5.4 all_min_dist

DESCRIPTION LINKS GRAPH

Origin [310]

Constraint all_min_dist(MINDIST, VARIABLES)

Synonyms minimum_distance, inter_distance.

Arguments

MINDIST	:	int
VARIABLES	:	collection(var−dvar)

Restrictions

MINDIST >0
required(VARIABLES, var)
VARIABLES.var ≥ 0

Purpose Enforce for each pair $(\texttt{var}_i, \texttt{var}_j)$ of distinct variables of the collection VARIABLES that $|\texttt{var}_i - \texttt{var}_j| \geq$ MINDIST.

Example $(2, \langle 5, 1, 9, 3\rangle)$

The all_min_dist constraint holds since the following expressions $|5-1|$, $|5-9|$, $|5-3|$, $|1-9|$, $|1-3|$, $|9-3|$ are all greater than or equal to the first argument MINDIST $=2$ of the all_min_dist constraint.

Typical

MINDIST >1
$|$VARIABLES$|>1$

Symmetries

- MINDIST can be decreased to any value ≥ 1.
- Items of VARIABLES are permutable.
- Two distinct values of VARIABLES.var can be swapped.
- One and the same constant can be added to the var attribute of all items of VARIABLES.

Usage The all_min_dist constraint was initially created for handling frequency allocation problems. In [10] it is used for scheduling tasks that all have the same fixed duration in the context of air traffic management in the terminal radar control area of airports.

Remark The all_min_dist constraint can be modelled as a set of tasks that should not overlap. For each variable var of the VARIABLES collection we create a task t where var and MINDIST respectively correspond to the origin and the duration of t.

Some solvers use in a pre-processing phase, while stating constraints of the form $|X_i - X_j| \geq D_{ij}$ (where X_i and X_j are domain variables and D_{ij} is a constant), an algorithm for automatically extracting large cliques [79] from such inequalities in order to state all_min_dist constraints.

Algorithm K. Artiouchine and P. Baptiste came up with a cubic time complexity algorithm achieving bound-consistency in [10, 11] based on the adaptation of a feasibility test algorithm from M.R. Garey *et al.* [164]. Later on, C.-G. Quimper *et al.*, proposed a quadratic algorithm achieving the same level of consistency in [299].

See also **generalisation:** `diffn` *(*`line segment`*, of same length, replaced by orthotope)*, `disjunctive` *(*`line segment`*, of same length, replaced by* `line segment`*)*.

implies: `alldifferent_interval`.

related: `distance`.

specialisation: `alldifferent` *(*`line segment`*, of same length, replaced by* `variable`*)*.

Keywords **application area:** frequency allocation problem, air traffic management.

constraint type: value constraint, decomposition, scheduling constraint.

filtering: bound-consistency.

final graph structure: acyclic.

problems: maximum clique.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE(<) \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	abs(variables1.var − variables2.var) ≥ MINDIST
Graph property(ies)	NARC= |VARIABLES| ∗ (|VARIABLES| − 1)/2
Graph class	• ACYCLIC • NO_LOOP

Graph model We generate a *clique* with a minimum distance constraint between each pair of distinct vertices and state that the number of arcs of the final graph should be equal to the number of arcs of the initial graph.

Parts (A) and (B) of Figure 5.3 respectively show the initial and final graph associated with the **Example** slot. The all_min_dist constraint holds since all the arcs of the initial graph belong to the final graph: all the minimum distance constraints are satisfied.

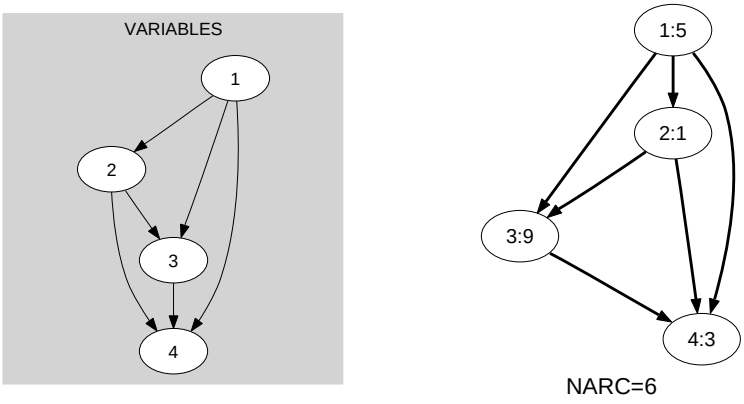

Figure 5.3: Initial and final graph of the all_min_dist constraint