

### 5.309 `soft_used_by_interval_var`

	DESCRIPTION	LINKS	GRAPH
<b>Origin</b>	Derived from <code>used_by_interval</code> .		
<b>Constraint</b>	<code>soft_used_by_interval_var(C, VARIABLES1, VARIABLES2, SIZE_INTERVAL)</code>		
<b>Synonym</b>	<code>soft_used_by_interval</code> .		
<b>Arguments</b>	<p>C : <code>dvar</code></p> <p>VARIABLES1 : <code>collection(var-dvar)</code></p> <p>VARIABLES2 : <code>collection(var-dvar)</code></p> <p>SIZE_INTERVAL : <code>int</code></p>		
<b>Restrictions</b>	<p><math>C \geq 0</math></p> <p><math>C \leq  \text{VARIABLES2} </math></p> <p><math> \text{VARIABLES1}  \geq  \text{VARIABLES2} </math></p> <p><code>required(VARIABLES1, var)</code></p> <p><code>required(VARIABLES2, var)</code></p> <p><math>\text{SIZE\_INTERVAL} &gt; 0</math></p>		
<b>Purpose</b>	<p>Let <math>N_i</math> (respectively <math>M_i</math>) denote the number of variables of the collection <code>VARIABLES1</code> (respectively <code>VARIABLES2</code>) that take a value in the interval <math>[\text{SIZE\_INTERVAL} \cdot i, \text{SIZE\_INTERVAL} \cdot i + \text{SIZE\_INTERVAL} - 1]</math>. <math>C</math> is the minimum number of values to change in the <code>VARIABLES1</code> and <code>VARIABLES2</code> collections so that for all integer <math>i</math> we have <math>M_i &gt; 0 \Rightarrow N_i &gt; 0</math>.</p>		
<b>Example</b>	<div style="border: 1px solid blue; padding: 5px; display: inline-block;"> <math display="block">\left( \begin{array}{l} 2, \langle 9, 1, 1, 8, 8 \rangle, \\ \langle 9, 9, 9, 1 \rangle, 3 \end{array} \right)</math> </div> <p>In the example, the fourth argument <code>SIZE_INTERVAL = 3</code> defines the following family of intervals <math>[3 \cdot k, 3 \cdot k + 2]</math>, where <math>k</math> is an integer. Consequently the values of the collections <code><math>\langle 9, 1, 1, 8, 8 \rangle</math></code> and <code><math>\langle 9, 9, 9, 1 \rangle</math></code> are respectively located within intervals <math>[9, 11]</math>, <math>[0, 2]</math>, <math>[0, 2]</math>, <math>[6, 8]</math>, <math>[6, 8]</math> and intervals <math>[9, 11]</math>, <math>[9, 11]</math>, <math>[9, 11]</math>, <math>[0, 2]</math>. Since there is a correspondence between two pairs of intervals we must unset at least <math>4 - 2</math> items (4 is the number of items of the <code>VARIABLES2</code> collection). Consequently, the <code>soft_used_by_interval_var</code> constraint holds since its first argument <code>C</code> is set to <math>4 - 2</math>.</p>		
<b>Symmetries</b>	<ul style="list-style-type: none"> <li>• Items of <code>VARIABLES1</code> are <a href="#">permutable</a>.</li> <li>• Items of <code>VARIABLES2</code> are <a href="#">permutable</a>.</li> <li>• An occurrence of a value of <code>VARIABLES1.var</code> that belongs to the <math>k</math>-th interval, of size <code>SIZE_INTERVAL</code>, can be <a href="#">replaced</a> by any other value of the same interval.</li> <li>• An occurrence of a value of <code>VARIABLES2.var</code> that belongs to the <math>k</math>-th interval, of size <code>SIZE_INTERVAL</code>, can be <a href="#">replaced</a> by any other value of the same interval.</li> </ul>		
<b>Usage</b>	A soft <code>used_by_interval</code> constraint.		

**See also**

**hard version:** [used\\_by\\_interval](#).

**implied by:** [soft\\_same\\_interval\\_var](#).

**Keywords**

**constraint arguments:** constraint between two collections of variables.

**constraint type:** soft constraint, relaxation, variable-based violation measure.

**modelling:** interval.

<b>Arc input(s)</b>	VARIABLES1 VARIABLES2
<b>Arc generator</b>	<i>PRODUCT</i> $\mapsto$ <code>collection(variables1, variables2)</code>
<b>Arc arity</b>	2
<b>Arc constraint(s)</b>	$\text{variables1.var}/\text{SIZE\_INTERVAL} = \text{variables2.var}/\text{SIZE\_INTERVAL}$
<b>Graph property(ies)</b>	<u>NSINK_NSOURCE</u> = $ \text{VARIABLES2}  - C$

**Graph model**

Parts (A) and (B) of Figure 5.552 respectively show the initial and final graph associated with the **Example** slot. Since we use the NSINK\_NSOURCE graph property, the source and sink vertices of the final graph are stressed with a double circle. The `soft_used_by_interval_var` constraint holds since the cost 2 corresponds to the difference between the number of variables of VARIABLES2 and the sum over the different connected components of the minimum number of sources and sinks.

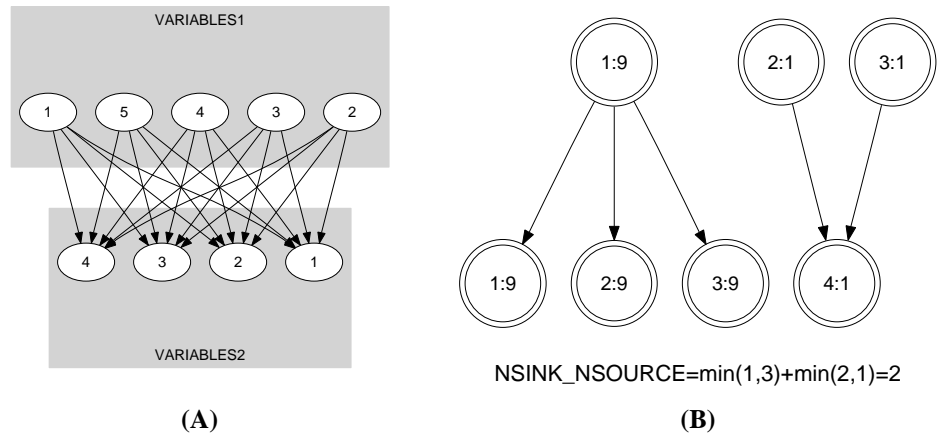


Figure 5.552: Initial and final graph of the `soft_used_by_interval_var` constraint

20050507

1591