

**5.296 sliding\_time\_window\_from\_start**

	DESCRIPTION	LINKS	GRAPH
<b>Origin</b>	Used for defining <code>sliding_time_window</code> .		
<b>Constraint</b>	<code>sliding_time_window_from_start(WINDOW_SIZE, LIMIT, TASKS, START)</code>		
<b>Arguments</b>	<pre>WINDOW_SIZE : int LIMIT       : int TASKS       : collection(origin-dvar, duration-dvar) START       : dvar</pre>		
<b>Restrictions</b>	<pre>WINDOW_SIZE &gt; 0 LIMIT ≥ 0 required(TASKS, [origin, duration]) TASKS.duration ≥ 0</pre>		
<b>Purpose</b>	The sum of the intersections of all the tasks of the TASKS collection with interval $[START, START + WINDOW\_SIZE - 1]$ is less than or equal to LIMIT.		
<b>Example</b>	$\left( 9, 6, \left\langle \begin{array}{ll} \text{origin} - 10 & \text{duration} - 3, \\ \text{origin} - 5 & \text{duration} - 1, \\ \text{origin} - 6 & \text{duration} - 2 \end{array} \right\rangle, 5 \right)$ <p>The intersections of tasks <math>\langle \text{id} - 1 \text{ origin} - 10 \text{ duration} - 3 \rangle</math>, <math>\langle \text{id} - 2 \text{ origin} - 5 \text{ duration} - 1 \rangle</math>, and <math>\langle \text{id} - 3 \text{ origin} - 6 \text{ duration} - 2 \rangle</math> with interval <math>[START, START + WINDOW\_SIZE - 1] = [5, 5 + 9 - 1] = [5, 13]</math> are respectively equal to 3, 1, and 2 (i.e., the three tasks of the TASKS collection are in fact included within interval <math>[5, 13]</math>). Consequently, the <code>sliding_time_window_from_start</code> constraint holds since the sum <math>3 + 1 + 2</math> of these intersections does not exceed the value of its second argument <math>LIMIT = 6</math>.</p>		
<b>Symmetries</b>	<ul style="list-style-type: none"> <li>• WINDOW_SIZE can be <b>decreased</b>.</li> <li>• LIMIT can be <b>increased</b>.</li> <li>• Items of TASKS are <b>permutable</b>.</li> <li>• TASKS.duration can be <b>decreased</b> to any value <math>\geq 0</math>.</li> <li>• One and the same constant can be <b>added</b> to START as well as to the origin attribute of all items of TASKS.</li> </ul>		
<b>Reformulation</b>	Similar to the reformulation of <code>sliding_time_window</code> .		
<b>Used in</b>	<code>sliding_time_window</code> .		
<b>Keywords</b>	<b>characteristic of a constraint:</b> derived collection. <b>constraint type:</b> sliding sequence constraint, temporal constraint.		

**Derived Collection**

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

**Arc input(s)**

S TASKS

**Arc generator**

$\text{PRODUCT} \mapsto \text{collection}(s, \text{tasks})$

**Arc arity**

2

**Arc constraint(s)**

TRUE

**Graph property(ies)**

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

**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  $[\text{START}, \text{START} + \text{WINDOW\_SIZE} - 1]$ . 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  $[\text{START}, \text{START} + \text{WINDOW\_SIZE} - 1]$ . 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 5.533 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  $[\text{START}, \text{START} + \text{WINDOW\_SIZE} - 1]$ . The constraint `sliding_time_window_from_start` holds since the sum of the previous intersections does not exceed LIMIT.

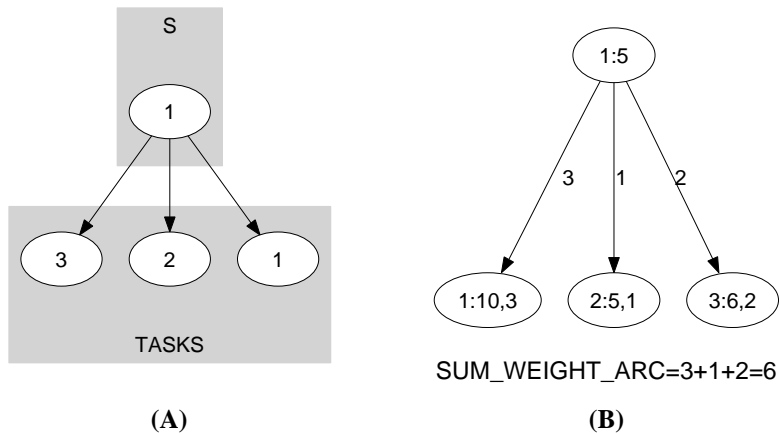


Figure 5.533: Initial and final graph of the `sliding_time_window_from_start` constraint