Title: Eliminating Interchangeable Values in Constraint Satisfaction Problems

Authors: E.C. Freuder

Proc.: AAAI

Year: 1991

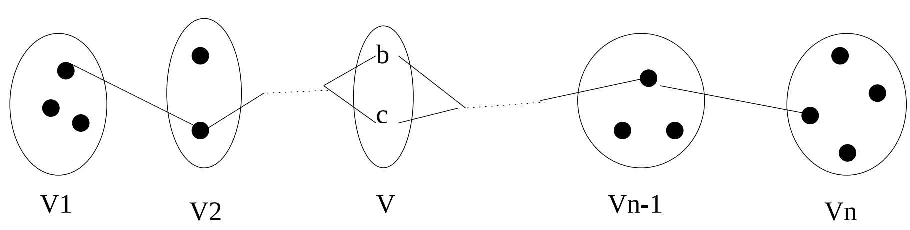
Foundations of Constraint Processing
CSCE421/821, Fall 2005
`www.cse.unl.edu/~choueiry/F05-421-821/`

Berthe Y. Choueiry (Shu-we-ri)
Avery Hall, Room 123B
`choueiry@cse.unl.edu`, Tel: (402)472-5444

Fully interchangeable (FI) values (I)

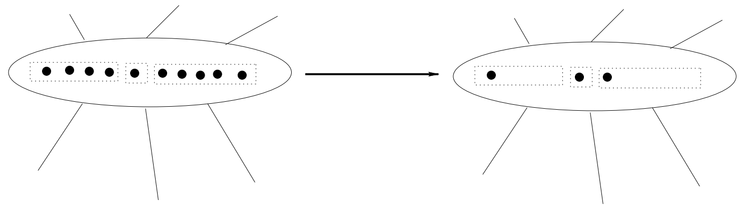
A value b for a CSP variable V is fully interchangeable with a value c for V iff:

1. Every solution to the CSP that contains b remains a solution when c is substituted for b
2. Every solution to the CSP that contains c remains a solution when b is substituted for c.



Full interchangeability (II)

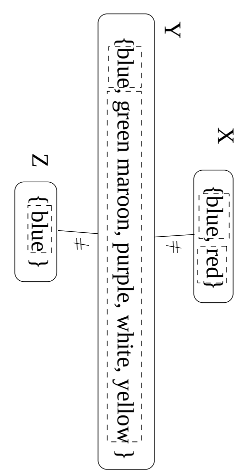
FI partition the domain of a CSP variable into sets of equivalent values



We can replace a set of FI values with a single representative of the set without effectively losing any solutions

Also, we need not retain equivalent values: solutions involving the representative can be transformed into solutions involving any other value in the set

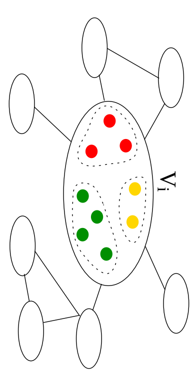
Examples

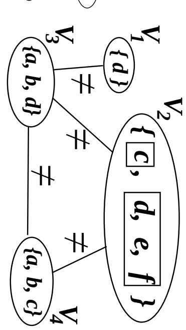


X← red

Y← green, maroon, purple, white, yellow

Z← blue





Advantages

- Eliminating interchangeable values can prune a great deal of effort from a backtrack search tree.
- If we are seeking all solutions, interchangeability allows us to find a family of similar solutions and avoid duplication of effort
- Basis for explanation, concept formation and problem decomposition

Contents of paper

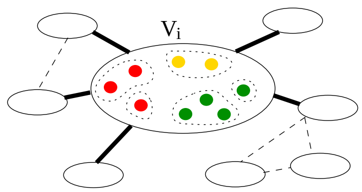
- Introduces interchangeability: symmetry between solutions (across variables) ♣ ♣ ♣
- Introduces the simplest symmetry, full interchangeability (FI), as a symmetry among the values of one CSP variable
- Approximation:
  - Neighborhood (NI), k-interchangeability (from global to **local**) *+algorithm*
  - Extended:
    * Substitutability, partial interchangeability (PI), subproblem interchangeability (from strong to **weak**)
    * Meta interchangeability
    * Dynamic interchangeability

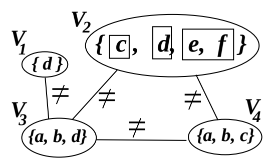
Neighborhood interchangeability (NI)

Two values, a and b for a CSP variable V are NI iff for every C on V:

$$\{i \mid (a, i) \text{ satisfies } C\} = \{i \mid (b, i) \text{ satisfies } C\}$$

NI is restricted to $\texttt{Neigh}(V)$





Theorem 1: NI $\Rightarrow$ FI

Neighborhood interchangeability is a sufficient, but not a necessary condition for full interchangeability

Proof:

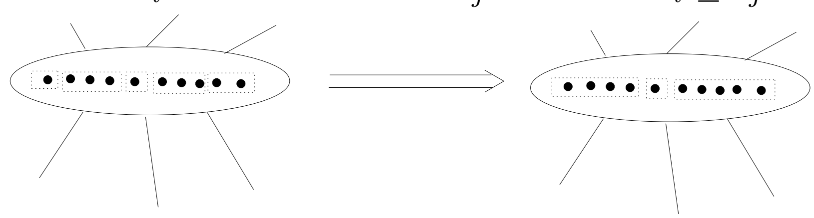
If two values are NI, there is no way they could fail to be interchangeable in any complete solution since there is not constraint that one could satisfy and not the other

But, FI values may not be NI: previous example, insoluble problem

NI as an approximation for FI

NI considers only constraints in neighborhood of the variable
FI considers the effects of all constraints in the problem on the variable

NI $\Rightarrow$ FI
For each NI-set s_i there is an FI-set s_j such that $s_i \subseteq s_j$



Finding the NI-partition only needs polynomial time
Finding FI-partition may require finding all solutions to problem..

Computing NI: discrimination tree (DT)

Algorithm DT for V_i (D_{V_i}, `Neigh`(V_i))

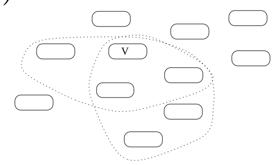
Create the root of the discrimination tree
Repeat for each value $v_i \in D_{V_i}$:
  Repeat for each variable $V_j \in$ `Neigh`(V_i):
    Repeat for each $v_j \in D_{V_j}$ consistent with v_i for V_i:
      Move to if present, construct and move to if not,
      a child node in the tree corresponding to '$V_j = v_j$'.
  Add '$V_i, \{v_i\}$' to annotation of the node (or root),
  Go back to the root of the discrimination tree.

The annotations in the tree give the NI-sets for V_i

Example: on board.
Time complexity:...
Space complexity:...

K-interchangeability

Two values, a and b for a CSP variable V are k-interchangeable iff a and b are fully interchangeable in **every** subproblem of the CSP induced by V and $(k-1)$ other variables



- NI = x-interchangeability, $x =$?
- FI = x-interchangeability, $x =$?
- When $k < n$, k-interchangeability is local
- **Theorem 2**: $\forall i < j$, i-interchangeability $\Rightarrow$ j-inter.
- Algorithm: generalization of DT, requires solving subproblems of size $k-1$
- Complexity: $O(n^k d^k)$, $n =$#variables and $d =$ domain size

Interchangeability types

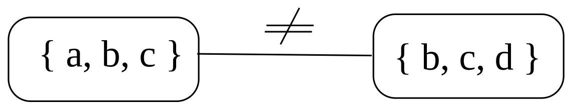
From local to global: NI $\rightarrow$ k-interchangeability $\rightarrow$ FI

Extended interchangeability:

- Weak Interchangeability: valid in some solutions to the CSP, not all
  - Substitutability
  - Partial interchangeability
  - Subproblem interchangeability
- Meta-interchangeability
- Dynamic interchangeability
- Functional interchangeability (the real thing)

Substitutability: One-way interchangeability

Given two values, a and b for a CSP variable V, a is substitutable for b iff substituting a in any solution involving b yields a solution



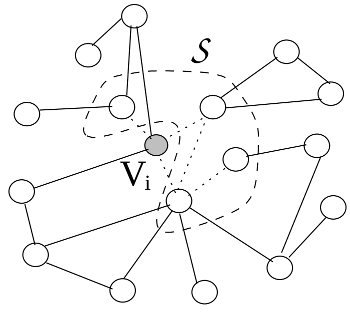
Note:

- a is substitutable for $b \not\Rightarrow b$ is substitutable for a
- One can define neighborhood substitutability

Partial Interchangeability (PI)

The values for variables may differ among themselves, but be fully interchangeable with respect to the rest of the world
Idea: define a boundary to confine change

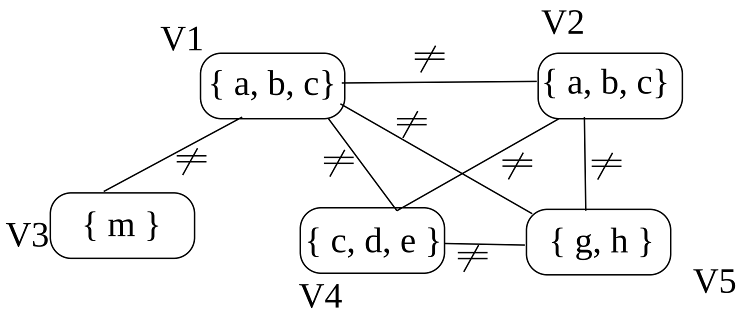
Two values are partially interchangeable with respect to a subset S of variables, iff any solution involving one implies a solution involving the other with possibly different values for variables in S



When S is empty, what do we have??

PI: example

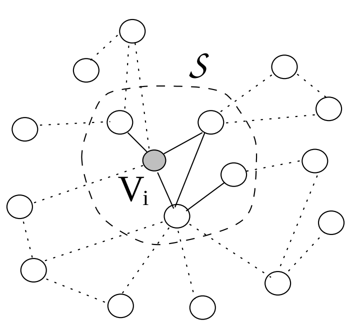
PI values for $V1$ given the boundary $S = \{V2\}$



One can define neighborhood partial interchangeability (NPI)

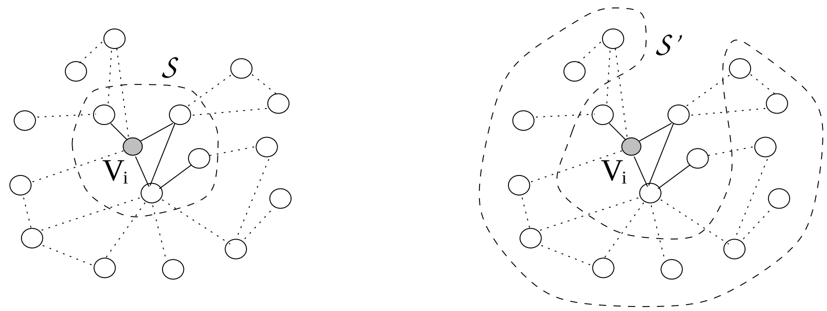
Subproblem interchangeability

Two values are subproblem interchangeable with respect to a subset of variables S iff they are fully interchangeable with respect to the solution of the subproblem of the CSP induced by S



Subproblem interchangeability vs. PI

Theorem 5: Given S, and $S' = \mathcal{V} - S$
PI WRT $S' \Rightarrow$ Subproblem interchangeability WRT S
Subproblem interchangeability WRT $S \not\Leftarrow$ PI WRT S'



Proof: the key idea is that a solution to a subproblem may fail to appear as a portion of any solution to the complete problem

Note: Theorem is inverted in paper, proof is correct.

Meta-interchangeability

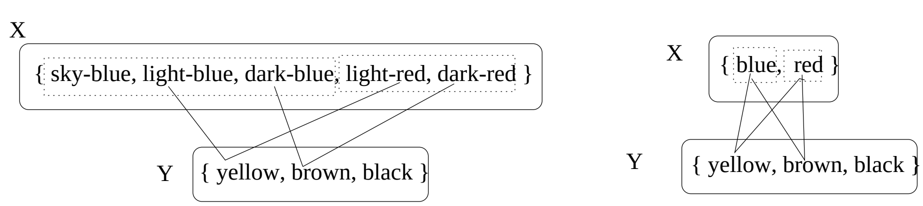
By grouping:
- variables into meta-variables or
- values into meta-values

we can introduce interchangeability into a higher level representation of the original CSP.

Variables $\rightarrow$ meta-variables

Values $\rightarrow$ meta-values

CSP $\rightarrow$ meta-CSP



Yellow and brown for Y become fully interchangeable

Dynamic interchangeability

Interchangeability can be recalculated after choices are made for variables values during backtrack search

Functional interchangeability

Notation: $S_{v|V}$ be the set of solutions with $V \leftarrow v$.

Two values a for variable V and b for variable W are functionally interchangeable iff there exist functions f_a and f_b such that:

$$f_a(S_{a|V}) = S_{b|W} \text{ and } f_b(S_{b|W}) = S_{a|V}$$

Special case: $V = W$

In general, find all solutions, define your mappings :—(

Isomorphic interchangeability

Two values a and b for a CSP variable are isomorphically interchangeable iff there exists a 1-1 function f such that:

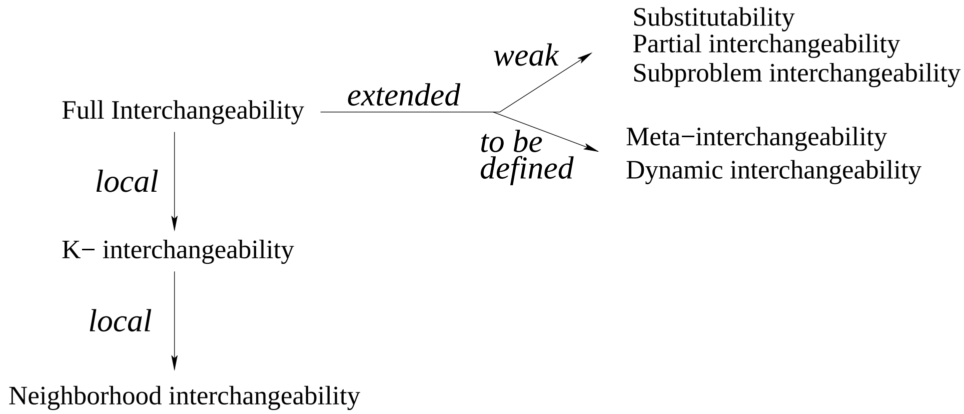
1. $b = f(a)$
2. for any solution S involving a, $\{f(v) | v \in S\}$ is a solution
3. for any solution S involving b, $\{f^{-1}(v) | v \in S\}$ is a solution

Example: 8-queens

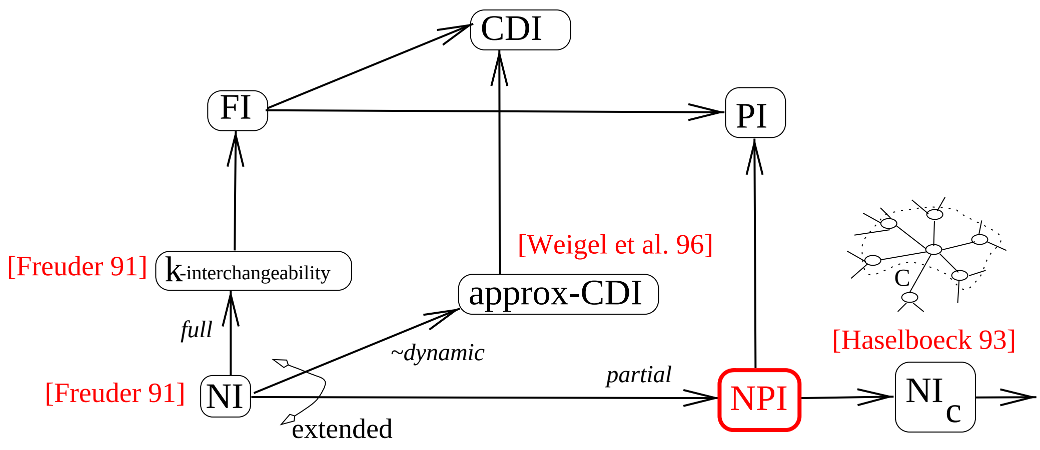
- For each row, values in col are isomorphically inter.
  $f : i \rightarrow (9 - i)$
- Also, $f : (i, j) \rightarrow (9 - i, 9 - j)$ *double-check*

Types of interchangeability: summary

- Most general: Functional interchangeability
- Restricting the function: Isomorphic interchangeability
- Restricting to **<u>one</u>** variable:



Lattice of some interchangeability relations



→ NPI is ‘tunable’

Summary

- Formalizes interchangeability in CSPs
- Introduces many types of interchangeability relations and show some relations among them
- Provides an algorithm for computing NI and k-interchangeability
- Recent developments:
  - NI_C and bundling: Haselboeck, IJCAI’93
  - CDI: Weigel, Faltings, Choueiry ECAI’96
  - NPI: Choueiry, Noubir, AAAI’98
  - Dynamic bundling (DynBundl): Beckwith, Choueiry, AusJCAI’01
  - Bundling for join query computation: Lal, Choueiry, CDB’04
  - DynBundl for non-binary CSPs: Lal, Choueiry, Freuder AAAI’05
  - etc.