5.42 bin_packing_capa

DESCRIPTION LINKS

Origin Derived from bin_packing.

Constraint bin_packing_capa(BINS, ITEMS)

Arguments

BINS	:	collection(id−int, capa−int)
ITEMS	:	collection(bin−dvar, weight−int)

Restrictions

required(BINS, [id, capa])
distinct(BINS, id)
BINS.id ≥ 1
BINS.id ≤ |BINS|
BINS.capa ≥ 0
required(ITEMS, [bin, weight])
ITEMS.weight ≥ 0

Purpose Given several items of the collection ITEMS (each of them having a specific weight), and different bins described the the items of collection BINS (each of them having a specific capacity capa), assign each item to a bin so that the total weight of the items in each bin does not exceed the capacity of the bin.

Example

$$\left(\begin{array}{l} \left\langle \begin{array}{ll} \mathtt{id}-1 & \mathtt{capa}-4,\\ \mathtt{id}-2 & \mathtt{capa}-3,\\ \mathtt{id}-3 & \mathtt{capa}-5,\\ \mathtt{id}-4 & \mathtt{capa}-3,\\ \mathtt{id}-5 & \mathtt{capa}-3 \end{array} \right\rangle,\\ \left\langle \begin{array}{ll} \mathtt{bin}-3 & \mathtt{weight}-4,\\ \mathtt{bin}-1 & \mathtt{weight}-3,\\ \mathtt{bin}-3 & \mathtt{weight}-1 \end{array} \right\rangle \end{array}\right)$$

The bin_packing_capa constraint holds since the sum of the height of items that are assigned to bins 1 and 3 is respectively equal to 3 and 5. The previous quantities are respectively less than or equal to the maximum capacities 4 and 5 of bins 1 and 3. Figure 5.79 shows the solution associated with the example.

Typical

|BINS| > 1
range(BINS.capa) > 1
BINS.capa >maxval(ITEMS.weight)
BINS.capa ≤sum(ITEMS.weight)
|ITEMS| > 1
range(ITEMS.bin) > 1
range(ITEMS.weight) > 1
ITEMS.weight > 0

Symmetries

- Items of `BINS` are permutable.
- Items of `ITEMS` are permutable.
- `BINS.capa` can be increased.
- `ITEMS.weight` can be decreased to any value ≥ 0.
- All occurrences of two distinct values in `BINS.id` or `ITEMS.bin` can be swapped; all occurrences of a value in `BINS.id` or `ITEMS.bin` can be renamed to any unused value.

Systems

`pack` in **Choco**.

See also

generalisation: `indexed_sum` *(negative contribution also allowed)*.

specialisation: `bin_packing` *(non-fixed capacity replaced by fixed overall capacity)*.

Keywords

application area: assignment.

constraint type: predefined constraint, resource constraint.

modelling: assignment dimension, assignment to the same set of values.

modelling exercises: assignment to the same set of values.

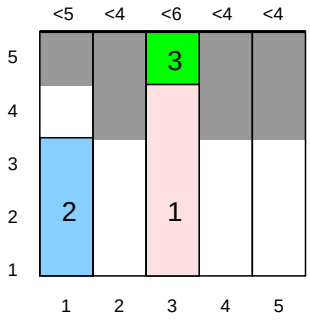

Figure 5.79: Bin-packing solution