Value Symmetry Breaking

Toby Walsh
NICTA and UNSW

Break symmetry

- LEX LEADER method for variable symmetries [Crawford, Ginsberg, Luks and Roy KR96]
- Post constraints to eliminate all but one solution in each symmetry class
  - $(X[1],X[2],...) \leq_{lex} (X[\sigma(1)],X[\sigma(2)],...)$
  - reversal symmetry:
    $(X[1],...,X[n]) \leq_{lex} (X[n],...,X[1])$

LEX LEADER

- Works with value symmetries
  - $(X[1],X[2],...) \leq_{lex} (\sigma(X[1]),\sigma(X[2]),...)$
- Suppose all values interchangeable
  - LEX LEADER equivalent to VALUE PRECEDENCE

LEX LEADER

- Too many constraints in general
  - For instance, *m* interchangeable values gives *m!* symmetry breaking constraints
  - In general, NP-hard to eliminate all symmetric solutions
  - Look for special types of symmetry where we can do better

Value symmetry

- Polynomial method to eliminate all symmetric solutions [Puget CP05]
  - Introduce variable for first occurrence of each value
    - X[i]=j implies Z[j]≤i
    - Z[j]=i implies X[i]=j
  - Value symmetry on X[i] maps to variable symmetry on Z[j]

Value symmetry

- Break symmetry with LEX LEADER method on introduced Z[j] vars
  - $(Z[1],Z[2],...) \leq_{lex} (Z[\sigma(1)],Z[\sigma(2)],...)$
  - But Z[j] take all different values
  - Hence exists k where $j<k$ implies $j=\sigma(k)$, $k \neq \sigma(k)$, and $Z[k] < Z[\sigma(k)]$

Value symmetry

- Break symmetry with LEX LEADER method on introduced Z[j] vars
  - (Z[1],Z[2],...) $\leq_{lex}$ (Z[σ(1)],Z[σ(2)],...)
  - Simplifies to quadratic number of ordering constraints: Z[k] < Z[σ(k)]
  - By exploiting transitivity, linear number

Value symmetry

- So, we're done?
  - Value symmetry maps onto variable symmetry
  - And this type of variable symmetry (on vars that are all-different) broken with linear number of ordering constraints

Value symmetry

- No!
  - This eliminates all symmetric solutions in polynomial time
  - But eliminating all symmetric subtrees (or equivalently, pruning all symmetric values) is NP-hard [Walsh CP07]

Value symmetry

- Suppose ValSym(Σ,[Z(1),..,Z[n]]) breaks all value symmetry
- i.e. (Z[1],Z[2],...) $\leq_{lex}$ (σ(Z[1]),σ(Z[2]),...) for all σ∈ Σ
- Thm: Checking if ValSym is satisfied (an exponential number of lex constraints) is polynomial [Puget CP05]

Value symmetry

- Suppose ValSym(Σ,[Z(1),..,Z[n]]) breaks all value symmetry
- i.e. $(Z[1],Z[2],...) \leq_{lex} (\sigma(Z[1]),\sigma(Z[2]),...)$ for all $\sigma \in \Sigma$
- Thm: Enforcing domain consistency on ValSym is NP-hard [Walsh CP07]

Pruning symmetric values is NP-hard

- Proof uses reduction from 3SAT
- Consider 3SAT problem {a,-a v b}
- Values represent Boolean truth values
  - 1,2 for a=true, and 3,4 for a=false
  - 5,6 for b=true, and 7,8 for b=false ...

Pruning symmetric values is NP-hard

- Proof uses reduction from 3SAT
- Consider 3SAT problem {a,-a v b}
- Values are (partially) interchangeable
  - 1,2 interchangeable, and 3,4 interchangeable
  - 5,6 interchangeable, and 7,8 interchangeable ...

Pruning symmetric values is NP-hard

- Truth assignment:
  - X[1] $\in$ {1,2,3,4} for truth value for a
  - X[2] $\in$ {5,6,7,8} for truth value for b
- Clauses:
  - X[3] $\in$ {1,2} for clause: a
  - X[4] $\in$ {3,4,5,6} for clause: -a v b
- Switch: X[5] $\in$ {7,8}

Pruning symmetric values is NP-hard

- Constraints:
  - odd(X[5]) implies odd(X[1]) & odd(X[1])
  - odd(X[5]) implies even(X[3]) & even(X[4])
  - odd(X[5]) implies UNSAT
  - even(X[5]) implies SAT
- Note: CSP has symmetric domains and solutions!

Pruning symmetric values is NP-hard

- Suppose we branch on X[5]=7
- Truth assignment:
  - X[1] ∈ {1,2,3,4} for truth value for a
  - X[2] ∈ {5,6,7,8} for truth value for b
- Clauses:
  - X[3] ∈ {1,2} for clause: a
  - X[4] ∈ {3,4,5,6} for clause: -a v b

Pruning symmetric values is NP-hard

- Suppose we branch on X[5]=7
- Forward checking on problem constraints
  - X[1] $\in$ {1,3} for truth value for a
  - X[2] $\in$ {5,7} for truth value for b
- Clauses:
  - X[3] $\in$ {2} for clause: a
  - X[4] $\in$ {4,6} for clause: -a v b

Pruning symmetric values is NP-hard

- Suppose we branch on X[5]=7
- ValSym ensures value precedence
  - X[1] $\in$ {1,3} for truth value for a
  - X[2] $\in$ {5,7} for truth value for b
- Clauses:
  - X[3] $\in$ {2} for clause: a
  - X[4] $\in$ {4,6} for clause: -a v b

Pruning symmetric values is NP-hard

- Suppose we branch on X[5]=7
- ValSym ensures value precedence
  - X[1] $\in$ {1} for truth value for a (=true)
  - X[2] $\in$ {5,7} for truth value for b
- Clauses:
  - X[3] $\in$ {2} for clause: a
  - X[4] $\in$ {4,6} for clause: -a v b

Pruning symmetric values is NP-hard

- Suppose we branch on X[5]=7
- ValSym ensures value precedence
  - X[1] $\in$ {1} for truth value for a (=true)
  - X[2] $\in$ {5} for truth value for b (=true)
- Clauses:
  - X[3] $\in$ {2} for clause: a
  - X[4] $\in$ {6} for clause: -a v b

Pruning symmetric values is NP-hard

- Suppose we branch on X[5]=7
- Solutions of ValSym are models of 3SAT
  - X[1] $\in$ {1} for truth value for a (=true)
  - X[2] $\in$ {5} for truth value for b (=true)
- Hence enforcing domain consistency on ValSym is NP-hard!

Fixed parameter tractability

- Intractability of pruning all symmetric values depends on existence of large number of values
- Thm: Enforcing domain consistency on ValSym takes O(n) time if number of values is fixed

Dynamic methods

- +ve
  - don’t conflict with branching heuristics
- -ve
  - no propagation between problem and symmetry breaking constraints

Dynamic methods

- Static methods can be exponentially faster than dynamic methods
- Thm: exists model of *n* pigeonhole problem that take $O(n^2)$ time with static methods like value precdence but $O(2^n)$ time with dynamic method like GE-tree

Conclusions

- Symmetry breaking is intractable in general
  - Value symmetry is (fixed parameter) tractable if we can bound number of values
  - Static methods for value symmetry can be exponentially faster than dynamic methods

Questions?