Optimal Selection of Preemption Points to Reduce Preemption Overhead

Venkannagari Shirisha Reddy (2020102054)

Kethu Sesha Sarath Reddy (2020102028)

Problem Statement:

- Considering real time systems, we need to have proper schedulability analysis which needs accurate evaluation of execution times.
- Execution time of task is dependent on the task code and input. Especially for the preemptive tasks, execution time is highly effected by preemption overheads. So, it is important to have proper estimation of execution time of preemptive tasks.
- For this purpose, there are timing tools but, most of them have pessimistic considerations leading to lose bound for WCET.
- This paper focuses on reducing preemption overheads by selecting optimal locations at which preemption can occur, which leads to decrease in WCET. Thus, increasing schedulability of the task set

Theory Proposed:

- Schedulability can be increased by using suitable scheduling algorithms to limit the number of preemptions as much as possible and to reduce preemption overhead.
- General approaches like the fully preemptive model create problems regarding overhead and WCET analysis and the fully non-preemptive model imposes large no of blockings on higher priority tasks which may result in deadline misses.
- Thus, hybrid preemption strategies are adopted and one such model is applied to reduce preemption overhead.
- Hybrid preemption strategy used in this problem includes deferring the preemption request by higher priority task until a point which results in less CRPD is reached (EPPs).
- Core idea of selecting these EPPs is discussed.

Proposed method:

Let the maximum time for which task t_i can execute non preemptively be Q_i units. Where WCET of any NPR consisting of δ_j ,δ_{j+1}, …., δ_k is q. For feasibility q < Q. where

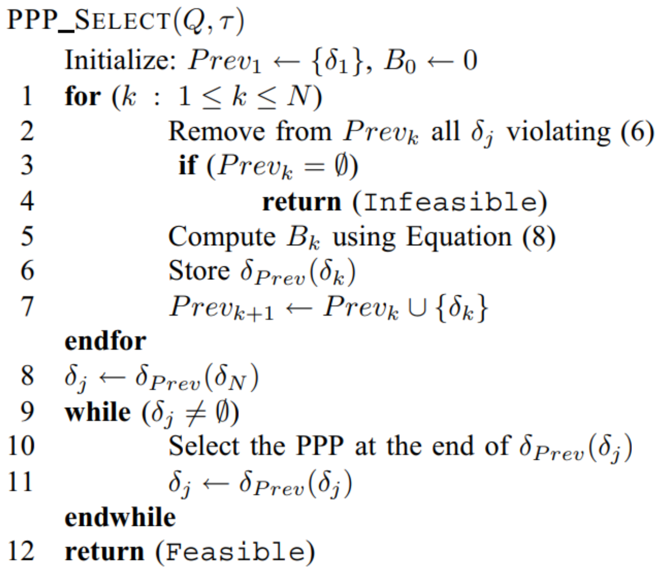
$$q = \xi_{j-1} + \sum_{\ell=j}^{k} b_\ell \quad , \quad B_k = B_{j-1} + q$$

Let B_k be WCET of first k BBs, including preemption overhead if there is EPPs in first k BBs. Therefore, with selection of EPPs, the possible WCET of the task t_i is B_N. Optimal selection of EPPs leads to minimum possibility of B_k's.

$$B_k = \min_{\delta_j \in Prev_k} \left\{ B_{j-1} + \xi_{j-1} + \sum_{\ell=j}^{k} b_\ell \right\} , \quad \delta^*(\delta_k) = \arg\min_{\delta_j \in Prev_k} \left\{ B_{j-1} + \xi_{j-1} + \sum_{\ell=j}^{k} b_\ell \right\}$$

Let $Prev_k$ be defined as the set of preceding BBs δ_j<=k that satisfy the feasibility condition (q<Q) and let $\delta_{Prev}(\delta_k)$ be the basic block preceding $\delta^*(\delta_k)$. Such that the last EPP will be at the end of $\delta_{Prev}(\delta_N)$, penultimate EPP will be $\delta_{Prev}(\delta_{Prev}(\delta_N))$ and so on.

The Algorithm for the optimal selection of PPPs of a task is as below



```
PPP_SELECT($Q, \tau$)
    Initialize: $Prev_1 \leftarrow \{\delta_1\}$, $B_0 \leftarrow 0$
    for ($k$ : $1 \leq k \leq N$)
        Remove from $Prev_k$ all $\delta_j$ violating (6)
        if ($Prev_k = \emptyset$)
            return (Infeasible)
        Compute $B_k$ using Equation (8)
        Store $\delta_{Prev}(\delta_k)$
        $Prev_{k+1} \leftarrow Prev_k \cup \{\delta_k\}$
    endfor
    $\delta_j \leftarrow \delta_{Prev}(\delta_N)$
    while ($\delta_j \neq \emptyset$)
        Select the PPP at the end of $\delta_{Prev}(\delta_j)$
        $\delta_j \leftarrow \delta_{Prev}(\delta_j)$
    endwhile
    return (Feasible)
```

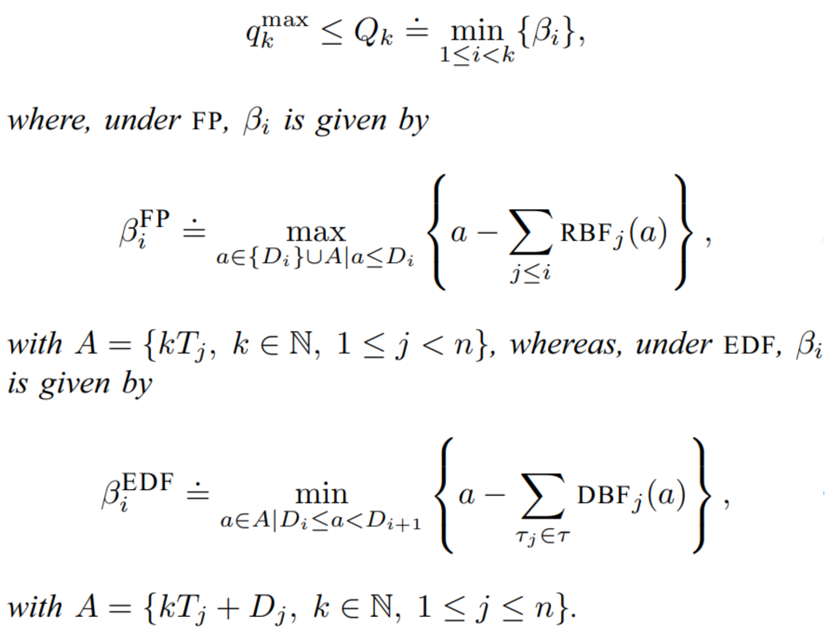
Implementation:

In the code that we have submitted the implementation of the above algorithm is done under the name of the function "PPP_SELECT(Q, T)" where for every basic block Bk is computed using the function "Compute_Bk(Prev,b,OH,B,Q)" where time complexity of the PPP_SELECT function is O(N) where N total number of the basic blocks.

Task set is generated by modifying a task generation algorithm given by UUniFast algorithm under the name of the function "gen_ripoll(nsets, MaxCompute, MaxSlack, Maxdelay, target_util)". Where

- Computation time of each job is taken as a random number between 100 and a user given max computation time of a task in multiples of 10.
- Deadline is considered as computation time plus a random number between 0 and a user given max slack time of a task in multiples of 10.
- Period is considered as deadline plus a random number between 0 and a user given max delay after deadline of a task in multiples of 10

We computed Q such that there is atleast one basic block present in non-pre-emptive region. But in the paper Q is computed by considering feasibility into account for EDF and FP. Where for all k | 1 < k ≤ n + 1



$$q_k^{\max} \leq Q_k \doteq \min_{1 \leq i < k} \{\beta_i\},$$

where, under FP, β_i *is given by*

$$\beta_i^{\text{FP}} \doteq \max_{a \in \{D_i\} \cup A | a \leq D_i} \left\{ a - \sum_{j \leq i} \text{RBF}_j(a) \right\},$$

with $A = \{kT_j,\ k \in \mathbb{N},\ 1 \leq j < n\}$, *whereas, under* EDF, β_i *is given by*

$$\beta_i^{\text{EDF}} \doteq \min_{a \in A | D_i \leq a < D_{i+1}} \left\{ a - \sum_{\tau_j \in \tau} \text{DBF}_j(a) \right\},$$

with $A = \{kT_j + D_j,\ k \in \mathbb{N},\ 1 \leq j \leq n\}$.

But here with our implementation, for most of the tasks generated, the set of values for ‘a’ were empty which results in no update of Q value which led the entire task set from that point onwards leading into non feasible mode. To be in detail, for example our task set is generated in such a way that deadlines are randomized and to schedule for EDF we arranged the set of tasks in priority order based on deadline values. And here the consecutive deadlines are being noticeably short ranged, and the value set ‘A’ doesn’t have any values in that range.

We computed Overhead (CRPD) values using a random function which gives values between 1 and 10. But in the paper CRPD values are computed using a priori by timing analysis tools (but not mentioned clearly).

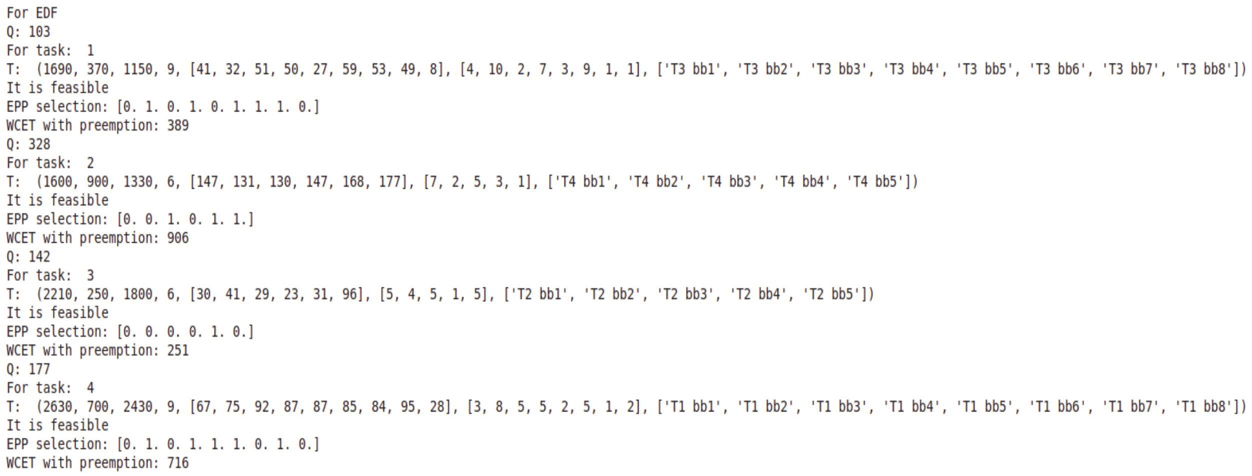
Experimentation:

Experiment is done to check working of implemented algorithm on many generated tasks sets, few of them are shown below:

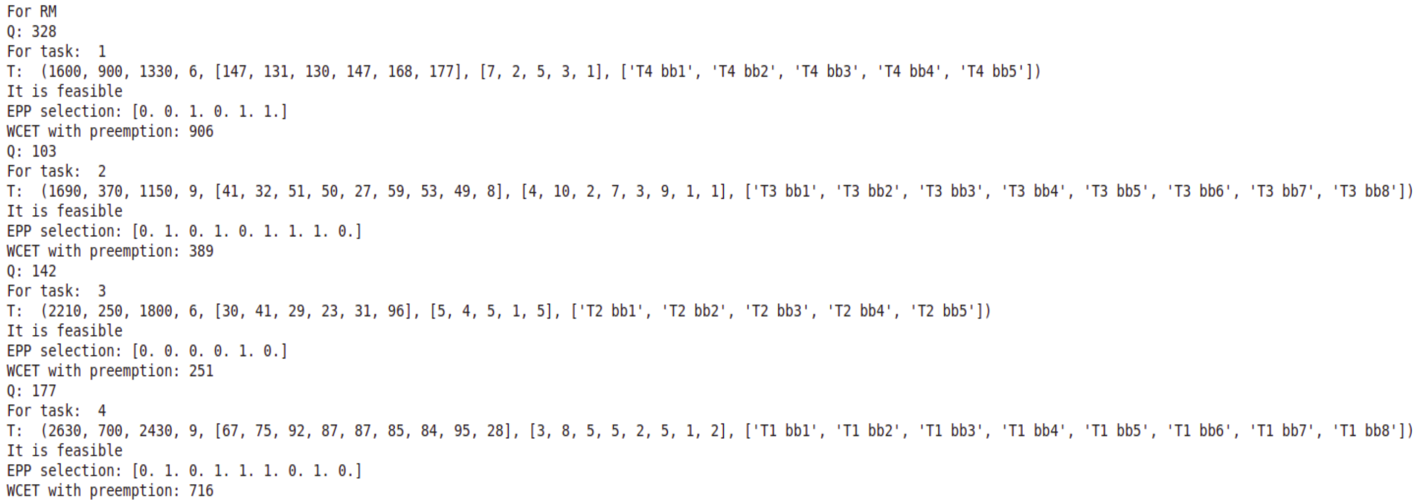
Notation: T = (period, execution time, deadline, no.of BBs, length of each BB, overhead at each PPP, Block indications)

- For generated task set t1: [(2630, 700, 2430, 9, [67, 75, 92, 87, 87, 85, 84, 95, 28], [3, 8, 5, 5, 2, 5, 1, 2], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8']), (2210, 250, 1800, 6, [30, 41, 29, 23, 31, 96], [5, 4, 5, 1, 5], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5']), (1690, 370, 1150, 9, [41, 32, 51, 50, 27, 59, 53, 49, 8], [4, 10, 2, 7, 3, 9, 1, 1], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8']), (1600, 900, 1330, 6, [147, 131, 130, 147, 168, 177], [7, 2, 5, 3, 1], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5'])]

Result obtained:



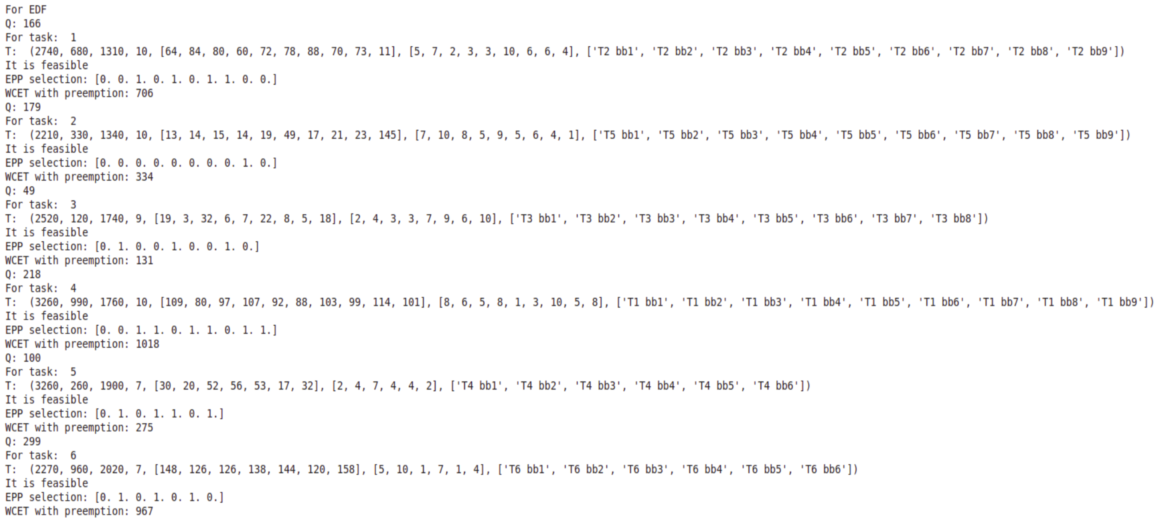
```
For EDF
Q: 103
For task:  1
T:  (1690, 370, 1150, 9, [41, 32, 51, 50, 27, 59, 53, 49, 8], [4, 10, 2, 7, 3, 9, 1, 1], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8'])
It is feasible
EPP selection: [0. 1. 0. 1. 0. 1. 1. 1. 0.]
WCET with preemption: 389
Q: 328
For task:  2
T:  (1600, 900, 1330, 6, [147, 131, 130, 147, 168, 177], [7, 2, 5, 3, 1], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5'])
It is feasible
EPP selection: [0. 0. 1. 0. 1. 1.]
WCET with preemption: 906
Q: 142
For task:  3
T:  (2210, 250, 1800, 6, [30, 41, 29, 23, 31, 96], [5, 4, 5, 1, 5], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5'])
It is feasible
EPP selection: [0. 0. 0. 0. 1. 0.]
WCET with preemption: 251
Q: 177
For task:  4
T:  (2630, 700, 2430, 9, [67, 75, 92, 87, 87, 85, 84, 95, 28], [3, 8, 5, 5, 2, 5, 1, 2], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8'])
It is feasible
EPP selection: [0. 1. 0. 1. 1. 1. 0. 1. 0.]
WCET with preemption: 716
```



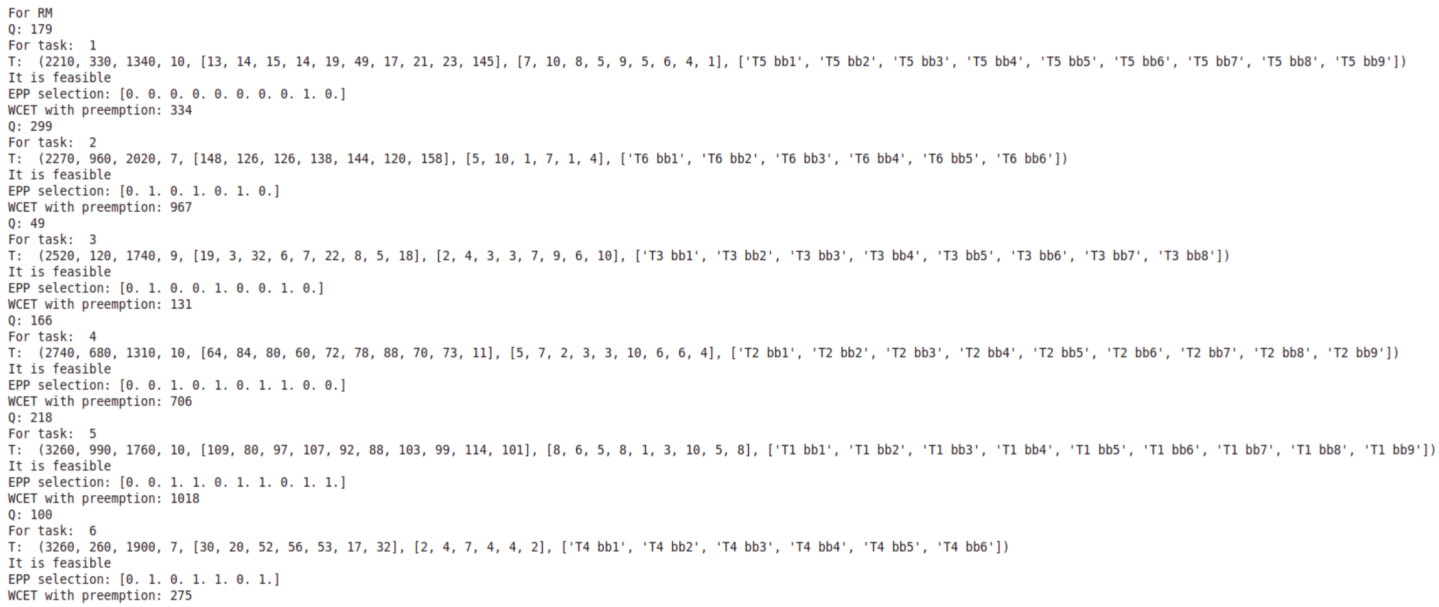
```
For RM
Q: 328
For task:  1
T:  (1600, 900, 1330, 6, [147, 131, 130, 147, 168, 177], [7, 2, 5, 3, 1], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5'])
It is feasible
EPP selection: [0. 0. 1. 0. 1. 1.]
WCET with preemption: 906
Q: 103
For task:  2
T:  (1690, 370, 1150, 9, [41, 32, 51, 50, 27, 59, 53, 49, 8], [4, 10, 2, 7, 3, 9, 1, 1], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8'])
It is feasible
EPP selection: [0. 1. 0. 1. 0. 1. 1. 1. 0.]
WCET with preemption: 389
Q: 142
For task:  3
T:  (2210, 250, 1800, 6, [30, 41, 29, 23, 31, 96], [5, 4, 5, 1, 5], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5'])
It is feasible
EPP selection: [0. 0. 0. 0. 1. 0.]
WCET with preemption: 251
Q: 177
For task:  4
T:  (2630, 700, 2430, 9, [67, 75, 92, 87, 87, 85, 84, 95, 28], [3, 8, 5, 5, 2, 5, 1, 2], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8'])
It is feasible
EPP selection: [0. 1. 0. 1. 1. 1. 0. 1. 0.]
WCET with preemption: 716
```

- For generated task set t2: [(3260, 990, 1760, 10, [109, 80, 97, 107, 92, 88, 103, 99, 114, 101], [8, 6, 5, 8, 1, 3, 10, 5, 8], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8', 'T1 bb9']), (2740, 680, 1310, 10, [64, 84, 80, 60, 72, 78, 88, 70, 73, 11], [5, 7, 2, 3, 3, 10, 6, 6, 4], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5', 'T2 bb6', 'T2 bb7', 'T2 bb8', 'T2 bb9']), (2520, 120, 1740, 9, [19, 3, 32, 6, 7, 22, 8, 5, 18], [2, 4, 3, 3, 7, 9, 6, 10], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8']), (3260, 260, 1900, 7, [30, 20, 52, 56, 53, 17, 32], [2, 4, 7, 4, 4, 2], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5', 'T4 bb6']), (2210, 330, 1340, 10, [13, 14, 15, 14, 19, 49, 17, 21, 23, 145], [7, 10, 8, 5, 9, 5, 6, 4, 1], ['T5 bb1', 'T5 bb2', 'T5 bb3', 'T5 bb4', 'T5 bb5', 'T5 bb6', 'T5 bb7', 'T5 bb8', 'T5 bb9']), (2270, 960, 2020, 7, [148, 126, 126, 138, 144, 120, 158], [5, 10, 1, 7, 1, 4], ['T6 bb1', 'T6 bb2', 'T6 bb3', 'T6 bb4', 'T6 bb5', 'T6 bb6'])]

Result obtained:



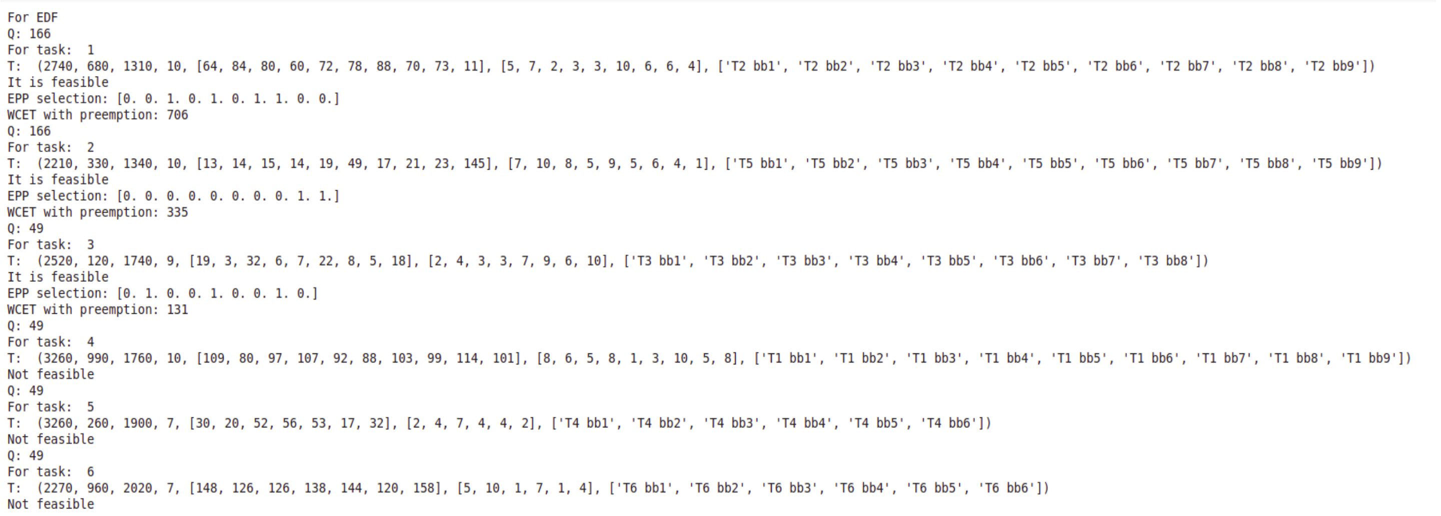
```
For EDF
Q: 166
For task:  1
T:  (2740, 680, 1310, 10, [64, 84, 80, 60, 72, 78, 88, 70, 73, 11], [5, 7, 2, 3, 3, 10, 6, 6, 4], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5', 'T2 bb6', 'T2 bb7', 'T2 bb8', 'T2 bb9'])
It is feasible
EPP selection: [0. 0. 1. 0. 1. 0. 1. 1. 0. 0.]
WCET with preemption: 706
Q: 179
For task:  2
T:  (2210, 330, 1340, 10, [13, 14, 15, 14, 19, 49, 17, 21, 23, 145], [7, 10, 8, 5, 9, 5, 6, 4, 1], ['T5 bb1', 'T5 bb2', 'T5 bb3', 'T5 bb4', 'T5 bb5', 'T5 bb6', 'T5 bb7', 'T5 bb8', 'T5 bb9'])
It is feasible
EPP selection: [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
WCET with preemption: 334
Q: 49
For task:  3
T:  (2520, 120, 1740, 9, [19, 3, 32, 6, 7, 22, 8, 5, 18], [2, 4, 3, 3, 7, 9, 6, 10], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8'])
It is feasible
EPP selection: [0. 1. 0. 0. 1. 0. 0. 1. 0.]
WCET with preemption: 131
Q: 218
For task:  4
T:  (3260, 990, 1760, 10, [109, 80, 97, 107, 92, 88, 103, 99, 114, 101], [8, 6, 5, 8, 1, 3, 10, 5, 8], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8', 'T1 bb9'])
It is feasible
EPP selection: [0. 0. 1. 1. 0. 1. 1. 0. 1. 1.]
WCET with preemption: 1018
Q: 100
For task:  5
T:  (3260, 260, 1900, 7, [30, 20, 52, 56, 53, 17, 32], [2, 4, 7, 4, 4, 2], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5', 'T4 bb6'])
It is feasible
EPP selection: [0. 1. 0. 1. 1. 0. 1.]
WCET with preemption: 275
Q: 299
For task:  6
T:  (2270, 960, 2020, 7, [148, 126, 126, 138, 144, 120, 158], [5, 10, 1, 7, 1, 4], ['T6 bb1', 'T6 bb2', 'T6 bb3', 'T6 bb4', 'T6 bb5', 'T6 bb6'])
It is feasible
EPP selection: [0. 1. 0. 1. 0. 1. 0.]
WCET with preemption: 967
```



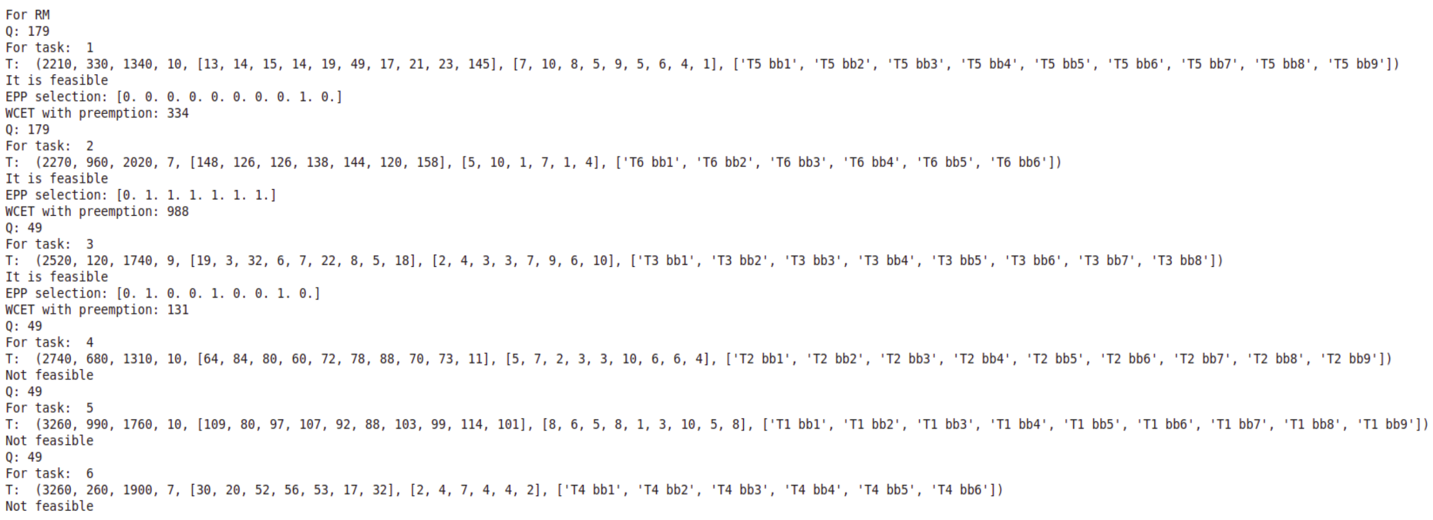
```
For RM
Q: 179
For task:  1
T:  (2210, 330, 1340, 10, [13, 14, 15, 14, 19, 49, 17, 21, 23, 145], [7, 10, 8, 5, 9, 5, 6, 4, 1], ['T5 bb1', 'T5 bb2', 'T5 bb3', 'T5 bb4', 'T5 bb5', 'T5 bb6', 'T5 bb7', 'T5 bb8', 'T5 bb9'])
It is feasible
EPP selection: [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
WCET with preemption: 334
Q: 299
For task:  2
T:  (2270, 960, 2020, 7, [148, 126, 126, 138, 144, 120, 158], [5, 10, 1, 7, 1, 4], ['T6 bb1', 'T6 bb2', 'T6 bb3', 'T6 bb4', 'T6 bb5', 'T6 bb6'])
It is feasible
EPP selection: [0. 1. 0. 1. 0. 1. 0.]
WCET with preemption: 967
Q: 49
For task:  3
T:  (2520, 120, 1740, 9, [19, 3, 32, 6, 7, 22, 8, 5, 18], [2, 4, 3, 3, 7, 9, 6, 10], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8'])
It is feasible
EPP selection: [0. 1. 0. 0. 1. 0. 0. 1. 0.]
WCET with preemption: 131
Q: 166
For task:  4
T:  (2740, 680, 1310, 10, [64, 84, 80, 60, 72, 78, 88, 70, 73, 11], [5, 7, 2, 3, 3, 10, 6, 6, 4], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5', 'T2 bb6', 'T2 bb7', 'T2 bb8', 'T2 bb9'])
It is feasible
EPP selection: [0. 0. 1. 0. 1. 0. 1. 1. 0. 0.]
WCET with preemption: 706
Q: 218
For task:  5
T:  (3260, 990, 1760, 10, [109, 80, 97, 107, 92, 88, 103, 99, 114, 101], [8, 6, 5, 8, 1, 3, 10, 5, 8], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8', 'T1 bb9'])
It is feasible
EPP selection: [0. 0. 1. 1. 0. 1. 1. 0. 1. 1.]
WCET with preemption: 1018
Q: 100
For task:  6
T:  (3260, 260, 1900, 7, [30, 20, 52, 56, 53, 17, 32], [2, 4, 7, 4, 4, 2], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5', 'T4 bb6'])
It is feasible
EPP selection: [0. 1. 0. 1. 1. 0. 1.]
WCET with preemption: 275
```

- For generated task set t3: [(3260, 990, 1760, 10, [109, 80, 97, 107, 92, 88, 103, 99, 114, 101], [8, 6, 5, 8, 1, 3, 10, 5, 8], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8', 'T1 bb9']), (2740, 680, 1310, 10, [64, 84, 80, 60, 72, 78, 88, 70, 73, 11], [5, 7, 2, 3, 3, 10, 6, 6, 4], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5', 'T2 bb6', 'T2 bb7', 'T2 bb8', 'T2 bb9']), (2520, 120, 1740, 9, [19, 3, 32, 6, 7, 22, 8, 5, 18], [2, 4, 3, 3, 7, 9, 6, 10], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8']), (3260, 260, 1900, 7, [30, 20, 52, 56, 53, 17, 32], [2, 4, 7, 4, 4, 2], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5', 'T4 bb6']), (2210, 330, 1340, 10, [13, 14, 15, 14, 19, 49, 17, 21, 23, 145], [7, 10, 8, 5, 9, 5, 6, 4, 1], ['T5 bb1', 'T5 bb2', 'T5 bb3', 'T5 bb4', 'T5 bb5', 'T5 bb6', 'T5 bb7', 'T5 bb8', 'T5 bb9']), (2270, 960, 2020, 7, [148, 126, 126, 138, 144, 120, 158], [5, 10, 1, 7, 1, 4], ['T6 bb1', 'T6 bb2', 'T6 bb3', 'T6 bb4', 'T6 bb5', 'T6 bb6'])]

Result obtained (Here we are getting not feasible cases)



```
For EDF
Q: 166
For task:  1
T:  (2740, 680, 1310, 10, [64, 84, 80, 60, 72, 78, 88, 70, 73, 11], [5, 7, 2, 3, 3, 10, 6, 6, 4], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5', 'T2 bb6', 'T2 bb7', 'T2 bb8', 'T2 bb9'])
It is feasible
EPP selection: [0. 0. 1. 0. 1. 0. 1. 1. 0. 0.]
WCET with preemption: 706
Q: 166
For task:  2
T:  (2210, 330, 1340, 10, [13, 14, 15, 14, 19, 49, 17, 21, 23, 145], [7, 10, 8, 5, 9, 5, 6, 4, 1], ['T5 bb1', 'T5 bb2', 'T5 bb3', 'T5 bb4', 'T5 bb5', 'T5 bb6', 'T5 bb7', 'T5 bb8', 'T5 bb9'])
It is feasible
EPP selection: [0. 0. 0. 0. 0. 0. 0. 0. 1. 1.]
WCET with preemption: 335
Q: 49
For task:  3
T:  (2520, 120, 1740, 9, [19, 3, 32, 6, 7, 22, 8, 5, 18], [2, 4, 3, 3, 7, 9, 6, 10], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8'])
It is feasible
EPP selection: [0. 1. 0. 0. 1. 0. 0. 1. 0.]
WCET with preemption: 131
Q: 49
For task:  4
T:  (3260, 990, 1760, 10, [109, 80, 97, 107, 92, 88, 103, 99, 114, 101], [8, 6, 5, 8, 1, 3, 10, 5, 8], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8', 'T1 bb9'])
Not feasible
Q: 49
For task:  5
T:  (3260, 260, 1900, 7, [30, 20, 52, 56, 53, 17, 32], [2, 4, 7, 4, 4, 2], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5', 'T4 bb6'])
Not feasible
Q: 49
For task:  6
T:  (2270, 960, 2020, 7, [148, 126, 126, 138, 144, 120, 158], [5, 10, 1, 7, 1, 4], ['T6 bb1', 'T6 bb2', 'T6 bb3', 'T6 bb4', 'T6 bb5', 'T6 bb6'])
Not feasible
```



```
For RM
Q: 179
For task:  1
T:  (2210, 330, 1340, 10, [13, 14, 15, 14, 19, 49, 17, 21, 23, 145], [7, 10, 8, 5, 9, 5, 6, 4, 1], ['T5 bb1', 'T5 bb2', 'T5 bb3', 'T5 bb4', 'T5 bb5', 'T5 bb6', 'T5 bb7', 'T5 bb8', 'T5 bb9'])
It is feasible
EPP selection: [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
WCET with preemption: 334
Q: 179
For task:  2
T:  (2270, 960, 2020, 7, [148, 126, 126, 138, 144, 120, 158], [5, 10, 1, 7, 1, 4], ['T6 bb1', 'T6 bb2', 'T6 bb3', 'T6 bb4', 'T6 bb5', 'T6 bb6'])
It is feasible
EPP selection: [0. 1. 1. 1. 1. 1. 1.]
WCET with preemption: 988
Q: 49
For task:  3
T:  (2520, 120, 1740, 9, [19, 3, 32, 6, 7, 22, 8, 5, 18], [2, 4, 3, 3, 7, 9, 6, 10], ['T3 bb1', 'T3 bb2', 'T3 bb3', 'T3 bb4', 'T3 bb5', 'T3 bb6', 'T3 bb7', 'T3 bb8'])
It is feasible
EPP selection: [0. 1. 0. 0. 1. 0. 0. 1. 0.]
WCET with preemption: 131
Q: 49
For task:  4
T:  (2740, 680, 1310, 10, [64, 84, 80, 60, 72, 78, 88, 70, 73, 11], [5, 7, 2, 3, 3, 10, 6, 6, 4], ['T2 bb1', 'T2 bb2', 'T2 bb3', 'T2 bb4', 'T2 bb5', 'T2 bb6', 'T2 bb7', 'T2 bb8', 'T2 bb9'])
Not feasible
Q: 49
For task:  5
T:  (3260, 990, 1760, 10, [109, 80, 97, 107, 92, 88, 103, 99, 114, 101], [8, 6, 5, 8, 1, 3, 10, 5, 8], ['T1 bb1', 'T1 bb2', 'T1 bb3', 'T1 bb4', 'T1 bb5', 'T1 bb6', 'T1 bb7', 'T1 bb8', 'T1 bb9'])
Not feasible
Q: 49
For task:  6
T:  (3260, 260, 1900, 7, [30, 20, 52, 56, 53, 17, 32], [2, 4, 7, 4, 4, 2], ['T4 bb1', 'T4 bb2', 'T4 bb3', 'T4 bb4', 'T4 bb5', 'T4 bb6'])
Not feasible
```

Results and discussion:

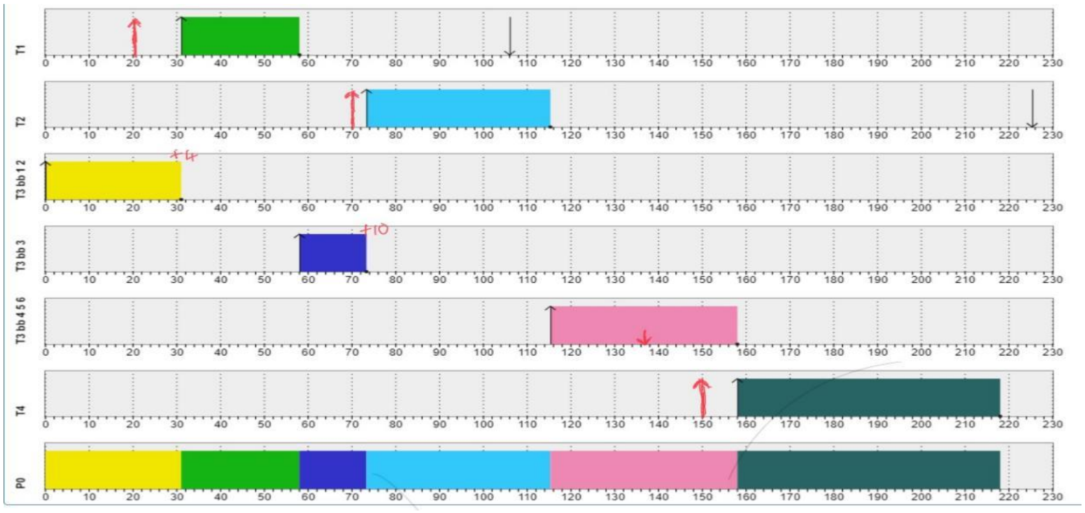
Notation:(period, computation time, relative deadline, no of basic blocks, lengths of BBs, preemption overheads at each PPP's)

For Task set

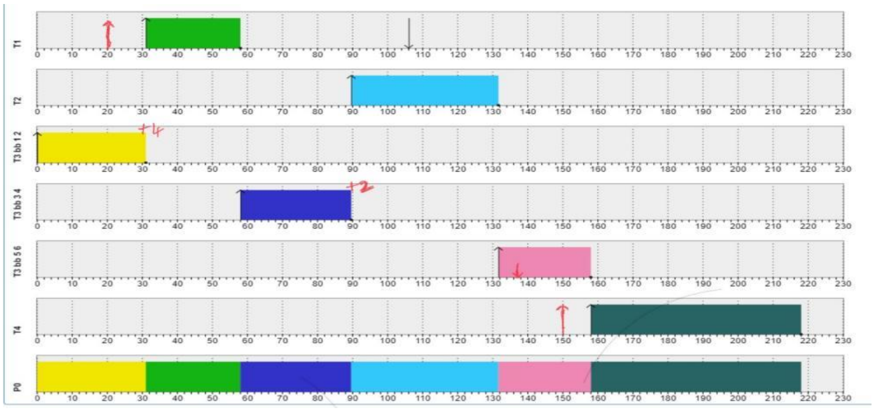
# ([1250, 270, 750, 5, [46, 36, 69, 39, 80], [7, 9, 10, 10]] , [1560, 420, 1520, 10, [36, 33, 53, 57, 38, 52, 58, 31, 42, 20], [3, 4, 3, 9, 8, 5, 1, 10, 5]] , [1770, 890, 1370, 6, [149, 161, 153, 163, 141, 123], [6, 4, 10, 2, 8]] , [2150, 600, 860, 7, [89, 103, 88, 76, 74, 105, 65], [6, 2, 5, 7, 7, 5]]).

Arrival times: A1 = 200, A2 = 700, A3 = 0, A4 = 1500

Preemption takes place at PPP:



Preemption takes place at EPP:



We can clearly see the overhead added while preemption takes place at EPP's is lesser than the overhead added while preemption takes place at PPP's.

Also, from the experimental results we can observe that there can be chances of task set being not feasible based on the values of Q.

However, our main aim is to reduce preemption overhead this can be achieved optimally with our algorithm.

Conclusion:

- Our method relaxes the assumption of taking equal preemption costs throughout the code, accounting for more realistic scenario.
- Smart preemption point selection can significantly reduce the WCET, cache related delays and context switch time of each task.
- Finds a much more efficient solution that has a linear complexity both in space and time. (Assumptions considered).
- If a feasible schedule is not found by the proposed method, then no other strategy can lead to a feasible solution. (Assumptions considered)

Expressing the preemption overhead as a function of the preempting task would significantly complicate the analysis and to avoid this, following assumptions are considered

A1: - The cache is cold after each context switch (fixed overhead)

A2: - Each EPP leads to a preemption.

However, the above assumptions are pessimistic.

A1: Preempting task may have a smaller footprint than the overhead resulted by this assumption.

A2: There might be cases in which not all EPPs lead to preemption.