

5.280 same_and_global_cardinality_low_up

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from same and global_cardinality_low_up		
Constraint	<code>same_and_global_cardinality_low_up(VARIABLES1, VARIABLES2, VALUES)</code>		
Arguments	VARIABLES1 : <code>collection</code> (var-dvar) VARIABLES2 : <code>collection</code> (var-dvar) VALUES : <code>collection</code> (val-int, omin-int, omax-int)		
Restrictions	$ VARIABLES1 = VARIABLES2 $ <code>required</code> (VARIABLES1, var) <code>required</code> (VARIABLES2, var) <code>required</code> (VALUES, [val, omin, omax]) <code>distinct</code> (VALUES, val) $VALUES.omin \geq 0$ $VALUES.omax \leq VARIABLES1 $ $VALUES.omin \leq VALUES.omax$		
Purpose	<div style="border: 1px solid pink; padding: 5px;"> The variables of the VARIABLES2 collection correspond to the variables of the VARIABLES1 collection according to a permutation. In addition, each value $VALUES[i].val$ ($1 \leq i \leq VALUES$) should be taken by at least $VALUES[i].omin$ and at most $VALUES[i].omax$ variables of the VARIABLES1 collection. </div>		

Example

$$\left(\begin{array}{l}
 \text{var - 1,} \\
 \text{var - 9,} \\
 \left\langle \begin{array}{l} \text{var - 1,} \\ \text{var - 5,} \end{array} \right\rangle, \\
 \text{var - 2,} \\
 \text{var - 1} \\
 \text{var - 9,} \\
 \text{var - 1,} \\
 \left\langle \begin{array}{l} \text{var - 1,} \\ \text{var - 1,} \end{array} \right\rangle, \\
 \text{var - 2,} \\
 \text{var - 5} \\
 \text{val - 1} \quad \text{omin - 2} \quad \text{omax - 3,} \\
 \left\langle \begin{array}{l} \text{val - 2} \quad \text{omin - 1} \quad \text{omax - 1,} \\ \text{val - 5} \quad \text{omin - 1} \quad \text{omax - 1,} \\ \text{val - 7} \quad \text{omin - 0} \quad \text{omax - 2,} \end{array} \right\rangle \\
 \text{val - 9} \quad \text{omin - 1} \quad \text{omax - 1}
 \end{array} \right)$$

The `same_and_global_cardinality_low_up` constraint holds since:

- The values 1, 9, 1, 5, 2, 1 assigned to $|VARIABLES1|$ correspond to a permutation of the values 9, 1, 1, 1, 2, 5 assigned to $|VARIABLES2|$.

- The values 1, 2, 5, 7 and 6 are respectively used 3 ($2 \leq 3 \leq 3$), 1 ($1 \leq 1 \leq 1$), 1 ($1 \leq 1 \leq 1$), 0 ($0 \leq 0 \leq 2$) and 1 ($1 \leq 1 \leq 1$) times.

Symmetries

- Arguments are [permutable](#) w.r.t. permutation (VARIABLES1, VARIABLES2).
- Items of VARIABLES1 are [permutable](#).
- Items of VARIABLES2 are [permutable](#).
- An occurrence of a value of VARIABLES1.var or VARIABLES2.var that does not belong to VALUES.val can be [replaced](#) by any other value that also does not belong to VALUES.val.
- Items of VALUES are [permutable](#).
- VALUES.omin can be [decreased](#) to any value ≥ 0 .
- VALUES.omax can be [increased](#) to any value $\leq |\text{VARIABLES1}|$.
- All occurrences of two distinct values in VARIABLES1.var, VARIABLES2.var or VALUES.val can be [swapped](#); all occurrences of a value in VARIABLES1.var, VARIABLES2.var or VALUES.val can be [renamed](#) to any unused value.

Usage

The `same_and_global_cardinality_low_up` constraint can be used for modelling the following [assignment](#) problem with one single constraint. The organisation Doctors Without Borders has a list of doctors and a list of nurses, each of whom volunteered to go on one rescue mission. Each volunteer specifies a list of possible dates and each mission should include one doctor and one nurse. In addition we have for each date the minimum and maximum number of missions that should be effectively done. The task is to produce a list of pairs such that each pair includes a doctor and a nurse who are available on the same date and each volunteer appears in exactly one pair so that for each day we build the required number of missions.

Algorithm

In [46], the [flow](#) network that was used to model the [same](#) constraint [43, 44] is extended to support the cardinalities. Then, algorithms are developed to compute [arc-consistency](#) and [bound-consistency](#).

See also

generalisation: `same_and_global_cardinality` (fixed interval replaced by variable).

implies: `global_cardinality_low_up`, `global_cardinality_low_up_no_loop`, `same`.

Keywords

application area: assignment.

combinatorial object: permutation, multiset.

constraint arguments: constraint between two collections of variables.

constraint type: value constraint.

filtering: bound-consistency, arc-consistency, flow.

modelling: equality between multisets.

problems: demand profile.

Arc input(s)	VARIABLES1 VARIABLES2
Arc generator	$\text{PRODUCT} \mapsto \text{collection}(\text{variables1}, \text{variables2})$
Arc arity	2
Arc constraint(s)	$\text{variables1.var} = \text{variables2.var}$
Graph property(ies)	<ul style="list-style-type: none"> • for all connected components: $\overline{\text{NSOURCE}} = \overline{\text{NSINK}}$ • $\overline{\text{NSOURCE}} = \text{VARIABLES1}$ • $\overline{\text{NSINK}} = \text{VARIABLES2}$

For all items of VALUES:

Arc input(s)	VARIABLES1
Arc generator	$\text{SELF} \mapsto \text{collection}(\text{variables})$
Arc arity	1
Arc constraint(s)	$\text{variables.var} = \text{VALUES.val}$
Graph property(ies)	<ul style="list-style-type: none"> • $\overline{\text{NVERTEX}} \geq \text{VALUES.omin}$ • $\overline{\text{NVERTEX}} \leq \text{VALUES.omax}$

Graph model

Parts (A) and (B) of Figure 5.512 respectively show the initial and final graph associated with the first graph constraint of the **Example** slot. Since we use the $\overline{\text{NSOURCE}}$ and $\overline{\text{NSINK}}$ graph properties, the source and sink vertices of the final graph are stressed with a double circle. Since there is a constraint on each connected component of the final graph we also show the different connected components. Each of them corresponds to an equivalence class according to the arc constraint.

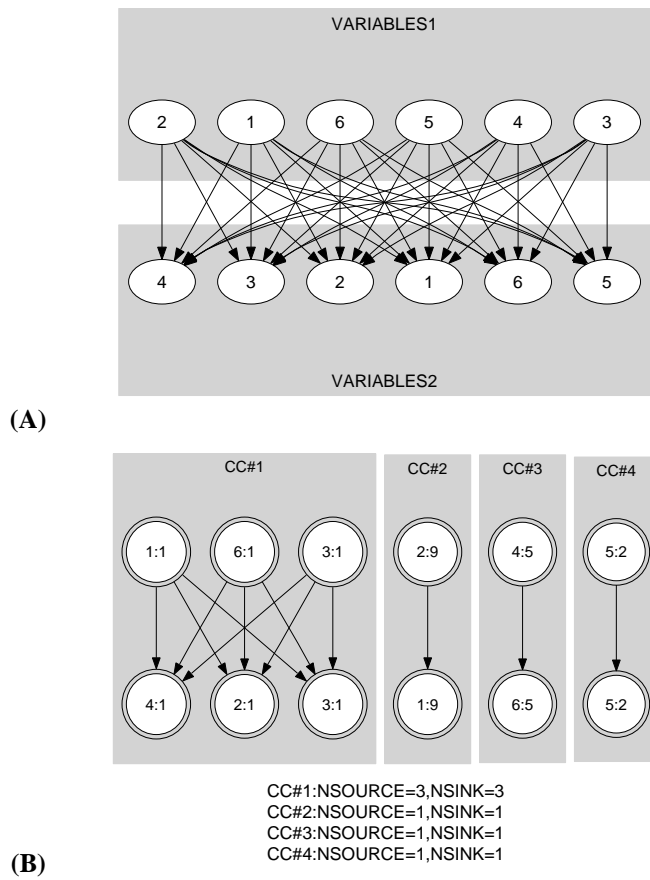


Figure 5.512: Initial and final graph of the same_and_global_cardinality_low_up constraint