5.221 minimum_weight_alldifferent

DESCRIPTION LINKS GRAPH

Origin [151]

Constraint minimum_weight_alldifferent(VARIABLES, MATRIX, COST)

Synonyms minimum_weight_alldiff, minimum_weight_alldistinct, min_weight_alldiff, min_weight_alldifferent, min_weight_alldistinct.

Arguments

VARIABLES	:	collection(var−dvar)
MATRIX	:	collection(i−int, j−int, c−int)
COST	:	dvar

Restrictions

$|\texttt{VARIABLES}| > 0$
required(VARIABLES, var)
$\texttt{VARIABLES.var} \geq 1$
$\texttt{VARIABLES.var} \leq |\texttt{VARIABLES}|$
required(MATRIX, [i, j, c])
increasing_seq(MATRIX, [i, j])
$\texttt{MATRIX.i} \geq 1$
$\texttt{MATRIX.i} \leq |\texttt{VARIABLES}|$
$\texttt{MATRIX.j} \geq 1$
$\texttt{MATRIX.j} \leq |\texttt{VARIABLES}|$
$|\texttt{MATRIX}| = |\texttt{VARIABLES}| * |\texttt{VARIABLES}|$

Purpose All variables of the VARIABLES collection should take a distinct value located within interval $[1, |\texttt{VARIABLES}|]$. In addition COST is equal to the sum of the costs associated with the fact that we assign value i to variable j. These costs are given by the matrix MATRIX.

Example

$$\left(\begin{array}{l} \langle 2,3,1,4\rangle, \\ \left\langle \begin{array}{lll} \mathtt{i}-1 & \mathtt{j}-1 & \mathtt{c}-4, \\ \mathtt{i}-1 & \mathtt{j}-2 & \mathtt{c}-1, \\ \mathtt{i}-1 & \mathtt{j}-3 & \mathtt{c}-7, \\ \mathtt{i}-1 & \mathtt{j}-4 & \mathtt{c}-0, \\ \mathtt{i}-2 & \mathtt{j}-1 & \mathtt{c}-1, \\ \mathtt{i}-2 & \mathtt{j}-2 & \mathtt{c}-0, \\ \mathtt{i}-2 & \mathtt{j}-3 & \mathtt{c}-8, \\ \mathtt{i}-2 & \mathtt{j}-4 & \mathtt{c}-2, \\ \mathtt{i}-3 & \mathtt{j}-1 & \mathtt{c}-3, \\ \mathtt{i}-3 & \mathtt{j}-2 & \mathtt{c}-2, \\ \mathtt{i}-3 & \mathtt{j}-3 & \mathtt{c}-1, \\ \mathtt{i}-3 & \mathtt{j}-4 & \mathtt{c}-6, \\ \mathtt{i}-4 & \mathtt{j}-1 & \mathtt{c}-0, \\ \mathtt{i}-4 & \mathtt{j}-2 & \mathtt{c}-0, \\ \mathtt{i}-4 & \mathtt{j}-3 & \mathtt{c}-6, \\ \mathtt{i}-4 & \mathtt{j}-4 & \mathtt{c}-5 \end{array} \right\rangle, 17 \end{array}\right)$$

The `minimum_weight_alldifferent` constraint holds since the cost 17 corresponds to the sum $\texttt{MATRIX}[(1-1)\cdot 4+2].\texttt{c}+\texttt{MATRIX}[(2-1)\cdot 4+3].\texttt{c}+\texttt{MATRIX}[(3-1)\cdot 4+1].\texttt{c}+\texttt{MATRIX}[(4-1)\cdot 4+4].\texttt{c} = \texttt{MATRIX}[2].\texttt{c}+\texttt{MATRIX}[7].\texttt{c}+\texttt{MATRIX}[9].\texttt{c}+\texttt{MATRIX}[16].\texttt{c} = 1+8+3+5$.

Algorithm The Hungarian method for the assignment problem [216] can be used for evaluating the bounds of the `COST` variable. A filtering algorithm is described in [343]. It can be used for handling both side of the `minimum_weight_alldifferent` constraint:

- Evaluating a lower bound of the `COST` variable and pruning the variables of the `VARIABLES` collection in order to not exceed the maximum value of `COST`.
- Evaluating an upper bound of the `COST` variable and pruning the variables of the `VARIABLES` collection in order to not be under the minimum value of `COST`.

Systems `assignment` in **SICStus**.

See also **attached to cost variant:** `alldifferent`.

common keyword: `global_cardinality_with_costs` *(cost filtering constraint,weighted assignment)*, `sum_of_weights_of_distinct_values` *(weighted assignment)*, `weighted_partial_alldiff` *(cost filtering constraint,weighted assignment)*.

Keywords **application area:** assignment.

characteristic of a constraint: core.

filtering: cost filtering constraint, Hungarian method for the assignment problem.

final graph structure: one_succ.

modelling: cost matrix.

problems: weighted assignment.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.key
Graph property(ies)	• **NTREE** $= 0$ • **SUM_WEIGHT_ARC** $\left(\text{MATRIX}\left[\sum \left(\begin{array}{l} (\text{variables1.key}-1)*\lvert\text{VARIABLES}\rvert, \\ \text{variables1.var} \end{array} \right)\right].\text{c} \right) = \text{COST}$

Graph model

Since each variable takes one value, and because of the arc constraint variables1 = variables.key, each vertex of the initial graph belongs to the final graph and has exactly one successor. Therefore the sum of the out-degrees of the vertices of the final graph is equal to the number of vertices of the final graph. Since the sum of the in-degrees is equal to the sum of the out-degrees, it is also equal to the number of vertices of the final graph. Since **NTREE** $= 0$, each vertex of the final graph belongs to a circuit. Therefore each vertex of the final graph has at least one predecessor. Since we saw that the sum of the in-degrees is equal to the number of vertices of the final graph, each vertex of the final graph has exactly one predecessor. We conclude that the final graph consists of a set of vertex-disjoint elementary circuits.

Finally the graph constraint expresses the fact that the COST variable is equal to the sum of the elementary costs associated with each variable-value assignment. All these elementary costs are recorded in the MATRIX collection. More precisely, the cost c_{ij} is recorded in the attribute c of the $((i-1)\cdot|\text{VARIABLES}|+j)^{th}$ entry of the MATRIX collection. This is ensured by the increasing restriction that enforces the fact that the items of the MATRIX collection are sorted in lexicographically increasing order according to attributes i and j.

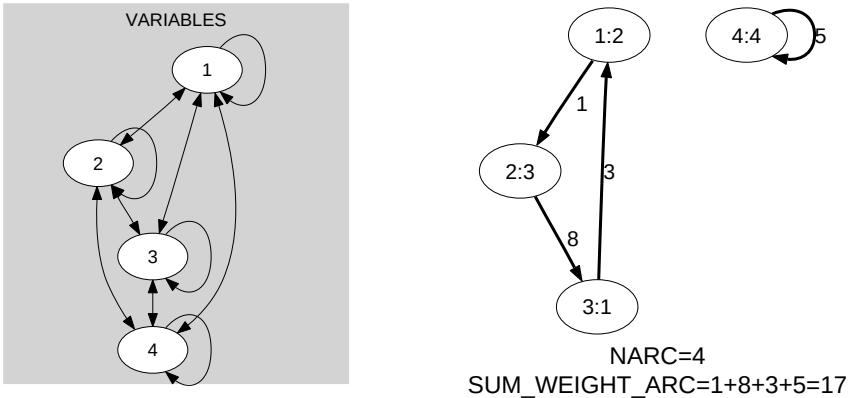

Figure 5.426: Initial and final graph of the minimum_weight_alldifferent constraint

Parts (A) and (B) of Figure 5.426 respectively show the initial and final graph associated with the **Example** slot. Since we use the **SUM_WEIGHT_ARC** graph property, the

arcs of the final graph are stressed in bold. We also indicate their corresponding weight.