$	$450 \cdot 7 = 3.150s$	43	42

Table 4.1: Math results and simulator results

Pipelined method

In the *Pipelined* method, *ExaminationTime* is the sum of *AnswerTime* and the *WaitingTime*. The latter is a random variable of an unknown distribution: obtaining a mathematical formulation for it to explain the simulation results is very difficult. However, to evaluate the correctness of the system's behavior in this scenario, OMNeT++ event logs are used. Thanks to the simulator's graphical interface it's possible to real-time check if what's happening during the simulation is what reflects what manual computations yield.

Considerations

The results show that when employing the *Parallel* examination mode, by increasing the number of teachers in the exam committee (and therefore the number of questions asked to each student), the total number of examined students remains quite the same.
This behavior can be explained by the following formula:

$$
\begin{aligned}
\#ExaminedStudents &= N \cdot \left\lfloor \frac{Simulation\ Time}{E[Examination\ Time]} \right\rfloor \\
&= N \cdot \left\lfloor \frac{Simulation\ Time}{E[\sum_{i=1}^{N} tAnswerTime_i]} \right\rfloor \\
&= N \cdot \left\lfloor \frac{Simulation\ Time}{\sum_{i=1}^{N} E[AnswerTime_i]} \right\rfloor \\
&= N \cdot \left\lfloor \frac{Simulation\ Time}{N \cdot E[AnswerTime_i]} \right\rfloor
\end{aligned}
$$

Warm-up time estimation

The *Warm-up Time* parameter has been estimated by considering the *Throughput* (i.e., the number of examined students in a given interval of time) of the network over a *Simulation Time* interval.

The used values are the following:

Simulation Time	600.000s ≈ 1 week
Repetitions	10

Table 5: warm-up time estimation parameters

The above values yield the following results:

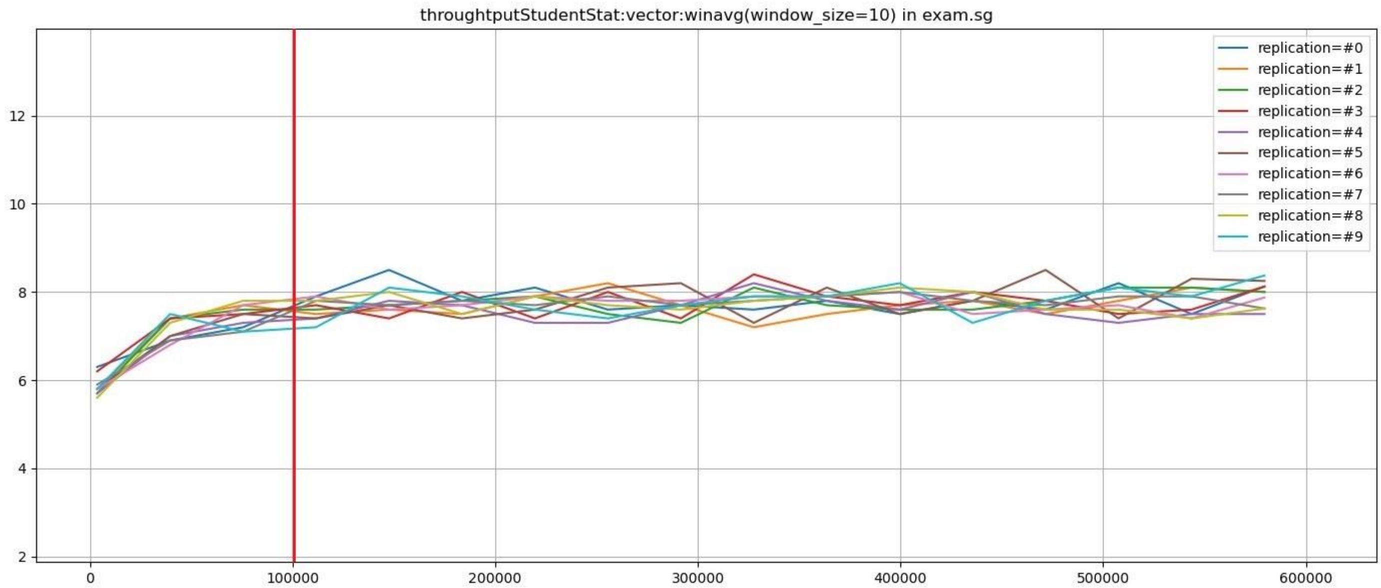

Figure 4: Warm-up time estimation, the red line marks the chosen value

The chosen value for the *Warm-up Time* parameter is then 100.000s.

After choosing a *Warm-up Time* value, the following parameters are set:

Simulation Time	700.000s ≈ 8,1 days
Repetitions	35

Table 5.1: Simulation Time and Repetitions values

An alternative way of choosing an appropriate *Warm-up Time* period could be setting a threshold value S of completely examined students (e.g., $S = 3$), and to use it in the following way: the *Warm-Up Time* parameter will be set as the interval of time between the moment the simulation starts and the moment in which the system reaches a total of completely examined students (i.e., students which have answered to N questions) of S. The chosen method can, instead, lead to a (negligible) overestimation of the *Warm-Up Time* parameter, but has the advantage of ensuring that the system will be fully operational when data will begin being collected.

Data analysis

All the following analysis have been carried on by considering the 95% confidence interval for the data.

Examination Time

For this analysis, data has been collected from the simulator by considering the possible combinations of number of teachers in the exam committee (N) and the chosen distributions (uniform and lognormal). Data has been aggregated by computing the sample mean of the collected values. Plots can be seen in Figure 5.1 and Figure 5.2.

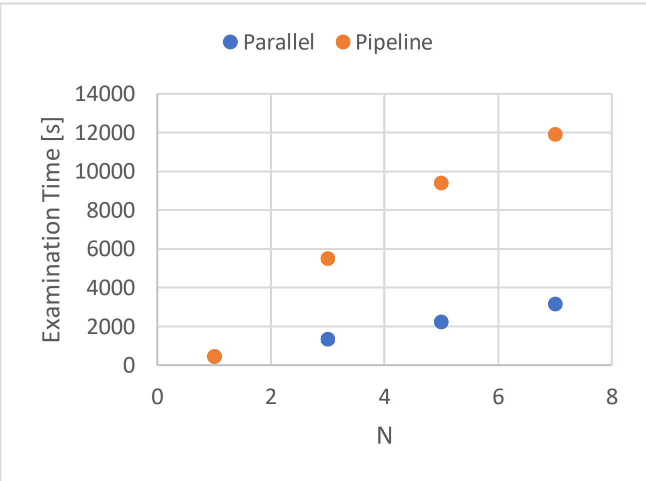

Figure 5.1: Uniform scenario

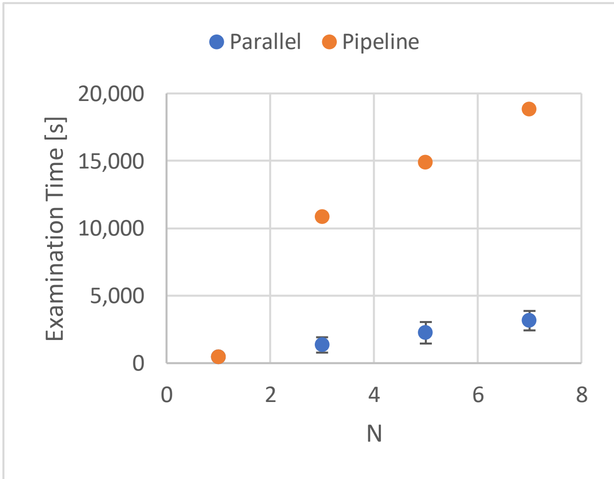

Figure 5.2: Lognormal scenario

In the uniform scenario (Figure 5.1), confidence levels are too small to be visible on the graph. The values are the following (all numbers are rounded to two decimal places, for readability purposes):

Parallel method	450,23 ± 0,76	1.349,67 ± 1,28	2.249,61 ± 1,65	3.151,24 ± 1,93

Table 6: Confidence levels for the Examination Time in a uniform scenario

For what concerns the *Pipelined* examination mode measurements (in both the uniform and lognormal scenarios), the collected values turn out to be unbounded (the sample variance keeps growing as the sample width increases), hence the confidence intervals for the sample mean cannot be computed. Anyway, by keep on collecting other samples for increasing values of N, the trend follows the one shown in the above graphs.

From Figure 5.1 and 5.2 it's clearly visible that *Examination Time* increases when increasing the number of teachers in the committee (N). Anyways, its value is always smaller when employing the *Parallel* method than when using the *Pipeline* one, and this can be said for both the used scenarios (uniform and lognormal). This is of course due to the fact that the *Pipeline* method introduces queueing: this generates a *Waiting Time*, which increases as the number of teachers increase. A more detailed analysis of the *Waiting Time* index can be found in the ***Waiting Time*** section.

Moreover, the graphs show that the *Examination Time* measured in the lognormal scenario is generally higher than the one measured for the uniform one. This is because of the way lognormal parameters have been chosen: each student has a 15% probability of answering each question in more than 600s (10 minutes).

Number of examined students

The collected data about the average number of examined students is shown in Figures 6.1 and 6.2. Recall that a student is considered examined once he/she has answered all the N questions.

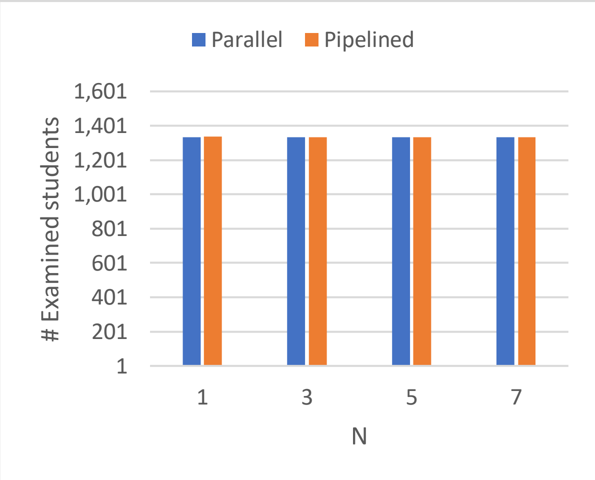

Figure 6.1: Uniform scenario

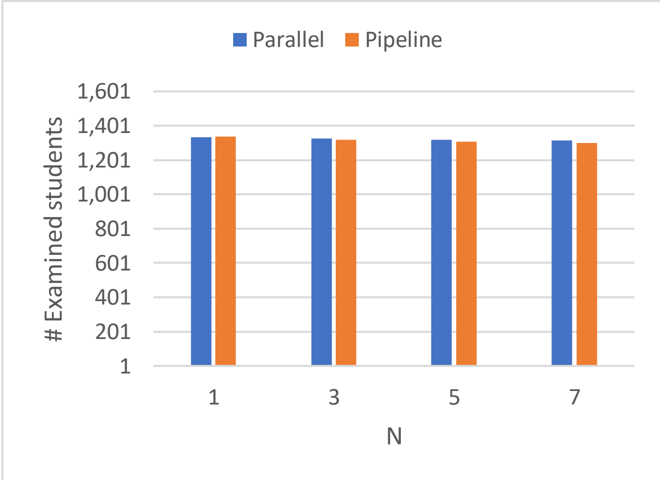

Figure 6.2: Lognormal scenario

A quick analysis can be done for this metric. The graphs clearly show that the number of examined students doesn't depend neither on the chosen examination method (*Parallel* or *Pipelined*), nor on the chosen distribution (*uniform* or *lognormal*). Moreover, in the *Pipeline* scenario, by increasing the value of N to very unrealistic values the average number of examined students tends to greatly decrease: e.g., for $N = 500$, in the *Pipeline-Lognormal* scenario, it reaches an average value of 692.

The average value depends also, of course, on the chosen *Simulation Time*.

The confidence levels are, in both graphs, too small to be visible: they can be consulted here.

Parallel method	1.333,29 ± 2,26	1.333,86 ± 1,34	1.332,74 ± 1,04	1.333,06 ± 0,87
Pipeline method	1.337,26 ± 4,03	1.332,91 ± 1,98	1.333,23 ± 1,31	1.333,23 ± 3,03

Table 7.1: Confidence levels for the Number of examined students in a uniform scenario

Parallel method	1.333,29 ± 2,26	1.324,94 ± 1,86	1.319,74 ± 1,93	1.314,00 ± 1,83
Pipeline method	1.337,26 ± 4,03	1.317,60 ± 3,65	1.308,74 ± 3,21	1.300,69 ± 2,69

Table 7.2: Confidence levels for the Number of examined students in a lognormal scenario

Waiting Time

The *Waiting Time* metric is only analyzed for the *Pipeline* execution mode, since students never have to wait when examined in the *Parallel* execution mode.

Moreover, only scenarios employing a number of teachers $N > 1$ are considered. *Waiting Time* is always null when the exam committee is only composed of a single teacher.

An additional fact concerning the structure of the model has to be addressed. The exam committee can be seen as a queueing network composed of N teachers (service centers). Let's consider the first of them: he will receive a new student as soon as the previous one has answered his question, and he will start examining that new student. This behavior means that his arrival rate of students (let's call it λ) and his service time (let's call it μ) are equal. Moreover, the first teacher will always have an empty queue. After the first answer, students will queue up in the other teachers' queues. This means that the first teacher's *utilization* will always be $\rho = \frac{\lambda}{\mu} = 1$. This and the fact that each teacher picks the different *Answer Times* from the same distribution make the system unstable: it may happen that once in a while a student will need a very large *Answer Time* to answer a given teacher's question. This can potentially make the latter's queue increase very heavily, depending on how much time other students need to answer the previous teacher's question. This behavior makes the *Waiting Time* metric unbounded, as it can grow very large depending on how large the simulation window is set to: this behavior is confirmed by Figure 7.1:

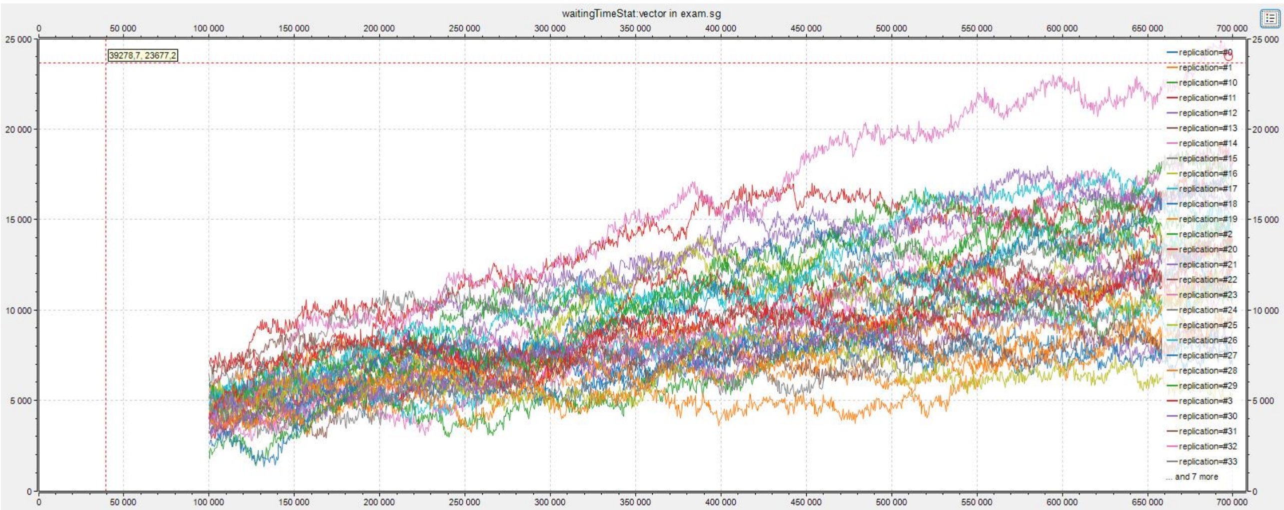

Figure 7.1: Waiting Time metric

The variability of the *Waiting Time* metric is evaluated by using Lorenz Curves, in the uniform scenario and in the lognormal one as well.

Uniform scenario

Lorenz Curves for the *Waiting Time* in the pipelined examination mode employing a uniform distribution of answer times are visible in Figure 7.2:

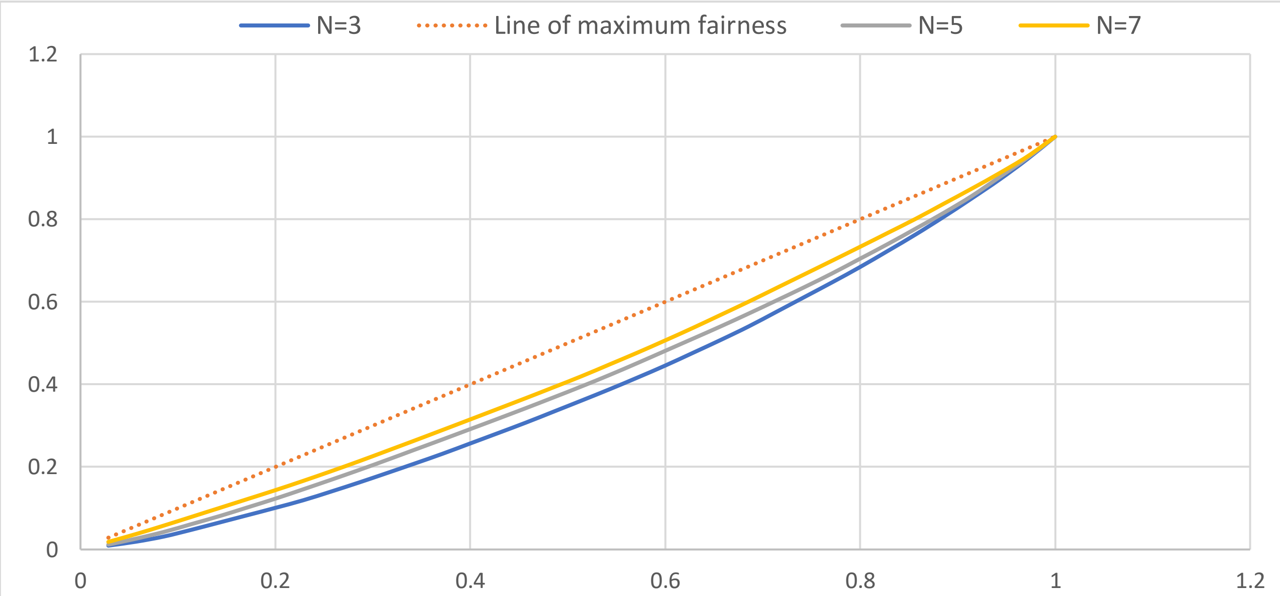

Figure 7.2: Lorenz curves for N=3,5,7, uniform scenario, pipelined method

The graph shows that as the number of teachers in the exam committee (N) increases, the *Waiting Time* tends to become fairer, and students will, on average, have to wait for a much less variable amount of time.

Lognormal scenario

The same reasoning can be done for the lognormal answer time scenario. The Lorenz Curves are shown in Figure 7.3:

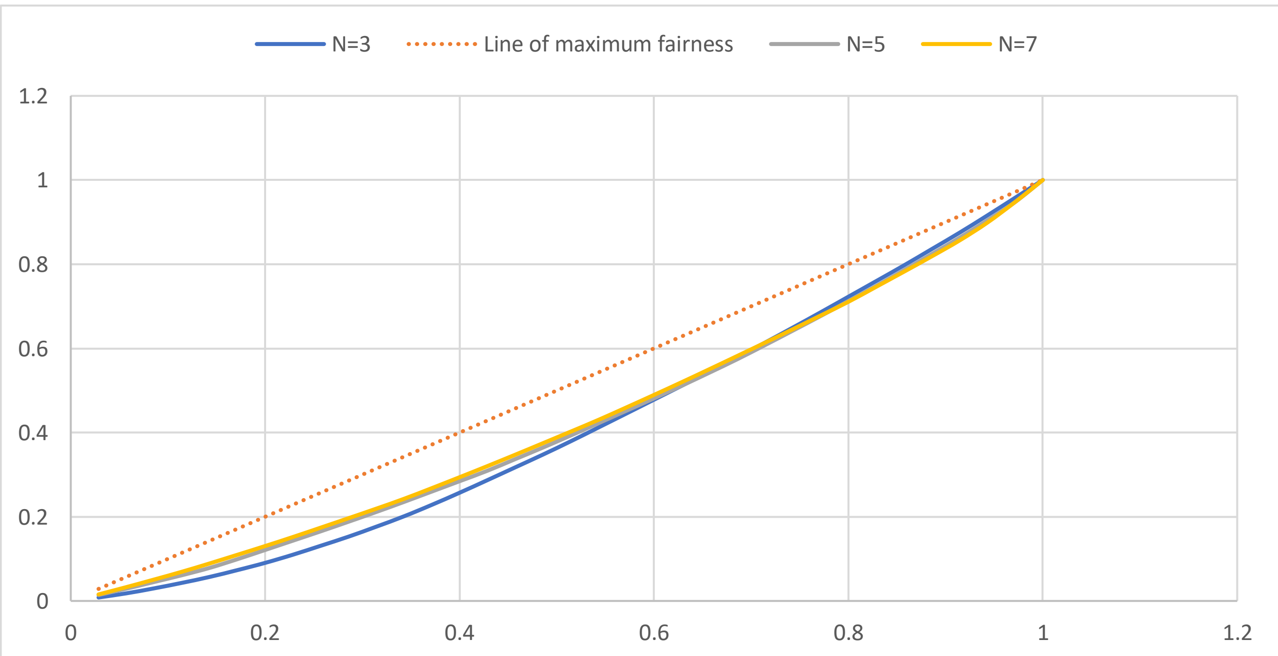

Figure 7.3: Lorenz curves for N=3,5,7, lognormal scenario, pipelined method

Idle Time

A professor's *Idle Time* consists of the sum of the intervals of time in which he is not examining a student (hence, in which he's being idle). Moreover, it's only analyzed for the *Pipeline* execution mode, since when employing a *Parallel* method, a teacher can never be idle.

The data has been collected and aggregated in the following way:

- A total of $N \cdot 35$ values of *Idle Time* (35 for each professor) have been gathered from the simulation's output
- For each repetition, each professor's *Idle Time* values have been aggregated by averaging them, to obtain a total of 35 values
- The obtained values have once again been aggregated to obtain $\bar{X}$ (the sample mean)

The above computations are repeated for $N = 1, 3, 5, 7$.

Figure 8 shows a comparison between the computed mean *Idle Time* ($\bar{X}$) in function of the dimension of the exam committee, in the uniform and in the lognormal scenarios:

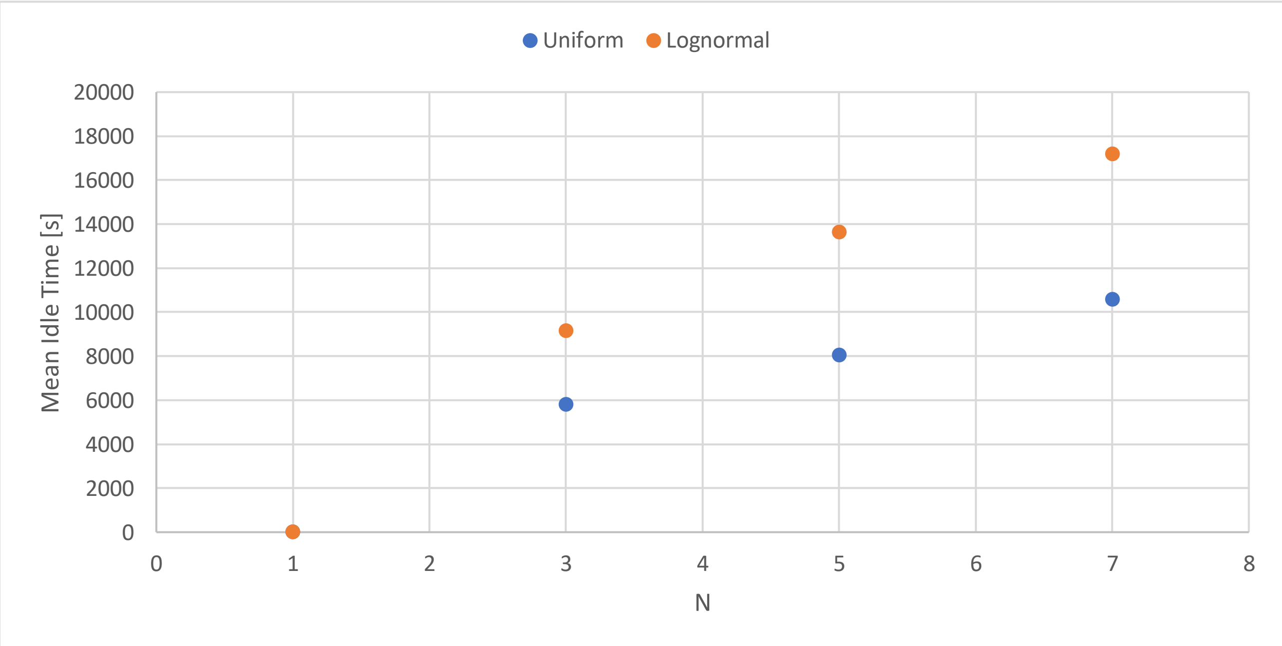

Figure 8: Mean Idle Time comparison in the different scenarios

The collected values turn out to be unbounded (the sample variance keeps growing as the sample width increases), hence the confidence intervals for the sample mean cannot be computed. Anyway, by keep on collecting other samples for increasing values of N, the trend follows the one shown in the above graphs.

The graph clearly shows that, by increasing the dimension of the exam committee, the mean *Idle Time* increases, and it's always higher in the lognormal scenario than in the uniform one (except, of course, in the case $N = 1$). This is because of the reasons explained in the previous paragraphs.

One possible explanation for this phenomenon is that, by increasing N, the probability for a student to experience queueing increases. Assuming that each professor is identified by an id P_i, $1 \leq i \leq 7$, let's consider the following scenario: professors P_1, P_2, P_3 all take an above-the-mean time to examine a student. This time can be enough for the remaining professors to empty their queue (if they have a non-empty one) and to remain idle for the residual interval of time.

The mean *Idle Time* in the different scenarios can be put in relation to the total *Simulation Time*, to obtain the percentage of time in which the exam committee has remained idle. With a *Simulation Time* of 700.000s and a *Warm-Up Time* interval of 100.000s, the results for the different distributions are shown in Table 9.1 and Table 9.2:

N	***Idle Time percentage***
1	0,000 %
3	0,969 %
5	1,342 %
7	1,763 %

Table 9.1: Uniform scenario

N	***Idle Time percentage***
1	0,000 %
3	1,526 %
5	2,273 %
7	2,863 %

Table 9.2: Lognormal scenario

An additional analysis can be done considering the average *Idle Time* of each professor. Data is shown in Figure 9, and, for brevity purposes, only the case for $N = 7$ is considered (confidence intervals cannot be computed for the same reason as before):

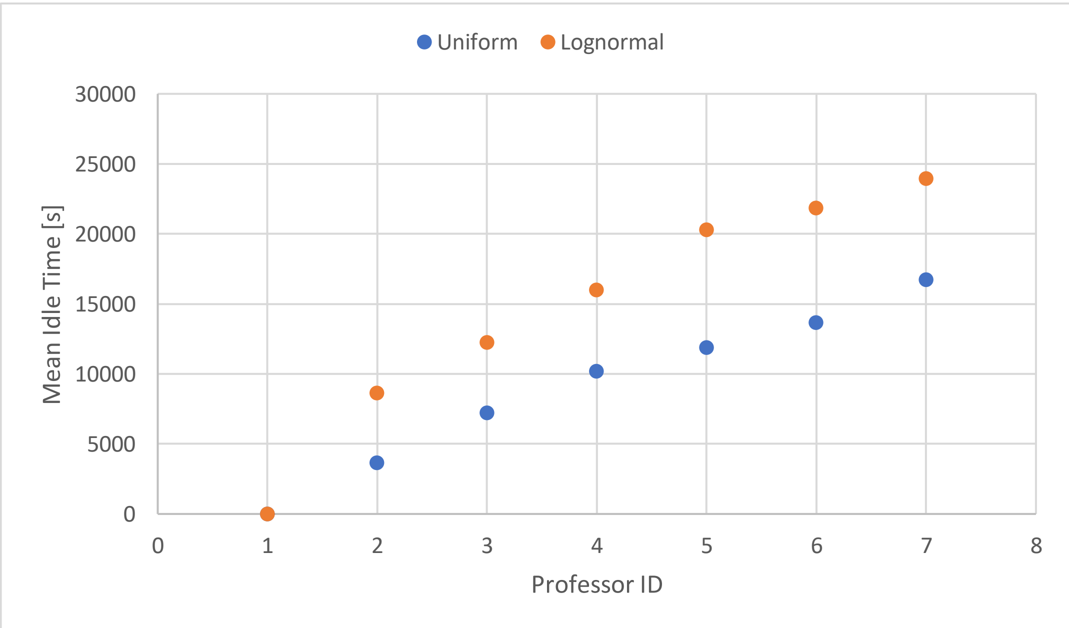

Figure 9: Mean Idle Time of each professor, case N = 7

Once again, it can be seen that the *Idle Times* are higher when employing the lognormal distribution. Moreover, Figure 9 also shows that, by moving forward in the pipeline, the mean *Idle Time* of teachers tends to increase. This is yet another proof of the phenomenon which has been described before.

The last study consists in the analysis of the *Idle Time* metric by varying the behavior of the first teacher, and then in comparing the system's performance with the previous case. For the sake of brevity, only one but significant enough case is considered: the first teacher of the pipeline will now pick the *Answer Time* t from a different distribution, which has been chosen to be $t \sim U(420, 550)$s. In this way, the distribution's mean value will be higher than the previous one, and again, this will only be valid for the *first* teacher.

Data has been collected for the described scenario, and has been plotted in Figure 10:

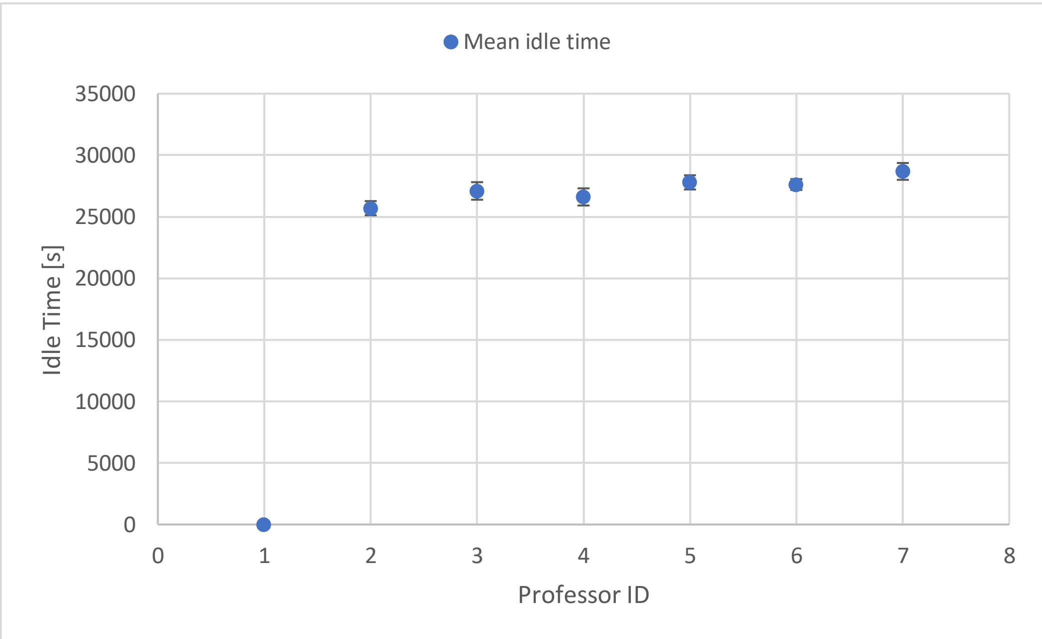

Figure 10: Mean Idle Time of professors from 1 to 7, in a Pipelined uniform scenario

The graph shows that, by varying the first teacher's mean value of *Answer Time*, which was shifted from 450s to 485s, the system changes its behavior: in the previous analysis, as the *Professor ID* was increasing, his/her *Mean Idle Time* was increasing as well (Figure 9).

In this scenario, instead, teachers' *Mean Idle Time* is bounded (for professor IDs from 2 to 7) and is independent of his/her position in the pipeline: teachers tend to remain idle for the same amount of time. The latter is, however, higher in this scenario than in the previously analyzed ones.

This little change in the first teacher's *Answer Time* uniform interval reflects also in the *Total Examination Time* metric's behavior, as shown in Figure 11 (recall that $Total\ Examination\ Time = Total\ Answer\ Time + Waiting\ Time$):

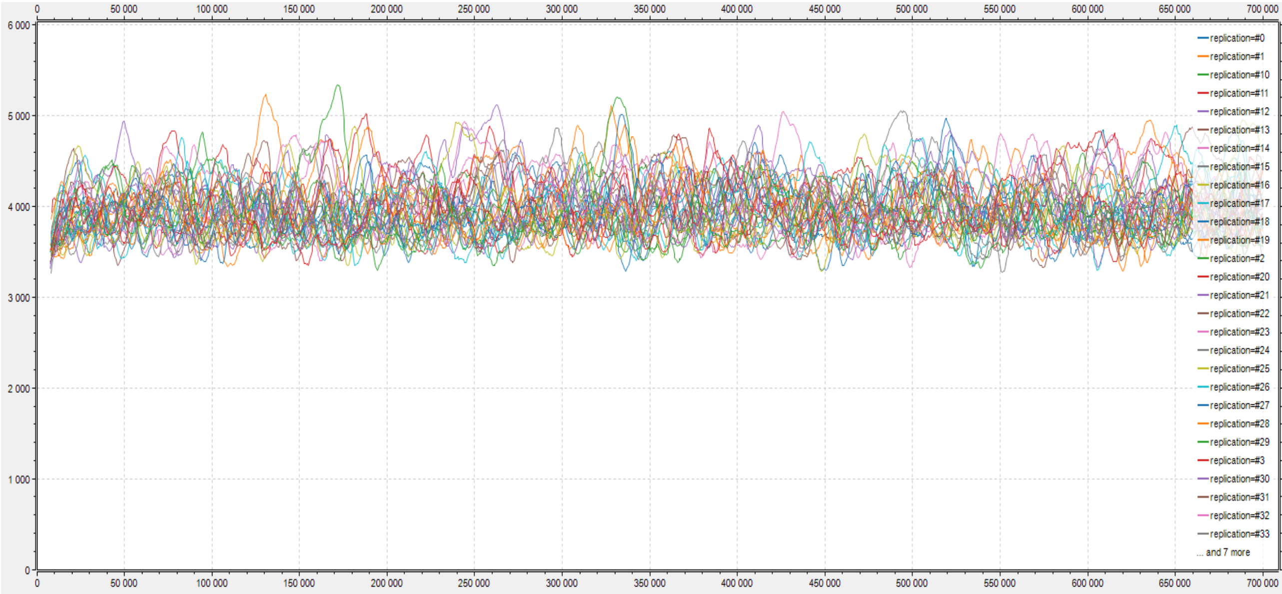

Figure 11: Total Examination Time of 35 repetitions, pipeline method, uniform scenario, with the above-described change

Since the first professor is responsible for "injecting" a new student (job) in the rest of the pipeline, this gives a viable solution to correct the system's instability: increasing the mean *Answer Time* at the first professor only (i.e., reducing the interarrival rate between two consecutive students in the pipeline). In this way there will still be a student that needs to be examined, waiting for the previous one to answer his first question, but the first professor will "inject" him in the rest of committee with a lower speed.

Final considerations

Examination Time

The analysis has revealed that a student's *Examination Time* increases when the number of professors in the exam committee increases, but it's always lower in the *Parallel* examination method. This is explained of course by the fact that, as N increases, the number of questions that need to be answered increases as well, and because by switching to the *Pipeline* method some queueing time is introduced. The latter makes the *Waiting Time* of a student become relevant, when analyzing the *Pipeline* method.

When instead talking about distributions of answer times, the choice was between a *Uniform distribution scenario* and a *Lognormal distribution* one (for both, parameters have been discussed in the "Calibration" section). Data shows that, because of the higher variability of the lognormal one, the total *Examination Time* tends to be higher as well when employing this kind of distribution.

Number of examined students

Graphs look almost identical in the *Parallel* and *Lognormal* scenarios for both examination methods (for the considered values of N). This means that the *Number of Examined Students* will not depend neither on the chosen examination mode, nor on the chosen scenario: only the duration of the simulation run will have an impact on the metric, since more *Simulation Time* would mean more opportunities for other students to finish their exam. These considerations are valid until a low-enough value for N is employed: as this parameter increases, the effects of queueing are not negligible anymore, and the *Number of Examined Students* tends to decrease in the *Pipeline* examination mode (by maintaining the same *Simulation Time*).

Waiting Time

The *Waiting Time* analytic is affected by the problem of high utilization of the service centers (professors). As a result of this, its values are not bounded as explained in the relative paragraph.

Idle Time

From the first analysis done for this metric, we can see that the *Idle Time* values for each professor are not so high. For example, the maximum value for the *Lognormal* scenario is only 2.86% of the duration of the whole simulation. This translates in about 5.5 hours in an (hypothetical) exam session of 8 days.
The values are very low: this is a clear demonstration of the high utilization of each service center of the network (the professors), and another proof of the system's instability. Moreover, the *Idle Time* of the first professor is always 0 (i.e., it has a 100% of utilization), and this is not good for the

rest of the network: this has been proven by increasing his/her mean *Answer Time,* as described in the dedicated section.