5.17 alldifferent_interval

DESCRIPTION LINKS GRAPH AUTOMATON

Origin Derived from alldifferent.

Constraint alldifferent_interval(VARIABLES, SIZE_INTERVAL)

Synonyms alldiff_interval, alldistinct_interval.

Arguments

VARIABLES	:	collection(var−dvar)
SIZE_INTERVAL	:	int

Restrictions

required(VARIABLES, var)
SIZE_INTERVAL > 0

Purpose Enforce all variables of the collection VARIABLES to belong to distinct intervals. The intervals are defined by $[\texttt{SIZE\_INTERVAL} \cdot k, \texttt{SIZE\_INTERVAL} \cdot k + \texttt{SIZE\_INTERVAL} - 1]$ where k is an integer.

Example $(\langle 2, 4, 10\rangle, 3)$

In the example, the second argument SIZE_INTERVAL $= 3$ defines the following family of intervals $[3 \cdot k, 3 \cdot k + 2]$, where k is an integer. Since the three variables of the collection VARIABLES take values that are respectively located within the three following distinct intervals $[0, 2]$, $[3, 5]$ and $[9, 11]$, the alldifferent_interval constraint holds.

All solutions Figure 5.39 gives all solutions to the following non ground instance of the alldifferent_interval0 constraint: $\texttt{V}_1 \in [0, 7]$, $\texttt{V}_2 \in [1, 2]$, $\texttt{V}_3 \in [2, 3]$, $\texttt{V}_4 \in [0, 9]$, alldifferent_interval0$(\langle \texttt{V}_1, \texttt{V}_2, \texttt{V}_3, \texttt{V}_4\rangle, 3)$.

① $(\langle 6, 1, 3, 9\rangle, 3)$
② $(\langle 6, 2, 3, 9\rangle, 3)$
③ $(\langle 7, 1, 3, 9\rangle, 3)$
④ $(\langle 7, 2, 3, 9\rangle, 3)$

Figure 5.39: All solutions corresponding to the non ground example of the alldifferent_interval0 constraint of the **All solutions** slot

Typical

$|\texttt{VARIABLES}| > 1$
SIZE_INTERVAL > 1
SIZE_INTERVAL $<$range(VARIABLES.var)

Symmetries

- Items of `VARIABLES` are permutable.
- A value of `VARIABLES.var` that belongs to the k-th interval, of size `SIZE_INTERVAL`, can be renamed to any unused value of the same interval.
- Two distinct values of `VARIABLES.var` that belong to two distinct intervals, of size `SIZE_INTERVAL`, can be swapped.

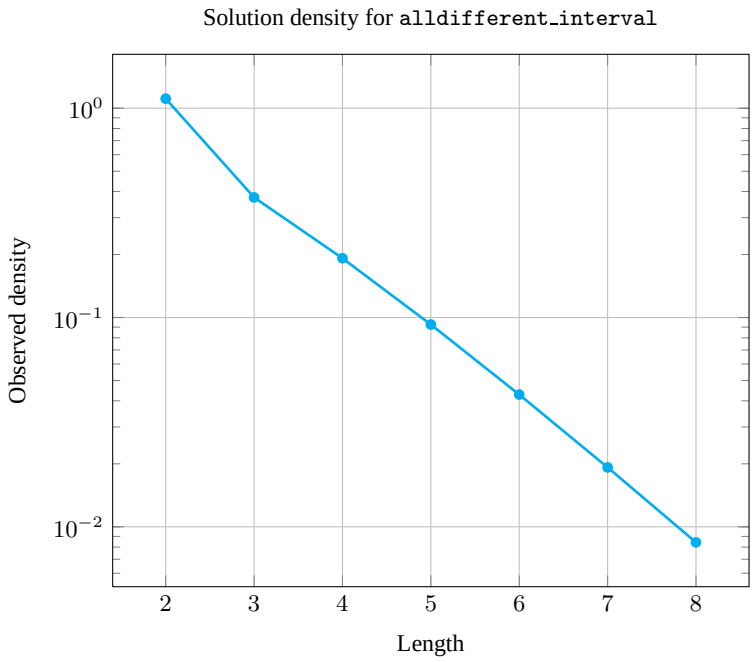
Arg. properties

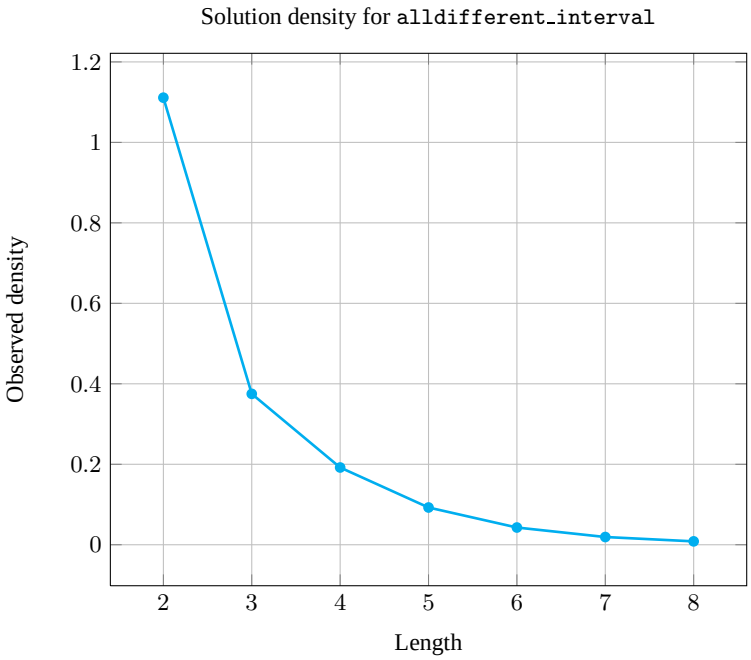
Contractible wrt. `VARIABLES`.

Counting

Length (n)	2	3	4	5	6	7	8
Solutions	10	24	120	720	5040	40320	362880

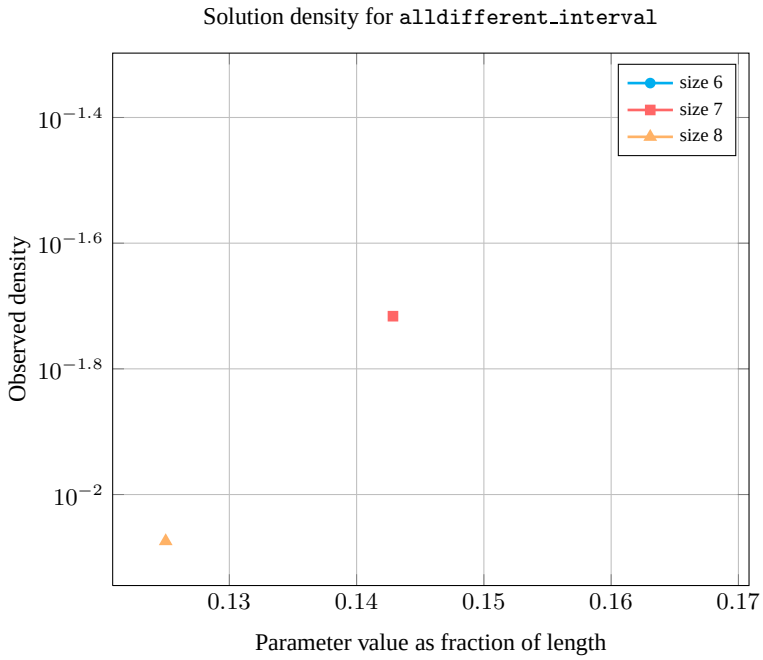
Number of solutions for `alldifferent_interval`: domains $0..n$

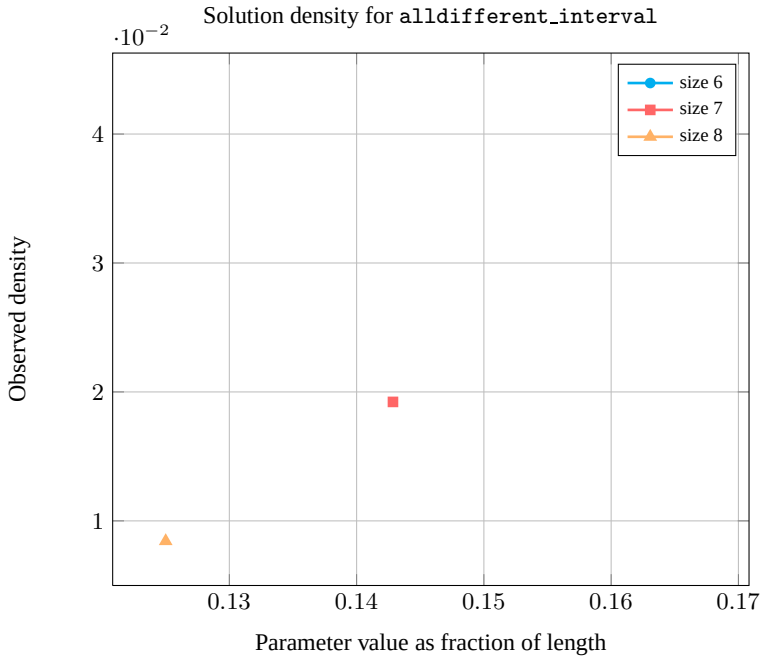




Length (n)		2	3	4	5	6	7	8
Total		10	24	120	720	5040	40320	362880
Parameter	1	6	24	120	720	5040	40320	362880
value	2	4	-	-	-	-	-	-

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





See also

implied by: `all_min_dist`.

specialisation: `alldifferent` (`variable`/`constant` *replaced by* `variable`).

Keywords

characteristic of a constraint: all different, sort based reformulation, automaton, automaton with array of counters.

constraint type: value constraint.

filtering: arc-consistency.

final graph structure: one_succ.

modelling: interval.

Arc input(s)	VARIABLES
Arc generator	$CLIQUE \mapsto$ collection(variables1, variables2)
Arc arity	2
Arc constraint(s)	variables1.var/SIZE_INTERVAL = variables2.var/SIZE_INTERVAL
Graph property(ies)	MAX_NSCC ≤ 1
Graph class	ONE_SUCC

Graph model Similar to the alldifferent constraint, but we replace the binary *equality* constraint of the alldifferent constraint by the fact that two variables are respectively assigned to two values that belong to the same interval. We generate a *clique* with a *belong to the same interval* constraint between each pair of vertices (including a vertex and itself) and state that the size of the largest strongly connected component should not exceed 1.

Parts (A) and (B) of Figure 5.40 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 component of the final graph.

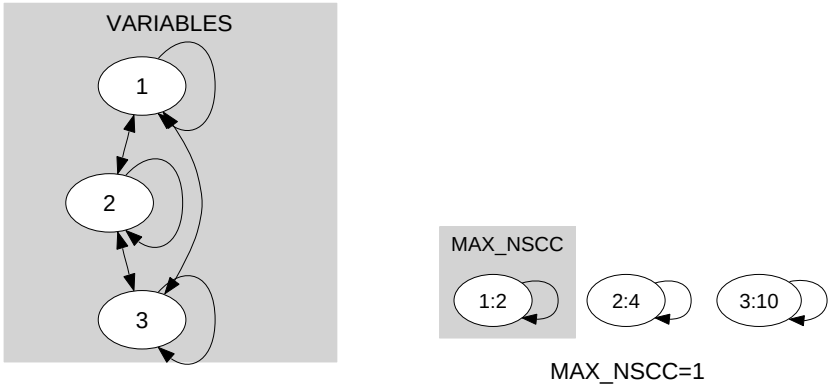


Figure 5.40: Initial and final graph of the alldifferent_interval constraint

Automaton

Figure 5.41 depicts the automaton associated with the `alldifferent_interval` constraint. To each item of the collection `VARIABLES` corresponds a signature variable $\mathtt{S}_i$ that is equal to 1. For each interval $[\mathtt{SIZE\_INTERVAL}\cdot k, \mathtt{SIZE\_INTERVAL}\cdot k+\mathtt{SIZE\_INTERVAL}-1]$ of values the automaton counts the number of occurrences of its values and finally imposes that the values of an interval are taken at most once.

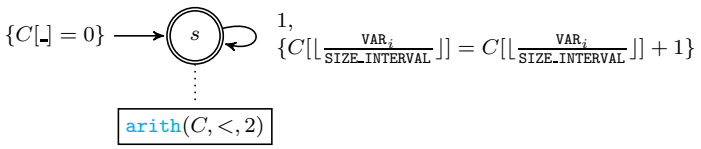


Figure 5.41: Automaton of the `alldifferent_interval` constraint