

5.87 cumulatives

	DESCRIPTION	LINKS	GRAPH
Origin	[31]		
Constraint	cumulatives(TASKS, MACHINES, CTR)		
Arguments	$\begin{array}{l} \text{TASKS} \quad : \quad \text{collection} \left(\begin{array}{l} \text{machine-dvar,} \\ \text{origin-dvar,} \\ \text{duration-dvar,} \\ \text{end-dvar,} \\ \text{height-dvar} \end{array} \right) \\ \text{MACHINES} \quad : \quad \text{collection}(\text{id-int, capacity-int}) \\ \text{CTR} \quad \quad : \quad \text{atom} \end{array}$		
Restrictions	<pre>required(TASKS, [machine, height]) require_at_least(2, TASKS, [origin, duration, end]) in_attr(TASKS, machine, MACHINES, id) TASKS.duration ≥ 0 TASKS.origin ≤ TASKS.end required(MACHINES, [id, capacity]) distinct(MACHINES, id) CTR ∈ [≤, ≥]</pre>		
Purpose	<p>Consider a set \mathcal{T} of tasks described by the TASKS collection. When CTR is equal to \leq (respectively \geq), the cumulatives constraint enforces the following condition for each machine m: At each point in time, where at least one task assigned on machine m is present, the cumulated height of the set of tasks that both overlap that point and are assigned to machine m should be less than or equal to (respectively greater than or equal to) the capacity associated with machine m. A task overlaps a point i if and only if (1) its origin is less than or equal to i, and (2) its end is strictly greater than i. It also imposes for each task of \mathcal{T} the constraint $\text{origin} + \text{duration} = \text{end}$.</p>		
Example	$\left(\begin{array}{l} \text{machine-1 origin-2 duration-2 end-4 height--2,} \\ \text{machine-1 origin-1 duration-4 end-5 height-1,} \\ \left\langle \begin{array}{l} \text{machine-1 origin-4 duration-2 end-6 height--1,} \\ \text{machine-1 origin-2 duration-3 end-5 height-2,} \\ \text{machine-1 origin-5 duration-2 end-7 height-2,} \\ \text{machine-2 origin-3 duration-2 end-5 height--1,} \\ \text{machine-2 origin-1 duration-4 end-5 height-1} \end{array} \right\rangle, \\ \langle \text{id-1 capacity-0, id-2 capacity-0} \rangle, \geq \end{array} \right)$		

Figure 5.176 shows with a thick line the cumulated profile on the two machines described by the MACHINES collection. Within this profile a task with a positive (respectively negative) height is represented by a pink (respectively blue) rectangle, where the length of the rectangle corresponds to the duration of the task. The cumulatives constraint holds since, both on machines 1 and 2, we have that at each point in time the cumulated

resource consumption is greater than or equal to the limit 0 enforced by the last argument (i.e., the attribute capacity of the items of the MACHINES collection) of the `cumulatives` constraint (i.e., we have a limit of 0 both on machines 1 and 2).

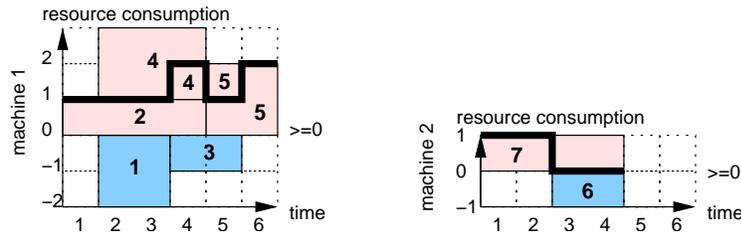


Figure 5.176: Resource consumption profile on the different machines

Typical

```
|TASKS| > 1
range(TASKS.machine) > 1
range(TASKS.origin) > 1
range(TASKS.duration) > 1
range(TASKS.end) > 1
range(TASKS.height) > 1
TASKS.duration > 0
TASKS.height ≠ 0
|MACHINES| > 1
MACHINES.capacity < sum(TASKS.height)
|TASKS| > |MACHINES|
```

Symmetries

- Items of TASKS are [permutable](#).
- Items of MACHINES are [permutable](#).
- All occurrences of two distinct values in TASKS.machine or MACHINES.id can be [swapped](#); all occurrences of a value in TASKS.machine or MACHINES.id can be [renamed](#) to any unused value.

Usage

As shown in the **Example** slot, the `cumulatives` constraint is useful for covering problems where different demand profiles have to be covered by a set of tasks. This is modelled in the following way:

- To each demand profile is associated a given machine m and a set of tasks for which all attributes (`machine`, `origin`, `duration`, `end`, `height`) are fixed; moreover the `machine` attribute is fixed to m and the `height` attribute is strictly negative. For each machine m the cumulated profile of all the previous tasks constitutes the demand profile to cover.
- To each task that can be used to cover the demand is associated a task for which the `height` attribute is a positive integer; the `height` attribute describes the amount of demand that can be covered by the task at each instant during its execution (between its `origin` and its `end`) on the demand profile associated with the `machine` attribute.
- In order to express the fact that each demand profile should completely be covered, we set the `capacity` attribute of each machine to 0. We can also relax the constraint

by setting the `capacity` attribute to a negative number that specifies the maximum allowed uncovered demand at each instant.

The demand profiles might also not be completely fixed in advance.

When all the heights of the tasks are non-negative, one other possible use of the `cumulatives` constraint is to enforce to reach a minimum level of resource consumption. This is imposed on those time points that are overlapped by at least one task.

By introducing a dummy task of height 0, of origin the minimum origin of all the tasks and of end the maximum end of all the tasks, this can also be imposed between the first and the last utilisation of the resource.

Finally the `cumulatives` constraint is also useful for scheduling problems where several *cumulative* machines are available and where you have to assign each task on a specific machine.

Algorithm

Three filtering algorithms for this constraint are described in [31].

Systems

`cumulatives` in **Gecode**, `cumulatives` in **SICStus**.

See also

assignment dimension removed: `cumulative` (*negative heights not allowed*).

common keyword: `calendar` (*scheduling constraint*),
`coloured_cumulatives` (*resource constraint*).

generalisation: `diffn` (*task with machine assignment and origin attributes replaced by orthotope*).

used in graph description: `sum_ctr`.

Keywords

application area: workload covering.

characteristic of a constraint: derived collection.

complexity: sequencing with release times and deadlines.

constraint type: scheduling constraint, resource constraint, temporal constraint, timetabling constraint.

filtering: compulsory part, sweep.

modelling: assignment dimension, assignment to the same set of values, scheduling with machine choice, calendars and preemption, zero-duration task.

modelling exercises: assignment to the same set of values, scheduling with machine choice, calendars and preemption.

problems: producer-consumer, demand profile.

Derived Collection

$$\text{col} \left(\begin{array}{c} \text{TIME_POINTS} - \text{collection}(\text{idm} - \text{int}, \text{duration} - \text{dvar}, \text{point} - \text{dvar}), \\ \left[\begin{array}{c} \text{item} \left(\begin{array}{c} \text{idm} - \text{TASKS.machine}, \\ \text{duration} - \text{TASKS.duration}, \\ \text{point} - \text{TASKS.origin} \end{array} \right), \\ \text{item} \left(\begin{array}{c} \text{idm} - \text{TASKS.machine}, \\ \text{duration} - \text{TASKS.duration}, \\ \text{point} - \text{TASKS.end} \end{array} \right) \end{array} \right] \end{array} \right)$$

Arc input(s)	TASKS
Arc generator	$\text{SELF} \mapsto \text{collection}(\text{tasks})$
Arc arity	1
Arc constraint(s)	$\text{tasks.origin} + \text{tasks.duration} = \text{tasks.end}$
Graph property(ies)	$\text{NARC} = \text{TASKS} $

For all items of MACHINES:

Arc input(s)	TIME_POINTS TASKS
Arc generator	$\text{PRODUCT} \mapsto \text{collection}(\text{time_points}, \text{tasks})$
Arc arity	2
Arc constraint(s)	<ul style="list-style-type: none"> • $\text{time_points.idm} = \text{MACHINES.id}$ • $\text{time_points.idm} = \text{tasks.machine}$ • $\text{time_points.duration} > 0$ • $\text{tasks.origin} \leq \text{time_points.point}$ • $\text{time_points.point} < \text{tasks.end}$
Graph class	<ul style="list-style-type: none"> • ACYCLIC • BIPARTITE • NO_LOOP
Sets	$\text{SUCC} \mapsto$ $\left[\begin{array}{c} \text{source}, \\ \text{variables} - \text{col} \left(\begin{array}{c} \text{VARIABLES} - \text{collection}(\text{var} - \text{dvar}), \\ [\text{item}(\text{var} - \text{TASKS.height})] \end{array} \right) \end{array} \right]$
Constraint(s) on sets	$\text{sum_ctr}(\text{variables}, \text{CTR}, \text{MACHINES.capacity})$

Graph model

Within the context of the second graph constraint, part (A) of Figure 5.177 shows the initial graphs associated with machines 1 and 2 of the **Example** slot. Part (B) of Figure 5.177 shows the corresponding final graphs associated with machines 1 and 2. On the one hand, each source vertex of the final graph can be interpreted as a time point p on a specific machine m . On the other hand the successors of a source vertex correspond to those tasks that both overlap that time point p and are assigned to machine m . Since they do not have any successors we have eliminated those vertices corresponding to the end of the last three tasks of the TASKS collection. The *cumulatives* constraint holds since for each successor set \mathcal{S} of the final graph the sum of the height of the tasks in \mathcal{S} is greater than or equal to the capacity of the machine corresponding to the time point associated with \mathcal{S} .

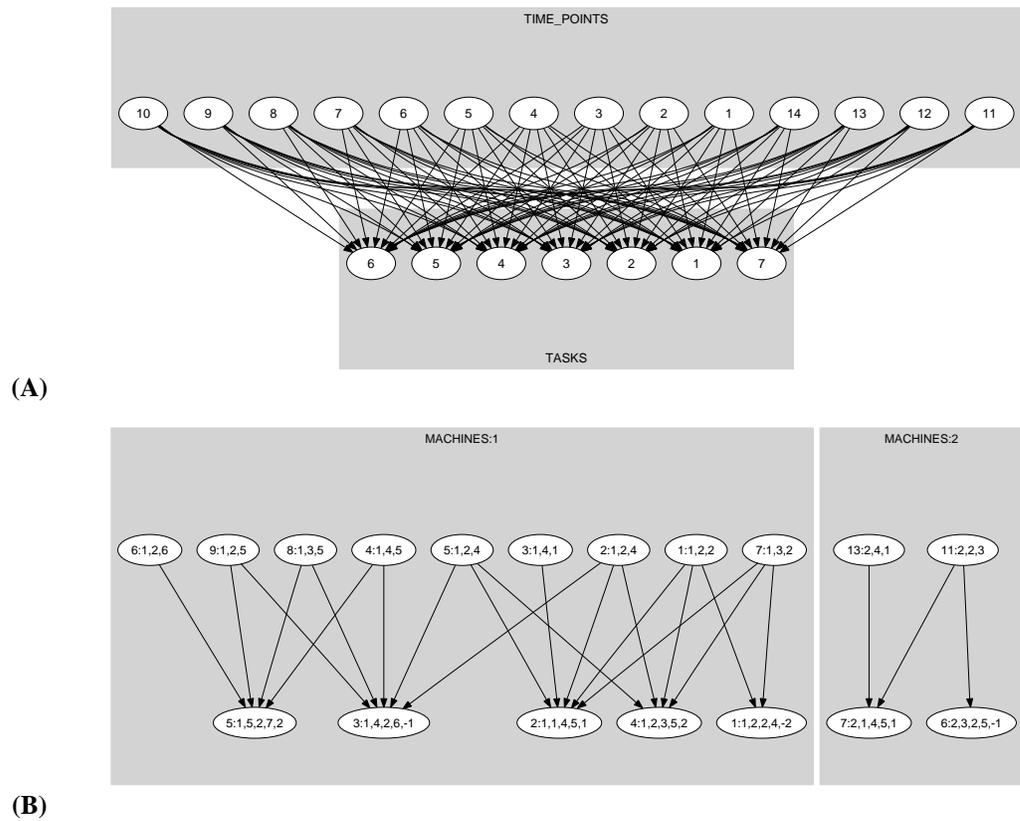


Figure 5.177: Initial and final graph of the cumulatives constraint

Signature

Since $NARC$ is the maximum number of vertices of the final graph of the first graph constraint we can rewrite $NARC = |TASKS|$ to $NARC \geq |TASKS|$. This leads to simplify $NARC$ to $NARC$.