5.167 global_cardinality_with_costs

DESCRIPTION LINKS GRAPH

Origin [344]

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 variable i 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.

All solutions

Figure 5.358 gives all solutions to the following non ground instance of the `global_cardinality_with_costs` constraint:
$V_1 \in [3,4]$, $V_2 \in [2,3]$, $V_3 \in [1,2]$, $V_4 \in [2,4]$, $V_5 \in [2,3]$, $V_6 \in [1,2]$,
$O_1 \in [1,1]$, $O_2 \in [2,3]$, $O_3 \in [0,1]$, $O_4 \in [2,3]$,
$C \in [0,16]$,
`global_cardinality_with_costs`$(\langle V_1,V_2,V_3,V_4,V_5,V_6\rangle,$
$\langle 1\,O_1,\ 2\,O_2,\ 3\,O_3,\ 4\,O_4\rangle,$
$\langle 1\,1\,5,\ 1\,2\,0,\ 1\,3\,1,\ 1\,4\,1,$
$2\,1\,2,\ 2\,2\,7,\ 2\,3\,0,\ 2\,4\,2,$
$3\,1\,3,\ 3\,2\,3,\ 3\,3\,6,\ 3\,4\,6,$
$4\,1\,4,\ 4\,2\,3,\ 4\,3\,0,\ 4\,4\,0,$
$5\,1\,2,\ 5\,2\,0,\ 5\,3\,6,\ 5\,4\,3,$
$6\,1\,5,\ 6\,2\,4,\ 6\,3\,5,\ 6\,4\,4\rangle, C)$.

Typical

$|$`VARIABLES`$| > 1$
range(`VARIABLES.var`) > 1
$|$`VALUES`$| > 1$
range(`VALUES.noccurrence`) > 1
range(`MATRIX.c`) > 1
$|$`VARIABLES`$| > |$`VALUES`$|$

Arg. properties

- Functional dependency: `VALUES.noccurrence` determined by `VARIABLES`.
- Functional dependency: `COST` determined by `VARIABLES`, `VALUES` and `MATRIX`.

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

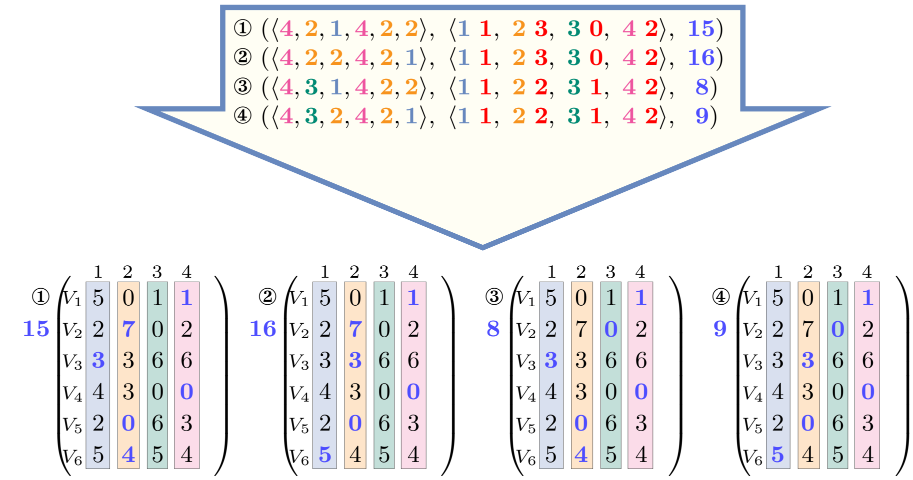


Figure 5.358: All solutions corresponding to the non ground example of the `global_cardinality_with_costs` constraint of the **All solutions** slot

$\mathcal{J}$ 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 a 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 `alldifferent`$(\mathtt{X_1}, \mathtt{X_2}, \ldots, \mathtt{X_n})$ and $\alpha_1 \cdot \mathtt{X_1} + \alpha_2 \cdot \mathtt{X_2} + \cdots + \alpha_n \cdot \mathtt{X_n} = \mathtt{COST}$. For this purpose we set the domain of the `noccurrence` variables to $\{0, 1\}$ and the cost attribute `c` of a variable $\mathtt{X_i}$ and one of its potential value `j` to $\alpha_i \cdot \mathtt{j}$. 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 A filtering algorithm achieving arc-consistency independently on each side (i.e., the *greater than or equal to* side and the *less than or equal to* side) of the `global_cardinality_with_costs` constraint is described in [344, 346]. This algorithm assumes for each value a fixed minimum and maximum number of occurrences. If we rather have occurrence variables, the **Reformulation slot** explains how to also obtain some propagation from the cost variable back to the occurrence variables.

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 $\mathtt{VALUES}[1].\mathtt{val}, \mathtt{VALUES}[2].\mathtt{val}, \ldots, \mathtt{VALUES}[m].\mathtt{val}$. In addition let $LINE_i$ (with $i \in [1, n]$) denote the values $\langle \mathtt{MATRIX}[m \cdot (i-1)+1].\mathtt{c}, \mathtt{MATRIX}[m \cdot (i-1)+2].\mathtt{c}, \ldots, \mathtt{MATRIX}[m \cdot i].\mathtt{c} \rangle$, i.e., line i 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 (with $i \in [1, |\texttt{VARIABLES}|]$) 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 $C_1 + C_2 + \cdots + C_n$.

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$(3, \langle 0, 0, 6\rangle, 6),$
$14 = 4 + 1 + 3 + 6.$

We now show how to add implied constraints that can also propagate from the cost variable back to the occurrence variables. Let $O_1, O_2, \ldots, O_m$ respectively denote the variables VALUES$[1]$.noccurrence, VALUES$[2]$.noccurrence, $\ldots$, VALUES$[m]$.noccurrence. The idea is to get for each value v_i (with $i \in [1, m]$) an idea of its minimum and maximum contribution in the total cost COST that is linked to the number of times it is assigned to a variables of VARIABLES. E.g., if value v_i (with $i \in [1, m]$) is used twice, then the corresponding minimum (respectively maximum) contribution in the total cost COST will be at least equal to the sum of the two smallest (respectively largest) costs attached to row i. Let D_i (with $i \in [1, m]$) denotes the contribution that stems from the variables of VARIABLES that are assigned value v_i. For each value v_i (with $i \in [1, m]$) we create one element constraint for linking $O_i + 1$ to the corresponding minimum contribution LOW_i. The table of that element constraint has $n + 1$ entries, where entry j (with $j \in [0, n]$) corresponds to the sum of the j^{th} smallest entries of row i of the cost matrix MATRIX. Similarly we create for each value v_i (with $i \in [1, m]$) one element constraint for linking $O_i + 1$ to the corresponding maximum contribution UP_i. The table of that element constraint also has $n + 1$ entries, where entry j (with $j \in [0, n]$) corresponds to the sum of the j^{th} largest entries of row i of the cost matrix MATRIX.

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

COST $= D_1 + D_2 + D_3,$
$n = O_1 + O_2 + O_3,$
$P_1 = O_1 + 1,$
$P_2 = O_2 + 1,$
$P_3 = O_3 + 1,$
element$(P_1, \langle 0, 0, 1, 4, 8\rangle, LOW_1),$
element$(P_2, \langle 0, 0, 0, 1, 3\rangle, LOW_2),$

element($P_3, \langle 0, 1, 7, 14, 22\rangle, LOW_3$),
element($P_1, \langle 0, 4, 7, 8, 8\rangle, UP_1$),
element($P_2, \langle 0, 2, 3, 3, 3\rangle, UP_2$),
element($P_3, \langle 0, 8, 15, 21, 22\rangle, UP_3$),
$LOW_1 \leq D_1$, $D_1 \leq UP_1$,
$LOW_2 \leq D_2$, $D_2 \leq UP_2$,
$LOW_3 \leq D_3$, $D_3 \leq UP_3$.

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.

constraint arguments: pure functional dependency.

filtering: cost filtering constraint.

heuristics: regret based heuristics, regret based heuristics in matrix problems.

modelling: cost matrix, scalar product, functional dependency.

problems: weighted assignment.

puzzles: magic square, magic hexagon.

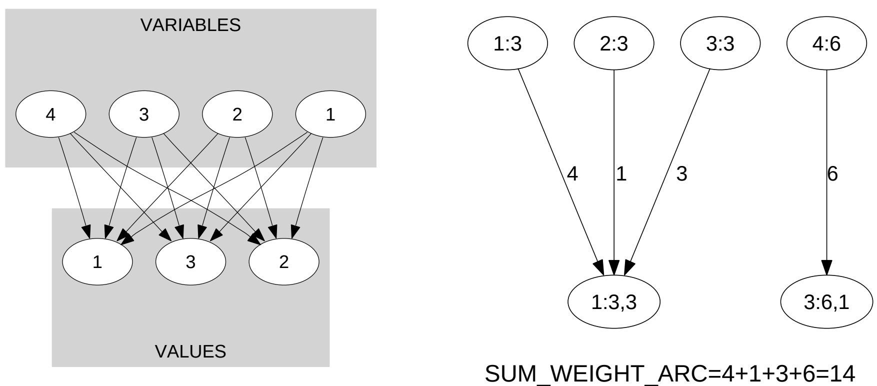
For all items of `VALUES`:

Arc input(s)	`VARIABLES`
Arc generator	*SELF* ↦ `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* ↦ `collection`(`variables`, `values`)
Arc arity	2
Arc constraint(s)	`variables.var` = `values.val`
Graph property(ies)	$\mathbf{SUM\_WEIGHT\_ARC}\left(\texttt{MATRIX}\left[\sum\left(\begin{array}{l}(\texttt{variables.key}-1)*|\texttt{VALUES}|,\\ \texttt{values.key}\end{array}\right)\right].\texttt{c}\right)=\texttt{COST}$

Graph model The first graph constraint forces 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 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 5.359 respectively show the initial and final graph associated with the second graph constraint of the **Example** slot.



(A) **(B)**

Figure 5.359: Initial and final graph of the `global_cardinality_with_costs` constraint