5.192   indexed_sum

Origin   N. Beldiceanu

Constraint   `indexed_sum(ITEMS, TABLE)`

Arguments   ITEMS : collection(index−dvar, weight−dvar)
                    TABLE : collection(index−int, summation−dvar)

Restrictions $|\text{ITEMS}| > 0$
                    $|\text{TABLE}| > 0$
                    required(ITEMS, [index, weight])
                    $\text{ITEMS.index} \geq 1$
                    $\text{ITEMS.index} \leq |\text{TABLE}|$
                    required(TABLE, [index, summation])
                    $\text{TABLE.index} \geq 1$
                    $\text{TABLE.index} \leq |\text{TABLE}|$
                    increasing_seq(TABLE, index)

Purpose   Given several items of the collection ITEMS (each of them having a specific fixed index
                    as well as a weight that may be negative or positive), and a table TABLE (each entry of
                    TABLE corresponding to a summation variable), assign each item to an entry of TABLE
                    so that the sum of the weights of the items assigned to that entry is equal to the corre-
                    sponding summation variable.

Example
$$\left(\begin{array}{c} \left\langle \begin{array}{ll} \text{index} - 3 & \text{weight} - -4, \\ \text{index} - 1 & \text{weight} - 6, \\ \text{index} - 3 & \text{weight} - 1 \end{array} \right\rangle, \\ \left\langle \begin{array}{ll} \text{index} - 1 & \text{summation} - 6, \\ \text{index} - 2 & \text{summation} - 0, \\ \text{index} - 3 & \text{summation} - -3 \end{array} \right\rangle \end{array} \right)$$

The `indexed_sum` constraint holds since the summation variables associated with
each entry of TABLE are equal to the sum of the weights of the items assigned to the
corresponding entry:

- TABLE[1].summation = ITEMS[2].weight = 6 (since TABLE[1].index = ITEMS[2].index = 1),
- TABLE[2].summation = 0 (since TABLE[2].index = 2 does not occur as a value of the index attribute of an item of ITEMS),
- TABLE[3].summation = ITEMS[1].weight + ITEMS[3].weight = $-4 + 1 = -3$ (since TABLE[3].index = ITEMS[1].index = ITEMS[3].index = 3).

Typical $|\text{ITEMS}| > 1$
                    range(ITEMS.index) > 1
                    $|\text{TABLE}| > 1$
                    range(TABLE.summation) > 1

Symmetries

- Items of ITEMS are permutable.
- Items of TABLE are permutable.

Reformulation

The indexed_sum(ITEMS, TABLE) constraint can be expressed in term of a set of reified constraints and of |TABLE| arithmetic constraints (i.e., scalar_product constraints).

1. For each item ITEMS$[i]$ ($i \in [1, |\text{ITEMS}|]$) and for each table entry j ($j \in [1, |\text{TABLE}|]$) of TABLE we create a 0-1 variable B_{ij} that will be set to 1 if and only if ITEMS$[i]$.index is fixed to j (i.e., $B_{ij} \Leftrightarrow$ ITEMS$[i]$.index $= j$).

2. For each entry j of the table TABLE, we impose the sum ITEMS$[1]$.weight $\cdot B_{1j} +$ ITEMS$[2]$.weight $\cdot B_{2j} + \cdots +$ ITEMS$[|\text{ITEMS}|]$.weight $\cdot B_{|\text{ITEMS}|j}$ to be equal to TABLE$[j]$.summation.

See also

implied by: elements_alldifferent.

specialisation: bin_packing *(negative contribution not allowed, effective use variable for each bin replaced by an overall fixed capacity)*, bin_packing_capa *(negative contribution not allowed, effective use variable for each bin replaced by a fixed capacity for each bin)*.

used in graph description: sum_ctr.

Keywords

application area: assignment.

modelling: variable indexing, variable subscript.

For all items of `TABLE`:

Arc input(s)	`ITEMS TABLE`
Arc generator	$PRODUCT \mapsto \mathsf{collection}(\mathtt{items}, \mathtt{table})$
Arc arity	2
Arc constraint(s)	$\mathtt{items.index} = \mathtt{table.index}$
Sets	$\mathsf{SUCC} \mapsto$ $\left[\begin{array}{l} \mathtt{source,} \\ \mathtt{variables-col} \left(\begin{array}{l} \mathtt{VARIABLES-collection}(\mathtt{var-dvar}), \\ [\mathtt{item}(\mathtt{var-ITEMS.weight})] \end{array} \right) \end{array} \right]$
Constraint(s) on sets	$\mathsf{sum\_ctr}(\mathtt{variables}, =, \mathtt{TABLE.summation})$

Graph model   We enforce the `sum_ctr` constraint on the weight of the items that are assigned to the same entry. Within the context of the **Example** slot, part (A) of Figure 5.444 shows the initial graphs associated with entries 1, 2 and 3 (i.e., one initial graph for each item of the `TABLE` collection). Part (B) of Figure 5.444 shows the corresponding final graphs associated with entries 1 and 3. Each source vertex of the final graph can be interpreted as an item assigned to a specific entry of `TABLE`.
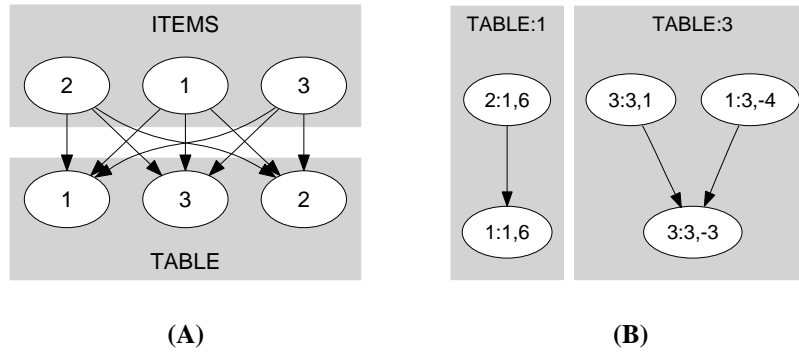

(A)   **(B)**

Figure 5.444: Initial and final graph of the `indexed_sum` constraint