5.262 min_width_valley

DESCRIPTION LINKS AUTOMATON

Origin derived from `valley`

Constraint `min_width_valley(MIN_WIDTH, VARIABLES)`

Synonym `min_base_valley`.

Arguments

`MIN_WIDTH`	:	`dvar`
`VARIABLES`	:	`collection(var−dvar)`

Restrictions

`MIN_WIDTH` ≥ 0
`MIN_WIDTH` $\leq$ |`VARIABLES`| $- 2$
`required(VARIABLES, var)`

Purpose Given a sequence `VARIABLES` constraint `MIN_WIDTH` to be fixed to the width of the smallest valley, or to 0 if no valley exists.

Example

$(5, \langle 3, 3, 5, 5, 4, 2, 2, 3, 4, 6, 6, 5, 5, 5, 5, 5, 5, 6\rangle)$
$(0, \langle 3, 8, 8, 5, 0, 0\rangle)$
$(4, \langle 9, 8, 8, 0, 0, 2\rangle)$

The first `min_width_valley` constraint holds since the sequence $3\ 3\ 5\ 5\ 4\ 2\ 2\ 3\ 4\ 6\ 6\ 5\ 5\ 5\ 5\ 5\ 5\ 6$ contains two valleys of respective width 5 and 6 (see Figure 5.569) and since its argument `MIN_WIDTH` is fixed to the smallest value 5.

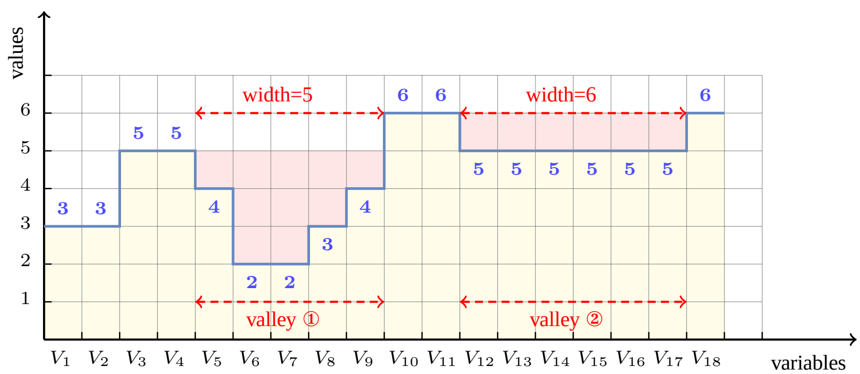


Figure 5.569: Illustration of the first example of the **Example** slot: a sequence of eighteen variables V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}, V_{12}, V_{13}, V_{14}, V_{15}, V_{16}, V_{17}, V_{18} respectively fixed to values 3, 3, 5, 5, 4, 2, 2, 3, 4, 6, 6, 5, 5, 5, 5, 5, 5, 6 and its two valleys of width 5 and 6.

Typical

$\texttt{MIN\_WIDTH} > 1$
$|\texttt{VARIABLES}| > 2$

Symmetries

- Items of `VARIABLES` can be reversed.
- One and the same constant can be added to the `var` attribute of all items of `VARIABLES`.

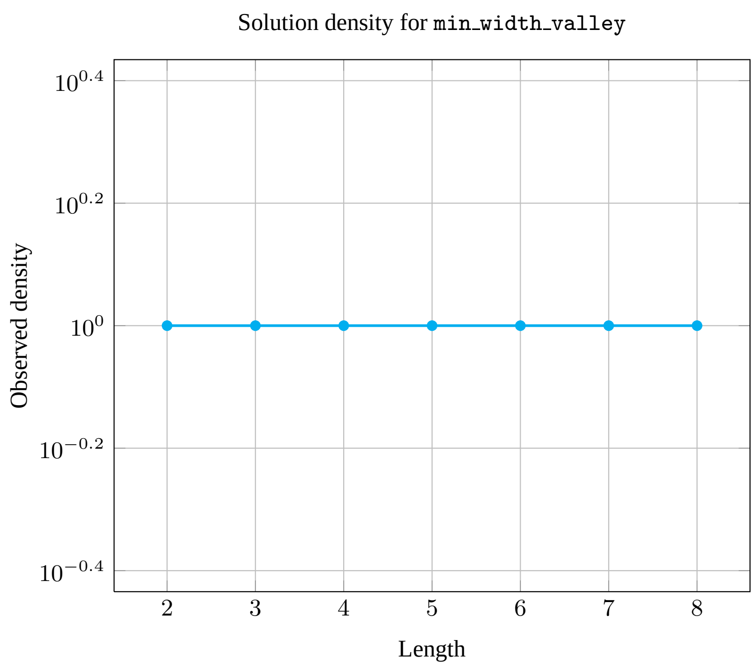
Arg. properties

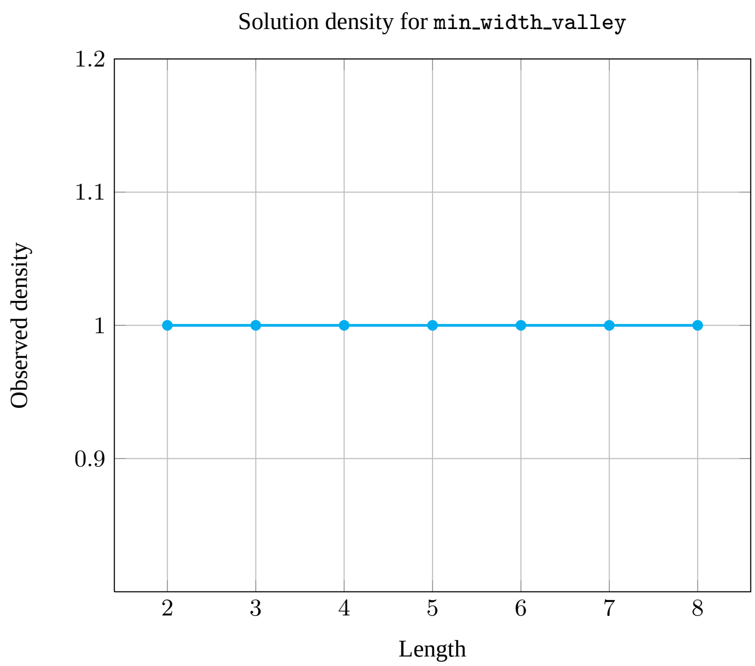
Functional dependency: `MIN_WIDTH` determined by `VARIABLES`.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	9	64	625	7776	117649	2097152	43046721

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

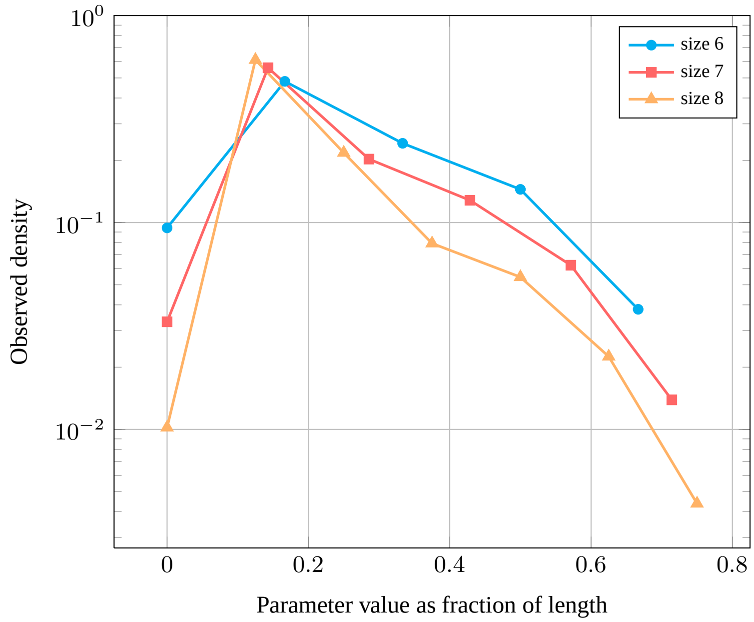




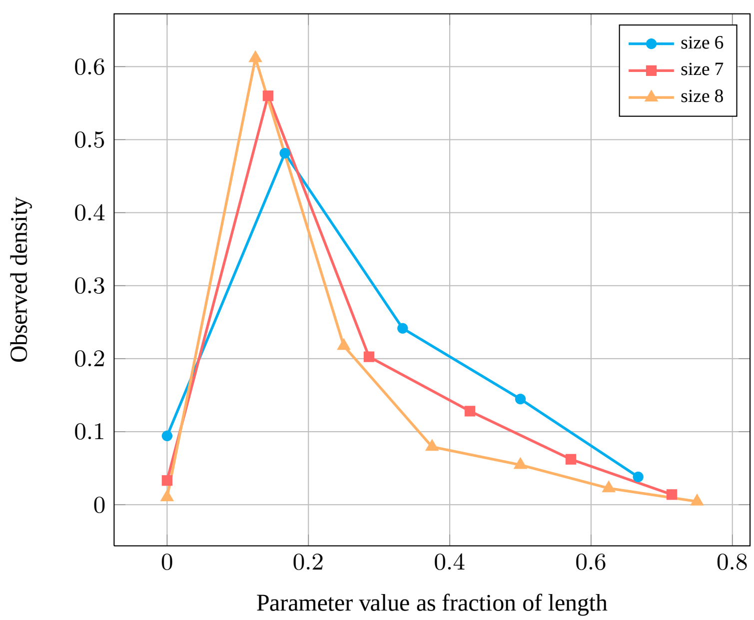
Length (n)		2	3	4	5	6	7	8
Total		9	64	625	7776	117649	2097152	43046721
Parameter value	0	9	50	295	1792	11088	69498	439791
	1	-	14	230	3205	56637	1174398	26327058
	2	-	-	100	2100	28420	424928	9363060
	3	-	-	-	679	17024	268722	3413256
	4	-	-	-	-	4480	130452	2345982
	5	-	-	-	-	-	29154	968946
	6	-	-	-	-	-	-	188628

Solution count for min_width_valley: domains $0..n$

Solution density for `min_width_valley`



Solution density for `min_width_valley`



See also **common keyword:** `valley` (*sequence*).

Keywords **characteristic of a constraint:** automaton, automaton with counters.

combinatorial object: sequence.

constraint arguments: reverse of a constraint, pure functional dependency.

filtering: glue matrix.

modelling: functional dependency.

Automaton Figure 5.570 depicts the automaton associated with the `min_width_valley` constraint. To each pair of consecutive variables $(\texttt{VAR}_i, \texttt{VAR}_{i+1})$ of the collection `VARIABLES` corresponds a signature variable S_i. The following signature constraint links $\texttt{VAR}_i$, $\texttt{VAR}_{i+1}$ and S_i: $(\texttt{VAR}_i < \texttt{VAR}_{i+1} \Leftrightarrow S_i = 0) \wedge (\texttt{VAR}_i = \texttt{VAR}_{i+1} \Leftrightarrow S_i = 1) \wedge (\texttt{VAR}_i > \texttt{VAR}_{i+1} \Leftrightarrow S_i = 2)$.

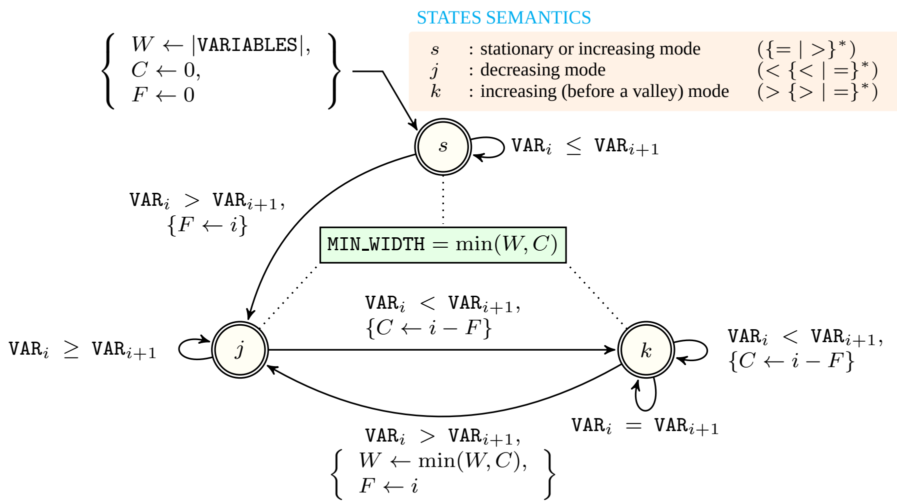


Figure 5.570: Automaton of the `min_width_valley` constraint: the start of the first potential valley is discovered while triggering the transition from s to j, the bottom of a valley is discovered while triggering the transition from j to k, the end of a valley and the start of the next potential valley are discovered while triggering the transition from k to j; the counters W, C and F respectively stand for *min_width*, *current* and *first*.

Glue matrix where $\overrightarrow{W}$, $\overrightarrow{C}$, $\overrightarrow{F}$ and $\overleftarrow{W}$, $\overleftarrow{C}$, $\overleftarrow{F}$ resp. represent the counters values W, C, F at the end of a prefix and at the end of the corresponding reverse suffix that partitions the sequence `VARIABLES`; $\overrightarrow{\texttt{MIN\_WIDTH}}$ (resp. $\overleftarrow{\texttt{MIN\_WIDTH}}$) stands for $\min(\overrightarrow{W}, \overrightarrow{C})$ (resp. $\min(\overleftarrow{W}, \overleftarrow{C})$).

	s $(\{<\mid=\}^*)$	j $(>\{>\mid=\}^*)$	k $(<\{<\mid=\}^*)$
s $(\{<\mid=\}^*)$	0	$\overleftarrow{\texttt{MIN\_WIDTH}}$	$\overleftarrow{\texttt{MIN\_WIDTH}}$
j $(>\{>\mid=\}^*)$	$\overrightarrow{\texttt{MIN\_WIDTH}}$	$\min\left(\overrightarrow{W}, n-\overrightarrow{F}-\overleftarrow{F}, \overleftarrow{W}\right)$	$\min\left(\overrightarrow{\texttt{MIN\_WIDTH}}, n-\overrightarrow{F}-\overleftarrow{F}, \overleftarrow{\texttt{MIN\_WIDTH}}\right)$
k $(<\{<\mid=\}^*)$	$\overrightarrow{\texttt{MIN\_WIDTH}}$	$\min\left(\overrightarrow{\texttt{MIN\_WIDTH}}, n-\overrightarrow{F}-\overleftarrow{F}, \overleftarrow{\texttt{MIN\_WIDTH}}\right)$	$\min\left(\overrightarrow{\texttt{MIN\_WIDTH}}, \overleftarrow{\texttt{MIN\_WIDTH}}\right)$

Figure 5.571: Glue matrix associated with the automaton of the `min_width_valley` constraint, where n stands for $|\texttt{VARIABLES}|$

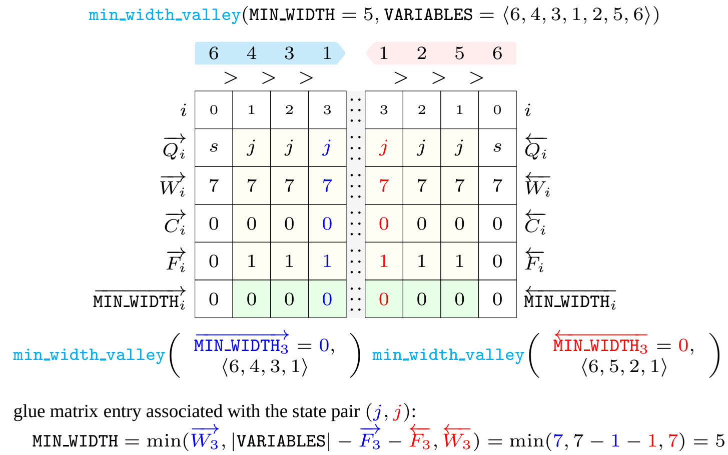


Figure 5.572: Illustrating the use of the state pair (j, j) of the glue matrix for linking MIN_WIDTH with the counters variables obtained after reading the prefix $6, 4, 3, 1$ and corresponding suffix $1, 2, 5, 6$ of the sequence $6, 4, 3, 1, 2, 5, 6$; note that the suffix $1, 2, 5, 6$ (in pink) is proceed in reverse order; the left (resp. right) table shows the initialisation (for $i = 0$) and the evolution (for $i > 0$) of the state of the automaton and its counters W, C and F upon reading the prefix $6, 4, 3, 1$ (resp. the reverse suffix $6, 5, 2, 1$).

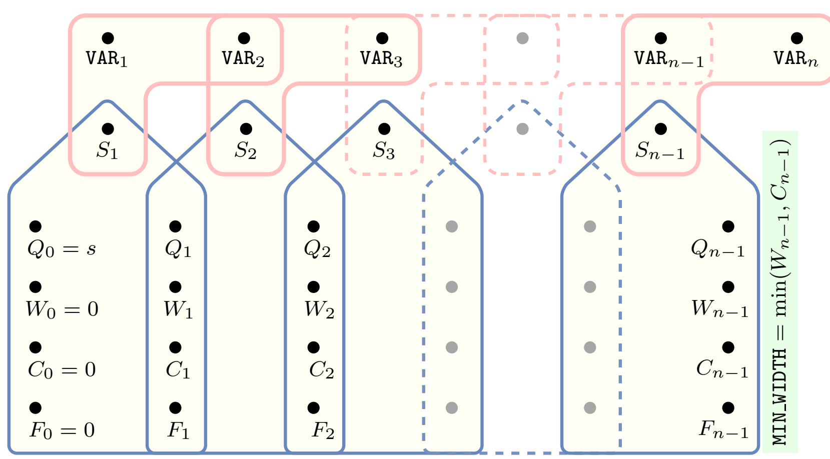


Figure 5.573: Hypergraph of the reformulation corresponding to the automaton of the min_width_valley constraint