5.420 weighted_partial_alldiff

DESCRIPTION LINKS GRAPH

Origin [406, page 71]

Constraint `weighted_partial_alldiff(VARIABLES, UNDEFINED, VALUES, COST)`

Synonyms `weighted_partial_alldifferent`, `weighted_partial_alldistinct`, `wpa`.

Arguments

```
VARIABLES  : collection(var−dvar)
UNDEFINED  : int
VALUES     : collection(val−int, weight−int)
COST       : dvar
```

Restrictions

```
required(VARIABLES, var)
|VALUES| > 0
required(VALUES, [val, weight])
in_attr(VARIABLES, var, VALUES, val)
distinct(VALUES, val)
```

Purpose All variables of the `VARIABLES` collection that are not assigned to value `UNDEFINED` must have pairwise distinct values from the `val` attribute of the `VALUES` collection. In addition `COST` is the sum of the `weight` attributes associated with the values assigned to the variables of `VARIABLES`. Within the `VALUES` collection, value `UNDEFINED` must be explicitly defined with a weight of 0.

Example

$$\left(\begin{array}{l}\langle 4,0,1,2,0,0\rangle, 0,\\ \left\langle\begin{array}{ll}\texttt{val}-0 & \texttt{weight}-0,\\ \texttt{val}-1 & \texttt{weight}-2,\\ \texttt{val}-2 & \texttt{weight}--1,\\ \texttt{val}-4 & \texttt{weight}-7,\\ \texttt{val}-5 & \texttt{weight}--8,\\ \texttt{val}-6 & \texttt{weight}-2\end{array}\right\rangle, 8\end{array}\right)$$

The `weighted_partial_alldiff` constraint holds since:

- No value, except value `UNDEFINED` $= 0$, is used more than once.
- `COST` $= 8$ is equal to the sum of the weights 2, -1 and 7 of the values 1, 2 and 4 assigned to the variables of `VARIABLES` $= \langle 4, 0, 1, 2, 0, 0\rangle$.

Typical

```
|VARIABLES| > 0
atleast(1, VARIABLES, UNDEFINED)
|VARIABLES| ≤ |VALUES| + 2
```

Symmetries

- Items of VARIABLES are permutable.
- Items of VALUES are permutable.
- All occurrences of two distinct values in VARIABLES.var or VALUES.val that are both different from UNDEFINED can be swapped; all occurrences of a value in VARIABLES.var or VALUES.val that is different from UNDEFINED can be renamed to any unused value that is also different from UNDEFINED.

Arg. properties

Functional dependency: COST determined by VARIABLES and VALUES.

Usage

In his PhD thesis [406, pages 71–72], Sven Thiel describes the following three potential scenarios of the weighted_partial_alldiff constraint:

- Given a set of tasks (i.e., the items of the VARIABLES collection), assign to each task a resource (i.e., an item of the VALUES collection). Except for the resource associated with value UNDEFINED, every resource can be used at most once. The cost of a resource is independent from the task to which the resource is assigned. The cost of value UNDEFINED is equal to 0. The total cost COST of an assignment corresponds to the sum of the costs of the resources effectively assigned to the tasks. Finally we impose an upper bound on the total cost.
- Given a set of persons (i.e., the items of the VARIABLES collection), select for each person an offer (i.e., an item of the VALUES collection). Except for the offer associated with value UNDEFINED, every offer should be selected at most once. The profit associated with an offer is independent from the person that selects the offer. The profit of value UNDEFINED is equal to 0. The total benefit COST is equal to the sum of the profits of the offers effectively selected. In addition we impose a lower bound on the total benefit.
- The last scenario deals with an application to an over-constraint problem involving the alldifferent constraint. Allowing some variables to take an ”undefined” value is done by setting all weights of all the values different from UNDEFINED to 1. As a consequence all variables assigned to a value different from UNDEFINED will have to take distinct values. The COST variable allows to control the number of such variables.

Remark

It was shown in [406, page 104] that, finding out whether the weighted_partial_alldiff constraint has a solution or not is NP-hard. This was achieved by reduction from subset sum.

Algorithm

A filtering algorithm is given in [406, pages 73–104]. After showing that, deciding whether the weighted_partial_alldiff has a solution is NP-complete, [406, pages 105–106] gives the following results of his filtering algorithm with respect to consistency under the 3 scenarios previously described:

- For scenario 1, if there is no restriction of the lower bound of the COST variable, the filtering algorithm achieves arc-consistency for all variables of the VARIABLES collection (but not for the COST variable itself).
- For scenario 2, if there is no restriction of the upper bound of the COST variable, the filtering algorithm achieves arc-consistency for all variables of the VARIABLES collection (but not for the COST variable itself).

- Finally, for scenario 3, the filtering algorithm achieves arc-consistency for all variables of the `VARIABLES` collection as well as for the `COST` variable.

See also

attached to cost variant: `alldifferent`, `alldifferent_except_0`.

common keyword: `global_cardinality_with_costs` (*weighted assignment*), `minimum_weight_alldifferent` (*cost filtering constraint*,*weighted assignment*), `soft_alldifferent_var` (*soft constraint*), `sum_of_weights_of_distinct_values` (*weighted assignment*).

Keywords

application area: assignment.

characteristic of a constraint: all different, joker value.

complexity: subset sum.

constraint type: soft constraint, relaxation.

filtering: cost filtering constraint.

modelling: functional dependency.

problems: weighted assignment.

Arc input(s)	`VARIABLES VALUES`
Arc generator	*PRODUCT*$\mapsto$`collection(variables, values)`
Arc arity	2
Arc constraint(s)	• `variables.var` $\neq$ `UNDEFINED` • `variables.var` $=$ `values.val`
Graph property(ies)	• **MAX_ID**≤ 1 • **SUM**(`VALUES`, `weight`) $=$ `COST`

Graph model Parts (A) and (B) of Figure 5.804 respectively show the initial and final graph associated with the **Example** slot. Since we also use the **SUM** graph property we show the vertices of the final graph from which we compute the total cost in a box.

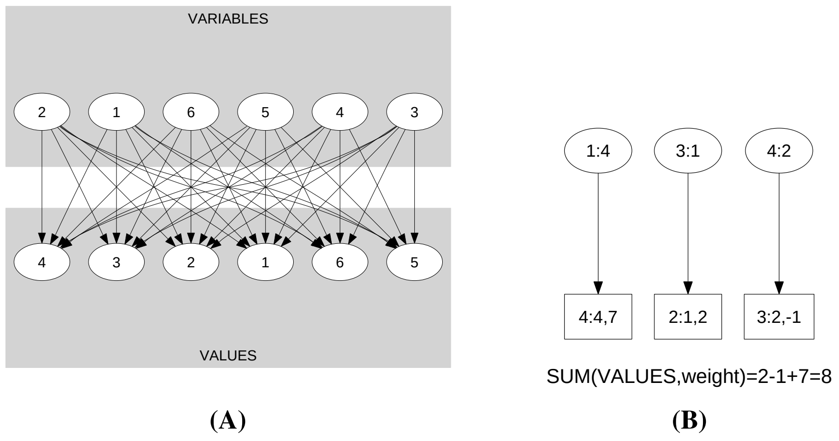

Figure 5.804: Initial and final graph of the `weighted_partial_alldiff` constraint