

5.54 bin_packing_capa

	DESCRIPTION	LINKS
Origin	Derived from bin_packing .	
Constraint	<code>bin_packing_capa(BINS, ITEMS)</code>	
Arguments	<pre>BINS : collection(id-int, capa-int) ITEMS : collection(bin-dvar, weight-int)</pre>	
Restrictions	<pre> BINS > 0 required(BINS, [id, capa]) distinct(BINS, id) BINS.id ≥ 1 BINS.id ≤ BINS BINS.capa ≥ 0 required(ITEMS, [bin, weight]) in_attr(ITEMS, bin, BINS, id) ITEMS.weight ≥ 0</pre>	
Purpose	<p>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.</p>	
Example	$\left(\begin{array}{l} \text{id} - 1 \quad \text{capa} - 4, \\ \left\langle \begin{array}{l} \text{id} - 2 \quad \text{capa} - 3, \\ \text{id} - 3 \quad \text{capa} - 5, \\ \text{id} - 4 \quad \text{capa} - 3, \end{array} \right\rangle, \\ \text{id} - 5 \quad \text{capa} - 3 \\ \left\langle \begin{array}{l} \text{bin} - 3 \quad \text{weight} - 4, \\ \text{bin} - 1 \quad \text{weight} - 3, \\ \text{bin} - 3 \quad \text{weight} - 1 \end{array} \right\rangle \end{array} \right)$	
	<p>The <code>bin_packing_capa</code> 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 5.123 shows the solution associated with the example.</p>	
Typical	<pre> 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</pre>	

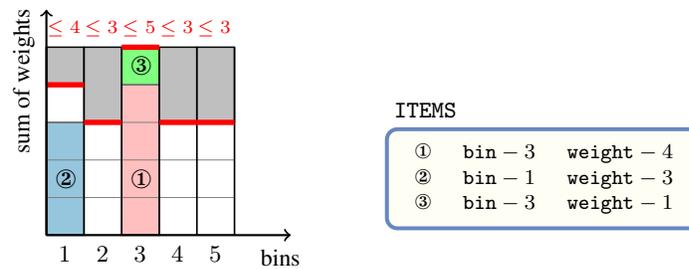


Figure 5.123: Bin-packing solution to the **Example** slot

Symmetries

- Items of BINS are [permutable](#).
- Items of ITEMS are [permutable](#).
- BINS.capacity 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.

Arg. properties

[Contractible](#) wrt. ITEMS.

Remark

In [MiniZinc](http://www.minizinc.org/) (<http://www.minizinc.org/>) there is also a constraint called `bin_packing_load` which, for each bin has a domain variable that is equal to the sum of the weights assigned to the corresponding bin.

Systems

`pack` in [Choco](#), `binpacking` in [Gecode](#), `bin_packing_capa` in [MiniZinc](#).

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.