5.290 nvector

DESCRIPTION LINKS GRAPH

Origin Introduced by G. Chabert as a generalisation of `nvalue`

Constraint `nvector(NVEC, VECTORS)`

Synonyms `nvectors`, `npoint`, `npoints`.

Type

`VECTOR : collection(var−dvar)`

Arguments

`NVEC : dvar`
`VECTORS : collection(vec − VECTOR)`

Restrictions

$|\texttt{VECTOR}| \geq 1$
$\texttt{NVEC} \geq \texttt{min}(1, |\texttt{VECTORS}|)$
$\texttt{NVEC} \leq |\texttt{VECTORS}|$
`required(VECTORS, vec)`
`same_size(VECTORS, vec)`

Purpose

NVEC is the number of distinct tuples of values taken by the vectors of the collection VECTORS. Two tuples of values $\langle A_1, A_2, \ldots, A_m\rangle$ and $\langle B_1, B_2, \ldots, B_m\rangle$ are *distinct* if and only if there exist an integer $i \in [1, m]$ such that $A_i \neq B_i$.

Example

$$\left(2, \left\langle \begin{array}{l} \texttt{vec} - \langle 5, 6\rangle, \\ \texttt{vec} - \langle 5, 6\rangle, \\ \texttt{vec} - \langle 9, 3\rangle, \\ \texttt{vec} - \langle 5, 6\rangle, \\ \texttt{vec} - \langle 9, 3\rangle \end{array} \right\rangle \right)$$

The `nvector` constraint holds since its first argument NVEC = 2 is set to the number of distinct tuples of values (i.e., tuples $\langle 5, 6\rangle$ and $\langle 9, 3\rangle$) occurring within the collection VECTORS. Figure 5.604 depicts with a thick rectangle a possible initial domain for each of the five vectors and with a grey circle each tuple of values of the corresponding solution.

Typical

$|\texttt{VECTOR}| > 1$
$\texttt{NVEC} > 1$
$\texttt{NVEC} < |\texttt{VECTORS}|$
$|\texttt{VECTORS}| > 1$

Symmetries

- Items of VECTORS are permutable.
- Items of VECTORS.vec are permutable (*same permutation used*).
- All occurrences of two distinct tuples of values of VECTORS.vec can be swapped; all occurrences of a tuple of values of VECTORS.vec can be renamed to any unused tuple of values.

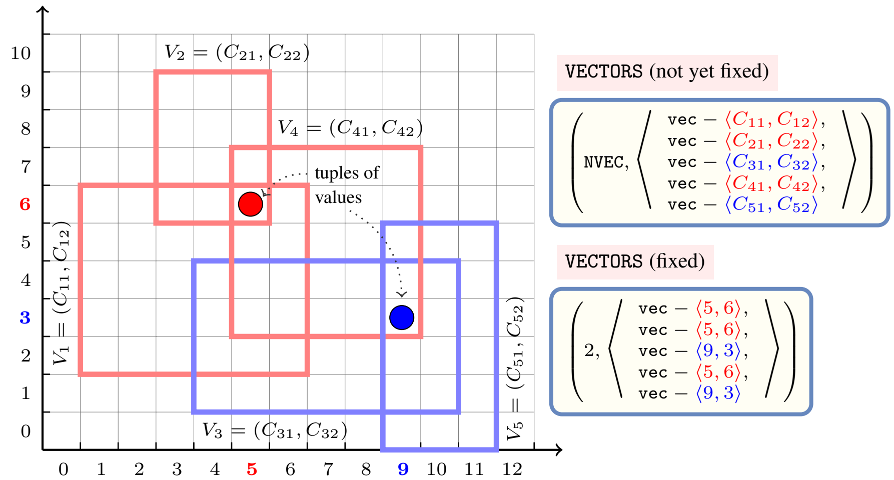


Figure 5.604: Possible initial domains ($C_{11} \in [1,6]$, $C_{12} \in [2,6]$, $C_{21} \in [3,5]$, $C_{22} \in [6,9]$, $C_{31} \in [4,10]$, $C_{32} \in [1,4]$, $C_{41} \in [5,9]$, $C_{42} \in [3,7]$, $C_{51} \in [9,11]$, $C_{52} \in [0,5]$) and solution corresponding to the **Example** slot: we have two distinct vectors ($\texttt{NVEC} = 2$)

Arg. properties

- Functional dependency: NVEC determined by VECTORS.
- Contractible wrt. VECTORS when $\texttt{NVEC} = 1$ and $|\texttt{VECTORS}| > 0$.
- Contractible wrt. VECTORS when $\texttt{NVEC} = |\texttt{VECTORS}|$.

Remark

It was shown in [109, 108] that, finding out whether a `nvector` constraint has a solution or not is NP-hard (i.e., the restriction to the rectangle case and to the atmost side of the `nvector` were considered for this purpose). This was achieved by reduction from the rectangle clique partition problem.

Reformulation

Assume the collection VECTORS is not empty (otherwise $\texttt{NVEC} = 0$). In this context, let n and m respectively denote the number of vectors of the collection VECTORS and the number of components of each vector. Furthermore, let $\alpha_i = \min(\underline{C_{1i}}, \underline{C_{2i}}, \ldots, \underline{C_{ni}})$, $\beta_i = \max(\overline{C_{1i}}, \overline{C_{2i}}, \ldots, \overline{C_{ni}})$, $\gamma_i = \beta_i - \alpha_i + 1$, $(i \in [1, m])$. By associating to each vector

$$\langle C_{k1}, C_{k2}, \ldots, C_{km}\rangle, \ \ (k \in [1, n])$$

a variable

$$D_k = \sum_{1 \le i \le m} \left(\left(\prod_{i<j\le m} \gamma_j \right) \cdot (C_{ki} - \alpha_i) \right),$$

the constraint

nvector(NVEC,
 $\langle$vec − $\langle C_{11}, C_{12}, \ldots, C_{1m}\rangle$,
 vec − $\langle C_{21}, C_{22}, \ldots, C_{2m}\rangle$,
 ....
 vec − $\langle C_{n1}, C_{n2}, \ldots, C_{nm}\rangle\rangle$)

can be expressed in term of the constraint
nvalue(NVEC, $\langle D_1, D_2, \ldots, D_n\rangle$).

Note that the previous reformulation does not work anymore if the variables have a continuous domain, or if an overflow occurs while propagating the equality constraint $D_k = \sum_{1\leq i\leq m} \left(\left(\prod_{i<j\leq m} \gamma_j\right) \cdot (C_{ki} - \alpha_i)\right)$ (i.e., the number of components m is too big).

When using this reformulation with respect to the **Example** slot we first introduce $D_1 = 1{\cdot}6{-}3{+}(4{\cdot}5{-}20)) = 3$, $D_2 = 1{\cdot}6{-}3{+}(4{\cdot}5{-}20)) = 3$, $D_3 = 1{\cdot}3{-}3{+}(4{\cdot}9{-}20)) = 16$, $D_4 = 1 \cdot 6 - 3 + (4 \cdot 5 - 20)) = 3$, $D_5 = 1 \cdot 3 - 3 + (4 \cdot 9 - 20)) = 16$ and then get the constraint nvalue$(2, \langle 3, 3, 16, 3, 16\rangle)$.

See also

common keyword: lex_equal, ordered_atleast_nvector, ordered_atmost_nvector *(vector)*.

generalisation: nvectors *(replace an equality with the number of distinct vectors by a comparison with the number of distinct nvectors)*.

implied by: ordered_nvector.

implies: atleast_nvector *(= NVEC replaced by ≥ NVEC)*, atmost_nvector *(= NVEC replaced by ≤ NVEC)*.

specialisation: nvalue *(vector replaced by variable)*.

Keywords

application area: SLAM problem.

characteristic of a constraint: vector.

complexity: rectangle clique partition.

constraint arguments: pure functional dependency.

constraint type: counting constraint, value partitioning constraint.

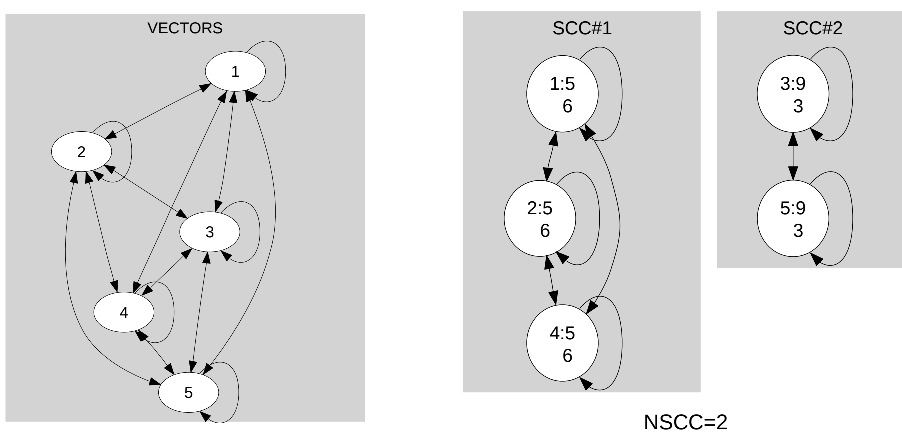
final graph structure: strongly connected component, equivalence.

modelling: number of distinct equivalence classes, functional dependency.

problems: domination.

Arc input(s)	`VECTORS`
Arc generator	*CLIQUE*↦`collection`(`vectors1`, `vectors2`)
Arc arity	2
Arc constraint(s)	`lex_equal`(`vectors1.vec`, `vectors2.vec`)
Graph property(ies)	**NSCC**= `NVEC`
Graph class	`EQUIVALENCE`

Graph model Parts (A) and (B) of Figure 5.605 respectively show the initial and final graph associated with the **Example** slot. Since we use the **NSCC** graph property we show the different strongly connected components of the final graph. Each strongly connected component corresponds to a tuple of values that is assigned to some vectors of the `VECTORS` collection. The 2 following tuple of values $\langle 5, 6\rangle$ and $\langle 9, 3\rangle$ are used by the vectors of the `VECTORS` collection.



(A) **(B)**

Figure 5.605: Initial and final graph of the `nvector` constraint