Automated Assignment of Lab Assistants to Student Presentations at KTH Lab Sessions

JACOB SCHOERNER
MIGUEL MÜLLER
Automated Assignment of Lab Assistants to Student Presentations at KTH Lab Sessions

MIGUEL MÜLLER
JACOB SCHOERNER

Computer Science
Date: June 5, 2018
Supervisor: Pawel Herman
Examiner: Örjan Ekeberg
Swedish title: Automatisk tilldelning av labbassistenter till studentpresentationer under KTH-labbsessioner
School of Electrical Engineering and Computer Science
Abstract

The authors study the case of lab presentations at KTH. Currently, lab assistants follow the First In First Out principle when deciding what student presentation to visit first. The authors study whether the Shortest Job Next discipline can give a significant advantage in presentation throughput. They create a simulation of a lab session, and simulate sessions with both SJN and FIFO, and varying values of time per presentation, number of assistants and number of lab rooms in the session. It is shown that Shortest Job Next gives a consistently higher number of presentations, but also on average a longer maximum waiting time. The effect on number of handled presentations is the highest when the time per presentation is low and the number of rooms is high.
Sammanfattning

Författarna studerar labbpresentationer vid KTH. I nuläget baserar sig labbassistenter på First In First Out-principen när de bestämmer vilken ordning de ska besöka de olika studentgrupperna som har markerat sig som redo att presentera. Författarna undersöker huruvida Shortest Job Next-principen kan ge ett betydande övertag i antal presentationer per tidsenhet. De skapar en simulation av ett labbtillfälle och testar simulationen mot både FIFO och SJN, under ett antal olika värden på tid per presentation, antal tillgängliga övningsassistenter och antal rum i labbsessionen. Det visar sig att Shortest Job Next ger ett konsekvent högre antal presentationer per tidsenhet, men att den också medför en risk att vissa grupper får vänta väldigt länge på att presentera. Effekten på antal avklarade presentationer är som störst när tiden per presentation är låg och antalet rum är stort.
## Contents

1 Introduction ................................................. 1  
  1.1 Research Question ........................................ 1  
  1.2 Scope ..................................................... 2  
  1.3 Outline .................................................... 2  

2 Background ................................. 3  
  2.1 KTH Lab Sessions ........................................ 3  
  2.2 Queueing Theory ........................................ 3  
      2.2.1 Service Disciplines .................................. 4  
  2.3 Simulation ................................................. 4  
  2.4 Related Work .............................................. 5  
      2.4.1 University Timetabling ...................... 5  
      2.4.2 Similar Applications in Other Fields ........ 6  

3 Method ............................................. 7  
  3.1 Data Gathering .......................................... 7  
  3.2 Model ................................................... 8  
      3.2.1 Queue ................................................ 8  
      3.2.2 Location and Walking Distance ............ 8  
      3.2.3 Parameters ......................................... 9  
  3.3 Simulator ............................................... 10  
      3.3.1 Queue ................................................. 10  
  3.4 ANOVA .................................................. 11  

4 Results ....................................... 13  
  4.1 Raw output data .......................................... 13  
  4.2 Statistical analysis .................................... 17  
      4.2.1 Percentual improvement ..................... 17  
      4.2.2 Applying ANOVA ................................. 17
Chapter 1

Introduction

A characteristic feature of the Royal Institute of Technology (KTH) Computer Science undergraduate studies is the lab session. Many lab sessions employ presentation queues, where students wait in line to present their lab work and be graded.

However, the queue can become long. Many student pairs enter the session having already completed the task on their own time, and immediately enter the queue. Even this may not be a guarantee for actually being able to perform a presentation, as there may simply be too many presentations for the limited number of lab assistants to handle them all in the limited amount of time. Thus, lab sessions can be stressful to both students, lab assistants, and the course supervisors.

To combat this problem, we have studied an automated system for assigning assistants to presentations. Such a system would streamline the assistants’ work flows, and hopefully lead to a higher throughput of student presentations per lab assistant per unit of time by optimizing the assistant assignment.

1.1 Research Question

In the case of KTH lab sessions, can performance in assignment of lab assistants to various student groups and their presentations, with respect to the student throughput (i.e., how many student presentations can be performed in a given amount of time), be significantly improved by using the Shortest Job Next service discipline, instead of the currently used First In First Out?
1.2 Scope

The principal goal of the project is to maximize throughput of student presentations. While we provide output data on the longest waiting times that individual students may face, we do not attempt to minimize these waiting times, as that is already the way the system works in practice.

We do not attempt to implement the NP-hard problem of scheduling assistants for maximal throughput with all the active assistants as variables. Instead, our system, when assigning an assistant, takes into account only the best use of that particular assistant given the current situation. In practice, this means that we use a greedy algorithm.

1.3 Outline

The report starts with the “Introduction”, introducing the reader to the general idea of the project. The introduction also contains the formal research question, a section on the scope and delimitations of the project, and this outline. Then follows “Background”, which gives a background information about the specific case of KTH lab sessions, about the mathematics behind Queuing theory (which is the mathematical framework that fits closest to the problem) and about other applications which have tackled similar problems in other fields using similar methods.

A “Method”-section accounts for the actual methodology of the project, both regarding the data gathering phase, the simulation model, and the data analysis. The results are presented and analysed in “Results”. In “Discussion”, the results are examined from various perspectives, and finally everything is summed up in “Conclusion”.
Chapter 2

Background

2.1 KTH Lab Sessions

A lab session is a form of examination in which students are tasked to solve some problem. Students typically work in pairs or larger groups. The students finally present their work, and receive a grading based on their performance.[5]

As the number of students on a course may be vast, and time limited, the students generally do not make their presentations for the course supervisor, but rather, to one of the so-called “övningsassistenten”, or lab assistants.[3] The number of lab assistants vary from course to course.

Upon completing their task, a student group reports to an online digital queue. The lab assistants check the queue, and one of them goes to the students at their computer desk and handles the presentation.[4] Students are served on a First In, First Out basis.

2.2 Queueing Theory

Mathematically, the problem in question falls under the field of queueing theory. Queuing theory deals with situations in which a number of customers attempt to visit a queuing node. At a node awaits a number of servers, and each customer must be assigned to a server, who is then occupied for a specified duration, called the size of the job.

Notationally, queues may be described as variations of “A/S/c”, where A represents the pattern of arrival of jobs (ie, does arrival time
follow a Poisson distribution, or some other statistical pattern? \( S \) represents the size of jobs, and \( c \) represents the number of servers. In this notation, the KTH lab session queue may be described as “\( M/G/k \)”, or Markovian (indicating random arrival time distribution), General size distribution (indicating independent service time) \( k \)-number of servers (indicating a fixed and limited number of available servers. [2]

Queuing theory supplies a number of possible algorithms, or Service Disciplines. [10] Among them are some which can be disregarded as they are not applicable in the case of KTH lab queues (such as Round Robin’, which requires that jobs may be interrupted at regular intervals, and then resumed, possibly by a different server, which for obvious reasons is inconvenient).

However, others map quite naturally, such as First In First Out (i.e., the student that first entered the queue is served first), SIRO (the lab assistants are assigned randomly) or Shortest Job First, which can be implemented differently depending on the specifics of how the KTH queue is modeled.

### 2.2.1 Service Disciplines

**First In, First Out**

First in, first out (FIFO) resembles a regular everyday queue; customers are served in the order that they arrived.[2] FIFO minimizes the longest waiting time by always providing service to the customer that has waited the longest.[8]

**Shortest Job Next**

In Shortest Job Next (SJN), customers are served in the order of the length of their service times. The customer with the shortest service time is served first, and so on. This optimizes throughput of customers, but can lead to long waiting times for especially long jobs as these are constantly passed over in favour of shorter jobs.[8]

### 2.3 Simulation

A simulation system is often used to study a real-time system that currently does not exist. More precisely, the simulation of a real-time sys-
tem is used to study the performance of said system and how it depends on the values of its input parameters. When simulating a real-time system, one has to model coherent parts of the system at various levels of detail. To build an accurate model, the basic events whose occurrence will alter the status of the system must be identified. For example, in a system where one repairman handles a queue of broken down machines, one of the status variables may be the number of broken down machines. The events that will directly affect the number of broken down machines will in this case be when a machine breaks down and when a machine gets fixed. In order to handle the basic events in our simulation, we need to keep track of the time when an event occurs. This is done with a set of variables called clocks. In addition to the clocks associated with the occurrence of an event, a master clock keeps track of the simulated time.

A simulator can advance the time in various ways. If a simulator is utilizing the event-advance design, the simulator decides which event occurs next, and the master clock is advanced to that time instant. In unit-time advance design, the master clock is advanced in fixed time units. This means that each time the master clock advances, the simulator has to check if any event is scheduled to occur.[11]

2.4 Related Work

2.4.1 University Timetabling

The problem of scheduling in a school/university context has been heavily studied from the perspective of timetabling, i.e., from a position of scheduling lectures into time slots based on various constraints, such as teacher availability, room availability, and so on.[9][1][13] However, the specific situation of queueing student presentations at a lab session seems to be unexplored. Seeing as different universities might have other models for lab presentations, it is possible that this specific application has not been extensively studied.

There exists a number of documents that are tangentially related, such as a U.S patent application for a “Student interaction management system”[14], which is described as a digital tool for students to apply for assistance from teachers, where the teachers are assigned to students based on “priority for an affiliate institution that uses the system, tutor subject weight, and student time in queue”. However, the
patent application does not cite any academic publications as sources, and as the connection is vague at best, we have not investigated it further.

2.4.2 Similar Applications in Other Fields

The field of elevator scheduling resembles our problem formulation, with the elevator cars analogous to the lab assistants and the floors analogous to student groups. We have made use of Anton Jansson and Kristoffer Uggl Lingval’s “Elevator Control Using Reinforcement Learning to Select Strategy”[7] for inspiration on methodology.

In “Elevator Control Using Reinforcement Learning to Select Strategy”, the authors seek to create a system for scheduling elevators. They list a number of possible algorithms (Longest Queue First, Zoning, Round-Robin, Up-Peak Group Elevator and Estimated Time of Arrival) for assigning elevators to floors, and make use of reinforcement learning to create a system which chooses the best assignment algorithm based on the circumstances. To train their system, they create a real-time simulation, with Poisson-distributed user arrival times. Ultimately, they find that under certain circumstances, reinforcement learning can have a strong and meaningful effect on customer waiting times within their system.

Another simulation study focusing on optimizing the use of a group of elevators is “Real-Time Optimal Scheduling of a Group of Elevators in a Multi-Story Robotic Fully-Automated Parking Structure” [6]. The simulation used here models the elevator system for commercial multi-storied parking structures during morning rush hours. The customer arrivals are Poisson distributed, and the monitored variables are mean customer waiting time, maximum customer waiting time, and elevator service times.

The field of medicine also makes use of queues. For an instance, in a hospital, the incoming patients may be considered as analogous to our student groups, and the nurses/doctors as analogous to the lab assistants. “A Survey of Queueing Theory Applications in Healthcare”[12] provides a queueing theory model of this situation, and introduces various considerations to take into account.

Information on the relevant Service Disciplines (FIFO and SJN) can be found in queueing theory textbooks and elementary texts, such as “A Survey of Scheduling Rules”.[10]
Chapter 3

Method

We have studied whether an automated, SJN-based assignment system could effectivise lab sessions by simulating a lab environment and testing assignment based on the SJN service discipline. The SJN discipline was chosen because it guarantees the highest possible throughput. The SJN benchmark was compared to FIFO, which represents how lab assistant assignment currently functions. Relevant parameters, such as number of available assistants, average presentation time, number of rooms are varied, and results presented for each variation. The results include:

- The number of presentations handled in a two-hour lab session.
- The longest waiting time, i.e. the longest time a student group has to wait to present after joining the queue.

Finally, we have compiled the results of the simulation to determine under which circumstances, if any, that a change to a SJN discipline over the current FIFO would provide a significant advantage in student throughput.

3.1 Data Gathering

From interviews with course supervisors as well as lab assistants, the following data has been drawn (full answers per course can be found in the appendix):

- How long does each presentation take?
• How many assistants are present at a typical lab session?
• How many rooms are used in a typical lab session?

3.2 Model

To create a model of the lab environment, a number of decisions have to be made. First of all, we decided to model each presentation as taking the same amount of time. In reality, of course, the time per presentation varies, but since the time that a given presentation will take is impossible to know before the presentation, the service discipline cannot take this into account. Thus, a discipline that differentiates between jobs must do so on some other criterion.

In fact, the only two meaningful factors that can be said to differentiate between two presentations, are the physical location of the students holding the presentation, and their place in the queue. This is the only information that is available to the Service Discipline at the time of assignment, and thus, this is the information that the assignment must be based on. Consequentially, in our model, comparing “job time” translates directly to comparing “distance to presentation” - as all presentation times are considered equal, the actual presentation times cancel out, leaving only the distance as a deciding factor when comparing job length.

3.2.1 Queue

We assume that new students enter the queue at the same pace as students leave it, so that the queue is effectively endless (but never contains more than a specified number of students at a time). This is done because this situation - one where there are too many students for the assistants to handle - is the problematic one.

3.2.2 Location and Walking Distance

We model the location of each student group in order to take walking distance into consideration when comparing the different service disciplines.

We do this by assigning a random index between \(-\frac{\text{delta}_{\text{max}}}{2}\) and \(\frac{\text{delta}_{\text{max}}}{2}\) to each student group joining the queue, where \(\text{delta}_{\text{max}}\) is the...
maximum time in seconds it can take for an assistant to walk to a
student group. This means we get a one-dimensional location space
where student groups are placed. Our own measurements show that
it takes approximately one minute to go from one room to the next
in the computer labs (measured in the E-building computer rooms).
Thus:

$$delta_{max} = \text{number of rooms used in lab session} \cdot 60.$$  

Before a lab session starts, all the assistants are placed at index 0, which
is the middle of the location space. This is to resemble the behaviour
of assistants in a real lab session. Consequently, the walking distance
$delta$ between a lab assistant at index $a$ and a student group at index $b$
becomes $delta = |a - b|$

3.2.3 Parameters

We assume session lengths of two hours, as this is the most common
case. With these assumptions, a number of input data remains. These
are:

- How many assistants are available?
- How many rooms are booked for the session? (this determines
  the maximum distance that an assistant may have to walk be-
  tween presentations)
- Average length of a presentation?

These parameters are varied, based on the statistics (gathered as
described under “Statistics”). The boundaries are as outlined below.
The average values are rounded upwards.
Table 3.1: Minimum, average and maximum values for all input parameters

<table>
<thead>
<tr>
<th></th>
<th>Min value</th>
<th>Average value</th>
<th>Max value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of assistants available</td>
<td>4</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Number of rooms available</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Length of a presentation (min)</td>
<td>5</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

3.3 Simulator

The simulator simulates one lab session. Since we are dealing with a real-time system, we need a way to represent time. We do this by counting the number of seconds from the start of the lab session. The simulator consists of a step()-function that simulates one second of the lab session, thus our program can call the step()-function until the simulated lab session ends. This way of advancing the simulation is called unit-time advance.[11] By running the simulation 1000 times for each discipline, we get an average number of simulated presentations for the model.

3.3.1 Queue

In addition to our free variables, the simulator requires a number of other input data, specifically:

- Queue join rate
- Initial queue size
- Session length

Session length can be safely fixed at 2 hours (or 7200 seconds), given that this is the overwhelmingly the most common case. The initial queue size and queue join rate must ensure that the number of students in the queue at any given time is higher than the number of
available assistants (given that these are the lab cases that we are interested in - if the assistants were to outnumber the students, the session would be unproblematic). However, they must also limit the number of students in the queue at any given time. Should the number become too high, the students will fill the entire lab area, creating a situation in which every single study spot is occupied by a student group. In such a situation, a lab assistant following SJN will always receive a travel distance of 0 to their next assignment, skewing the results. To ensure reasonable results, we set the initial queue size to three times the number of assistants:

\[ \text{initial queue size} = \text{number of assistants} \cdot 3 \]

This is also reasonable from a course planning perspective; it is likely that course supervisors will seek more lab assistants for courses with large amounts of students enrolled. The constant 3 is an arbitrary choice, but we have found that it gives reasonable numbers, given our input data (a range of 12 to 21 student groups in the queue). The queue join rate is set so that the number of students in the queue at any given time will be approximately the same. This, again, ensures that the queue will be neither overpopulated nor emptied.

\[ \text{queue join rate} = \frac{1}{\text{time per presentation} + \text{average walking time} \cdot \text{number of assistants}} \]

Note that this is the average queue join frequency - the chance, for any given second, that a new student will join the queue that particular second. Since our simulator is designed with unit-time advance design, this can directly be used in the step()-function and will ensure independency between different student groups. The only issue here is that two student groups can’t join the queue during the same time step. However, since the time step is only one second, this is a minor issue that should not have any significant effect on the simulation. In event-advance design, a Poisson-distributed probability function would be required.

### 3.4 ANOVA

To ensure the statistical significance of our results, we test them with the ANOVA (Analysis of variance) method. So-called “One way ANOVA”
measures how altering one variable, called the explanatory variable, affects a response variable. The response variable is measured at various levels of the explanatory variable. The variability of the results inside each level is measured, as well as the variability of the levels themselves. The numbers are then compared. If the variability of the levels themselves exceeds the variability of the results inside the levels to a certain extent, there is reason to believe that there is a correlation between the explanatory variable and the response variable.

Mathematically, this can be briefly described as follows. We define variability between levels as:

\[ SSG = \sum_{i=1}^{k} n_i \cdot (\bar{x}_i - \bar{x})^2 \]

where \( k \) is the number of levels of the explanatory variable, \( n_i \) the number of observations of the response variable at explanatory variable level \( i \), \( \bar{x}_i \) the mean of the response value for observations at level \( i \) and \( \bar{x} \) the mean of the response variable across all observations.

We define variability within levels as:

\[ SSE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2 \]

We divide each number by their respective degrees of freedom, and finally receive an output value (called the \( F \) value) according to:

\[ F = \frac{MSG}{MSE} = \frac{(SSG)}{k-1} \frac{(SSE)}{(n-k)} \]
Chapter 4

Results

4.1 Raw output data

Running the simulation as described under "Method" generates the following output data:

Table 4.1: Output data for varying values of number of assistants. Number of rooms and presentation time kept at average.

<table>
<thead>
<tr>
<th>Number of assistants</th>
<th>Average number of presentations (FIFO)</th>
<th>Average number of presentations (SJN)</th>
<th>Average longest wait time (FIFO) (minutes)</th>
<th>Average longest wait time (SJN) (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>45.4</td>
<td>47.9</td>
<td>43.3</td>
<td>82.6</td>
</tr>
<tr>
<td>6</td>
<td>68.3</td>
<td>71.9</td>
<td>42.5</td>
<td>89.4</td>
</tr>
<tr>
<td>7</td>
<td>79.7</td>
<td>84.0</td>
<td>42.6</td>
<td>91.7</td>
</tr>
</tbody>
</table>
Figure 4.1: The relation between the number of assistants and the number of handled presentations.
Table 4.2: Output data for varying values of number of rooms. Number of assistants and presentation time kept at average.

<table>
<thead>
<tr>
<th>Number of rooms</th>
<th>Average number of presentations (FIFO)</th>
<th>Average number of presentations (SJN)</th>
<th>Average longest wait time (FIFO) (minutes)</th>
<th>Average longest wait time (SJN) (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>71.6</td>
<td>72.0</td>
<td>40.1</td>
<td>87.6</td>
</tr>
<tr>
<td>3</td>
<td>68.3</td>
<td>71.9</td>
<td>42.5</td>
<td>89.4</td>
</tr>
<tr>
<td>4</td>
<td>66.4</td>
<td>71.9</td>
<td>44.5</td>
<td>90.1</td>
</tr>
</tbody>
</table>

Figure 4.2: The relation between the number of rooms and the number of handled presentations.
Table 4.3: Output data for varying values of presentation time. Number of assistants and number of rooms kept at average.

<table>
<thead>
<tr>
<th>Presentation time (minutes)</th>
<th>Average number of presentations (FIFO)</th>
<th>Average number of presentations (SJN)</th>
<th>Average longest wait time (FIFO) (minutes)</th>
<th>Average longest wait time (SJN) (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>123.2</td>
<td>141.5</td>
<td>35.1</td>
<td>69.3</td>
</tr>
<tr>
<td>10</td>
<td>68.3</td>
<td>71.9</td>
<td>42.5</td>
<td>89.4</td>
</tr>
<tr>
<td>15</td>
<td>48.0</td>
<td>48.0</td>
<td>54.1</td>
<td>93.6</td>
</tr>
</tbody>
</table>

Figure 4.3: The relation between the presentation time and the number of handled presentations.
4.2 Statistical analysis

4.2.1 Percentual improvement

One way to gain an intuitive understanding of the results is to relate the difference in number of handled presentations, to the number of presentations overall. This can be done with the formula:

\[ p = \frac{\text{number of presentations}_{\text{SJN}} - \text{number of presentations}_{\text{FIFO}}}{\text{number of presentations}_{\text{FIFO}}} \]

That is, “for a certain input data, how many more presentations can SJN handle per presentation that FIFO handles”. This gives the following results:

Table 4.4: Percentual improvement of SJN over FIFO for different levels of the input parameters

<table>
<thead>
<tr>
<th>Number of assistants</th>
<th>Number of rooms</th>
<th>Presentation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>5.3%</td>
<td>0.2%</td>
</tr>
<tr>
<td>Average</td>
<td>5.3%</td>
<td>5.3%</td>
</tr>
<tr>
<td>High</td>
<td>5.4%</td>
<td>8.3%</td>
</tr>
</tbody>
</table>

4.2.2 Applying ANOVA

In our case, the explanatory variable of ANOVA represents the service discipline used, and has two potential values: FIFO or SJN. Our response variable is the number of presentations handled in a two-hour session. Since we have 1000 observations of each level of the explanatory variable, our F-values will likely be very high (if a correlation exists).

Applying ANOVA as described above generates the following F-values for the different values of each parameter (with the other two parameters kept at average):
Table 4.5: Anova F-values for various values of the input parameters

<table>
<thead>
<tr>
<th></th>
<th>Number of assistants</th>
<th>Number of rooms</th>
<th>Presentation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>3391.1</td>
<td>197.1</td>
<td>35943.9</td>
</tr>
<tr>
<td>Average</td>
<td>7926.7</td>
<td>7926.7</td>
<td>7926.7</td>
</tr>
<tr>
<td>High</td>
<td>10752.5</td>
<td>42769.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>

The table value for F having ~1000 degrees of freedom in the denominator and one in the numerator, is 3.8508 for certainty of $p = 0.05$ [15]. Thus, our result is statistically significant in all cases except for the highest value on the presentation time parameter.
Chapter 5

Discussion

Under all parameter configurations, SJN gives a higher throughput. The effect increases when the number of rooms increase, and when the presentation time decreases. This is expected - more rooms means higher possible walking distances, which leads to an advantage for SJN. With a high presentation time, the walking distances become more and more insignificant compared to the actual presentation time, and the difference between SJN and FIFO decreases. With a low presentation time, however, SJN can handle almost one more presentation per 5 FIFO presentations, certainly a significant advantage. The correlation with number of assistants is much, much fainter. From this data, it seems likely that the strongest effect should be found with a maximum number of rooms and a minimum presentation time.

The ANOVA analysis shows that the advantage of SJN over FIFO in throughput, for all parameter configurations - except high presentation time - is statistically significant.

When comparing how long a student group has to wait in order to present, FIFO always outperforms SJN by a significant margin for every configuration of the simulator. This is the trade-off that is made; increased throughput means less “fairness”. The average longest waiting time was approximately doubled when switching from FIFO to SJN, which makes SJN less attractive in practice.

Further studies could try some sort of combination of FIFO and SJN to ensure a somewhat fair assignment system with higher throughput than FIFO.
5.1 Potential sources of error

Our way of representing location in combination with our way of making new student groups join the queue makes it possible for several student groups to have the same location in our one-dimensional space. This may not be an accurate depiction of reality, since it makes it possible for dozens of student groups to be in the same place. It is possible that this aspect of our model may have a positive effect on the throughput, especially for SJN since an assistant in that kind of location will only have to walk when there are no student groups left in that location. In FIFO, it’s much less probable for an assistant to be assigned a student in the same location as the last.

Another potential error, which relates to the one above, is how we decide the size of the initial queue. The initial queue size was set to a value of $3 \cdot \text{number of assistants}$. A larger (initial) queue would cover more space, which would benefit SJN. This makes it relevant to find out how the queue size affects our results. Since there are many factors that affect the initial queue size in practice, it is not clear how this should be modelled. Even if it’s modelled after a normal distributed probability function, deciding an accurate mean would be hard.

If one was to implement SJN for KTH lab sessions, students would very likely adapt in some way or another. This has not been taken into account in our study. For example, our model assumes that student groups will join the queue at the same rate no matter which service discipline is used. In addition, our model always assumes students will choose a random location in one of the available rooms, even if SJN is used. Therefore, one should interpret our results with care and not assume the same results would be seen in practice.
Chapter 6

Conclusion

The problem statement of this thesis asked the question “Can student throughput in lab sessions be significantly improved by using the Shortest Job Next service discipline instead of First In First Out?”. The results show that SJN outperforms FIFO in almost every use case, with respect to throughput. Only in labs with a high average time per presentation, does the difference become negligible. There are, of course, other factors to take into account for course supervisors deciding what policy to use in their labs. The results also show that waiting times are negatively affected by switching to SJN, which may in turn have the effect of demotivating students to even show up to the labs, or simply be viewed as "unfair". With this said, the results certainly confirm the hypothesis of the problem statement.
Bibliography


# Appendix A

## Course statistics

<table>
<thead>
<tr>
<th></th>
<th>DD1389 Inter-netprogram-mering</th>
<th>DD2350 Algo-ritmer, data-strukturer och komplexitet</th>
<th>SF1514 Nu-meriska Metoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of lab assistants</td>
<td>6.3</td>
<td>7</td>
<td>4-6</td>
</tr>
<tr>
<td>Number of rooms</td>
<td>1.7</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Presentation time (minutes)</td>
<td>8</td>
<td>5-10</td>
<td>15</td>
</tr>
</tbody>
</table>

**Table A.1:** Information gathered from course supervisors in various lab courses