# *Symmetry and Optimization*

François Margot
Tepper School of Business
Carnegie Mellon University

February 25, 2010

# *Symmetry and Optimization*

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g(x) \leq 0 \\ & x_i \in \mathbb{Z} \quad \text{for} \quad i \in I \\ & x \in \mathbb{R}^n \end{array}$$

$\pi$ : permutation of $\{1, \ldots, n\}$
$\pi(x) = \pi(x_1, \ldots, x_n) = (x_{\pi(1)}, \ldots, x_{\pi(n)})$

$\pi$ is a symmetry of the problem if

- $x$ feasible $\Leftrightarrow$ $\pi(x)$ feasible
- $f(x) = f(\pi(x))$

# *Special Case: Integer Linear Programming*

Integer Linear Program (ILP):

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \geq b \\ & x \in \{0, \ldots, k\}^n \end{array} \qquad A : m \times n$$

$\pi$ is a symmetry of the ILP if

- $x$ feasible $\Leftrightarrow$ $\pi(x)$ feasible
- $c^T x = c^T \pi(x)$

# *Symmetry Group of the ILP*

Example:

$$
\begin{array}{lrrrrcl}
\min & -x_1 & -x_2 & -x_3 & -x_4 & & \\
\text{s.t.} & x_1 & +x_2 & & +2x_4 & \leq & 2 \\
 & & x_2 & +x_3 & +2x_4 & \leq & 2 \\
 & x_1 & & +x_3 & +2x_4 & \leq & 2
\end{array}
$$

$$x \in \{0,1\}^4$$

Feasible solutions:

$$(x_1, x_2, x_3, x_4) \in \{(0,0,0,0), (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1), (1,1,0,0), (1,0,1,0), (0,1,1,0), (1,1,1,0)\}$$

$G$ : set of all symmetries of the ILP

$$G = \{[1,2,3,4], [1,3,2,4], [2,1,3,4], [3,2,1,4], [2,3,1,4], [3,1,2,4]\}$$

# *Symmetry Group*

$G$ with composition of permutation is a group:

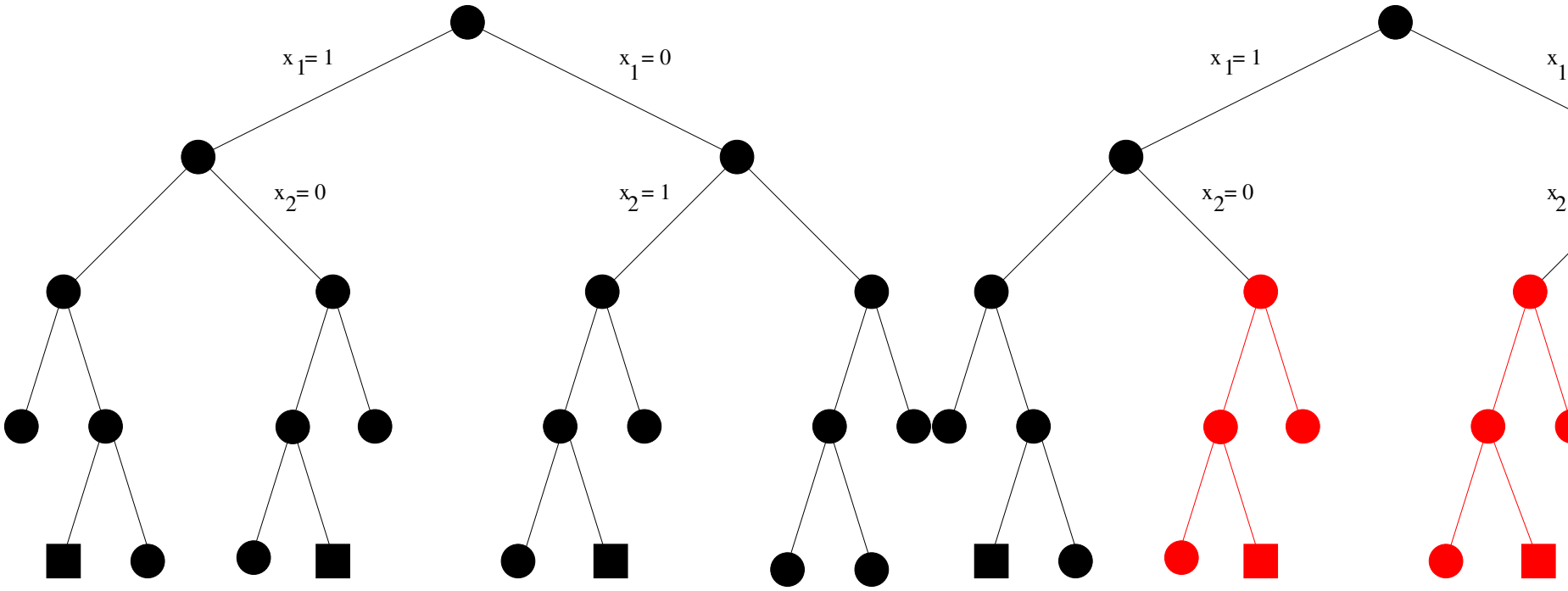
- composition of permutations in $G$ is a permutation in $G$
- $G$ has a neutral element (identity permutation $I$)
- $g \in G \Rightarrow$ there exists $h \in G$ such that $g \cdot h = h \cdot g = I$

Examples of symmetric ILPs:

- Coloring problems
- Network Design (with symmetric network)
- Flexible Manufacturing Operations (with identical machines)
- Design of statistical experiments
- Benchmark problems for QAP, Steiner Tree Problems

# *Branch-and-Bound for ILP*

Branch-and-Bound tree for a symmetric problem: $x_1, x_2$ symmetric




Problem:

$|G|$ large $\quad \Rightarrow \quad$ Most solution techniques become inefficient

# *Problem Characteristics*

| Problem | $n$ | $\hat{z}$ | LP | Group order | Comm. Solver |
|---|---|---|---|---|---|
| $OA_7(7,2,4,7)$ | 128 | – | 112 | 645,120 | 45m |
| $CA_7(7,2,4,7)$ | 128 | 113 | 112 | 645,120 | 2.5h |
| $PA_7(7,2,4,7)$ | 128 | -108 | -112 | 645,120 | $>$ 4h |
| $OA_2(6,3,3,2)$ | 729 | – | 54 | 33,592,320 | $>$ 4h |
| *cov*1054 | 252 | 51 | 50 | 3,628,800 | $>$ 4h |
| *cov*1174 | 330 | 17 | 15.71 | 39,916,800 | $>$ 4h |
| *cod*93 | 512 | -40 | -51.20 | 185,794,560 | $>$ 4h |
| *cod*105 | 1024 | -12 | -18.29 | 371,5891,200 | $>$ 4h |
| *STS*81 | 81 | 61 | 27 | 1,965,150,720 | $>$ 4h |

- "small" number of variables
- "small" integrality gap
- large group

# *Detecting Symmetric ILPs*

Automatic detection is hard:

- Symmetry is a property of the feasible set (empty $\Rightarrow S^n$)
- May consider symmetry of the LP formulation (drawback: easy to destroy)

Generating $G$:

- Known from model
- Compute from formulation : Graph automorphism problem
- Usually, work with a subgroup

# *General Setting and Problems*

Settings:

- Arbitrary symmetry group
- General integer variables

Problems:

- Finding optimal solution
- Optimality proof of known solution
- Finding all nonisomorphic optimal solutions

# *Approaches*

"Exact" Algorithms:

- I: Perturbation
- II: Reformulations (column generation)
- III: Symmetry breaking inequalities
- IV: Symmetry breaking during search
- V: Pruning the enumeration tree

"Approximate" Algorithms:

- VI: Orbital branching
- VII: Dominance relations

Can be used to enumerate all nonisomorphic solution  Use local symmetry information

# *Approach I: Perturbation*

Modify the objective function:

- Add small perturbation to destroy symmetry
  - Counterproductive when trying to prove infeasibility
  - Once optimal solution found, all problems are of this type
- Replace by lexicographic minimization ($c_i = 2^i$ for $i = 1, \ldots n$)
  - Works only for some problems
  - Numerical issues

# *Approach II: Reformulations for $G = \mathcal{S}^n$*

- Column generation
- Dantzig-Wolfe Decomposition

Example: Minimize # of identical machines to perform $m$ tasks

- Column: subset of tasks that can be assigned to a single machine
- Dantzig-Wolfe Decomposition: Minimize # subsets to cover all tasks

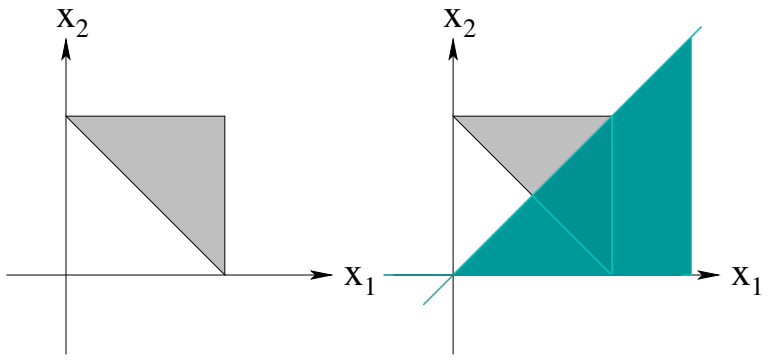
[Barnhart, Johnson, Nemhauser, 1998]

Edge Coloring: [Nemhauser, Park, 1991]
Vertex Coloring: [Mehrotra, Trick, 1995]

# *Approach III: Adding inequalities*

Idea:

- Add inequalities remove some of the symmetry, keeping at least one optimal solution
- Usually: Intersect original formulation with a cone pointed at the origin




Drawbacks:

- Isomorphic solutions may remain feasible
- May create highly fractional LP relaxations

# *III Adding Inequalities (cont.): Orbitopes*

For packing or partitioning problems of the form:

$$\begin{array}{lcll} Ax & \leq & b & \\ \sum_{j=1}^{n} x_{ij} & \{=;\leq\} & 1 & \forall i = 1, \ldots, m \\ x_{ij} & \geq & 0 & \end{array}$$

Collect all variables in a 2-dimensional matrix:

$$X = \begin{array}{|c|c|c|c|c|} \hline x_{1,1} & x_{1,2} & x_{1,3} & \ldots & x_{1,n} \\ \hline x_{2,1} & x_{2,2} & x_{2,3} & \ldots & x_{2,n} \\ \hline \ldots & \ldots & \ldots & \ldots & \ldots \\ \hline x_{m,1} & x_{m,2} & x_{m,3} & \ldots & x_{m,n} \\ \hline \end{array}$$

# *Orbitopes*

If any permutation of the columns of $X$ is a symmetry of the problem:

- a family of symmetry breaking inequalities is known (*shifted column inequalities*)
- polynomial time separation algorithm
- Describes the convex hull of non isomorphic solutions of

$$\begin{array}{lll} \sum_{j=1}^{n} x_{ij} & \{=;\leq\} & 1 \qquad \forall i = 1,\ldots,m \\ x_{ij} & \geq & 0 \end{array}$$

[Kaibel, Pfetsch, 2006]
[Kaibel, Peinhardt, Pfetsch 2007]

# *III (cont.): Asymetric Representatives*

Graph Coloring: Color the vertices of a graph $G = (V, E)$ with the minimum number of colors such that any two adjacent vertices receive distinct colors

Note: Can be generalized to more general partitioning problems

Symmetry: Color permutation

$$z_{ij} = \begin{cases} 1 & \text{if } i \text{ is the representative of } j, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

# *Asymetric Representatives (cont.)*

$$\begin{aligned}
\min \quad & \sum_{i=1}^{s} z_{ii} \\
\text{s.t.} \quad & \sum_{(i,j)\notin E} z_{ij} = 1, && \text{for all } j = 1,\ldots,s, \\
& z_{ij} + z_{ik} \le z_{ii}, && \text{for all distinct } i,j,k, \\
& && \quad \text{with } (i,j),(i,k) \notin E, (j,k) \in E, \\
& z_{ij} \in \{0,1\}, && \text{for all } i = 1,\ldots,s, j = 1,\ldots,s, \\
& z_{ij} = 0, && \text{for all } i = 1,\ldots,s, j = 1,\ldots,s, \text{ with } i > j, \\
& z_{ij} = 0, && \text{for all } (i,j) \in E.
\end{aligned}$$

[Campêlo, Corrêa, Frota, 2004]
[Campêlo, Campos, Corrêa, 2005, 2008]

# *Isomorphism-free backtracking enumeration*

[Butler, Ivanov, Kreher, Lam, Leon, McKay, Read, Stinson]

Example: Solving an ILP with 0, 1 variables:
$a$ : node of the enumeration tree

$$F_1^a = \{i \mid x_i \text{ fixed to } 1 \text{ at } a\}$$

$$F_0^a = \{i \mid x_i \text{ fixed to } 0 \text{ at } a\}$$

Problems at $a$ and $b$ are isomorphic if

$$\exists\, g \in G \text{ with}$$

$$g(F_1^a) = F_1^b \qquad \text{and} \qquad g(F_0^a) = F_0^b$$

$\Rightarrow$ May prune one of $a$ or $b$

[Bazaraa, Kirca, 1983]

# *Approach IV: Symmetry Breaking During Search (SBDS)*

Constraint Programming Approach:

- Add constraints for each created node of the tree to forbid isomorphic ones
- May require huge number of constraints
- Need to keep track of some of the visited nodes

- Symmetry Breaking During Search:
  [Gent, Smith, 2002]
  [Gent, Kelsey, et al. 2005]
- GAP_SBDS: Group representation of the symmetries:
  [Gent, Harvey, Kelsey 2002]
- SBDS-CP-LP hybrid [Petrie, Smith, 2004]

# *Approach V: Pruning*

Assumptions:

- Branch by partitioning the domain of a variable into $k \geq 2$ subdomains
- Complete ordering of the sons (topological in drawing of tree)

Sought: Minimum interference with usual operations

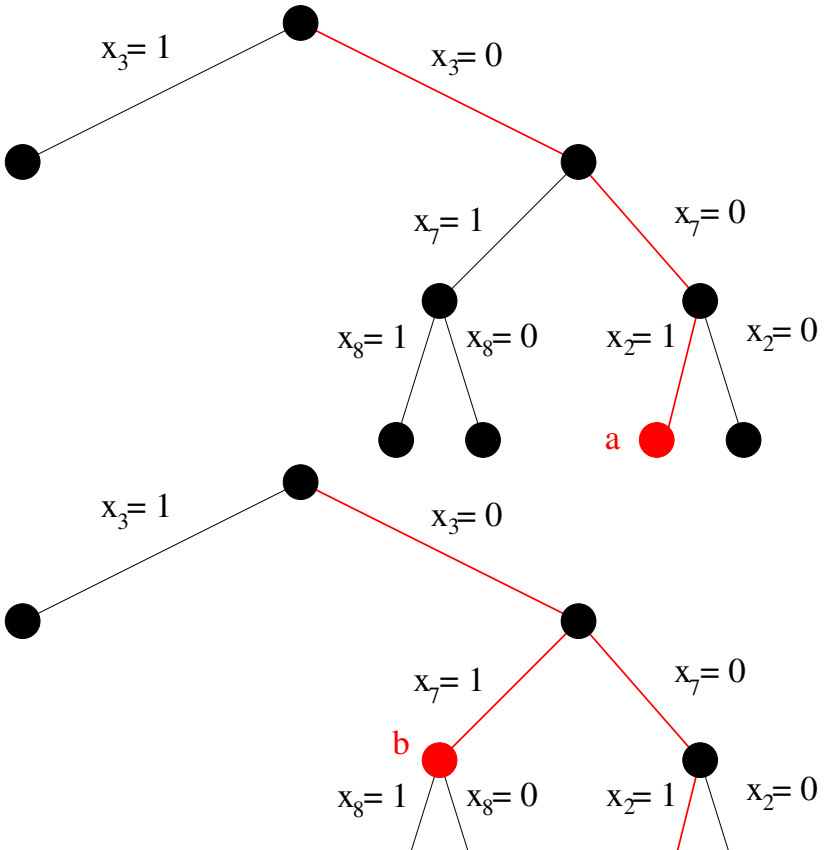
- Freedom to restrict variable domains
- Freedom to choose the branching variable
- Freedom to choose the partitioning
- Pruning uses only information available at a single node of tree

Achievable if:

- Algorithm for restrictions is not be based on symmetry considerations

# *V.I: Left-of-path Mapping*

- $a$: node of the tree
- $F_0^a = \{i \mid x_i \text{ fixed to } 0 \text{ at } a\}$ $F_1^a = \{i \mid x_i \text{ fixed to } 1 \text{ at } a\}$
- Compare fixing at $a$ with left sons issued from ancestors of $a$
- Prune $a$ if there exists $g \in G$ mapping a subset of $F_1^a$ to $F_1^b$ and a subset of $F_0^a$ to $F_0^b$




$F_1^b = \{7\}$ $F_0^b = \{3\}$

$F_1^a = \{2\}$ $F_0^a = \{3, 7\}$

# *V.I: Left-of-path Mapping (cont.)*

- Backtrack Searching with Symmetry (BSS)
  [Brown, Finkelstein, Purdom 1988, 1995]
  Paper Description:
  - General integer variables
  - Branch by creating one son for each possible value of branching variable

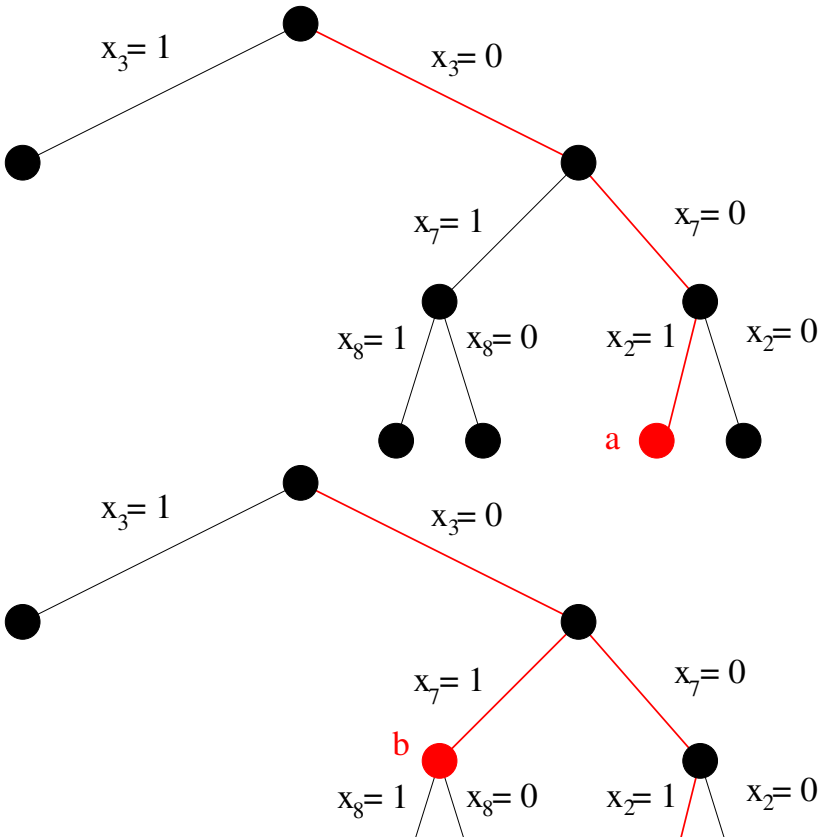
  Implementation:
  - Generic code working for any group
  - Few numerical results
- Symmetry Breaking By Dominance Detection (SBDD)
  [Fahle, Shamberger, Sellman 2001]
  Paper Description:
  - General integer variables
  - Branch by arbitrary partition of domain of branching variable

  Implementation:
  - Ad hoc code for several applications

# *V.II: Lexicomin Support Pruning*

[Butler, Ivanov, Kreher, Lam, Leon, McKay, Read, Stinson]

- $a$: node of the tree
- $F_1^a = \{i \mid x_i \text{ fixed to 1 at } a\}$
- Compare fixing at $a$ with left sons issued from ancestors of $a$
- Prune $a$ if there exists $g \in G$ mapping a subset of $F_1^a$ to $F_1^b$




$F_1^b = \{7\}$

$F_1^a = \{2\}$

# *V.II: Lexicomin Support Pruning (cont.)*

- Isomorphism Pruning (IP)
  [M 2002, 2003, 2003b, 2007]

  Paper Description:
  - General integer variables
  - Branch by creating one son for each possible value of branching variable
  - Rigid branching scheme
    Can be relaxed [Ostrowski 2007]

  Implementation:
  - Generic code working for any group
  - Applications:
    - covering designs [M 2003]
    - orthogonal arrays [Bulutoglu, M 2007]
    - edge coloring [M 2007]
    - codes [Linderoth, Thain, M 2007]

# *Left-of-path Mapping vs. Lexicomin Support*

Bare bone comparison:

- Left-of-path mapping pruning $\subseteq$ Lexicomin Support pruning
- Algorithms based on group representation are backtracking algorithms with depth $|F_1^a \cup F_0^a|$ and $|F_1^a|$ respectively
- Lexicomin Support clear winner

However:

Can set variables to 0 during backtracking. If there exists:

- $F \subseteq F_1^a$
- $b$ a left-ancestor of $a$
- $g \in G$ with $g(F) = F_1^b$

then set to 0 all vars in $g^{-1}(F_0^b)$ (0-setting)

# *Left-of-path Mapping vs. Lexicomin Support*

If full 0-setting is done in Left-of-path mapping then

- Left-of-path mapping pruning = Lexicomin Support pruning
- More variables set to 0 by Left-of-path mapping than with Lexicomin Support
- Much slower Left-of-path mapping checking than Lexicomin Support checking for deep trees

BSS implementation of [Brown, Finkelstein, Purdom 1988, 1995]

- Fast comput. of generators of stabilizer of $(F_1^a \cup j, F_0^a)$ in $G$
- Does not always do full 0-setting

IP implementation of [M 2007]

- Fast computation of one orbit of stabilizer of $F_1^a \cup j$ in $G$
- Does not always do full 0-setting

# *Approach VI: Orbital Branching*

Recompute symmetry group at each node; use orbit information for branching

Orbital Branching:

- Recompute symmetry group for free variables
- Compute partition of free variables into orbits
- Select one orbit $\mathcal{O}$ and one $x_i \in \mathcal{O}$;
- Branch:
  either all vars in $\mathcal{O}$ fixed to 0 or $x_i = 1$

[Ostrowski, Linderoth, Rossi, Smriglio 2006]

# *Orbital Branching vs. Isomorphism Pruning*

Orbital Branching:

- may keep isomorphic solutions
- at some nodes, it may use a larger group than the symmetry group used by isomorphism pruning
- In general slower than Isomorphism Pruning

# *Approach VII: Dominance Relations*

- Use MIP to detect assignment of variables that are dominated
- Limited efficiency for highly symmetric problems

[Fischetti, Toth, 1988]
[Fischetti, Salvagnin, 2007]