Non-Preemptive Scheduling with History-Dependent Execution Time

Björn Andersson, Sagar Chaki, Dionisio de Niz
Software Engineering Institute

Brian Dougherty, Jules White
Virginia Tech

Russell Kegley
Lockheed Martin Aeronautics
Example of Task Set in the Old Model

n=3

\[ T_1 = 50, D_1 = 10 \]
\[ T_2 = 150, D_2 = 15 \]
\[ T_3 = 500, D_3 = 500 \]

\[ C_1 = 4 \]
\[ C_2 = 7 \]
\[ C_3 = 5 \]
Example of Task Set in the New Model
Example of Task Set in the New Model

n=3
Example of Task Set in the New Model

\( n=3 \)

\[ \begin{align*}
T_1 &= 50, \quad D_1 = 10 \\
T_2 &= 150, \quad D_2 = 15 \\
T_3 &= 500, \quad D_3 = 500
\end{align*} \]
Example of Task Set in the New Model

n=3

\[ T_1 = 50, D_1 = 10 \quad T_2 = 150, D_2 = 15 \quad T_3 = 500, D_3 = 500 \]

Assume non-preemptive scheduling
Example of Task Set in the New Model

\[ n=3 \]

\[ T_1=50, D_1=10 \quad T_2=150, D_2=15 \quad T_3=500, D_3=500 \]

\[ \text{ubc}_2=6, \text{lhubc}_2=1, \text{ihubc}_2=1 \]

Job of task 1  |  Job of task 2

6 time units

Execution time of a job when a pre-specified history is matched at run-time
Example of Task Set in the New Model

$n=3$

- $T_1=50, D_1=10$
- $T_2=150, D_2=15$
- $T_3=500, D_3=500$

- $\text{nhlbc}_1=1$
- $\text{nhlbc}_2=1$
- $\text{nhlbc}_3=1$

- $\text{lbc}_1=0, \text{lhlbc}_1=0$
- $\text{lbc}_2=0, \text{lhlbc}_2=0$
- $\text{lbc}_3=0, \text{lhlbc}_3=0$

- $\text{nhubc}_1=2$
- $\text{nhubc}_2=2$
- $\text{nhubc}_3=1$

- $\text{ubc}_1=4, \text{lhubc}_1=0$
- $\text{ubc}_2=7, \text{lhubc}_2=0$
- $\text{ubc}_3=5, \text{lhubc}_3=0$

- $\text{ubc}_1^1=3, \text{lhubc}_1^1=1, \text{ihubc}_1^1=2$
- $\text{ubc}_2^1=6, \text{lhubc}_2^1=1, \text{ihubc}_2^1=1$
- $\text{ubc}_3^1=5, \text{lhubc}_3^1=0$

Job of task 2

7 time units

Execution time of a job when the zero-length pre-specified history is matched at run-time
Example of Task Set in the New Model

\[\begin{align*}
n &= 3 \\
T_1 &= 50, D_1 = 10 & T_2 &= 150, D_2 = 15 & T_3 &= 500, D_3 = 500 \\
n_{hubc_2} &= 2 \\
uc_2^1 &= 7, lhubc_2^1 = 0 \\
uc_2^2 &= 6, lhubc_2^2 = 1, ihubc_2^{2,1} = 1
\end{align*}\]

Pre-specified histories of task \( \tau_2 \) and their associated upper bounds on execution times
Example of Task Set in the New Model

\( n = 3 \)

\[
\begin{align*}
T_1 &= 50, \ D_1 = 10 \\
T_2 &= 150, \ D_2 = 15 \\
T_3 &= 500, \ D_3 = 500 \\
nhbc_1 &= 2 \\
nhbc_2 &= 2 \\
nhbc_3 &= 1 \\
ubc_1^1 &= 4, \ lhubc_1^1 = 0 \\
ubc_1^2 &= 3, \ lhubc_1^2 = 1, \ ihubc_1^{2,1} = 2 \\
ubc_2^1 &= 7, \ lhubc_2^1 = 0 \\
ubc_2^2 &= 6, \ lhubc_2^2 = 1, \ ihubc_2^{2,1} = 1 \\
ubc_3^1 &= 5, \ lhubc_3^1 = 0
\end{align*}
\]

Upper bounds on execution times of a job as a function of jobs that executed before it
Example of Task Set in the New Model

\( n=3 \)

\[
\begin{align*}
T_1 &= 50, \ D_1 = 10 \\
nhlbc_1 &= 1 \\
lbc_1^1 &= 0, \ lhlc_1^1 = 0
\end{align*}
\]

\[
\begin{align*}
T_2 &= 150, \ D_2 = 15 \\
nhlbc_2 &= 1 \\
lbc_2^1 &= 0, \ lhlc_2^1 = 0
\end{align*}
\]

\[
\begin{align*}
T_3 &= 500, \ D_3 = 500 \\
nhlbc_3 &= 1 \\
lbc_3^1 &= 0, \ lhlc_3^1 = 0
\end{align*}
\]

Lower bound on execution times of a job
Example of Task Set in the New Model

n=3

\[
\begin{align*}
T_1 &= 50, D_1 = 10 \\
nhlbc_1 &= 1 \\
lbc_1 &= 0, lhlbc_1 = 0 \\
nhubc_1 &= 2 \\
ubc_1 &= 4, lhubc_1 = 0 \\
ubc_1^2 &= 3, lhubc_1^2 = 1, ihubc_1^{2,1} = 2 \\
\end{align*}
\]

\[
\begin{align*}
T_2 &= 150, D_2 = 15 \\
nhlbc_2 &= 1 \\
lbc_2 &= 0, lhlbc_2 = 0 \\
nhubc_2 &= 2 \\
ubc_2 &= 7, lhubc_2 = 0 \\
ubc_2^2 &= 6, lhubc_2^2 = 1, ihubc_2^{2,1} = 1 \\
\end{align*}
\]

\[
\begin{align*}
T_3 &= 500, D_3 = 500 \\
nhlbc_3 &= 1 \\
lbc_3 &= 0, lhlbc_3 = 0 \\
nhubc_3 &= 1 \\
ubc_3 &= 5, lhubc_3 = 0 \\
\end{align*}
\]

Upper and lower bounds on execution times of a job as a function of jobs that executed before it
Example of Task Set in the New Model

\( n=3 \)

\[
\begin{align*}
T_1 &= 50, D_1 = 10 \\
\text{nhlbc}_1 &= 1 \\
\text{lbc}_1^1 &= 0, \text{lhlbc}_1^1 &= 0 \\
\text{nhubc}_1 &= 2 \\
\text{ubc}_1^1 &= 4, \text{lhubc}_1^1 &= 0 \\
\text{ubc}_1^2 &= 3, \text{lhubc}_1^2 &= 1, \text{ihubc}_1^2 &= 2 \\
T_2 &= 150, D_2 = 15 \\
\text{nhlbc}_2 &= 1 \\
\text{lbc}_2^1 &= 0, \text{lhlbc}_2^1 &= 0 \\
\text{nhubc}_2 &= 2 \\
\text{ubc}_2^1 &= 7, \text{lhubc}_2^1 &= 0 \\
\text{ubc}_2^2 &= 6, \text{lhubc}_2^2 &= 1, \text{ihubc}_2^2 &= 1 \\
T_3 &= 500, D_3 = 500 \\
\text{nhlbc}_3 &= 1 \\
\text{lbc}_3^1 &= 0, \text{lhlbc}_3^1 &= 0 \\
\text{nhubc}_3 &= 1 \\
\text{ubc}_3^1 &= 5, \text{lhubc}_3^1 &= 0
\end{align*}
\]

Schedule these tasks with non-preemptive fixed priority scheduling

\[ \tau_1 \uparrow \quad \quad \tau_1 \quad \downarrow \]

\[ \tau_2 \uparrow \quad \quad \quad \quad \quad \tau_2 \quad \downarrow \]

\[ \tau_3 \uparrow \quad \quad \tau_3 \quad \downarrow \]

This is the worst-case arrival for \( \tau_2 \). With new model: deadline of \( \tau_2 \) is met.
Example of Task Set in the Old Model

\[ n=3 \]

\[ T_1=50, \; D_1=10 \]
\[ T_2=150, \; D_2=15 \]
\[ T_3=500, \; D_3=500 \]

\[ C_1=4 \]
\[ C_2=7 \]
\[ C_3=5 \]

Schedule these tasks with non-preemptive fixed priority scheduling

\[ \tau_1 \]
\[ \tau_2 \]
\[ \tau_3 \]

This is the worst-case arrival for \( \tau_2 \). With old model: deadline of \( \tau_2 \) is missed.
Example of Task Set in the New Model

n=3

<table>
<thead>
<tr>
<th>Task</th>
<th>Deadline</th>
<th>Lateness</th>
<th>Remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_1</td>
<td>50</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>T_2</td>
<td>150</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>T_3</td>
<td>500</td>
<td>500</td>
<td>1</td>
</tr>
</tbody>
</table>

nhlc_1 = 1

nhlc_2 = 1

nhlc_3 = 1

nhubc_1 = 2

nhubc_2 = 2

nhubc_3 = 1

ubc_1^1 = 4, lhubc_1^1 = 0

ubc_2^1 = 7, lhubc_2^1 = 0

ubc_3^1 = 5, lhubc_3^1 = 0

ubc_1^2 = 3, lhubc_1^2 = 1, ihubc_1^2,1 = 2

ubc_2^2 = 6, lhubc_2^2 = 1, ihubc_2^2,1 = 1

ubc_3^2 = 1, lhubc_3^2 = 0

ubc_1^3 = 1, lhubc_1^3 = 0

ubc_2^3 = 1, lhubc_2^3 = 0

ubc_3^3 = 1, lhubc_3^3 = 0

Schedule these tasks with non-preemptive fixed priority scheduling

This is the worst-case arrival for τ_2. With new model: deadline of τ_2 is met.
Example of Task Set in the New Model

n=3

\[ T_1 = 50, \ D_1 = 10 \]

\[ T_2 = 150, \ D_2 = 15 \]

\[ T_3 = 500, \ D_3 = 500 \]

\[ nhlbc_1 = 1 \]

\[ nhlbc_2 = 1 \]

\[ nhlbc_3 = 1 \]

\[ lbc_1^1 = 0, \ lhlbc_1^1 = 0 \]

\[ lbc_2^1 = 0, \ lhlbc_2^1 = 0 \]

\[ lbc_3^1 = 0, \ lhlbc_3^1 = 0 \]

\[ nhubc_1 = 2 \]

\[ nhubc_2 = 2 \]

\[ nhubc_3 = 1 \]

\[ ubc_1^1 = 4, \ lhubc_1^1 = 0 \]

\[ ubc_2^1 = 7, \ lhubc_2^1 = 0 \]

\[ ubc_3^1 = 5, \ lhubc_3^1 = 0 \]

\[ ubc_1^2 = 3, \ lhubc_1^2 = 1, \ ihubc_1^{2,1} = 2 \]

\[ ubc_2^2 = 6, \ lhubc_2^2 = 1, \ ihubc_2^{2,1} = 1 \]

Schedule these tasks with non-preemptive fixed priority scheduling

For another arrival for \( \tau_2 \). Job of \( \tau_2 \) has longer execution time but deadline of \( \tau_2 \) is met.
Problem discussed in this presentation

How to perform schedulability analysis of tasks described with the new model under fixed-priority non-preemptive scheduling?
Ideas that do not work

- L&L critical instant
Ideas that do not work

- L&L critical instant
- Level-\(i\) busy period
busy period

\[ \tau_{4,1} \quad \tau_{1,1} \quad \tau_{2,1} \quad \tau_{3,1} \quad \tau_{1,2} \quad \tau_{3,2} \]
1. \( L := \) Compute the maximum duration of a busy period
2. \( Q_i := \left\lfloor \frac{L}{T_i} \right\rfloor \)
3. \textbf{for} \( q := 1 \) to \( Q_i \) \textbf{do}
4. \( R_{i,q} := \) compute the maximum response time of \( \tau_{i,q} \) — the \( q \):th job of task \( \tau_i \) — in every busy period of length at most \( L \)
5. \( R_i := \max_{q=1..Q} R_{i,q} \)
1. \( L := \) Compute the maximum duration of a busy period
2. \( Q_i := \left\lceil \frac{L}{T_i} \right\rceil \)
3. \textbf{for} \( q := 1 \) to \( Q_i \) \textbf{do}
4. \( R_{i,q} := \) compute the maximum response time of \( \tau_{i,q} \) — the \( q \):th job of task \( \tau_i \) — in every busy period of length at most \( L \)
5. \( R_i := \max_{q=1..Q} R_{i,q} \)
1. $L := \text{Compute the maximum duration of a busy period}$
2. $Q_i := \left\lceil \frac{L}{T_i} \right\rceil$
3. \textbf{for} $q := 1$ to $Q_i$ \textbf{do}
4. \hspace{1em} $<R_{i,q}, valid_{i,q}> := \text{compute the maximum response time of } \tau_{i,q}$
   \hspace{1em} — the $q$:th job of task $\tau_i$ — in every busy period of length
   \hspace{1em} at most $L$ and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$
An idea that works for computing the response time of task \( \tau_i \):

1. \( L := \) Compute the maximum duration of a busy period.
2. \( Q_i := \left\lceil \frac{L}{T_i} \right\rceil \)
3. \textbf{for} \( q := 1 \) to \( Q_i \) \textbf{do}
4. \( <R_{i,q}, valid_{i,q}> := \) compute the maximum response time of — the \( q:th \) job of task \( \tau_i \) — in every busy period of length at most \( L \) and compute whether \( \tau_{i,q} \) exist in the busy period.
5. \( R_i = \max_{q=1..Q} \text{and valid}_{i,q} R_{i,q} \)
An idea that works for computing the response time of task $\tau_i$

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \left\lceil \frac{L}{T_i} \right\rceil$
3. for $q := 1$ to $Q_i$ do
4.   $<R_{i,q}, valid_{i,q}> :=$ compute the maximum response time of $\tau_{i,q}$ — the $q$:th job of task $\tau_i$ — in every busy period of length at most $L$ and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q and valid_{i,q}} R_{i,q}$
An idea that works for computing the response time of task $\tau_i$

1. $L := \text{Compute the maximum duration of a busy period}$
2. $Q_i := \left\lceil \frac{L}{T_i} \right\rceil$
3. for $q := 1$ to $Q_i$ do
4. $<R_{i,q}, valid_{i,q}> := \text{compute the maximum response time of } \tau_{i,q}$
   — the $q$:th job of task $\tau_i$ — in every busy period of length at most $L$ and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$
An idea that works for computing the response time of task $\tau_i$

1. $L := $ Compute the maximum duration of a busy period
2. $Q_i := \left\lceil \frac{L}{T_i} \right\rceil$
3. for $q := 1$ to $Q_i$ do
4. \[<R_{i,q}, valid_{i,q}> := \text{compute the maximum response time of } \tau_{i,q} \]
   the $q$:th job of task $\tau_i$ — in every busy period of length
   at most $L$ and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$
An idea that works for computing the response time of task $\tau_i$

1. $L := \text{Compute the maximum duration of a busy period}$
2. $Q_i := \left\lceil L/T_i \right\rceil$
3. for q := 1 to $Q_i$ do
4. \begin{align*}
   & <R_{i,q}, valid_{i,q}> := \text{compute the maximum response time of } \tau_{i,q} \\
   & \text{— the } q:\text{th job of task } \tau_i \text{— in every busy period of length at most } L \text{ and compute whether } \tau_{i,q} \text{ exist in the busy period.}
\end{align*}
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$
An idea that works for computing the response time of task $\tau_i$

1. $L := \text{Compute the maximum duration of a busy period}$

2. $Q_i := \left\lceil \frac{L}{T_i} \right\rceil$

3. for $q := 1$ to $Q_i$ do

4. $<R_{i,q}, valid_{i,q}> := \text{compute the maximum response time of } \tau_{i,q}$

   — the $q$:th job of task $\tau_i$ — in every busy period of length

   at most $L$ and compute whether $\tau_{i,q}$ exist in the busy period.

5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$
1. \( L := \) Compute the maximum duration of a busy period
2. \( Q_i := \left\lfloor \frac{L}{T_i} \right\rfloor \)
3. \textbf{for} \( q := 1 \) \textbf{to} \( Q_i \) \textbf{do}
4. \( <R_{i,q}, \text{valid}_{i,q}> := \) compute the maximum response time of \( \tau_{i,q} \) — the \( q \)-th job of task \( \tau_i \) — in every busy period of length at most \( L \) and compute whether \( \tau_{i,q} \) exist in the busy period.
5. \( R_i = \max_{q=1..Q \text{ and } \text{valid}_{i,q}} R_{i,q} \)
Perform step 1: Compute the maximum duration of a busy period

Represent a schedule.

- $x_j^p = 1$ iff a job of task $\tau_j$ executes in position $p$ in busy period
- $y_{j,k}^p = 1$ iff job $\tau_{j,k}$ executes in position $p$ in busy period
- $t_k$ = time of $k$:th context switch in busy period

Other variables

- $A_{i,k}$ = arrival time of $\tau_{i,k}$
- $f_{j,k}$ = finishing time of $\tau_{j,k}$
- $ft_{\text{last job}}$ = time when the busy period ends

Maximize $ft_{\text{last job}} - t_1$ subject to constraints .... [see paper]

1. $L := \text{Compute the maximum duration of a busy period}$
2. $Q_i := \left\lceil \frac{L}{T_i} \right\rceil$
3. for $q := 1$ to $Q_i$ do
4. \begin{itemize}
   \item $<R_{i,q}, valid_{i,q}> := \text{compute the maximum response time of } \tau_{i,q}$
   \item the $q$:th job of task $\tau_i$ — in every busy period of length at most $L$ and compute whether $\tau_{i,q}$ exist in the busy period.
\end{itemize}
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$
Perform step 4: Compute the maximum response time of $\tau_{i,q}$ during every busy period of duration at most $L$

Represent a schedule.

- $x_j^p = 1$ iff a job of task $\tau_j$ executes in position $p$ in busy period
- $y_{j,k}^p = 1$ iff job $\tau_{j,k}$ executes in position $p$ in busy period
- $t_k$ = time of $k$:th context switch in busy period

Other variables
- $A_{j,k}$ = arrival time of $\tau_{j,k}$
- $f_{j,k}$ = finishing time of $\tau_{j,k}$
- $ft_{lastjob}$ = time when the busy period ends

Maximize $f_{j,k} - A_{j,k}$ subject to constraints .... [see paper]

1. $L :=$ Compute the maximum duration of a busy period
2. $Q_i := \lceil L/T_i \rceil$
3. $\textbf{for } q := 1 \textbf{ to } Q_i \textbf{ do}$
4. $<R_{i,q}, valid_{i,q}> :=$ compute the maximum response time of $\tau_{i,q}$ — the $q$:th job of task $\tau_i$ — in every busy period of length at most $L$ and compute whether $\tau_{i,q}$ exist in the busy period.
5. $R_i = \max_{q=1..Q \text{ and } valid_{i,q}} R_{i,q}$
Conclusion

It is possible to compute exact response times of tasks scheduled by non-preemptive fixed-priority scheduling where execution times depend on history.
Thanks for listening!