Recent Results in Symmetry Breaking

Justin Pearson

Uppsala University

May 27, 2009

Joint work (only one name appears on all papers) between:

- Pierre Flener(Uppsala), Ian Gent (St Andrews), Pascal Van Hentenryck (Brown), Tom Kelsey (St Andrws), Steve Linton (St Andrews),
- Justin Pearson (Uppsala), Colva M. Roney-Dougal (St Andrews), Meinolf Sellmann (Brown),
- and Magnus Ågren (Uppsala)

Outline of ideas in the talk

- Try to understand the boundary between tractable and intractable symmetry breaking.

Outline of ideas in the talk

- Try to understand the boundary between tractable and intractable symmetry breaking.
- Specialised search procedures to break symmetry,

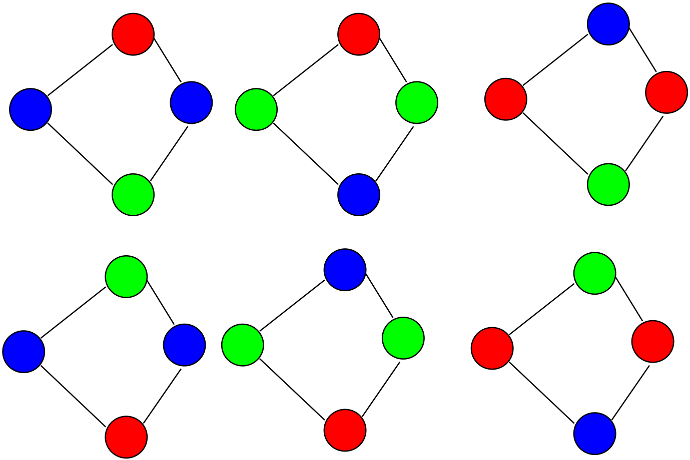
Outline of ideas in the talk

- Try to understand the boundary between tractable and intractable symmetry breaking.
- Specialised search procedures to break symmetry,
- Better $O(n^{??})$ algorithms for symmetry breaking via matching rather than general computational group theory.

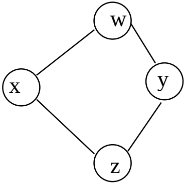
Outline of ideas in the talk

- Try to understand the boundary between tractable and intractable symmetry breaking.
- Specialised search procedures to break symmetry,
- Better $O(n^{??})$ algorithms for symmetry breaking via matching rather than general computational group theory.
- Dealing with conditional symmetry.

What is a Symmetry?



What is a Symmetry?

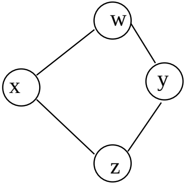


- The most general idea of a symmetry is a bijection that takes *candidate solutions* to *candidate solutions* such that *solutions* are preserved.

A solution is a map from $V = \{w, x, y, z\}$ to $D = \{R(=0), G(=1), B(=2)\}$. A symmetry is a bijection on the set of maps that preserves solutions.

Example: $\sigma(f : V \rightarrow D)(v) = (f(v) + 1) \mod 3$

What is a Symmetry?



- Given any permutation π on $\{R, G, B\}$ then:

$$\sigma_\pi(f : V \longrightarrow D) = \lambda v(\pi(f(v))$$

- Also we can swap around x and y. So

$$\sigma_{x \leftrightarrow y} f(v) = \begin{cases} f(y) & \text{if } v = x \\ f(x) & \text{if } v = y \\ f(v) & \text{otherwise} \end{cases}$$

Types of Symmetry

- A CSP is a set of variables V and a domain of assignments D subject to some constraints. A solution is a map f from V to D such that all constraints are satisfied.

Types of Symmetry

- A CSP is a set of variables V and a domain of assignments D subject to some constraints. A solution is a map f from V to D such that all constraints are satisfied.
- A symmetry group, G, of a CSP acts on the set of maps $\mathrm{Hom}(V, D)$ that preserves solution

$$f \in \mathrm{Sol}(P) \wedge \pi \in G \text{ implies } f^{\pi} \in \mathrm{Sol}(P)$$

Types of Symmetry

- A CSP is a set of variables V and a domain of assignments D subject to some constraints. A solution is a map f from V to D such that all constraints are satisfied.
- A symmetry group, G, of a CSP acts on the set of maps $\mathrm{Hom}(V, D)$ that preserves solution

$$f \in \mathrm{Sol}(P) \wedge \pi \in G \text{ implies } f^{\pi} \in \mathrm{Sol}(P)$$

- This group is too big.

Types of Symmetry

- A CSP is a set of variables V and a domain of assignments D subject to some constraints. A solution is a map f from V to D such that all constraints are satisfied.
- A symmetry group, G, of a CSP acts on the set of maps $\mathrm{Hom}(V, D)$ that preserves solution

$$f \in \mathrm{Sol}(P) \wedge \pi \in G \text{ implies } \; f^{\pi} \in \mathrm{Sol}(P)$$

- This group is too big.
- You can make it smaller by requiring good behaviour of the group action w.r.t. partial maps (getting essentially variable value pairs map to variable value pairs)

Types of Symmetry

- A CSP is a set of variables V and a domain of assignments D subject to some constraints. A solution is a map f from V to D such that all constraints are satisfied.
- A symmetry group, G, of a CSP acts on the set of maps $\mathrm{Hom}(V, D)$ that preserves solution

$$f \in \mathrm{Sol}(P) \wedge \pi \in G \text{ implies } f^{\pi} \in \mathrm{Sol}(P)$$

- This group is too big.
- You can make it smaller by requiring good behaviour of the group action w.r.t. partial maps (getting essentially variable value pairs map to variable value pairs)
- Or simply declare which subgroups you are interested in.

How to Break Symmetry?

Modify the CSP or the search procedure so that the following two conditions are met:

- If there is a solution then the modified CSP has, or the search procedure produces, a symmetrically equivalent solution.
- We only pick out one representative from each orbit.

Sometimes this is too hard so the second condition is relaxed.

What's on the Market?

Adding constraints before search.
Sometimes it is obvious from the problem which constraints to add.
A list of n variables $[x_1, \ldots, x_n]$ representing a set, so all the variables are interchangeable. Then simply assert that:

$$x_1 < x_2 < \cdots < x_n$$

Other symmetry breaking predicates include `lex`, `lex-chain`, `multi-set ordering`.

Lex Constraints - Crawford and Company

When there is a group of variable permutation symmetries, for each variable permutation add a constraint that lexicographically picks out one of the solutions.

For each permutation π of the variables of the problem add the constraint:

$$[x_1, \ldots, x_n] \leq_{\text{lex}} [x_{\pi(1)}, \ldots, x_{\pi(n)}]$$

Example

Consider the following matrix problem

$$\begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix}$$

where rows and columns can be swapped, that is each solution has 4 (that is 3) symmetric solutions:

$$\begin{pmatrix} x_2 & x_1 \\ x_4 & x_3 \end{pmatrix} \quad \begin{pmatrix} x_3 & x_4 \\ x_1 & x_2 \end{pmatrix} \quad \begin{pmatrix} x_4 & x_3 \\ x_2 & x_1 \end{pmatrix}$$

Then add the constraints: $[x_1, x_2, x_3, x_4] \leq [x_2, x_1, x_4, x_3]$, $[x_1, x_2, x_3, x_4] \leq [x_3, x_4, x_1, x_2]$, $[x_1, x_2, x_3, x_4] \leq [x_4, x_3, x_2, x_1]$.

Problems with Lex-leader

- Big groups have many lex-constraints.
- Lots of constraints, large overhead.

Hard Problem: find the smallest set of constraints breaking all symmetry or even small sets breaking a large amount of the symmetry.
(NP-complete in general)
Big groups often have simple symmetry breaking predicates.

Symmetry Breaking During Search

Two main methods on the market.

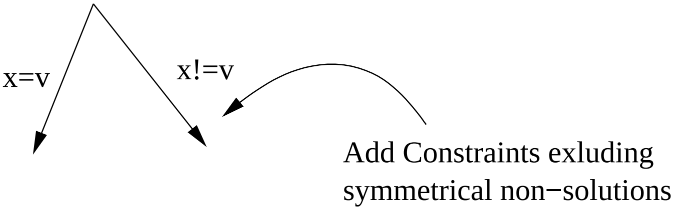
- Add constraints during search that break the symmetry.
- Dominance checking (Global Cuts)

and also

- Modifying the search procedure directly for a specific symmetry group.

Symmetry Breaking During Search

Gent & Smith and Backofen & Will.



The idea is that we take the current no-good and add the symmetric set of no-goods, one for each symmetry in the symmetry group.

Symmetry Breaking by Dominance Checking

Fahle, Schamberger & Sellmann and Focacci & Milano.

- In some sense dual to the SBDS.
- At each level in the search tree check if there is a symmetrically equivalent non-solution.

Of course it is more complicated ...

Types of Symmetry

- We are interested in certain combinations of variable and value symmetries.

Types of Symmetry

- We are interested in certain combinations of variable and value symmetries.
  - A variable symmetry is a bijection $\sigma : V \longrightarrow V$ such that $f : V \longrightarrow D$ is a solution implies that $f \circ \sigma$ is a solution.

Types of Symmetry

- We are interested in certain combinations of variable and value symmetries.
  - A variable symmetry is a bijection $\sigma : V \rightarrow V$ such that $f : V \rightarrow D$ is a solution implies that $f \circ \sigma$ is a solution.
  - A value symmetry is a bijection $\pi : D \rightarrow D$ such that $f : V \rightarrow D$ is a solution implies that $\pi \circ f$ is a solution.

Types of Symmetry

- We are interested in certain combinations of variable and value symmetries.
  - A variable symmetry is a bijection $\sigma : V \to V$ such that $f : V \to D$ is a solution implies that $f \circ \sigma$ is a solution.
  - A value symmetry is a bijection $\pi : D \to D$ such that $f : V \to D$ is a solution implies that $\pi \circ f$ is a solution.
- Combinations f is a solution implies $\pi \circ f \circ \sigma$ is a solution.

Types of Symmetry Groups

We have to start somewhere:

- the symmetric group, consider CSPs where either or both π and σ come from the full symmetric group S_V or S_D.

Types of Symmetry Groups

We have to start somewhere:

- the symmetric group, consider CSPs where either or both π and σ come from the full symmetric group S_V or S_D. (sometimes called full symmetries)

Types of Symmetry Groups

We have to start somewhere:

- the symmetric group, consider CSPs where either or both π and σ come from the full symmetric group S_V or S_D. (sometimes called full symmetries)
- Partial symmetries. Given a partition $P = P_1 \cup \cdots \cup P_n$ study symmetries the preserve the partition $x \in P_i$ implies $x.\pi$ in P_i.

Types of Symmetry Groups

We have to start somewhere:

- the symmetric group, consider CSPs where either or both π and σ come from the full symmetric group S_V or S_D. (sometimes called full symmetries)
- Partial symmetries. Given a partition $P = P_1 \cup \cdots \cup P_n$ study symmetries the preserve the partition $x \in P_i$ implies $x.\pi$ in P_i.
- Various subgroups of wreath products. In the social golfer you can either permute whole weeks or within each week you can permute the groups.

The idea is that these symmetries are easy to understand and capture some interesting problems.

Intractability

- A set valued CSP, $(V, D = 2^U, \{R_i(S_i)\})$ is where variables can be assigned subsets of a domain not just single elements.

Intractability

- A set valued CSP, $(V, D = 2^U, \{R_i(S_i)\})$ is where variables can be assigned subsets of a domain not just single elements.
- A set valued CSP is fully variable and value interchangeable if given $\pi \in S_V$ and $\sigma \in S_U$ and whenever f is a solution then

$$2^\sigma \circ f \circ \pi$$

is a solution.

Intractability

- A set valued CSP, $(V, D = 2^U, \{R_i(S_i)\})$ is where variables can be assigned subsets of a domain not just single elements.
- A set valued CSP is fully variable and value interchangeable if given $\pi \in S_V$ and $\sigma \in S_U$ and whenever f is a solution then

$$2^\sigma \circ f \circ \pi$$

  is a solution.

The dominance checking problem for such CSPs is NP-hard. The proof is not as easy as it first might seem. You have to show the NP-hardness for dominance detecting problems that arise during search.

Tractability

- One strand of research involves specialising the SBDX and seeing what you get.

Tractability

- One strand of research involves specialising the SBDX and seeing what you get.
- With full value symmetry the answer is quite simple.

Tractability

- One strand of research involves specialising the SBDX and seeing what you get.
- With full value symmetry the answer is quite simple.
- You only have to symmetry check against siblings.

Tractability

- One strand of research involves specialising the SBDX and seeing what you get.
- With full value symmetry the answer is quite simple.
- You only have to symmetry check against siblings.
- With full value symmetry the dominance check is very simple.

Tractability

- One strand of research involves specialising the SBDX and seeing what you get.
- With full value symmetry the answer is quite simple.
- You only have to symmetry check against siblings.
- With full value symmetry the dominance check is very simple.
  - At each node if you have used the values $d_1, \ldots, d_k$ then the equivalence classes are:

$$\{d_1\}, \ldots, \{d_k\}, \{d_{k+1}, \ldots, \}$$

  - Pick an previously used value or one new value.

Labelling Algorithm

```
bool fValllabel(𝒫) {
    return fValllabelA(𝒫,ε);
}
bool fValllabelA(⟨V, D, C⟩,θ) {
    if scope(θ) = V then
        return C(θ);
    select v in V \ scope(θ);
    A := image(θ);
    if A ≠ D then
        select f in D \ A; A := A ∪ {f};
    forall(d ∈ A)
        θ′ := θ & v = d;
        if ¬Failure(⟨V, D, C⟩, θ′) then
            if fValllabelA(⟨V, D, C⟩,θ′) then
                return true;
    return false;
```

Value Symmetry: Modified Search Procedures

- Generalises to wreath problems

Value Symmetry: Modified Search Procedures

- Generalises to wreath problems
- Generalised into a much more general context in GE-trees.

Value Symmetry: Modified Search Procedures

- Generalises to wreath problems
- Generalised into a much more general context in GE-trees.
- Although specialised search procedures often have better complexity bounds.

More Tractability: Piecewise value and Piecewise variable interchangeable

- The basic idea is to turn the dominance check into a matching problem.

More Tractability: Piecewise value and Piecewise variable interchangeable

- The basic idea is to turn the dominance check into a matching problem.
- Key idea build signatures the count how many times values appears in variable partitions.

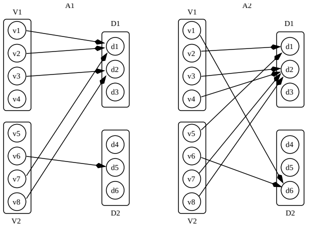
More Tractability: Piecewise value and Piecewise variable interchangeable

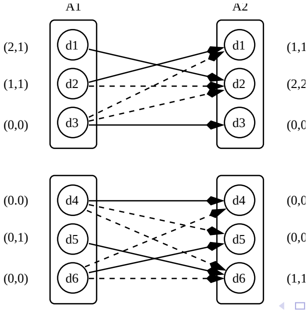
- The basic idea is to turn the dominance check into a matching problem.
- Key idea build signatures the count how many times values appears in variable partitions.
- Define $sig(d_i)$ to be a vector of values of how many times d_i appears in each variable partition.

More Tractability: Piecewise value and Piecewise variable interchangeable

- The basic idea is to turn the dominance check into a matching problem.
- Key idea build signatures the count how many times values appears in variable partitions.
- Define $sig(d_i)$ to be a vector of values of how many times d_i appears in each variable partition.
- Build a bipartite graph the has a perfect matching iff one assignments dominates the other.

Value Symmetry: Modified Search Procedures





Conditional Symmetry

- A conditional symmetry is symmetry with a precondition.
- For example

$$(v_1 = 1) \rightarrow V_2 \text{ and } v_3 \text{ are symmetric}$$

- A set of conditional symmetries forms a groupoid.
- The dominance checking problem with groupoids is much harder than for groups.

References

All of the papers can possibly be found at `http://www.it.uu.se/research/astra` or more likely by following the links from `http://user.it.uu.se/~justin`.