Cumulative Scheduling with Calendars and Overtime

Samuel Cloutier
Claude-Guy Quimper
September 4th, 2024

UNIVERSITÉ LAVAL

RCPSP - Reminder

Resource Constrained Project Scheduling Problem

Schedule tasks subject to :

- Precedence constraints
- Resource usage constraints

RCPSP - Reminder

Resource Constrained Project Scheduling Problem

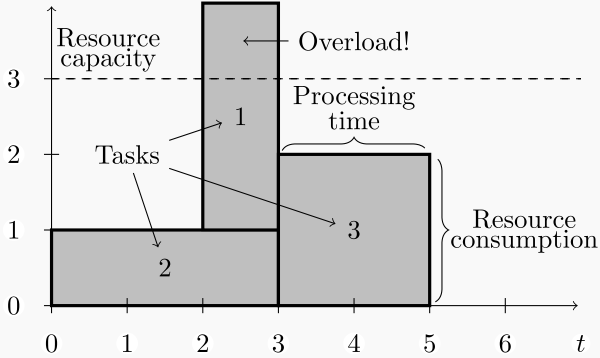
Schedule tasks subject to :

- Precedence constraints
- Resource usage constraints

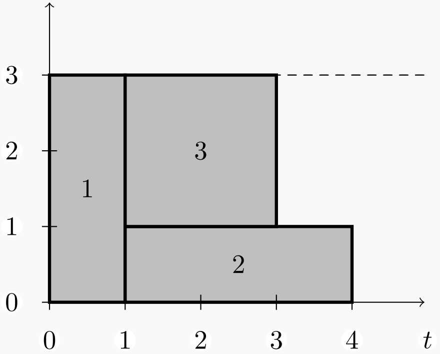
Definition of task $i \in \mathcal{I}$

- S_i : Starting time variable
- p_i : Processing time
- h_i : Resource usage

Resource usage constraints



Resource usage constraints



The CUMULATIVE constraint models this.

Calendars with Scheduled Overtime

Motivation

- In practice, not all time points are worked the same (such as weekends)
- Some time points may represent work hours that are out of the regular work schedule.

Calendars with Scheduled Overtime

Motivation

- In practice, not all time points are worked the same (such as weekends)
- Some time points may represent work hours that are out of the regular work schedule.

Inspiration

A Constraint Programming Approach to Ship Refit Project Scheduling, Boudreault et. al. (CP22) :

- Overtime is available to finish tasks earlier
- Every task follows some calendar periodic on weeks

In our generalization, calendars are arbitrary.

Symbol definitions

-  : Closed time point
-  : Regular time point
-  : Overtime time point

Calendars with Scheduled Overtime

In our generalization, calendars are arbitrary.

Symbol definitions

- 😴 : Closed time point
- ☀️🔨 : Regular time point
- 🌙🔨 : Overtime time point

Example of a calendar

A 8-16 work day with a 4 overtime hour time slot :

😴😴😴😴😴😴😴😴☀️🔨☀️🔨☀️🔨☀️🔨☀️🔨☀️🔨☀️🔨☀️🔨🌙🔨🌙🔨🌙🔨🌙🔨😴😴😴😴

Effects of calendars with overtime

New variables

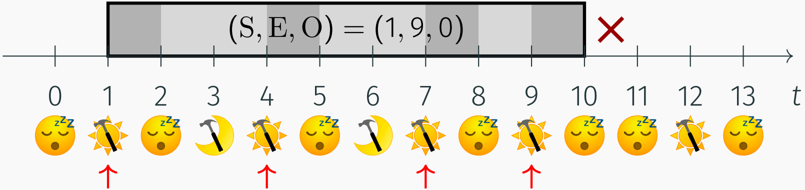
- E_i : Elapsed time (actual duration)
- O_i : Amount of overtime used

Rules

- A task may not start nor end with an unworked hour
- Time worked in an execution window is exactly the processing time of the task
- We cannot work more overtime than available in the execution window

Effects of calendars with overtime

Task of processing time 4



Domain variation

Initial domains :

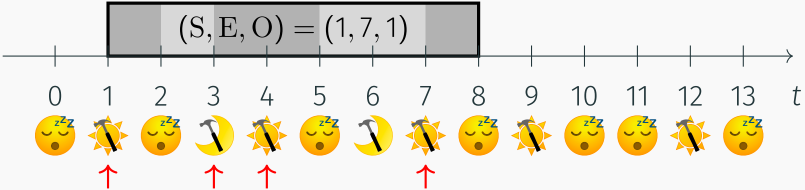
$\mathrm{dom}(S) = \{0..8\}$, $\mathrm{dom}(E) = \{0..8\}$, and $\mathrm{dom}(O) = \{0..8\}$

Filtered domain :

$\mathrm{dom}(S) = \{1..8\}$, $\mathrm{dom}(E) = \{0..8\}$, and $\mathrm{dom}(O) = \{0..8\}$

Effects of calendars with overtime

Task of processing time 4



Domain variation

Initial domains :

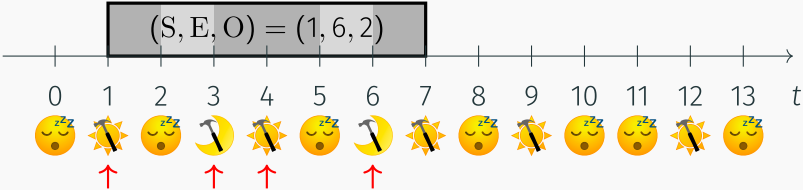
$\mathrm{dom}(\mathrm{S}) = \{0..8\}$, $\mathrm{dom}(\mathrm{E}) = \{0..8\}$, and $\mathrm{dom}(\mathrm{O}) = \{0..8\}$

Filtered domain :

$\mathrm{dom}(\mathrm{S}) = \{1..8\}$, $\mathrm{dom}(\mathrm{E}) = \{0..8\}$, and $\mathrm{dom}(\mathrm{O}) = \{0..8\}$

Effects of calendars with overtime

Task of processing time 4



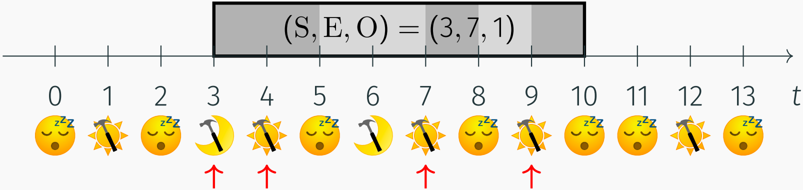
Domain variation

Initial domains :
$\mathrm{dom}(S) = \{0..8\}$, $\mathrm{dom}(E) = \{0..8\}$, and $\mathrm{dom}(O) = \{0..8\}$

Filtered domain :
$\mathrm{dom}(S) = \{1..8\}$, $\mathrm{dom}(E) = \{0..8\}$, and $\mathrm{dom}(O) = \{0..8\}$

Effects of calendars with overtime

Task of processing time 4



Domain variation

Initial domains :

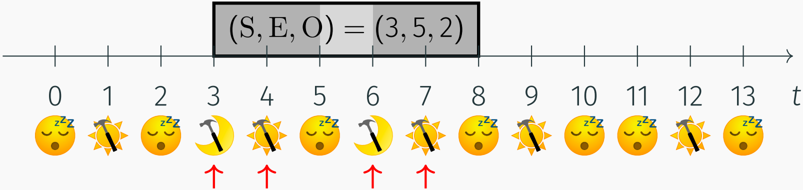
$\mathrm{dom}(\mathrm{S}) = \{0..8\}$, $\mathrm{dom}(\mathrm{E}) = \{0..8\}$, and $\mathrm{dom}(\mathrm{O}) = \{0..8\}$

Filtered domain :

$\mathrm{dom}(\mathrm{S}) = \{1..8\}$, $\mathrm{dom}(\mathrm{E}) = \{0..8\}$, and $\mathrm{dom}(\mathrm{O}) = \{0..8\}$

Effects of calendars with overtime

Task of processing time 4



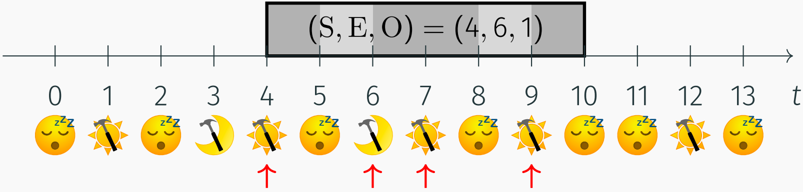
Domain variation

Initial domains :
$\mathrm{dom}(\mathrm{S}) = \{0..8\}$, $\mathrm{dom}(\mathrm{E}) = \{0..8\}$, and $\mathrm{dom}(\mathrm{O}) = \{0..8\}$

Filtered domain :
$\mathrm{dom}(\mathrm{S}) = \{1..8\}$, $\mathrm{dom}(\mathrm{E}) = \{0..8\}$, and $\mathrm{dom}(\mathrm{O}) = \{0..8\}$

Effects of calendars with overtime

Task of processing time 4



Domain variation

Initial domains :

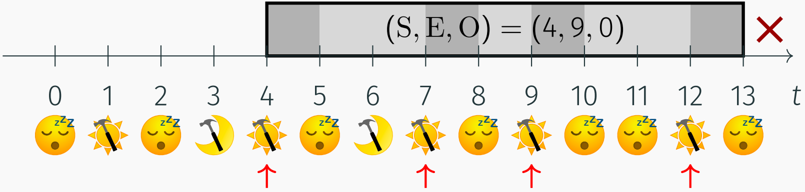
$\mathrm{dom}(S) = \{0..8\}$, $\mathrm{dom}(E) = \{0..8\}$, and $\mathrm{dom}(O) = \{0..8\}$

Filtered domain :

$\mathrm{dom}(S) = \{1..8\}$, $\mathrm{dom}(E) = \{0..8\}$, and $\mathrm{dom}(O) = \{0..8\}$

Effects of calendars with overtime

Task of processing time 4



Domain variation

Initial domains :

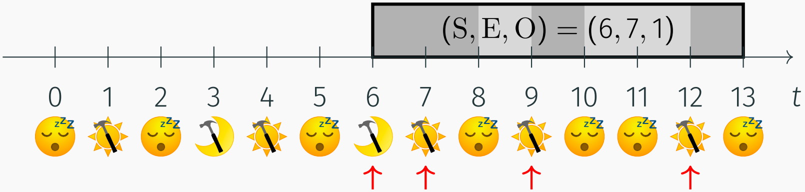
$\mathrm{dom}(\mathrm{S}) = \{0..8\}$, $\mathrm{dom}(\mathrm{E}) = \{0..8\}$, and $\mathrm{dom}(\mathrm{O}) = \{0..8\}$

Filtered domain :

$\mathrm{dom}(\mathrm{S}) = \{1..8\}$, $\mathrm{dom}(\mathrm{E}) = \{0..8\}$, and $\mathrm{dom}(\mathrm{O}) = \{0..8\}$

Effects of calendars with overtime

Task of processing time 4



Domain variation

Initial domains :
$\text{dom}(S) = \{0..8\}$, $\text{dom}(E) = \{0..8\}$, and $\text{dom}(O) = \{0..8\}$

Filtered domain :
$\text{dom}(S) = \{1..6\}$, $\text{dom}(E) = \{0..8\}$, and $\text{dom}(O) = \{0..8\}$

Effects of calendars with overtime

Task of processing time 4

Domain variation

Initial domains :

$\mathrm{dom}(S) = \{0..8\}$, $\mathrm{dom}(E) = \{0..8\}$, and $\mathrm{dom}(O) = \{0..8\}$

Filtered domain :

$\mathrm{dom}(S) = \{1..6\}$, $\mathrm{dom}(E) = \{5..7\}$, and $\mathrm{dom}(O) = \{1..2\}$

First Contribution - The CalendarOvertime constraint

Definition

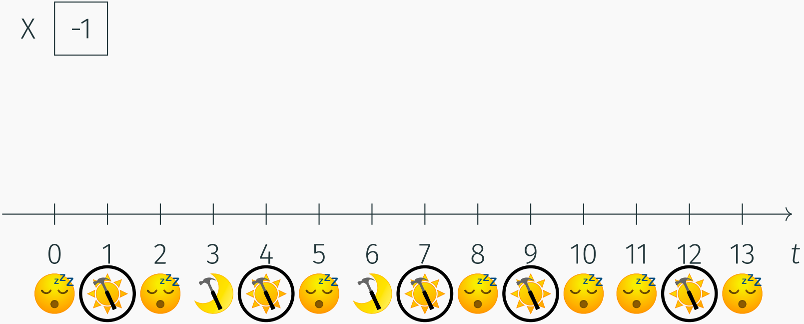
$$\textsc{CalendarOvertime}(\mathrm{S}, \mathrm{E}, \mathrm{O}, \mathrm{p}, \mathrm{Cal}) \overset{def}{\iff} \mathrm{Cal}[\mathrm{S}] \neq \text{😴}$$
$$\wedge\ \mathrm{Cal}[\mathrm{S}+\mathrm{E}-1] \neq \text{😴}$$
$$\wedge\ \mathrm{O} = \mathrm{p} - |\{t \in [\mathrm{S}, \mathrm{S}+\mathrm{E}) \mid \mathrm{Cal}[t] = \text{☀🔨}\}|$$
$$\wedge\ \mathrm{O} \leq |\{t \in [\mathrm{S},\ \mathrm{S}+\mathrm{E}) \mid \mathrm{Cal}[t] = \text{🌙🔨}\}|$$
$$\wedge\ |\{t \in \{\mathrm{S},\ \mathrm{S}+\mathrm{E}-1\} \mid \mathrm{Cal}[t] = \text{🌙🔨}\}| \leq \mathrm{O}$$

Context

- One constraint per task
- Does not deal with resource usages
- Its propagator applies bounds consistency

Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time



Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X	-1	0

Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X	-1	0	0

Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X	-1	0	0	0

Key data-structures for the propagator

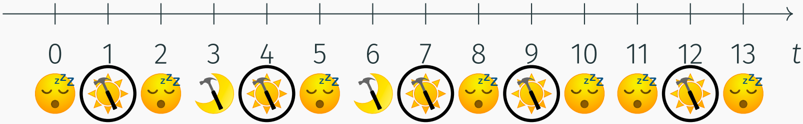
- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X				
-1	0	0	0	1

Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4



Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

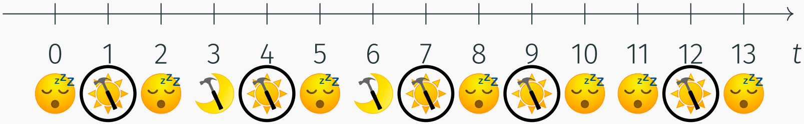
Y	1

Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y	1	4



Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X														
X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y	1	4	7

Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X														
	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y				
	1	4	7	9

Key data-structures for the propagator

- We want to perform fast propagation
- We precompute four arrays that permit performing various computations in constant time

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y	1	4	7	9	12

Counting the regular time points in an interval

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y	1	4	7	9	12

Utility functions

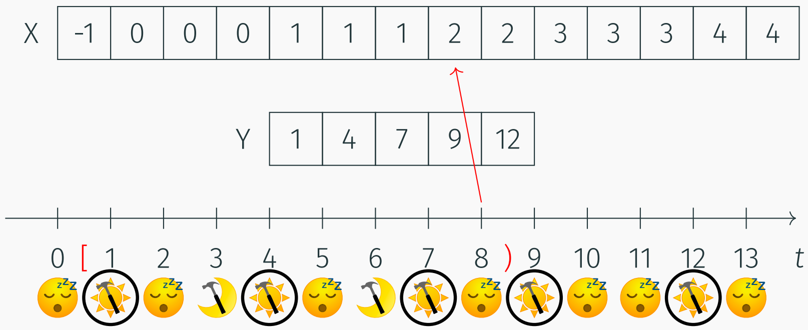
Counting the regular time points in an interval

X														
	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y					
	1	4	7	9	12

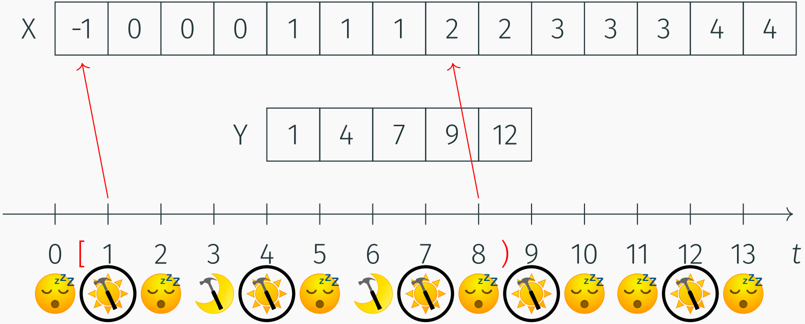
Utility functions

Counting the regular time points in an interval



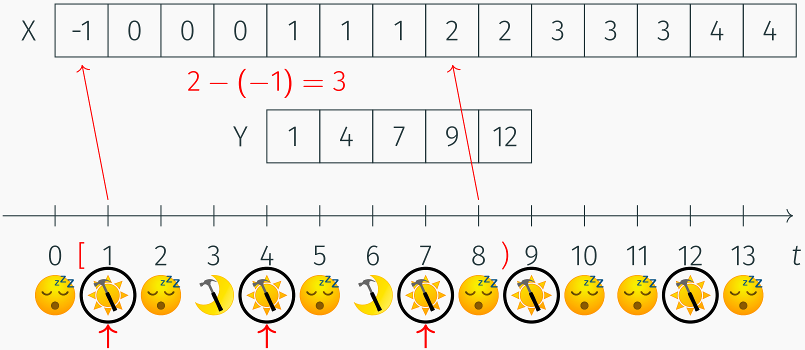
Utility functions

Counting the regular time points in an interval



Utility functions

Counting the regular time points in an interval



Utility functions

Finding the (non-strict) previous regular time point from any point.

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

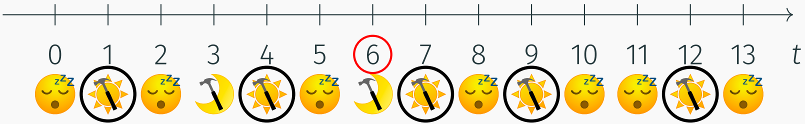
Y	1	4	7	9	12

Utility functions

Finding the (non-strict) previous regular time point from any point.

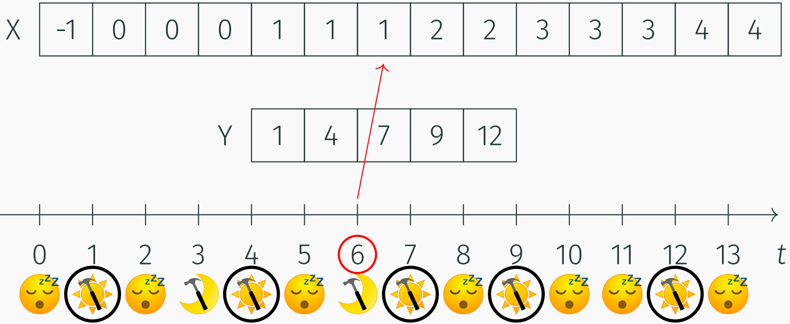
X														
	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y					
	1	4	7	9	12



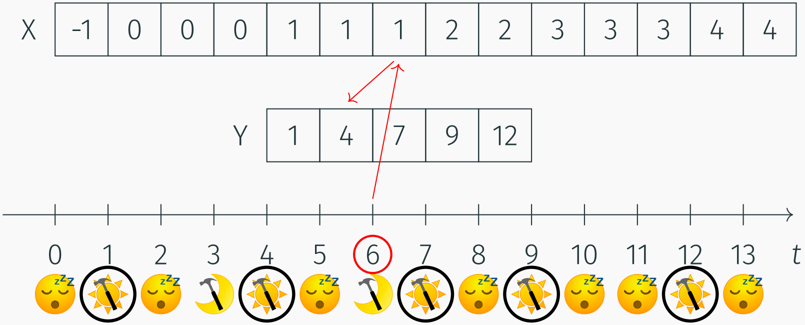
Utility functions

Finding the (non-strict) previous regular time point from any point.



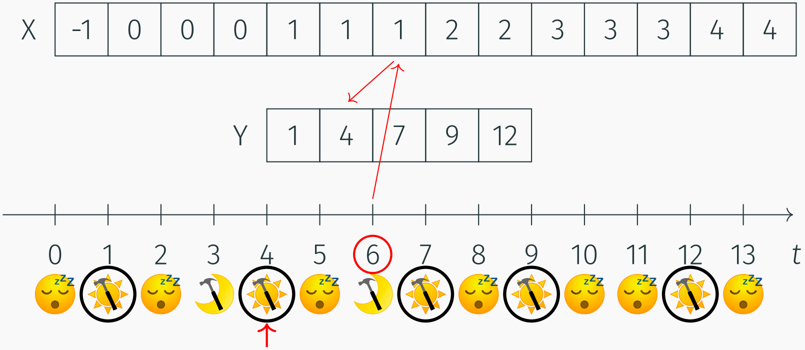
Utility functions

Finding the (non-strict) previous regular time point from any point.



Utility functions

Finding the (non-strict) previous regular time point from any point.



Utility functions

Finding the smallest end for an interval containing a given amount of regular time

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y	1	4	7	9	12

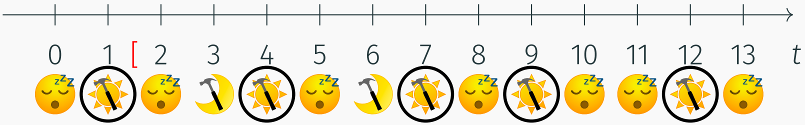
Utility functions

Finding the smallest end for an interval containing a given amount of regular time

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

Y	1	4	7	9	12

$N = 3$



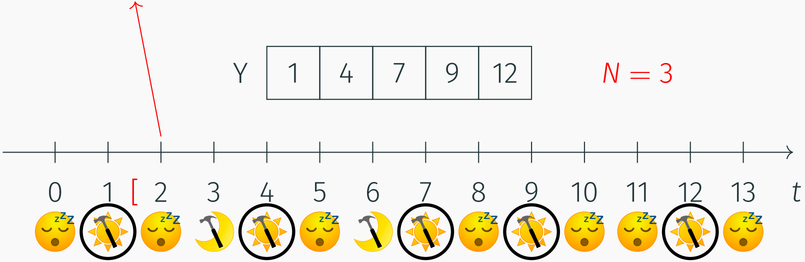
Utility functions

Finding the smallest end for an interval containing a given amount of regular time

X	-1	0	0	0	1	1	1	2	2	3	3	3	4	4

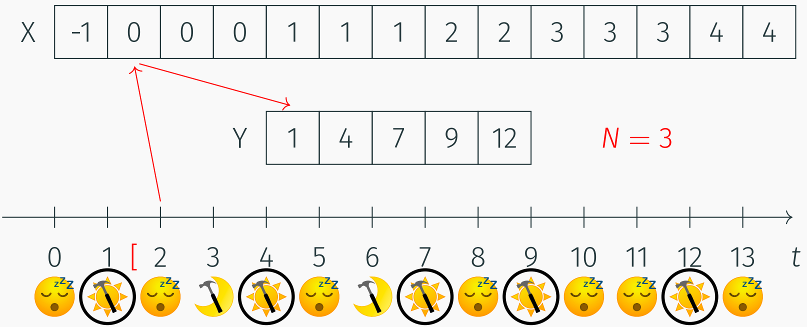
Y	1	4	7	9	12

$N = 3$

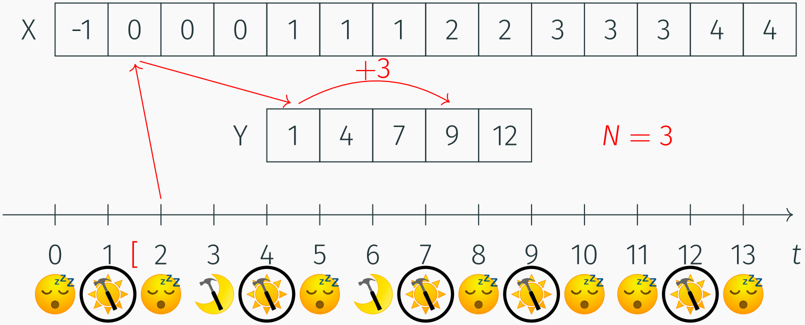


Utility functions

Finding the smallest end for an interval containing a given amount of regular time

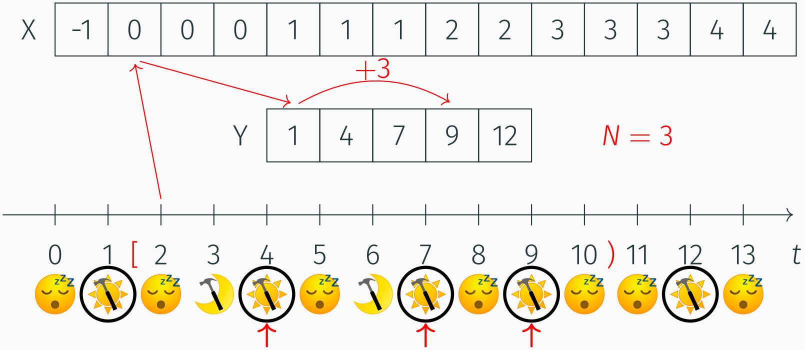


Finding the smallest end for an interval containing a given amount of regular time



Utility functions

Finding the smallest end for an interval containing a given amount of regular time



Propagation trace

Steps to create a minimal execution window for a given S

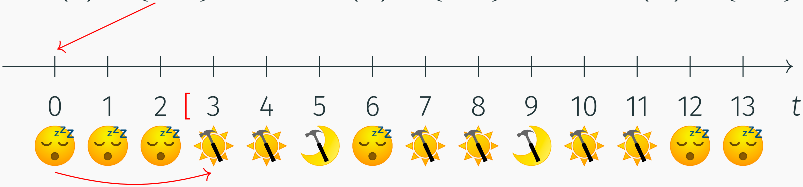
1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

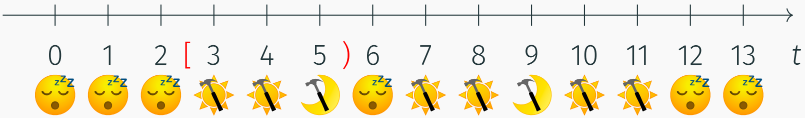


Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$



Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom(S)} = \{0..8\}$ $\mathrm{dom(E)} = \{3..6\}$ $\mathrm{dom(O)} = \{0..1\}$

Ok

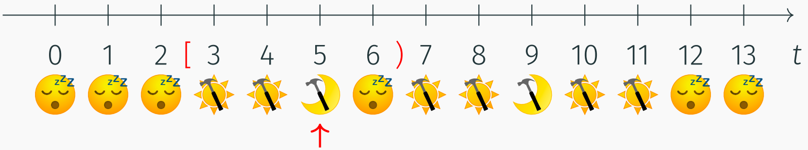
Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

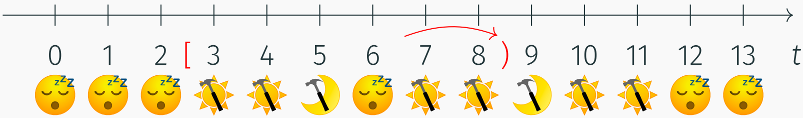
Not Ok



Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$



Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\quad$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\quad$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

Ok

Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

Not Ok

Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

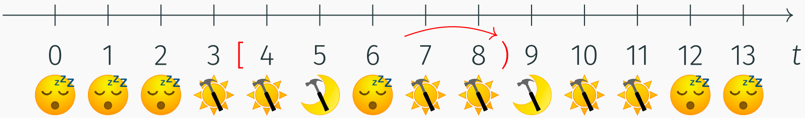
$\mathrm{dom(S)} = \{0..8\}$ $\mathrm{dom(E)} = \{3..6\}$ $\mathrm{dom(O)} = \{0..1\}$

Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$



Propagation trace

> Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

Ok

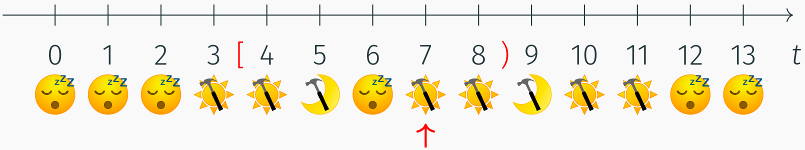
Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom(S)} = \{0..8\}$ $\mathrm{dom(E)} = \{3..6\}$ $\mathrm{dom(O)} = \{0..1\}$

Ok



Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

Ok

Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

Ok

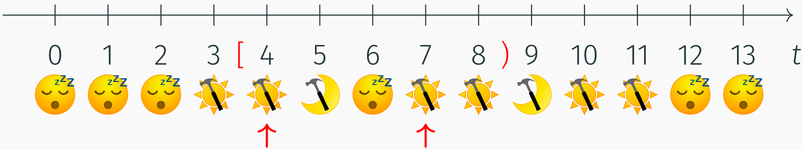
Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{0..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

Ok



Propagation trace

Filtering the lower bound of S, for a task with $\mathrm{p} = 2$

1. Add p non-closed time points
2. Adjust to reach the minimal duration (verify $\underline{\mathrm{E}}$)
3. Add any missing regular time (verify $\overline{\mathrm{O}}$)
4. Correct any potential unworked overtime tail
5. Verify $\overline{\mathrm{E}}$, $\underline{\mathrm{O}}$, and the unworked overtime head/tail

$\mathrm{dom}(\mathrm{S}) = \{4..8\}$ $\mathrm{dom}(\mathrm{E}) = \{3..6\}$ $\mathrm{dom}(\mathrm{O}) = \{0..1\}$

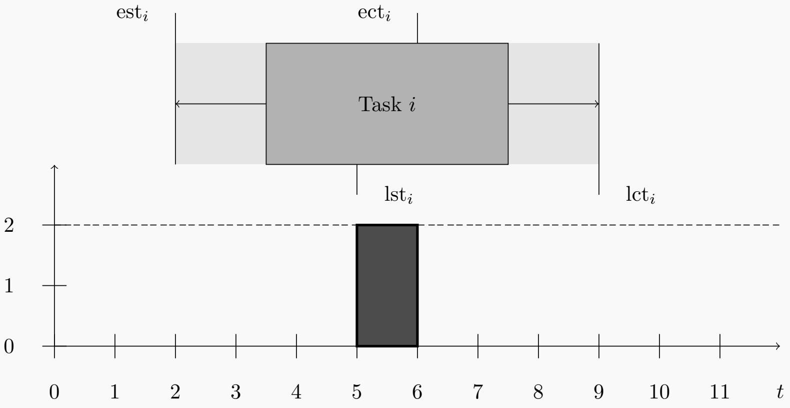
Time-Tabling rule

Notation

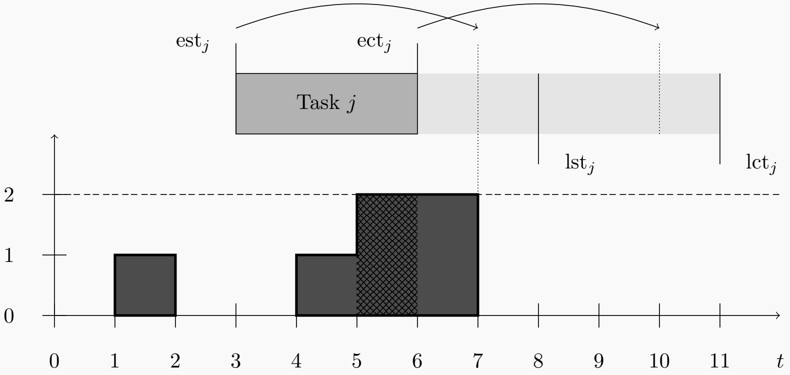
- est : earliest starting time
- lst : latest starting time
- $\mathrm{ect} = \mathrm{est} + \mathrm{p}$: earliest completion time
- $\mathrm{lct} = \mathrm{lst} + \mathrm{p}$: latest completion time

The Time-Tabling rule

The compulsory part is the interval $[\mathrm{lst}, \mathrm{ect})$, if non-empty

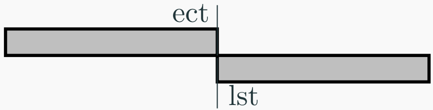


The profile (the aggregation of compulsory parts) helps deduce invalid values



$$p = 4$$

$$\mathrm{dom}(S) = \{1..5\} \qquad \mathrm{dom}(E) = \{4..7\} \qquad \mathrm{dom}(O) = \{0..1\}$$

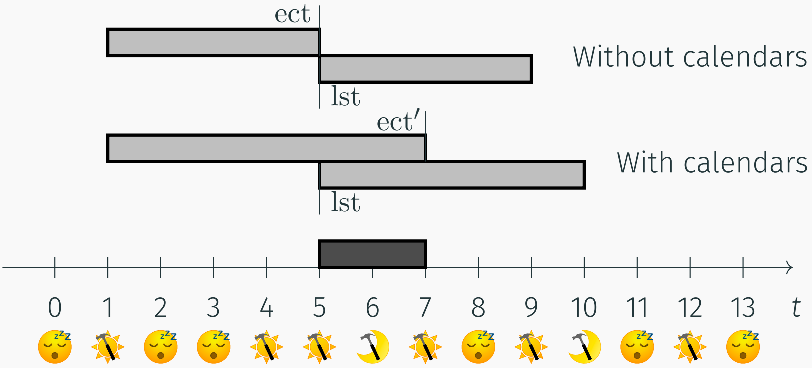


Without calendars

Second Contribution - The CUMULATIVEOVERTIME constraint

$$\mathrm{p} = 4$$

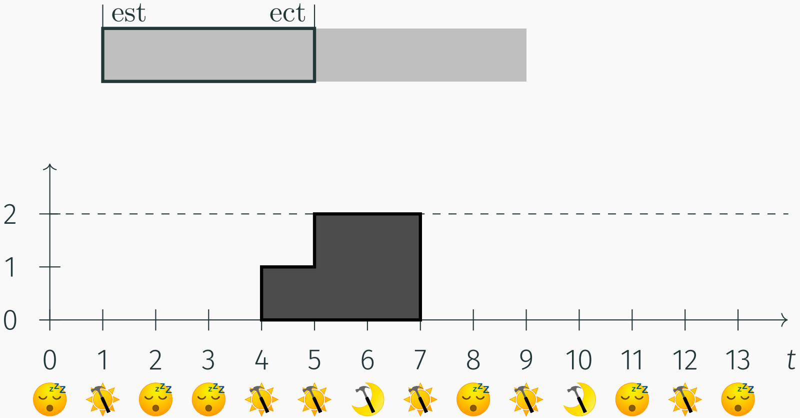
$\mathrm{dom(S)} = \{1..5\}$ $\mathrm{dom(E)} = \{4..7\}$ $\mathrm{dom(O)} = \{0..1\}$



Second Contribution - The CumulativeOvertime constraint

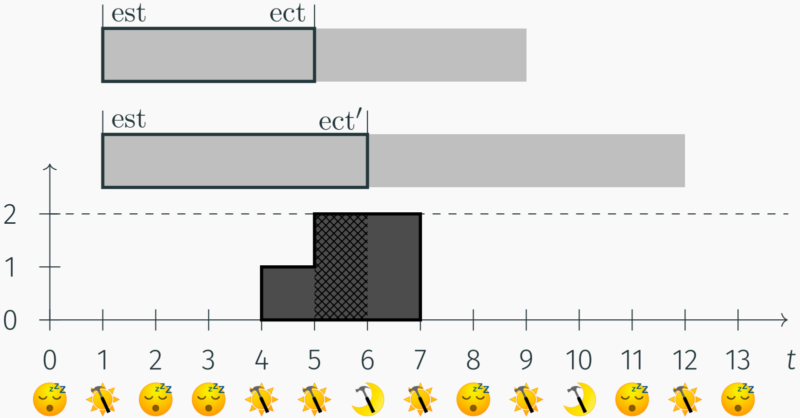
$$p = 3$$

$$\mathrm{dom}(S) = \{1..9\} \qquad \mathrm{dom}(E) = \{4..5\} \qquad \mathrm{dom}(O) = \{0..1\}$$



$$p = 3$$

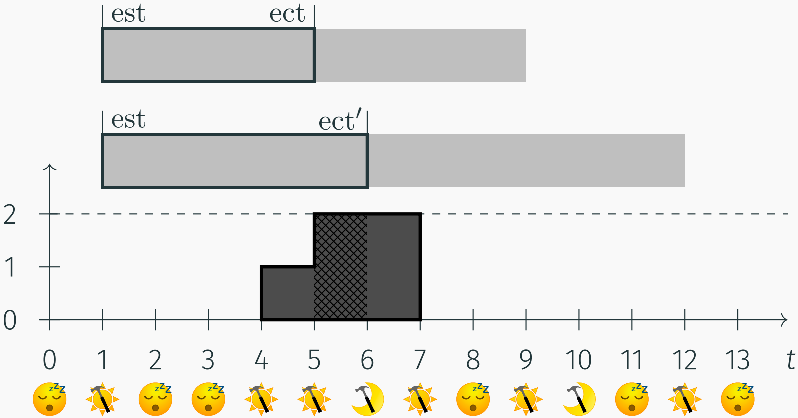
$$\mathrm{dom}(S) = \{1..9\} \qquad \mathrm{dom}(E) = \{4..5\} \qquad \mathrm{dom}(O) = \{0..1\}$$



Second Contribution - The CumulativeOvertime constraint

$$p = 3$$

$\mathrm{dom}(S) = \{7..9\}$ $\mathrm{dom}(E) = \{4..5\}$ $\mathrm{dom}(O) = \{0..1\}$



Experimentation

Problem

We solve the RCPSP augmented with calendars and overtime.
There are resource, release and deadline, precedence, and calendar constraints.
We minimize overtime costs with a restrained horizon.

Instances

We use instances from PSPLIB, BL, and PACK
They are augmented with pseudo-realistic calendars.

Experimentation

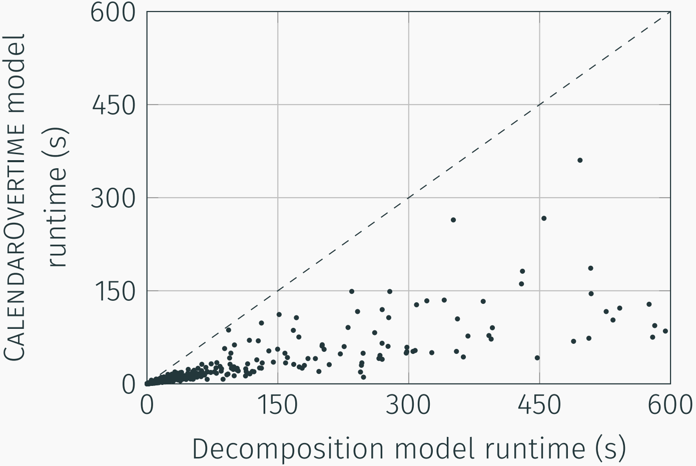
The 3 MiniZinc models

- Decomposition model : Decomposes the calendar constraints into ELEMENT constraints.
- CALENDAROVERTIME model : Uses our CALENDAROVERTIME constraints.
- CUMULATIVEOVERTIME model : Uses both our new constraints.

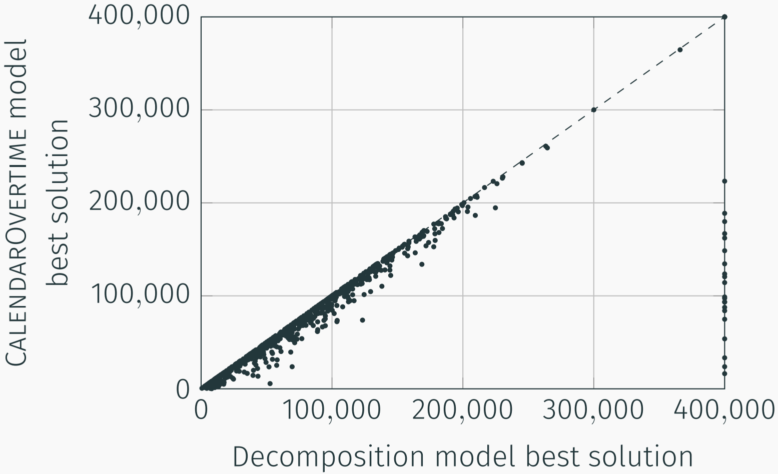
Implementation

Our constraints are implemented in the Chuffed solver (which does lazy clause generation).

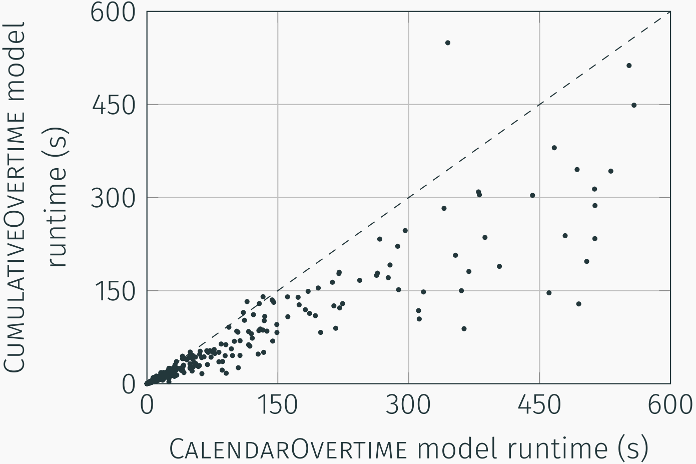
Results - Decomposition vs. CalendarOvertime



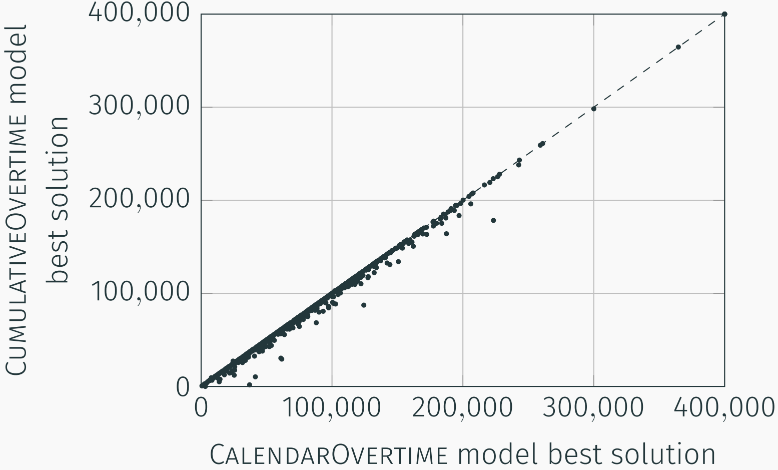
Results - Decomposition vs. CalendarOvertime



Results - CalendarOvertime vs. CumulativeOvertime



Results - CalendarOvertime vs. CumulativeOvertime



Conclusion

Contributions

- The CalendarOvertime constraint (with a propagator that applies bounds consistency of the S, E and O variables in $\mathcal{O}(|\mathrm{dom}(\mathrm{S})|)$).
- The CumulativeOvertime constraint (with a propagator that incorporate calendars in the Time-Tabling rule).

Advantages of the new constraints

- Make modeling with calendars simpler
- Lead to better resolutions

Conclusion

The new constraints could be used in other problems than the RCPSP augmented with calendars.

Thank you for your attention

Link to the code, instances, and models :