## 5.13 alldifferent_partition

**DESCRIPTION** **LINKS** **GRAPH**

**Origin** Derived from `alldifferent`.

**Constraint** `alldifferent_partition(VARIABLES, PARTITIONS)`

**Synonyms** `alldiff_partition`, `alldistinct_partition`.

**Type** `VALUES : collection(val-int)`

**Arguments**

```
VARIABLES  : collection(var-dvar)
PARTITIONS : collection(p - VALUES)
```

**Restrictions**

$|\texttt{VALUES}| \geq 1$
`required(VALUES, val)`
`distinct(VALUES, val)`
$|\texttt{VARIABLES}| \leq |\texttt{PARTITIONS}|$
`required(VARIABLES, var)`
$|\texttt{PARTITIONS}| \geq 2$
`required(PARTITIONS, p)`

**Purpose** Enforce all variables of the collection `VARIABLES` to take values that belong to distinct partitions.

**Example**

$$\left(\begin{array}{l}\langle 6, 3, 4\rangle, \\ \left\langle\begin{array}{l}\texttt{p} - \langle 1, 3\rangle, \\ \texttt{p} - \langle 4\rangle, \\ \texttt{p} - \langle 2, 6\rangle\end{array}\right\rangle\end{array}\right)$$

Since all variables take values that are located within distinct partitions the `alldifferent_partition` constraint holds.

**Typical** $|\texttt{VARIABLES}| > 2$

**Symmetries**

- 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.

**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.

**constraint type:** value constraint.

**filtering:** arc-consistency.

**final graph structure:** one_succ.

**modelling:** incompatible pairs of values.

| | |
|---|---|
| **Arc input(s)** | VARIABLES |
| **Arc generator** | *CLIQUE*$\mapsto$collection(variables1, variables2) |
| **Arc arity** | 2 |
| **Arc constraint(s)** | in_same_partition(variables1.var, variables2.var, PARTITIONS) |
| **Graph property(ies)** | **MAX_NSCC** $\leq 1$ |
| **Graph class** | ONE_SUCC |

**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 5.18 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 component of the final graph.

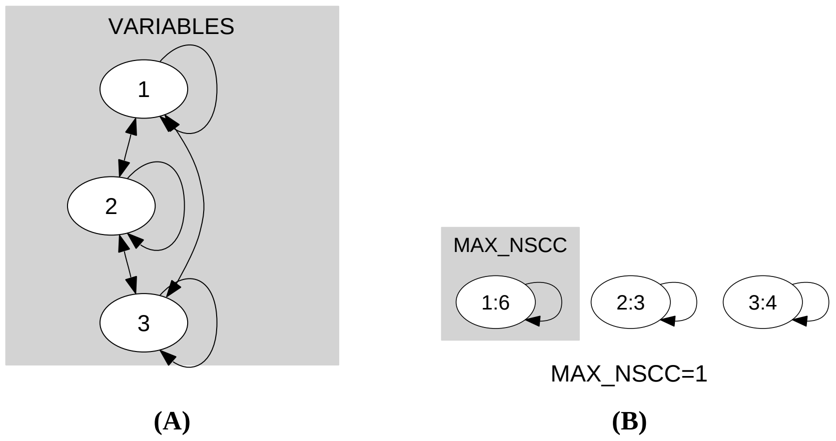

Figure 5.18: Initial and final graph of the alldifferent_partition constraint