

5.366 soft_used_by_interval_var

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from <code>used_by_interval</code> .		
Constraint	<code>soft_used_by_interval_var(C, VARIABLES1, VARIABLES2, SIZE_INTERVAL)</code>		
Synonym	<code>soft_used_by_interval</code> .		
Arguments	<pre> C : dvar VARIABLES1 : collection(var-dvar) VARIABLES2 : collection(var-dvar) SIZE_INTERVAL : int </pre>		
Restrictions	<pre> C ≥ 0 C ≤ VARIABLES2 VARIABLES1 ≥ VARIABLES2 required(VARIABLES1, var) required(VARIABLES2, var) SIZE_INTERVAL > 0 </pre>		
Purpose	<p>Let N_i (respectively M_i) denote the number of variables of the collection <code>VARIABLES1</code> (respectively <code>VARIABLES2</code>) that take a value in the interval $[SIZE_INTERVAL \cdot i, SIZE_INTERVAL \cdot i + SIZE_INTERVAL - 1]$. <code>C</code> is the minimum number of values to change in the <code>VARIABLES1</code> and <code>VARIABLES2</code> collections so that for all integer i we have $M_i > 0 \Rightarrow N_i \geq M_i$.</p>		
Example	<pre>(2, (9, 1, 1, 8, 8), (9, 9, 9, 1), 3)</pre> <p>In the example, the fourth argument <code>SIZE_INTERVAL = 3</code> defines the following family of intervals $[3 \cdot k, 3 \cdot k + 2]$, where k is an integer. Consequently the values of the collections <code>(9, 1, 1, 8, 8)</code> and <code>(9, 9, 9, 1)</code> are respectively located within intervals $[9, 11]$, $[0, 2]$, $[0, 2]$, $[6, 8]$, $[6, 8]$ and intervals $[9, 11]$, $[9, 11]$, $[9, 11]$, $[0, 2]$. Since there is a correspondence between two pairs of intervals we must unset at least $4 - 2$ 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 $4 - 2$.</p>		
Typical	<pre> C > 0 VARIABLES1 > 1 VARIABLES2 > 1 range(VARIABLES1.var) > 1 range(VARIABLES2.var) > 1 SIZE_INTERVAL > 1 SIZE_INTERVAL < range(VARIABLES1.var) SIZE_INTERVAL < range(VARIABLES2.var) </pre>		

Symmetries

- Items of VARIABLES1 are [permutable](#).
- Items of VARIABLES2 are [permutable](#).
- An occurrence of a value of VARIABLES1.var that belongs to the k -th interval, of size SIZE_INTERVAL, can be [replaced](#) by any other value of the same interval.
- An occurrence of a value of VARIABLES2.var that belongs to the k -th interval, of size SIZE_INTERVAL, can be [replaced](#) by any other value of the same interval.

Usage

A soft [used_by_interval](#) 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](#).

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

Graph model

Parts (A) and (B) of Figure 5.709 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.

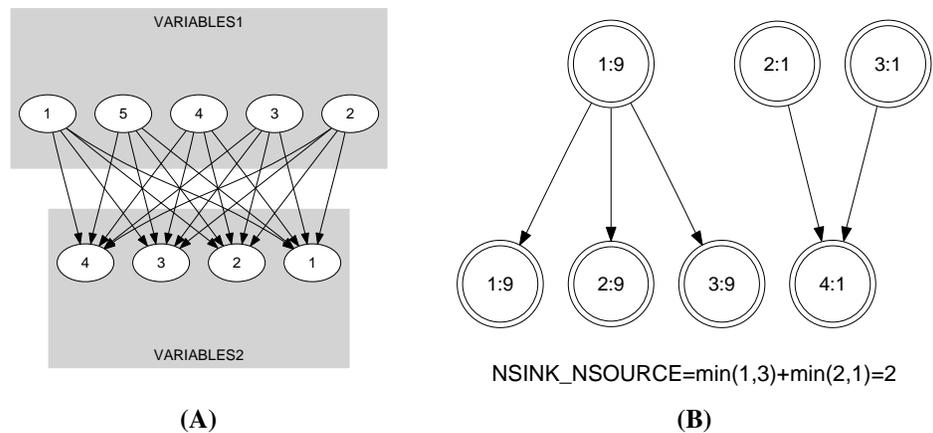


Figure 5.709: Initial and final graph of the *soft_used_by_interval_var* constraint

20050507

2167