5.267 minimum_weight_alldifferent

DESCRIPTION LINKS GRAPH

Origin [171]

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

|VARIABLES| > 0
required(VARIABLES, var)
VARIABLES.var ≥ 1
VARIABLES.var ≤ |VARIABLES|
required(MATRIX, [i, j, c])
increasing_seq(MATRIX, [i, j])
MATRIX.i ≥ 1
MATRIX.i ≤ |VARIABLES|
MATRIX.j ≥ 1
MATRIX.j ≤ |VARIABLES|
|MATRIX| = |VARIABLES| ∗ |VARIABLES|

Purpose

All variables of the VARIABLES collection should take a distinct value located within interval $[1, |\text{VARIABLES}|]$. 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 $\texttt{MATRIX}[(1-1)\cdot 4+2].\texttt{c}+\texttt{MATRIX}[(2-1)\cdot 4+3].\texttt{c}+\texttt{MATRIX}[(3-1)\cdot 4+1].\texttt{c}+\texttt{MATRIX}[(4-1)\cdot 4+4].\texttt{c} = \texttt{MATRIX}[2].\texttt{c}+\texttt{MATRIX}[7].\texttt{c}+\texttt{MATRIX}[9].\texttt{c}+\texttt{MATRIX}[16].\texttt{c} = 1+8+3+5$.

All solutions

Figure 5.588 gives all solutions to the following non ground instance of the minimum_weight_alldifferent constraint:
$\texttt{V}_1 \in [2,4]$, $\texttt{V}_2 \in [2,3]$, $\texttt{V}_3 \in [1,6]$, $\texttt{V}_4 \in [2,5]$, $\texttt{V}_5 \in [2,3]$, $\texttt{V}_6 \in [1,6]$, $\texttt{V} \in [0,25]$,
$\texttt{minimum\_weight\_alldifferent}(\langle \texttt{V}_1,\texttt{V}_2,\texttt{V}_3,\texttt{V}_4,\texttt{V}_5,\texttt{V}_6\rangle,$
$\langle 1\,1\,5,\ 1\,2\,0,\ 1\,3\,1,\ 1\,4\,1,\ 1\,5\,3,\ 1\,6\,0,$
$2\,1\,2,\ 2\,2\,7,\ 2\,3\,0,\ 2\,4\,2,\ 2\,5\,5,\ 2\,6\,1,$
$3\,1\,3,\ 3\,2\,3,\ 3\,3\,6,\ 3\,4\,6,\ 3\,5\,0,\ 3\,6\,9,$
$4\,1\,4,\ 4\,2\,3,\ 4\,3\,0,\ 4\,4\,0,\ 4\,5\,0,\ 4\,6\,2,$
$5\,1\,2,\ 5\,2\,0,\ 5\,3\,6,\ 5\,4\,3,\ 5\,5\,7,\ 5\,6\,2,$
$6\,1\,5,\ 6\,2\,4,\ 6\,3\,5,\ 6\,4\,4,\ 6\,5\,5,\ 6\,6\,4\rangle, \texttt{C})$.

① $(\langle 4,2,1,5,3,6\rangle,\ \mathbf{21})$
② $(\langle 4,3,1,5,2,6\rangle,\ \mathbf{8})$
③ $(\langle 4,3,6,5,2,1\rangle,\ \mathbf{15})$

$$\begin{array}{c} ① \\ \mathbf{21} \end{array}
\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline V_1 & 5 & 0 & 1 & \mathbf{1} & 3 & 0 \\ V_2 & 2 & \mathbf{7} & 0 & 2 & 5 & 1 \\ V_3 & \mathbf{3} & 3 & 6 & 6 & 0 & 9 \\ V_4 & 4 & 3 & 0 & 0 & \mathbf{0} & 2 \\ V_5 & 2 & 0 & \mathbf{6} & 3 & 7 & 2 \\ V_6 & 5 & 4 & 5 & 4 & 5 & \mathbf{4} \end{array}
\qquad
\begin{array}{c} ② \\ \mathbf{8} \end{array}
\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline V_1 & 5 & 0 & 1 & \mathbf{1} & 3 & 0 \\ V_2 & 2 & 7 & \mathbf{0} & 2 & 5 & 1 \\ V_3 & \mathbf{3} & 3 & 6 & 6 & 0 & 9 \\ V_4 & 4 & 3 & 0 & 0 & \mathbf{0} & 2 \\ V_5 & 2 & \mathbf{0} & 6 & 3 & 7 & 2 \\ V_6 & 5 & 4 & 5 & 4 & 5 & \mathbf{4} \end{array}
\qquad
\begin{array}{c} ③ \\ \mathbf{15} \end{array}
\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline V_1 & 5 & 0 & 1 & \mathbf{1} & 3 & 0 \\ V_2 & 2 & 7 & \mathbf{0} & 2 & 5 & 1 \\ V_3 & 3 & 3 & 6 & 6 & 0 & \mathbf{9} \\ V_4 & 4 & 3 & 0 & 0 & \mathbf{0} & 2 \\ V_5 & 2 & \mathbf{0} & 6 & 3 & 7 & 2 \\ V_6 & \mathbf{5} & 4 & 5 & 4 & 5 & 4 \end{array}$$

Figure 5.588: All solutions corresponding to the non ground example of the minimum_weight_alldifferent constraint of the **All solutions** slot

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{MATRIX.c}) > 1$
$\texttt{MATRIX.c} > 0$

Arg. properties

Functional dependency: COST determined by VARIABLES and MATRIX.

Algorithm

The Hungarian method for the assignment problem [243] can be used for evaluating the bounds of the COST variable. A filtering algorithm is described in [377]. 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 all_different in **SICStus**, all_distinct 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, functional dependency.

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** $\left(\texttt{MATRIX} \left[\sum \left(\begin{array}{l} (\texttt{variables1.key} - 1) * |\texttt{VARIABLES}|, \\ \texttt{variables1.var} \end{array} \right) \right] .\texttt{c} \right) = \texttt{COST}$

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 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 $((i-1) \cdot |\texttt{VARIABLES})| + j)^{th}$ entry of the MATRIX collection. This is ensured by the increasing restriction that enforces that the items of the MATRIX collection are sorted in lexicographically increasing order according to attributes i and j.

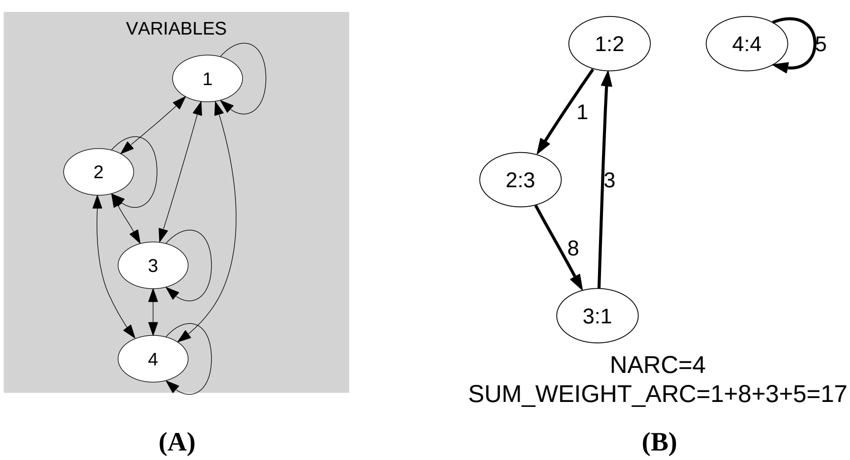

Figure 5.589: Initial and final graph of the minimum_weight_alldifferent constraint

Parts (A) and (B) of Figure 5.589 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.