Amacker Olivier & Savioz Yolan 15.10.2024

OS-Scheduler

This document serves as a small performance analysis, comparing 4 algorithms regarding task schedule.

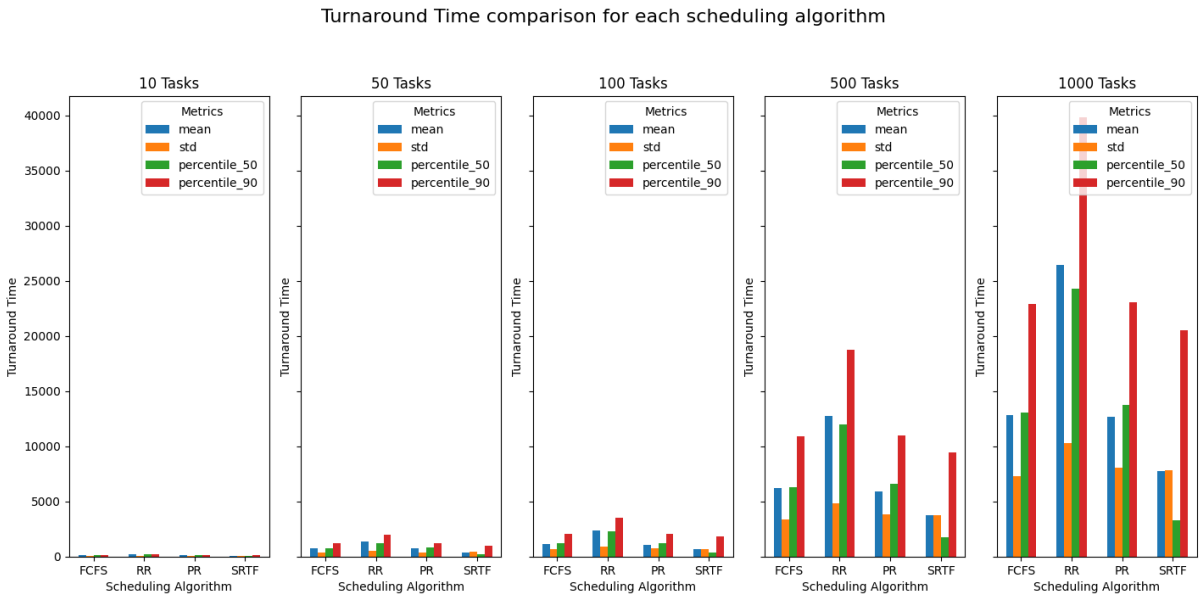
1. First come first served (fcfs)
2. Round robin (rr)
3. Priority scheduling
4. Shortest Remaining Time First (SRTF)

For every algorithm, we generated tasks with the file generator.py. We generated 5 batches with different numbers of tasks.

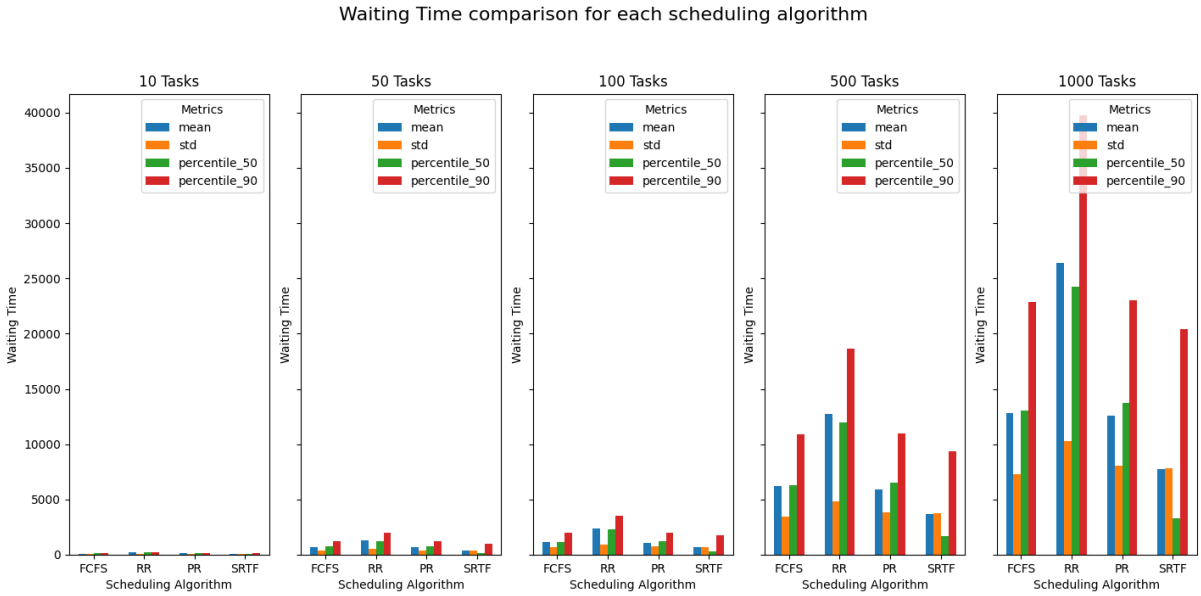
1. 10 tasks
2. 50 tasks
3. 100 tasks
4. 500 tasks
5. 1000 tasks

We then ran stats.py to extract useful information and generate different graphs.

Here are some graphics made with matplotlib that compare each algorithm for process scheduling. (They can be found on a larger scale the end of this document)

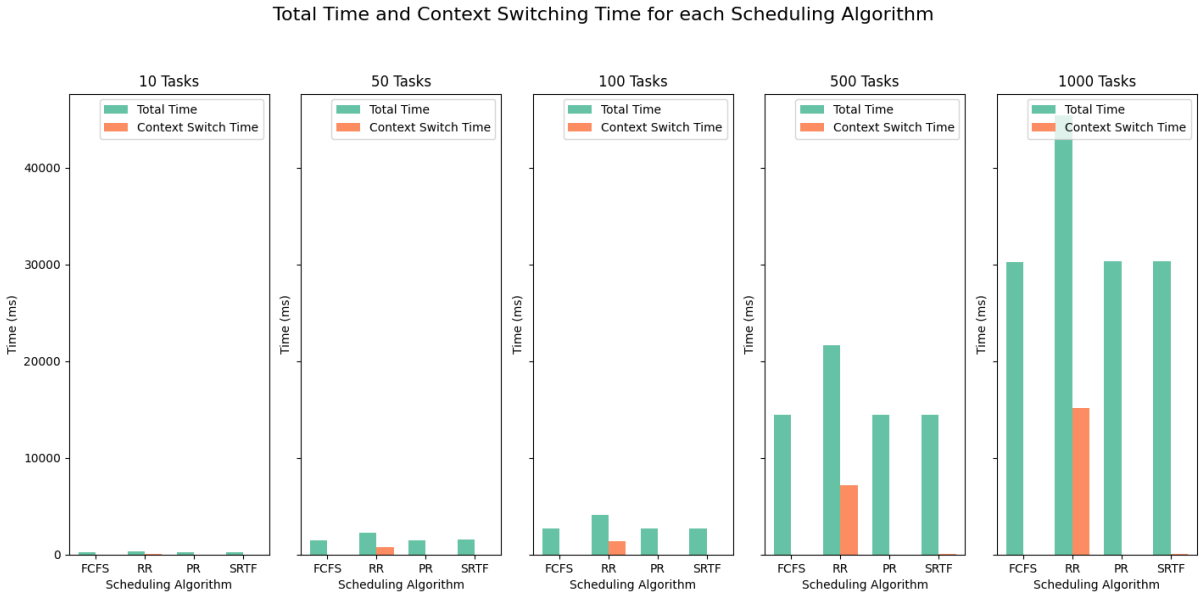


This first chart highlights how each algorithm are pretty much equals regarding turnaround time. However, the Round Robin (RR) algorithm shows a steep increase in turnaround time due to its preemptive nature, making it less efficient for larger task sets. FCFS provides reasonable performance for smaller numbers of tasks but suffers in the higher percentiles as task size grows, and Pr falls somewhere in between, with variable results across metrics. The percentile 90 column especially highlights how longer tasks in RR can cause substantial delays for the remaining tasks.



This plot graph shows a waiting time comparison for the different algorithms.

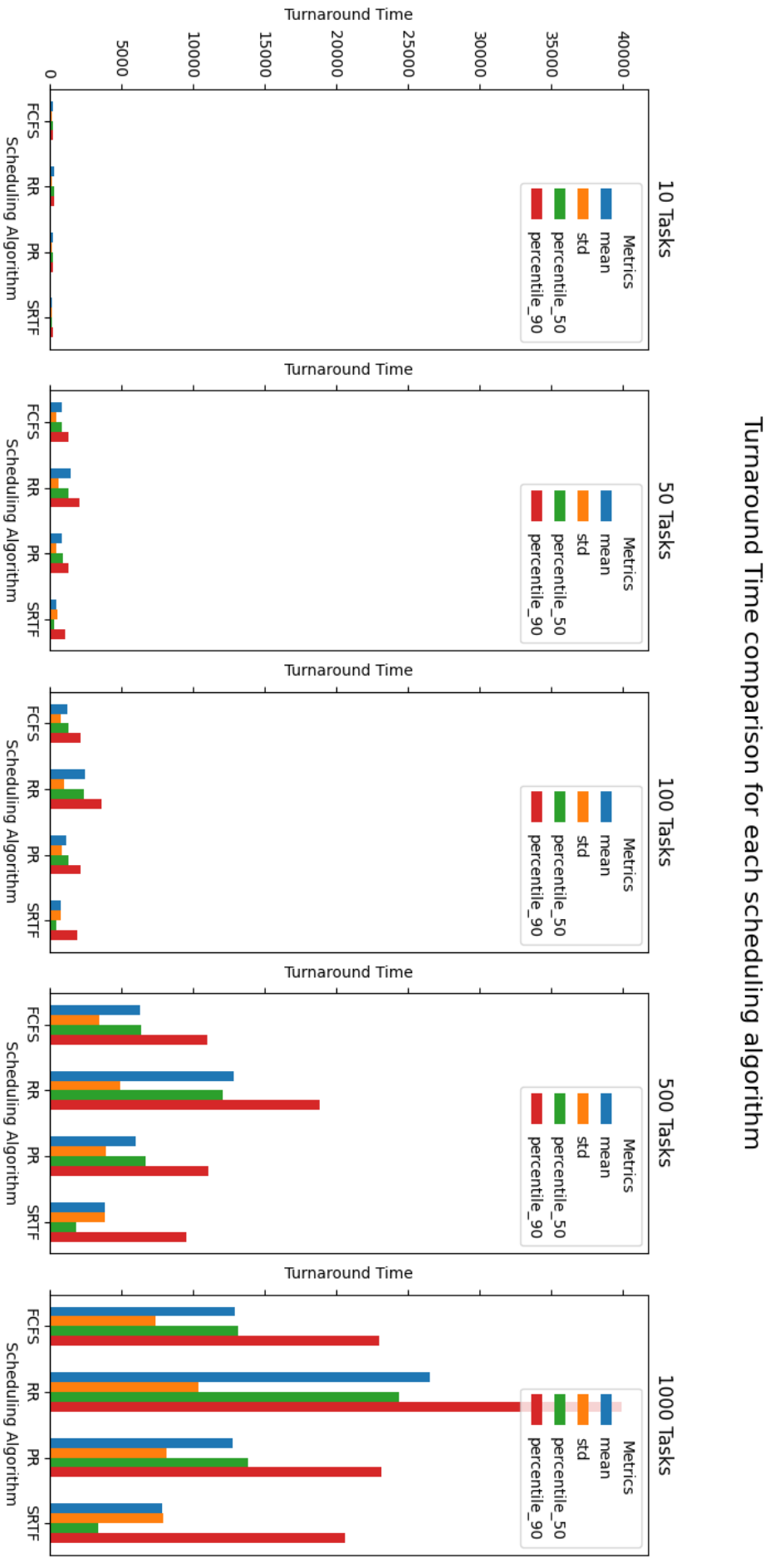
As you can see, the two previous graphics are quite the same. This is due to the fact that the turnaround time is directly influenced by the waiting time. Turnaround time includes both the waiting time and the execution time of a task, so if the waiting time increases, the turnaround time tends to follow the same pattern. Therefore, variations in the scheduling algorithm affect both metrics in a similar way, leading to graphs that are visually alike, especially if the execution the is short compared to turnaround time.

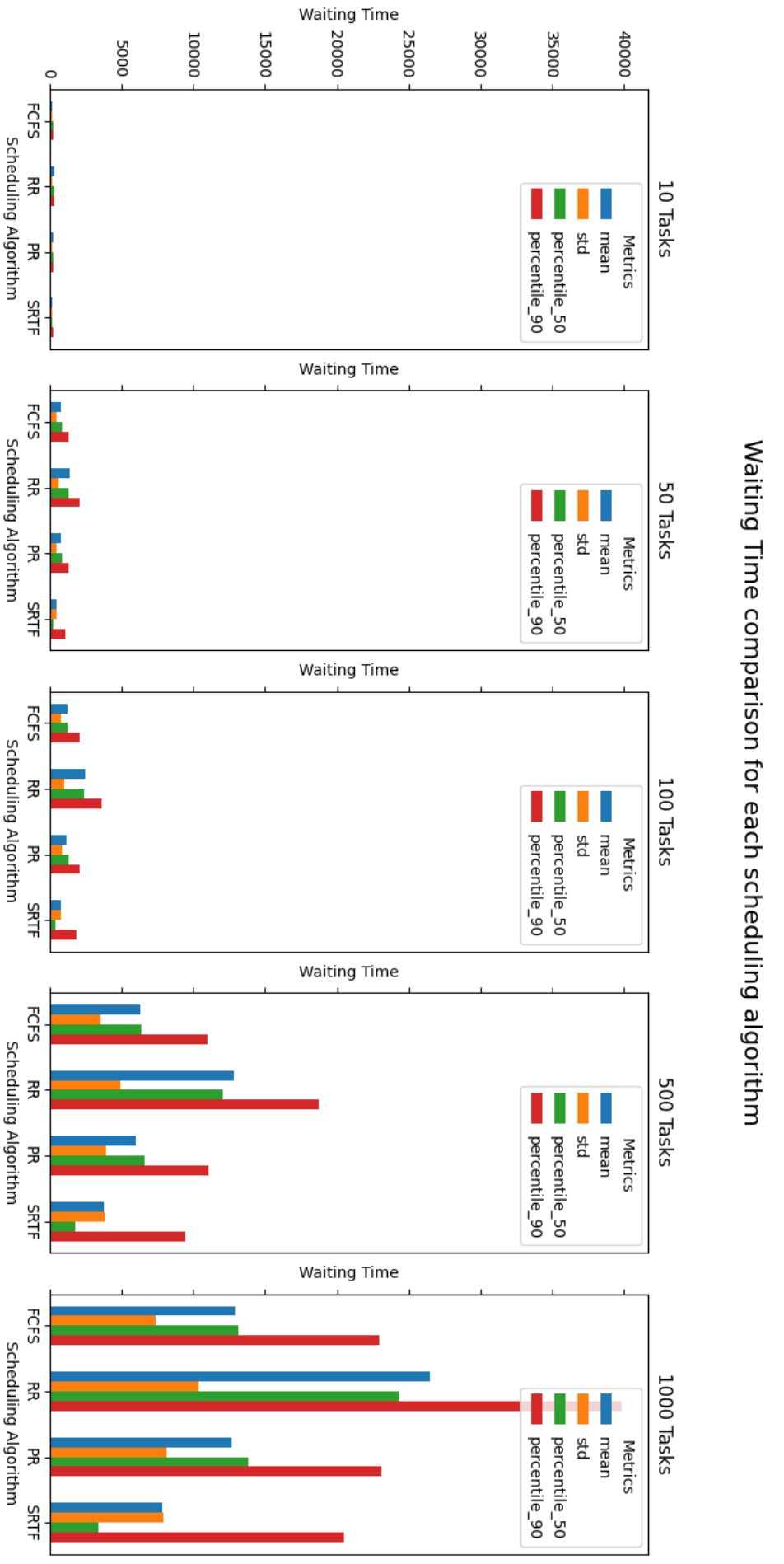


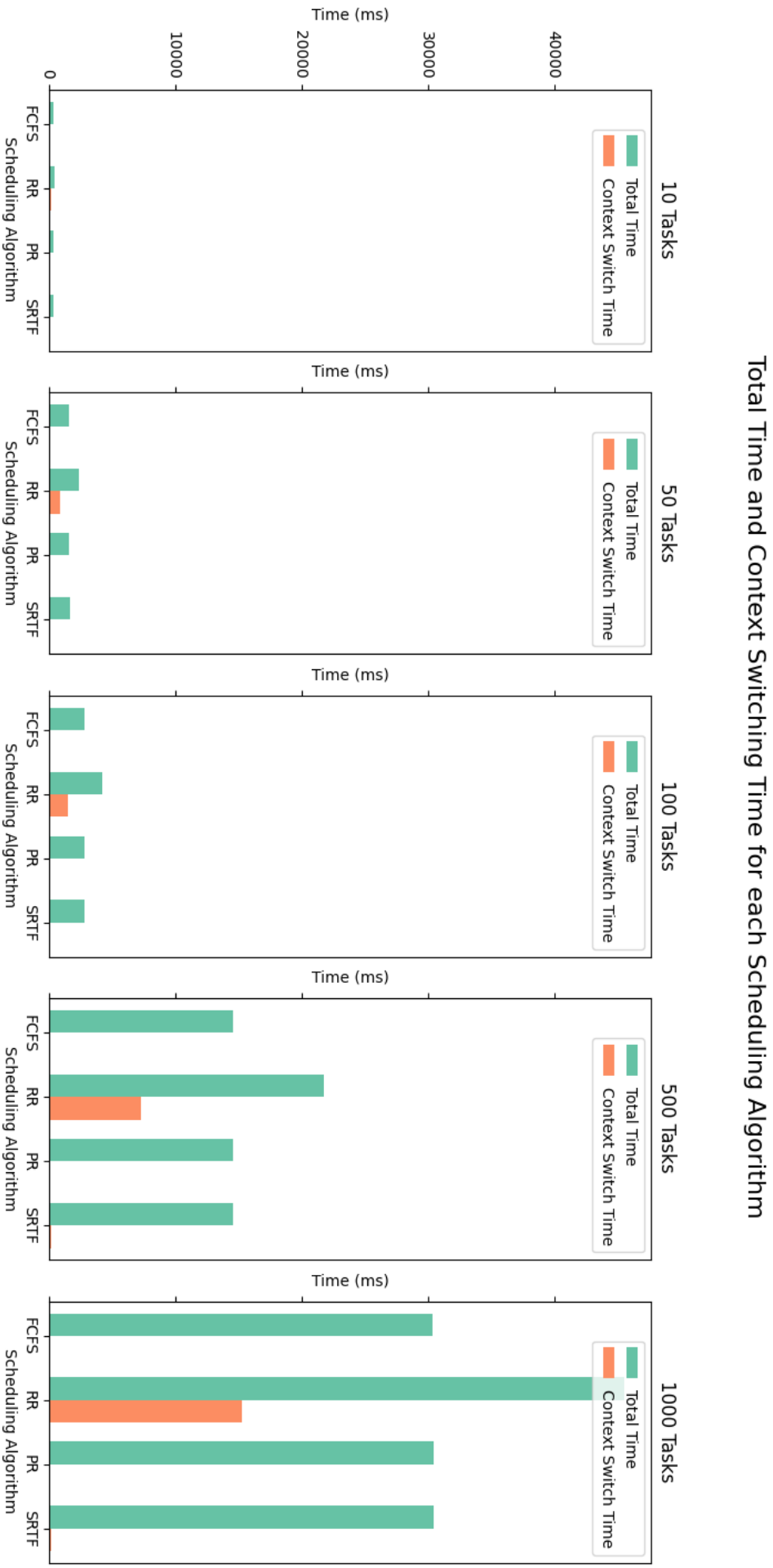
This chart clearly illustrates the impact of increasing the number of tasks on total execution time and context switching time for different scheduling algorithms. While differences are minimal with a small number of tasks (10 or 50), as the number of tasks grows (100, 500, 1000), context switching time becomes significantly more pronounced, especially for the Round Robin (RR) algorithm, which incurs a large overhead due to its frequent preemptions. In contrast, the SRTF algorithm remains more efficient in terms of total time, while FCFS shows variable performance depending on the number of

tasks, with lower total execution time, but longer waiting times for subsequent tasks. This chart highlights the trade-offs between responsiveness, efficiency, and context switching cost as the number of tasks increases.

In conclusion, the choice of scheduling algorithm depends heavily on the specific system requirements. Round Robin (RR), while providing the fastest response time, incurs the highest total execution time due to frequent context switches, making it ideal for interactive systems but inefficient in terms of overall performance. First Come First Serve (FCFS), though simple and efficient with minimal context switching, suffers from long waiting times and poor responsiveness, especially when handling long tasks. Priority Scheduling (Pr) offers flexibility by allowing high-priority tasks to be processed quickly, but it introduces variability in waiting and turnaround times. Lastly, Shortest Remaining Time First (SRTF) is the most optimal in minimizing waiting and turnaround times, making it suitable for environments with shorter tasks, though it may lead to frequent preemptions and reduce responsiveness for longer tasks. Therefore, selecting the best algorithm is a trade-off between responsiveness, efficiency, and task characteristics**.**


Turnaround Time comparison for each scheduling algorithm
10 Tasks
50 Tasks
100 Tasks
500 Tasks
1000 Tasks
Turnaround Time
Scheduling Algorithm
FCFS
RR
PR
SRTF
Metrics
mean
std
percentile_50
percentile_90
0
5000
10000
15000
20000
25000
30000
35000
40000


Waiting Time comparison for each scheduling algorithm
10 Tasks
50 Tasks
100 Tasks
500 Tasks
1000 Tasks
Waiting Time
Scheduling Algorithm
FCFS
RR
PR
SRTF
Metrics
mean
std
percentile_50
percentile_90
0
5000
10000
15000
20000
25000
30000
35000
40000


Total Time and Context Switching Time for each Scheduling Algorithm
10 Tasks
50 Tasks
100 Tasks
500 Tasks
1000 Tasks
Time (ms)
0
10000
20000
30000
40000
Total Time
Context Switch Time
Scheduling Algorithm
FCFS
RR
PR
SRTF