5.164   global_cardinality_low_up

Origin   Used for defining sliding_distribution.

Constraint   global_cardinality_low_up(VARIABLES, VALUES)

Synonyms   gcc_low_up, gcc.

Arguments
```
VARIABLES  :  collection(var−dvar)
VALUES     :  collection(val−int, omin−int, omax−int)
```

Restrictions
required(VARIABLES, var)
$|\text{VALUES}| > 0$
required(VALUES, [val, omin, omax])
distinct(VALUES, val)
$\text{VALUES.omin} \geq 0$
$\text{VALUES.omax} \leq |\text{VARIABLES}|$
$\text{VALUES.omin} \leq \text{VALUES.omax}$

Purpose   Each value $\text{VALUES}[i].\text{val}$ $(1 \leq i \leq |\text{VALUES}|)$ should be taken by at least $\text{VALUES}[i].\text{omin}$ and at most $\text{VALUES}[i].\text{omax}$ variables of the VARIABLES collection.

Example
$$\left(\begin{array}{l} \langle 3, 3, 8, 6 \rangle, \\ \left\langle \begin{array}{lll} \text{val} - 3 & \text{omin} - 2 & \text{omax} - 3, \\ \text{val} - 5 & \text{omin} - 0 & \text{omax} - 1, \\ \text{val} - 6 & \text{omin} - 1 & \text{omax} - 2 \end{array} \right\rangle \end{array} \right)$$

The global_cardinality_low_up constraint holds since values 3, 5 and 6 are respectively used 2 $(2 \leq 2 \leq 3)$, 0 $(0 \leq 0 \leq 1)$ and 1 $(1 \leq 1 \leq 2)$ times within the collection $\langle 3, 3, 8, 6 \rangle$ and since no constraint was specified for value 8.

Typical
$|\text{VARIABLES}| > 1$
range(VARIABLES.var) > 1
$|\text{VALUES}| > 1$
$\text{VALUES.omin} \leq |\text{VARIABLES}|$
$\text{VALUES.omax} > 0$
$\text{VALUES.omax} < |\text{VARIABLES}|$
$|\text{VARIABLES}| > |\text{VALUES}|$
in_attr(VARIABLES, var, VALUES, val)

Symmetries	• Items of VARIABLES are permutable.
	• An occurrence of a value of VARIABLES.var that does not belong to VALUES.val can be replaced by any other value that also does not belong to VALUES.val.
	• Items of VALUES are permutable.
	• VALUES.omin can be decreased to any value ≥ 0.
	• VALUES.omax can be increased to any value $\leq |\text{VARIABLES}|$.
	• All occurrences of two distinct values in VARIABLES.var or VALUES.val can be swapped; all occurrences of a value in VARIABLES.var or VALUES.val can be renamed to any unused value.

Arg. properties

Contractible wrt. VALUES.

Remark

Within the context of linear programming [215, page 376] provides relaxations of the global_cardinality_low_up constraint.

In **MiniZinc** (http://www.minizinc.org/) there is also a global_cardinality_low_up_closed constraint where all variables must be assigned a value from the val attribute.

Algorithm

A filtering algorithm achieving arc-consistency for the global_cardinality_low_up constraint is given in [342]. This algorithm is based on a flow model of the global_cardinality_low_up constraint where there is a one-to-one correspondence between feasible flows in the flow model and solutions of the global_cardinality_low_up constraint. The leftmost part of Figure 3.30 illustrates this flow model.

The global_cardinality_low_up constraint is entailed if and only if for each value v equal to VALUES[i].val (with $1 \leq i \leq |\text{VALUES}|$) the following two conditions hold:

1. The number of variables of the VARIABLES collection assigned value v is greater than or equal to VALUES[i].omin.

2. The number of variables of the VARIABLES collection that can potentially be assigned value v is less than or equal to VALUES[i].omax.

Reformulation

A reformulation of the global_cardinality_low_up, involving linear constraints, preserving bound-consistency was introduced in [71]. For each potential interval $[l, u]$ of consecutive values this model uses $|\text{VARIABLES}|$ 0-1 variables $B_{1,l,u}, B_{2,l,u}, \ldots, B_{|\text{VARIABLES}|,l,u}$ for modelling the fact that each variable of the collection VARIABLES is assigned a value within interval $[l, u]$ (i.e., $\forall i \in [1, |\text{VARIABLES}|] : B_{i,l,u} \Leftrightarrow l \leq \text{VARIABLES}[i].\text{var} \wedge \text{VARIABLES}[i].\text{var} \leq u$), as well as one domain variable $C_{l,u}$ for counting how many values of $[l, u]$ are assigned to variables of VARIABLES (i.e. $C_{l,u} = B_{1,l,u} + B_{2,l,u} + \cdots + B_{|\text{VARIABLES}|,l,u}$). The lower and upper bounds of variable $C_{l,u}$ are respectively initially set with respect to the minimum and maximum number of possible occurrences of the values of interval $[l, u]$. Finally, assuming that s is the smallest value that can be assigned to the variables of VARIABLES, the constraint $C_{s,u} = C_{s,k} + C_{k+1,u}$ is stated for each $k \in [s, u-1]$.

Systems

globalCardinality in **Choco**, global_cardinality_low_up in **MiniZinc**.

Used in

sliding_distribution.

See also   **common keyword:** open_global_cardinality *(assignment,counting constraint)*.

generalisation: global_cardinality *(*fixed interval *replaced by* variable*)*.

implied by: increasing_global_cardinality *(a* global_cardinality_low_up *constraint where the* variables *are increasing)*, same_and_global_cardinality_low_up.

related: ordered_global_cardinality *(restrictions are done on nested sets of values, all starting from first value)*.

shift of concept: global_cardinality_low_up_no_loop *(assignment of a* variable *to its position is ignored)*.

soft variant: open_global_cardinality_low_up *(a* set variable *defines the set of variables that are actually considered)*.

specialisation: alldifferent *(each value should occur at most once)*.

system of constraints: sliding_distribution *(one* global_cardinality_low_up *constraint for each sliding sequence of* SEQ *consecutive* variables*)*.

Keywords   **application area:** assignment.

constraint type: value constraint, counting constraint.

filtering: flow, arc-consistency, bound-consistency, DFS-bottleneck, entailment.
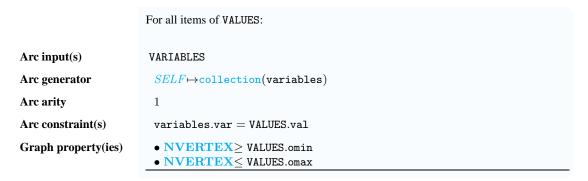
Cond. implications   global_cardinality_low_up(VARIABLES, VALUES)
   with increasing(VARIABLES)
  implies increasing_global_cardinality(VARIABLES, VALUES).

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**$\geq$ `VALUES.omin` • **NVERTEX**$\leq$ `VALUES.omax`

Graph model   Since we want to express one unary constraint for each value we use the "For all items of `VALUES`" iterator. Part (A) of Figure 5.372 shows the initial graphs associated with each value 3, 5 and 6 of the `VALUES` collection of the **Example** slot. Part (B) of Figure 5.372 shows the two corresponding final graphs respectively associated with values 3 and 6 that are both assigned to the variables of the `VARIABLES` collection (since value 5 is not assigned to any variable of the `VARIABLES` collection the final graph associated with value 5 is empty). Since we use the **NVERTEX** graph property, the vertices of the final graphs are stressed in bold.
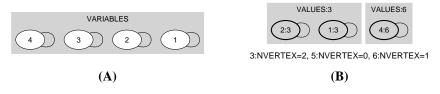

(A)   **(B)**

Figure 5.372: Initial and final graph of the `global_cardinality_low_up` constraint