5.353 sliding_time_window_from_start

DESCRIPTION LINKS GRAPH

Origin

Used for defining `sliding_time_window`.

Constraint

$\texttt{sliding\_time\_window\_from\_start}(\texttt{WINDOW\_SIZE}, \texttt{LIMIT}, \texttt{TASKS}, \texttt{START})$

Arguments

WINDOW_SIZE	:	int
LIMIT	:	int
TASKS	:	collection(origin−dvar, duration−dvar)
START	:	dvar

Restrictions

$\texttt{WINDOW\_SIZE} > 0$
$\texttt{LIMIT} \geq 0$
$\texttt{required}(\texttt{TASKS}, [\texttt{origin}, \texttt{duration}])$
$\texttt{TASKS.duration} \geq 0$

Purpose

The sum of the intersections of all the tasks of the TASKS collection with interval $[\texttt{START}, \texttt{START} + \texttt{WINDOW\_SIZE} - 1]$ is less than or equal to LIMIT.

Example

$$\left(9, 6, \left\langle \begin{array}{ll} \texttt{origin}-10 & \texttt{duration}-3, \\ \texttt{origin}-5 & \texttt{duration}-1, \\ \texttt{origin}-6 & \texttt{duration}-2 \end{array} \right\rangle, 5\right)$$

The intersections of tasks $\langle \texttt{id}-1\ \texttt{origin}-10\ \texttt{duration}-3\rangle$, $\langle \texttt{id}-2\ \texttt{origin}-5\ \texttt{duration}-1\rangle$, and $\langle \texttt{id}-3\ \texttt{origin}-6\ \texttt{duration}-2\rangle$ with interval $[\texttt{START}, \texttt{START}+\texttt{WINDOW\_SIZE}-1] = [5, 5+9-1] = [5, 13]$ are respectively equal to 3, 1, and 2 (i.e., the three tasks of the TASKS collection are in fact included within interval $[5, 13]$). Consequently, the `sliding_time_window_from_start` constraint holds since the sum $3+1+2$ of these intersections does not exceed the value of its second argument $\texttt{LIMIT} = 6$.

Typical

$\texttt{WINDOW\_SIZE} > 1$
$\texttt{LIMIT} > 0$
$\texttt{LIMIT} < \texttt{WINDOW\_SIZE}$
$|\texttt{TASKS}| > 1$
$\texttt{TASKS.duration} > 0$

Symmetries

- WINDOW_SIZE can be decreased.
- LIMIT can be increased.
- Items of TASKS are permutable.
- TASKS.duration can be decreased to any value ≥ 0.
- One and the same constant can be added to START as well as to the origin attribute of all items of TASKS.

Arg. properties

Contractible wrt. TASKS.

Reformulation

Similar to the reformulation of `sliding_time_window`.

Used in

`sliding_time_window`.

Keywords

characteristic of a constraint: derived collection.

constraint type: sliding sequence constraint, temporal constraint.

Derived Collection $\texttt{col}(\texttt{S}-\texttt{collection}(\texttt{var}-\texttt{dvar}),[\texttt{item}(\texttt{var}-\texttt{START})])$

Arc input(s) S TASKS

Arc generator $PRODUCT \mapsto \texttt{collection}(\texttt{s},\texttt{tasks})$

Arc arity 2

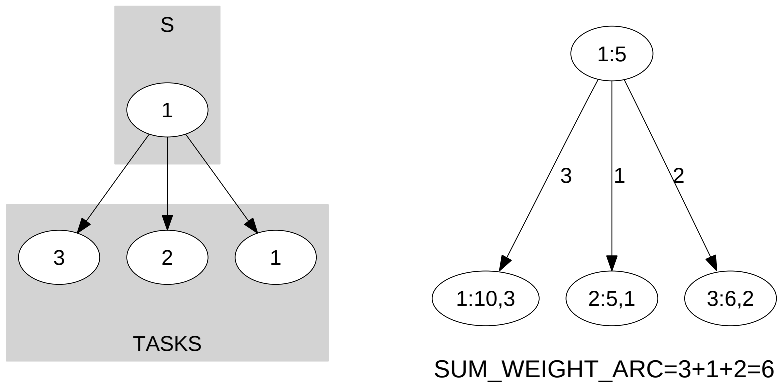
Arc constraint(s) TRUE

Graph property(ies)

$$\mathbf{SUM\_WEIGHT\_ARC}\left(\texttt{max}\left(0,\ \begin{array}{l}\texttt{min}\left(\begin{array}{l}\texttt{s.var}+\texttt{WINDOW\_SIZE},\\ \texttt{tasks.origin}+\texttt{tasks.duration}\end{array}\right)-\\ \texttt{max}(\texttt{s.var},\texttt{tasks.origin})\end{array}\right)\right)\leq \texttt{LIMIT}$$

Graph model Since we use the TRUE arc constraint the final and the initial graph are identical. The unique source of the final graph corresponds to the interval $[\texttt{START},\texttt{START}+\texttt{WINDOW\_SIZE}-1]$. Each sink of the final graph represents a given task of the TASKS collection. We associate to each arc the value given by the intersection of the task associated with one of the extremities of the arc with the time window $[\texttt{START},\texttt{START}+\texttt{WINDOW\_SIZE}-1]$. Finally, the graph property **SUM_WEIGHT_ARC** sums up all the valuations of the arcs and check that it does not exceed a given limit.

Parts (A) and (B) of Figure 5.690 respectively show the initial and final graph associated with the **Example** slot. To each arc of the final graph we associate the intersection of the corresponding sink task with interval $[\texttt{START},\texttt{START}+\texttt{WINDOW\_SIZE}-1]$. The constraint sliding_time_window_from_start holds since the sum of the previous intersections does not exceed LIMIT.



(A) **(B)**

Figure 5.690: Initial and final graph of the sliding_time_window_from_start constraint