

5.359 soft_alldifferent_ctr

	DESCRIPTION	LINKS	GRAPH
Origin	[314]		
Constraint	soft_alldifferent_ctr(C, VARIABLES)		
Synonyms	soft_alldiff_ctr, soft_alldistinct_ctr, soft_alldiff_min_ctr, soft_alldifferent_min_ctr, soft_alldistinct_min_ctr, soft_all_equal_max_ctr.		
Arguments	C : dvar VARIABLES : collection(var-dvar)		
Restrictions	C ≥ 0 required(VARIABLES, var)		
Purpose	Consider the <i>disequality</i> constraints involving two distinct variables VARIABLES[i].var and VARIABLES[j].var ($i < j$) of the collection VARIABLES. Among the previous set of constraints, C is greater than or equal to the number of <i>disequality</i> constraints that do not hold.		
Example	(4, (5, 1, 9, 1, 5, 5)) (1, (5, 1, 9, 1, 2, 6)) (0, (5, 1, 9, 0, 2, 6))		
	Within the collection (5, 1, 9, 1, 5, 5) the first and fifth values, the first and sixth values, the second and fourth values, and the fifth and sixth values are identical. Consequently, the argument C = 4 is greater than or equal to the number of <i>disequality</i> constraints that do not hold (i.e. 4) and the soft_alldifferent_ctr constraint holds.		
Typical	C > 0 C ≤ VARIABLES * (VARIABLES - 1)/2 VARIABLES > 1		
Symmetries	<ul style="list-style-type: none"> • 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	The soft_alldifferent_ctr constraint is called soft_alldiff_min_ctr or soft_all_equal_max_ctr in [149].		

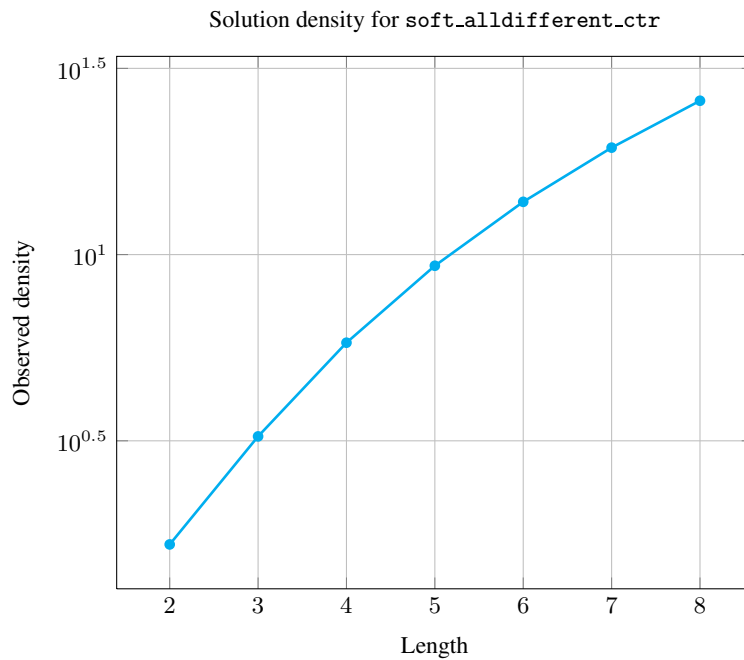
Algorithm

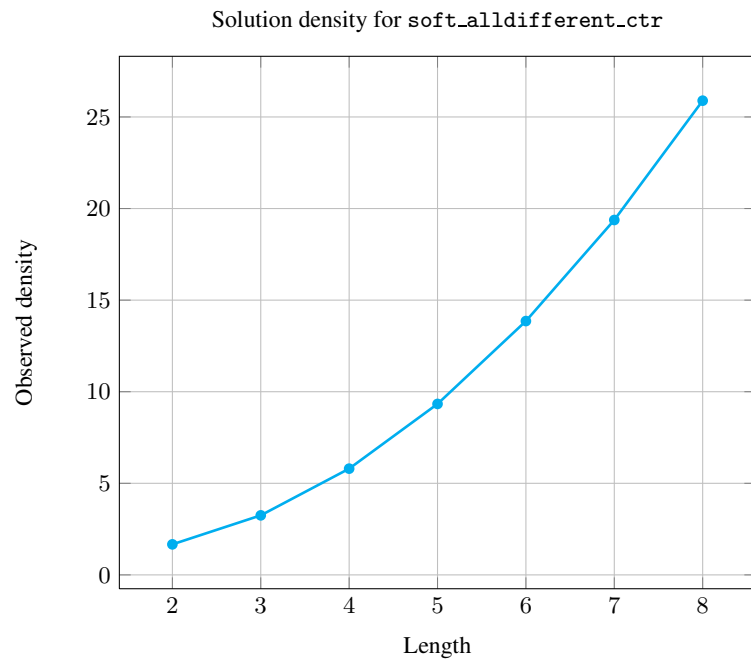
Since it focus on the soft aspect of the `alldifferent` constraint, the original article [314] that introduces this constraint describes how to evaluate the minimum value of C and how to prune according to the maximum value of C . The corresponding filtering algorithm does not achieve `arc-consistency`. W.-J. van Hove [422] presents a new filtering algorithm that achieves `arc-consistency`. This algorithm is based on a reformulation into a `minimum-cost flow` problem.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	15	208	3625	72576	1630279	40632320	1114431777

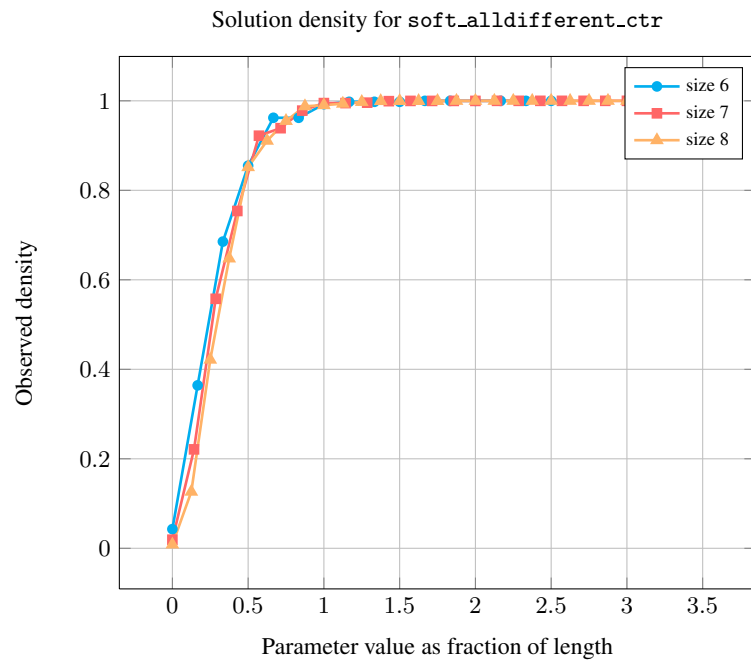
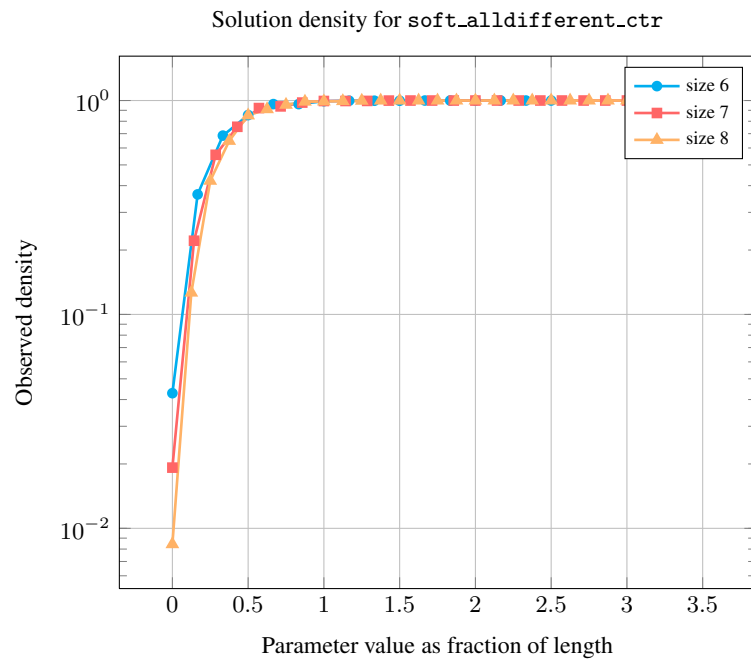
Number of solutions for `soft_alldifferent_ctr`: domains $0..n$





Length (n)		2	3	4	5	6	7	8
Total		15	208	3625	72576	1630279	40632320	1114431777
Parameter value	0	6	24	120	720	5040	40320	362880
	1	9	60	480	4320	42840	463680	5443200
	2	-	60	540	6120	80640	1169280	18144000
	3	-	64	620	7320	100590	1580880	27881280
	4	-	-	620	7620	113190	1933680	36666000
	5	-	-	620	7620	113190	1968960	39206160
	6	-	-	625	7770	116760	2051280	41111280
	7	-	-	-	7770	117390	2086560	42522480
	8	-	-	-	7770	117390	2086560	42628320
	9	-	-	-	7770	117390	2088520	42769440
	10	-	-	-	7776	117642	2095576	42938784
	11	-	-	-	-	117642	2096752	43023456
	12	-	-	-	-	117642	2096752	43025976
	13	-	-	-	-	117642	2096752	43030008
	14	-	-	-	-	117642	2096752	43030008
	15	-	-	-	-	117649	2097144	43044120
	16	-	-	-	-	-	2097144	43046136
	17	-	-	-	-	-	2097144	43046136
	18	-	-	-	-	-	2097144	43046136
	19	-	-	-	-	-	2097144	43046136
	20	-	-	-	-	-	2097144	43046136
	21	-	-	-	-	-	2097152	43046712
	22	-	-	-	-	-	-	43046712
	23	-	-	-	-	-	-	43046712
	24	-	-	-	-	-	-	43046712
	25	-	-	-	-	-	-	43046712
	26	-	-	-	-	-	-	43046712
	27	-	-	-	-	-	-	43046712
28	-	-	-	-	-	-	43046721	

Solution count for soft_alldifferent_ctr: domains 0.. n



See also

common keyword: `soft_all_equal_max_var`, `soft_all_equal_min_ctr`,
`soft_all_equal_min_var`, `soft_alldifferent_var` (*soft constraint*).

hard version: `alldifferent`.

implied by: equivalent, imply.

implies: soft_alldifferent_var.

related: atmost_nvalue.

Keywords

characteristic of a constraint: all different, disequality.

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

filtering: minimum cost flow.

modelling: degree of diversity of a set of solutions.

modelling exercises: degree of diversity of a set of solutions.

Arc input(s)	VARIABLES
Arc generator	<i>CLIQUE</i> (\langle) \mapsto collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	<u>NARC</u> \leq C

Graph model

We generate an initial graph with binary *equalities* constraints between each vertex and its successors. We use the arc generator *CLIQUE*(\langle) in order to avoid counting twice the same *equality* constraint. The graph property states that C is greater than or equal to the number of *equalities* that hold in the final graph.

Parts (A) and (B) of Figure 5.701 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NARC** graph property, the arcs of the final graph are stressed in bold. Since four equality constraints remain in the final graph the *cost* variable C is greater than or equal to 4.

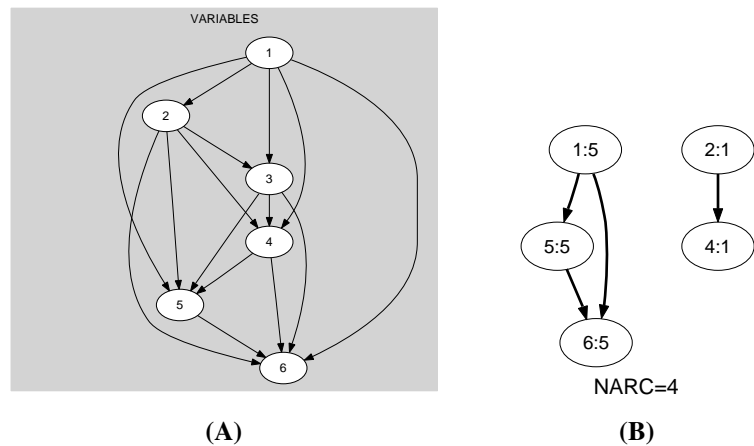


Figure 5.701: Initial and final graph of the soft_alldifferent_ctr constraint

20030820

2137