

5.388 sum_of_weights_of_distinct_values

	DESCRIPTION	LINKS	GRAPH
Origin	[40]		
Constraint	<code>sum_of_weights_of_distinct_values(VARIABLES, VALUES, COST)</code>		
Synonym	<code>swdv</code> .		
Arguments	VARIABLES : <code>collection(var-dvar)</code> VALUES : <code>collection(val-int, weight-int)</code> COST : <code>dvar</code>		
Restrictions	<code>required(VARIABLES, var)</code> <code> VALUES > 0</code> <code>required(VALUES, [val, weight])</code> <code>VALUES.weight ≥ 0</code> <code>distinct(VALUES, val)</code> <code>in_attr(VARIABLES, var, VALUES, val)</code> <code>COST ≥ 0</code>		
Purpose	All variables of the VARIABLES collection take a value in the VALUES collection. In addition COST is the sum of the weight attributes associated with the distinct values taken by the variables of VARIABLES.		
Example	$\left(\left\langle \langle 1, 6, 1 \rangle, \left\langle \begin{array}{ll} \text{val} - 1 & \text{weight} - 5, \\ \text{val} - 2 & \text{weight} - 3, \\ \text{val} - 6 & \text{weight} - 7 \end{array} \right\rangle, 12 \right\rangle \right)$		
	The <code>sum_of_weights_of_distinct_values</code> constraint holds since its last argument <code>COST = 12</code> is equal to the sum $5 + 7$ of the weights of the values 1 and 6 that occur within the $\langle 1, 6, 1 \rangle$ collection.		
Typical	<code> VARIABLES > 1</code> <code>range(VARIABLES.var) > 1</code> <code> VALUES > 1</code> <code>range(VALUES.weight) > 1</code> <code>maxval(VALUES.weight) > 0</code>		
Symmetries	<ul style="list-style-type: none"> • Items of VARIABLES are permutable. • All occurrences of two distinct values of VARIABLES.var can be swapped. • Items of VALUES are permutable. • 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

Functional dependency: COST determined by VARIABLES and VALUES.

See also

attached to cost variant: `nvalue` (*all values have a weight of 1*).

common keyword: `global_cardinality_with_costs`,
`minimum_weight_alldifferent`, `weighted_partial_alldiff` (*weighted assignment*).

Keywords

application area: assignment.

constraint arguments: pure functional dependency.

constraint type: relaxation.

filtering: cost filtering constraint.

modelling: functional dependency.

problems: domination, weighted assignment, facilities location problem.

Arc input(s)	VARIABLES VALUES
Arc generator	<i>PRODUCT</i> \mapsto <code>collection(variables, values)</code>
Arc arity	2
Arc constraint(s)	<code>variables.var = values.val</code>
Graph property(ies)	<ul style="list-style-type: none"> • NSOURCE = VARIABLES • SUM(VALUES, weight) = COST

Signature

Since we use the *PRODUCT* arc generator, the number of sources of the final graph cannot exceed the number of sources of the initial graph. Since the initial graph contains |VARIABLES| sources, this number is an upper bound of the number of sources of the final graph. Therefore we can rewrite **NSOURCE** = |VARIABLES| to **NSOURCE** \geq |VARIABLES| and simplify NSOURCE to **NSOURCE**.

Parts (A) and (B) of Figure 5.747 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NSOURCE** graph property, the source vertices of the final graph are shown in a double circle. Since we also use the **SUM** graph property we show the vertices from which we compute the total cost in a box.

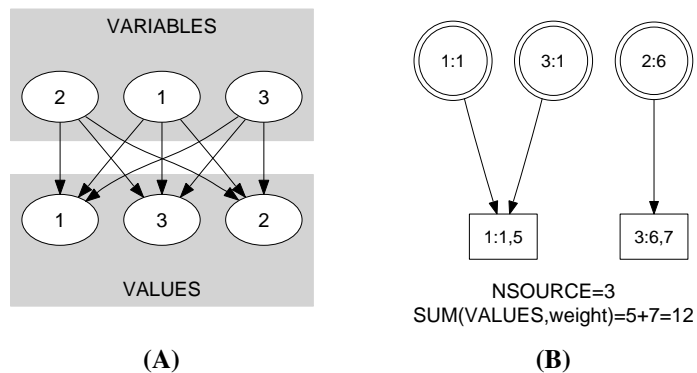


Figure 5.747: Initial and final graph of the `sum_of_weights_of_distinct_values` constraint

20030820

2265