

5.20 alldifferent_partition

	DESCRIPTION	LINKS	GRAPH
Origin	Derived from alldifferent .		
Constraint	<code>alldifferent_partition(VARIABLES, PARTITIONS)</code>		
Synonyms	<code>alldiff_partition</code> , <code>alldistinct_partition</code> .		
Type	VALUES : <code>collection(val-int)</code>		
Arguments	VARIABLES : <code>collection(var-dvar)</code> PARTITIONS : <code>collection(p - VALUES)</code>		
Restrictions	$ VALUES \geq 1$ required (VALUES, val) distinct (VALUES, val) $ VARIABLES \leq PARTITIONS $ required (VARIABLES, var) $ PARTITIONS \geq 2$ required (PARTITIONS, p)		
Purpose	Enforce all variables of the collection VARIABLES to take values that belong to distinct partitions.		
Example	$((\langle 6, 3, 4 \rangle, \langle p - \langle 1, 3 \rangle, p - \langle 4 \rangle, p - \langle 2, 6 \rangle \rangle))$		
	Since all variables take values that are located within distinct partitions the <code>alldifferent_partition</code> constraint holds.		
Typical	$ VARIABLES > 2$		
Symmetries	<ul style="list-style-type: none"> Items of VARIABLES are permutable. Items of PARTITIONS are permutable. Items of PARTITIONS.p are permutable. A value of VARIABLES.var can be renamed to any value that belongs to the same partition of PARTITIONS. Two distinct values of VARIABLES.var that do not belong to the same partition of PARTITIONS can be swapped. 		
Arg. properties	Contractible wrt. VARIABLES.		

See also

common keyword: `in_same_partition` (*partition*).

specialisation: `alldifferent` (`variable` \in `partition` *replaced by* `variable`).

used in graph description: `in_same_partition`.

Keywords

characteristic of a constraint: `partition`, all different, sort based reformulation.

constraint type: value constraint.

filtering: arc-consistency.

final graph structure: `one_succ`.

modelling: incompatible pairs of values.

Arc input(s)	VARIABLES
Arc generator	<i>CLIQUE</i> \mapsto <code>collection(variables1, variables2)</code>
Arc arity	2
Arc constraint(s)	<code>in_same_partition(variables1.var, variables2.var, PARTITIONS)</code>
Graph property(ies)	<code>MAX_NSCC</code> ≤ 1
Graph class	<code>ONE_SUCC</code>

Graph model

Similar to the `alldifferent` constraint, but we replace the binary *equality* constraint of the `alldifferent` constraint by the fact that two variables are respectively assigned to two values that belong to the same partition. We generate a *clique* with a `in_same_partition` constraint between each pair of vertices (including a vertex and itself) and state that the size of the largest strongly connected component should not exceed 1.

Parts (A) and (B) of Figure 5.48 respectively show the initial and final graph associated with the **Example** slot. Since we use the `MAX_NSCC` graph property we show one of the largest strongly connected components of the final graph.

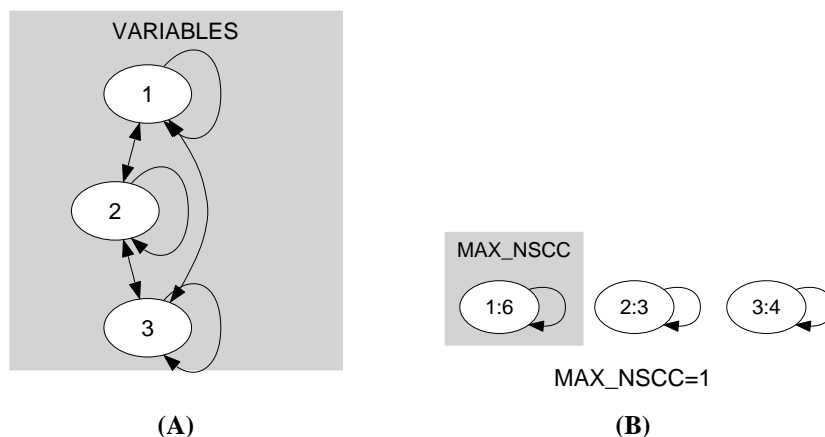


Figure 5.48: Initial and final graph of the `alldifferent_partition` constraint

