On the Computational Complexity of Periodic Scheduling

PhD defense

Thomas Rothvoß

DISCRETE OPTIMIZATION

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \ldots, \tau_n$ with

$$\tau_i = (c(\tau_i),\, d(\tau_i),\, p(\tau_i))$$

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \ldots, \tau_n$ with

$$\tau_i = (c(\tau_i),\, d(\tau_i),\, p(\tau_i))$$

running time

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \ldots, \tau_n$ with

$$\tau_i = (c(\tau_i),\ d(\tau_i),\ p(\tau_i))$$

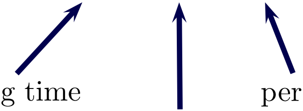
running time

(relative) deadline

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \ldots, \tau_n$ with

$$\tau_i = (c(\tau_i),\, d(\tau_i),\, p(\tau_i))$$



running time

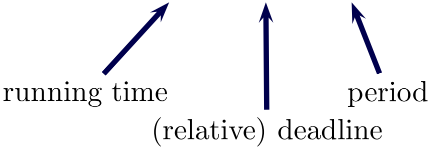
(relative) deadline

period

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \ldots, \tau_n$ with

$$\tau_i = (c(\tau_i),\, d(\tau_i),\, p(\tau_i))$$



running time

(relative) deadline

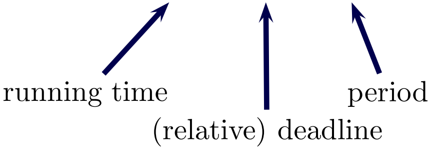
period

W.l.o.g.: Task τ_i releases job of length $c(\tau_i)$ at $z \cdot p(\tau_i)$ and absolute deadline $z \cdot p(\tau_i) + d(\tau_i)$ $(z \in \mathbb{N}_0)$

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \dots, \tau_n$ with

$$\tau_i = (c(\tau_i),\, d(\tau_i),\, p(\tau_i))$$



running time

(relative) deadline

period

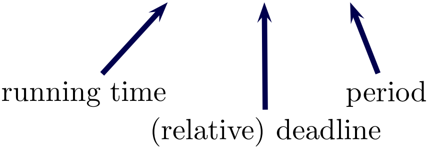
W.l.o.g.: Task τ_i releases job of length $c(\tau_i)$ at $z \cdot p(\tau_i)$ and absolute deadline $z \cdot p(\tau_i) + d(\tau_i)$ $(z \in \mathbb{N}_0)$

Utilization: $u(\tau_i) = \frac{c(\tau_i)}{p(\tau_i)}$

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \ldots, \tau_n$ with

$$\tau_i = (c(\tau_i),\, d(\tau_i),\, p(\tau_i))$$



running time

(relative) deadline

period

W.l.o.g.: Task τ_i releases job of length $c(\tau_i)$ at $z \cdot p(\tau_i)$ and absolute deadline $z \cdot p(\tau_i) + d(\tau_i)$ $(z \in \mathbb{N}_0)$

Utilization: $u(\tau_i) = \frac{c(\tau_i)}{p(\tau_i)}$

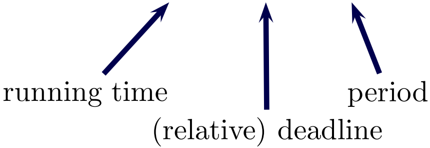
Settings:

- static priorities ↔ dynamic priorities

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \dots, \tau_n$ with

$$\tau_i = (c(\tau_i),\, d(\tau_i),\, p(\tau_i))$$



running time

(relative) deadline

period

W.l.o.g.: Task τ_i releases job of length $c(\tau_i)$ at $z \cdot p(\tau_i)$ and absolute deadline $z \cdot p(\tau_i) + d(\tau_i)$ ($z \in \mathbb{N}_0$)

Utilization: $u(\tau_i) = \frac{c(\tau_i)}{p(\tau_i)}$

Settings:

- static priorities ↔ dynamic priorities
- single-processor ↔ multi-processor

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \dots, \tau_n$ with

$$\tau_i = (c(\tau_i),\, d(\tau_i),\, p(\tau_i))$$

running time

(relative) deadline

period

W.l.o.g.: Task τ_i releases job of length $c(\tau_i)$ at $z \cdot p(\tau_i)$ and absolute deadline $z \cdot p(\tau_i) + d(\tau_i)$ $(z \in \mathbb{N}_0)$

Utilization: $u(\tau_i) = \frac{c(\tau_i)}{p(\tau_i)}$

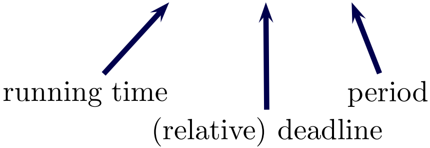
Settings:

- static priorities ↔ dynamic priorities
- single-processor ↔ multi-processor
- preemptive scheduling ↔ non-preemptive

Real-time Scheduling

Given: (synchronous) tasks $\tau_1, \ldots, \tau_n$ with

$$\tau_i = (c(\tau_i),\ d(\tau_i),\ p(\tau_i))$$



running time

(relative) deadline

period

W.l.o.g.: Task τ_i releases job of length $c(\tau_i)$ at $z \cdot p(\tau_i)$ and absolute deadline $z \cdot p(\tau_i) + d(\tau_i)$ $(z \in \mathbb{N}_0)$

Utilization: $u(\tau_i) = \frac{c(\tau_i)}{p(\tau_i)}$

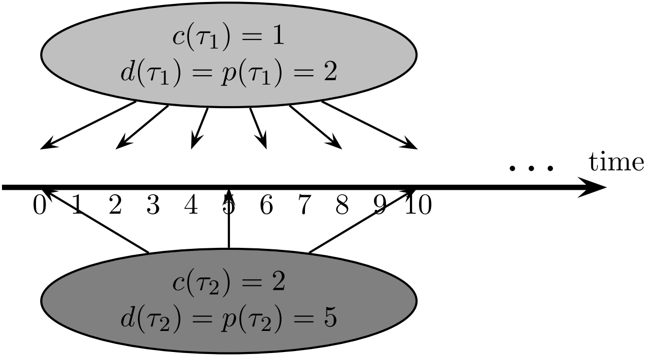
Settings:

- static priorities ↔ dynamic priorities
- single-processor ↔ multi-processor
- preemptive scheduling ↔ non-preemptive

Implicit deadlines: $d(\tau_i) = p(\tau_i)$

Constrained deadlines: $d(\tau_i) \leq p(\tau_i)$

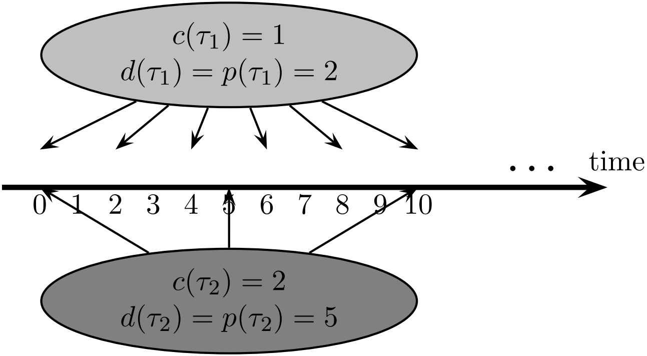
Example: Implicit deadlines & static priorities



Example: Implicit deadlines & static priorities

Theorem (Liu & Layland '73)

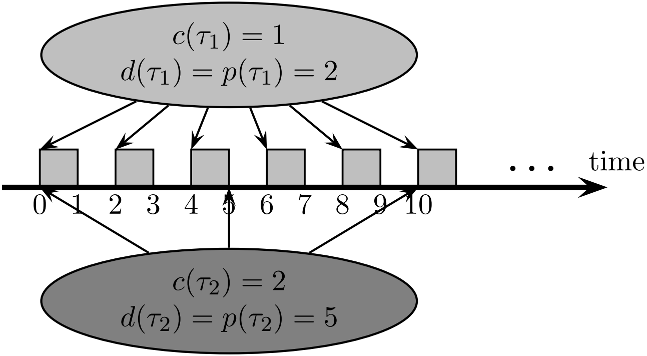
Optimal priorities: $\frac{1}{p(\tau_i)}$ *for* τ_i *(***Rate-monotonic** *schedule)*



Example: Implicit deadlines & static priorities

Theorem (Liu & Layland '73)

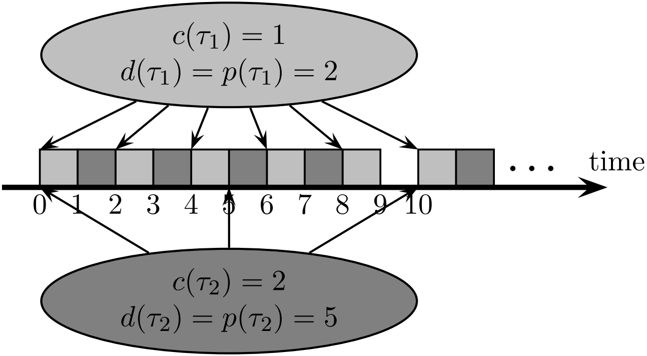
*Optimal priorities: $\frac{1}{p(\tau_i)}$ for τ_i (**Rate-monotonic** schedule)*



Example: Implicit deadlines & static priorities

Theorem (Liu & Layland '73)

*Optimal priorities: $\frac{1}{p(\tau_i)}$ for τ_i (**Rate-monotonic** schedule)*



Feasibility test for implicit-deadline tasks

Theorem (Lehoczky et al. '89)

If $p(\tau_1) \leq \ldots \leq p(\tau_n)$ *then the* **response time** $r(\tau_i)$ *in a Rate-monotonic schedule is the smallest non-negative value s.t.*

$$c(\tau_i) + \sum_{j<i} \left\lceil \frac{r(\tau_i)}{p(\tau_j)} \right\rceil c(\tau_j) \leq r(\tau_i)$$

1 machine suffices $\Leftrightarrow \forall i : r(\tau_i) \leq p(\tau_i)$.

Simultaneous Diophantine Approximation (SDA)

Given:

- $\alpha_1, \ldots, \alpha_n \in \mathbb{Q}$
- bound $N \in \mathbb{N}$
- error bound $\varepsilon > 0$

Decide:

$$\exists Q \in \{1, \ldots, N\} : \max_{i=1,\ldots,n} \left| \alpha_i - \frac{\mathbb{Z}}{Q} \right| \leq \frac{\varepsilon}{Q}$$

Simultaneous Diophantine Approximation (SDA)

Given:

- $\alpha_1, \ldots, \alpha_n \in \mathbb{Q}$
- bound $N \in \mathbb{N}$
- error bound $\varepsilon > 0$

Decide:

$$\begin{aligned} & \exists Q \in \{1, \ldots, N\} : \max_{i=1,\ldots,n} \left| \alpha_i - \frac{\mathbb{Z}}{Q} \right| \leq \frac{\varepsilon}{Q} \\ \Leftrightarrow \quad & \exists Q \in \{1, \ldots, N\} : \max_{i=1,\ldots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq \varepsilon \end{aligned}$$

Simultaneous Diophantine Approximation (SDA)

Given:

- $\alpha_1, \dots, \alpha_n \in \mathbb{Q}$
- bound $N \in \mathbb{N}$
- error bound $\varepsilon > 0$

Decide:

$$\exists Q \in \{1, \dots, N\} : \max_{i=1,\dots,n} \left| \alpha_i - \frac{\mathbb{Z}}{Q} \right| \leq \frac{\varepsilon}{Q}$$

$$\Leftrightarrow \quad \exists Q \in \{1, \dots, N\} : \max_{i=1,\dots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq \varepsilon$$

- **NP**-hard [Lagarias '85]

Simultaneous Diophantine Approximation (SDA)

Given:

- $\alpha_1, \dots, \alpha_n \in \mathbb{Q}$
- bound $N \in \mathbb{N}$
- error bound $\varepsilon > 0$

Decide:

$$\exists Q \in \{1, \dots, N\} : \max_{i=1,\dots,n} \left| \alpha_i - \frac{\mathbb{Z}}{Q} \right| \leq \frac{\varepsilon}{Q}$$

$$\Leftrightarrow \quad \exists Q \in \{1, \dots, N\} : \max_{i=1,\dots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq \varepsilon$$

- **NP**-hard [Lagarias ’85]
- Gap version **NP**-hard [Rössner & Seifert ’96, Chen & Meng ’07]

Simultaneous Diophantine Approximation (2)

Theorem (Rössner & Seifert '96, Chen & Meng '07)

Given $\alpha_1, \ldots, \alpha_n$, N, $\varepsilon > 0$ *it is* **NP**-*hard to distinguish*

- $\exists Q \in \{1, \ldots, N\} : \max\limits_{i=1,\ldots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq \varepsilon$
- $\nexists Q \in \{1, \ldots, n^{\frac{O(1)}{\log\log n}} N\} : \max\limits_{i=1,\ldots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq n^{\frac{O(1)}{\log\log n}} \varepsilon$

Simultaneous Diophantine Approximation (2)

Theorem (Rössner & Seifert '96, Chen & Meng '07)

Given $\alpha_1, \ldots, \alpha_n$, N, $\varepsilon > 0$ *it is* **NP***-hard to distinguish*

- $\exists Q \in \{1, \ldots, N\} : \max_{i=1,\ldots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq \varepsilon$
- $\nexists Q \in \{1, \ldots, n^{\frac{O(1)}{\log\log n}} N\} : \max_{i=1,\ldots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq n^{\frac{O(1)}{\log\log n}} \varepsilon$

Theorem (Eisenbrand & R. - APPROX'09)

Given $\alpha_1, \ldots, \alpha_n$, N, $\varepsilon > 0$ *it is* **NP***-hard to distinguish*

- $\exists Q \in \{N/2, \ldots, N\} : \max_{i=1,\ldots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq \varepsilon$
- $\nexists Q \in \{1, \ldots, 2^{n^{\mathcal{O}(1)}} \cdot N\} : \max_{i=1,\ldots,n} |\lceil Q\alpha_i \rfloor - Q\alpha_i| \leq n^{\frac{O(1)}{\log\log n}} \cdot \varepsilon$

even if $\varepsilon \leq (\frac{1}{2})^{n^{\mathcal{O}(1)}}$.

Directed Diophantine Approximation (DDA)

Theorem (Eisenbrand & R. - APPROX'09)

Given $\alpha_1, \dots, \alpha_n$, N, $\varepsilon > 0$ *it is* **NP**-*hard to distinguish*

- $\exists Q \in \{N/2, \dots, N\} : \max_{i=1,\dots,n} |\lceil Q\alpha_i \rceil - Q\alpha_i| \leq \varepsilon$
- $\nexists Q \in \{1, \dots, n^{\frac{O(1)}{\log \log n}} \cdot N\} : \max_{i=1,\dots,n} |\lceil Q\alpha_i \rceil - Q\alpha_i| \leq 2^{n^{\mathcal{O}(1)}} \cdot \varepsilon$

even if $\varepsilon \leq (\frac{1}{2})^{n^{\mathcal{O}(1)}}$.

Directed Diophantine Approximation (DDA)

Theorem (Eisenbrand & R. - APPROX'09)

Given $\alpha_1, \ldots, \alpha_n$*,* N*,* $\varepsilon > 0$ *it is* **NP***-hard to distinguish*

- $\exists Q \in \{N/2, \ldots, N\} : \max_{i=1,\ldots,n} |\lceil Q\alpha_i \rceil - Q\alpha_i| \leq \varepsilon$
- $\nexists Q \in \{1, \ldots, n^{\frac{O(1)}{\log\log n}} \cdot N\} : \max_{i=1,\ldots,n} |\lceil Q\alpha_i \rceil - Q\alpha_i| \leq 2^{n^{\mathcal{O}(1)}} \cdot \varepsilon$

even if $\varepsilon \leq (\frac{1}{2})^{n^{\mathcal{O}(1)}}$.

Theorem (Eisenbrand & R. - SODA'10)

Given $\alpha_1, \ldots, \alpha_n$*,* $w_1, \ldots, w_n \geq 0$*,* N*,* $\varepsilon > 0$ *it is* **NP***-hard to distinguish*

- $\exists Q \in [N/2, N] : \sum_{i=1}^n w_i \, (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon$
- $\nexists Q \in [1, n^{\frac{O(1)}{\log\log n}} \cdot N] : \sum_{i=1}^n w_i \, (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq 2^{n^{\mathcal{O}(1)}} \cdot \varepsilon$

even if $\varepsilon \leq (\frac{1}{2})^{n^{\mathcal{O}(1)}}$.

Hardness of Response Time Computation

Theorem (Eisenbrand & R. - RTSS'08)

Computing response times for implicit-deadline tasks w.r.t. to a Rate-monotonic schedule, i.e. solving

$$\min \left\{ r \geq 0 \mid c(\tau_n) + \sum_{i=1}^{n-1} \left\lceil \frac{r}{p(\tau_i)} \right\rceil c(\tau_i) \leq r \right\}$$

$(p(\tau_1) \leq \ldots \leq p(\tau_n))$ is **NP***-hard (even to approximate within a factor of $n^{\frac{\mathcal{O}(1)}{\log \log n}}$).*

- Reduction from Directed Diophantine Approximation

Mixing Set

$$
\begin{aligned}
\min c_s s + c^T y & & \\
s + a_i y_i & \geq & b_i \quad \forall i = 1, \ldots, n \\
s & \in & \mathbb{R}_{\geq 0} \\
y & \in & \mathbb{Z}^n
\end{aligned}
$$

Mixing Set

$$\begin{aligned} \min c_s s + c^T y & & \\ s + a_i y_i &\geq b_i \quad \forall i = 1, \ldots, n \\ s &\in \mathbb{R}_{\geq 0} \\ y &\in \mathbb{Z}^n \end{aligned}$$

- Complexity status? [Conforti, Di Summa & Wolsey '08]

Mixing Set

$$\begin{aligned} \min c_s s + c^T y & & & \\ s + a_i y_i &\geq & b_i & \quad \forall i = 1, \ldots, n \\ s &\in & \mathbb{R}_{\geq 0} & \\ y &\in & \mathbb{Z}^n & \end{aligned}$$

- Complexity status? [Conforti, Di Summa & Wolsey '08]

Theorem (Eisenbrand & R. - APPROX'09)

Solving Mixing Set is **NP**-*hard.*

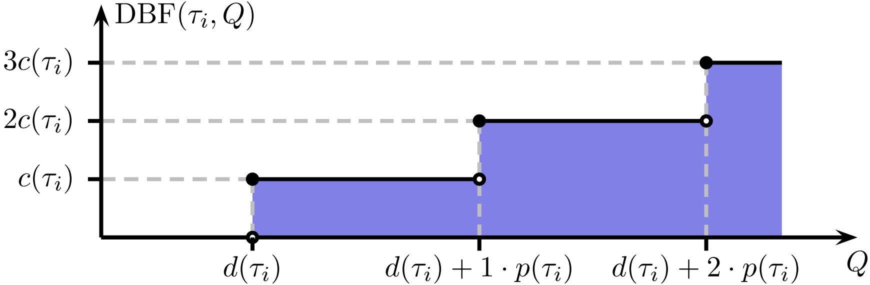
1. Model Directed Diophantine Approximation (almost) as Mixing Set
2. Simulate missing constraint with Lagrangian relaxation

Testing EDF-schedulability

Setting: Constrained deadline tasks, i.e. $d(\tau_i) \leq p(\tau_i)$

Theorem (Baruah, Mok & Rosier '90)

$$\underbrace{\left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) \cdot c(\tau_i)}_{=:\, DBF(\tau_i, Q)}$$



Testing EDF-schedulability

Setting: Constrained deadline tasks, i.e. $d(\tau_i) \leq p(\tau_i)$

Theorem (Baruah, Mok & Rosier '90)

Earliest-Deadline-First schedule is feasible iff

$$\forall Q \geq 0 : \sum_{i=1}^{n} \underbrace{\left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) \cdot c(\tau_i)}_{=: DBF(\tau_i, Q)} \leq Q$$

Testing EDF-schedulability

Setting: Constrained deadline tasks, i.e. $d(\tau_i) \leq p(\tau_i)$

Theorem (Baruah, Mok & Rosier '90)

Earliest-Deadline-First schedule is feasible iff

$$\forall Q \geq 0 : \sum_{i=1}^{n} \underbrace{\left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) \cdot c(\tau_i)}_{=: DBF(\tau_i, Q)} \leq Q$$

- Complexity status is "Problem 2" in *Open Problems in Real-Time Scheduling* [Baruah & Pruhs '09]

Testing EDF-schedulability

Setting: Constrained deadline tasks, i.e. $d(\tau_i) \leq p(\tau_i)$

Theorem (Baruah, Mok & Rosier '90)

Earliest-Deadline-First schedule is feasible iff

$$\forall Q \geq 0 : \sum_{i=1}^{n} \underbrace{\left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) \cdot c(\tau_i)}_{=: DBF(\tau_i, Q)} \leq Q$$

- Complexity status is "Problem 2" in *Open Problems in Real-Time Scheduling* [Baruah & Pruhs '09]

Theorem (Eisenbrand & R. - SODA'10)

Testing EDF-schedulability is **coNP***-hard.*

Reduction (1)

Given: DDA instance $\alpha_1, \ldots, \alpha_n$, $w_1, \ldots, w_n$, $N \in \mathbb{N}$, $\varepsilon > 0$

Reduction (1)

Given: DDA instance $\alpha_1, \dots, \alpha_n,\ w_1, \dots, w_n,\ N \in \mathbb{N},\ \varepsilon > 0$

Define: Task set $\mathcal{S} = \{\tau_0, \dots, \tau_n\}$ s.t.

- **Yes:** $\exists Q \in [N/2, N] : \sum_{i=1}^{n} w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon$
  $\Rightarrow \mathcal{S}$ not EDF-schedulable $(\exists Q \geq 0 : \mathrm{DBF}(\mathcal{S}, Q) > \beta \cdot Q)$
- **No:** $\nexists Q \in [N/2, 3N] : \sum_{i=1}^{n} w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq 3\varepsilon$
  $\Rightarrow \mathcal{S}$ EDF-schedulable $(\forall Q \geq 0 : \mathrm{DBF}(\mathcal{S}, Q) \leq \beta \cdot Q)$

Reduction (1)

Given: DDA instance $\alpha_1, \dots, \alpha_n,\ w_1, \dots, w_n,\ N \in \mathbb{N},\ \varepsilon > 0$

Define: Task set $\mathcal{S} = \{\tau_0, \dots, \tau_n\}$ s.t.

- **Yes:** $\exists Q \in [N/2, N] : \sum_{i=1}^{n} w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon$
  $\Rightarrow \mathcal{S}$ not EDF-schedulable ($\exists Q \geq 0 : \mathrm{DBF}(\mathcal{S}, Q) > \beta \cdot Q$)
- **No:** $\nexists Q \in [N/2, 3N] : \sum_{i=1}^{n} w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq 3\varepsilon$
  $\Rightarrow \mathcal{S}$ EDF-schedulable ($\forall Q \geq 0 : \mathrm{DBF}(\mathcal{S}, Q) \leq \beta \cdot Q$)

Task system:

$$\tau_i = (c(\tau_i), d(\tau_i), p(\tau_i)) := \left(w_i, \frac{1}{\alpha_i}, \frac{1}{\alpha_i}\right) \quad \forall i = 1, \dots, n$$

$$U := \sum_{i=1}^{n} \frac{c(\tau_i)}{p(\tau_i)}$$

Reduction (2)

$$Q \cdot U - \sum_{i=1}^{n} \left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) c(\tau_i)$$

Reduction (2)

$$Q \cdot U - \sum_{i=1}^{n} \left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) c(\tau_i) \quad = \quad \sum_{i=1}^{n} \left(\frac{Q}{p(\tau_i)} - \left\lfloor \frac{Q}{p(\tau_i)} \right\rfloor \right) c(\tau_i)$$

Reduction (2)

$$\begin{aligned} Q \cdot U - \sum_{i=1}^{n} \left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) c(\tau_i) &= \sum_{i=1}^{n} \left(\frac{Q}{p(\tau_i)} - \left\lfloor \frac{Q}{p(\tau_i)} \right\rfloor \right) c(\tau_i) \\ &= \sum_{i=1}^{n} w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \end{aligned}$$

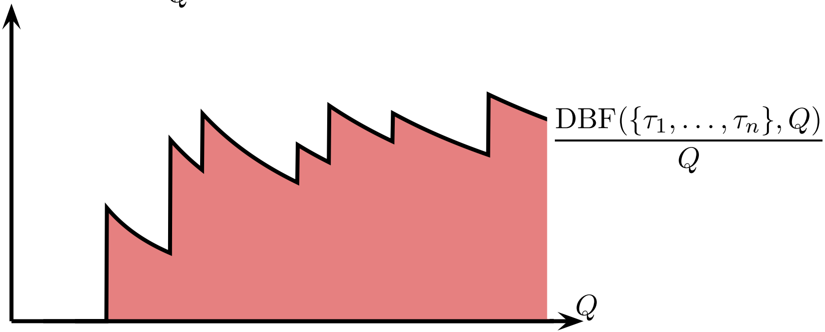
Reduction (2)

$$\begin{aligned}
Q \cdot U - \sum_{i=1}^{n} \left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) c(\tau_i) \quad &= \quad \sum_{i=1}^{n} \left(\frac{Q}{p(\tau_i)} - \left\lfloor \frac{Q}{p(\tau_i)} \right\rfloor \right) c(\tau_i) \\
&= \quad \sum_{i=1}^{n} w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \\
\Rightarrow \frac{\mathrm{DBF}(\{\tau_1, \ldots, \tau_n\}, Q)}{Q} \approx U \quad &\Leftrightarrow \quad \text{approx. error small}
\end{aligned}$$

Reduction (2)

$$
\begin{aligned}
Q \cdot U - \sum_{i=1}^{n} \left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) c(\tau_i) \quad &= \quad \sum_{i=1}^{n} \left(\frac{Q}{p(\tau_i)} - \left\lfloor \frac{Q}{p(\tau_i)} \right\rfloor \right) c(\tau_i) \\
&= \quad \sum_{i=1}^{n} w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor)
\end{aligned}
$$

$$\Rightarrow \frac{\mathrm{DBF}(\{\tau_1, \ldots, \tau_n\}, Q)}{Q} \approx U \quad \Leftrightarrow \quad \text{approx. error small}$$

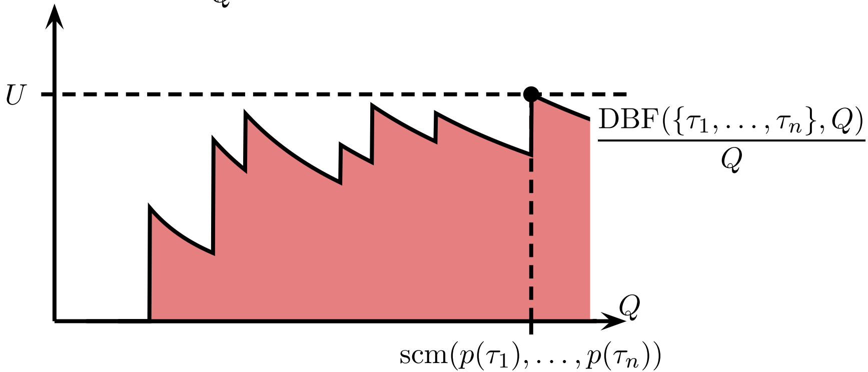


Reduction (2)

$$Q \cdot U - \sum_{i=1}^{n} \left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) c(\tau_i) \quad = \quad \sum_{i=1}^{n} \left(\frac{Q}{p(\tau_i)} - \left\lfloor \frac{Q}{p(\tau_i)} \right\rfloor \right) c(\tau_i)$$

$$= \quad \sum_{i=1}^{n} w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor)$$

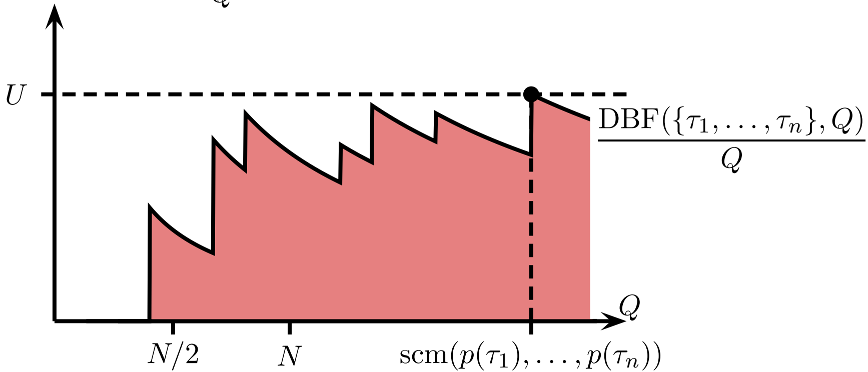
$$\Rightarrow \frac{\mathrm{DBF}(\{\tau_1, \ldots, \tau_n\}, Q)}{Q} \approx U \quad \Leftrightarrow \quad \text{approx. error small}$$



Reduction (2)

$$\begin{aligned} Q \cdot U - \sum_{i=1}^{n} \left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) c(\tau_i) \quad &= \quad \sum_{i=1}^{n} \left(\frac{Q}{p(\tau_i)} - \left\lfloor \frac{Q}{p(\tau_i)} \right\rfloor \right) c(\tau_i) \\ &= \quad \sum_{i=1}^{n} w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \end{aligned}$$

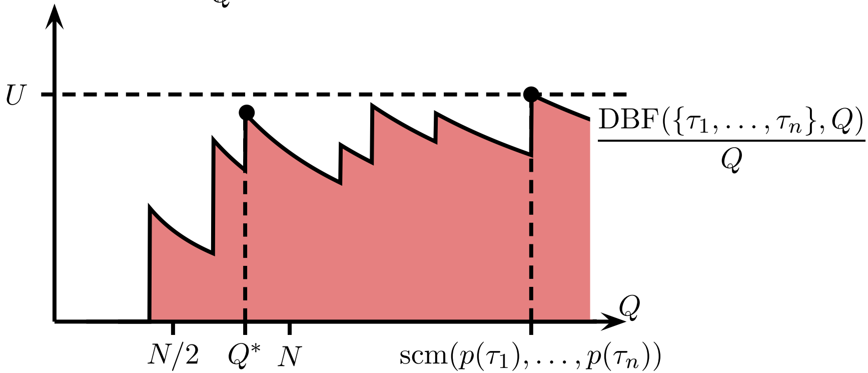
$$\Rightarrow \frac{\mathrm{DBF}(\{\tau_1, \ldots, \tau_n\}, Q)}{Q} \approx U \quad \Leftrightarrow \quad \text{approx. error small}$$



Reduction (2)

$$
\begin{aligned}
Q \cdot U - \sum_{i=1}^{n} \left(\left\lfloor \frac{Q - d(\tau_i)}{p(\tau_i)} \right\rfloor + 1 \right) c(\tau_i) &= \sum_{i=1}^{n} \left(\frac{Q}{p(\tau_i)} - \left\lfloor \frac{Q}{p(\tau_i)} \right\rfloor \right) c(\tau_i) \\
&= \sum_{i=1}^{n} w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor)
\end{aligned}
$$

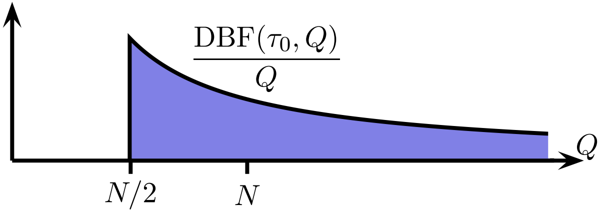
$$
\Rightarrow \frac{\mathrm{DBF}(\{\tau_1, \ldots, \tau_n\}, Q)}{Q} \approx U \quad \Leftrightarrow \quad \text{approx. error small}
$$



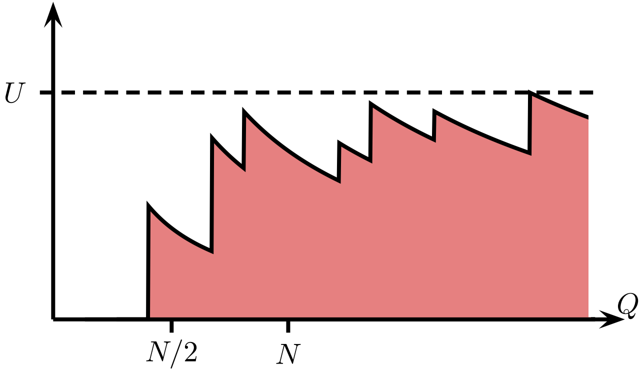
Reduction (3)

Add a special task

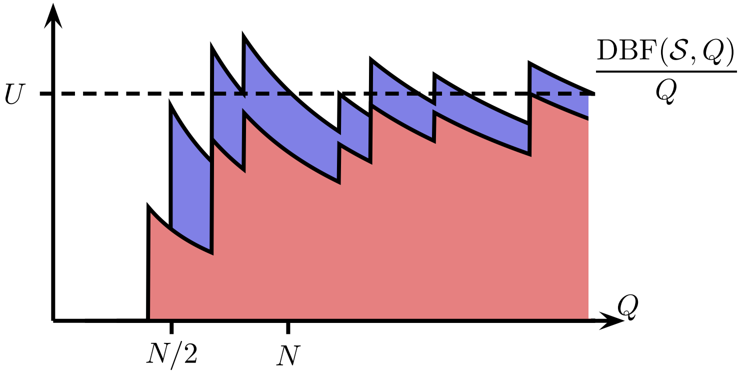
$$\tau_0 = (c(\tau_0), d(\tau_0), p(\tau_0)) := (3\varepsilon, N/2, \infty)$$



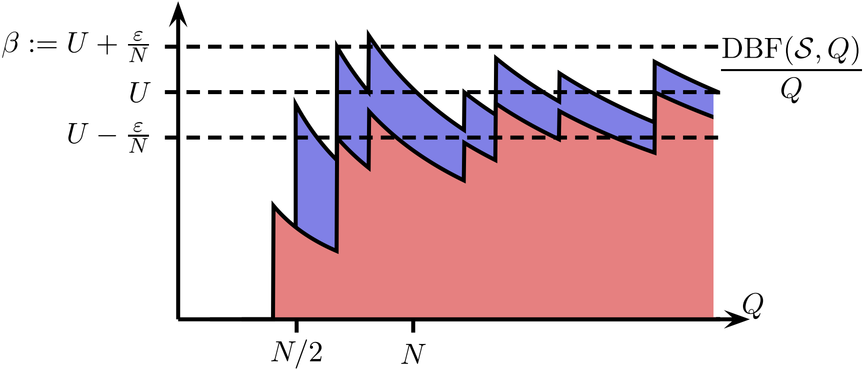
Reduction (4)



Reduction (4)



Reduction (4)


$\beta := U + \frac{\varepsilon}{N}$
U
$U - \frac{\varepsilon}{N}$
$\frac{\mathrm{DBF}(\mathcal{S},Q)}{Q}$
Q
$N/2$
N

Algorithms for Multi-processor Scheduling

Setting: Implicit deadlines ($d(\tau_i) = p(\tau_i)$), multi-processor

Algorithms for Multi-processor Scheduling

Setting: Implicit deadlines ($d(\tau_i) = p(\tau_i)$), multi-processor

Known results:

- **NP**-hard (generalization of Bin Packing)

Algorithms for Multi-processor Scheduling

Setting: Implicit deadlines ($d(\tau_i) = p(\tau_i)$), multi-processor

Known results:

- **NP**-hard (generalization of Bin Packing)
- asymp. 1.75-apx [Burchard et al. '95]

Algorithms for Multi-processor Scheduling

Setting: Implicit deadlines ($d(\tau_i) = p(\tau_i)$), multi-processor

Known results:

- **NP**-hard (generalization of Bin Packing)
- asymp. 1.75-apx [Burchard et al. '95]

Theorem (Eisenbrand & R. - ICALP'08)

In time $\mathcal{O}_\varepsilon(1) \cdot n^{(1/\varepsilon)^{\mathcal{O}(1)}}$ one can find an assignment to

$$APX \leq (1+\varepsilon) \cdot OPT + \mathcal{O}(1)$$

machines such that the RM-schedule is feasible if the machines are speed up by a factor of $1+\varepsilon$ (resource augmentation).

1. Rounding, clustering & merging small tasks
2. Use relaxed feasibility notion
3. Dynamic programming

Algorithms for Multi-processor Scheduling (2)

Theorem

Let $k \in \mathbb{N}$ be an arbitrary parameter. In time $\mathcal{O}(n^3)$ one can find an assignment of implicit-deadline tasks to

$$APX \leq \left(\frac{3}{2} + \frac{1}{k}\right) OPT + 9k$$

machines ($\rightarrow$ asymptotic $\frac{3}{2}$-apx).

1. Create graph $G = (\mathcal{S}, E)$ with tasks as nodes and edges

   $$(\tau_1, \tau_2) \in E :\Leftrightarrow \{\tau_1, \tau_2\} \text{ RM-schedulable (on 1 machine)}$$

2. Define suitable edge weights
3. Mincost matching $\Rightarrow$ good assignment

Algorithms for Multi-processor Scheduling (3)

$$\mathcal{P} \quad = \quad \{x \in \{0,1\}^n \mid \{\tau_i \mid x_i = 1\} \text{ RM-schedulable}\}$$

Algorithms for Multi-processor Scheduling (3)

$$\begin{aligned}
\mathcal{P} &= \{x \in \{0,1\}^n \mid \{\tau_i \mid x_i = 1\} \text{ RM-schedulable}\} \\
\lambda_x &= \begin{cases} 1 & \text{pack machine with } \{\tau_i \mid x_i = 1\} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

Algorithms for Multi-processor Scheduling (3)

$$\begin{aligned}
\mathcal{P} &= \{x \in \{0,1\}^n \mid \{\tau_i \mid x_i = 1\} \text{ RM-schedulable}\} \\
\lambda_x &= \begin{cases} 1 & \text{pack machine with } \{\tau_i \mid x_i = 1\} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

Column-based LP (or configuration LP):

$$\begin{aligned}
\min \mathbf{1}^T \lambda & \\
\sum_{x \in \mathcal{P}} \lambda_x x &\geq \mathbf{1} \\
\lambda &\geq \mathbf{0}
\end{aligned}$$

Algorithms for Multi-processor Scheduling (3)

$$
\begin{aligned}
\mathcal{P} &= \{x \in \{0,1\}^n \mid \{\tau_i \mid x_i = 1\} \text{ RM-schedulable}\} \\
\lambda_x &= \begin{cases} 1 & \text{pack machine with } \{\tau_i \mid x_i = 1\} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

Column-based LP (or configuration LP):

$$
\begin{aligned}
\min \mathbf{1}^T \lambda & \\
\sum_{x \in \mathcal{P}} \lambda_x x &\geq \mathbf{1} \\
\lambda &\geq \mathbf{0}
\end{aligned}
$$

- For Bin Packing: $OPT - OPT_f \leq \mathcal{O}(\log^2 n)$ [Karmarkar & Karp '82]

Algorithms for Multi-processor Scheduling (3)

$$\begin{aligned}\mathcal{P} &= \{x \in \{0,1\}^n \mid \{\tau_i \mid x_i = 1\} \text{ RM-schedulable}\} \\ \lambda_x &= \begin{cases} 1 & \text{pack machine with } \{\tau_i \mid x_i = 1\} \\ 0 & \text{otherwise} \end{cases}\end{aligned}$$

Column-based LP (or configuration LP):

$$\begin{aligned} \min \mathbf{1}^T \lambda & \\ \sum_{x \in \mathcal{P}} \lambda_x x &\geq \mathbf{1} \\ \lambda &\geq \mathbf{0} \end{aligned}$$

- For Bin Packing: $OPT - OPT_f \leq \mathcal{O}(\log^2 n)$ [Karmarkar & Karp '82]

Theorem

$$1.33 \approx \frac{4}{3} \leq \sup_{instances} \left\{ \frac{OPT}{OPT_f} \right\} \leq 1 + \ln(2) \approx 1.69$$

Algorithms for Multi-processor Scheduling (4)

Theorem (Eisenbrand & R. - ICALP'08)

For all $\varepsilon > 0$, there is no polynomial time algorithm with

$$APX \leq OPT + n^{1-\varepsilon}$$

unless $\mathbf{P} = \mathbf{NP}$ *($\Rightarrow$ no AFPTAS).*

- Cloning of 3-Partition instances

Algorithms for Multi-processor Scheduling (5)

- FFMP is a simple First Fit like heuristic with running time $\mathcal{O}(n \log n)$

Algorithms for Multi-processor Scheduling (5)

- FFMP is a simple First Fit like heuristic with running time $\mathcal{O}(n \log n)$

Theorem (Karrenbauer & R. - ESA'09)

For n tasks $\mathcal{S} = \{\tau_1, \dots, \tau_n\}$ with $u(\tau_i) \in [0,1]$ uniformly at random

$$E[FFMP(\mathcal{S})] \leq E[u(\mathcal{S})] + \mathcal{O}(n^{3/4}(\log n)^{3/8})$$

(average utilization $\rightarrow 100\%$).

- Reduce to known results from the average case analysis of Bin Packing

Dynamic vs. static priorities

- **Setting:** constrained-deadline tasks, i.e. $d(\tau_i) \leq p(\tau_i)$

Dynamic vs. static priorities

- **Setting:** constrained-deadline tasks, i.e. $d(\tau_i) \leq p(\tau_i)$
- Worst-case speedup factor

$$f := \sup_{\text{instances}} \left\{ \frac{\text{proc. speed for static priority schedule}}{\text{proc. speed for arbitrary schedule}} \right\} \geq 1$$

Dynamic vs. static priorities

- **Setting:** constrained-deadline tasks, i.e. $d(\tau_i) \leq p(\tau_i)$
- Worst-case speedup factor

$$f := \sup_{\text{instances}} \left\{ \frac{\text{proc. speed for static priority schedule}}{\text{proc. speed for arbitrary schedule}} \right\} \geq 1$$

- Known: $\frac{3}{2} \leq f \leq 2$ [Baruah & Burns '08]

Dynamic vs. static priorities

- **Setting:** constrained-deadline tasks, i.e. $d(\tau_i) \leq p(\tau_i)$
- Worst-case speedup factor

$$f := \sup_{\text{instances}} \left\{ \frac{\text{proc. speed for static priority schedule}}{\text{proc. speed for arbitrary schedule}} \right\} \geq 1$$

- Known: $\frac{3}{2} \leq f \leq 2$ [Baruah & Burns '08]

Theorem (Davis, R., Baruah & Burns - RT Systems '09)

$$f = \frac{1}{\Omega} \approx 1.76$$

where $\Omega \approx 0.567$ *is the unique positive real root of* $x = \ln(1/x)$.

The end

Thanks for your attention