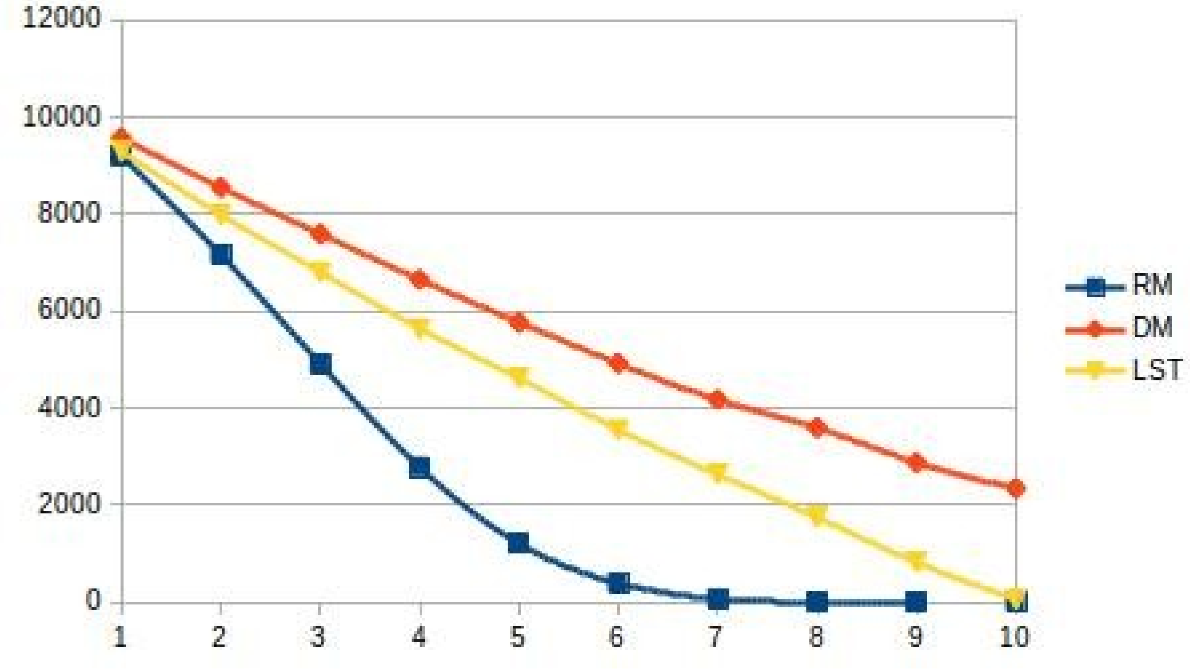
# Real time embedded System
# Assignment-2
# Algorithm Comparision

Shivanshu Shukla
1207888353

Plot 1:
12 tasks in each task set and the deadline distribution of [C i, ,T i ]



| plot1 | RM | DM | LST |
|---|---|---|---|
| 0.05 | 9193 | 9565 | 9290 |
| 0.15 | 7162 | 8547 | 7986 |
| 0.25 | 4915 | 7590 | 6805 |
| 0.35 | 2784 | 6650 | 5640 |
| 0.45 | 1210 | 5766 | 4614 |
| 0.55 | 395 | 4922 | 3557 |
| 0.65 | 75 | 4177 | 2635 |
| 0.75 | 6 | 3600 | 1750 |
| 0.85 | 0 | 2879 | 826 |
| 0.95 | 0 | 2346 | 68 |

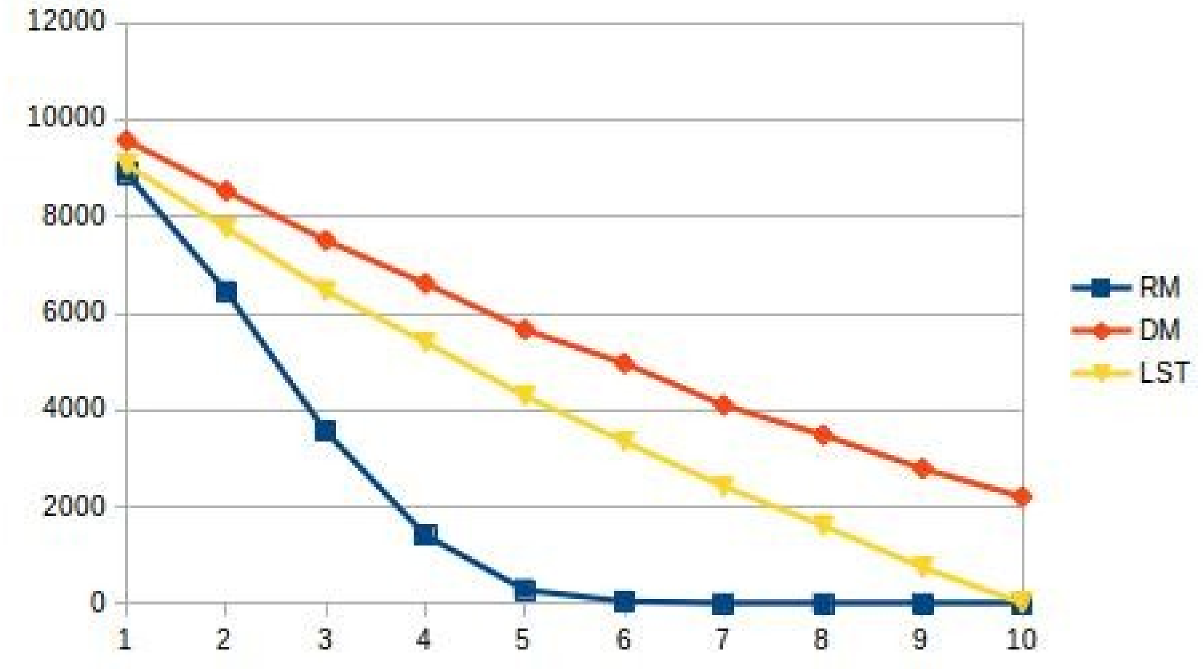
Rare monotonic : as utilization increases it decreases rapidly because here Deadline can be any where between Ci and Ti. If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases rapidly.

Least slack time: as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti. If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases.

Deadline monotonic : as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti. If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases. DM performs better than LST because it gives priority t tasks with earlier deadline, where as LST will schedule the task with least slack time in which case the tasks with similar slack time but will different deadline have probability to miss its deadline.

## Plot 2:

## 24 tasks in each task set and the deadline distribution of [C i, ,T i ]



| plot2 | RM | DM | LST |
|---|---|---|---|
| 0.05 | 8890 | 9569 | 9065 |
| 0.15 | 6450 | 8523 | 7761 |
| 0.25 | 3557 | 7506 | 6465 |
| 0.35 | 1418 | 6615 | 5402 |
| 0.45 | 292 | 5664 | 4295 |
| 0.55 | 56 | 4968 | 3351 |
| 0.65 | 1 | 4102 | 2422 |
| 0.75 | 0 | 3486 | 1616 |
| 0.85 | 0 | 2792 | 754 |
| 0.95 | 0 | 2210 | 21 |

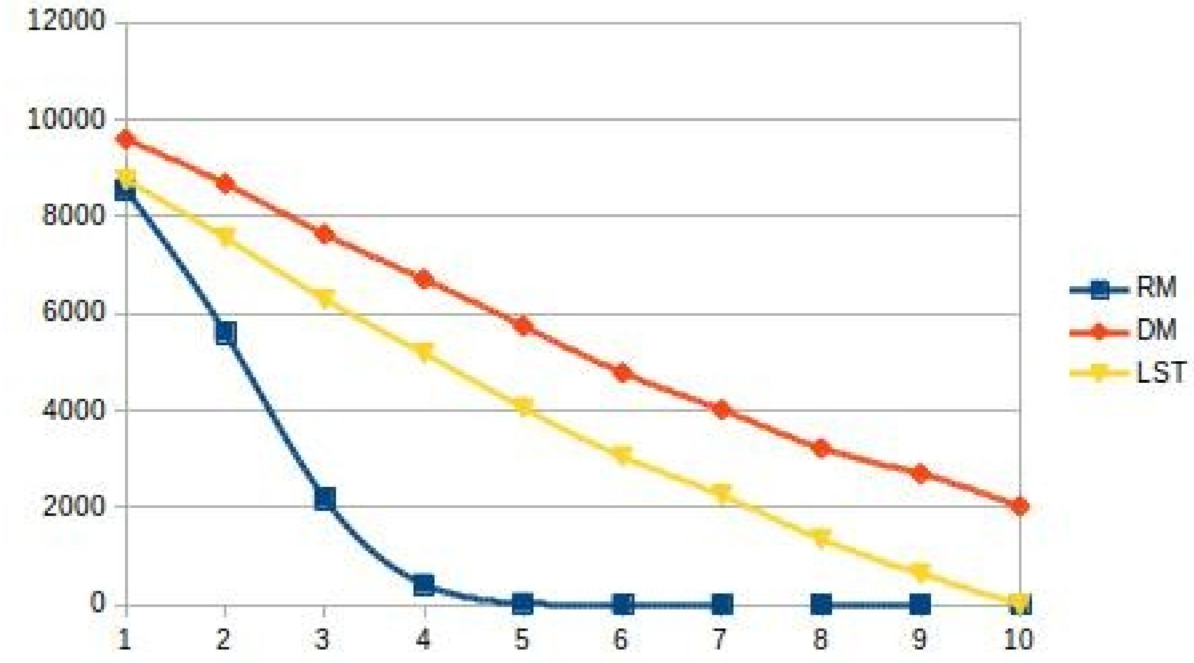
Rare monotonic : as utilization increases it decreases rapidly because here Deadline can be any where between Ci and Ti and the number of tasks are more than before making scheduling difficult through rate monotonic. If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases rapidly.

Least slack time: as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti, with increase in numbe of tasks performance is not effected drascically. If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases.

Deadline monotonic : as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti with increase in numbe of tasks performance is not effected drascically . If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases. DM performs better than LST because it gives priority t tasks with earlier deadline, where as LST will schedule the task with least slack time in which case the tasks with similar slack time but will different deadline have probability to miss its deadline.

## Plot 3:

48 tasks in each task set and the deadline distribution of [C i, ,T i ]



| plot3 | RM | DM | LST |
|---|---|---|---|
| 0.05 | 8556 | 9595 | 8752 |
| 0.15 | 5601 | 8676 | 7558 |
| 0.25 | 2190 | 7635 | 6296 |
| 0.35 | 427 | 6706 | 5180 |
| 0.45 | 36 | 5744 | 4055 |
| 0.55 | 1 | 4782 | 3054 |
| 0.65 | 0 | 4022 | 2258 |
| 0.75 | 0 | 3229 | 1356 |
| 0.85 | 0 | 2699 | 645 |
| 0.95 | 0 | 2032 | 6 |

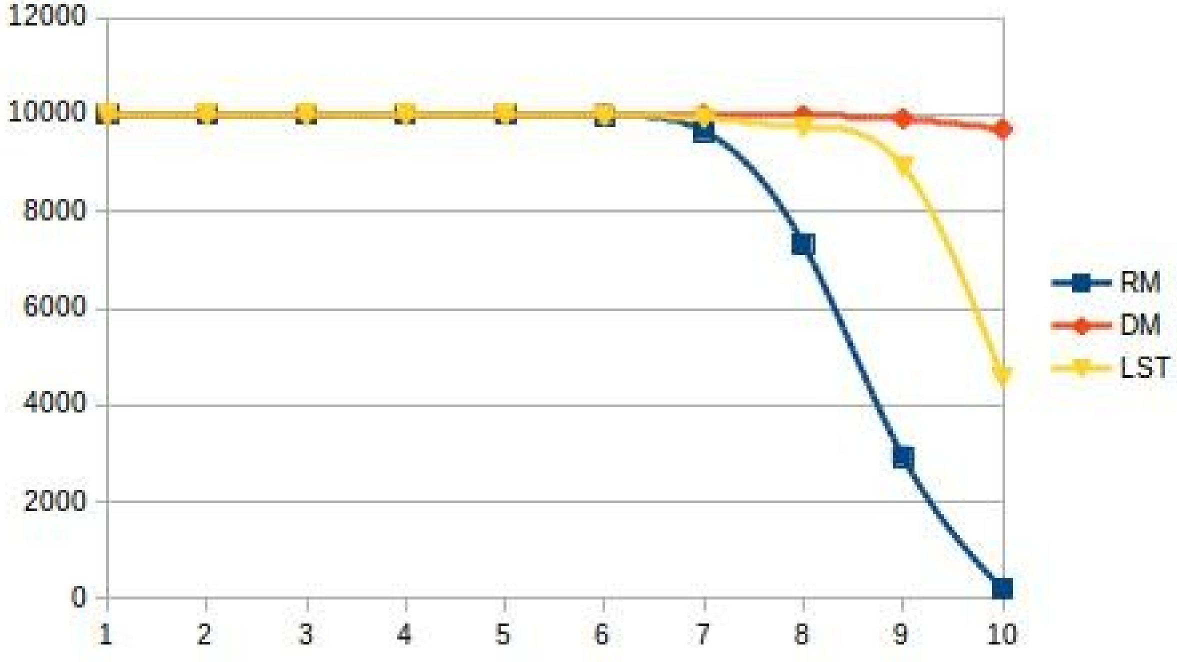
Rare monotonic : as utilization increases it decreases rapidly because here Deadline can be any where between Ci and Ti and the number of tasks are more than before making scheduling difficult through rate monotonic. If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases rapidly.

Least slack time: as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti, with increase in numbe of tasks performance is not effected drascically. If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases.

Deadline monotonic : as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti with increase in numbe of tasks performance is not effected drascically . If Di is slightly greater than Ci other tasks can not be scheduled hence number of schedulable tasks decreases. DM performs better than LST because it gives priority t tasks with earlier deadline, where as LST will schedule the task with least slack time in which case the tasks with similar slack time but will different deadline have probability to miss its deadline.

## Plot 4:

12 tasks in each task set and the deadline distribution of [C i +(T i -C i )/2 , ,T i ]



| plot4 | RM | DM | LST | |
|---|---|---|---|---|
| | 0.05 | 10000 | 10000 | 10000 |
| | 0.15 | 10000 | 10000 | 10000 |
| | 0.25 | 10000 | 10000 | 10000 |
| | 0.35 | 10000 | 10000 | 10000 |
| | 0.45 | 10000 | 10000 | 10000 |
| | 0.55 | 9990 | 10000 | 9981 |
| | 0.65 | 9657 | 9999 | 9930 |
| | 0.75 | 7324 | 9982 | 9742 |
| | 0.85 | 2933 | 9915 | 8935 |
| | 0.95 | 201 | 9701 | 4571 |

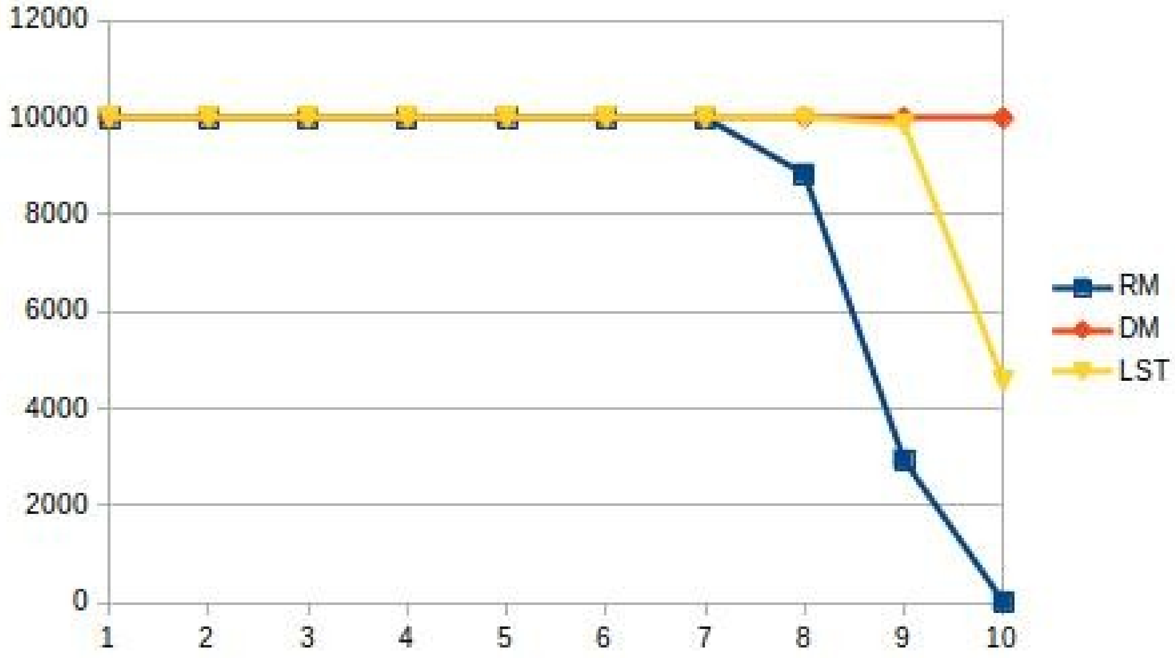
Rare monotonic : as utilization increases its schedulability decreases slowly initially there it falls rapidly becasuse here Deadline can be any where between (Ci+(Ti-Ci))/2 and Ti giving room to other high priority tasks to preempt and still be able to schedule. Rapid drop is due to upper bound of Rate monotonic.

Least slack time: as utilization increases number of schedulable tasks do. because here Deadline can be any where between Ci and Ti, with increase in number of tasks performance is not effected drascically.

Deadline monotonic : as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti with increase in number of tasks performance is not effected drascically . DM performs better than LST because it gives priority t tasks with earlier deadline, where as LST will schedule the task with least slack time in which case the tasks with similar slack time but will different deadline have probability to miss its deadline.

## Plot 5:

24 tasks in each task set and the deadline distribution of [C i +(T i -C i )/2 , ,T i ]



| plot5 | RM | DM | LST |
|---|---|---|---|
| 0.05 | 10000 | 10000 | 10000 |
| 0.15 | 10000 | 10000 | 10000 |
| 0.25 | 10000 | 10000 | 10000 |
| 0.35 | 10000 | 10000 | 10000 |
| 0.45 | 10000 | 10000 | 10000 |
| 0.55 | 10000 | 10000 | 10000 |
| 0.65 | 9988 | 10000 | 9998 |
| 0.75 | 8832 | 10000 | 9987 |
| 0.85 | 2941 | 9998 | 9865 |
| 0.95 | 27 | 9987 | 4561 |

Rare monotonic : as utilization increases its schedulability decreases slowly initially there it falls rapidly becasuse here Deadline can be any where between (Ci+(Ti-Ci))/2 and Ti giving room to other high priority tasks to preempt and still be able to schedule. Rapid drop is due to upper bound of Rate monotonic for 24 tasks.

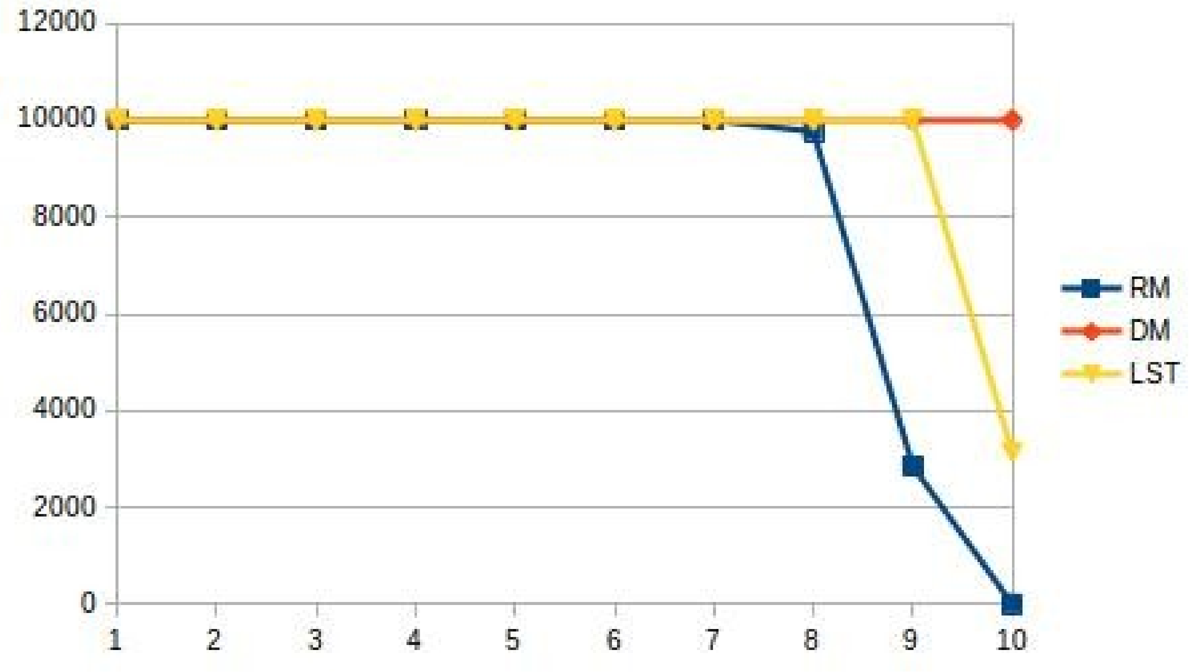
Least slack time: as utilization increases number of schedulable tasks do. because here Deadline can be any where between Ci and Ti, with increase in number of tasks performance is not effected drascically.

Deadline monotonic : as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti with increase in number of tasks performance is not effected drascically . DM performs better than LST because it gives priority t tasks with earlier deadline, where as LST will schedule the task with least slack time in which case the tasks with similar slack time but will different deadline have probability to miss its deadline.

## Plot 6:

48 tasks in each task set and the deadline distribution of [C i +(T i -C i )/2 , ,T i ]

| plot6 | RM | DM | LST |
|---|---|---|---|
| 0.05 | 10000 | 10000 | 10000 |
| 0.15 | 10000 | 10000 | 10000 |
| 0.25 | 10000 | 10000 | 10000 |
| 0.35 | 10000 | 10000 | 10000 |
| 0.45 | 10000 | 10000 | 10000 |
| 0.55 | 10000 | 10000 | 10000 |
| 0.65 | 10000 | 10000 | 10000 |
| 0.75 | 9762 | 10000 | 10000 |
| 0.85 | 2848 | 10000 | 9995 |
| 0.95 | 2 | 10000 | 3142 |



Rare monotonic : as utilization increases its schedulability decreases slowly initially there it falls rapidly becasuse here Deadline can be any where between (Ci+(Ti-Ci))/2 and Ti giving room to other high priority tasks to preempt and still be able to schedule. Rapid drop is due to upper bound of Rate monotonic for 24 tasks.

Least slack time: as utilization increases number of schedulable tasks do. because here Deadline can be any where between Ci and Ti, with increase in number of tasks performance is not effected drascically.

Deadline monotonic : as utilization increases number of schedulable tasks linearly decreases. because here Deadline can be any where between Ci and Ti with increase in number of tasks performance is not effected drascically . DM performs better than LST because it gives priority t tasks with earlier deadline, where as LST will schedule the task with least slack time in which case the tasks with similar slack time but will different deadline have probability to miss its deadline.

## Conclusion:

Deadline monotonic is the most optimal algorithm as it gives priority to task with earlier deadline.
Least slack time  does not perform as good as deadline monotonic because  there can be a task with smaller period and bigger slack time making it a lower priority task than the one with bigger period and smaller slack time. And is better than Rate monotonic because priority is dependent on slack time, giving priority to tasks having deadline closer to worst case execution time covering many failure cases of rate monotonic.

Rate monotonic fails in a few conditions as it gives priority to tasks with lower period rather than taking deadline into consideration leading to some tasks missing the deadline.