5.322 permutation

DESCRIPTION LINKS GRAPH

Origin Derived from alldifferent_consecutive_values.

Constraint permutation(VARIABLES)

Argument VARIABLES : collection(var−dvar)

Restrictions
required(VARIABLES, var)
minval(VARIABLES.var) = 1
maxval(VARIABLES.var) = |VARIABLES|

Purpose Enforce all variables of the collection VARIABLES to take distinct values between 1 and the total number of variables.

Example $(\langle 3, 2, 1, 4\rangle)$

The permutation constraint holds since all the values 3, 2, 1 and 4 are distinct, and since they all belong to interval $[1, 4]$ where 4 is the total number of variables.

Typical |VARIABLES| > 2

Symmetries
- Items of VARIABLES are permutable.
- Two distinct values of VARIABLES.var can be swapped.

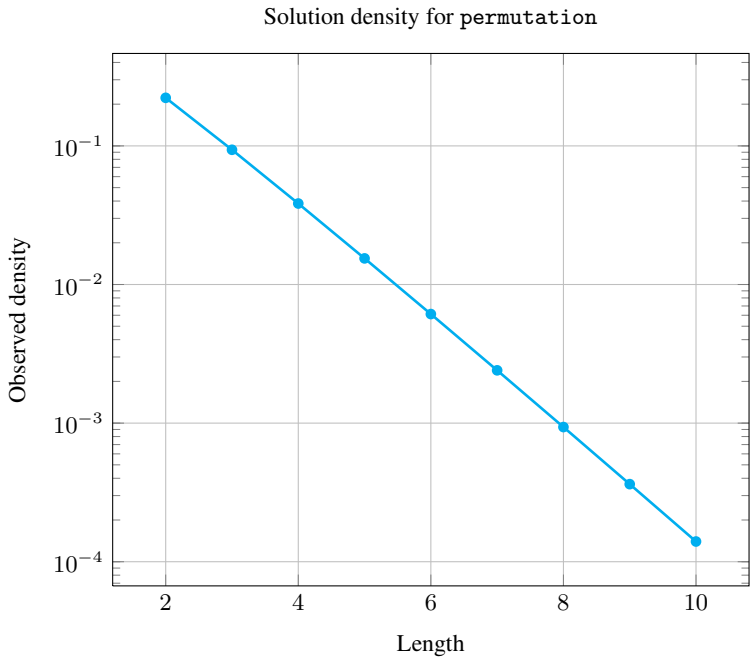
Usage See **Usage** slot of alldifferent.

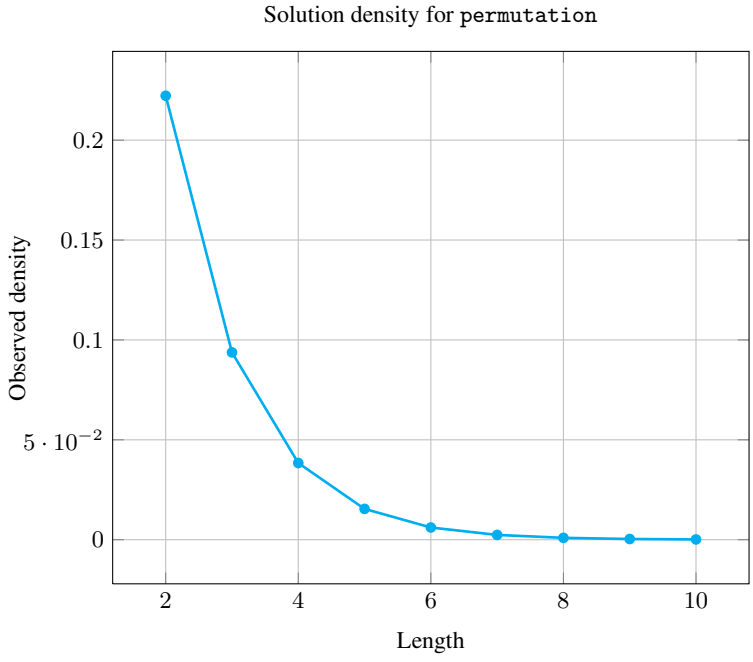
Algorithm See **Algorithm** slot of alldifferent.

Counting

Length (n)	2	3	4	5	6	7	8	9	10
Solutions	2	6	24	120	720	5040	40320	362880	3628800

Number of solutions for permutation: domains $0..n$





See also

implied by: `proper_circuit`.

implies: `alldifferent_consecutive_values`.

Keywords

characteristic of a constraint: all different, disequality, sort based reformulation.

combinatorial object: permutation.

constraint type: value constraint.

final graph structure: one_succ.

Cond. implications

- permutation(VARIABLES)
  implies balance(BALANCE, VARIABLES)
  when BALANCE = 0.

- permutation(VARIABLES)
  implies change(NCHANGE, VARIABLES, CTR)
  when NCHANGE = |VARIABLES| − 1
  and CTR ∈ [≠].

- permutation(VARIABLES)
  implies circular_change(NCHANGE, VARIABLES, CTR)
  when NCHANGE = |VARIABLES|
  and CTR ∈ [≠].

- permutation(VARIABLES)
  implies length_last_sequence(LEN, VARIABLES)
  when LEN = 1.

- permutation(VARIABLES)
  implies length_first_sequence(LEN, VARIABLES)
  when LEN = 1.

- permutation(VARIABLES)
  implies longest_change(SIZE, VARIABLES, CTR)
  when SIZE = |VARIABLES|
  and CTR ∈ [≠].

- permutation(VARIABLES)
  implies max_n(MAX, RANK, VARIABLES)
  when MAX = |VARIABLES| − RANK.

- permutation(VARIABLES)
  implies min_n(MIN, RANK, VARIABLES)
  when MIN = RANK + 1.

- permutation(VARIABLES)
  implies min_nvalue(MIN, VARIABLES)
  when MIN = 1.

- permutation(VARIABLES)
  implies min_size_full_zero_stretch(MINSIZE, VARIABLES)
  when MINSIZE = |VARIABLES|.

- permutation(VARIABLES)
  implies ninterval(NVAL, VARIABLES, SIZE_INTERVAL)
  when NVAL = (|VARIABLES| + SIZE_INTERVAL)/SIZE_INTERVAL.

- permutation(VARIABLES)
  implies range_ctr(VARIABLES, CTR, R)
  when CTR $\in [\leq]$
  and R = |VARIABLES|.

- permutation(VARIABLES)
  implies soft_alldifferent_ctr(C, VARIABLES).

- permutation(VARIABLES)
  implies soft_all_equal_max_var(N, VARIABLES)
  when N $\leq$ |VARIABLES| $-$ 1.

- permutation(VARIABLES)
  implies soft_all_equal_min_var(N, VARIABLES)
  when N $\geq$ |VARIABLES| $-$ 1.

- permutation(VARIABLES)
  implies sum_ctr(VARIABLES, CTR, VAR)
  when CTR $\in [=]$
  and VAR = |VARIABLES| $*$ (|VARIABLES| + 1)/2.

- permutation(VARIABLES)
  with |VARIABLES| $>$ 2
  and first(VARIABLES.var) $>$minval(VARIABLES.var)
  and last(VARIABLES.var) $>$minval(VARIABLES.var)
  implies deepest_valley(DEPTH, VARIABLES)
  when DEPTH =minval(VARIABLES.var).

- permutation(VARIABLES)
  with |VARIABLES| $>$ 2
  and first(VARIABLES.var) = 1
  implies deepest_valley(DEPTH, VARIABLES)
  when DEPTH = 2.

- permutation(VARIABLES)
  with |VARIABLES| $>$ 2
  and last(VARIABLES.var) = 1
  implies deepest_valley(DEPTH, VARIABLES)
  when DEPTH = 2.

- permutation(VARIABLES)
  with |VARIABLES| $>$ 2
  and first(VARIABLES.var) $<$maxval(VARIABLES.var)
  and last(VARIABLES.var) $<$maxval(VARIABLES.var)
  implies highest_peak(HEIGHT, VARIABLES)
  when HEIGHT =maxval(VARIABLES.var).

- permutation(VARIABLES)
  with |VARIABLES| $>$ 2
  and first(VARIABLES.var) = |VARIABLES|
  implies highest_peak(HEIGHT, VARIABLES)
  when HEIGHT = |VARIABLES| $-$ 1.

- permutation(VARIABLES)
  with $|\text{VARIABLES}| > 2$
  and $\text{last}(\text{VARIABLES.var}) = |\text{VARIABLES}|$
  implies highest_peak(HEIGHT, VARIABLES)
  when $\text{HEIGHT} = |\text{VARIABLES}| - 1$.

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

Graph model We generate a *clique* with an *equality* 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 one. Finally the restrictions express the fact that all values are between 1 and the total number of variables.

Parts (A) and (B) of Figure 5.651 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. The permutation holds since all the strongly connected components have at most one vertex: a value is used at most once.

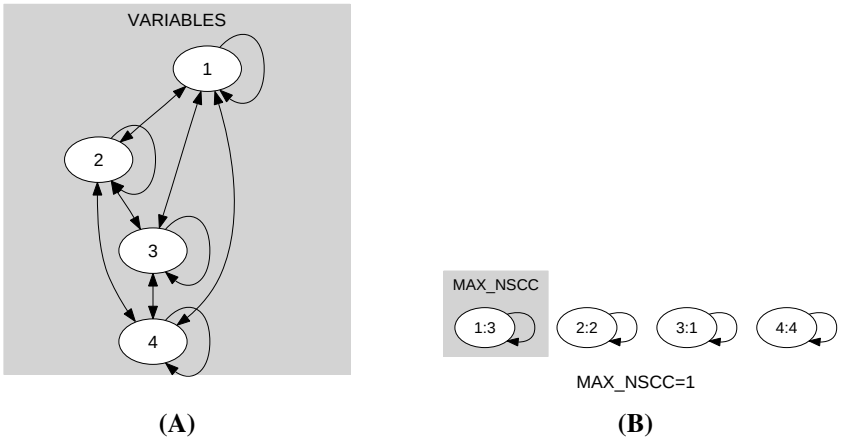


Figure 5.651: Initial and final graph of the permutation constraint