5.401 tour

DESCRIPTION **LINKS** **GRAPH**

Origin [5]

Constraint `tour(NODES)`

Synonyms `atour`, `cycle`.

Argument `NODES : collection(index-int, succ-svar)`

Restrictions
$|\texttt{NODES}| \geq 3$
`required(NODES, [index, succ])`
$\texttt{NODES.index} \geq 1$
$\texttt{NODES.index} \leq |\texttt{NODES}|$
`distinct(NODES, index)`

Purpose Enforce to cover an undirected graph G described by the `NODES` collection with a Hamiltonian cycle.

Example

$$\left(\left\langle\begin{array}{ll}\texttt{index}-1 & \texttt{succ}-\{2,4\},\\ \texttt{index}-2 & \texttt{succ}-\{1,3\},\\ \texttt{index}-3 & \texttt{succ}-\{2,4\},\\ \texttt{index}-4 & \texttt{succ}-\{1,3\}\end{array}\right\rangle\right)$$

The `tour` constraint holds since its `NODES` argument depicts the following Hamiltonian cycle visiting successively the vertices 1, 2, 3 and 4.

Symmetry Items of `NODES` are permutable.

Algorithm When the number of vertices is odd (i.e., $|\texttt{NODES}|$ is odd) a necessary condition is that the graph is not bipartite. Other necessary conditions for filtering the `tour` constraint are given in [131, 130].

See also **common keyword:** `circuit` *(graph partitioning constraint, Hamiltonian)*, `cycle` *(graph constraint)*, `link_set_to_booleans` *(constraint involving set variables)*.

used in graph description: `in_set`.

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

combinatorial object: matching.

constraint arguments: constraint involving set variables.

constraint type: graph constraint.

filtering: DFS-bottleneck, linear programming.

problems: Hamiltonian.

Arc input(s)	NODES
Arc generator	$CLIQUE(\neq) \mapsto$ collection(nodes1, nodes2)
Arc arity	2
Arc constraint(s)	in_set(nodes2.index, nodes1.succ) $\Leftrightarrow$ in_set(nodes1.index, nodes2.succ)
Graph property(ies)	**NARC**$= \vert\texttt{NODES}\vert * \vert\texttt{NODES}\vert - \vert\texttt{NODES}\vert$

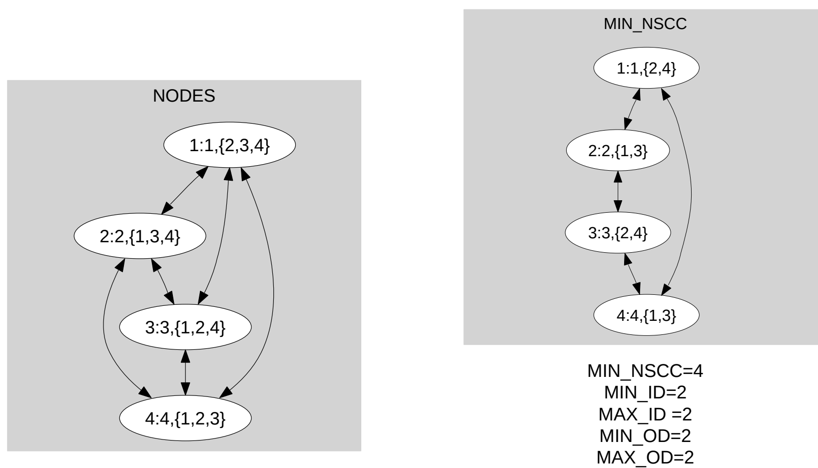
Arc input(s)	NODES
Arc generator	$CLIQUE(\neq) \mapsto$ collection(nodes1, nodes2)
Arc arity	2
Arc constraint(s)	in_set(nodes2.index, nodes1.succ)
Graph property(ies)	• **MIN_NSCC**$= \vert\texttt{NODES}\vert$ • **MIN_ID**$= 2$ • **MAX_ID**$= 2$ • **MIN_OD**$= 2$ • **MAX_OD**$= 2$

Graph model The first graph property enforces the subsequent condition: If we have an arc from the i^{th} vertex to the j^{th} vertex then we have also an arc from the j^{th} vertex to the i^{th} vertex. The second graph property enforces the following constraints:

- We have one strongly connected component containing |NODES| vertices,
- Each vertex has exactly two predecessors and two successors.

Part (A) of Figure 5.760 shows the initial graph from which we start. It is derived from the set associated with each vertex. Each set describes the potential values of the succ attribute of a given vertex. Part (B) of Figure 5.760 gives the final graph associated with the **Example** slot. The tour constraint holds since the final graph corresponds to a Hamiltonian cycle.

Signature Since the maximum number of vertices of the final graph is equal to |NODES|, we can rewrite the graph property **MIN_NSCC** $=$ |NODES| to **MIN_NSCC** $\geq$ |NODES| and simplify $\underline{\overline{\textbf{MIN\_NSCC}}}$ to $\overline{\textbf{MIN\_NSCC}}$.



(A) **(B)**

Figure 5.760: Initial and final graph of the `tour` set constraint