

5.34 atmost_nvalue

	DESCRIPTION	LINKS	GRAPH
Origin	[56]		
Constraint	atmost_nvalue(NVAL, VARIABLES)		
Synonyms	soft_alldiff_max_var, soft_alldistinct_max_var.	soft_alldifferent_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	$(4, \langle 3, 1, 3, 1, 6 \rangle)$		
	The 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	<ul style="list-style-type: none"> • 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. 		
Remark	This constraint was introduced together with the atleast_nvalue constraint by C. Bessière <i>et al.</i> in a article [56] providing filtering algorithms for the nvalue constraint. It was shown in [63] 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	[26] provides an algorithm that achieves bound consistency . [36] provides two filtering algorithms, while [56] provides a greedy algorithm and a graph invariant for evaluating the minimum number of distinct values. [56] also gives a linear relaxation for approximating the minimum number of distinct values.		

Systems	<code>atMostNValue</code> in Choco .
See also	comparison swapped: <code>atleast_nvalue</code> . implied by: <code>nvalue</code> (\leq NVAL replaced by = NVAL). related: <code>soft_all_equal_max_var</code> , <code>soft_all_equal_min_ctr</code> , <code>soft_all_equal_min_var</code> , <code>soft_alldifferent_ctr</code> , <code>soft_alldifferent_var</code> .
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	<code>CLIQUE</code> \mapsto <code>collection</code> (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 5.63 respectively show the initial and final graph associated with 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.

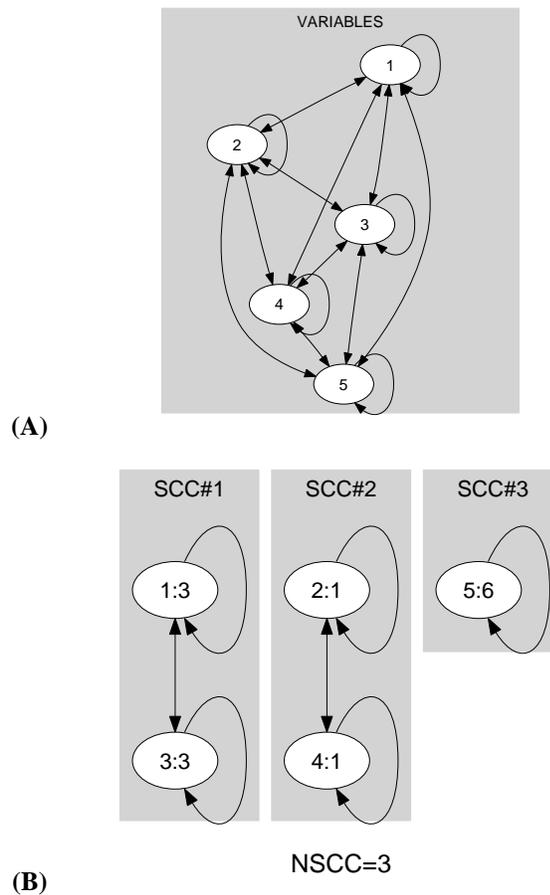


Figure 5.63: Initial and final graph of the `atmost_nvalue` constraint

