Assignment 2 Report

Section 1: Report of the analysis program

An analysis program was written to read a given input file for a number of task sets and analyze if the task sets are schedulable using Rate Monotonic scheduling algorithm, Deadline Monotonic scheduling algorithm and fixed priority scheduling with least slack time. . A sample input file was made, named "Sample_1.txt" which had four tasks in it.

As the deadline is not necessarily equal to the period, we do the "Utilization Bound" test using the task density (Ci/Di) rather than task utilization (Ci/Pi). This tells us that a task is schedulable if its task density is less than the 'Utilization Bound'. However, nothing can be said about the task schedulability if the value is more than the bound. Under this condition, we do a "Response Time" test to find whether the task is schedulable or not. The exact output obtained is available in the file "output.txt".

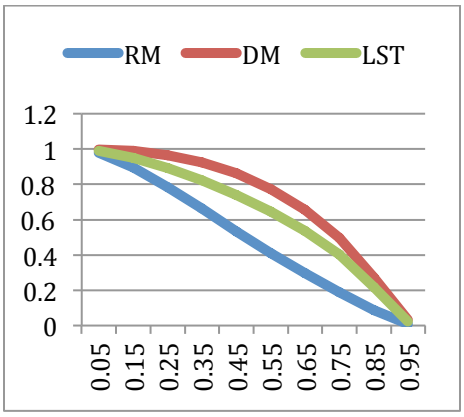
The first task set has three tasks in it. The deadline of each task in this set was equal to its period. We used the "Utilization Bound" test to find whether it is schedulable. It turned out that the utilization (density) of the tasks was less than the utilization bound. The utilization of the lowest priority task in this set was calculated to be 0.752 whereas the utilization bound for three tasks is 0. 779. So, we concluded the task set is schedulable for all three algorithms.

The second task set also has three tasks. The deadline of each task in this set was equal to its period. However, the "Utilization Bound" test was inconclusive, because the utilization (density) was more than the utilization bound. So, we followed the Utilization Bound test with "Response Time" test. The response time for the lowest priority task was calculated to be 300 for all the three algorithms. Its deadline was 350 for the lowest priority task, and hence the task set was schedulable for all the three algorithms.

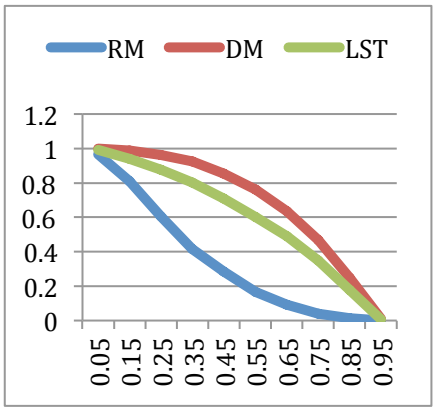
The third task set does not have the deadlines of its task equal to its period. It is schedulable only with DM algorithm. The result of the "Utilization Bound" test is inconclusive. The "Response Time" test gives the response time of 19.1 for the lowest priority task for all the three algorithms. The deadline of the lowest priority task is 10 when using RM, and hence the task set is not schedulable using the RM algorithm. The deadline, if we use DM algorithm, is 20.8 for the lowest priority task. So it is schedulable with the DM algorithm. Fixed priority scheduling with least slack time gives a deadline of 18.9 for the lowest priority task, which again makes it unschedulable.

The fourth task set also does not have the deadlines of its tasks equal to its period. This is schedulable with DM and fixed priority scheduling with least slack time. The response time for all the three algorithms is 18.1. However the deadlines for RM, DM and fixed priority scheduling with least slack time are 10, 20.8 and 18.9 respectively. So we can observe that it is schedulable using DM and fixed priority scheduling with least slack time, but not with RM.

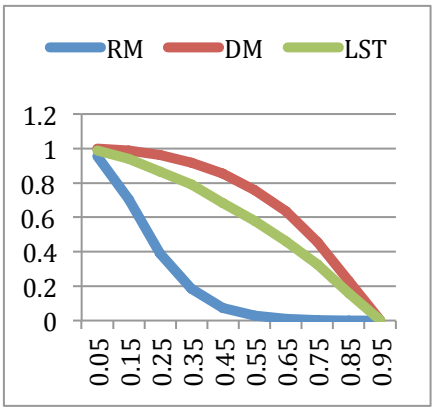
Section 2: Report of Comparative Analysis



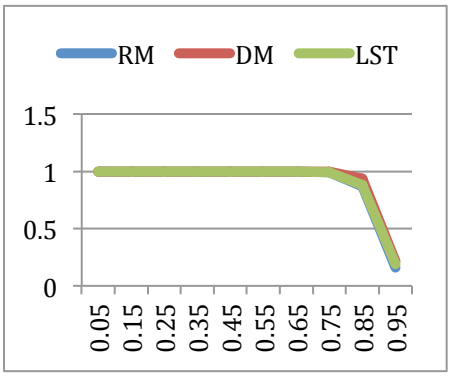
Plot 1: 12 Tasks, Di in [Ci, Ti]



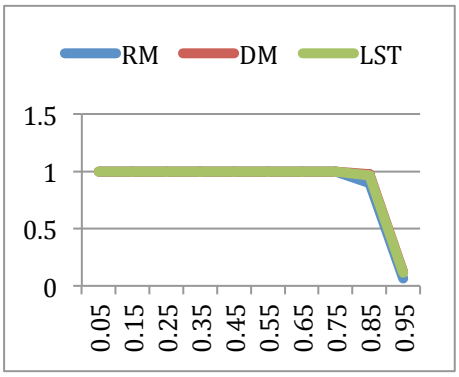
Plot 2: 24 Tasks, Di in [Ci, Ti]



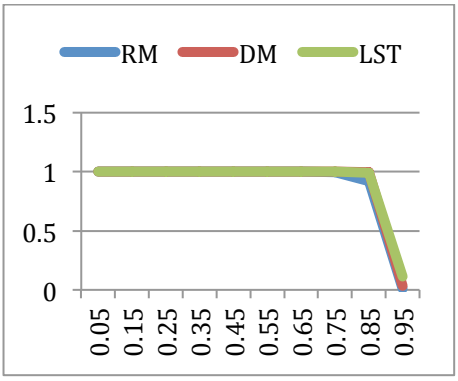
Plot 3: 48 tasks, Di in [Ci, Ti]



Plot 4: 12 Tasks, Di in [Ci+(Ti-Ci)/2, Ti]



Plot 5: 24 Tasks, Di in [Ci+(Ti-Ci)/2, Ti]



Plot 6: 48 Tasks, Di in [Ci+(Ti-Ci)/2, Ti]

A program for analyzing the performance of different scheduling algorithms was written. Random task sets with deadline varying from [Ci, Ti] (Plots 1, 2 and 3) was made, and the percentage of task sets schedulable using Rate Monotonic Scheduling algorithm, Deadline Monotonic Scheduling Algorithm and fixed priority scheduling algorithm with least slack time was calculated. The results can be seen in Plots 1-3, having the number of tasks per task set equal to 12, 24 and 48 respectively. It may be pointed out here that the "LST" in the plots stand for "Fixed priority scheduling with Least Slack Time", and not the Priority Driven LST algorithm.

The range of the deadline was then changed to [Ci + (Ti-Ci)/2, Ti], and the performance of the three algorithms was then analyzed. The results can be seen in Plots 4-6, having the number of tasks per task set equal to 12, 24 and 48 respectively.

We observe that if the deadline of a task is evenly distributed in [Ci, Ti], the performance of RM algorithm is inferior to the other two algorithms. The performance declines further

if we increase the number of tasks per task set. This can be because the 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. The DM algorithm has the best performance. It is the optimal fixed priority-scheduling algorithm. We can prove this by considering that a feasible, fixed priority-scheduling algorithm that is not DM can always be converted into DM scheduling algorithm by recursively swapping the priorities of two tasks, which have deadlines and priorities in reverse order. The resulting schedule is consistent with DM scheduling algorithm and is still schedulable. The Fixed Priority algorithm with least slack time also has a good performance, as it does consider the deadline of the tasks while assigning them their priorities. However, it is not optimal, and so the performance is not as good ad DM algorithm.

The performance of RM algorithm is much better if we set the deadline near to the period, as can be seen in the Plots 3-6. The performance of DM algorithm is still the best. However, the deadline of the tasks being close to the period improves the performance of RM. The performance of fixed priority scheduling with least slack time is better than RM, but not as good as DM.

As stated earlier, the DM algorithm is the optimal fixed priority-scheduling algorithm. If we choose the deadline of the tasks close to its period, the priority of the task assigned by RM is relatively similar to that assigned by DM. This causes the performance of RM to increase, as can be seen in the plots. Fixed priority scheduling algorithm assigns priorities using the slack time of the tasks. The priority assigned in this manner is similar to the priority assigned by DM, regardless of when the deadline is. So, its performance is similar to DM for all deadline distributions. However, it is not equal to the performance of DM because it sometimes assigns priorities in a different order, and can be seen in an example of the third task set used as an input in the analysis program.