5.35 assign_and_nvalues

DESCRIPTION LINKS GRAPH

Origin Derived from assign_and_counts and nvalues.

Constraint assign_and_nvalues(ITEMS, RELOP, LIMIT)

Arguments

ITEMS	:	collection(bin−dvar, value−dvar)
RELOP	:	atom
LIMIT	:	dvar

Restrictions

required(ITEMS, [bin, value])
RELOP $\in [=, \neq, <, \geq, >, \leq]$

Purpose Given several items (each of them having a specific value that may not be initially fixed), and different bins, assign each item to a bin, so that the number n of distinct values in each bin satisfies the condition n RELOP LIMIT.

Example

$$\left(\begin{array}{l} \{\texttt{bin}-2 \quad \texttt{value}-3,\\ \left\langle \begin{array}{ll} \texttt{bin}-1 & \texttt{value}-5,\\ \texttt{bin}-2 & \texttt{value}-3,\\ \texttt{bin}-2 & \texttt{value}-3, \end{array} \right\rangle, \leq, 2\\ \texttt{bin}-2 \quad \texttt{value}-4\} \end{array}\right)$$

Figure 5.99 depicts the solution corresponding to the example.

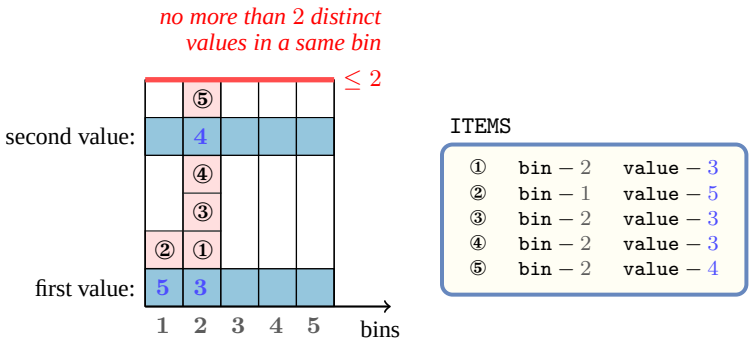


Figure 5.99: An assignment with at most two distinct values in parallel (values 3 and 4 in bin 2 and value 5 in bin 1)

The assign_and_nvalues constraint holds since for each used bin (i.e., namely bins 1 and 2) the number of distinct colours of the corresponding assigned items is less than or equal to the limit 2.

Typical

$|\texttt{ITEMS}| > 1$
$\texttt{range}(\texttt{ITEMS.bin}) > 1$
$\texttt{range}(\texttt{ITEMS.value}) > 1$
$\texttt{RELOP} \in [<, \leq]$
$\texttt{LIMIT} > 1$
$\texttt{LIMIT} < |\texttt{ITEMS}|$

Symmetries

- Items of ITEMS are permutable.
- All occurrences of two distinct values of ITEMS.bin can be swapped; all occurrences of a value of ITEMS.bin can be renamed to any unused value.

Arg. properties

- Contractible wrt. ITEMS when $\texttt{RELOP} \in [<, \leq]$.
- Extensible wrt. ITEMS when $\texttt{RELOP} \in [\geq, >]$.

Usage

Let us give two examples where the assign_and_nvalues constraint is useful:

- Quite often, in bin-packing problems, each item has a specific type, and one wants to assign items of similar type to each bin.
- In a vehicle routing problem, one wants to restrict the number of towns visited by each vehicle. Note that several customers may be located at the same town. In this example, each bin would correspond to a vehicle, each item would correspond to a visit to a customer, and the colour of an item would be the location of the corresponding customer.

See also

assignment dimension removed: nvalue, nvalues.

common keyword: nvalues_except_0 *(number of distinct values)*.

related: roots.

used in graph description: nvalues.

Keywords

application area: assignment.

final graph structure: acyclic, bipartite, no loop.

modelling: assignment dimension, number of distinct values.

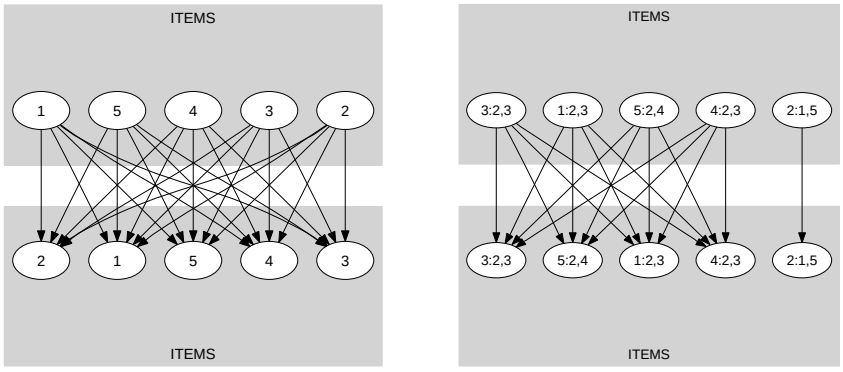
Arc input(s)	ITEMS ITEMS
Arc generator	$PRODUCT \mapsto$ collection(items1, items2)
Arc arity	2
Arc constraint(s)	items1.bin = items2.bin
Graph class	• ACYCLIC • BIPARTITE • NO_LOOP
Sets	$\text{SUCC} \mapsto \left[\begin{array}{l}\text{source},\\ \text{variables} - \text{col}\left(\begin{array}{l}\text{VARIABLES}-\text{collection}(\text{var}-\text{dvar}),\\ [\text{item}(\text{var} - \text{ITEMS.value})]\end{array}\right)\end{array}\right]$
Constraint(s) on sets	nvalues(variables, RELOP, LIMIT)

Graph model

We enforce the nvalues constraint on the items that are assigned to the same bin.

Parts (A) and (B) of Figure 5.100 respectively show the initial and final graph associated with the **Example** slot. The final graph consists of the following two connected components:

- The connected component containing 8 vertices corresponds to the items that are assigned to bin 2.
- The connected component containing 2 vertices corresponds to the items that are assigned to bin 1.



(A) **(B)**

Figure 5.100: Initial and final graph of the assign_and_nvalues constraint

The assign_and_nvalues constraint holds since for each set of successors of the vertices of the final graph no more than two distinct values are used:

- The unique item assigned to bin 1 uses value 5.

- Items assigned to bin 2 use values 3 and 4.