5.330 proper_forest

DESCRIPTION LINKS GRAPH

Origin Derived from tree, [46].

Constraint proper_forest(NTREES, NODES)

Arguments

NTREES	:	dvar
NODES	:	collection(index−int, neighbour−svar)

Restrictions

NTREES ≥ 0
required(NODES, [index, neighbour])
|NODES| mod $2 = 0$
NODES.index ≥ 1
NODES.index $\leq$ |NODES|
distinct(NODES, index)
NODES.neighbour ≥ 1
NODES.neighbour $\leq$ |NODES|
NODES.neighbour $\neq$ NODES.index

Purpose Cover an undirected graph G by a set of NTREES trees (i.e., a *tree* is a connected graph without cycles that contains at least two vertices [105]) in such a way that each vertex of G belongs to one distinct tree.

Example

$$\left(\begin{array}{cc} & \\ & \\ & \\ & \\ 3, \left\langle \begin{array}{ll} \texttt{index} - 1 & \texttt{neighbour} - \{3,6\}, \\ \texttt{index} - 2 & \texttt{neighbour} - \{9\}, \\ \texttt{index} - 3 & \texttt{neighbour} - \{1,5,7\}, \\ \texttt{index} - 4 & \texttt{neighbour} - \{9\}, \\ \texttt{index} - 5 & \texttt{neighbour} - \{3\}, \\ \texttt{index} - 6 & \texttt{neighbour} - \{1\}, \\ \texttt{index} - 7 & \texttt{neighbour} - \{3\}, \\ \texttt{index} - 8 & \texttt{neighbour} - \{10\}, \\ \texttt{index} - 9 & \texttt{neighbour} - \{2,4\}, \\ \texttt{index} - 10 & \texttt{neighbour} - \{8\} \end{array} \right\rangle & \\ & \\ & \\ & \\ & \end{array}\right)$$

The proper_forest constraint holds since the undirected graph associated with the items of the NODES collection corresponds to a forest containing NTREES $= 3$ trees: each tree respectively involves the vertices $\{1, 3, 5, 6, 7\}$, $\{2, 4, 9\}$ and $\{8, 10\}$.

Typical

NTREES > 0
|NODES| > 1

Symmetry Items of NODES are permutable.

Arg. properties

Functional dependency: NTREES determined by NODES.

Algorithm A filtering algorithm for the `proper_forest` constraint was proposed by N. Beldiceanu *et al.* in [46]. It achieves hybrid-consistency and its running time is dominated by the complexity of finding all edges that do not belong to any maximum cardinality matching in an undirected n-vertex, m-edge graph, i.e., $O(m \cdot n)$.

Systems `tree` in **Choco**.

See also **common keyword:** `tree` (*connected component*,*tree*).

Keywords **characteristic of a constraint:** undirected graph.

constraint arguments: constraint involving set variables.

constraint type: graph constraint.

filtering: hybrid-consistency.

final graph structure: connected component, tree, no cycle, symmetric.

modelling: functional dependency.

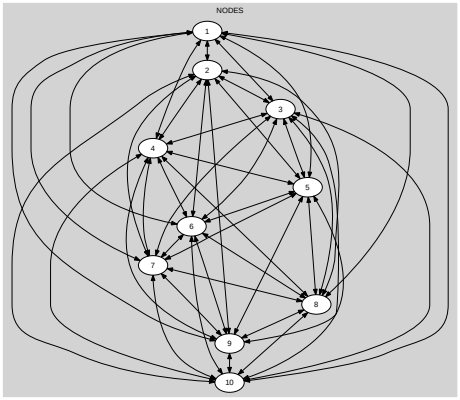
Arc input(s)	NODES
Arc generator	$CLIQUE(\neq) \mapsto$ collection(nodes1, nodes2)
Arc arity	2
Arc constraint(s)	in_set(nodes2.index, nodes1.neighbour)
Graph property(ies)	• $\mathbf{NVERTEX} = (\mathbf{NARC} + 2 * \text{NTREES})/2$ • $\mathbf{NCC} = \text{NTREES}$ • $\mathbf{NVERTEX} = \lvert\text{NODES}\rvert$
Graph class	SYMMETRIC

Graph model The graph constraint forces the following conditions:

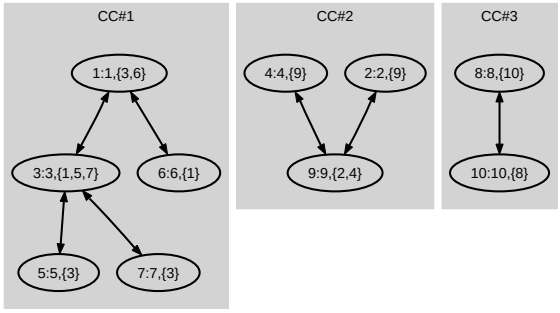
- Each connected component of the final graph has n vertices and $2 \cdot (n-1)$ arcs. This is equivalent to the fact that each connected component has not any cycle.
- Since we use the $CLIQUE(\neq)$ arc-generator and since, by definition, the final graph does not contain any isolated vertex, each connected component of the final graph involves more than one vertex.
- The number of connected components of the final graph is equal to $\mathbf{NCC}$.
- All the vertices of the initial graph belong to the final graph.
- The final graph is symmetric.

Parts (A) and (B) of Figure 5.677 respectively show the initial and final graph associated with the **Example** slot. For each connected component we display its number of arcs as well as its number of vertices. The proper_forest constraint holds since the final graph has $\text{NTREES} = \mathbf{NCC} = 3$ connected components and no cycle.

(A)



(B)



CC#1:NARC=8,NVERTEX=5
CC#2:NARC=4,NVERTEX=3
CC#3:NARC=2,NVERTEX=2
NCC=3

Figure 5.677: Initial and final graph of the `proper_forest` constraint