ECRTS 2012 in Pisa, Italy
11-13 July 2012

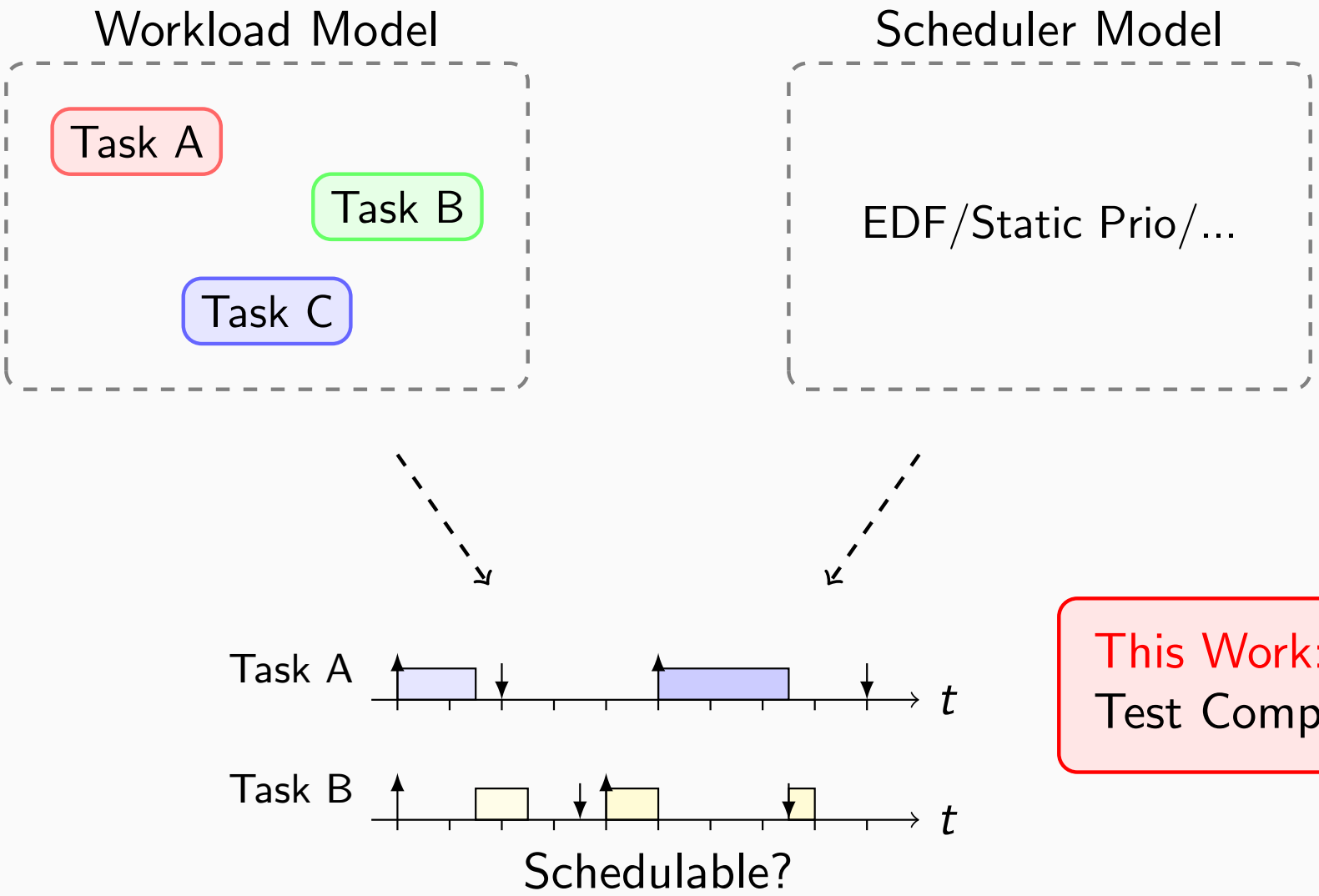
Hardness Results for Static Priority Real-Time Scheduling

Martin Stigge

Uppsala University, Sweden

Joint work with Wang Yi

Problem Overview



Model/Design Choices

Task Models:

- Periodic (L&L)
- Generalized Multiframe (GMF)
- Digraph Real-Time (DRT)
- ...

Schedulers:

- Dynamic Priorities: EDF
- Static Priorities

Different combinations = Different complexity

(Here: Uniprocessor, preemption, precise tests)

Complexity of Schedulability Test

- Efficient schedulability tests possible?
  - ("Efficient" = "pseudo-polynomial")

	EDF	Static
L&L	Yes	Yes
GMF	Yes	~~Yes*~~ No!
DRT	Yes	No
EDRT	No	No

- * = Takada & Sakamura, 1997
- *Flawed!*

Theorem (Our technical result)

For GMF task systems, the schedulability problem for static priority schedulers is strongly coNP-hard.

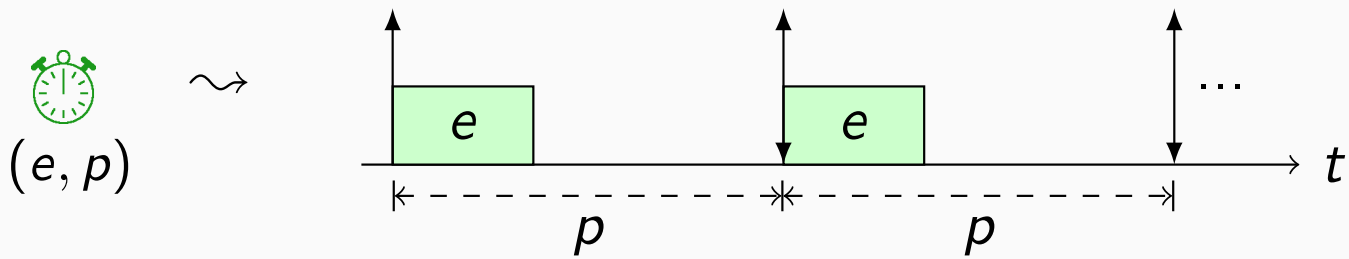
Fahrplan

1. Problem Overview
2. Task Models: L&L, GMF, DRT
3. Analysis Methods
   - EDF: Demand Bound Function
   - Static Priorities: Maximum Interference Function
4. Hardness Result

The Liu and Layland (L&L) Task Model

(Liu and Layland, 1973)

- Tasks are *periodic*
  - Job WCET e
  - Minimum inter-release delay p (implicit deadline)

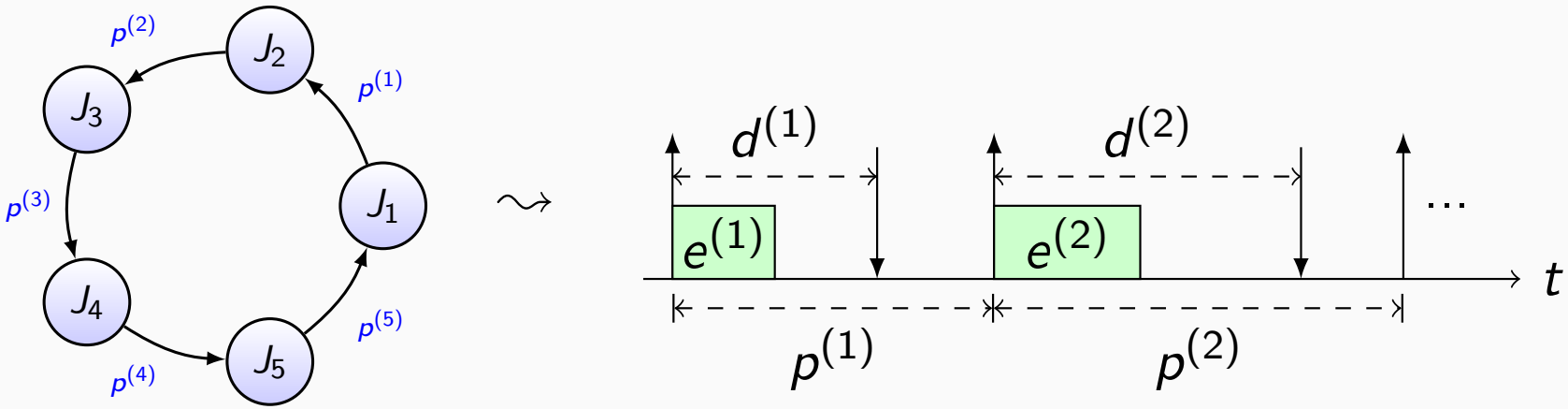


- Advantages: Well-known model; efficient schedulability tests
- However, *not everything is periodic*...

The General Multiframe (GMF) Task Model

(Baruah et al, 1999)

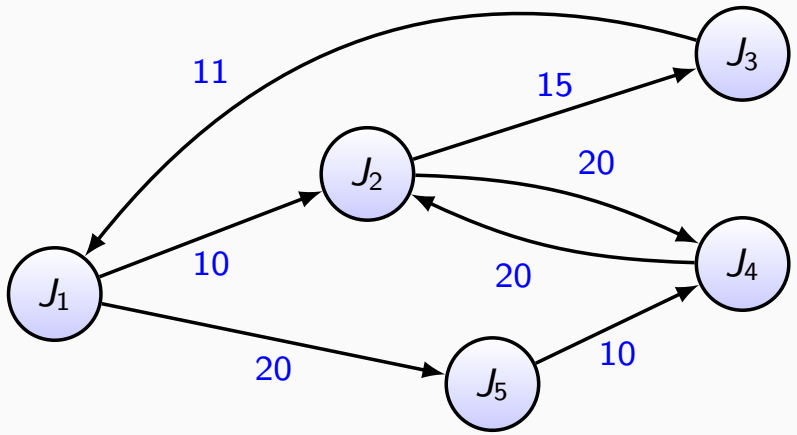
- Tasks *cycle* through job types, "frames"
  - Vector for WCET $(e^{(1)}, \ldots, e^{(n)})$
  - Vector for deadlines $(d^{(1)}, \ldots, e^{(n)})$
  - Vector for minimum inter-release delays $(p^{(1)}, \ldots, p^{(n)})$



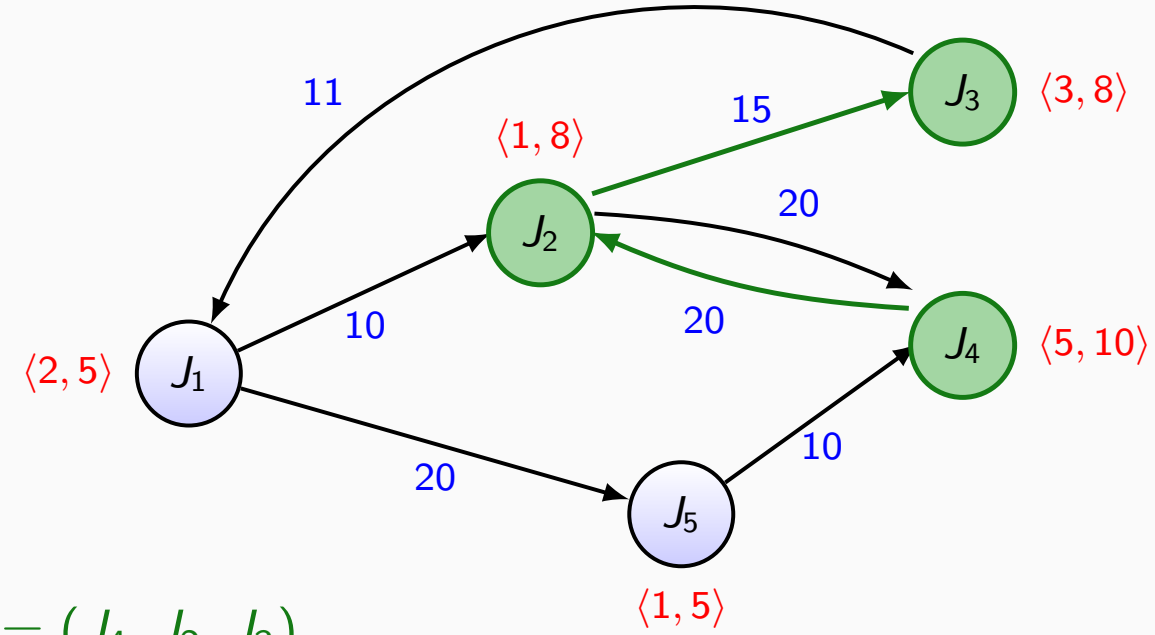
The Digraph Real-Time (DRT) Task Model

(S. et al, RTAS 2011)

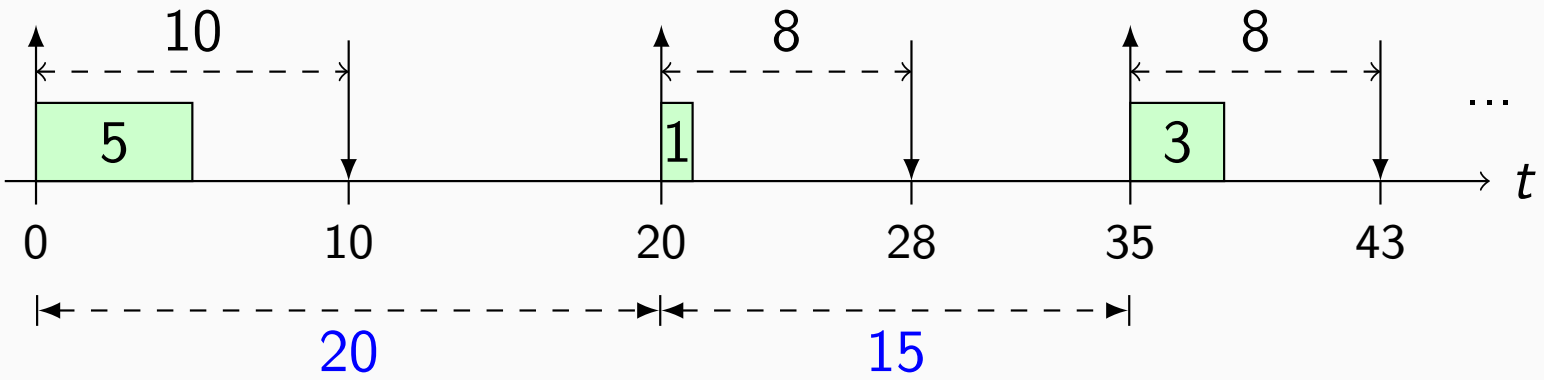
- Branching, cycles (loops), ...
- *Directed graph* for each task
  - Vertices J: jobs to be released (with WCET and deadline)
  - Edges (J_i, J_j): minimum inter-release delays $p(J_i, J_j)$



DRT: Semantics



Path $\pi = (J_4, J_2, J_3)$



Fahrplan

1. Problem Overview
2. Task Models: L&L, GMF, DRT
3. Analysis Methods
   - EDF: Demand Bound Function
   - Static Priorities: Maximum Interference Function
4. Hardness Result

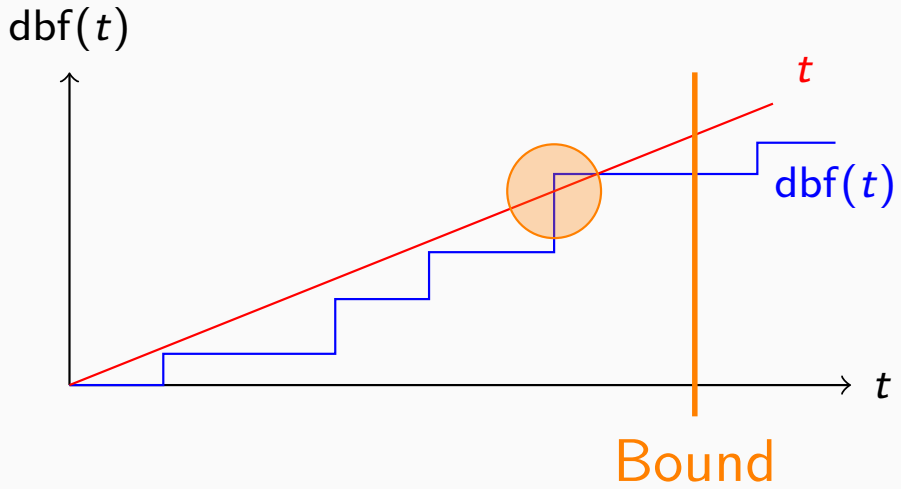
The Demand Bound Function

- Given a time interval length t
- $\mathrm{dbf}(t)$ bounds the *demand* for processor time within *any* t interval

Theorem

A task system τ is schedulable with EDF iff:

$$\forall t \geqslant 0 : \sum_{T \in \tau} \mathrm{dbf}_T(t) \leqslant t$$



Complexity of Schedulability Test

- Efficient schedulability tests:

	EDF	Static
L&L	Yes	Yes
GMF	Yes	?
DRT	Yes	
EDRT	No	

Schedulability for Static Priorities

- L&L tasks: *Response Time Analysis*

$$R_i = C_i + \underbrace{\sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \cdot C_j}_{\text{Interference Term}}$$

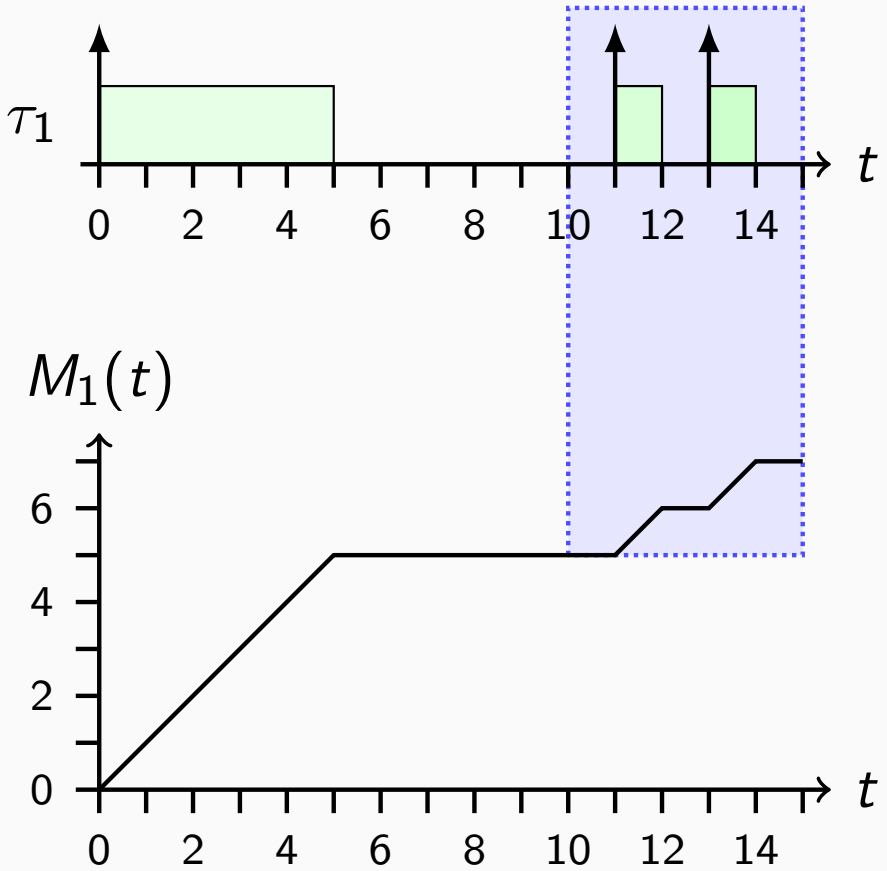
- Generalize for GMF: *Maximum Interference Function (MIF)*
  - $M_j(t)$: Maximum interference that τ_j can cause within t time units

$$R_i = C_i + \sum_{j \in hp(i)} M_j(R_i)$$

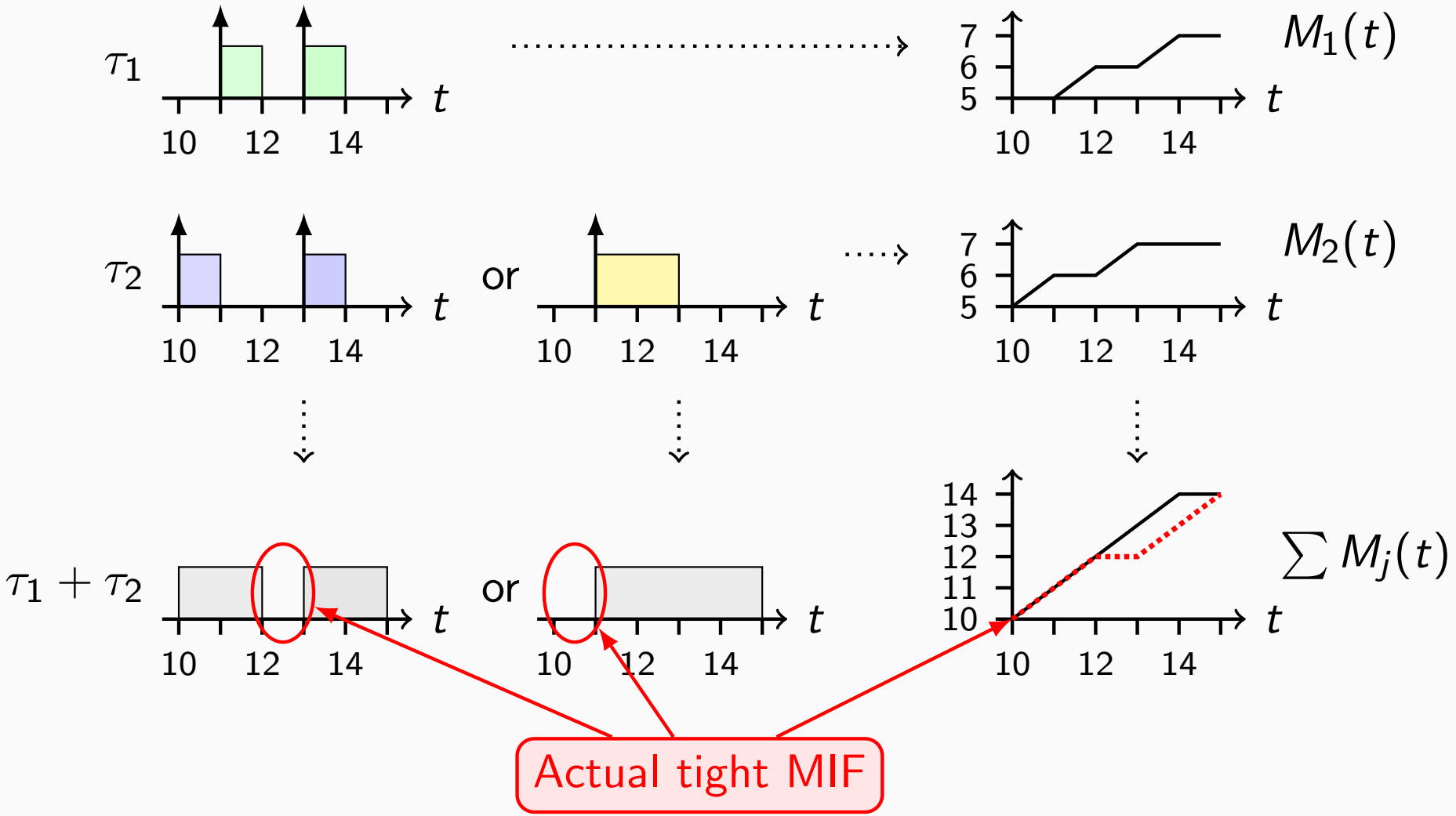
  - Efficiently computable for GMF/DRT/...

BUT: Inherently overapproximate!

MIF: Example



MIF: Combined Example



Schedulability for Static Priorities

- In summary: MIF is *pessimistic*
- Possible improvement?
  - Define MIF additions better? Precise?
  - Use another abstraction level?
  - ...?
- No!

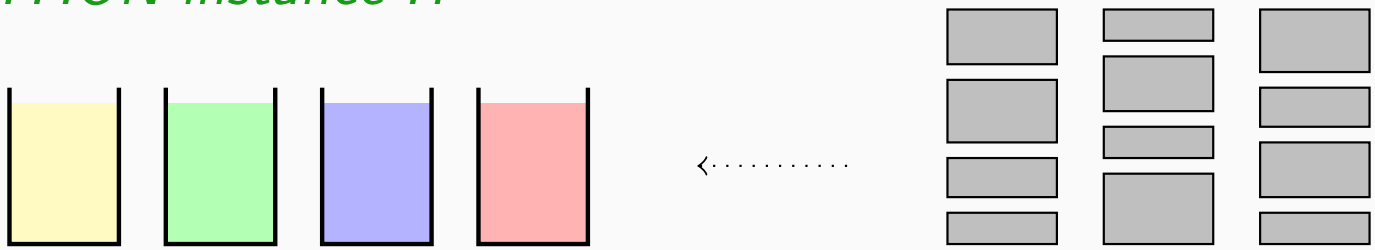
Theorem (Our technical result)

For GMF task systems, the schedulability problem for static priority schedulers is strongly coNP-hard.

- Thus: No precise efficient analysis possible.

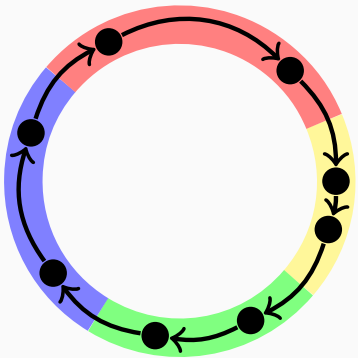
Hardness Result: Proof sketch

3-PARTITION instance I:



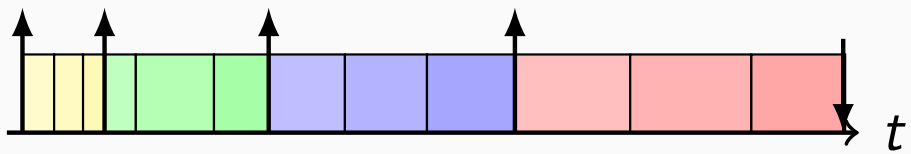
Possible to (exactly) fit all items? (strongly NP-hard)

Reduction to GMF schedulability:



One GMF task
for *each* item

Packing possible $\iff$ Busy interval



Thus: $\tau(I)$ unsched. $\iff$ $I \in$ 3-PARTITION

Summary and Outlook

	EDF	Static
L&L	Yes	Yes
GMF	Yes	~~Yes~~ No!
DRT	Yes	No
EDRT	No	No

- Showed *intractability* of static scheduling for GMF
- Insight:
  - For EDF, "simple" overload test suffices
    (Local worst cases combine.)
  - For static prio: More structure $\rightarrow$ more complex test
    (Local worst case unclear.)

- Ongoing work:
  - Solve anyway? Heuristics?
  - Use SAT-/SMT-solvers

Q & A

Thanks!