5.41 atmost_nvalue

DESCRIPTION LINKS GRAPH

Origin [62]

Constraint atmost_nvalue(NVAL, VARIABLES)

Synonyms soft_alldiff_max_var, soft_alldifferent_max_var, soft_alldistinct_max_var.

Arguments

NVAL	:	dvar
VARIABLES	:	collection(var−dvar)

Restrictions

NVAL $\geq$ min(1, |VARIABLES|)
required(VARIABLES, var)

Purpose The number of distinct values taken by the variables of the collection VARIABLES is less than or equal to NVAL.

Example

$(4, \langle 3, 1, 3, 1, 6\rangle)$
$(3, \langle 3, 1, 3, 1, 6\rangle)$
$(1, \langle 3, 3, 3, 3, 3\rangle)$

The first atmost_nvalue constraint holds since the collection $\langle 3, 1, 3, 1, 6\rangle$ involves at most 4 distinct values (i.e., in fact 3 distinct values).

Typical

NVAL > 1
NVAL $<$ |VARIABLES|
|VARIABLES| > 1

Symmetries

- NVAL 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.
- An occurrence of a value of VARIABLES.var can be replaced by any value of VARIABLES.var.

Arg. properties

Contractible wrt. VARIABLES.

Remark This constraint was introduced together with the atleast_nvalue constraint by C. Bessière *et al.* in an article [62] providing filtering algorithms for the nvalue constraint.

It was shown in [69] that, finding out whether a atmost_nvalue constraint has a solution or not is NP-hard. This was achieved by reduction from 3-SAT.

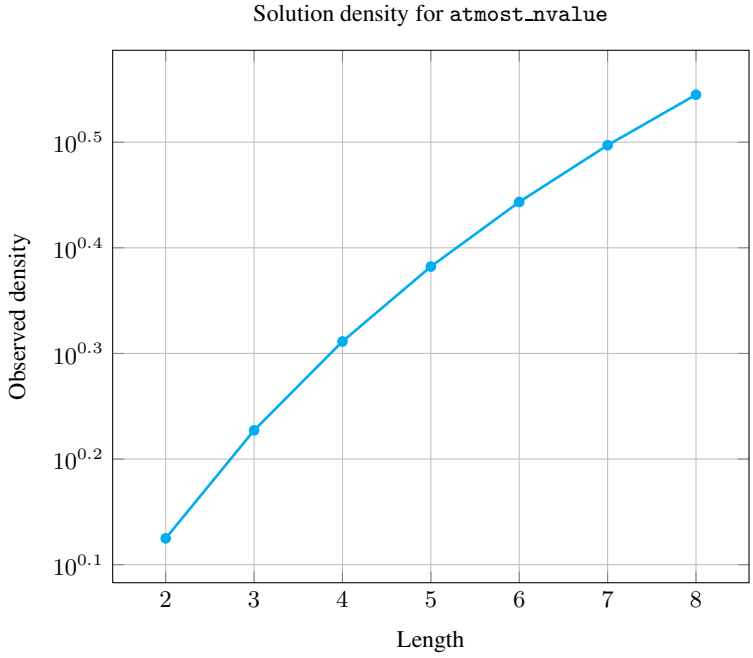
Algorithm

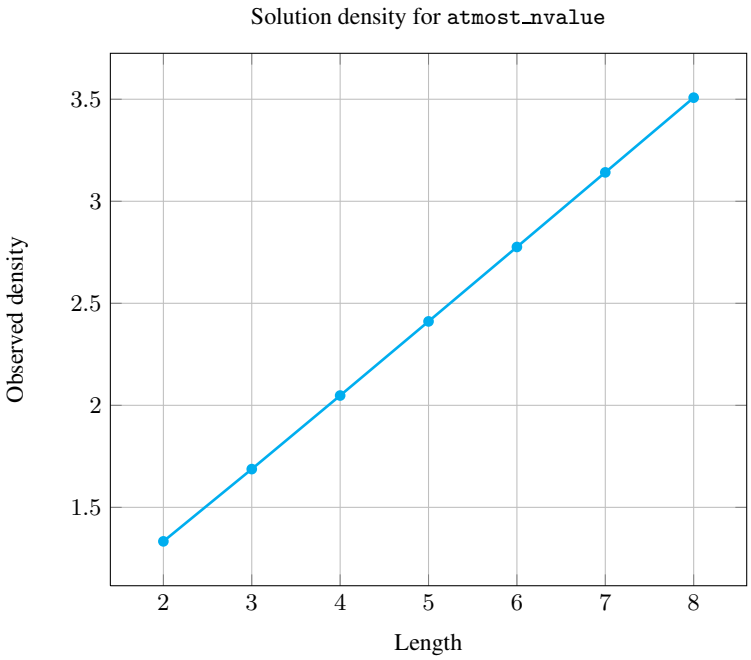
[27] provides an algorithm that achieves bound consistency. [40] provides two filtering algorithms, while [62] provides a greedy algorithm and a graph invariant for evaluating the minimum number of distinct values. [62] also gives a linear relaxation for approximating the minimum number of distinct values.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	12	108	1280	18750	326592	6588344	150994944

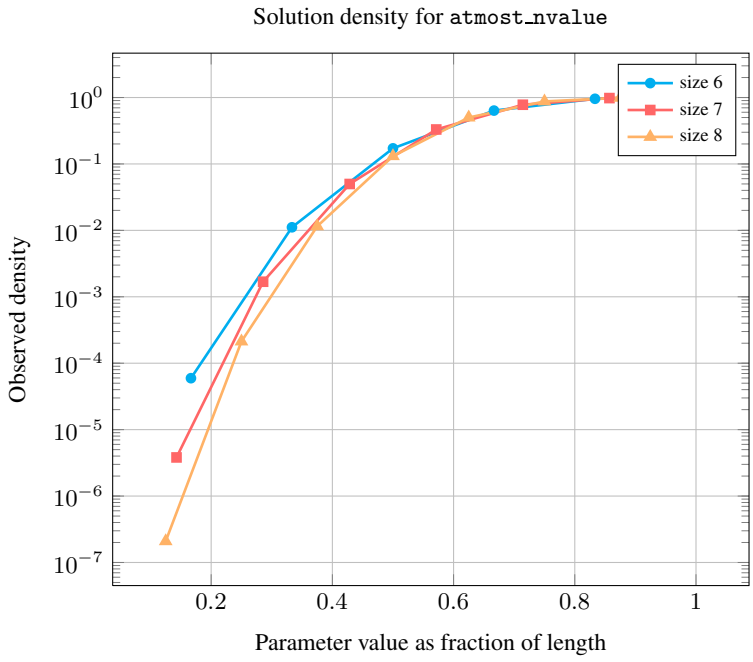
Number of solutions for `atmost_nvalue`: domains $0..n$

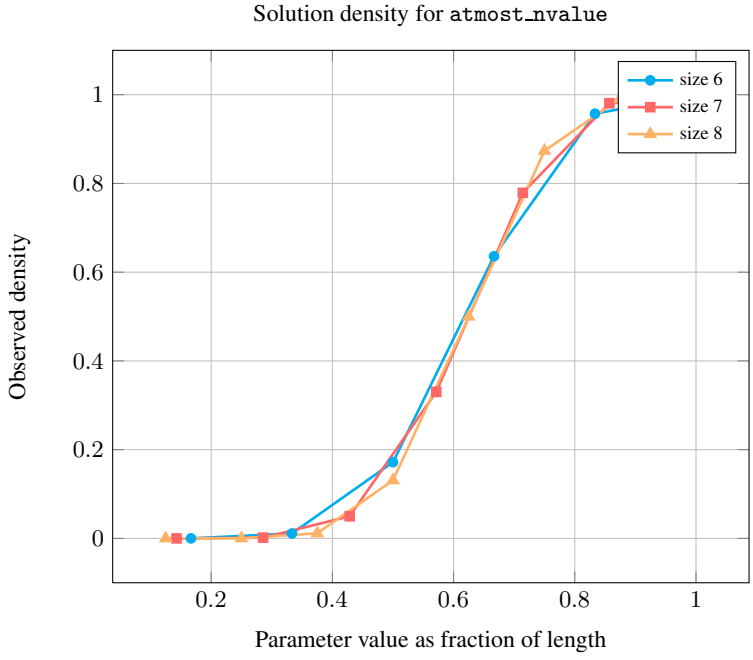




Length (n)		2	3	4	5	6	7	8
Total		12	108	1280	18750	326592	6588344	150994944
Parameter value	1	3	4	5	6	7	8	9
	2	9	40	145	456	1309	3536	9153
	3	-	64	505	3456	20209	104672	496017
	4	-	-	625	7056	74809	692672	5639841
	5	-	-	-	7776	112609	1633472	21515841
	6	-	-	-	-	117649	2056832	37603521
	7	-	-	-	-	-	2097152	42683841
	8	-	-	-	-	-	-	43046721

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





Systems atMostNValue in **Choco**.

See also **comparison swapped:** `atleast_nvalue`.

implied by: nvalue ($\leq$ NVAL *replaced by* $=$ NVAL).

related: soft_all_equal_max_var, soft_all_equal_min_ctr, soft_all_equal_min_var, soft_alldifferent_ctr, soft_alldifferent_var.

Keywords

complexity: 3-SAT.

constraint type: counting constraint, value partitioning constraint.

filtering: bound-consistency.

final graph structure: strongly connected component, equivalence.

modelling: number of distinct equivalence classes, number of distinct values.

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** $\leq$ NVAL
Graph class	EQUIVALENCE

Graph model Parts (A) and (B) of Figure 5.95 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 corresponds to a specific value that is assigned to some variables of the VARIABLES collection. The 3 following values 1, 3 and 6 are used by the variables of the VARIABLES collection.

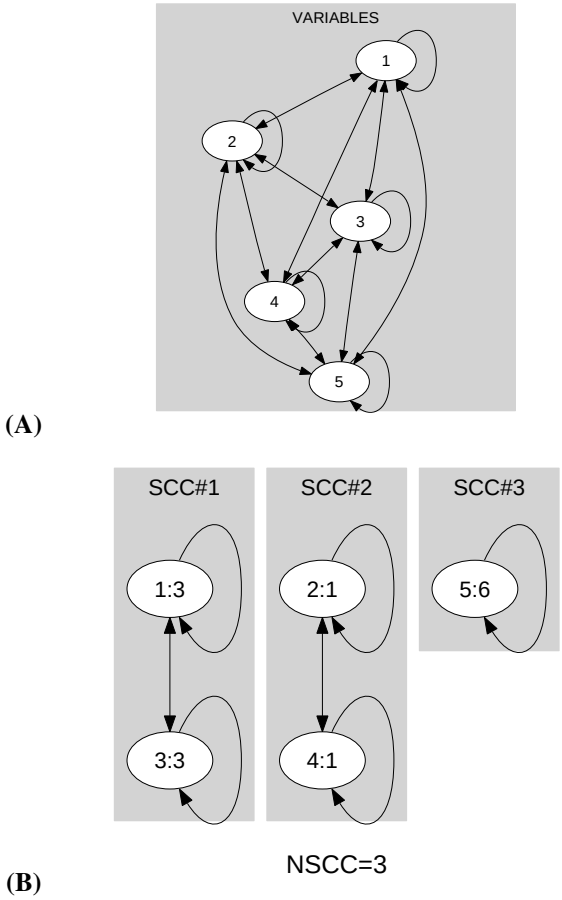


Figure 5.95: Initial and final graph of the `atmost_nvalue` constraint