5.142 global_cardinality_with_costs

DESCRIPTION LINKS GRAPH

Origin [311]

Constraint global_cardinality_with_costs(VARIABLES, VALUES, MATRIX, COST)

Synonyms gccc, cost_gcc.

Arguments

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

Restrictions

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

Purpose

Each value VALUES[i].val should be taken by exactly VALUES[i].noccurrence variables of the VARIABLES collection. In addition the COST of an assignment is equal to the sum of the elementary costs associated with the fact that we assign the i^{th} variable of the VARIABLES collection to the j^{th} value of the VALUES collection. These elementary costs are given by the MATRIX collection.

Example

$$\left(\begin{array}{l}\langle 3,3,3,6\rangle, \\ \left\langle\begin{array}{ll}\texttt{val}-3 & \texttt{noccurrence}-3, \\ \texttt{val}-5 & \texttt{noccurrence}-0, \\ \texttt{val}-6 & \texttt{noccurrence}-1\end{array}\right\rangle, \\ \left\langle\begin{array}{lll}\texttt{i}-1 & \texttt{j}-1 & \texttt{c}-4, \\ \texttt{i}-1 & \texttt{j}-2 & \texttt{c}-1, \\ \texttt{i}-1 & \texttt{j}-3 & \texttt{c}-7, \\ \texttt{i}-2 & \texttt{j}-1 & \texttt{c}-1, \\ \texttt{i}-2 & \texttt{j}-2 & \texttt{c}-0, \\ \texttt{i}-2 & \texttt{j}-3 & \texttt{c}-8, \\ \texttt{i}-3 & \texttt{j}-1 & \texttt{c}-3, \\ \texttt{i}-3 & \texttt{j}-2 & \texttt{c}-2, \\ \texttt{i}-3 & \texttt{j}-3 & \texttt{c}-1, \\ \texttt{i}-4 & \texttt{j}-1 & \texttt{c}-0, \\ \texttt{i}-4 & \texttt{j}-2 & \texttt{c}-0, \\ \texttt{i}-4 & \texttt{j}-3 & \texttt{c}-6\end{array}\right\rangle, 14\end{array}\right)$$

The `global_cardinality_with_costs` constraint holds since:

- Values 3, 5 and 6 respectively occur 3, 0 and 1 times within the collection $\langle 3,3,3,6\rangle$.
- The COST argument corresponds to the sum of the costs respectively associated with the first, second, third and fourth items of $\langle 3,3,3,6\rangle$, namely 4, 1, 3 and 6.

Typical

$|\texttt{VARIABLES}| > 1$
$\texttt{range}(\texttt{VARIABLES.var}) > 1$
$|\texttt{VALUES}| > 1$
$\texttt{range}(\texttt{VALUES.noccurrence}) > 1$
$\texttt{range}(\texttt{MATRIX.c}) > 1$
$|\texttt{VARIABLES}| > |\texttt{VALUES}|$

Usage

A classical utilisation of the `global_cardinality_with_costs` constraint corresponds to the following assignment problem. We have a set of persons $\mathcal{P}$ as well as a set of jobs $\mathcal{J}$ to perform. Each job requires a number of persons restricted to a specified interval. In addition each person p has to be assigned to one specific job taken from a subset $\mathcal{J}_p$ of $\mathcal{J}$. There is a cost C_{pj} associated with the fact that person p is assigned to job j. The previous problem is modelled with one single `global_cardinality_with_costs` constraint where the persons and the jobs respectively correspond to the items of the VARIABLES and VALUES collection.

The `global_cardinality_with_costs` constraint can also be used for modelling a conjunction $\texttt{alldifferent}(\texttt{X}_1,\texttt{X}_2,\ldots,\texttt{X}_n)$ and $\alpha_1\cdot\texttt{X}_1+\alpha_2\cdot\texttt{X}_2+\ldots+\alpha_n\cdot\texttt{X}_n=\texttt{COST}$. For this purpose we set the domain of the `noccurrence` variables to $\{0,1\}$ and the cost attribute `c` of a variable $\texttt{X}_i$ and one of its potential value `j` to $\alpha_i\cdot\texttt{j}$. In practice this can be used for the *magic squares* and the *magic hexagon* problems where all the α_i are set to 1.

Algorithm

[311, 313]

Reformulation

Let n and m respectively denote the number of items of the VARIABLES and of the VALUES collections. Let $v_1,v_2,\ldots,v_m$ denote the values $\texttt{VALUES}[1].\texttt{val},\texttt{VALUES}[2].\texttt{val},\ldots,\texttt{VALUES}[m].\texttt{val}$. In addition let $LINE_i$ $(1\le i\le n)$

denote the values $\langle \texttt{MATRIX}[m\cdot(i-1)+1].\texttt{c}, \texttt{MATRIX}[m\cdot(i-1)+2].\texttt{c}, \ldots, \texttt{MATRIX}[m\cdot i].\texttt{c}\rangle$, i.e., the i-th line of the matrix MATRIX.

By introducing $2 \cdot n$ auxiliary variables $U_1, U_2, \ldots, U_n$ and $C_1, C_2, \ldots, C_n$, the global_cardinality_with_costs(VARIABLES, VALUES, MATRIX, COST) constraint can be expressed in term of the conjunction of one global_cardinality(VARIABLES, VALUES) constraint, $2 \cdot n$ element constraints and one arithmetic constraint sum_ctr.

For each variable V_i ($1 \leq i \leq |\texttt{VARIABLES}|$) of the VARIABLES collection a first element$(U_i, \langle v_1, v_2, \ldots, v_m\rangle, V_i)$ constraint provides the correspondence between the variable V_i and the index of the value U_i to which it is assigned. A second element$(U_i, LINE_i, C_i)$ links the previous index U_i to the cost C_i variable associated with variable V_i. Finally the total cost COST is equal to the sum $C_1 + C_2 + \ldots + C_n$.

In the context of the **Example** slot we get the following conjunction of constraints:

global_cardinality($\langle 3, 3, 3, 6\rangle$,
$\langle$val $-\ 3$ noccurrence $-\ 3$,
val $-\ 5$ noccurrence $-\ 0$,
val $-\ 6$ noccurrence $-\ 1\rangle$),
element$(1, \langle 3, 5, 6\rangle, 3)$,
element$(1, \langle 3, 5, 6\rangle, 3)$,
element$(1, \langle 3, 5, 6\rangle, 3)$,
element$(3, \langle 3, 5, 6\rangle, 6)$,
element$(1, \langle 4, 1, 7\rangle, 4)$,
element$(1, \langle 1, 0, 8\rangle, 1)$,
element$(1, \langle 3, 2, 1\rangle, 3)$,
element$(3, \langle 0, 0, 6\rangle, 6)$,
$14 = 4 + 1 + 3 + 6$.

Systems global_cardinality in **SICStus**.

See also **attached to cost variant:** global_cardinality *(*cost *associated with each* variable,value *pair removed).*

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

implies: global_cardinality.

Keywords **application area:** assignment.

filtering: cost filtering constraint.

modelling: cost matrix, scalar product.

problems: weighted assignment.

puzzles: magic square, magic hexagon.

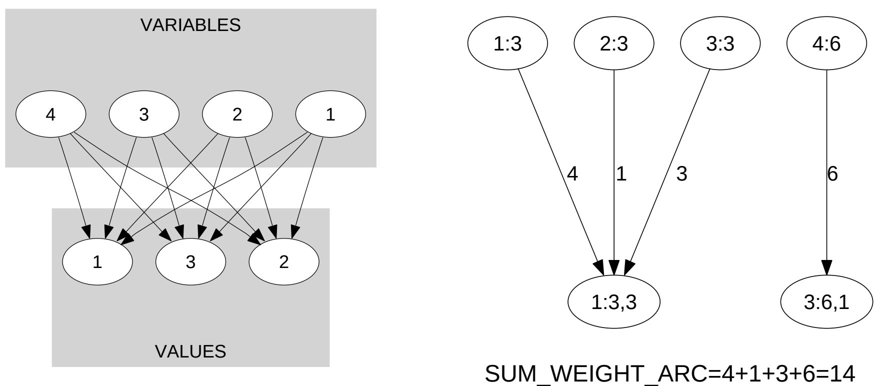
For all items of `VALUES`:

Arc input(s)	`VARIABLES`
Arc generator	*SELF* $\mapsto$ `collection`(`variables`)
Arc arity	1
Arc constraint(s)	`variables.var` = `VALUES.val`
Graph property(ies)	**NVERTEX** = `VALUES.noccurrence`

Arc input(s)	`VARIABLES` `VALUES`
Arc generator	*PRODUCT* $\mapsto$ `collection`(`variables`, `values`)
Arc arity	2
Arc constraint(s)	`variables.var` = `values.val`
Graph property(ies)	**SUM_WEIGHT_ARC**(`MATRIX`[(`variables.key` − 1) ∗ |`VALUES`| + `values.key`].`c`) = `COST`

Graph model The first graph constraint enforces each value of the `VALUES` collection to be taken by a specific number of variables of the `VARIABLES` collection. It is identical to the graph constraint used in the `global_cardinality` constraint. The second 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 $((i-1)\cdot|\texttt{VALUES})|+j)^{th}$ 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`.

Parts (A) and (B) of Figure 5.274 respectively show the initial and final graph associated with the second graph constraint of the **Example** slot.



(A) **(B)**

Figure 5.274: Initial and final graph of the `global_cardinality_with_costs` constraint