CPU Scheduling Report

This report investigates aspects of CPU scheduling algorithms. In this report, there are 3 experiments targeting different areas of focus:

- Experiment 1 – The effects of modifying the Time Quantum on turnaround time.
- Experiment 2 – The effects that high workload has on response times.
- Experiment 3 – Whether the First-Come First-Served scheduler out performs the Shortest Job First Scheduler in a prolonged period.

These experiments are outlined with their results and analysis below. By the end of this report, the report provides an insight into how the aspects explored in the experiments affect CPU scheduling alhgorithms and understand why this is.

The results and inputs are organised into experiment folders and then test folders for each test in an experiment, then seed folders if needed. These files are numbered sequentially for each test e.g. output1.out output2.out and so on. This is the same for parameters and inputs. This to allow ease of reading for what file you need to find. The graphs were produced in Jupyter Lab, using the pandas plugin.

Contents

Experiment 1

Objectives

To understand how when the Time Quantum is changed for Round Robin and Feedback Round Robin Schedulers, how this changes the turnaround time of the algorithms.

Turnaround time is the amount of time between submission and completion of a process. The Time Quantum is the amount of time allocated to each process for it to execute, before switching to another process if available.

Therefore, here the amount of processes executed is staying the same, along with all mean times and the seed number, for each set of tests. This is to increase the scope of the experiment and to eliminate anomolies.

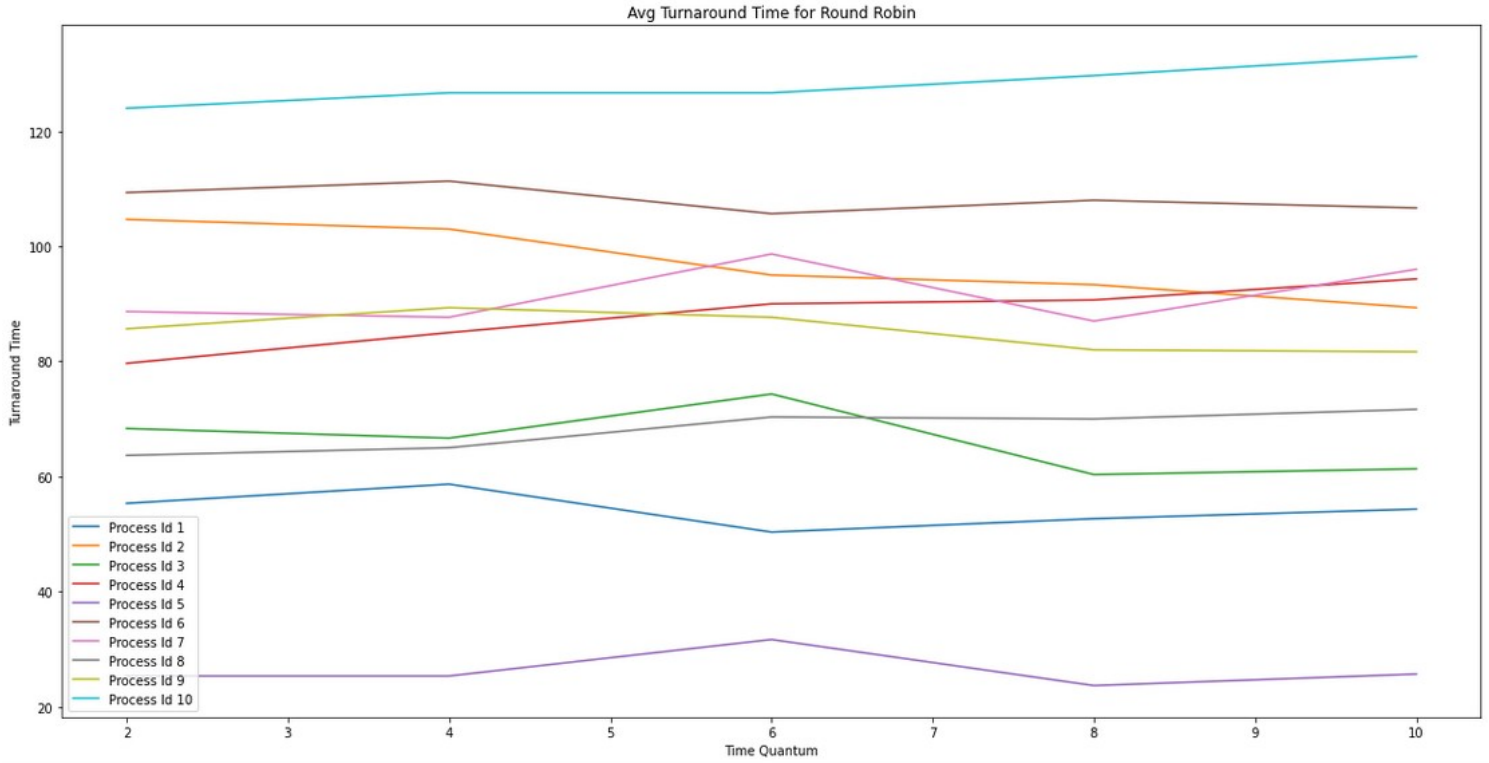
The prediction is that there is an optimal Time Quantum, for each set of data, and that the graphs will show a Gaussian (Normal) distribution, where the optimal Time Quantum is at the peak with the other values plateauing, to show worse execution times.

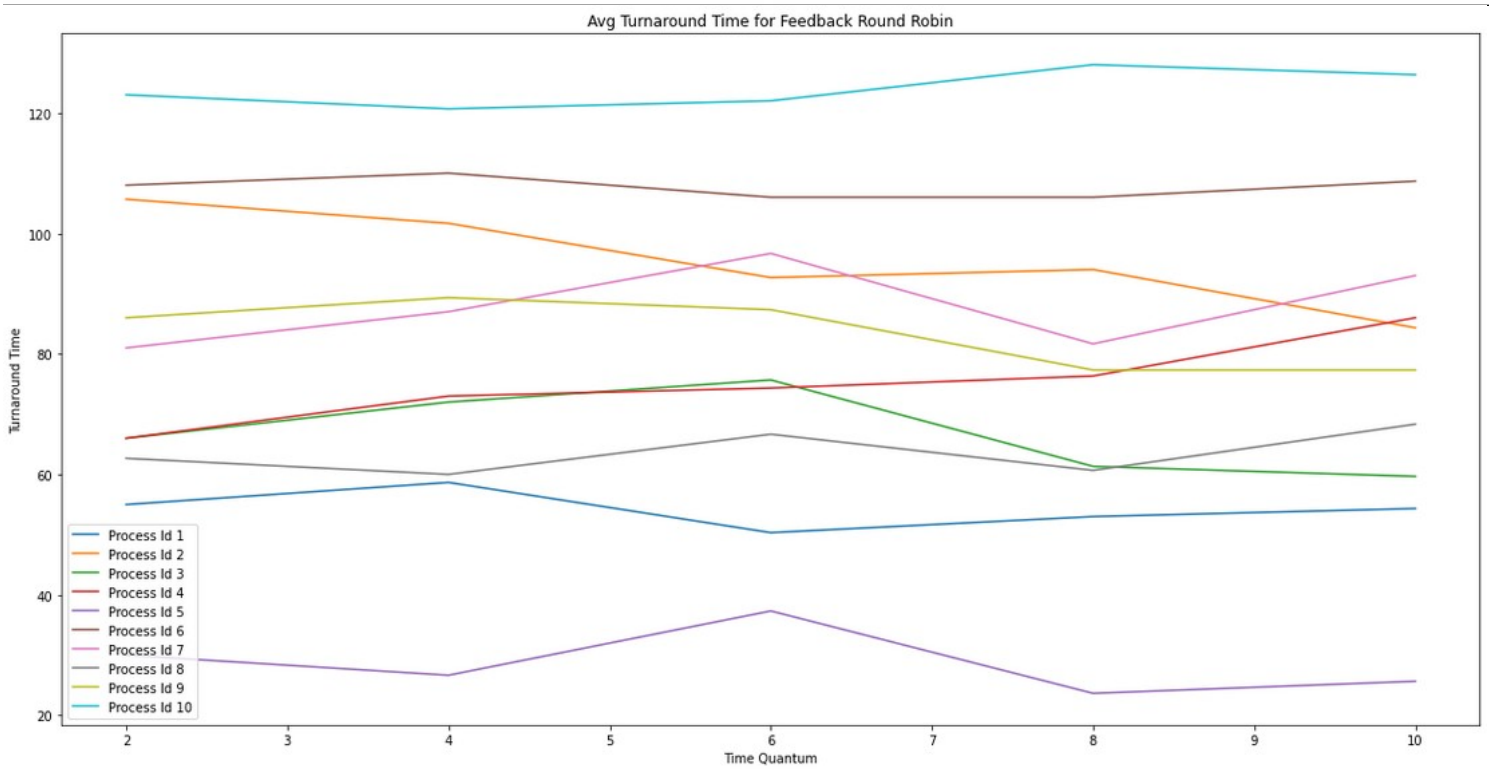
Setup

There will be 3 sets of tests, with different seeds for each set of tests, the amount of processes is set at 10. The time between processes sent is 0, in order to show the scheduler at full strength. All parameters are kept the same otherwise, except the Time Quantum which will rise by 2 from 2, 5 times. So; 2, 4, 6, 8, 10. This is to ensure the range and difference between the bigger and small Time Quantums is shown and accurately analyzed.

These outputs can then be analysed by inputting them into graphs to identify anomolies and test the prediction.

Graphs


Avg Turnaround Time for Round Robin
Turnaround Time
Time Quantum
Process Id 1
Process Id 2
Process Id 3
Process Id 4
Process Id 5
Process Id 6
Process Id 7
Process Id 8
Process Id 9
Process Id 10


Avg Turnaround Time for Feedback Round Robin
Turnaround Time
Time Quantum
Process Id 1
Process Id 2
Process Id 3
Process Id 4
Process Id 5
Process Id 6
Process Id 7
Process Id 8
Process Id 9
Process Id 10

Analysis

On reflection, the amount of time quantums tested was not enough to show a significant change and trend in Turnaround time. However, the graphs show that the different processes did respond to the change in time quantum, and in different ways for the schdulers tested.

The first graph, referring to the Round Robin Scheduler, shows processes 3, 5, and 7 confirming my predictions on the experiment, as they represent the curve of the Gaussian (normal) distribution, this is because these processes have a burst time near to the mean. The other processes however, don’t adhere to this prediction, due to their burst times being higher or lower than the mean. Examples of higher burst times are the processes; 10, 8 and 4. These have a positive increase and therefore perform better with a higher Time Quantum. Examples of lower burst times are the processes; 1, 2 and 9. These have a negative general increase on the graph, or have a peak at the lower Time Quantums, and therefore perform better here due to their lower burst times.

The second graph, referring to the Feedback Round Robin Scheduler, shows processes 5, 7 and 3 confirming my predictions on the experiment, as they represent the curve of the Gaussian (normal) distribution, due to these processes having a mean burst time. The other processes also don’t align with the prediction similar to the last graph. Positive increases in this graph include, the processes 10 and 4. Negative increases include the processes 2 and 9. The rest of the processes for this graph, have no trend. This suggests that the increase in Time Quantum, has no difference to the turnaround time of a process.

In summary, even though some of the processes do correlate with the prediction made, and due lack of distinct trends, the graphs show and therefore conclude that there is no difference that changing the Time Quantum has, at least in small amounts, in effecting the average Turnaround time of a process.

Experiment 2

Objectives

The effects that high workload has on response times. Response times being the time from creation to the first response. High workload is classed here as being a lot of processes in the same time frame. The experiment here is to determine how higher workloads effect reponse rates across all of our scheduling algorithms.

Therefore, the amount of processes are going to change throughout the tests, but all other factors will stay the same; e.g. Time Quantum, or alpha. This to get a true idea of what is happening. The process involves measuring the time from creation to termination for each of the processes, and working out how this changes with a greater amount of processes to execute. The time between arrival of the processes will be set to 0, to show a difference in times when more processes are added.

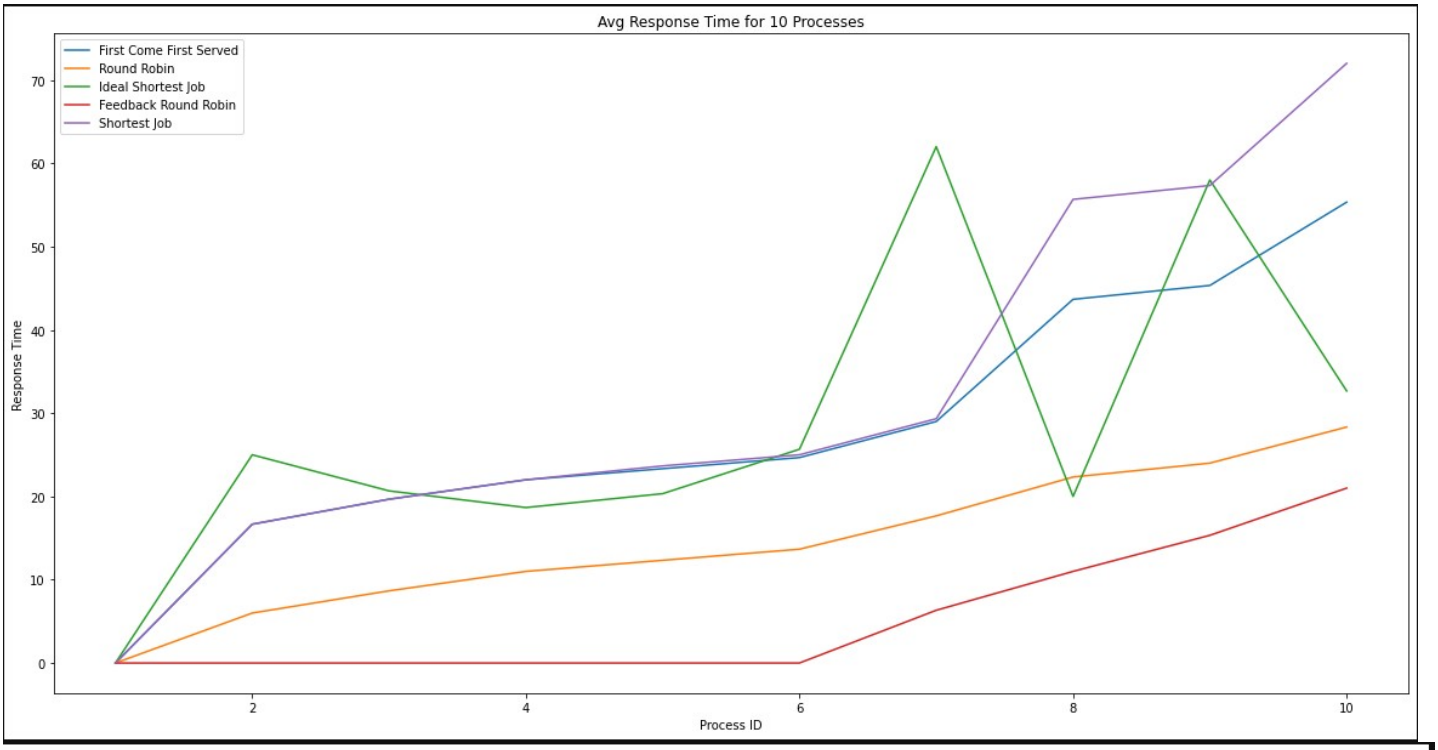
The prediction is that the higher the woakload the higher the overall response times. This could obviously be dependant on the scheuler that we are testing, but in general the prediction is that the more items in the scheduler the long it will take for them to be executed and discarded on average.

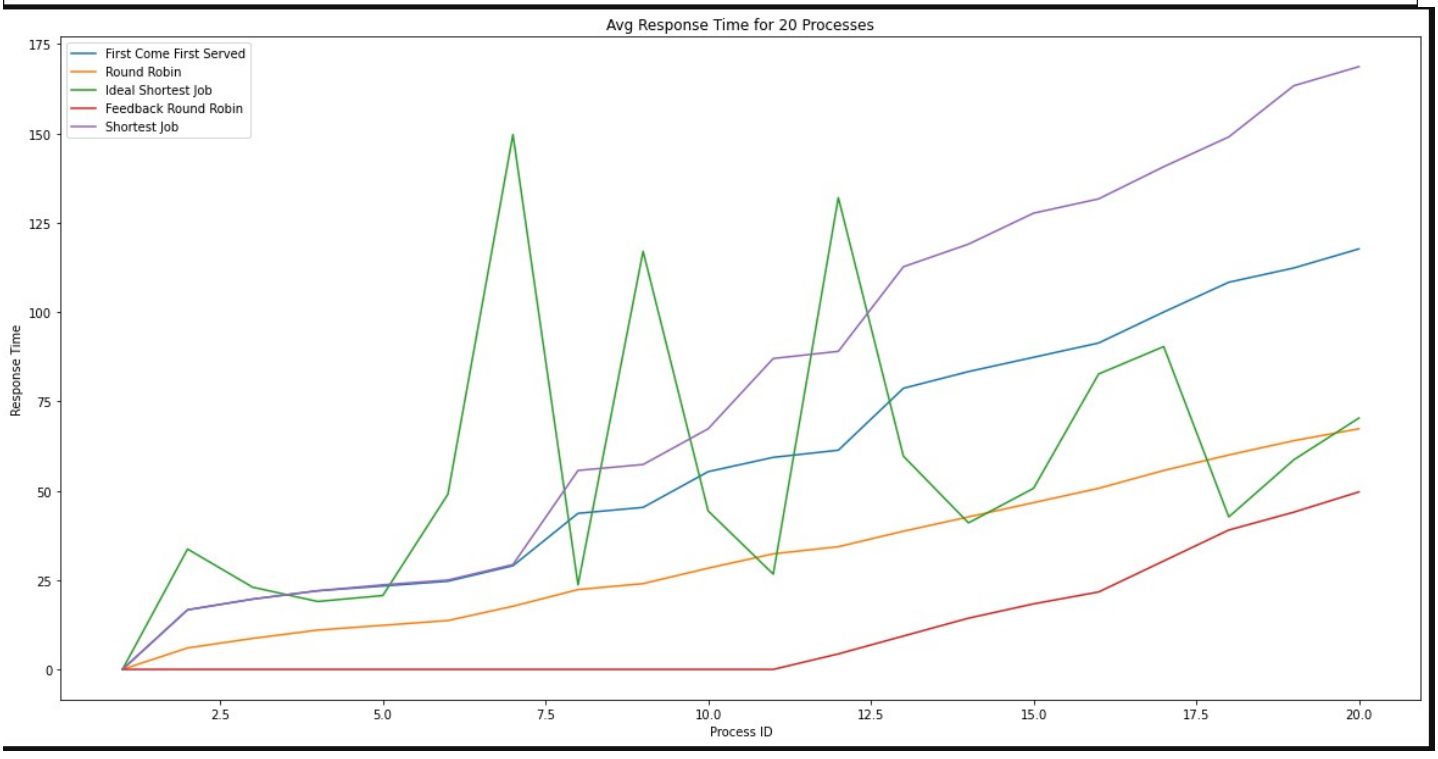
Setup

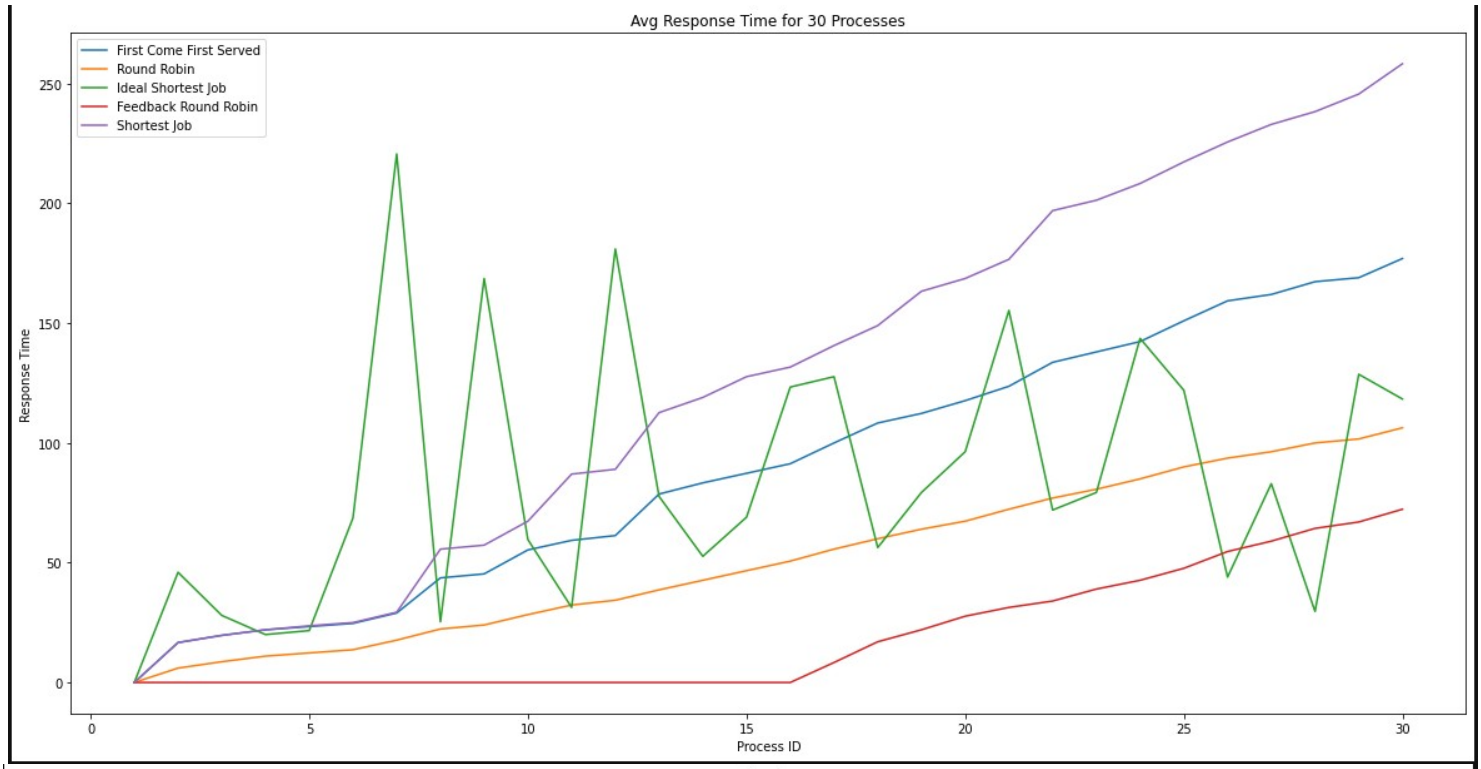
There are 5 tests for each scheduler that we can represent. For each test there are 5 indivual tests using different amounts of workloads, 3 times with different seeds to ensure accuracy, theses 3 seeds are however the same for each of the 5 tests to ensure comparison. The amount of processes running at the same time will go up from 10 by 10, so; 10, 20, 30, 40 ,50.

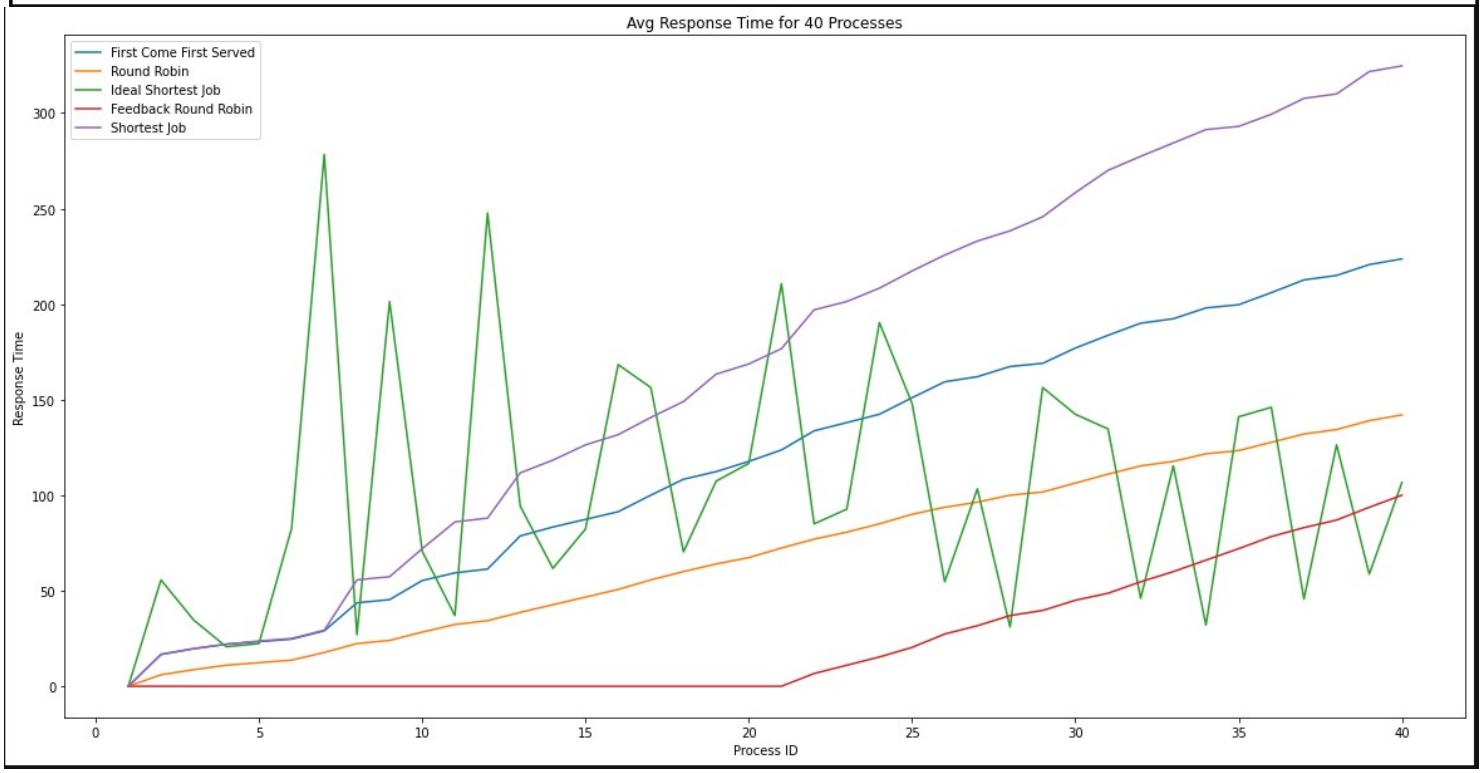
These outputs can then be analysed by inputting them into graphs to identify anomolies and test the prediction, for each of the tests and these can then be used for comparision.

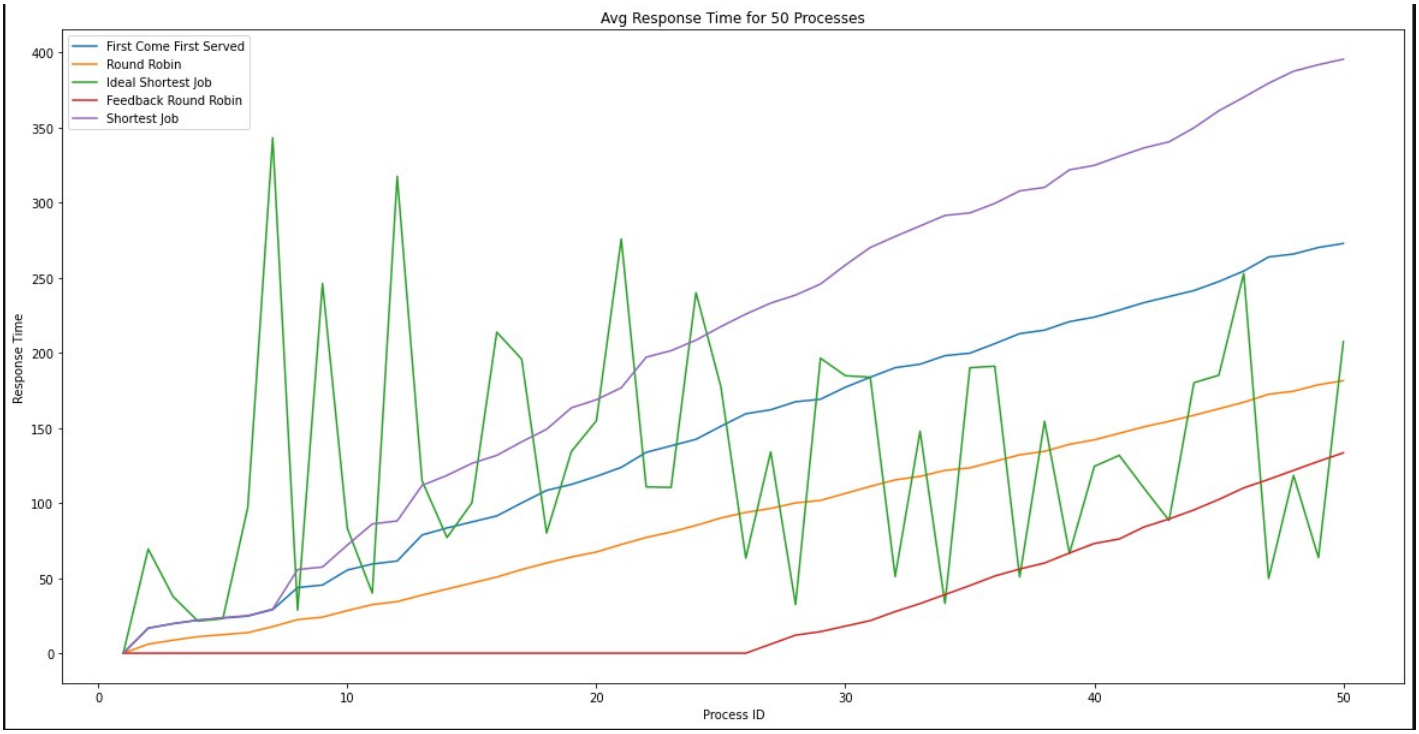
Graphs


Avg Response Time for 10 Processes
First Come First Served
Round Robin
Ideal Shortest Job
Feedback Round Robin
Shortest Job
Response Time
Process ID
0
10
20
30
40
50
60
70
2
4
6
8
10


Avg Response Time for 20 Processes
First Come First Served
Round Robin
Ideal Shortest Job
Feedback Round Robin
Shortest Job
Response Time
Process ID
0
25
50
75
100
125
150
175
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Avg Response Time for 30 Processes
First Come First Served
Round Robin
Ideal Shortest Job
Feedback Round Robin
Shortest Job
Response Time
Process ID
0
50
100
150
200
250
0
5
10
15
20
25
30


Avg Response Time for 40 Processes
First Come First Served
Round Robin
Ideal Shortest Job
Feedback Round Robin
Shortest Job
Response Time
Process ID
0
50
100
150
200
250
300
0
5
10
15
20
25
30
35
40



Analysis

On reflection, the turnaround time is not the easiest amount the measure as for feedback round robin a lot of the time it is 0, as they are instantly processed by the scheduler, which doesn't allow for much comparision.

The average response time with 10 processes graph's trends, are intresting as they are different to the other graphs, as the Ideal Shortest Job scheduler line has a positive trend, and doesn't change from this trend. But shows that the higher process IDs increase the response time, but this is because of the fact that all the processes are started at the same time, so the ones with a smaller ID are looked at first, which is staying true to the hypothesis of the experiment; the higher the process count the larger the response time.

The graphs for 20, 30 and 40 processes are similar to that of the 50 processes. This is because the response times don't change for each new amount of processes added due to each of the process tests using the same seed, so the first 10 are always the same 10 processes. However, due to the next 10 processes added each time, the order that these are processed changes due to the

processing time of the new processes being smaller. But, this doesn’t change for First Come First Served as it goes in order of process ID.

The average response time with 50 processes graph’s trends, for all the schedulers except Ideal Shortest Job are the same, with a slow positive trend, but Ideal Shortest Job’s trend is hard to correlate due to the erratic jaggers in the graph, which if we put a trend line is also a slow positive trend. However, it is represented like this as the scheduler’s tactic is too take the smallest processing time process first and then move on to the next smallest and so on. This therefore means the process list is not in order of ID but in order of processing time needed. Hence the trend of the graph.

In summary, Therefore, this shows that the hypothesis is correct as the trend of the graphs is positive, and as the processes increase the response times also increase. However, it is worth noteing that if done again the seed should have been changed for each number of processes, even if each of the graphs are averages of 3 seeds, these 3 seeds are the same. The time of creation for each of the processes should also be stagged in order to allow each of the algorithms to show their best response times. For example, Shortest Job, was the slowest at responding on average as all the processes were created at the same time, it couldn’t work out what was the shortest job effectively.

Experiment 3

Objectives

Whether the First-Come First-Served scheduler out performs the Shortest Job First Scheduler in a prolonged period. The prolonged period is defined as a decent amount of time, that allows both aspects of the schedulers to shine. By outperfoms this is in overall turnaround time. If one scheduler has a better average in all these areas then it would outperform the other.

The overall performance is the shortest amount of time the processes are executed, which is why the values of the mean bursts for CPU and IO are changing.

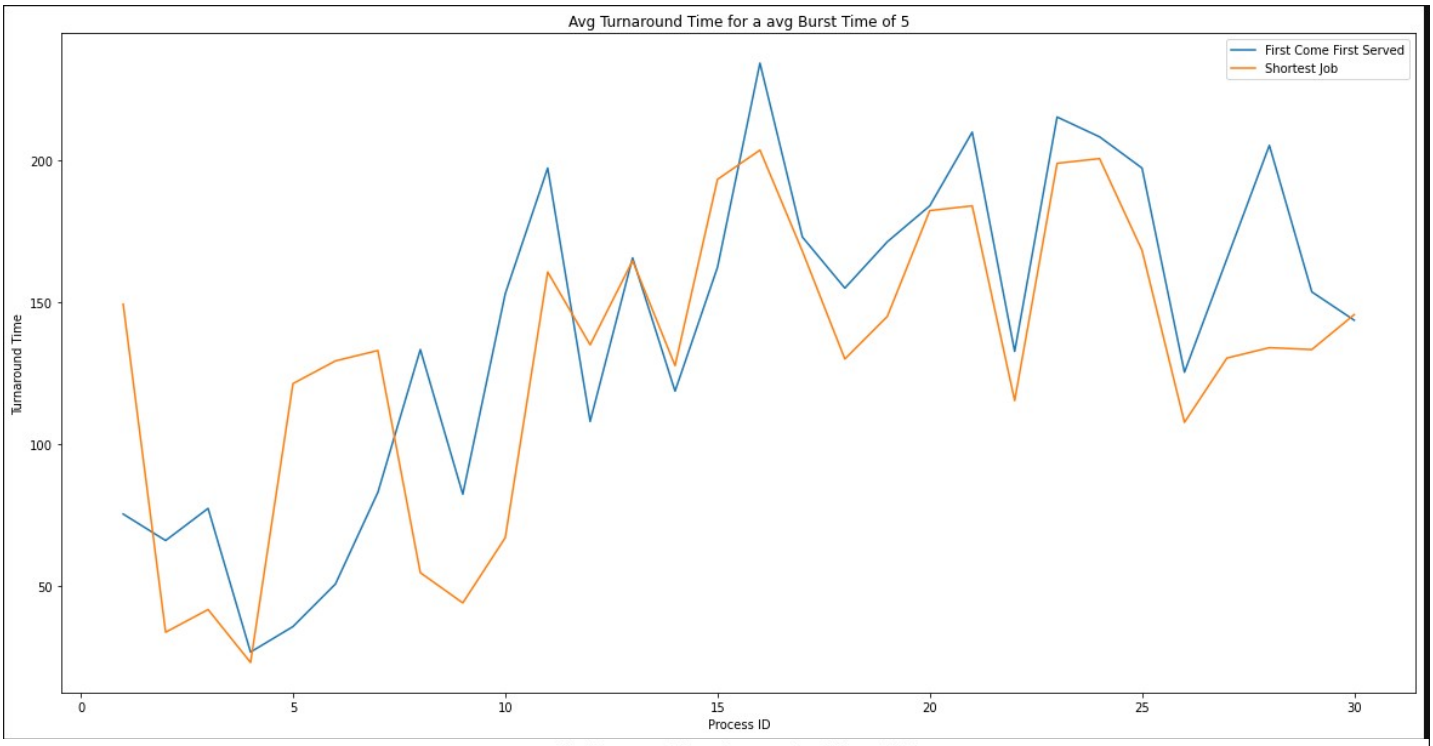
The prediction is that the First-Come-First-Served scheduler will perform better than Shortest Job First due to the limited overhead FCFS has; this allows it to execute quicker and therefore spend less time blocking, and making processes wait therefore increasing turnaround time.

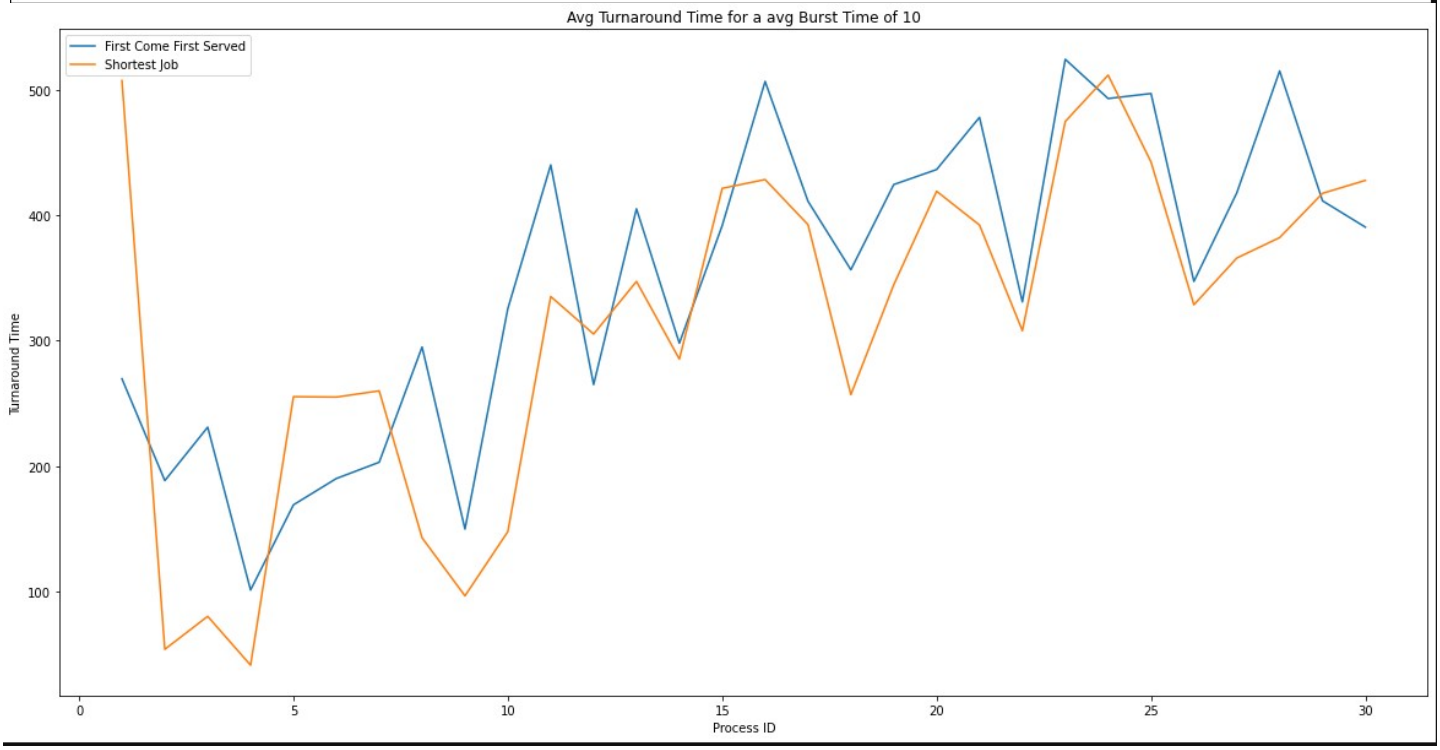
Setup

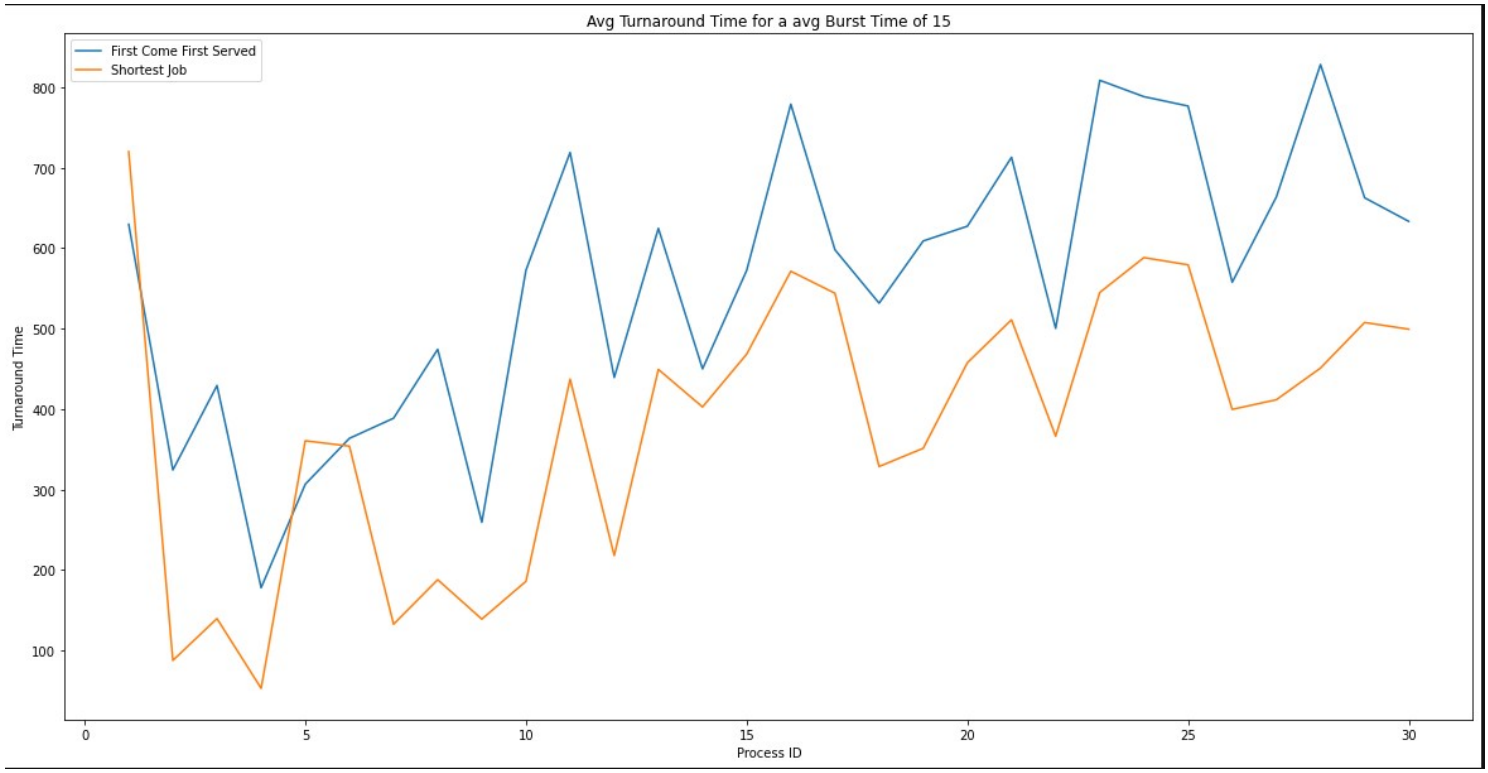
The number of processes will stay the same at 30, to give a reasonable amount of time for each test. Each test the amount of bursts for both CPU and IO will increase by 5 for each test from 5 upto 25. For Shortest Job First the initial burst estimate will be adjusted acoordingly to keep the schedulers fair.

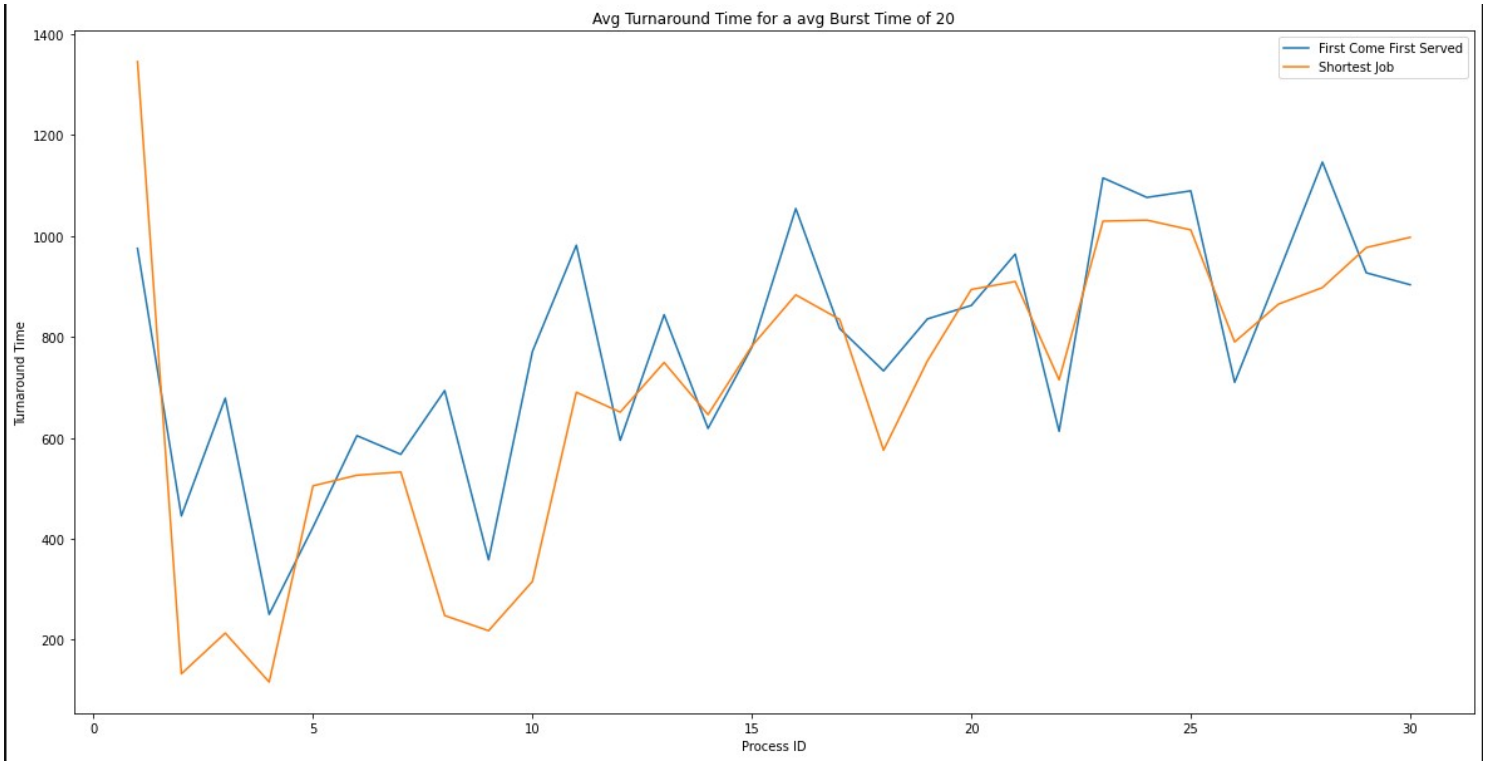
Each of the tests will have 3 seeds to elimate anomolies and there will be two tests in total for each scheduler. The outputs can then be analysed by inputting them into graphs to identify anomolies and test the prediction, for each of the tests and these can then be used for comparision.

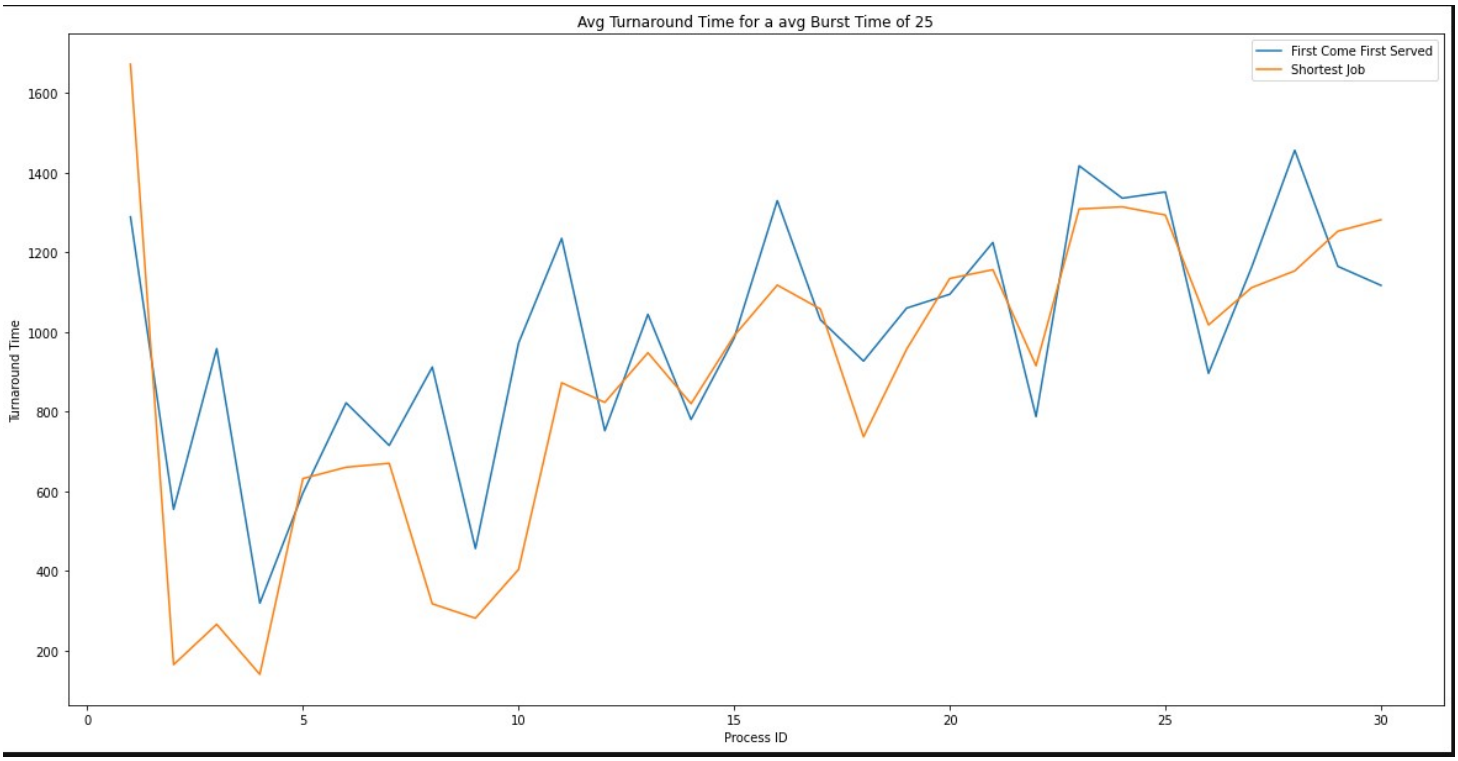
Graphs


Avg Turnaround Time for a avg Burst Time of 5
First Come First Served
Shortest Job
Turnaround Time
Process ID
0
5
10
15
20
25
30
50
100
150
200


Avg Turnaround Time for a avg Burst Time of 10
First Come First Served
Shortest Job
Turnaround Time
Process ID
0
5
10
15
20
25
30
100
200
300
400
500


Avg Turnaround Time for a avg Burst Time of 15
First Come First Served
Shortest Job
Turnaround Time
Process ID
100
200
300
400
500
600
700
800
0
5
10
15
20
25
30


Avg Turnaround Time for a avg Burst Time of 20
First Come First Served
Shortest Job
Turnaround Time
Process ID
200
400
600
800
1000
1200
1400
0
5
10
15
20
25
30



Analysis

At first glance, the graphs appear to be similar, however the y-axis scales up when the mean burst time is increased. On reflection, the burst times and waiting times could be included for this experiment, but the problem with this is that these times were 0 and sometimes 15, so wouldn't of averaged well, and therefore would not analyse well. Also, the seeds should have been changed between each burst time, to change the trend of the graph, however, this shows what happens when the average burst time changes, which means that the inputs are generated for each burst time and therefore in this case it isn't a problem.

When the average burst is 5 and 10, the two schedulers are tightly correlated with little between them, and a general positive trend as the processes increased in relation to turnaround time.

However, when the average burst time is 15, there is a stark difference between the turnaround times of First Come First Served compared to Shortest Job, there is similar trend lines, but Shortest Job response much quicker than First Come First Served. This is against what the hypothesis predicted.

When the average burst is 20 and 25, there seems to be no difference between 20 and 25 burst time, but the overall turnaround time is higher by 100. These graphs show Shortest Job being consitently lower in overall turnaround times than First Come First Served once again going against the hypothesis.

In summary, the experiment shows that Shortest Job overall over a prolonged period time out performs First Come First Served. This is because, shortest job responds to the shortest job available first, and therefore doesn’t let shorter processes hang when its working on a larger process, like First Come First Served does. An increase in the number of processes tested, around 50, would help to show a better trend between the two schedulers if the experiment was done again, but it should still show that Shortest Job out performs First Come First Served.