Real Time Embedded System
Assignment – 2

Schedulability Analysis

Team - 12
Chinmay Bolumbu (ASU ID: 1213329869)
Rama Kumar Kana Sundara (ASU ID: 1213347614)

Section 1:

A program for analysing the different types of scheduling was written to read a given input file for several task sets. Analysis includes figuring out if the task sets are schedulable using Earliest Deadline First, Rate Monotonic scheduling algorithm and Deadline Monotonic scheduling algorithm. We made use of a input file named "inputfile.txt" for testing our program.

A task is schedulable if its task density is less than the 'Utilization Bound'. We choose to do the Utilization Bound test using Task density (Ci/Di) over task utilization (Ci/Pi). The reason for preferring Task density is because deadline of a task is not necessarily equal to the period. If the task fails to pass the Utilization bound test (value is more than the bound), nothing can be said about the schedulability of task till we do a Response Time Test. A task can only be said that it is schedulable or not based on its success or failure in a Response Time Test. Output of all the tests are tabulated in a generated file called "output.txt".

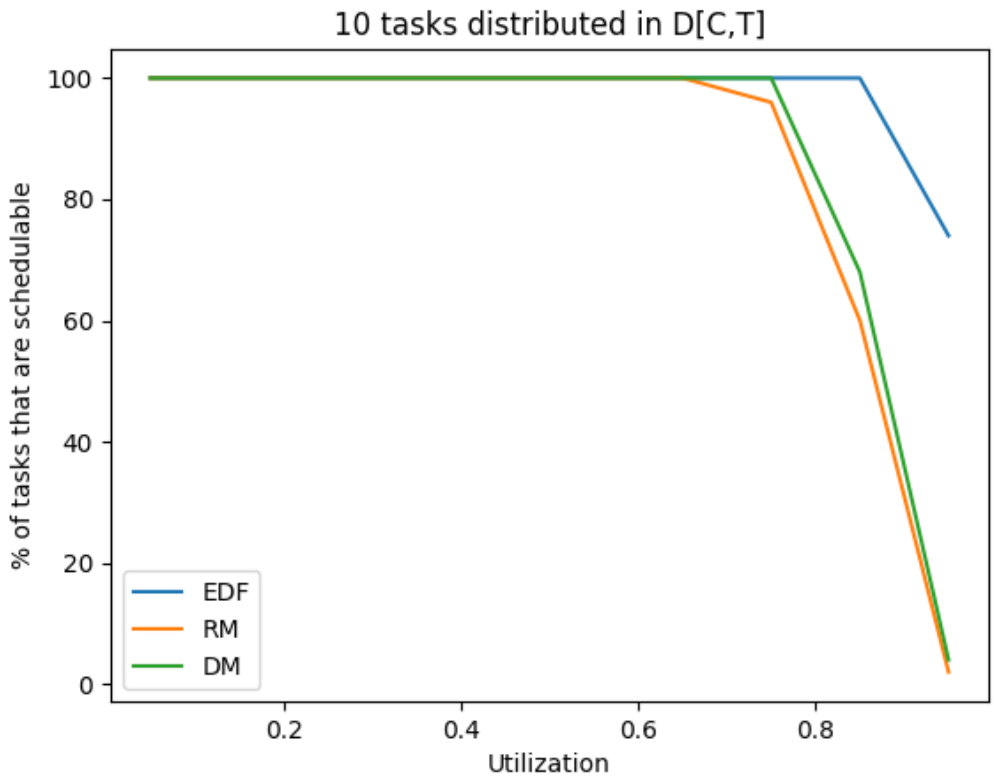
Following Test cases were checked using the program:

- The first task set had tasks such that the deadline of each task was equal to its period. We used Utilization Bound test as a first measure to find whether it is schedulable or not. It turned out that the Utilization of tasks was less than the Utilization Bound. So, we concluded that the task set is schedulable for EDF, RM and DM algorithms.
- The second task set had tasks such that deadline of each task in this set was equal to its period. However, we made sure that the Utilization Bound test would fail. Since the Utilization (density) was more than the utilization bound, our result about the schedulability of tasks with the initial UB Test was inconclusive. So, we followed the Utilization Bound test with Response time analysis. The response time for the lowest priority task was calculated to be less than its deadline for all the three algorithms. Hence, we concluded that the task set was schedulable in EDF, RM and DM algorithms.
- The third task set had tasks such that they do not have deadlines equal to its period. It is schedulable only with DM algorithm. The result of the Utilization Bound test was inconclusive. This is followed by a Response Time test which gave a response time for the lowest priority task for all the three algorithms. The deadline of the lowest priority task when using RM was less than the Response time calculated, and hence the task set is not schedulable using RM algorithm. The deadline, if we use DM algorithm for the lowest priority task was more than the calculated Response time. So, it is schedulable with DM algorithm.

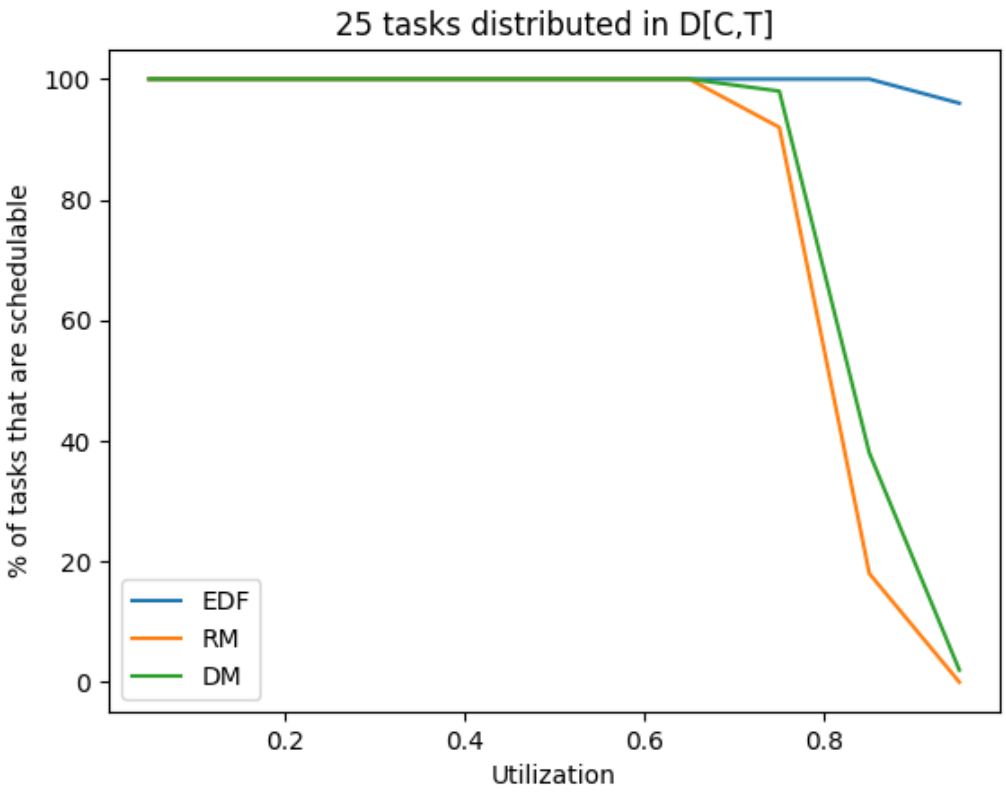
- Other task sets were tested using the program. The task sets were such that they fail the initial UB Test. We check for the Response time / Loading factor of the tasks and then compare the Utilization of every task set with calculated RT or LF. Thereby, we conclude if the Task set is schedulable in each of the scheduling policies. In Response Time Analysis, program reports the Worst-case response times and in Loading Factor approach it reports the first missing deadlines.

Section 2:

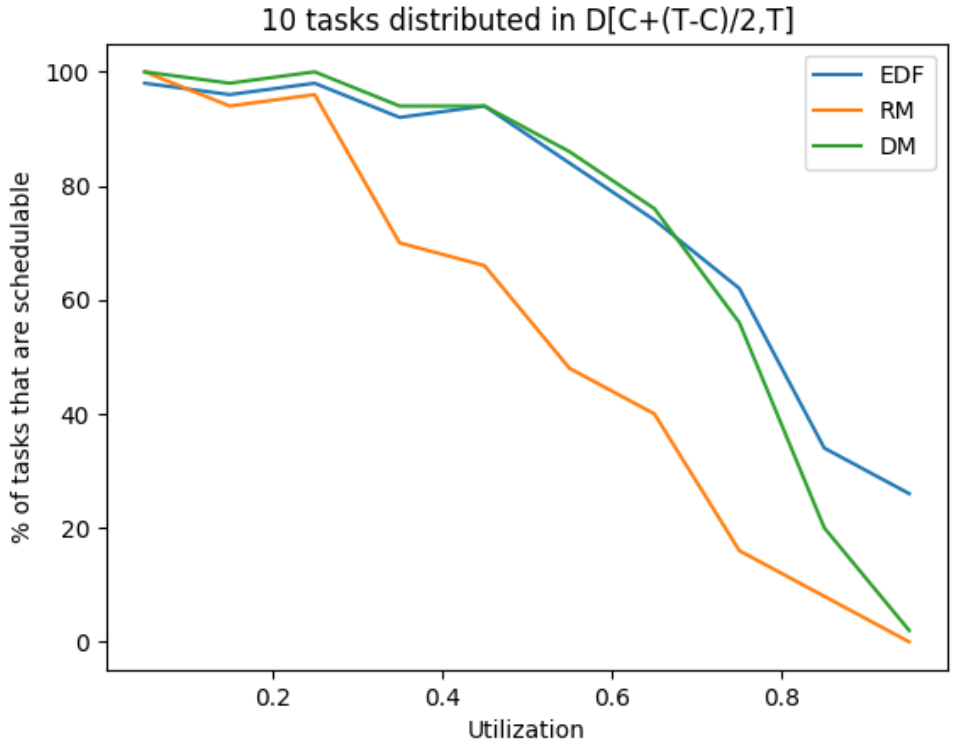
We analyse the performance of different scheduling algorithms in this section. A program was written to generate Random task sets with deadline varying (Di in [Ci, Ti] and Di in [Ci+(Ti-Ci)/2, Ti]). The percentage of task sets schedulable using Earliest Deadline First, Rate Monotonic Scheduling algorithm, Deadline Monotonic Scheduling Algorithm were calculated and were calculated. The results can be seen in Plots 1- 4, having the number of tasks per task set equal to 10 and 25 respectively. It may be observed from the plots that the Utilization in EDF algorithm is closer to 100% (Utilization factor: 1) when compared to other approaches.



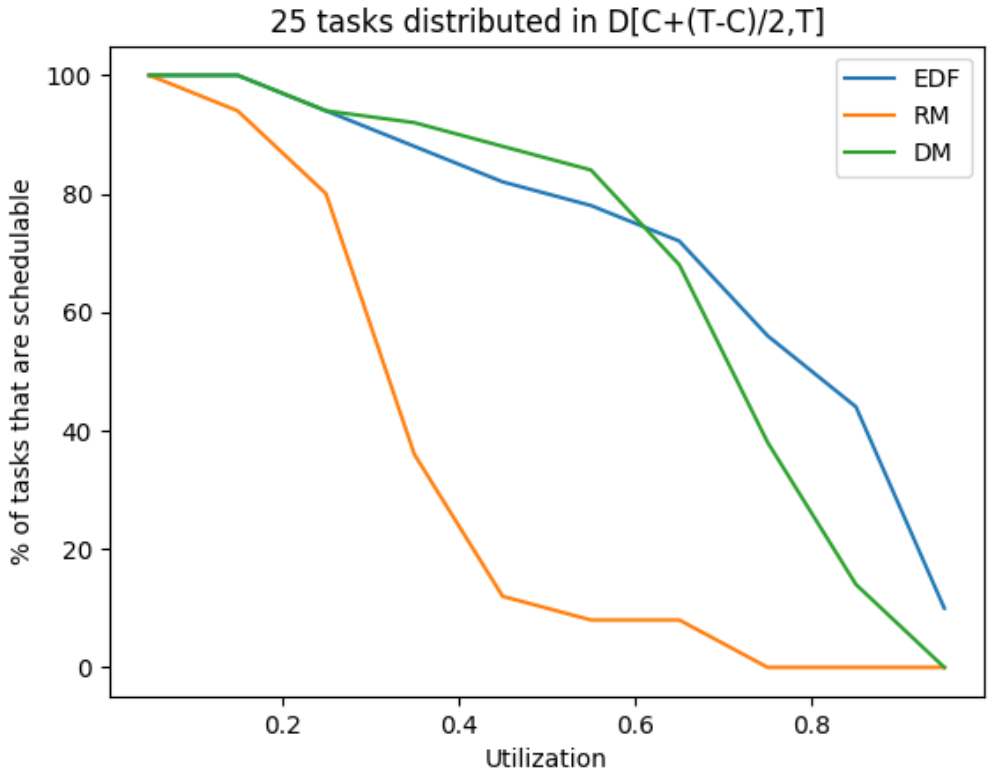
Plot 1: 10 Tasks, Di in [Ci, Ti]



Plot 2: 10 Tasks, Di in [Ci, Ti]



Plot 3: 10 Tasks, [Ci+(Ti-Ci)/2, Ti])



Plot 4: 25 Tasks, [Ci+(Ti-Ci)/2, Ti])

We observe that if the deadline of a task is evenly distributed in [Ci, Ti], the performance of RM algorithm is inferior compared to the other two algorithms. The performance declines further if we increase the number of tasks per task set. This is because RM algorithm does not consider the deadlines of the tasks while assigning them priorities. This degrades the performance, as the tasks having a large period can have relatively small deadline. On the other hand, EDF algorithm has the best performance since this considers the deadline of every task for scheduling.

The deadline was then changed to [Ci + (Ti-Ci)/2, Ti] and the performance of the three algorithms was then analysed. The results can be seen in Plots 3 - 4, having the number of tasks per task set equal to 10 and 25 respectively. RM approach shows a inferior performance compared to the other two even in this case since it does not consider the deadlines of the tasks while assigning them priorities.