Improved feasibility of fixed-priority scheduling with deferred preemption using preemption thresholds for preemption points

Reinder J. Bril, Martijn M.H.P. van den Heuvel, and Johan J. Lukkien

Dep. of Mathematics and Computer Science
Group System Architecture and Networking
Background and motivation

• Facts about fixed-priority scheduling (FPS):
  • described in standards, e.g. OSEK (Automotive);
  • supported by most COTS RTOS;
  • de facto standard in industry.

• Advantages of limited preemptive FPS:
  • Reduced memory requirements;
  • Reduced cost of arbitrary preemptions;
  • Improved schedulability of task sets.
Background and motivation

• Non-preemptive jobs (FPNS):
  • Pro: minimal memory requirements;
  • Pro: no preemption costs;
  • Cons: reduced schedulability.

• Two main alternatives:
  1. Sequence of non-preemptive sub-jobs (FPDS):
     – Preemption points
  2. Preemption thresholds for tasks (FPTS).

• Schedulability: neither dominates the other.
Background and motivation

• FPDS:
  • highest schedulability ratio;
  • generates less preemptions;
  • most predictable for estimating preemption costs
    – because the number of preemptions and their positions are fixed and known from the code.

• Question: can we improve on FPDS?

Background and motivation

• Generalization of FPTS and FPDS (FPGS):
  • Preemption thresholds for tasks and per sub-job
    – Pro: improved schedulability ratio;
    – Cons: arbitrary preemptions.

• Specialization of FPGS (FPDS\textsuperscript{\lambda}):
  • Preemption thresholds for tasks only
    – Pro: limits preemptions to preemption points;
    – Cons: limited schedulability ratio improvement.

Background and motivation

8 tasks, 5,000 task sets, $T_i \in [100, 10,000]$ (uniform), $U_i$ by UUnifast ($\Rightarrow C_i$), $D_i \in [0.5(T_i + C_i), T_i]$ (uniform);
Background and motivation

• Goal:
  • Limited preemptive FPS scheme
    - schedulability ratio comparable to FPGS;
    - preemptions limited to preemption points.

• Approach:
  • Preemption thresholds per preemption point.
Overview

• Background and motivation

• Real-time scheduling model (FPS)
  • Basic model
  • Enhanced model
  • Generalization graph

• An example
  • A schedulable configuration
  • Determining the configuration

• Contributions

• Conclusion
Basic FPS – model

- **Events**: implicit
- **Tasks** ($\tau$):
  - independent, no self-suspension
- **characteristics** ($R^+$):
  - minimal inter-arrival time ($T$);
  - computation time ($C$);
  - deadline ($D$);
- **Scheduling algorithm**:
  - fixed-priority ($\pi$) & non-idling;
  - [non-] preemptive
- **Platform**: single CPU

![Graph showing task scheduling](image)
Enhanced FPS – model

• Existing enhancements:
  • Job of $\tau_i$ is a sequence of $m_i$ sub-jobs.
  • Preemption threshold
    - per task: $\theta_i$, where $\pi_i \leq \theta_i \leq \pi_1$;
    - per sub-job: $\theta_{i,k}$, where $\pi_i \leq \theta_{i,k} \leq \pi_1$.

• Novel enhancement:
  - Preemption threshold
    - per preemption point: $\theta_{i,k}$, where $\pi_i \leq \theta_{i,k} \leq \pi_1$
    - superseding $\theta_i$. 
## Generalization graph for FPS algorithms

Job as a sequence of $m_i$ sub-jobs.

<table>
<thead>
<tr>
<th>$m_i = 1$</th>
<th>$m_i \geq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{i,k} = \pi_i$</td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \pi_i$</td>
</tr>
<tr>
<td>FPPS</td>
<td>FPPS$^+$</td>
</tr>
<tr>
<td>$\theta_{i,k} = \pi_1$</td>
<td>FPNS</td>
</tr>
</tbody>
</table>

Generalization graph for FPS algorithms

Preemption threshold per task.

<table>
<thead>
<tr>
<th></th>
<th>$m_i = 1$</th>
<th>$m_i \geq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{i,k} = \pi_i$</td>
<td>FPPS↑ ← FPPS+</td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \pi_i$</td>
</tr>
<tr>
<td></td>
<td>FPTS↓</td>
<td></td>
</tr>
<tr>
<td>$\theta_{i,k} = \pi_1$</td>
<td>FPNS ← FPDS</td>
<td></td>
</tr>
</tbody>
</table>

### Generalization graph for FPS algorithms

- **Preemption threshold per sub-job (only).**

<table>
<thead>
<tr>
<th>$m_i = 1$</th>
<th>$m_i \geq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{i,k} = \pi_i$</td>
<td>$\rightarrow$</td>
</tr>
<tr>
<td>$\theta_{i,a} = \pi_i$</td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \pi_i$</td>
</tr>
</tbody>
</table>

- **FPPS**
- **FPTS**
- **FPNS**
- **FPDS**

---


RTNS-2013 (Sophia Antipolis, France)
Preemption threshold per task and per sub-job. 

<table>
<thead>
<tr>
<th>$m_i = 1$</th>
<th>$m_i \geq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \pi_i$</td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \theta_i$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\theta_{i,k} = \pi_i$</th>
<th>FPPS</th>
<th>FPPS$^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPTS</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td>FPNS</td>
<td>←</td>
<td>←</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\theta_{i,k} = \pi_1$</th>
<th>FPNS</th>
<th>FPDS $\wedge$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPTS $^+$</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td>FPDS</td>
<td>←</td>
<td>←</td>
</tr>
</tbody>
</table>

**Generalization graph for FPS algorithms**

Preemption threshold per sub-job and per preemption point.

<table>
<thead>
<tr>
<th></th>
<th>$m_i = 1$</th>
<th>$m_i \geq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{i,k} = \pi_i$</td>
<td><strong>FPPS</strong> $\leftarrow$ <strong>FPPS</strong>$^+$</td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \pi_i$</td>
</tr>
<tr>
<td></td>
<td><strong>FPTS</strong> $\leftarrow$ <strong>FPTS</strong>$^+$ $\leftarrow$ <strong>FPGS</strong> $\leftarrow$ <strong>FPVS</strong></td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \theta_i$</td>
</tr>
<tr>
<td>$\theta_{i,k} = \pi_1$</td>
<td><strong>FPNS</strong> $\leftrightarrow$ **FPDS$^\wedge$ $\leftrightarrow$ <strong>FPDS</strong>$^*$</td>
<td></td>
</tr>
</tbody>
</table>

[0] This presentation/paper.
Generalization graph for FPS algorithms

<table>
<thead>
<tr>
<th>$m_i = 1$</th>
<th>$m_i \geq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \pi_i$</td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \theta_i$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\theta_{i,k} = \pi_i$</th>
<th>FPPS</th>
<th>FPPS$^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FPTS</td>
<td>FPTS$^+$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FPGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FPVS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\theta_{i,k} = \pi_1$</th>
<th>FPNS</th>
<th>FPDS</th>
<th>FPDS$^\wedge$</th>
<th>FPDS$^*$</th>
</tr>
</thead>
</table>

**FPVS:**  
*Pro:* highest schedulability ratio;  
*Cons:* potential of arbitrary preemptions;  
*Cons:* potential of higher memory requirements.
Generalization graph for FPS algorithms

<table>
<thead>
<tr>
<th></th>
<th>$m_i = 1$</th>
<th>$m_i \geq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \pi_i$</td>
<td>$m_i &gt; 1 \Rightarrow \theta_{i,a} = \theta_i$</td>
</tr>
<tr>
<td>$\theta_{i,k} = \pi_i$</td>
<td>FPPS $\leftarrow$ FPPS$^+$</td>
<td>FPPS$^+$ $\leftarrow$ FPPS</td>
</tr>
<tr>
<td></td>
<td>FPTS $\leftarrow$ FPTS$^+$</td>
<td>FPTS$^+$ $\leftarrow$ FPPS $\leftarrow$ FPGS $\leftarrow$ FPVS</td>
</tr>
<tr>
<td>$\theta_{i,k} = \pi_1$</td>
<td>FPNS $\leftarrow$ FPDS $\leftarrow$ FPDS$^\wedge$</td>
<td>FPDS$^\wedge$ $\leftarrow$ FPDS$^\bullet$</td>
</tr>
</tbody>
</table>

**FPDS$^\bullet$:**
- **Pro:** high schedulability ratio;
- **Pro:** predictable preemptions costs;
- **Pro:** predictable memory requirements.
Evaluation

8 tasks, 5,000 task sets,
\( T_i \in [100, 10,000] \) (uniform), \( U_i \) by UUnifast (\( \Rightarrow C_i \)),
\( D_i \in [0.5(T_i + C_i), T_i] \) (uniform);
An example task set $\mathcal{T}$

<table>
<thead>
<tr>
<th>Task</th>
<th>$T_i$</th>
<th>$D_i$</th>
<th>$C_i$</th>
<th>$\pi_i$</th>
<th>$C_{i,a}$</th>
<th>$\theta_{i,a}$</th>
<th>$\theta_{i,a}^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>23</td>
<td>23</td>
<td>8</td>
<td>2</td>
<td>3, 5</td>
<td>3, 3</td>
<td>2</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>29</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td>1, 4, 5</td>
<td>3, 3, 3</td>
<td>1, 2</td>
</tr>
</tbody>
</table>

- **Task set $\mathcal{T}$ is**
  - *not schedulable* with any existing limited preemptive FPS scheme, not even under any other priority assignment.
  - *schedulable* with both FPVS and FPDS$^*$.

- **Conclusion:**
  - FPVS strictly dominates all existing schemes.
An example task set $\mathcal{T}$ – configuration

<table>
<thead>
<tr>
<th>Task</th>
<th>$T_i$</th>
<th>$D_i$</th>
<th>$C_i$</th>
<th>$\pi_i$</th>
<th>$C_{i,a}$</th>
<th>$\theta_{i,a}$</th>
<th>$\theta^*_{i,a}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>23</td>
<td>23</td>
<td>8</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>29</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Determining the configuration of task $\tau_1$. 

RTNS-2013 (Sophia Antipolis, France)
An example task set $\mathcal{T}$ – configuration

<table>
<thead>
<tr>
<th>Task</th>
<th>$T_i$</th>
<th>$D_i$</th>
<th>$C_i$</th>
<th>$\pi_i$</th>
<th>$C_{i,a}$</th>
<th>$\theta_{i,a}$</th>
<th>$\theta^*_{i,a}$</th>
<th>$\beta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>23</td>
<td>23</td>
<td>8</td>
<td>2</td>
<td>3, 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>29</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Determining the configuration of task $\tau_2$.

• **Blocking tolerance** ($\beta_i$) [1]:
  • the maximum amount of time that a task ($\tau_i$) can be blocked without missing its deadline ($D_i$).

An example task set $\mathcal{T}$ – towards analysis

<table>
<thead>
<tr>
<th>Task</th>
<th>$T_i$</th>
<th>$D_i$</th>
<th>$C_i$</th>
<th>$\pi_i$</th>
<th>$C_{i,a}$</th>
<th>$\theta_{i,a}$</th>
<th>$\theta_{i,a}^*$</th>
<th>$\beta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>23</td>
<td>23</td>
<td>8</td>
<td>2</td>
<td>3, 5</td>
<td>3, 3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>29</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Determining the configuration of task $\tau_2$.

$C_{2,1} = 3$, $C_{2,2} = 5$, $C_2 = 8$, $\pi_1$, $\pi_2$, $\beta_1 = 5$.

FPTS$^+$ configuration
&
FPDS configuration
An example task set $\mathcal{T}$ – towards analysis

<table>
<thead>
<tr>
<th>Task</th>
<th>$T_i$</th>
<th>$D_i$</th>
<th>$C_i$</th>
<th>$\pi_i$</th>
<th>$C_{i,a}$</th>
<th>$\theta_{i,a}$</th>
<th>$\theta_{i,a}^*$</th>
<th>$\beta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>23</td>
<td>23</td>
<td>8</td>
<td>2</td>
<td>3, 5</td>
<td>3, 3</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>29</td>
<td>25</td>
<td>10</td>
<td>1</td>
<td>1, 4, 5</td>
<td>3, 3, 3</td>
<td>1, 2</td>
<td>1</td>
</tr>
</tbody>
</table>

- Determining the configuration of task $\tau_3$.

FPVS configuration & FPDS\textsuperscript{*} configuration

$C_3 = 10$

$C_{3,1} = 1$
$C_{3,2} = 4$
$C_{3,3} = 5$

$\beta_1 = 5$
$\beta_2 = 9$
Contributions

1. Novel scheduling algorithms: FPVS
   • Sub-jobs (similar to FPDS);
   • preemption thresholds
     – for sub-jobs and preemption points;
   • generalizes existing FPS algs.

2. Schedulability analysis for FPVS (similar to [11])
   • specializes to all existing FPS algs;

3. Algorithm to maximize schedulability under FPS:
   • given: $T_i$, $D_i$, $C_i$, and $\pi_i$;
   • determine: $C_{i,a}$, $\theta_{i,a}$, $\theta_{i,a}^\circ$ (inspired by [8]).


Conclusion (significance of the result)

- **FPVS and FPDS•**
  - have highest schedulability ratio;
  - can be emulated, e.g. with OSEK-standard (Automotive).

- **FPDS• meets our goal:**
  - Schedulability comparable to FPGS;
  - Preemptions limited to preemption points.

- **If schedulable(\(\mathcal{T},\text{FPDS}\)) then FPDS• may have**
  - reduced memory requirements;
  - tighter preemption cost estimation;
  - reduced power consumption.