Constraints, and how to satisfy them: Symmetry & Search

Colva M. Roney-Dougal
University of St Andrews

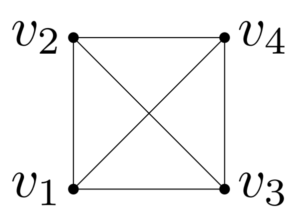
From Higman-Sims to Urysohn

Gracefully labelling a graph

A vertex labelling of a simple connected graph $G = (V, E)$ with distinct integers from $\{0, \ldots, |E|\}$ is **graceful** if

$$\{|v - w| \ : \ \{v, w\} \in E\} = \{1, \ldots, |E|\}.$$

Suppose we wish to gracefully label K_4.



- Variables: v_1, v_2, v_3, v_4.
- Values: $\{0, \ldots, 6\}$.
- Constraints:
  - `AllDifferent`(v_1, v_2, v_3, v_4).
  - $\{v_i, v_j\} \neq \{v_k, v_l\} \in E \Rightarrow |v_i - v_j| \neq |v_k - v_l|$.

Want to find one or all assignments of values to variables that satisfy the constraints.

This is a *constraint satisfaction problem*.

Constraint Satisfaction Problems

Defn: A (finite domain) **constraint satisfaction problem** (CSP) is a triple $P = (V, D, \mathcal{C})$ where:

- V – finite set of **variables**.
- D – finite set of **values**; domain of the variables.
- $\mathcal{C}$ – set of **constraints** on the variables.

Constraint $C = (S, R)$:
S – *scope*, a sequence of variables.
$R \subseteq D^{|S|}$ – permitted values of variables in S.
Constraints normally stated *intensionally*: $x < y$.

Standard applications: airline scheduling, hospital rostering, assigning of tasks to machines in factories.
NP-complete to solve.
Can apply to any combinatorial problem in NP.

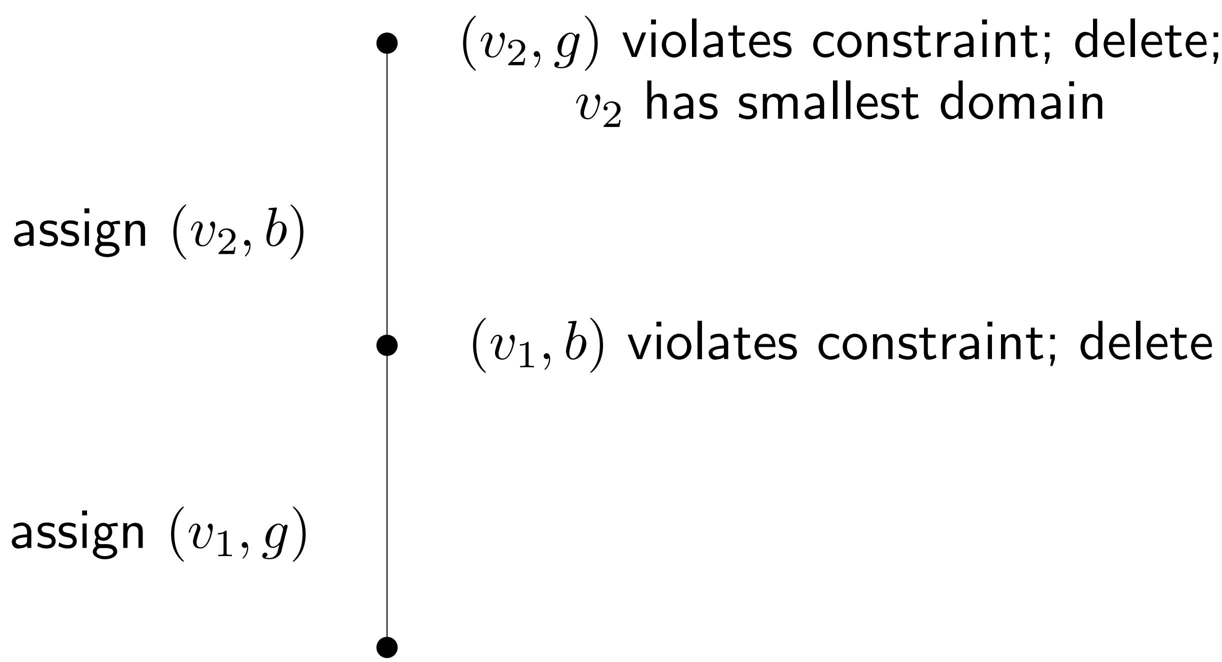
Assignment – $\{(v_1, a_1), \ldots, (v_k, a_k) : v_i \in V, v_i \neq v_j, a_i \in D\}$. **Full** assignment – size $|V|$.

Solution – full assignment satisfying all constraints.

How to solve a CSP

Backtrack search for all, or some, solutions.
Use *inference* to reduce search.

e.g: Colour K_2; colours $\{b, g\}$ for v_1, and $\{b\}$ for v_2.
$V = \{v_1, v_2\}$, $D = \{b, g\}$, $\mathcal{C} = \{v_2 \neq g, v_1 \neq v_2\}$.



Simplify tree to: node $\leftrightarrow$ variable; edge $\leftrightarrow$ value.
In practice search uses binary branching.

Symmetry in CSPs

Recall graceful K_4 problem:

For $1 \leqslant i \leqslant 4$, require $v_i \in \{0, \ldots, 6\}$, s.t. $v_i \neq v_j$ for $i \neq j$ and $|v_i - v_j| \neq |v_k - v_l|$ for $\{i, j\} \neq \{k, l\}$.

Only search inequivalent assignments: all vertices are interchangeable.
Important to minimise search of assignments that don't extend to solutions – most full assignments are non-solutions.

Definition 1 A **full assignment symmetry** is a bijection from the set of full assignments to itself that setwise stabilises the solutions.

- Symmetry group is $\mathrm{Sym}(\text{solutions}) \times \mathrm{Sym}(\text{nonsolutions})$.
- Can't find symmetries until have found some solutions/nonsolutions.
- Only a subset of this leads to an action on partial assignments which can reduce search.

Constraint symmetries

Let symmetries act on **literals**: $\chi := V \times D$.

Definition 2 A **constraint symmetry** is an element of $\mathrm{Sym}(\chi)$ that stabilises the set of constraints.

Interchanging $((v_i, \alpha), (v_j, \alpha))\ \forall\ \alpha \in \{0, \ldots, 6\}$ is a constraint symmetry in graceful K_4 problem.

4 queens Place 4 queens on a 4×4 chessboard so that no two queens attack one another.

Model: $V = \{r_1, r_2, r_3, r_4\} =$ rows;
$D = \{1, 2, 3, 4\} =$ column containing queen;
$\mathcal{C} =$ no two queens in same column or diagonal.

Rotation by 90° maps the first column constraint to a statement that only one queen in r_1.

NOT a constraint symmetry: doesn't even stabilise the set of full assignments.

Status of symmetries depends on model: but some models are better for search.

Solution symmetries

Definition 3 A **solution symmetry** is an element of $\mathrm{Sym}(\chi)$ whose induced action on sets of literals of size $|V|$ setwise stabilises the solutions.

May map assignments that extend to no solution to sets of literals with repeated variables.

If an assignment $\mathcal{A}$ extends to a solution then all images of $\mathcal{A}$ are assignments of size $|\mathcal{A}|$.

Theorem [Cohen–Jeavons–Jefferson–Petrie–Smith'06] The group of constraint symmetries is a subgroup of the group of solution symmetries. Each is the automorphism group of a graph naturally associated with the CSP.

- Hard to find symmetries, work with what you see.
- Often solution symmetries that are not constraint symmetries are most useful during search.
- Let $\mathrm{Aut}(P)$ be group of solution symmetries of P.

Value and variable symmetries

Value symmetry – symmetry preserving partition of χ by variables. $L(P)$ – group of value symmetries.

Pure value symmetry – value symmetry s.t. if $(v, \alpha)f = (v, \beta)$ then $(z, \alpha)f = (z, \beta)$ for all $z \in V$. $PL(P)$ – pure value symmetries, natural induced action on D.

Graceful K_4 problem has pure value symmetry interchanging α with $6 - \alpha$ for $\alpha \in \{0, \ldots, 6\}$.

Variable symmetry – symmetry preserving partition of χ by values. $R(P)$ – group of variable symmetries.

Pure variable symmetry – variable symmetry s.t. if $(v, \alpha)f = (w, \alpha)$ then $(v, \beta)f = (w, \beta)$ for all $\beta \in D$. $PR(P)$ – pure variable symmetries, natural induced action on V.

Graceful K_4 problem has pure variable symmetries permuting vertices.

The structure of $\mathrm{Aut}(P)$

Lemma [Kelsey–Linton–CMRD '04]
1. $PL(P) \leqslant N_{\mathrm{Aut}(P)}(R(P))$.
2. $PR(P) \leqslant N_{\mathrm{Aut}(P)}(L(P))$.
3. $PL(P) \leqslant \{(\sigma, \ldots, \sigma) \in \mathrm{Sym}(D)^{|V|}\} \leqslant \mathrm{Sym}(\chi)$.
4. $L(P) \leqslant \mathrm{Sym}(D)^{|V|} \leqslant \mathrm{Sym}(\chi)$.
5. Similar statements for variable symmetries.
6. $PL(P) \times PR(P) \leqslant \mathrm{Aut}(P)$.

Lemma
B – literals occurring in every solution of P; $\overline{B}$ – literals occurring in none.
$\mathrm{Sym}(B) \times \mathrm{Sym}(\overline{B}) \leqslant \mathrm{Aut}(P)$.

Open problems:
1. Which CSPs have automorphism group generated by value and variable symmetries?
2. Restriction to symmetries mapping assignments to assignments gives product action embedding. Are there any similar embeddings for weaker restrictions?

Breaking symmetries

A Group Equivalence tree (**GE-tree**) for a CSP P with respect to $G \leqslant \mathrm{Aut}(P)$ is any search tree satisfying the following:

(i) No node is isomorphic under G to any other node.

(ii) Given a full assignment $\mathcal{A}$, there is at least one leaf of the tree which lies in $\mathcal{A}^G$.

GE-tree is **minimal** if the deletion of any node (and its descendants) will delete at least one full assignment.

Want to **reduce search** to a GE-tree **without** excessive complexity overhead.

Symmetry breaking during search

Post additional constraints during search to prevent exploration of symmetric partial assignments.

$P = (V, D, \mathcal{C})$, $G \leqslant \mathrm{Aut}(P)$, $\mathcal{A}$ – assignment.

Suppose have explored $\mathcal{A} \wedge (v, a)$, conclude $\mathcal{A} \Rightarrow \neg(v, a)$.
Let $\mathcal{C} := \mathcal{C} \cup \{\mathcal{A}^g \Rightarrow \neg(v, a)^g \ : \ g \in G\}$.

- No restriction on structure of G.
- Produces a GE-tree.
- Results in exponential number of constraints.

Problem: Characterise minimal/small sets of symmetries that restrict search to a GE-tree.

- Breaking generators insufficient.
- Random selection not good. [McDonald–Smith'02]
- If $S_n \cong PR(P) = \mathrm{Aut}(P)$, sufficient to use all transpositions (subject to sensible search order).

A polynomial-time approach for value symmetries
[Gent–Kelsey–Linton–CMRD '04]

Lemma Let $\mathcal{A} = \{(v_i, \alpha_i) \ : \ 1 \leqslant i \leqslant k\}$ be a partial assignment. Then $L(P)_{\{\mathcal{A}\}} = L(P)_{(\mathcal{A})}$.

Assume $G = \mathrm{Aut}(P) = L(P)$. We describe how to construct the inequivalent descendants of $\mathcal{A}$.

1 Select any $v_{k+1} \notin V(\mathcal{A})$.

2 For each orbit of $G_{(\mathcal{A})}$ on $\{(v_{k+1}, \alpha) \ : \ \alpha \in D\}$:

  (a) Select a representative β and add an edge from $\mathcal{A}$ labelled (v_{k+1}, β).

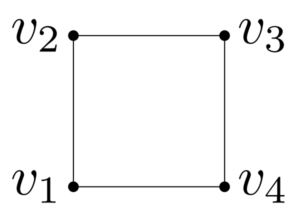
Theorem A tree $\mathbf{T}$ constructed as in the preceding paragraph is a minimal GE-tree.

Theorem The immediate descendants of each node can be constructed in polynomial time.

A polynomial-time approach for `AllDifferent` on variables

Lemma P – CSP with `AllDifferent` on variables. $\mathcal{A}$ – assignment. Then $R(P)_{\{\mathcal{A}\}} = R(P)_{(\mathcal{A})}$.

Example: Suppose that P has $PR(P) = D_{2\times 4} \leqslant \mathrm{Sym}(V)$, and `AllDifferent` on variables.



To break symmetry, first assume that $v_1 < v_2$, $v_1 < v_3$ and $v_1 < v_4$.

After fixing v_1, can still swap v_2 and v_4, so assume $v_2 < v_4$. Transitivity $\Rightarrow$ can remove $v_1 < v_4$.

To break **all** of $D_{2\times 4}$ post: $v_1 < v_2, v_1 < v_3, v_2 < v_4$.

Theorem [Puget '04] Let $P = (V, D, \mathcal{C})$ be a CSP with `AllDifferent` on variables. Then in polynomial time we can construct at most $|V| - 1$ binary $<$ constraints that break all symmetries of $PR(P)$.

Static symmetry breaking without `AllDifferent`

Static – removing symmetry before search, typically by adding constraints. e.g. previous slide.

Sometimes can remodel problem.

Theorem [Crawford–Ginsberg–Luks–Roy '96]
$P = (V, D, \mathcal{C})$; $V = \{v_1, \ldots, v_n\}$; $G \leqslant PR(P)$. Let $\mathcal{C}' = \{v_1 \ldots v_n \leqslant_{\mathrm{lex}} v_{1g^{-1}} \ldots v_{ng^{-1}} \ : \ g \in G\}$.
Then $(V, D, \mathcal{C} \cup \mathcal{C}')$ has one solution for each orbit of solutions of P under G.

- Works for all pure variable symmetries.
- Doesn't restrict search order.
- Produces exponentially many symmetry breaking constraints.

Problem: Characterise minimal sufficient sets of elements of G to break all of G.

Lex constraints – set of lexicographic constraints produced as in theorem.

Complexity issues

Theorem [Luks–Roy '04]
There are an infinite number of pure variable symmetry groups that require exponentially many lex constraints.

Theorem [Luks–Roy '04]
If $|V| = n$ and $PR(P)$ is abelian then there exists an ordering on V s.t. $O(n^3 \log n)$ lex constraints suffice.

Problem: Characterise the groups for which polynomially many lex constraints suffice.
Characterise the optimal variable ordering.

Minimal sets of lex constraints

Lemma [Frisch–Harvey'03/Öhrman'05/Jefferson–Grayland–Miguel–CMRD '07]
v, w – variables; A, B, C, D – strings of variables; $\mathcal{C}$ – lex constraints.
1. If $\mathcal{C} = \mathcal{C}' \cup \{Av \leqslant_{\text{lex}} Bw\}$ and $\mathcal{C}' \cup \{A = B\} \Rightarrow v \leqslant w$ then

$$\mathcal{C} \Leftrightarrow \mathcal{C}' \cup \{A \leqslant_{\text{lex}} B\}.$$

2. If $\mathcal{C} = \mathcal{C}' \cup \{AvB \leqslant_{\text{lex}} CwD\}$ and $\mathcal{C}' \cup \{A = C\} \Rightarrow v = w$ then

$$\mathcal{C} \Leftrightarrow \mathcal{C}' \cup \{AB \leqslant_{\text{lex}} CD\}.$$

Set of lex constraints is **minimal** if these reduction rules do not apply.

Lemma [Jefferson–Grayland–Miguel–CMRD '07]
These rules are not confluent.

Question: Are they confluent for lex constraints for transitive groups? Are there other reduction rules?

Some families of variable symmetries
[Jefferson–Grayland–Miguel–CMRD'07]

Proposition
1. Let $PR(P) = S_n$. A minimal set of lex constraints for $PR(P)$ is $\{v_i \leqslant v_{i+1} \ : \ 1 \leqslant i \leqslant n-1\}$.
2. Let $PR(P) = A_n$. A minimal set of lex constraints for $PR(P)$ is:
$v_i v_{n-1} \leqslant_{\text{lex}} v_{i+1} v_n \ (1 \leqslant i \leqslant n-3)$,
$v_{n-2} \leqslant v_{n-1}$, $v_{n-2} v_{n-1} \leqslant_{\text{lex}} v_n v_{n-2}$.
Minimal sets known for C_n, D_n.

Proposition
Let $G \times H \leqslant PR(P)$, disjoint action on $V = V_1 \cup V_2$.
X – minimal set of lex constraints for G on V_1.
Y – minimal set of lex constraints for H on V_2.
A minimal set of lex constraints for $G \times H$ is $X \cup Y$.
Construction known for imprimitive wreath product – may **not** result in minimality.

Investigating product action wreath product – variable ordering significant.

Happy birthday Peter!