5.11 all_min_dist

DESCRIPTION **LINKS** **GRAPH**

Origin [343]

Constraint `all_min_dist(MINDIST, VARIABLES)`

Synonyms `minimum_distance`, `inter_distance`.

Arguments

```
MINDIST   : int
VARIABLES : collection(var-dvar)
```

Restrictions

`MINDIST` > 0
$|$`VARIABLES`$| < 2 \vee$ `MINDIST` $<$ `range(VARIABLES.var)`
`required(VARIABLES, var)`

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

All solutions Figure 5.23 gives all solutions to the following non ground instance of the `all_min_dist` constraint: $V_1 \in [0, 5]$, $V_2 \in [3, 9]$, $V_3 \in [5, 7]$, $V_4 \in [2, 10]$, `all_min_dist`$(\mathbf{3}, \langle V_1, V_2, V_3, V_4 \rangle)$.

① $(\mathbf{3}, \langle 0, 3, 6, 9 \rangle)$
② $(\mathbf{3}, \langle 0, 3, 6, 10 \rangle)$
③ $(\mathbf{3}, \langle 0, 3, 7, 10 \rangle)$
④ $(\mathbf{3}, \langle 0, 4, 7, 10 \rangle)$
⑤ $(\mathbf{3}, \langle 0, 9, 6, 3 \rangle)$
⑥ $(\mathbf{3}, \langle 1, 4, 7, 10 \rangle)$

Figure 5.23: All solutions corresponding to the non ground example of the `all_min_dist` constraint of the **All solutions** slot

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.

Arg. properties

Contractible wrt. VARIABLES.

Usage

The all_min_dist constraint was initially created for handling frequency allocation problems. In [11] 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 [88] from such inequalities in order to state all_min_dist constraints.

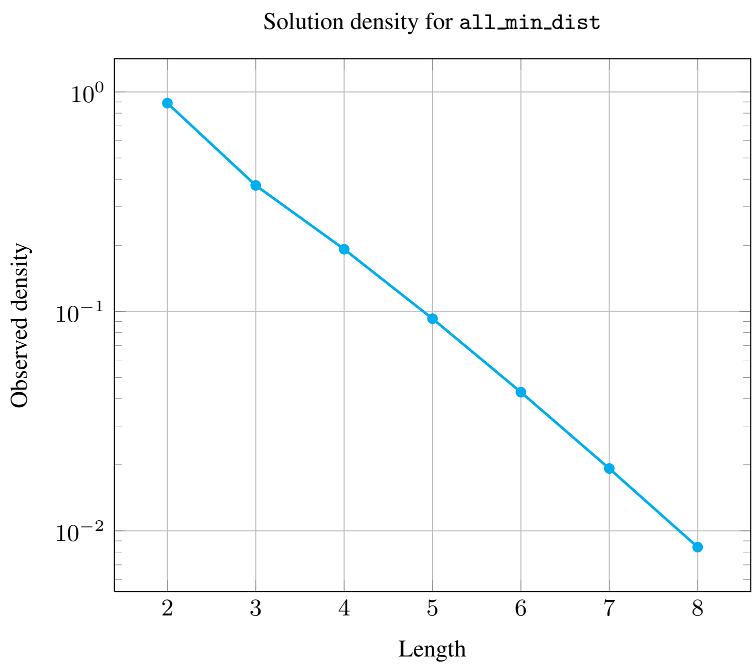
Algorithm

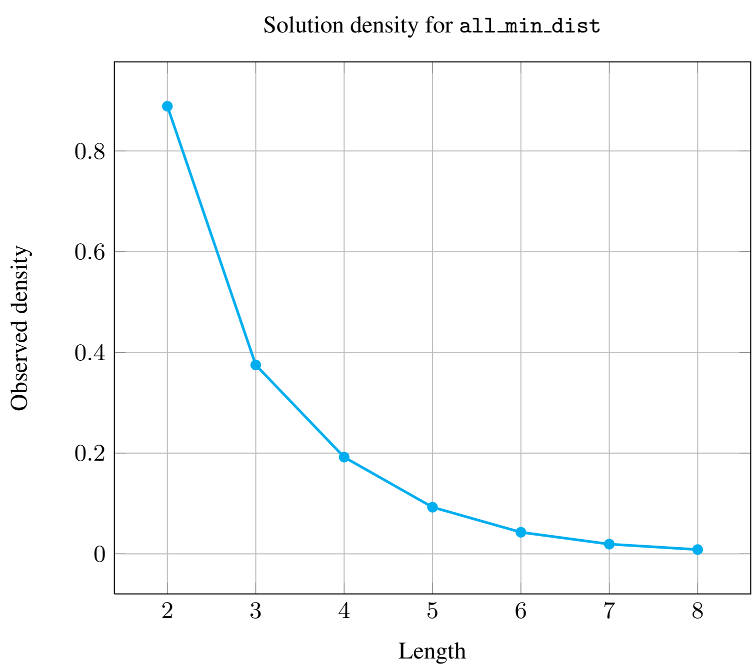
K. Artiouchine and P. Baptiste came up with a cubic time complexity algorithm achieving bound-consistency in [11, 12] based on the adaptation of a feasibility test algorithm from M.R. Garey *et al.* [185]. Later on, C.-G. Quimper *et al.*, proposed a quadratic algorithm achieving the same level of consistency in [332].

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	8	24	120	720	5040	40320	362880

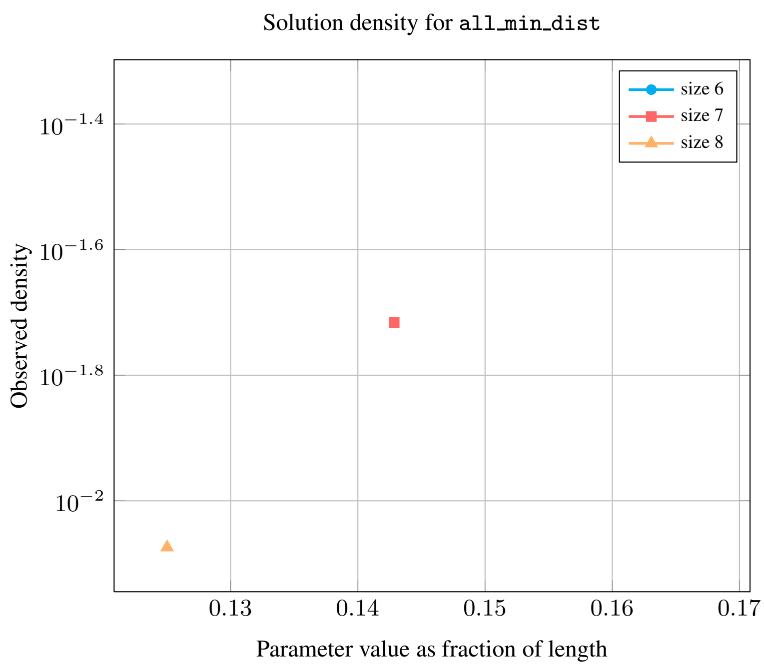
Number of solutions for all_min_dist: domains $0..n$

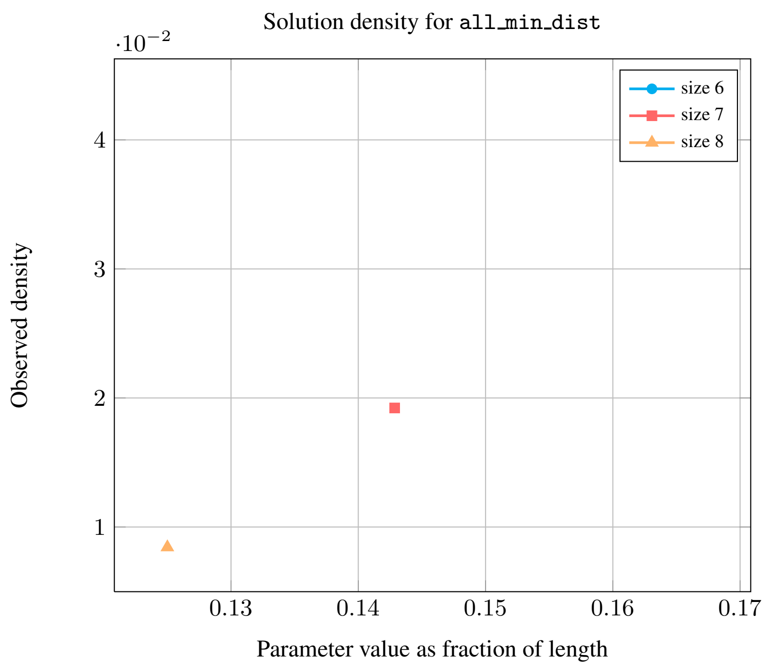

Solution density for all_min_dist
Observed density
10^0
10^{-1}
10^{-2}
2 3 4 5 6 7 8
Length


Solution density for all_min_dist
Observed density
0.8
0.6
0.4
0.2
0
2 3 4 5 6 7 8
Length

Length (n)		2	3	4	5	6	7	8
Total		8	24	120	720	5040	40320	362880
Parameter value	1	6	24	120	720	5040	40320	362880
	2	2	-	-	-	-	-	-

Solution count for `all_min_dist`: domains $0..n$





See also

generalisation: `diffn` *(*`line segment`*, of same length, replaced by orthotope)*, `disjunctive` *(*`line segment`*, of same length, replaced by* `line segment`*)*, `multi_inter_distance` *(*`LIMIT` *parameter introduced to specify capacity* ≥ 1*)*.

implies: `alldifferent_interval`, `soft_alldifferent_var`.

related: `distance`.

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

Keywords

application area: frequency allocation problem, air traffic management.

characteristic of a constraint: sort based reformulation.

constraint type: value constraint, decomposition, scheduling constraint.

filtering: bound-consistency.

final graph structure: acyclic.

problems: maximum clique.

Cond. implications

`all_min_dist`(`MINDIST`, `VARIABLES`)
implies `soft_all_equal_max_var`(`N`, `VARIABLES`)
when $\mathtt{N} \geq |\mathtt{VARIABLES}| - 1$.

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 5.24 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.

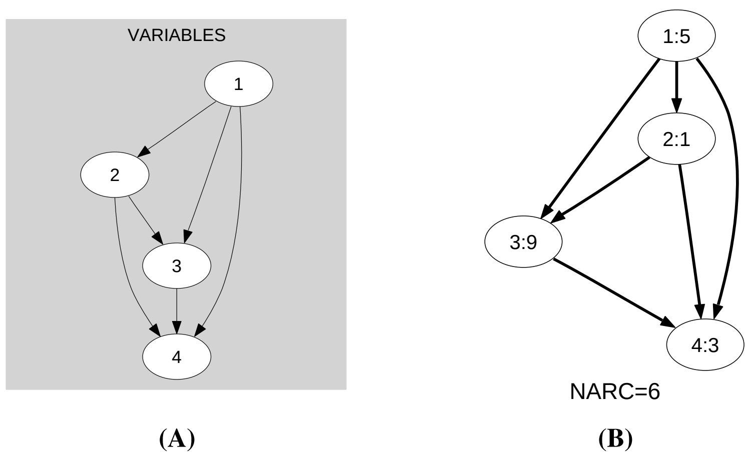


Figure 5.24: Initial and final graph of the `all_min_dist` constraint