Reformulations in Mathematical Programming: Symmetry

Leo Liberti

LIX, École Polytechnique, France
Framework: reformulations

- Given an optimization problem, many different *Mathematical Programming (MP) formulations* can describe its solution set.

- The performances of solution algorithms depend on the MP formulation.

Given an optimization problem and a solution algorithm, what is the MP formulation yielding the best performance?

How do we pass from one formulation to another that keeps some (all) of the mathematical properties of the old formulation?
Storing MP formulations

- Mathematical expressions as \( n \)-ary expression trees

\[
\sum_{i=1}^{3} x_i y_i - \log\left(\frac{x_1}{y_3}\right)
\]

- A formulation \( P \) is a 7-tuple \((\mathcal{P}, \mathcal{V}, \mathcal{E}, \mathcal{O}, \mathcal{C}, \mathcal{B}, \mathcal{T})\) = (parameters, variables, expression trees, objective functions, constraints, bounds on variables, variable types)

- Objectives are encoded as pairs \((d, f)\) where \(d \in \{-1, 1\}\) is the optimization direction and \(f\) is the function being optimized

- Constraints are encoded as triplets \(c \equiv (e, s, b)\) \((e \in \mathcal{E}, s \in \{\leq, \geq, =\}, b \in \mathbb{R})\)

- \(\mathcal{F}(P)\) = feasible set, \(\mathcal{L}(P)\) = local optima, \(\mathcal{G}(P)\) = global optima
Auxiliary problems

If problems $P, Q$ are related by a computable function $f$ through the relation $f(P, Q) = 0$, $Q$ is an auxiliary problem with respect to $P$.

- **Opt-reformulations** (or *exact reformulations*): preserve all optimality properties
- **Narrowings**: preserve some optimality properties
- **Relaxations**: provide bounds to an objective function value towards its optimization direction
- **Approximations**: formulation $Q$ depending on a parameter $k$ such that $\lim_{k \to \infty} Q(k)$ is an opt-reformulation, narrowing or relaxation
Main idea: if we find a global optimum of $Q$, we can map it back to a global optimum of $P$. There may be optima of $P$ without a corresponding optimum in $Q$. 

L., Reformulations in Mathematical Programming: Definitions and Systematics, RAIRO-RO (accepted)
The setting

- Most common solution algorithm for finding global optima: **Branch-and-Bound** (BB for MILPs, sBB for MINLPs)

- **BB (implicit enumeration)**: provides a certificate of optimality in the linear case, and of $\varepsilon$-approximation in the nonlinear case

- **If the problem has symmetries**: many BB nodes will contain (symmetric) optimal solutions $\Rightarrow$ pruning will occur rarely $\Rightarrow$ BB converges slowly

- Need a **reformulation** which is guaranteed to keep at least one global optimum (but hopefully excludes a lot of symmetric optima): a **narrowing**
Minimal group-theoretical toolbox

- Given a group $G$ acting on a set $X$ and $x \in X$, $Gx = \{ gx \mid g \in G \}$ is the orbit of $x$ w.r.t. $G$.

- Given $Y \subseteq X$, the point-wise stabilizer of $Y$ w.r.t. $G$ is a subgroup $H \leq G$ such that $hy = y$ for all $h \in H, y \in Y$.

- The set-wise stabilizer of $Y$ w.r.t. $G$ is a subgroup $H \leq G$ such that $HY = Y$ (denote $H$ by $\text{stab}(Y, G)$).

- Let $\pi \in S_n$ with disjoint cycle product $\sigma_1 \cdots \sigma_k$ and $N \subseteq \{1, \ldots, n\}$.

- Denote $\pi[N] = \prod_{\substack{j \leq k \\& \ \sigma_j \in \text{stab}(N, S_n)}} \sigma_j$. 
Motivating example

Consider an instance $P$:

\[
\begin{align*}
\text{min } & x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} \\
& x_{11} + x_{12} + x_{13} \geq 1 \\
& x_{21} + x_{22} + x_{23} \geq 1 \\
& x_{11} + x_{21} \geq 1 \\
& x_{12} + x_{22} \geq 1 \\
& x_{13} + x_{23} \geq 1
\end{align*}
\]

of the covering prob.

\[
\begin{align*}
\text{min } 1 & \times: \forall i \sum_j x_{ij} \geq 1 \land \forall j \sum_i x_{ij} \geq 1
\end{align*}
\]

The set of solutions is $\mathcal{G}(P) =$

\{(0, 1, 1, 1, 0, 0), (1, 0, 0, 0, 1, 1), (0, 0, 1, 1, 1, 0), (1, 1, 0, 0, 1, 0), (1, 0, 1, 0, 1, 0), (0, 1, 0, 1, 0, 1)\}

$G^* = \text{stab}(\mathcal{G}(P), S_n)$ is the solution group (var. permutations keeping $\mathcal{G}(P)$ fixed setwise)
Symmetries

For the above instance, $G^*$ is
\[ \langle (2, 3)(5, 6), (1, 2)(4, 5), (1, 4)(2, 5)(3, 6) \rangle \cong D_{12} \]

```gap
S := [[0,1,1,1,0,0],[0,1,0,1,0,1],[0,0,1,1,1,0],[1,1,0,0,0,1],[1,0,0,0,1,1],[1,0,1,0,1,0]];
G:=MatrixAutomorphisms(S); StructureDescription(G);
Group([ (2,3)(5,6), (1,2)(4,5), (1,4)(2,5)(3,6) ]); "D12"
```

For all $x^* \in G(P)$, $G^* x^* = G(P) \Rightarrow \exists$ only 1 orbit

$\exists$ only one solution in $G(P)$ (modulo symmetries)

```gap
Orbit(G,S[1],Permuted);
[[0,1,1,1,0,0],[1,1,0,0,0,1],[1,0,0,0,1,1],[0,0,1,1,1,0],[0,1,0,1,1,0]]
Orbit(G,S[2],Permuted);
[[0,1,0,1,0,1],[1,0,0,0,1,1],[0,0,1,1,1,0],[1,0,1,0,1,0],[0,1,1,1,1,0]]
Orbit(G,S[3],Permuted);
[[0,0,1,1,1,0],[0,1,0,1,0,1],[1,0,0,0,1,1],[1,1,0,0,0,1],[0,1,1,1,1,0]]
...
```
Symmetries

This is **bad** for Branch-and-Bound techniques: many branches will contain (symmetric) optimal solutions and therefore will not be pruned by bounding $\Rightarrow$ *deep and large BB trees*

![BB tree diagram]

- If we knew $G^*$ in advance, we might add constraints eliminating (some) symmetric solutions out of $G(P)$

- **Can we find $G^*$ (or a subgroup thereof) *a priori***?

- What constraints provide a valid narrowing of $P$ excluding symmetric solutions of $G(P)$?
Symmetries and formulation

- The cost vector $c^T = (1, 1, 1, 1, 1, 1)$ is fixed by all (column) permutations in $S_6$
- The vector $b = (1, 1, 1, 1, 1)$ is fixed by all (row) permutations in $S_5$
- Consider $P$’s constraint matrix:

$$
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1
\end{pmatrix}
$$

- Let $\pi \in S_6$ be a column permutation such that $\exists$ a row permutation $\sigma \in S_5$ with $\sigma(A\pi) = A$
- Then permuting the variables/columns in $P$ according to $\pi$ does not change the problem formulation
The formulation group

For a MILP with $c = 1_n$ and $b = 1_m$,

$$G_P = \{ \pi \in S_n \mid \exists \sigma \in S_m (\sigma A\pi = A) \}$$

(1)

is called the formulation group of $P$

In the example above, we get $G_P \cong D_{12} \cong G^*$

Thm.

For a covering/packing problem $P$, $G_P \leq G^*$.

Result can be extended to all MILPs [Margot: 2002, 2003 (Math. Prog.); 2007 (DO)]
Related results in MILP

- **Isomorphism pruning** [Margot 02,03], involves addition of linear inequalities of packing type *locally* to selected nodes of the BB tree (as well as var. fixing)

- **Orbitopes** [Kaibel et al. 07,08]: “polytopes modulo symmetries” for $C_n$ and $S_n$ groups only

- **Fundamental domains** [Friedman 07]: given a (discrete) domain $X$ and a group $G$ acting on $X$, a fundamental domain is a subset $F$ of $X$ such that $GF = X$ (determination of smallest FDs w.r.t. given ordering vectors $c$)

- **Orbital branching** [Ostrowski et al. 07,08] branching scheme taking advantage of the problem group (yields fewer branching disjunctions)
Related results in CP

- Much more work in CP than in MILP
- **Definitions**: Cohen et al., *Symmetry Definitions for Constraint Satisfaction Problems*, CP 2005. Relations between constraint and solution groups

---

**Survey**


---

In general, current literature on symmetry in mathematical programming concentrates on MILPs/SDPs and assumes the formulation group is given *a priori*
My contributions

1. **MILPs** (COCOA08 paper):
   - A MILP-based method for finding subgroups of the problem group
   - Some static symmetry-breaking constraints (narrowing reformulation)

2. **MINLPs** (new material):
   - Definition of the problem group
   - Reduction to **GRAPH ISOMORPHISM**
   - Orbit-based static symmetry-breaking constraints (narrowing reformulation)
Symmetries in MINLPs

Consider the following MINLP \( P \):

\[
\begin{align*}
\min & \ f(x) \\
g(x) & \leq 0 \\
x & \in X.
\end{align*}
\]

(2)

where \( X \) may contain integrality constraints on \( x \)

For a row permutation \( \sigma \in S_m \) and a column permutation \( \pi \in S_n \), we define \( \sigma P \pi \) as follows:

\[
\begin{align*}
\min & \ f(x\pi) \\
\sigma g(x\pi) & \leq 0 \\
x\pi & \in X.
\end{align*}
\]

(3)

Define \( G_P = \{ \pi \in S_n \mid \exists \sigma \in S_m \ (\sigma P \pi = P) \} \)
Representing $g(x\pi)$

In the linear case, writing $Ax\pi$ is easy — how do we deal with $g(x\pi)$?

How do we decide whether $g_i(x) = g_h(x\pi)$ for $i, h \leq m$?

**Answer:** consider the expression DAG representation of $g$

\[
\sum_{i=1}^{3} x_i y_i - \log(x_3/y_3)
\]

List of expressions $\equiv$ expression DAG sharing variable leaf nodes

Every function $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is represented by a DAG whose leaf nodes are variables and constants and whose intermediate nodes are mathematical operators

Look for relationships between the DAGs representing $g(x)$ and $\sigma g(x\pi)$
Example

\[ C_0 : x_6 x_7 + x_8 x_9 = 1 \]
\[ C_1 : x_6 x_8 + x_7 x_9 = 1 \]

\[ G_{DAG} = \text{set of automorphisms of expression DAG fixing: (a) root node set having same constr. direction and coeff. (constraint permutations), (b) operators with same label and rank and (c) leaf node set (variable permutations)} \]

Dreadnaut version 2.4 (32 bits).

> n=10 g 2 3; 4 5; 6 7; 8 9; 6 8; 7 9. f=[0:1|2:5|6:9] x

(4 5)(6 7)(8 9)  !variable permutations
(2 3)(6 8)(7 9)  !operator permutations
(0 1)(2 4)(3 5)(7 8)  !constraint permutation

\[ G_P \text{ is the projection of } G_{DAG} \text{ to variable indices} \]

\[ \langle (6, 7)(8, 9), (6, 8)(7, 9), (7, 8) \rangle \cong D_8 \]
Colors on the DAG nodes are used to identify those subsets of nodes which can be permuted

1. Root nodes (i.e. constraints) can be permuted if they have the same RHS
2. Operator nodes (including root nodes) can be permuted if they have the same DAG rank and label
3. If an operator node is non-commutative, then the order of the children node must be maintained
4. Constant nodes can be permuted if they have the same DAG rank level and value
5. Variable nodes can be permuted if they have the same bounds and integrality constraints
Node colors 2

Formalize by equivalence relations on sets: $R =$ roots, $O =$ operators, $C =$ constants, $V =$ variables

Let $V$ be the set of all nodes of the DAG; for all $x, y \in V$:

1. $x \sim_R y$ if $x, y \in R \land \text{RHS}(x) = \text{RHS}(y)$ or $x, y \notin R$
2. $x \sim_O y$ if $x, y \in O \land \text{level}(x) = \text{level}(y) \land \text{label}(x) = \text{label}(y) \land (\text{order}(x) = \text{order}(y)$ if $x, y$ noncommutative) or $x, y \notin O$
3. $x \sim_C y$ if $x, y \in C \land \text{value}(x) = \text{value}(y) \land \text{level}(x) = \text{level}(y)$ or $x, y \notin C$
4. $x \sim_V y$ if $x, y \in V \land \text{limits}(x) = \text{limits}(y) \land \text{integer}(x) = \text{integer}(y)$ or $x, y \notin V$

Define an integral function $\text{color} : V \to \mathbb{N}$ s.t. $\forall x, y \in V \ (\text{color}(x) = \text{color}(y) \iff x \sim_R y \land x \sim_O y \land x \sim_C y \land x \sim_V y)$

$\text{color}$ is itself an equivalence relation (call it $\sim$) and partitions $V$ in disjoint sets $V_1, \ldots, V_p$
MINLP problem groups

- Let $P$ be a MINLP and $D = (\mathcal{V}, \mathcal{A})$ be the DAG of $P$
- Let $G_{\text{DAG}}$ be the group of automorphisms of $D$ that fix each class in $\mathcal{V}/\sim$
- Define $\phi : G_{\text{DAG}} \to S_n$ by $\phi(\pi) =$ permutation on $\mathcal{V}$ (set of variable nodes) induced by $\pi$; then

**Thm.**

$\phi$ is a group homomorphism and $\text{Im} \phi \cong G_P$

- Hence can find $G_P$ by computing $\text{Im} \phi$
- Although the complexity status ($\text{P/NP}$-complete) of the $\text{GRAPH ISOMORPHISM}$ problem is currently unknown, $\text{nauty}$ is a practically efficient software for computing $G_{\text{DAG}}$
- Also, MILPs are MINLPs! (can apply same methods)
## Symmetries in the MIPLib3

<table>
<thead>
<tr>
<th>Instance</th>
<th>$G_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>air03.mod</td>
<td>$(C_2)^{13}$</td>
</tr>
<tr>
<td>arki001.mod</td>
<td>$S_3^8$</td>
</tr>
<tr>
<td>enigma.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>gen.mod</td>
<td>$(C_2)^2 \times D_8 \times S_6$</td>
</tr>
<tr>
<td>fiber.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>harp2.mod</td>
<td>$C_2 \times C_2$</td>
</tr>
<tr>
<td>mas74.mod</td>
<td>$C_2 \times C_2$</td>
</tr>
<tr>
<td>mas76.mod</td>
<td>$S_3^2 \times (S_3)^2 \times S_4$</td>
</tr>
<tr>
<td>misc03.mod</td>
<td>$(C_2)^7$</td>
</tr>
<tr>
<td>misc06.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>misc07.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>mitre.mod</td>
<td>$(C_2)^3 \times (S_3)^3$</td>
</tr>
<tr>
<td>noswot.mod</td>
<td>$(C_2)^7$</td>
</tr>
<tr>
<td>nw04.mod</td>
<td>$C_2 \times S_4$</td>
</tr>
<tr>
<td>p0201.mod</td>
<td>$S_5$</td>
</tr>
<tr>
<td>p0282.mod</td>
<td>$S_5$</td>
</tr>
<tr>
<td>p0548.mod</td>
<td>$((C_3)^3 \rtimes PSL(3, 3)) \rtimes C_2$</td>
</tr>
<tr>
<td>p2756.mod</td>
<td>$S_7^9$</td>
</tr>
<tr>
<td>qiu.mod</td>
<td>$S_{48}$</td>
</tr>
<tr>
<td>rgn.mod</td>
<td>$(S_3)^2 \times S_4 \times S_5$</td>
</tr>
<tr>
<td>rout.mod</td>
<td></td>
</tr>
<tr>
<td>stein27.mod</td>
<td></td>
</tr>
<tr>
<td>swath.mod</td>
<td></td>
</tr>
<tr>
<td>vpm1.mod</td>
<td></td>
</tr>
<tr>
<td>vpm2.mod</td>
<td></td>
</tr>
</tbody>
</table>

All others: $G_P = \{e\}$

All instances have been pre-solved by AMPL
## Symmetries in the MIPLib2003

<table>
<thead>
<tr>
<th>Instance</th>
<th>$G_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>arki001.mod</td>
<td>$S_{48}$</td>
</tr>
<tr>
<td>fiber.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>glass4.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>mas74.mod</td>
<td>$C_2 \times C_2$</td>
</tr>
<tr>
<td>mas76.mod</td>
<td>$C_2 \times C_2$</td>
</tr>
<tr>
<td>misc07.mod</td>
<td>$S_3$</td>
</tr>
<tr>
<td>mzzv11.mod</td>
<td>$(C_2)^{155}$</td>
</tr>
<tr>
<td>mzzv42z.mod</td>
<td>$(C_2)^{110}$</td>
</tr>
<tr>
<td>noswot.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>opt1217.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>p2756.mod</td>
<td>$(C_2)^{32}$</td>
</tr>
<tr>
<td>protfold.mod</td>
<td>$(C_2)^2$</td>
</tr>
<tr>
<td>qiu.mod</td>
<td>$C_2 \times S_4$</td>
</tr>
<tr>
<td>rout.mod</td>
<td>$S_5$</td>
</tr>
<tr>
<td>timtabl1.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>timtab2.mod</td>
<td>$C_2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>mkc.mod</td>
<td>RAM (36 gens)</td>
</tr>
<tr>
<td>seymour.mod</td>
<td>RAM (78 gens)</td>
</tr>
<tr>
<td>swath.mod</td>
<td>RAM (922 gens)</td>
</tr>
<tr>
<td>atlanta-ip</td>
<td>CPU time</td>
</tr>
<tr>
<td>dano3mip</td>
<td>CPU time</td>
</tr>
<tr>
<td>mod011.mod</td>
<td>CPU time</td>
</tr>
<tr>
<td>sp97ar.mod</td>
<td>CPU time</td>
</tr>
<tr>
<td>t1717.mod</td>
<td>CPU time</td>
</tr>
</tbody>
</table>

All others: $G_P = \{e\}$

**AMPL presolver disabled**
Symmetries in the MINLPLib

<table>
<thead>
<tr>
<th>Instance</th>
<th>$G_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cecil13.mod</td>
<td>$(C_2)^9$</td>
</tr>
<tr>
<td>elf.mod</td>
<td>$S_3$</td>
</tr>
<tr>
<td>gastrans.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>gear.mod</td>
<td>$D_8$</td>
</tr>
<tr>
<td>gear2.mod</td>
<td>$D_8$</td>
</tr>
<tr>
<td>gear3.mod</td>
<td>$D_8$</td>
</tr>
<tr>
<td>gear4.mod</td>
<td>$C_2$</td>
</tr>
<tr>
<td>hmittelman.mod</td>
<td>$(C_2)^7 \times S_{762}$</td>
</tr>
<tr>
<td>lop97icx.mod</td>
<td>$S_6$</td>
</tr>
<tr>
<td>nuclear14.mod</td>
<td>$S_6$</td>
</tr>
<tr>
<td>nuclear24.mod</td>
<td>$S_5$</td>
</tr>
<tr>
<td>nuclear25.mod</td>
<td>$S_7$</td>
</tr>
<tr>
<td>nuclear49.mod</td>
<td>$S_3$</td>
</tr>
<tr>
<td>nuclearva.mod</td>
<td>$S_3$</td>
</tr>
<tr>
<td>nuclearvb.mod</td>
<td>$S_3$</td>
</tr>
<tr>
<td>nuclearvc.mod</td>
<td>$S_3$</td>
</tr>
<tr>
<td>nuclearvd.mod</td>
<td>$S_3$</td>
</tr>
<tr>
<td>nuclearve.mod</td>
<td>$S_3$</td>
</tr>
<tr>
<td>nuclearvf.mod</td>
<td>$S_{10}$</td>
</tr>
<tr>
<td>nvs09.mod</td>
<td>$(C_2 \times S_3 \times S_6 \times S_{13})^3$</td>
</tr>
<tr>
<td>product.mod</td>
<td>$(C_2)^9 \times (S_3)^2$</td>
</tr>
<tr>
<td>risk2b.mod</td>
<td>$(C_2)^9 \times (S_3)^2$</td>
</tr>
<tr>
<td>super2.mod</td>
<td>$S_4$</td>
</tr>
<tr>
<td>super3.mod</td>
<td></td>
</tr>
<tr>
<td>synheat.mod</td>
<td></td>
</tr>
</tbody>
</table>

All others: $G_P = \{e\}$

<table>
<thead>
<tr>
<th>Instance</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>qap.mod</td>
<td>CPU time</td>
</tr>
<tr>
<td>qapw.mod</td>
<td>CPU time</td>
</tr>
</tbody>
</table>

All instances have been pre-solved by AMPL.
Breaking symmetries

**Defn.**

Given a permutation $\pi \leq S_n$ acting on the component indices of the vectors in a given set $X \subseteq \mathbb{R}^n$, the constraints $g(x) \leq 0$ (that is, $\{g_1(x) \leq 0, \ldots, g_q(x) \leq 0\}$) are symmetry breaking constraints (SBCs) with respect to $\pi$ and $X$ if there is $y \in X$ such that $g(y\pi) \leq 0$.

![Diagram](image)

$g(y) \nleq 0 \quad g(y\pi) \leq 0$

Usually $y\pi$ is an optimum, but not all optima satisfy the SBCs.

**Defn.**

Given a group $G$, $g(x) \leq 0$ are SBCs w.r.t $G$ and $X$ if there is $y \in XG$ such that $g(y) \leq 0$. 
SBCs and narrowings

Adjoining SBCs to an MP formulation provides a valid narrowing

Thm.

If \( g(x) \leq 0 \) are SBCs for any subgroup \( G \) of \( GP \) and \( G(P) \), then the problem \( Q \) obtained by adjoining \( g(x) \leq 0 \) to the constraints of \( P \) is a narrowing of \( P \).

**Notation:** \( g[B](x) \leq 0 \) if \( g(x) \) only involve variable indices in \( B \)

Conditions allowing adjunctions of many SBCs

Thm.

Let \( \omega, \theta \subseteq \{1, \ldots, n\} \) be such that \( \omega \cap \theta = \emptyset \). Consider \( \rho, \sigma \in GP \), and let \( g[\omega](x) \leq 0 \) be SBCs w.r.t. \( \rho, G(P) \) and \( h[\theta](x) \leq 0 \) be SBCs w.r.t. \( \sigma, G(P) \). If \( \rho[\omega], \sigma[\theta] \in GP[\omega \cup \theta] \) then the system of constraints \( c(x) \leq 0 \) consisting of \( g[\omega](x) \leq 0 \) and \( h[\theta](x) \leq 0 \) is an SBC system for \( \rho\sigma \).
Let $\Omega$ be the set of nontrivial orbits of the regular action of $G_P$ on $\{1, \ldots, n\}$

**Thm.**

Let $\omega \in \Omega$. The constraints

$$\forall j \in \omega \setminus \{\text{min}\ \omega\} \quad x_{\text{min}\ \omega} \leq x_j.$$  

are SBCs with respect to $G_P$.

**Lemma:** $G[\omega]$ is the transitive constituent of $G$ on its orbit $\omega$

**Thm.**

Let $\omega^- = \omega \setminus \{\text{max}\ \omega\}$ and for all $j \in \omega^- (j^+ = \min\{h \in \omega \mid h > j\})$. Provided $G_P[\omega] = \text{Sym}(\omega)$, the following constraints:

$$\forall j \in \omega^- \quad x_j \leq x_{j^+}$$

are SBCs with respect to $G_P$. 
Automatic SBC generation

1. Transform MINLP from AMPL input format into a DAG representation (ROSE)
2. Compute node colors according to relation $\sim$ defined above (ROSE)
3. Compute $G_{DAG}$ (nauty)
4. Compute $\text{Im}\phi$ (gap)
5. Compute nontrivial orbits $\Omega$ (gap)
6. Generate SBCs (4) or (5) according to the structure of $G_P[\omega]$, where $\omega$ is the longest orbit in $\Omega$ (gap)
7. If conditions hold, try to generate compatible SBCs from other orbits (gap)

ROSE=Reformulation/Optimization Software Engine; nauty=Graph Isomorphism software; gap=Group Theory software; data flow provided by Unix scripts
Tests

Computed group structures for 669 instances in MIPLib3 ∪ MIPLib2003 ∪ GlobalLib ∪ MINLPLib

Out of 18% instances with nontrivial groups, 74 could be solved by BB algorithms (CPLEX for MILPs; Couenne, BARON for (MI)NLPs)

\textit{Narrowing1}: only use (4) for longest orbit

\textit{Narrowing2}: also use (5) and attempt to combine SBCs if possible

Test 1: over all instances

Test 2: over a selection of 6 difficult instances with long BB runs

<table>
<thead>
<tr>
<th>T.</th>
<th>CPU</th>
<th>Best gap</th>
<th>Nodes</th>
<th>CPU</th>
<th>Best gap</th>
<th>Nodes</th>
<th>CPU</th>
<th>Best gap</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>157263</td>
<td>2.26E4%</td>
<td>21.44M</td>
<td>152338</td>
<td>2.26E4%</td>
<td>14.23M</td>
<td>153470</td>
<td>2.26E4%</td>
<td>15.72M</td>
</tr>
<tr>
<td>2</td>
<td>815018</td>
<td>242.88%</td>
<td>12.26M</td>
<td>888089</td>
<td>219.14%</td>
<td>14.63M</td>
<td>786406</td>
<td>217.05%</td>
<td>11.28M</td>
</tr>
</tbody>
</table>
## Test 2

<table>
<thead>
<tr>
<th>Instance</th>
<th>Slv</th>
<th>CPU</th>
<th>( f^* ) gap</th>
<th>nodes tree</th>
<th>CPU</th>
<th>( f^* ) gap</th>
<th>nodes tree</th>
<th>CPU</th>
<th>( f^* ) gap</th>
<th>nodes tree</th>
<th>R.t.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MILPLib(s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mkc(a)</td>
<td></td>
<td></td>
<td>-563.846%</td>
<td>1945500</td>
<td></td>
<td>-563.846%</td>
<td>2104500</td>
<td></td>
<td>-</td>
<td></td>
<td>2712.33</td>
</tr>
<tr>
<td>protfold(b)</td>
<td></td>
<td>30.51%</td>
<td>592000</td>
<td></td>
<td>16.54%</td>
<td>353823</td>
<td></td>
<td>-</td>
<td></td>
<td>592.14</td>
<td></td>
</tr>
<tr>
<td>seymour(a)</td>
<td></td>
<td></td>
<td>3992700</td>
<td>423</td>
<td></td>
<td>4343500</td>
<td>3960700</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>283311</td>
<td>262817</td>
<td></td>
<td>3038821</td>
<td>233643</td>
<td></td>
<td>1.0%</td>
<td>3064665</td>
<td>5.95</td>
</tr>
<tr>
<td>GlobalLib</td>
<td></td>
<td></td>
<td>-3500</td>
<td>5452500</td>
<td></td>
<td>-3500</td>
<td>7373700</td>
<td></td>
<td>-3500</td>
<td>4425400</td>
<td>1.40</td>
</tr>
<tr>
<td>ex5_2_5(a)</td>
<td></td>
<td></td>
<td>1259853</td>
<td>19805</td>
<td></td>
<td>18.5%</td>
<td>747262</td>
<td></td>
<td>17.41%</td>
<td>1076927</td>
<td></td>
</tr>
<tr>
<td>maxmin(a)</td>
<td></td>
<td></td>
<td>-0.366</td>
<td>237100</td>
<td></td>
<td>-0.366</td>
<td>238000</td>
<td></td>
<td>-</td>
<td></td>
<td>1.29</td>
</tr>
<tr>
<td>MINLPLib</td>
<td></td>
<td></td>
<td>150803</td>
<td>58643</td>
<td></td>
<td>144%</td>
<td>150355</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lop97icx(a)</td>
<td></td>
<td></td>
<td>27824</td>
<td>26903</td>
<td></td>
<td>39.14%</td>
<td>27189</td>
<td></td>
<td></td>
<td>14708</td>
<td>24.96</td>
</tr>
</tbody>
</table>

IMA MINLP Workshop, Nov. 2008 – p. 30
The KNP group

**Kissing Number Problem** (decision version): Given integers $D, N > 1$, can $N$ unit spheres be adjacent to a given unit sphere in $\mathbb{R}^d$?

**Formulation:**

\[
\begin{align*}
\max_{x, \alpha} & \quad \alpha \\
\forall i \leq N & \quad ||x_i||^2 = 1 \\
\forall i < j \leq N & \quad ||x_i - x_j||^2 \geq \alpha \\
\alpha & \in [0, 1], \forall i \leq N \ x_i \in [-1, 1]^D
\end{align*}
\]

If $\alpha \geq 1$, answer YES, otherwise NO

*The group $\text{Aut}(G(P))$ has infinite (uncountable) cardinality: each feasible solution can be rotated by any angle in $\mathbb{R}^D$; however, the problem group $G_P$ is finite (permutations of spheres and/or dimensions)*

Conjecture (formulated by software): $G_P \cong S_D$

Rewrite constraint: $||x_i - x_j||^2 = \sum_{k \leq D} (x_{ik} - x_{jk})^2 = \sum_{k \leq D} (x_{ik}^2 + x_{jk}^2 - 2x_{ik}x_{jk}) = 2(D - \sum_{k \leq D} x_{ik}x_{jk})$

**Conjecture becomes:** $G_P \cong S_D \times S_N$ (eventually proved correct)
Some KNP results

<table>
<thead>
<tr>
<th>Instance</th>
<th>Siv</th>
<th>CPU</th>
<th>$f^*$ gap</th>
<th>nodes tree</th>
<th>CPU</th>
<th>$f^*$ gap</th>
<th>nodes tree</th>
<th>CPU</th>
<th>$f^*$ gap</th>
<th>nodes tree</th>
<th>R.t.</th>
</tr>
</thead>
<tbody>
<tr>
<td>knp-6.2a</td>
<td>B</td>
<td>8.66</td>
<td>-1</td>
<td>1118</td>
<td>36000</td>
<td>-1</td>
<td>131</td>
<td>1.91</td>
<td>-1</td>
<td>186</td>
<td>1.43</td>
</tr>
<tr>
<td>knp-7.2</td>
<td>B</td>
<td>147.21</td>
<td>-0.753</td>
<td>13729</td>
<td>154.9</td>
<td>-0.753</td>
<td>14182</td>
<td>3.86</td>
<td>-0.753</td>
<td>260</td>
<td>1.47</td>
</tr>
<tr>
<td>knp-12.3</td>
<td>B</td>
<td>36000</td>
<td>-1.105</td>
<td>299241</td>
<td>36000</td>
<td>-1.105</td>
<td>241361</td>
<td>36000</td>
<td>-1.105</td>
<td>273923</td>
<td>3.39</td>
</tr>
<tr>
<td>knp-13.3</td>
<td>B</td>
<td>36000</td>
<td>-0.914</td>
<td>102150</td>
<td>36000</td>
<td>-0.914</td>
<td>97709</td>
<td>36000</td>
<td>-0.914</td>
<td>68248</td>
<td>3.38</td>
</tr>
<tr>
<td>knp-24.4</td>
<td>B</td>
<td>36000</td>
<td>-0.966</td>
<td>10156</td>
<td>36000</td>
<td>-0.966</td>
<td>11394</td>
<td>36000</td>
<td>-0.92</td>
<td>4059</td>
<td>5.62</td>
</tr>
</tbody>
</table>

Indicator:

\[
\frac{1}{4} \left( \frac{1}{CPU} + \frac{1}{GAP+1} + \frac{\text{nodes tree}}{} \right)
\]

“bigger is better”
The end

Thank you