5.358 soft_all_equal_min_var

DESCRIPTION LINKS GRAPH

Origin [149]

Constraint `soft_all_equal_min_var(N, VARIABLES)`

Arguments

`N : dvar`
`VARIABLES : collection(var-dvar)`

Restrictions

$\texttt{N} \geq 0$
`required(VARIABLES, var)`

Purpose

Let M be the number of occurrences of the most often assigned value to the variables of the `VARIABLES` collection. `N` is greater than or equal to the total number of variables of the `VARIABLES` collection minus M (i.e., `N` is greater than or equal to the minimum number of variables that need to be reassigned in order to obtain a solution where all variables are assigned a same value).

Example

$(1, \langle 5, 1, 5, 5 \rangle)$

Within the collection $\langle 5, 1, 5, 5 \rangle$, 3 is the number of occurrences of the most assigned value. Consequently, the `soft_all_equal_min_var` constraint holds since the argument $\texttt{N} = 1$ is greater than or equal to the total number of variables 4 minus 3.

Typical

$\texttt{N} > 0$
$\texttt{N} < |\texttt{VARIABLES}|$
$\texttt{N} < |\texttt{VARIABLES}|/10 + 2$
$|\texttt{VARIABLES}| > 1$

Symmetries

- `N` 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.

Algorithm

Let m denote the total number of potential values that can be assigned to the variables of the `VARIABLES` collection. In [149], E. Hebrard *et al.* provides an $O(m)$ filtering algorithm achieving arc-consistency on the `soft_all_equal_min_var` constraint. The same paper also provides an algorithm with a lower complexity for achieving range consistency. Both algorithms are based on the following ideas:

- In a first phase, they both compute an *envelope* of the union $\mathcal{D}$ of the domains of the variables of the `VARIABLES` collection, i.e., an array A that indicates for each potential value v of $\mathcal{D}$, the maximum number of variables that could possibly be assigned value v. Let max_occ denote the maximum value over the entries of array

A, and let $\mathcal{V}_{max\_occ}$ denote the set of values which all occur in max_occ variables of the `VARIABLES` collection. The quantity $|$`VARIABLES`$| - max\_occ$ is a lower bound of `N`.

- In a second phase, depending on the relative ordering between max_occ and the minimum value of $|$`VARIABLES`$| -$ `N`, i.e., $|$`VARIABLES`$| - \overline{\texttt{N}}$, we have the three following cases:
  1. When $max\_occ < |$`VARIABLES`$| - \overline{\texttt{N}}$, the constraint `soft_all_equal_min_var` simply fails since not enough variables of the `VARIABLES` collection can be assigned the same value.
  2. When $max\_occ = |$`VARIABLES`$| - \overline{\texttt{N}}$, the constraint `soft_all_equal_min_var` can be satisfied. In this context, a value v can be removed from the domain of a variable V of the `VARIABLES` collection if and only if:
     (a) value v does not belong to $\mathcal{V}_{max\_occ}$,
     (b) the domain of variable V contains all values of $\mathcal{V}_{max\_occ}$.

     On the one hand, the first condition can be understand as the fact that value v is not a value that allows to have at least $|$`VARIABLES`$| - \overline{\texttt{N}}$ variables assigned the same value. On the other hand, the second condition can be interpreted as the fact that variable V is absolutely required in order to have at least $|$`VARIABLES`$| - \overline{\texttt{N}}$ variables assigned the same value.
  3. When $max\_occ > |$`VARIABLES`$| - \overline{\texttt{N}}$, the constraint `soft_all_equal_min_var` can be satisfied, but no value can be pruned.

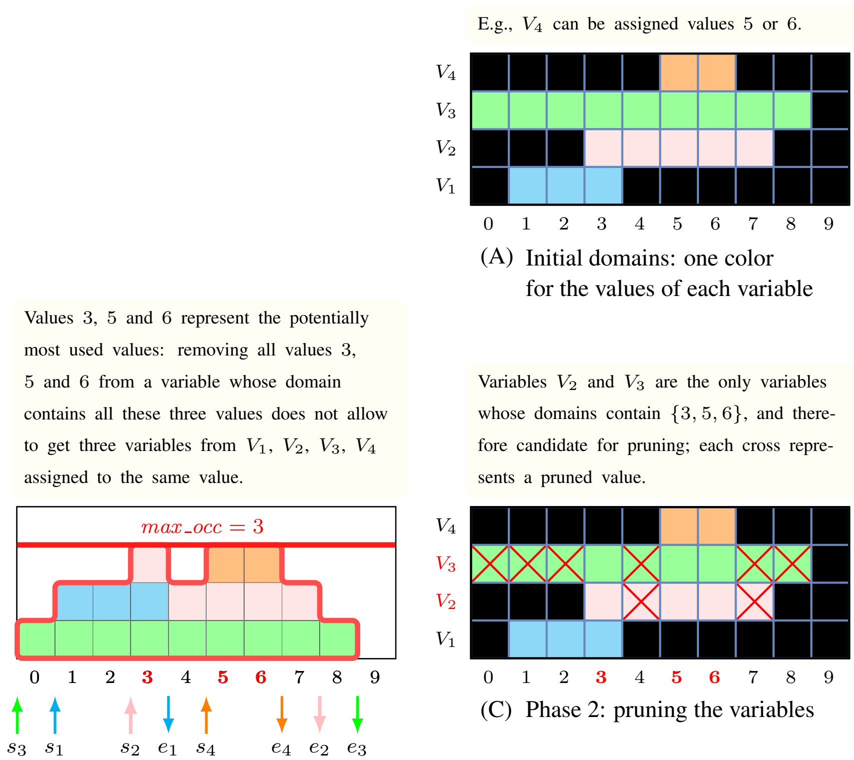
Note that, in the context of range consistency, the first phase of the filtering algorithm can be interpreted as a sweep algorithm were:

- On the one hand, the *sweep status* corresponds to the maximum number of occurrence of variables that can be assigned a given value.
- On the other hand, the *event point series* correspond to the minimum values of the variables of the `VARIABLES` collection as well as to the maximum values $(+1)$ of the same variables.

Figure 5.699 illustrates the previous filtering algorithm on an example where `N` is equal to 1, and where we have four variables V_1, V_2, V_3 and V_4 respectively taking their values within intervals $[1,3]$, $[3,7]$, $[0,8]$ and $[5,6]$ (see Part (A) of Figure 5.699, where the values of each variable are assigned a same colour that we retrieve in the other parts of Figure 5.699).

Part (B) of Figure 5.699 illustrates the first phase of the filtering algorithm, namely the computation of the envelope of the domains of variables V_1, V_2, V_3 and V_4. The *start events* s_1, s_2, s_3, s_4 (i.e., the events respectively associated with the minimum value of variables V_1, V_2, V_3, V_4) where the envelope is increased by 1 are represented by the character $\uparrow$. Similarly, the *end events* (i.e., the events e_1, e_2, e_3, e_4 respectively associated with the maximum value $(+1)$ of V_1, V_2, V_3, V_4 are represented by the character $\downarrow$). Since the highest peak of the envelope is equal to 3 we have that max_occ is equal to 3. The values that allow to reach this highest peak are equal to $\mathcal{V}_{max\_occ} = \{3,5,6\}$ (i.e., shown in red in Part (B) of Figure 5.699).

Finally, Part (C) of Figure 5.699 illustrates the second phase of the filtering algorithm. Since $max\_occ = 3$ is equal to $|$`VARIABLES`$| - \overline{\texttt{N}} = 4 - 1$ we remove from the variables whose domains contain $\mathcal{V}_{max\_occ} = \{3,5,6\}$ (i.e., variables V_2 and V_3) all values not in $\mathcal{V}_{max\_occ} = \{3,5,6\}$ (i.e., values 4, 7 for variable V_2 and values $0, 1, 2, 4, 7, 8$ for variable V_3).



(A) Initial domains: one color for the values of each variable

(B) Phase 1: computing the domains envelope (in red) from the sorted start and end events s_3, s_1, s_2, e_1, s_4, e_4, e_2, e_3

(C) Phase 2: pruning the variables

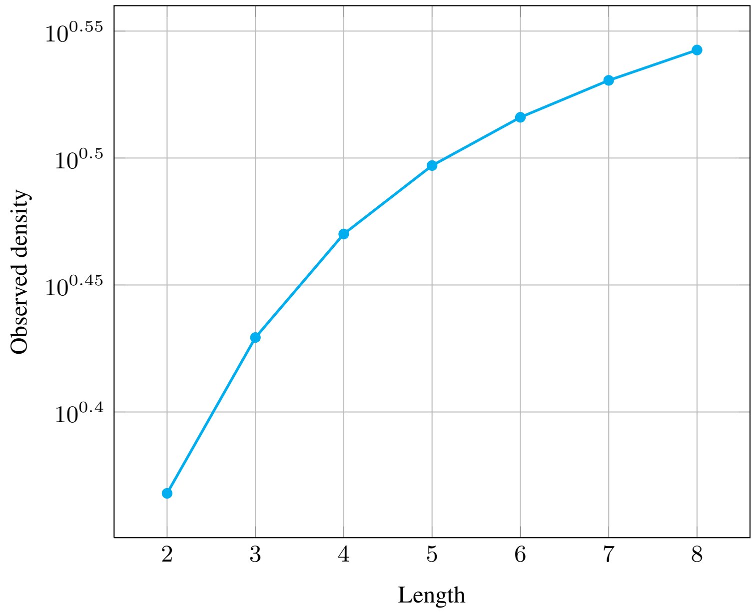
Figure 5.699: Illustration of the two phases filtering algorithm of the `soft_all_equal_min_var`$(1, \langle V_1, V_2, V_3, V_4\rangle)$ constraint with $V_1 \in [1, 3]$, $V_2 \in [3, 7]$, $V_3 \in [0, 8]$ and $V_4 \in [5, 6]$

Counting

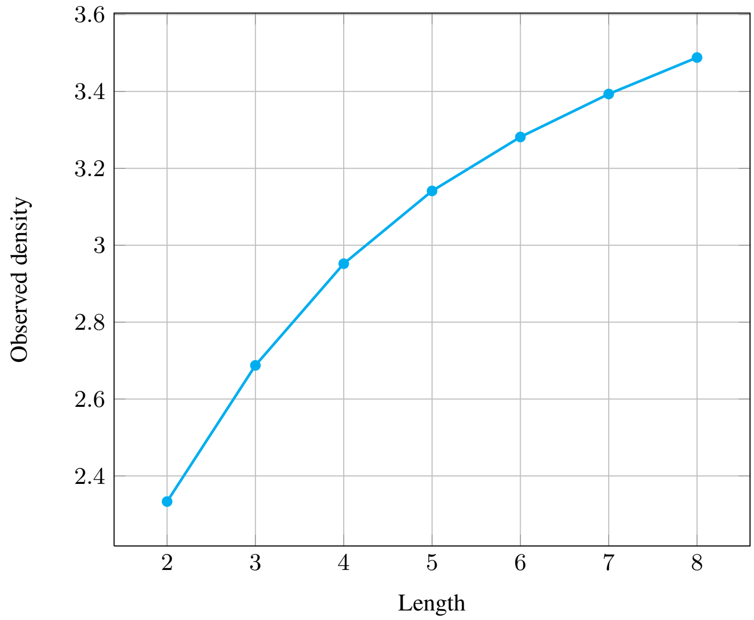
Length (n)	2	3	4	5	6	7	8
Solutions	21	172	1845	24426	386071	7116320	150156873

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

Solution density for `soft_all_equal_min_var`



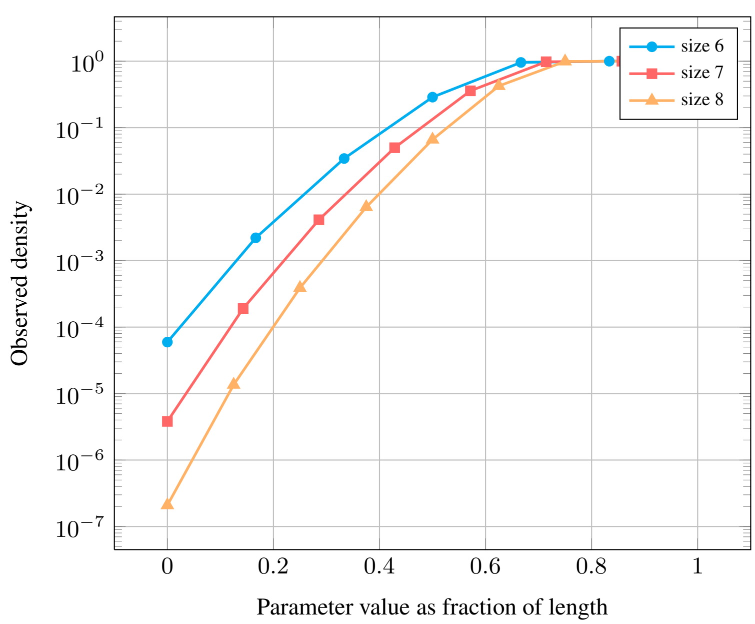
Solution density for `soft_all_equal_min_var`



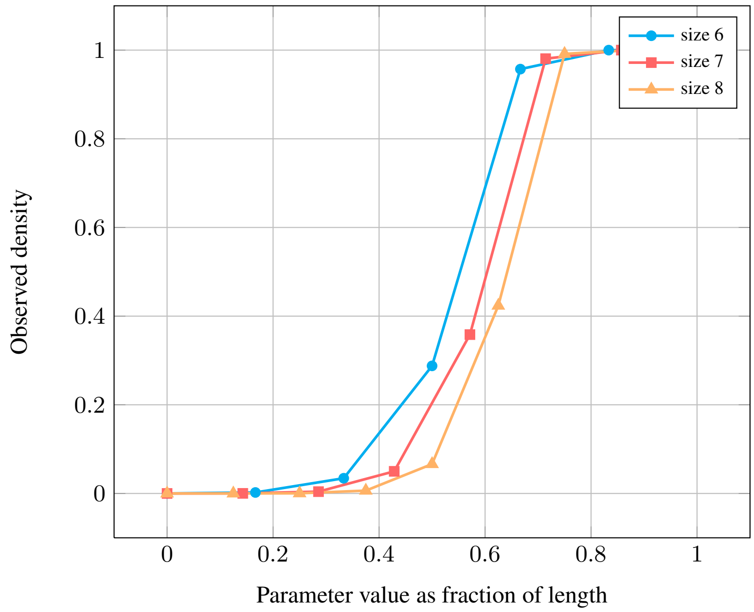
Length (n)		2	3	4	5	6	7	8
Total		21	172	1845	24426	386071	7116320	150156873
Parameter value	0	3	4	5	6	7	8	9
	1	9	40	85	156	259	400	585
	2	9	64	505	1656	4039	8632	16713
	3	-	64	625	7056	33859	104672	274761
	4	-	-	625	7776	112609	751472	2852721
	5	-	-	-	7776	117649	2056832	18234801
	6	-	-	-	-	117649	2097152	42683841
	7	-	-	-	-	-	2097152	43046721
	8	-	-	-	-	-	-	43046721

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

Solution density for `soft_all_equal_min_var`



Solution density for `soft_all_equal_min_var`



See also

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

hard version: `all_equal`.

implied by: `xor`.

related: `atmost_nvalue`.

Keywords

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

filtering: arc-consistency, sweep.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE$↦collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var = variables2.var
Graph property(ies)	$\overline{\mathbf{MAX\_NSCC}}$≥ |VARIABLES| − N

Graph model We generate an initial graph with binary *equalities* constraints between each vertex and its successors. The graph property states that N is greater than or equal to the difference between the total number of vertices of the initial graph and the number of vertices of the largest strongly connected component of the final graph.

Parts (A) and (B) of Figure 5.700 respectively show the initial and final graph associated with the **Example** slot. Since we use the **MAX_NSCC** graph property we show one of the largest strongly connected components of the final graph.

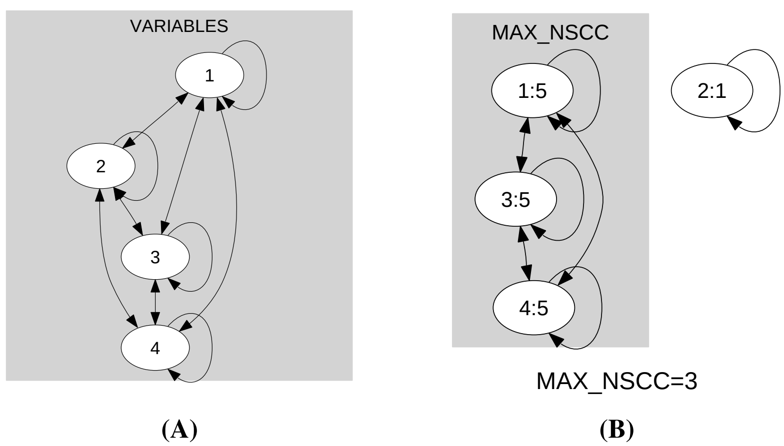


Figure 5.700: Initial and final graph of the soft_all_equal_min_var constraint