Understanding and Managing Symmetry in CSPs

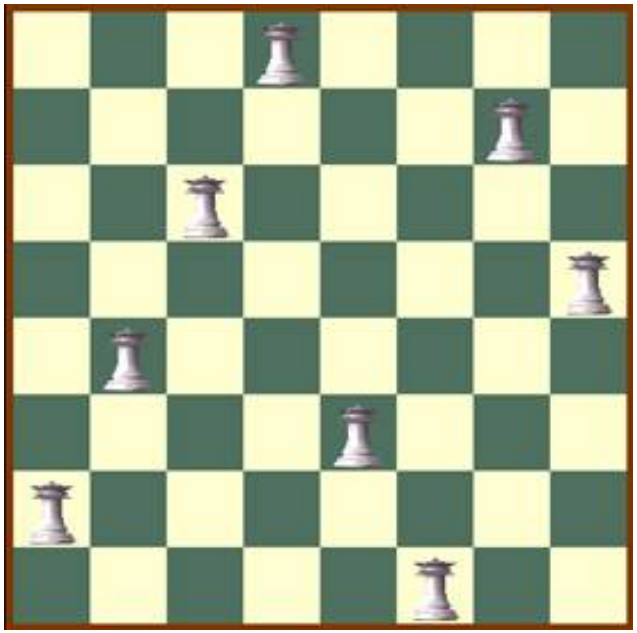
Steve Linton

Centre for Interdisciplinary Research in Computational Algebra
University of St Andrews

MathCSP, Oxford, March 2006

Outline

N Queens

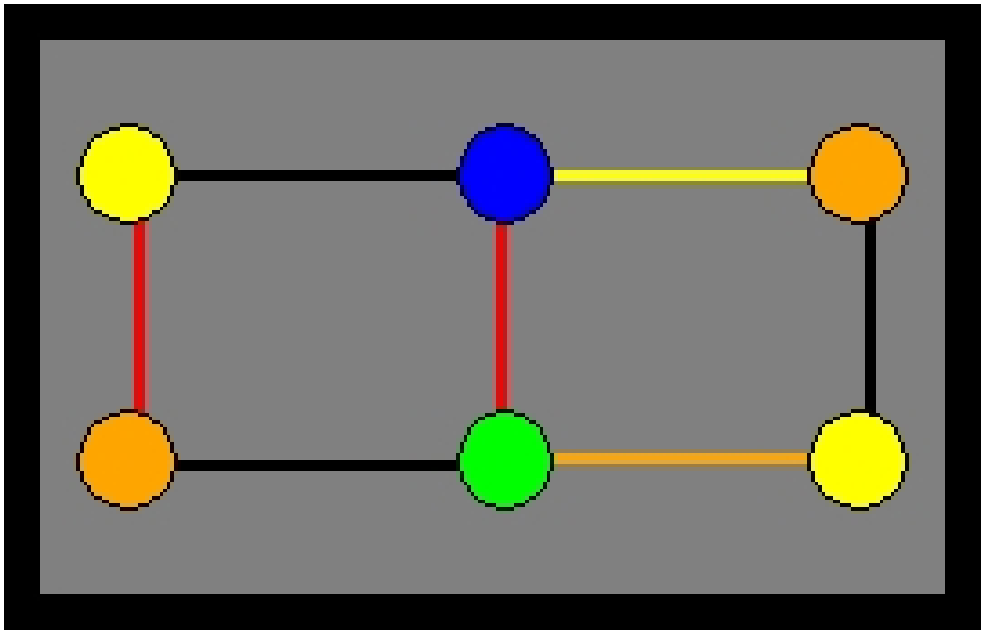


- $V = \{1, 2, 3, \ldots, 8\}$, $D = \{a, b, c, \ldots h\}$
- Symmetries of the square

Transport Planning

- Variables for positions of vehicles at various times, items loaded onto trucks, . . .
- Multiple interchangeable vehicles
- Symmetries permute variables

Graph Colouring



- Colours are interchangeable
- Symmetries permute values

Symmetry has Many Origins

- Inherent symmetry of the problem
  - n-queens
  - Hopefully the user can tell us these
- Artifacts of modelling
  - sets represented as sequences, . . .
  - record these during modelling
  - modelling can also lose symmetries
    - $a = b = c = d$ modelled as $a = b \wedge b = c \wedge c = d$

We Want to Handle the Symmetry

- To decide satisfiability
  - Can make intractable problems tractable
  - Finding the symmetries is Graph Isomorphism in general
    - Question: when is it easier than that?
- To find all inequivalent solutions
- To specify the problem more concisely

Pigeonhole Problem

$V = \{Loc_1, \ldots, Loc_n\}$ pigeons
$D = \{1, \ldots, m\}$ holes
$C = \{Loc_1 \neq Loc_2\}$
$G = S_V \times S_D$ imposed symmetries

Natural, but Unhelpful Definition

Natural Definition

Symmetries are permutations of the full assignments D^V which preserve satisfying assignments.

Problems

No interaction with search
Huge permutation degree

Literals and Partial Assignments

Literals

We call elements of $V \times D$ literals

Partial Assignments

A set of literals with no repeated variables is a partial assignment
The variables occurring in it are its scope
A partial assignment is called satisfying iff its projection onto the intersection of its scope with the scope of each constraint lies in the projection of the constraint

Satisfying partial assignments correspond to reasonable places you might reach in the search tree.

Symmetries 1

Symmetries

A permutation of the literals is a symmetry if it maps satisfying partial assignments to partial assignments and maps satisfying total assignments to satisfying total assignments

Strict symmetries

A symmetry is strict if it maps satisfying partial assignments to satisfying partial assignments

Instance symmetries

A symmetry is an instance symmetry if it fixes the constraints as a set.

Examples 1

Symmetries

Swapping a and c in $a = b \wedge b = c \wedge c = d$

Strict symmetries

Rotation of the chessboard in n-Queens

Instance symmetries

Reflection about vertical axis in n-Queens

Symmetries 2

Variable Symmetries

induced by a permutation of V

Value Symmetries

Map $(v, d) \rightarrow (v, d')$ for all v, d

Global Value Symmetries

induced by a permutation of D.

Any value symmetries can be made global by relabelling domains.

Examples 2

Variable Symmetries

Permutation of lorries, geometric symmetries of a graph

Value Symmetries

Recolouring a component of a graph, $x \rightarrow -x$ in $x^2 + y = 3$

Global Value Symmetries

Recolouring a graph

Symmetry Breaking

CSP Search algorithms

Take an instance, return every satisfying full assignment
if you just want satisfiability, interrupt it after the first one

CSP Search with (Complete) Symmetry Breaking

Take an instance and a group of symmetries, return (exactly) one satisfying full assignment in each orbit of the prescribed symmetry group

Static and dynamic

In static methods the representative(s) of the orbits are known *a priori*
In dynamic methods the representative(s) depend(s) on the search

Symmetry Breaking Constraints (SBC)

N Queens

Assume that the queen in the top row is on the left

- Replace $I = (V, D, C)$ by $I' = (V, D, C \cup C')$ such that I' has (at least) one satisfying assignment in each orbit of satisfying assignments of I.
- If it's exactly one then the symmetry is completely broken
- Crawford *et al* showed that exponentially many constraints can be needed to completely break even quite "well-behaved" symmetry groups
- Double-lex is an example of an efficient, but incomplete, SBC for $S_n \times S_n$ acting on an $n \times m$ matrix of variables.
- Open: what's left after double-lex?

Generalized Lex – Work in Progress

- Static symmetry breaking technique for any symmetry group G
- Fix an ordering of the literals
- Only explore partial assignments that are lexicographically least in their orbit under the group (as sets of literals)
- Checking this is a potentially exponential group theoretic computation at each node, but seems efficient in practice
- May be compatible with suitable SBCs

Value Symmetries

Graph Colouring Heuristic

Never use any new colour except the smallest one.

Theorem

Value symmetries can be broken in polynomial time (per node)

- At each node, compute the sequence-wise stabilizer of all the values in use already
- Take the orbits of this on the possible values of the next variable
- Only consider one representative of each orbit

On not Going to the Same Place Twice

Idea

Once we have explored the sub-search tree rooted at P, we never want to enter any node Q such that $\exists g \in G : g(P) \subseteq Q$

- SBDS posts constraints $\bigwedge_{g \in G} \neg g(P)$ after leaving P. Can be a lot of constraints.
- SBDD remembers maximal completed subtrees P and tests each node Q for a g. Can be exponential search per node.

Symmetrised Propagation

- If g is a symmetry of C and $C \implies v \neq d$ then $C \implies g(v \neq d)$
- C here has to include search decisions made so far
- For instance symmetries this just speeds up propagation that would happen anyway
- For other symmetries it can give stronger propagation
- If you want to use symmetry to specify problems, this becomes vital

Open Questions and Future Directions

- More experiments, better algorithms for existing methods
- Combining methods – when is it sound
- Tractability questions for symmetry detection and for (eg) SBDD or GenLex
- *G*-constraints
- Interaction of symmetry and inference.