Real-time Scheduling

Main Results on Periodic Task Scheduling
Homework

**Median**
HW1: 20
HW2: 9
Main Results in Real-time Scheduling of Periodic Tasks

Periodic Task Scheduling

- Rate Monotonic
  - Bound
  - Optimality
- EDF
  - Bound
  - Optimality
Earliest Deadline First (EDF) Optimality Result

- EDF is the optimal dynamic priority scheduling policy
  - It can meet all deadlines whenever the processor utilization is less than 100%
  - Intuition:
    - You have HW1 due tomorrow and HW2 due the day after, which one do you do first?
    - If you started with HW2 and met both deadlines you could have started with HW1 (in EDF order) and still met both deadlines
    - EDF can meet deadlines whenever anyone else can

![Diagram showing HW2 and HW1 with deadlines](image)
Earliest Deadline First (EDF) Optimality Result

- EDF is the optimal dynamic priority scheduling policy
  - It can meet all deadlines whenever the processor utilization is less than 100%
  - Intuition:
    - You have HW1 due tomorrow and HW2 due the day after, which one do you do first?
    - If you started with HW2 and met both deadlines you could have started with HW1 (in EDF order) and still met both deadlines
    - EDF can meet deadlines whenever anyone else can

Non-EDF Ok $\rightarrow$ EDF OK!

<table>
<thead>
<tr>
<th>HW1</th>
<th>HW2</th>
</tr>
</thead>
</table>

Deadline HW1

Deadline HW2
When can EDF Meet Deadlines?

- Consider a task set where:

  \[ \sum_i \frac{C_i}{P_i} = 1 \]

- Imagine a policy that reserves for each task \( i \) a fraction \( f_i \) of each clock tick, where \( f_i = \frac{C_i}{P_i} \)
Utilization Bound of EDF

- Imagine a policy that reserves for each task $i$ a fraction $f_i$ of each time unit, where $f_i = C_i/P_i$.

- This policy meets all deadlines, because within each period $P_i$ it reserves for task $i$ a total time $f_i P_i = (C_i / P_i) P_i = C_i$ (i.e., enough to finish).
Utilization Bound of EDF

- Pick any two execution chunks that are not in EDF order and swap them
Utilization Bound of EDF

- Pick any two execution chunks that are not in EDF order and swap them
- Still meets deadlines!
Utilization Bound of EDF

- Pick any two execution chunks that are not in EDF order and swap them

- Still meets deadlines!
- Repeat swap until all in EDF order
  → EDF meets deadlines
Rate Monotonic Scheduling

Rate monotonic scheduling is the optimal fixed-priority scheduling policy for periodic tasks (with period = deadline).
The Worst-Case Scenario

- Consider the worst case where all tasks arrive at the same time.

- If any fixed priority scheduling policy can meet deadline, rate monotonic can!
Optimality of Rate Monotonic

- If any other policy can meet deadlines so can RM

Policy X meets deadlines?
Optimality of Rate Monotonic

- If any other policy can meet deadlines so can RM

Policy X meets deadlines? \(\text{YES}\) 
\[\rightarrow \text{RM meets deadlines}\]
Utilization Bounds

- Intuitively:
  - The lower the processor utilization, $U$, the easier it is to meet deadlines.
  - The higher the processor utilization, $U$, the more difficult it is to meet deadlines.

- Question: is there a threshold $U_{\text{bound}}$ such that
  - When $U < U_{\text{bound}}$ deadlines are met
  - When $U > U_{\text{bound}}$ deadlines are missed
Example
(Rate-Monotonic Scheduling)

Task 1
\[ P_1 = 2 \]
\[ C_1 = 1 \]

Task 2
\[ P_2 = 3 \]
\[ C_2 = 1.01 \]

\[
U = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{1}{2} + \frac{1.01}{3} \approx 83.3\% 
\]

Question: is there a threshold \( U_{bound} \) such that
- When \( U < U_{bound} \) deadlines are met
- When \( U > U_{bound} \) deadlines are missed
Example
(Rate-Monotonic Scheduling)

Task 1
\[ P_1 = 2 \]
\[ C_1 = 1 \]

Task 2
\[ P_2 = 3 \]
\[ C_2 = 1.01 \]

\[ U = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{1}{2} + \frac{1.01}{3} \approx 83.3\% \]

- Question: is there a threshold \( U_{\text{bound}} \) such that
  - When \( U < U_{\text{bound}} \) deadlines are met
  - When \( U > U_{\text{bound}} \) deadlines are missed
Example
(Rate-Monotonic Scheduling)

Task 1
\( P_1 = 2 \)
\( C_1 = 1 \)

Task 2
\( P_2 = 3 \)
\( C_2 = 1.01 \)

\[ U = \frac{C_1}{P_1} + \frac{C_2}{P_2} + \frac{1}{2} + \frac{1.01}{3} \approx 83.3\% \]

Question: is there a threshold \( U_{\text{bound}} \) such that
- When \( U < U_{\text{bound}} \) deadlines are met
- When \( U > U_{\text{bound}} \) deadlines are missed
Another Example (Rate-Monotonic Scheduling)

Task 1
\(P_1 = 2\)
\(C_1 = 1\)

Task 2
\(P_2 = 6\)
\(C_2 = 2.4\)

\[ U = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{1}{2} + \frac{2.4}{6} = 0.90 = 90\% \]

Question: is there a threshold \(U_{\text{bound}}\) such that
- When \(U < U_{\text{bound}}\) deadlines are met
- When \(U > U_{\text{bound}}\) deadlines are missed
Another Example
(Rate-Monotonic Scheduling)

Task 1
$P_1 = 2$
$C_1 = 1$

Task 2
$P_2 = 6$
$C_2 = 2.4$

$U = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{1}{2} + \frac{2.4}{6} = 90\%$

Question: is there a threshold $U_{\text{bound}}$ such that
- When $U < U_{\text{bound}}$ deadlines are met
- When $U > U_{\text{bound}}$ deadlines are missed
Another Example
(Rate-Monotonic Scheduling)

Task 1
\[ P_1 = 2 \]
\[ C_1 = 1 \]

Task 2
\[ P_2 = 6 \]
\[ C_2 = 2.4 \]

\[ U = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{1}{2} + \frac{2.4}{6} = 90\% \]

Question: is there a threshold \( U_{\text{bound}} \) such that
- When \( U < U_{\text{bound}} \) deadlines are met
- When \( U > U_{\text{bound}} \) deadlines are missed

Schedulable!
Another Example
(Rate-Monotonic Scheduling)

Task 1
\[ P_1 = 2 \]
\[ C_1 = 1 \]

Task 2
\[ P_2 = 6 \]
\[ C_2 = 2.4 \]

\[ \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{1}{2} + \frac{2.4}{6} = 90\% \]

Schedulability depends on task set!
No clean utilization threshold between schedulable and unschedulable task sets!

Question: is there a threshold \( U \) such that:
- When \( U < U_{\text{bound}} \) deadlines are met
- When \( U > U_{\text{bound}} \) deadlines are missed
A Conceptual View of Schedulability

Utilization = \sum_{i} \frac{C_i}{P_i}

Question: is there a threshold \( U_{\text{bound}} \) such that
- When \( U < U_{\text{bound}} \), deadlines are met
- When \( U > U_{\text{bound}} \), deadlines are missed
A Conceptual View of Schedulability

Utilization = $\sum_i \frac{C_i}{P_i}$

- Modified Question: is there a threshold $U_{bound}$ such that
  - When $U < U_{bound}$ deadlines are met
  - When $U > U_{bound}$ deadlines may or may not be missed

All green area (schedulable)
A Conceptual View of Schedulability

Modified Question: is there a threshold $U_{\text{bound}}$ such that
- When $U < U_{\text{bound}}$ deadlines are met
- When $U > U_{\text{bound}}$ deadlines may or may not be missed

$$\text{Utilization} = \sum_i \frac{C_i}{P_i}$$

$U < U_{\text{bound}}$ is a sufficient but not necessary schedulability condition

All green area (schedulable)
A Conceptual View of Schedulability

Utilization = \[ \sum_{i} \frac{C_i}{P_i} \]

Equivalent question:
What’s the lowest utilization of an unschedulable task set?

All green area (schedulable)

- Modified Question: is there a threshold \( U_{\text{bound}} \) such that
  - When \( U < U_{\text{bound}} \) deadlines are met
  - When \( U > U_{\text{bound}} \) deadlines may or may not be missed
A Conceptual View of Schedulability

Utilization = $\sum_{i} \frac{C_i}{P_i}$

Equivalent question: What’s the lowest utilization of an unschedulable task set?

All green area (schedulable)

Modified Question: is there a threshold $U_{bound}$ such that

- When $U < U_{bound}$ deadlines are met
- When $U > U_{bound}$ deadlines may or may not be missed
The Schedulability Condition

For $n$ independent periodic tasks with periods equal to deadlines, the utilization bound is:

$$U = n \left( 2^{1/n} - 1 \right)$$

$$n \rightarrow \infty \quad U \rightarrow \ln 2$$
Periodic Task Scheduling

Rate Monotonic
- Bound
- Optimality

EDF
- Bound
- Optimality
Deriving the Utilization Bound for Rate Monotonic Scheduling

- Consider a simple case: 2 tasks

Find some task set parameter $x$ such that

Case (a): $x < x_o \rightarrow U(x)$ decreases with $x$
Case (b): $x > x_o \rightarrow U(x)$ increases with $x$

Thus $U(x)$ is minimum when $x = x_o$

Find $U(x_o)$
Consider a simple case: 2 tasks

Find some task set parameter $x$ such that
- Case (a): $x < x_o \rightarrow U(x)$ decreases with $x$
- Case (b): $x > x_o \rightarrow U(x)$ increases with $x$

Thus $U(x)$ is minimum when $x = x_o$

Find $U(x_o)$
Deriving the Utilization Bound for Rate Monotonic Scheduling

- Consider a simple case: 2 tasks

Find some task set parameter $x$ such that

Case (a): $x < x_o \rightarrow U(x)$ decreases with $x$
Case (b): $x > x_o \rightarrow U(x)$ increases with $x$

Thus $U(x)$ is minimum when $x = x_o$
Find $U(x_o)$
Deriving the Utilization Bound for Rate Monotonic Scheduling

Consider a simple case: 2 tasks

Find some task set parameter $x$ such that

Case (a): $x < x_o \rightarrow U(x)$ decreases with $x$
Case (b): $x > x_o \rightarrow U(x)$ increases with $x$

Thus $U(x)$ is minimum when $x = x_o$

Find $U(x_o)$
Deriving the Utilization Bound for Rate Monotonic Scheduling

- Consider a simple case: 2 tasks

Find some task set parameter $x$ such that
- Case (a): $x < x_o \rightarrow U(x)$ decreases with $x$
- Case (b): $x > x_o \rightarrow U(x)$ increases with $x$

Thus $U(x)$ is minimum when $x = x_o$

Find $U(x_o)$
Deriving the Utilization Bound for Rate Monotonic Scheduling

Consider a simple case: 2 tasks

Find some task set parameter $x$ such that

Case (a): $x < x_o \rightarrow U(x)$ decreases with $x$
Case (b): $x > x_o \rightarrow U(x)$ increases with $x$

Thus $U(x)$ is minimum when $x = x_o$

Find $U(x_o)$
Deriving the Utilization Bound for Rate Monotonic Scheduling

Consider these two sub-cases:

**Case (a):**

\[ P_2 - \left| \frac{P_2}{P_1} \right| P_1 \]

Task 1

\[ P_1 \]

\[ P_2 \]

Task 2

\[ C_1 \]

**Case (b):**

\[ P_2 - \left| \frac{P_2}{P_1} \right| P_1 \]

\[ P_1 \]

\[ P_2 \]

\[ C_1 \]

Find some task set parameter \( x \) such that

Case (a): \( x < x_o \) \( \rightarrow U(x) \) decreases with \( x \)

Case (b): \( x > x_o \) \( \rightarrow U(x) \) increases with \( x \)

Thus \( U(x) \) is minimum when \( x = x_o \)

Find \( U(x_o) \)
Deriving the Utilization Bound for Rate Monotonic Scheduling

Consider these two sub-cases:

**Case (a):**
\[ C_1 \leq P_2 - \left[ \frac{P_2}{P_1} \right] P_1 \]

**Case (b):**
\[ C_1 > P_2 - \left[ \frac{P_2}{P_1} \right] P_1 \]

Find some task set parameter \( x \) such that
- Case (a): \( x < x_o \) \( \Rightarrow \) \( U(x) \) decreases with \( x \)
- Case (b): \( x > x_o \) \( \Rightarrow \) \( U(x) \) increases with \( x \)
Thus \( U(x) \) is minimum when \( x = x_o \)
Find \( U(x_o) \)
Deriving the Utilization Bound for Rate Monotonic Scheduling

- Consider these two sub-cases:

**Case (a):**

\[ C_1 \leq P_2 - \left( \frac{P_2}{P_1} \right) P_1 \]

\[ C_2 = P_2 - C_1 \left( \frac{P_2}{P_1} \right) + 1 \]

\[ U = \frac{C_1}{P_1} + \frac{C_2}{P_2} = 1 + \frac{C_1}{P_2} \left[ \frac{P_2}{P_1} - \left( \frac{P_2}{P_1} \right) - 1 \right] \]

**Case (b):**

\[ C_1 > P_2 - \left( \frac{P_2}{P_1} \right) P_1 \]

\[ C_2 = (P_1 - C_1) \left( \frac{P_2}{P_1} \right) \]

\[ U = \frac{P_1}{P_2} \left[ \frac{P_2}{P_1} \right] + \frac{C_1}{P_2} \left[ \frac{P_2}{P_1} - \left( \frac{P_2}{P_1} \right) \right] \]
Deriving the Utilization Bound for Rate Monotonic Scheduling

- Consider these two sub-cases:

**Case (a):**

\[
C_1 \leq P_2 - \frac{P_2}{P_1} \cdot P_1
\]

\[
C_2 = P_2 - C_1 \left( \frac{P_2}{P_1} + 1 \right)
\]

\[
U = \frac{C_1}{P_1} + \frac{C_2}{P_2} = 1 + \frac{C_1}{P_2} \left[ \frac{P_2}{P_1} - \frac{P_2}{P_1} - 1 \right]
\]

**Case (b):**

\[
C_1 > P_2 - \frac{P_2}{P_1} \cdot P_1
\]

\[
C_2 = (P_1 - C_1) \frac{P_2}{P_1}
\]

\[
U = \frac{P_1}{P_2} \frac{P_2}{P_1} + \frac{C_1}{P_2} \left[ \frac{P_2}{P_1} - \frac{P_2}{P_1} \right] - 1
\]
Deriving the Utilization Bound for Rate Monotonic Scheduling

- The minimum utilization case:

\[ C_1 = P_2 - \left| \frac{P_2}{P_1} \right| P_1 \]

\[ U = 1 + \frac{C_1}{P_2} \left[ \frac{P_2}{P_1} - \left| \frac{P_2}{P_1} \right| - 1 \right] \]
Deriving the Utilization Bound for Rate Monotonic Scheduling

The minimum utilization case:

\[
C_1 = P_1 \left( \frac{P_2}{P_1} - \left\lfloor \frac{P_2}{P_1} \right\rfloor \right)
\]

\[
C_1 = P_2 - \left\lfloor \frac{P_2}{P_1} \right\rfloor P_1
\]

\[
U = 1 + \frac{P_1}{P_2} \left( \frac{P_2}{P_1} - \left\lfloor \frac{P_2}{P_1} \right\rfloor \right) \left[ \frac{P_2}{P_1} - \left\lfloor \frac{P_2}{P_1} \right\rfloor - 1 \right]
\]

\[
\Rightarrow \left\lfloor \frac{P_2}{P_1} \right\rfloor = 1
\]

\[
\Rightarrow U = 1 + \frac{P_1}{P_2} \left( \frac{P_2}{P_1} - 1 \right) \left( \frac{P_2}{P_1} - 2 \right)
\]

\[
\frac{dU}{d\left( \frac{P_2}{P_1} \right)} = 0
\]

\[
\Rightarrow \frac{P_2}{P_1} = \sqrt{2}
\]

\[
\Rightarrow U \approx 0.83
\]

Note that \( C_1 = P_2 - P_1 \)
Generalizing to N Tasks

\[ C_1 = P_2 - P_1 \]
\[ C_2 = P_3 - P_2 \]
\[ C_3 = P_4 - P_3 \]

\[ \ldots \]

\[ U = \frac{C_1}{P_1} + \frac{C_2}{P_2} + \frac{C_3}{P_3} + \ldots \]
Generalizing to N Tasks

\[
\begin{align*}
C_1 &= P_2 - P_1 \\
C_2 &= P_3 - P_2 \\
C_3 &= P_3 - P_2 \\
\vdots \\
U &= \frac{C_1}{P_1} + \frac{C_2}{P_2} + \frac{C_3}{P_3} + \ldots \\
\frac{dU}{d\left(\frac{P_2}{P_1}\right)} &= 0 \\
\frac{dU}{d\left(\frac{P_3}{P_2}\right)} &= 0 \\
\frac{dU}{d\left(\frac{P_4}{P_3}\right)} &= 0 \\
\vdots
\end{align*}
\]
Generalizing to N Tasks

\[ C_1 = P_2 - P_1 \]
\[ C_2 = P_3 - P_2 \]
\[ C_3 = P_3 - P_2 \]
\[ \cdots \]
\[ dU \left( \frac{P_2}{P_1} \right) = 0 \]
\[ dU \left( \frac{P_3}{P_2} \right) = 0 \]
\[ dU \left( \frac{P_4}{P_3} \right) = 0 \]
\[ \Rightarrow \frac{P_{i+1}}{P_i} = 2^{\frac{1}{n}} \]
\[ \Rightarrow U = n \left( 2^{\frac{1}{n}} - 1 \right) \]
Done Today

Periodic Task Scheduling

Rate Monotonic

Bound
Optimality

EDF

Bound
Optimality