## 5.303 soft_alldifferent_var

DESCRIPTION LINKS GRAPH

**Origin** [281]

**Constraint** soft_alldifferent_var(C, VARIABLES)

**Synonyms** soft_alldiff_var, soft_alldistinct_var, soft_alldiff_min_var, soft_alldifferent_min_var, soft_alldistinct_min_var.

**Arguments**

| | | |
|---|---|---|
| C | : | dvar |
| VARIABLES | : | collection(var−dvar) |

**Restrictions**

$C \geq 0$
required(VARIABLES, var)

**Purpose** C is greater than or equal to the minimum number of variables of the collection VARIABLES for which the value needs to be changed in order that all variables of VARIABLES take a distinct value.

**Example**

$$\left(\begin{array}{c} \mathtt{var}-5,\\ \mathtt{var}-1,\\ 3,\left\langle \begin{array}{c} \mathtt{var}-9,\\ \mathtt{var}-1,\\ \mathtt{var}-5, \end{array} \right\rangle \\ \mathtt{var}-5 \end{array}\right)$$

Within the collection $\langle 5,1,9,1,5,5\rangle$, 3 and 2 items are respectively fixed to values 5 and 1. Therefore one must change the values of at least $(3-1)+(2-1)=3$ items to get back to 6 distinct values. Consequently, the soft_alldifferent_var constraint holds since its first argument C is greater than or equal to 3.

**Symmetries**

- Items of VARIABLES are permutable.
- All occurrences of two distinct values of VARIABLES.var can be swapped; all occurrences of a value of VARIABLES.var can be renamed to any unused value.

**Usage** A soft alldifferent constraint.

**Remark** Since it focus on the soft aspect of the alldifferent constraint, the original article [281], which introduce this constraint, describes how to evaluate the minimum value of C and how to prune according to the maximum value of C.

The soft_alldifferent_var constraint is called soft_alldiff_min_var in [132].

**Algorithm** The filtering algorithm presented in [281] achieves arc-consistency.

**Reformulation** By introducing a variable $M$ that gives the number of distinct values used by variables of the collection VARIABLES, the soft_alldifferent_var(C, VARIABLES) constraint can be expressed as a conjunction of the nvalue($M$, VARIABLES) constraint and of the linear constraint C $\geq$ |VARIABLES| $- M$.

**See also** **common keyword:** soft_all_equal_max_var, soft_all_equal_min_ctr, soft_all_equal_min_var, soft_alldifferent_ctr, weighted_partial_alldiff *(soft constraint)*.

**hard version:** alldifferent.

**related:** atmost_nvalue, nvalue.

**Keywords** **characteristic of a constraint:** all different, disequality.

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

**final graph structure:** strongly connected component, equivalence.

| | |
|---|---|
| **Arc input(s)** | VARIABLES |
| **Arc generator** | $CLIQUE \mapsto$ collection(variables1, variables2) |
| **Arc arity** | 2 |
| **Arc constraint(s)** | variables1.var = variables2.var |
| **Graph property(ies)** | NSCC $\geq$ \|VARIABLES\| − C |

**Graph model** We generate a clique with binary *equalities* constraints between each pairs of vertices (this include an arc between a vertex and itself) and we state that C is equal to the difference between the total number of variables and the number of strongly connected components.

Parts (A) and (B) of Figure 5.545 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NSCC** graph property we show the different strongly connected components of the final graph. Each strongly connected component of the final graph includes all variables that take the same value. Since we have 6 variables and 3 strongly connected components the *cost* variable C is greater than or equal to $6 - 3$.

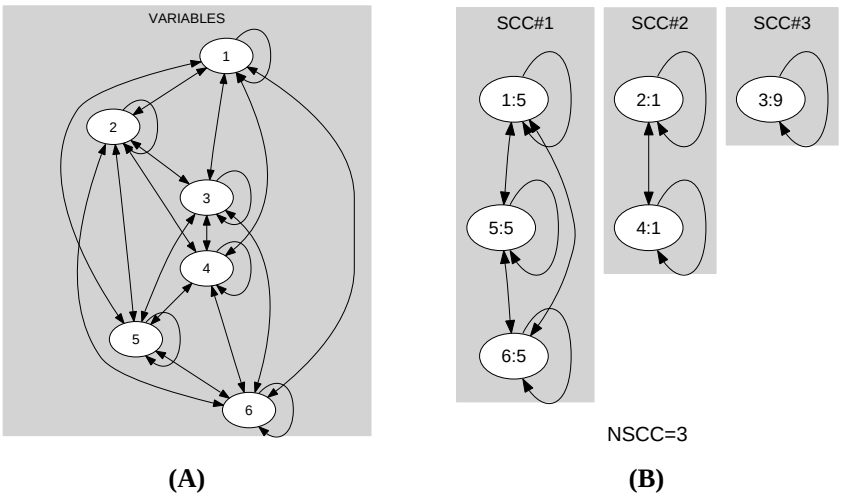


Figure 5.545: Initial and final graph of the soft_alldifferent_var constraint