5.360 soft_alldifferent_var

DESCRIPTION LINKS GRAPH

Origin [314]

Constraint soft_alldifferent_var(C, VARIABLES)

Synonyms soft_alldiff_var, soft_alldistinct_var, soft_alldiff_min_var, soft_alldifferent_min_var, soft_alldistinct_min_var.

Arguments

C	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

C ≥ 0
required(VARIABLES, var)

Purpose C is greater than or equal to the minimum number of variables of the collection VARIABLES for which the value needs to be changed in order that all variables of VARIABLES take a distinct value.

Example

$(3, \langle 5, 1, 9, 1, 5, 5\rangle)$
$(1, \langle 5, 1, 9, 6, 5, 3\rangle)$
$(0, \langle 8, 1, 9, 6, 5, 3\rangle)$

Within the collection $\langle 5, 1, 9, 1, 5, 5\rangle$ of the first example, 3 and 2 items are respectively fixed to values 5 and 1. Therefore one must change the values of at least $(3 - 1) + (2 - 1) = 3$ items to get back to 6 distinct values. Consequently, the corresponding soft_alldifferent_var constraint holds since its first argument C is greater than or equal to 3.

Typical

C > 0
2 ∗ C ≤ |VARIABLES|
|VARIABLES| > 1
some_equal(VARIABLES)

Symmetries

- C can be increased.
- Items of VARIABLES are permutable.
- All occurrences of two distinct values of VARIABLES.var can be swapped; all occurrences of a value of VARIABLES.var can be renamed to any unused value.

Arg. properties

Contractible wrt. VARIABLES.

Usage A soft alldifferent constraint.

Remark

Since it focus on the soft aspect of the `alldifferent` constraint, the original article [314], which introduce this constraint, describes how to evaluate the minimum value of `C` and how to prune according to the maximum value of `C`.

The `soft_alldifferent_var` constraint is called `soft_alldiff_min_var` in [149].

Algorithm

A first filtering algorithm presented in [314] achieves arc-consistency. A second filtering algorithm also achieving arc-consistency is described in [129, 130].

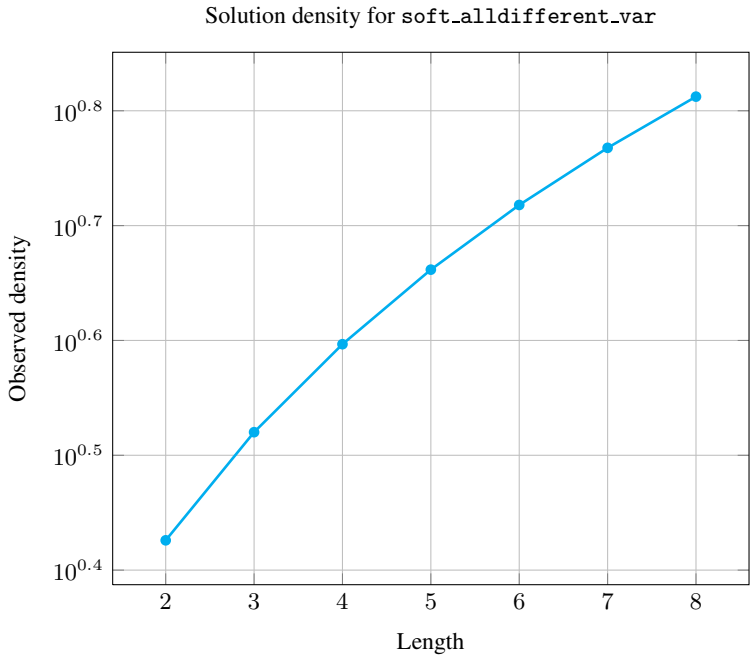
Reformulation

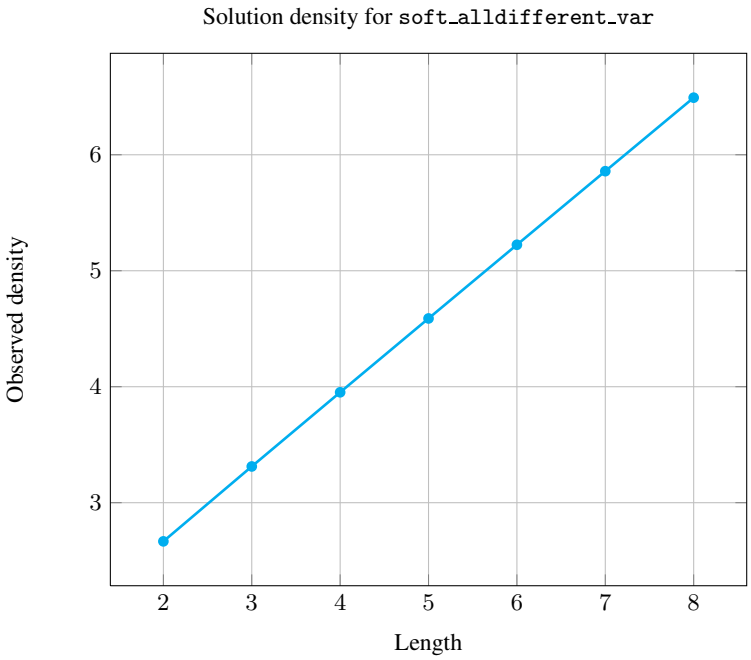
By introducing a variable M that gives the number of distinct values used by variables of the collection `VARIABLES`, the `soft_alldifferent_var`(`C`, `VARIABLES`) constraint can be expressed as a conjunction of the `nvalue`(M, `VARIABLES`) constraint and of the linear constraint `C` $\geq |$`VARIABLES`$| - M$.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	24	212	2470	35682	614600	12286024	279472266

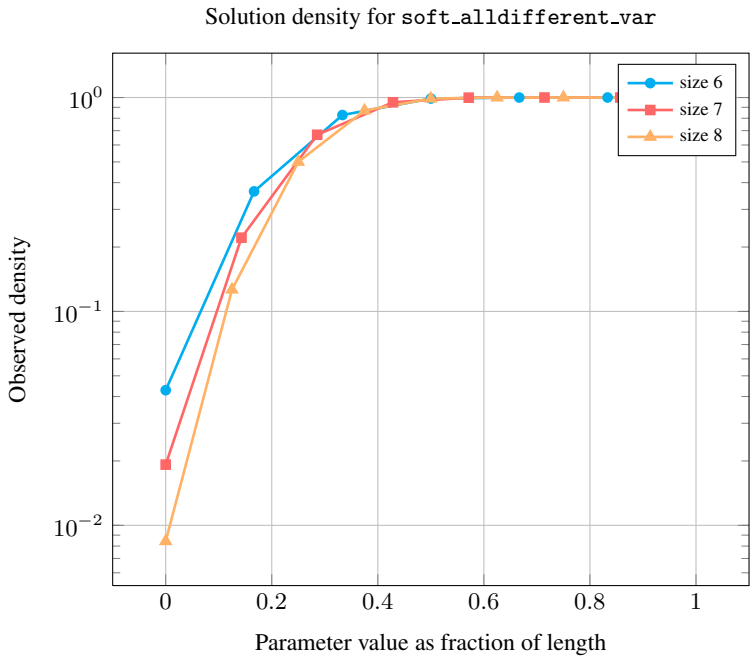
Number of solutions for `soft_alldifferent_var`: domains $0..n$

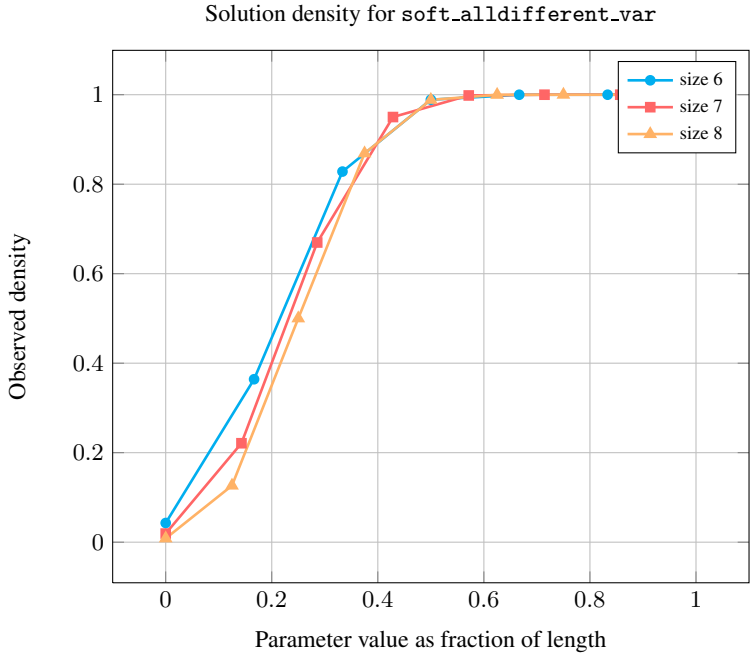




Length (n)		2	3	4	5	6	7	8
Total		24	212	2470	35682	614600	12286024	279472266
Parameter value	0	6	24	120	720	5040	40320	362880
	1	9	60	480	4320	42840	463680	5443200
	2	9	64	620	7320	97440	1404480	21530880
	3	-	64	625	7770	116340	1992480	37406880
	4	-	-	625	7776	117642	2093616	42550704
	5	-	-	-	7776	117649	2097144	43037568
	6	-	-	-	-	117649	2097152	43046712
	7	-	-	-	-	-	2097152	43046721
	8	-	-	-	-	-	-	43046721

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





See also **common keyword:** soft_all_equal_max_var, soft_all_equal_min_ctr, soft_all_equal_min_var, soft_alldifferent_ctr, weighted_partial_alldiff (*soft constraint*).

hard version: alldifferent.

implied by: all_min_dist, alldifferent_modulo, soft_alldifferent_ctr.

related: atmost_nvalue, nvalue.

Keywords

characteristic of a constraint: all different, disequality.

constraint type: soft constraint, value constraint, relaxation, variable-based violation measure.

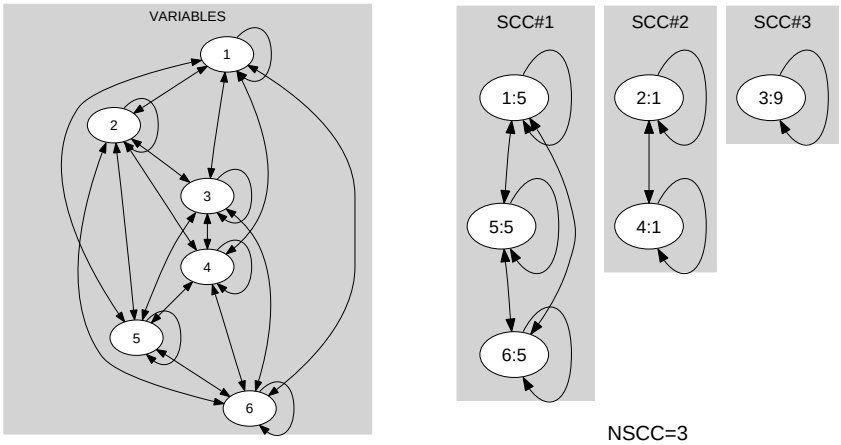
filtering: bipartite matching.

final graph structure: strongly connected component, equivalence.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	NSCC$\geq$ |VARIABLES| $-$ C

Graph model We generate a clique with binary *equalities* constraints between each pairs of vertices (this include an arc between a vertex and itself) and we state that C is equal to the difference between the total number of variables and the number of strongly connected components.

Parts (A) and (B) of Figure 5.702 respectively show the initial and final graph associated with the first example of the **Example** slot. Since we use the **NSCC** graph property we show the different strongly connected components of the final graph. Each strongly connected component of the final graph includes all variables that take the same value. Since we have 6 variables and 3 strongly connected components the *cost* variable C is greater than or equal to $6 - 3$.



(A) **(B)**

Figure 5.702: Initial and final graph of the soft_alldifferent_var constraint