Symmetry in Mathematical Programming

Leo Liberti

IBM Research, Yorktown Heights, USA

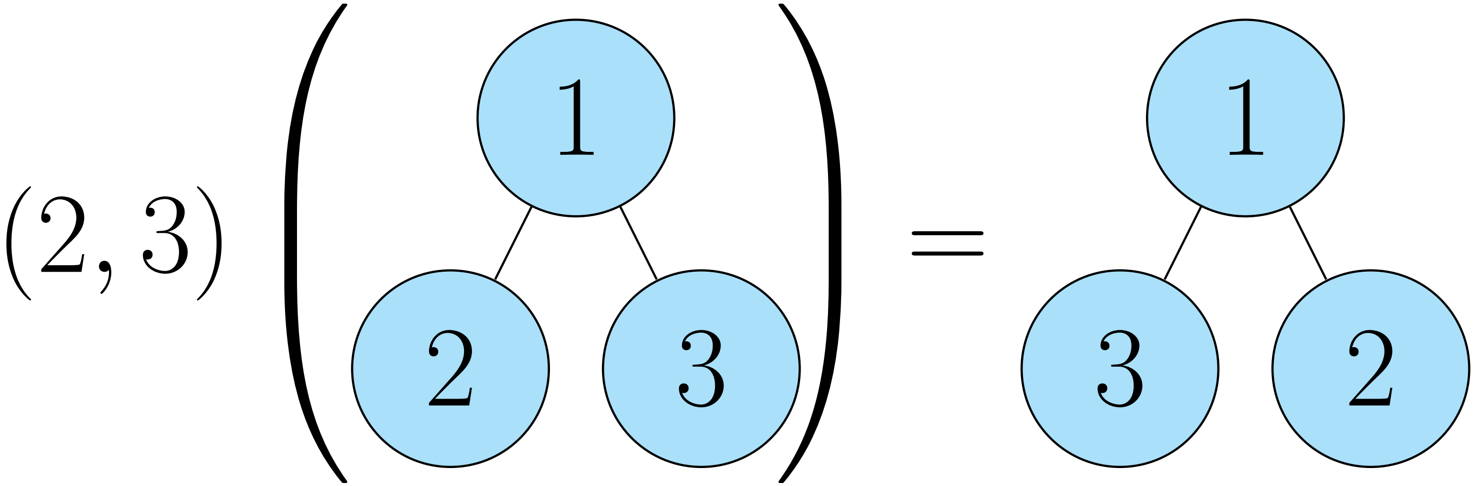
LIX, Ecole Polytechnique, Palaiseau, France

Summary

1. What is a symmetric mathematical program?
2. Detecting symmetries
   - Drawing mathematical programs
   - From instances to problems
3. Exploiting symmetries
   - The `sb3` reformulation
   - The `sbstab` reformulation
   - Orbital shrinking

What is a symmetric MP?

Permuting vertices in graphs




```
# apply permutation to edge set
gap> OnSetsSets([[1,2],[1,3]], (2,3));
[ [ 1, 2 ], [ 1, 3 ] ]
```

Action on the incidence matrix

$$(2,3)\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

the edge order is irrelevant

$$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}(1,2) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Notation

(*column permutation*)(MATRIX)(*row permutation*) = (PERMUTED MATRIX)

$$(2,3)\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}(1,2) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

```
# apply permutation to columns
gap> J := [[1,1,0],[1,0,1]];
gap> Apply(J, r->Permuted(r, (2,3))); J;
[ [ 1, 0, 1 ], [ 1, 1, 0 ] ]
# apply permutation to rows
gap> Permuted(J, (1,2));
[ [ 1, 1, 0 ], [ 1, 0, 1 ] ]
```

Permuting ILPs...

$$\pi \left(\begin{array}{ll} \min & (1,1,1)(x_1,x_2,x_3)^\top \\ & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ & \forall i \in \{1,2,3\} \quad x_i \in \{0,1\} \end{array} \right) =$$

... defined as permuting variables

$$\min \quad (1,1,1)\pi(x_1,x_2,x_3)^\mathsf{T}$$

$$= \quad \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \pi \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \geq \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\pi x \in \{0,1\}^3$$

Action on linear forms

$$\begin{aligned} (1,2,3)(c^{\mathsf{T}} x) &= \text{(def)} \\ = c^{\mathsf{T}}(1,2,3)(x_1, x_2, x_3) &= \\ = c_1 x_3 + c_2 x_1 + c_3 x_2 &= \\ = c_{(1,2,3)3} x_3 + c_{(1,2,3)1} x_1 + c_{(1,2,3)2} x_2 &= \text{(reorder)} \\ = c_{(1,2,3)1} x_1 + c_{(1,2,3)2} x_2 + c_{(1,2,3)3} x_3 &= \end{aligned}$$

$$= \left((1,2,3)(c_1, c_2, c_3)\right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Permuting vars $\Leftrightarrow$ permuting coeffs

Theorem

$$\pi \left(\begin{array}{rcl} \min c^{\mathsf{T}}x & & \\ Ax & \geq & b \\ x & \in & \mathbb{Z} \end{array} \right) = \begin{array}{rcl} \min (\pi c)^{\mathsf{T}}x & & \\ (\pi A)x & \geq & b \\ x & \in & \mathbb{Z} \end{array}$$

ILP isomorphism

π is an ILP isomorphism if

$$\left.\begin{array}{rcl} \min c^{\mathsf{T}}x & & \\ Ax & \geq & b \\ x & \in & \mathbb{Z} \end{array}\right\} \text{``}{=}\text{''} \left\{\begin{array}{rcl} \min (\pi c)^{\mathsf{T}}x & & \\ (\pi A)x & \geq & b \\ x & \in & \mathbb{Z} \end{array}\right.$$

the constraint order is irrelevant

$\Rightarrow$ "=" means:

$$\pi c = c \wedge \exists \sigma \; (\pi A \sigma = A \wedge b\sigma = b)$$

Example

$$\left.\begin{array}{lccc} \min & (1,1,1,1,1,1)^{\mathsf{T}} x & & \\ & \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} x & \geq & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \\ & \forall i \leq 6 \quad x_i & \in & \{0,1\} \end{array}\right\}$$

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

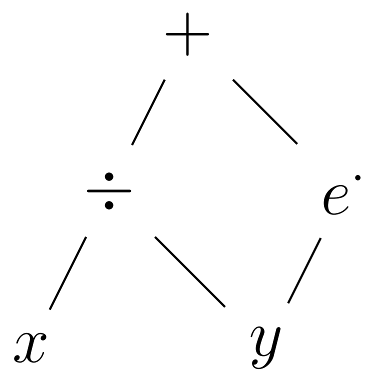
Another example

- Change LP data: $c = (2, 2, 1, 2, 2, 1)$ and $b = (2, 2, 1, 1, 1)^\mathsf{T}$
- Add a last row $(c, 0)$ and column $(b, 0)^\mathsf{T}$ to A
- Re-compute the matrix automorphism group

```
gap> for i in [1..Size(A)] do A[i] :=
   Concatenation(A[i], [1]); od;;
gap> A[1][7] := 2;; A[2][7] := 2;;
gap> Add(A, [2,2,1,2,2,1,0]);
gap> H := MatrixAutomorphisms(A);
Group([ (1,4)(2,5)(3,6), (1,2)(4,5) ])
gap> StructureDescription(H);
"C2 x C2"
gap> IsSubgroup(G,H);
true
```

Generalizations

- Similar definitions hold for LP, MILP, SDP
- For NLP/MINLP, need more work
  - Transform MINLP into a graph using expression DAGs
  - E.g. $\frac{x}{y} + e^y$ is represented by



- Define MINLP isomorphism as a DAG isomorphism

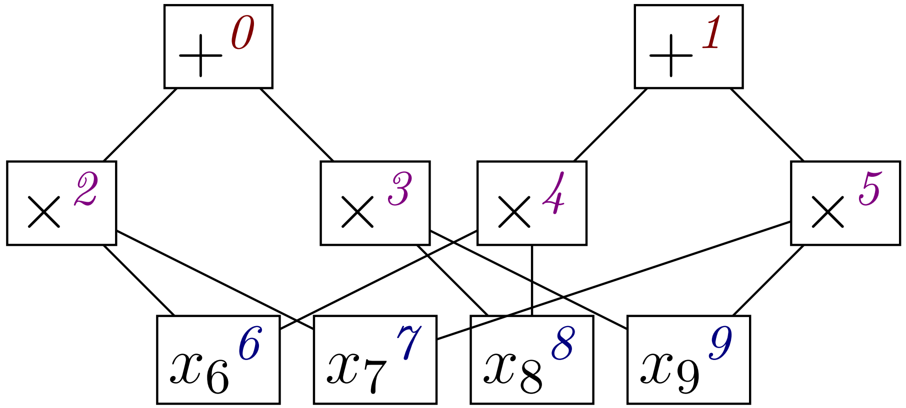
NLP example

- Consider the following NLP

$$
\begin{aligned}
\min x_6 + x_7 + x_8 + x_9 & & \\
\mathsf{c}_0 : x_6 x_7 + x_8 x_9 &= 1 \\
\mathsf{c}_1 : x_6 x_8 + x_7 x_9 &= 1 \\
x_6, x_7, x_8, x_9 &\in [-1, 1]
\end{aligned}
$$

- Objective $x_6 + x_7 + x_8 + x_9$ and box constraints $x_6, x_7, x_8, x_9 \in [-1, 1]$: invariant w.r.t. all permutations of $\{x_6, x_7, x_8, x_9\}$
- Just look at the symmetries of c_0, c_1

NLP example




```
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)      !blue:  variable permutations
(2 3)(6 8)(7 9)      !purple:  operator permutations
(0 1)(2 4)(3 5)(7 8) !red:  constraint permutations
```

$(0,1)(2,4)(3,5)(7,8)$: can swap x_7, x_8 if <u>at the same time</u> swap operator nodes $2,4$ and $3,5$, and constraint nodes $0,1$

NLP example

- Only look at action on variable indices

  x_7, x_8 *can be swapped* $\Rightarrow$ *(*$(1,0,1,1)$ *is a solution* $\Rightarrow$ $(1,1,0,1)$ *is also a solution)*
- But DAG nodes also include operators
- "Project" DAG group action onto variable nodes

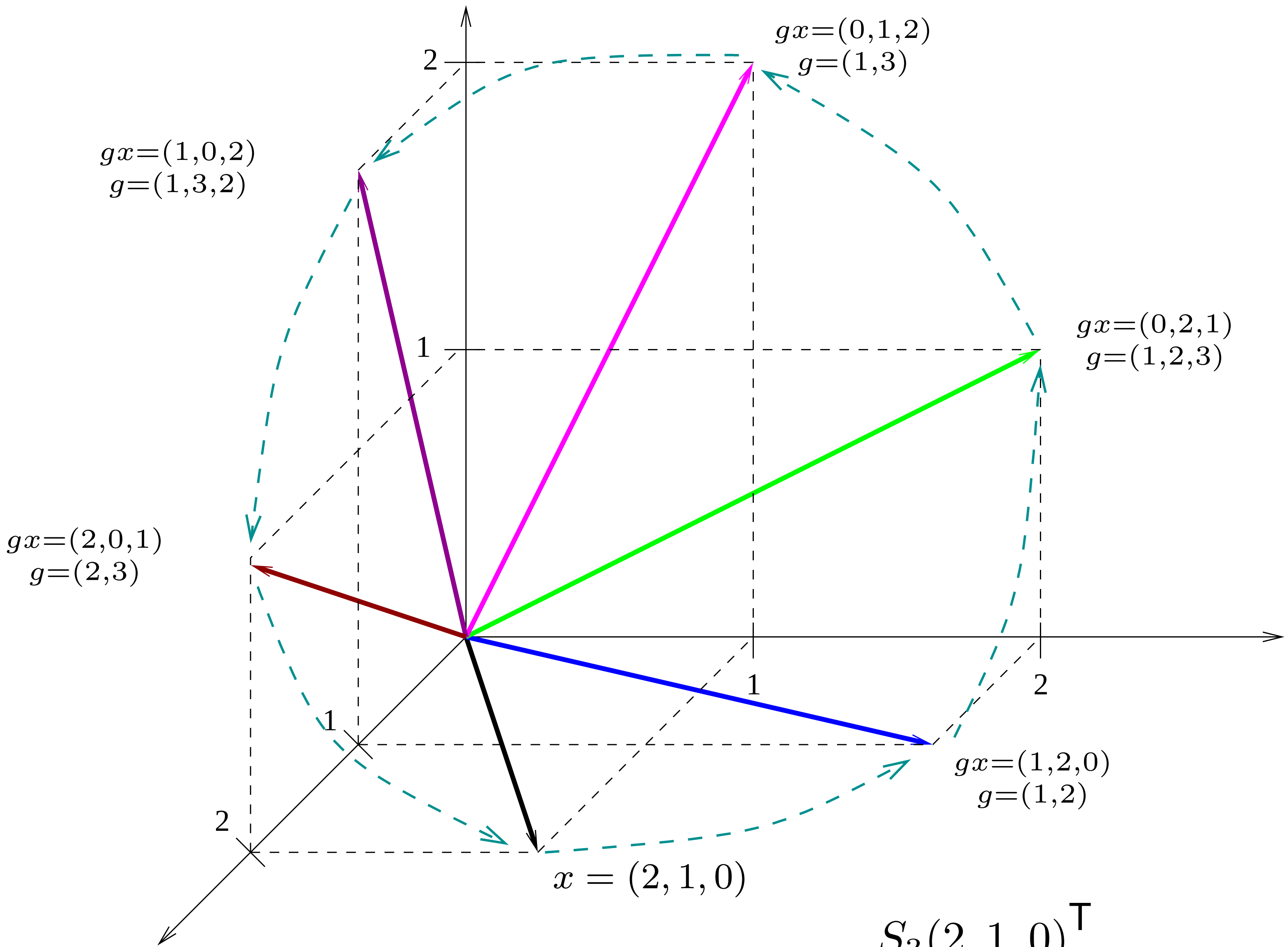
```
gap> L := [(4,5)(6,7)(8,9), (2,3)(6,8)(7,9),
     (10,1)(2,4)(3,5)(7,8)];;
gap> G := Group(L);;
gap> Omg := Orbits(G);
[ [ 1, 10 ], [ 2, 3, 4, 5 ], [6, 7, 8, 9 ] ]

gap> H := Action(G, Omg[3]);
Group([ (1,3)(2,4), (1,2)(3,4), (2,3) ])
gap> LoadPackage("SONATA");;
gap> IsIsomorphicGroup(H, G);
true
```

GAP perms. start from 1

variable indices

What are orbits?



$S_3(2,1,0)^{\mathsf{T}}$

What are orbits?

$$
\begin{aligned}
\pi_1 &= (1,2) \\
\pi_2 &= (4,5,6) \\
G &= \langle \pi_1, \pi_2 \rangle \\
X &= \{1,2,3,4,5,6\} \\
x_1 \sim x_2 &\Leftrightarrow \exists \pi \in G \; (\pi(x_1) = x_2) \\
\Omega(G) &\triangleq X/\sim \quad = \{\{1,2\},\{3\},\{4,5,6\}\}
\end{aligned}
$$

Formulation group

- Reduce a MP P to a coloured DAG $\mathcal{D}_P$
- Find the automorphism group $\mathsf{Aut}(\mathcal{D}_P)$ of the DAG
  Thm.
  The action of $\mathsf{Aut}(\mathcal{D}_P)$ on $\mathcal{D}_P$ can be "meaningfully projected" onto the set of variable indices
- "Meaningfully"$\Rightarrow$
  $\exists \Lambda \subseteq \Omega(\mathcal{D}_P)$ ($\bigcup \Lambda =$ set of variable indices)

  Notation: $\Omega(G) =$ set of orbits of G
- Projection G_P: **formulation group** of P
- Projection operator: look at generators of $\mathcal{D}_P$ as disjoint cycle products, only keep cycles acting on index variables

Solution group

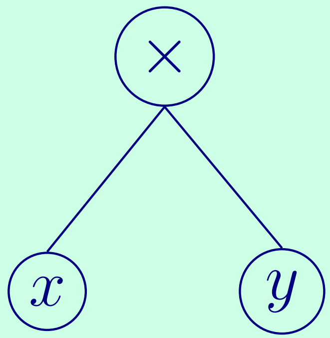
- Let $\mathcal{G}(P)$ be the set of globally optimal solutions of P
- The group $\mathsf{Aut}(\mathcal{G}(P))$ of automorphisms of $\mathcal{G}(P)$ is the **solution group** of P
  Thm.
  Formulation group $\leq$ the solution group
- Computing the solution group generally requires $\mathcal{G}(P)$
- The formulation group can be computed *a priori*

Detecting symmetries in MINLPs

MINLP

- Minimize an objective function $f(x)$ subject to constraints $\forall i \le m \; g_i(x) \le b_i$
- Some variables might be integer: $\forall i \in Z \; (x_i \in \mathbb{Z})$
- f, g_i are mathematical expressions representing functions $\mathbb{R}^n \to \mathbb{R}$
- We represent mathematical expressions using DAGs,

E.g. xy given by

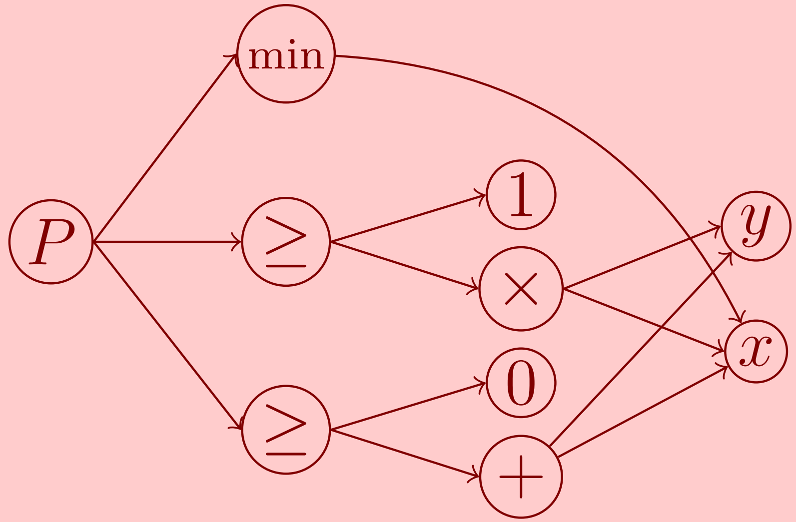


No black-box functions

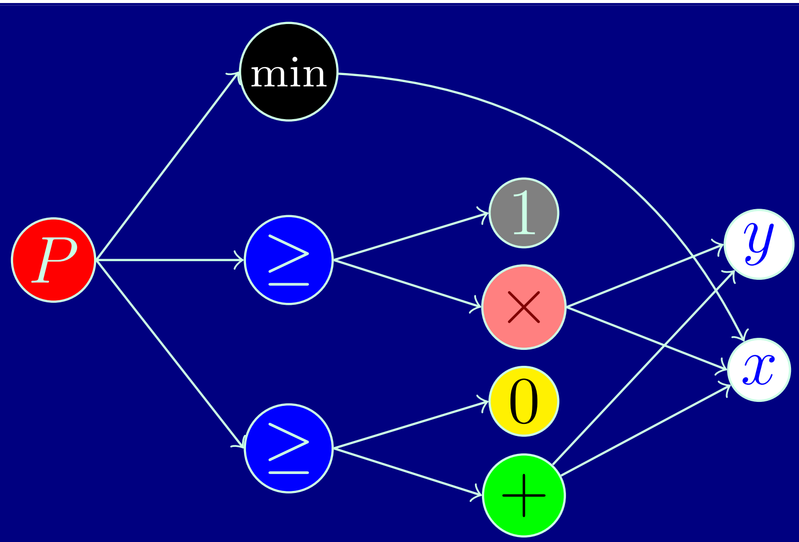
MINLP as DAG

- Introduce special nodes for objective & constraints

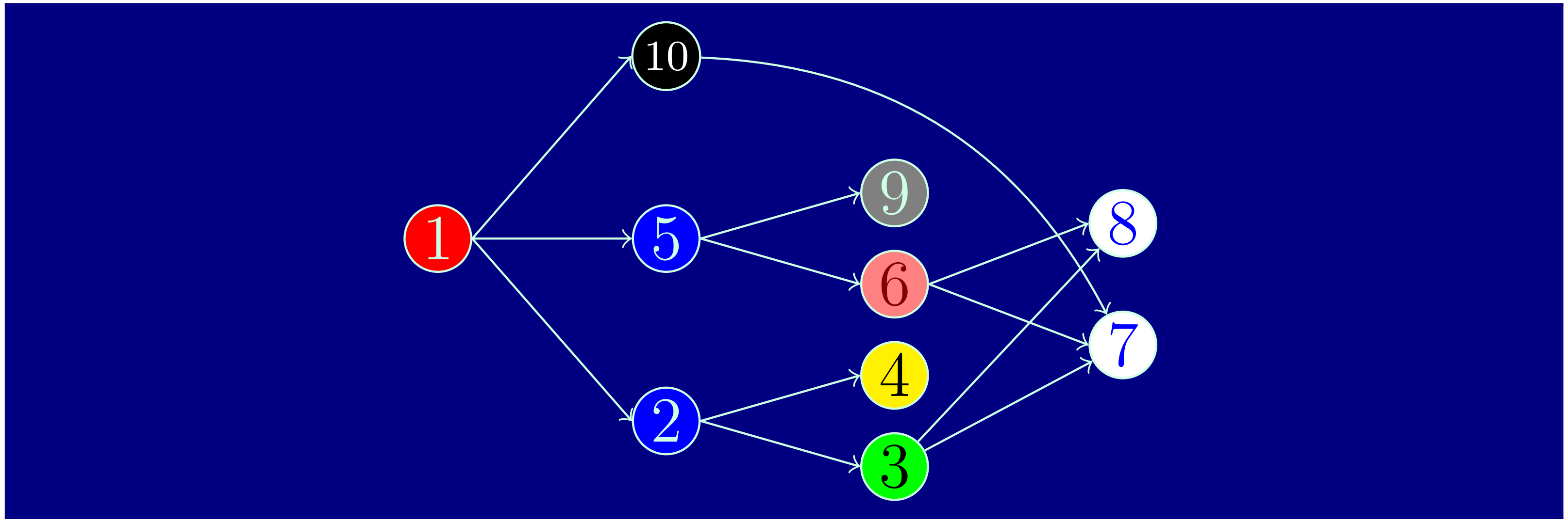
$$\min\{x \mid xy \geq 1 \wedge x+y \geq 0\}$$



- Color nodes the same if it makes sense to swap them



Finding graph automorphisms




```
Dreadnaut version 2.4 (32 bits).
n=10 $=1 g 2 5 10; 3 4; 7 8; ; 6 9; 7 8; ; ; ; 7; x
(2 5)(3 6)(4 9)
f=[1|2 5|3|4|6|9|7 8|10] x
()  !
```
trivial group: no symmetries

From the MINLPLib: `elf`

24 binary variables, 30 continuous variables ≥ 0

$$\min \sum_{i=25}^{48} x_i$$

$$\forall i \in \{25, \dots, 48\}, j \in \{49, 50, 51\}, \qquad k \in \{1, \dots, 8\}$$

$$x_i - (c_k - x_j)^2 \geq 100(x_{i-24} - 1)$$

$$\forall k \in \{1, \dots 8\} \quad \sum_{i=3(k-1)+1}^{3k} x_i \geq 1$$

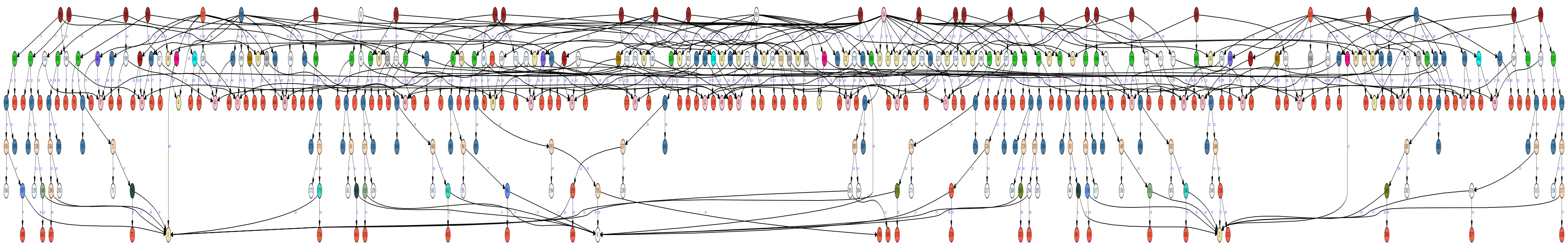
$$\forall h \in \{1, 2, 3\} \; x_{52+h} = \sum_{k=1}^{8} x_{3(k-1)+h}$$

$$\forall h \in \{1, 2, 3\} \; x_{48+h} x_{52+h} = \sum_{k=1}^{8} c_k x_{3(k-1)+h}$$

$$(c_1, \dots, c_8) = (8, 8.5, 8.3, 8.7, 8.6, 9, 9.2, 9.5)$$

The `elf`'s DAG

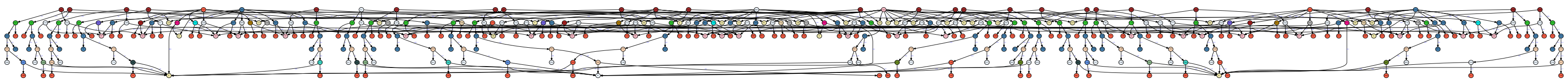
```
$ ( cat elf.mod ; echo "option solver roseamp;" ; \
   echo "option roseamp_options \"mainsolver=14
     symmgroupouttype=3\";" ; \
   echo "solve;" ) | ampl > /dev/null 2>&1
$ mv Rsymmgroup_out.gph elf.gph
$ gph2dot elf.gph > elf.dot
$ dot -Tgif elf.dot > elf.gif
```



The `elf`'s group

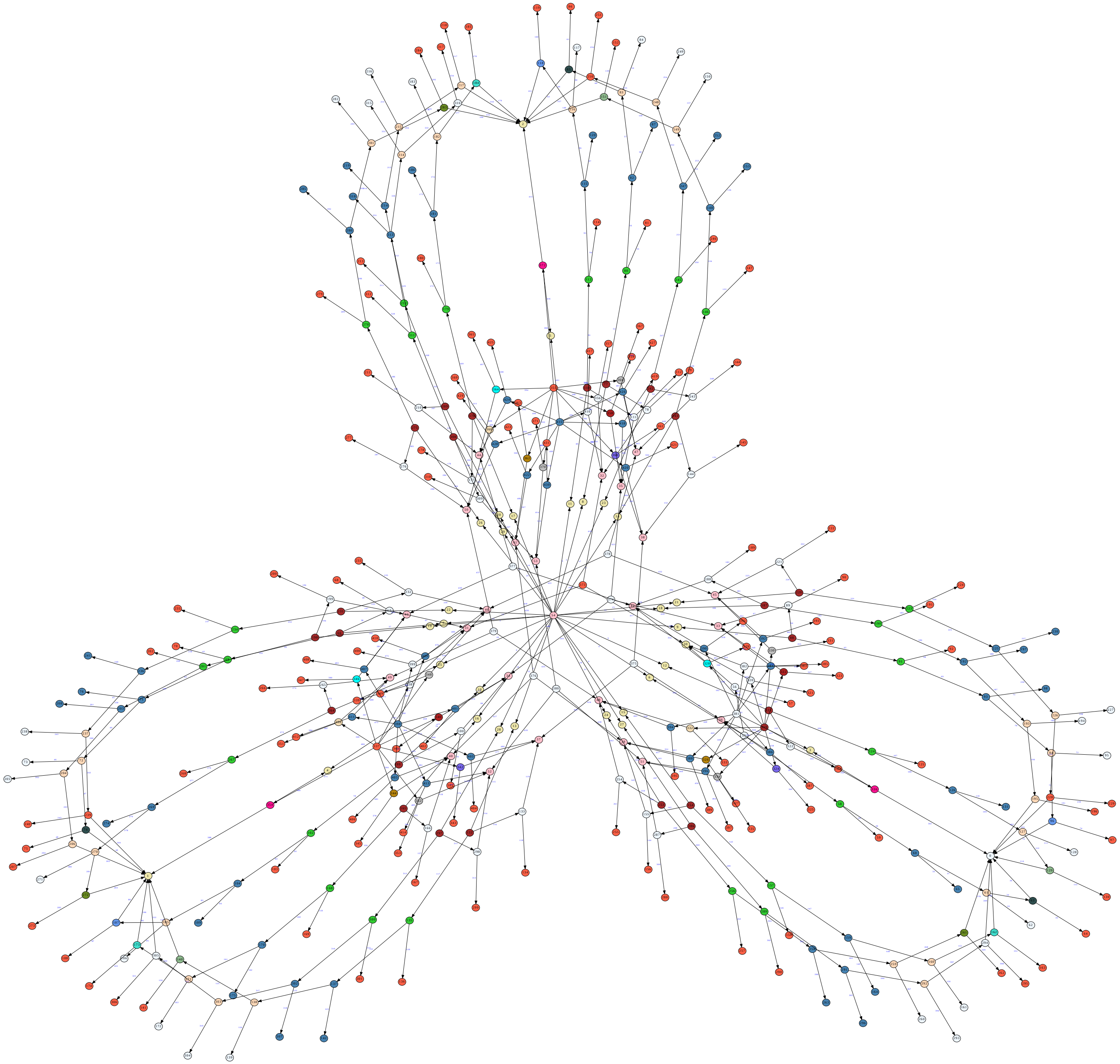
```
$ mod2group elf.mod
$ cat elf.gap | gap -b
<permutation group with 2 generators> G := Group( [ ( 2,
3)( 5, 6)( 8, 9)( 11, 12)( 14, 15)( 17, 18)( 20, 21) (
23, 24)( 26, 27)( 29, 30)( 32, 33)( 35, 36)( 38, 39)( 41,
42)( 44, 45) ( 47, 48)( 50, 51)( 53, 54), ( 1, 2)( 4, 5)(
7, 8)( 10, 11)( 13, 14) ( 16, 17)( 19, 20)( 22, 23)( 25,
26)( 28, 29)( 31, 32)( 34, 35)( 37, 38) ( 40, 41)( 43,
44)( 46, 47)( 49, 50)( 52, 53) ] )
gap> StructureDescription(G);
"S3"
```

The S_3 structure is not immediately evident from



Let's redraw the graph using `fdp`...

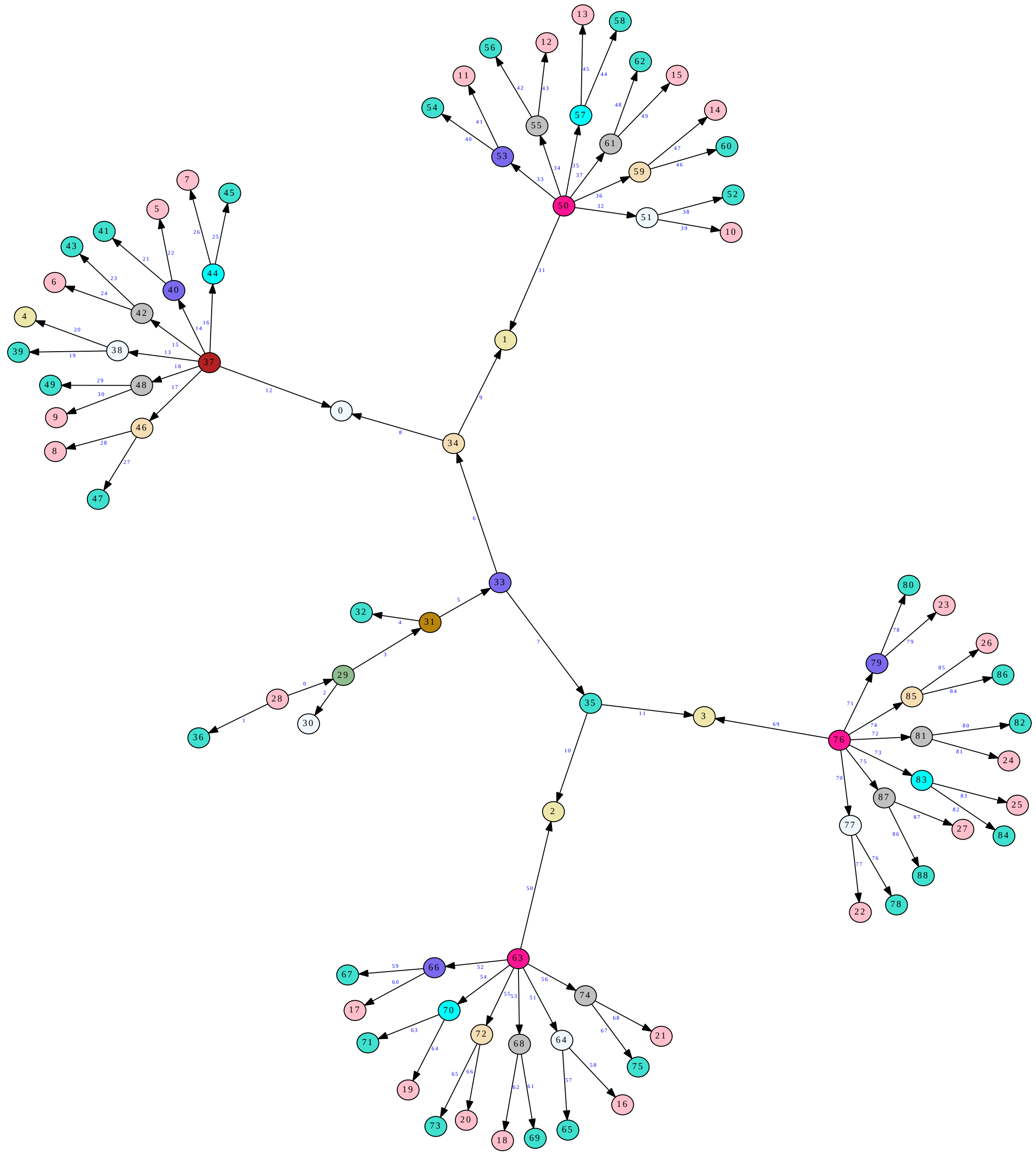
The `elf`'s DAG again



Drawing mathematical programs

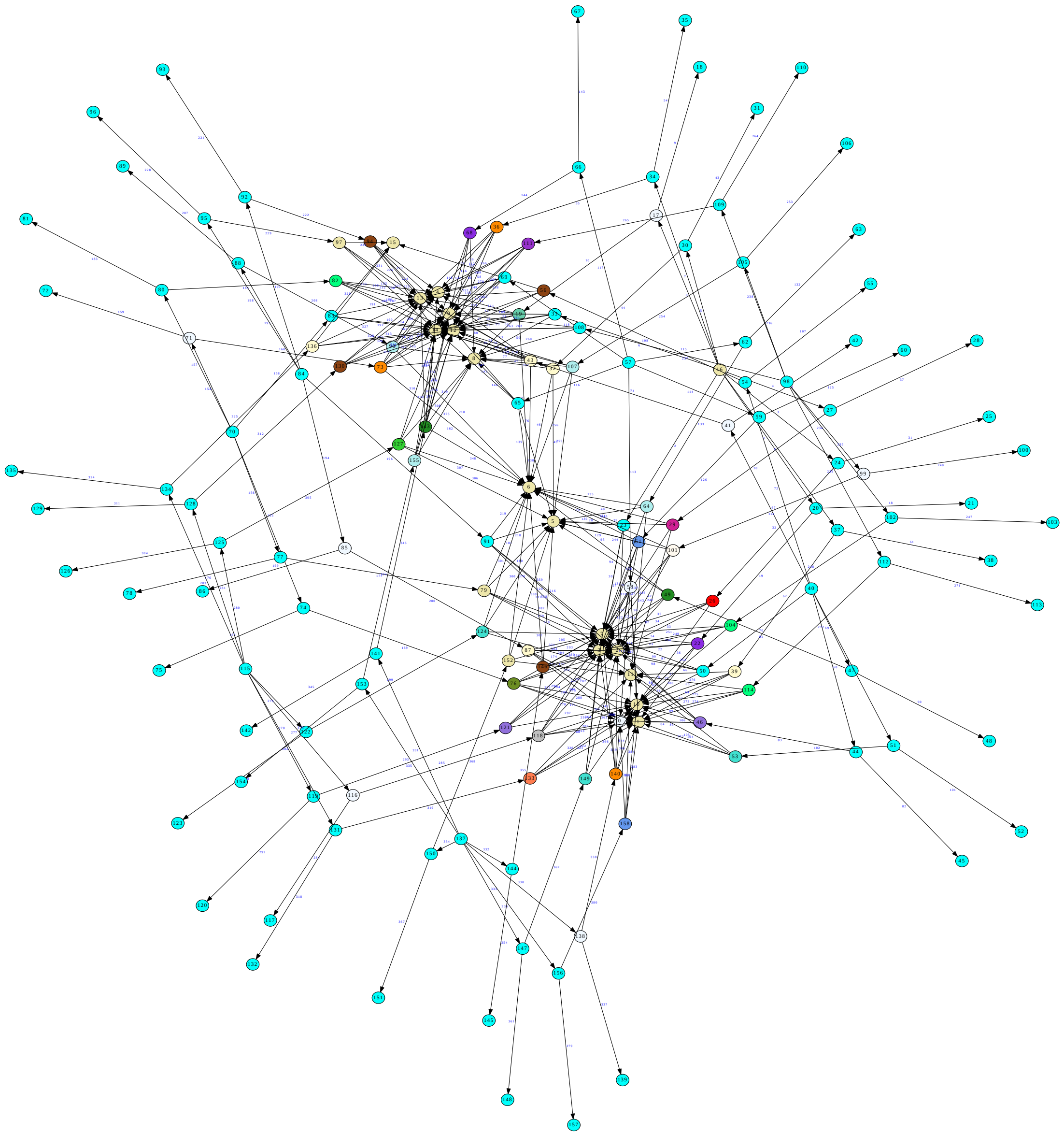
gear2 DAG and group

D_8



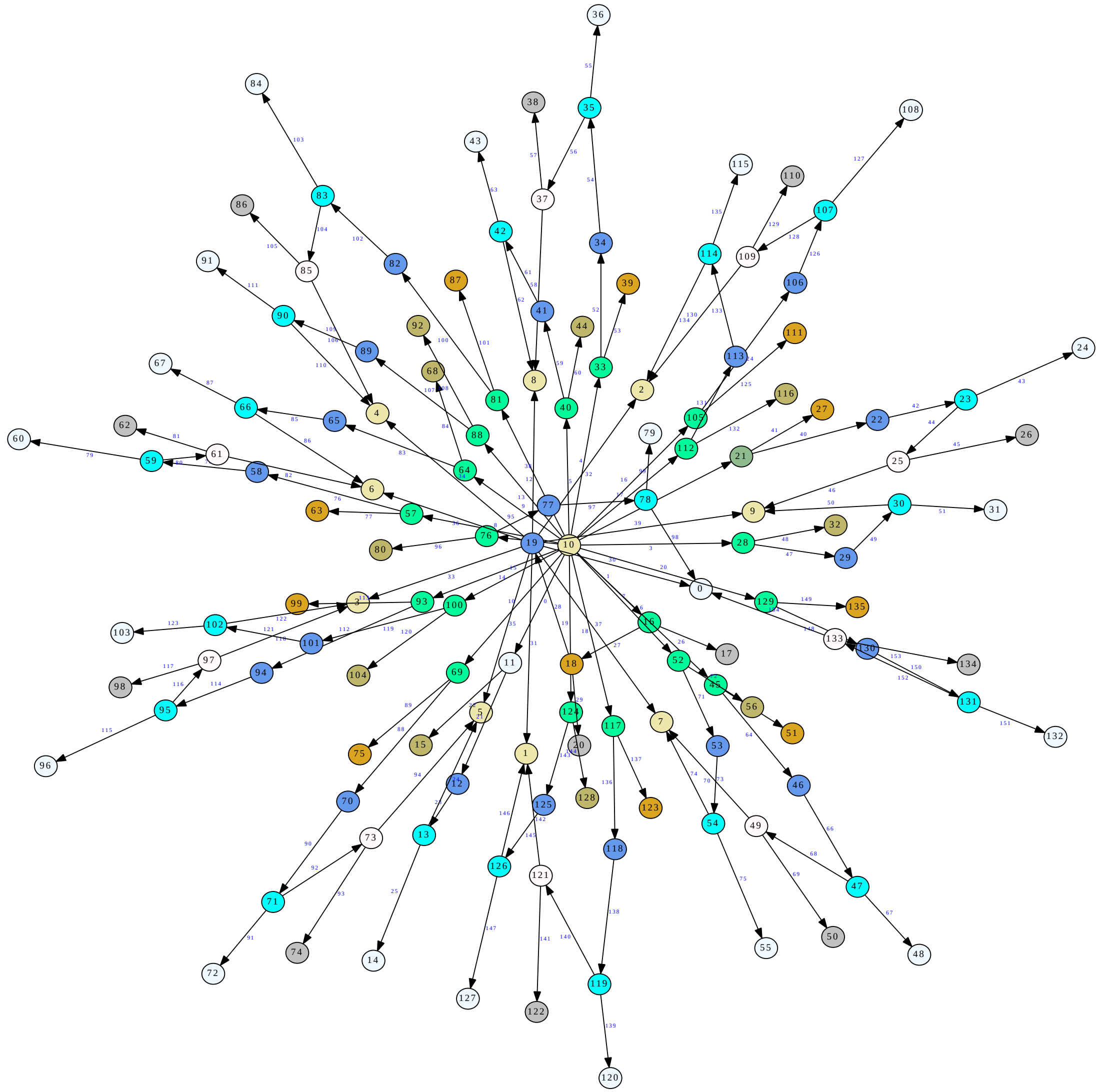
`hmittelman` DAG and group

C_2^3



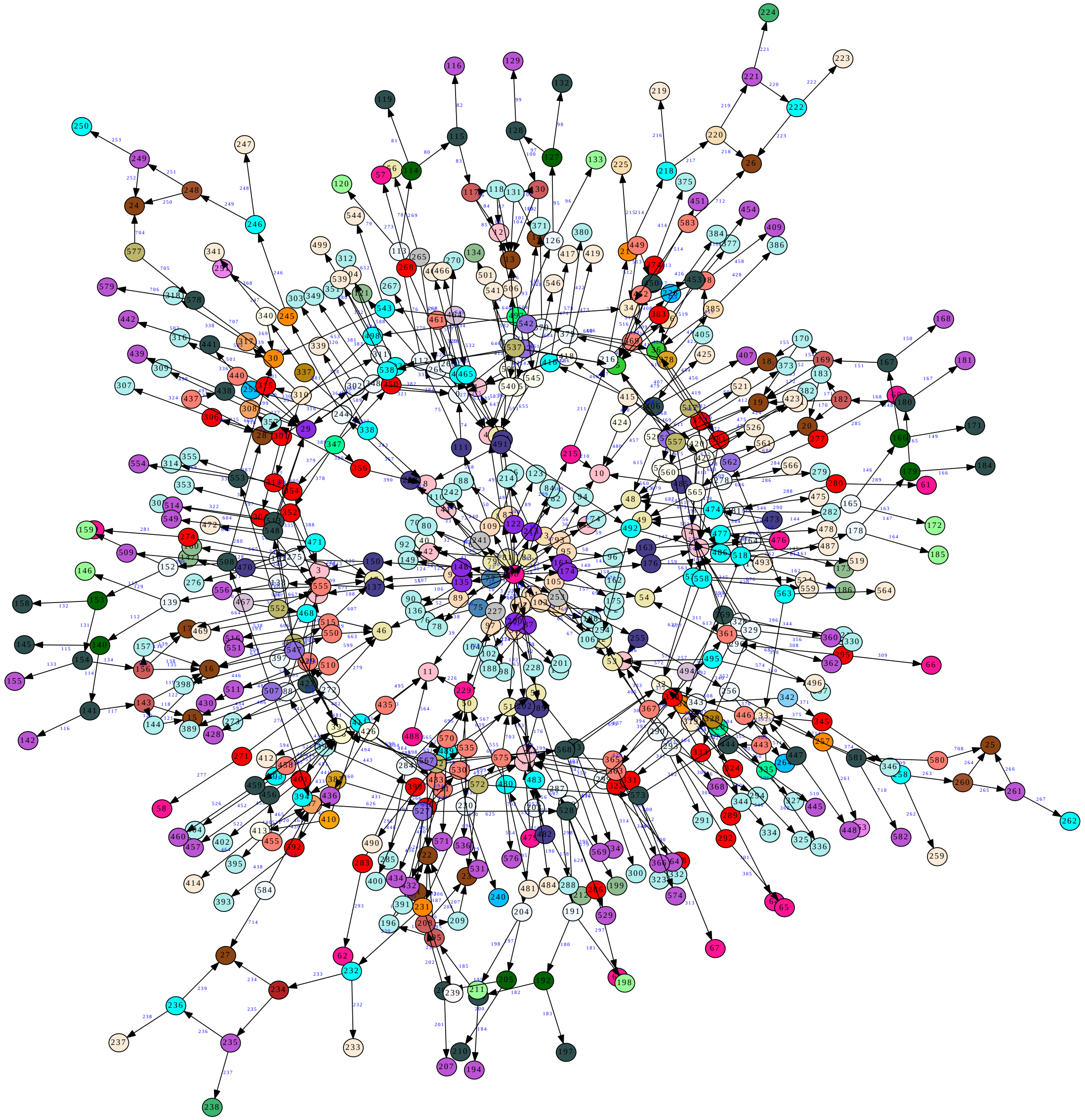
nvs09 **DAG and group**

S_{10}



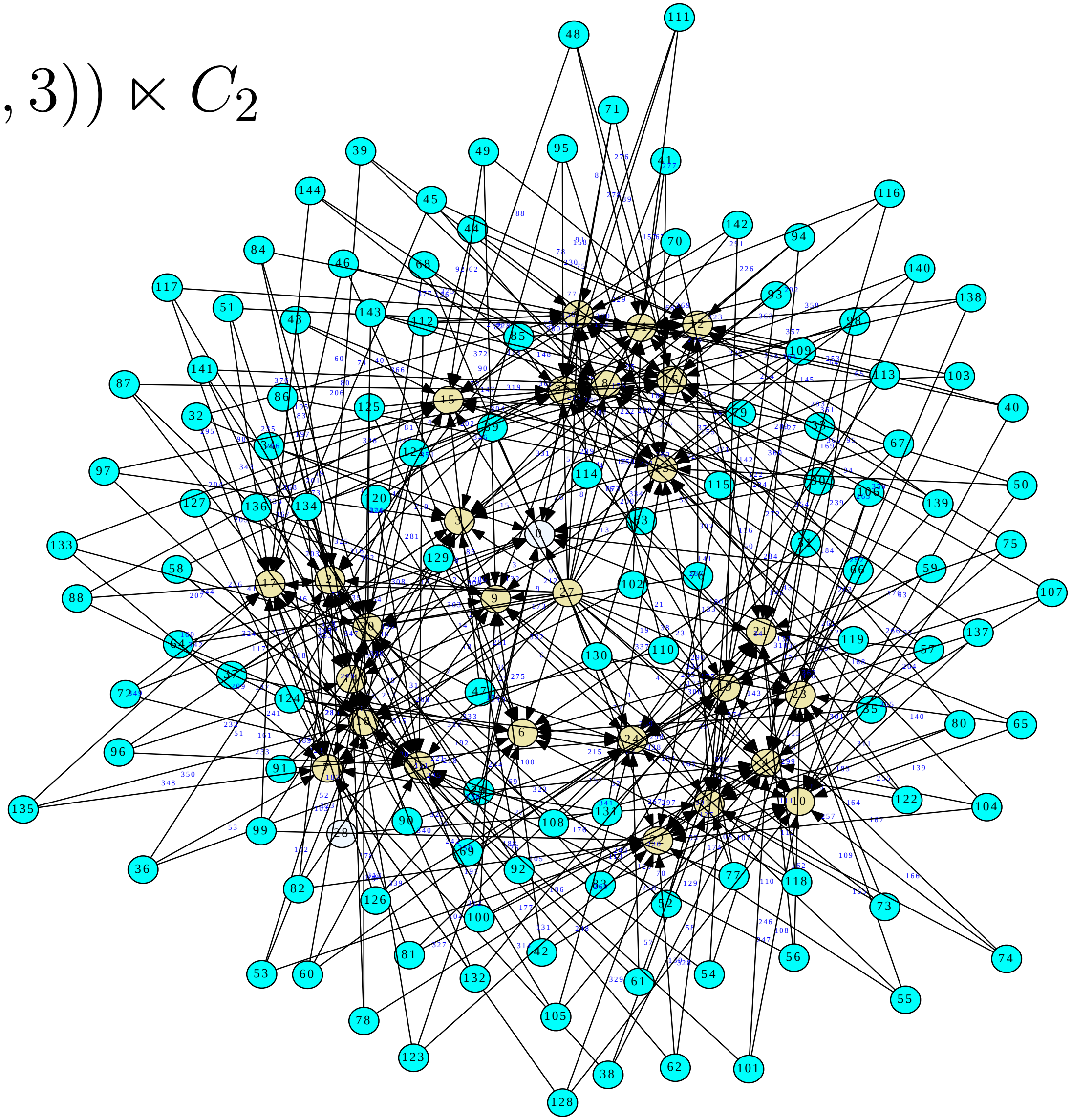
`synheat` DAG and group

S_4



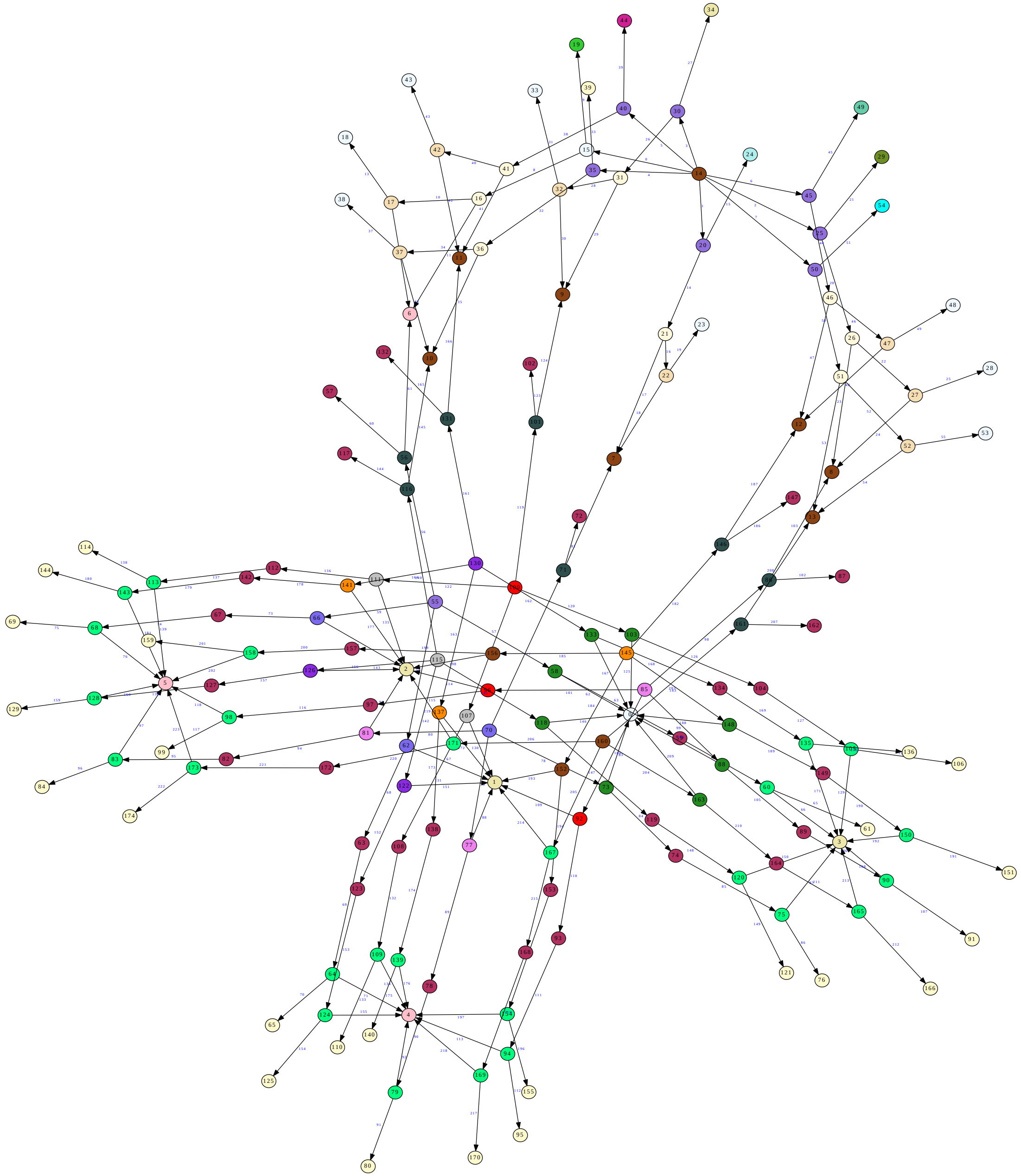
`sts27` DAG and group

$(C_3^3 \ltimes \mathsf{PSL}(3,3)) \ltimes C_2$



`ex8_4_6` DAG and group

S_3



From instances to problems

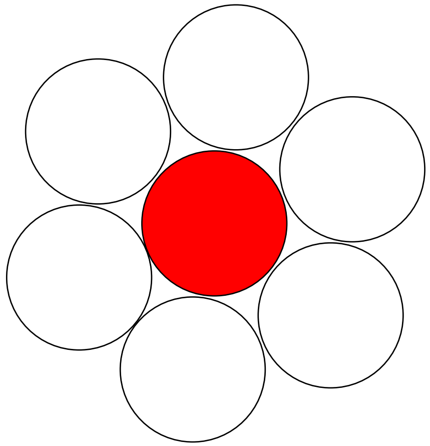
Conjecturing symmetries

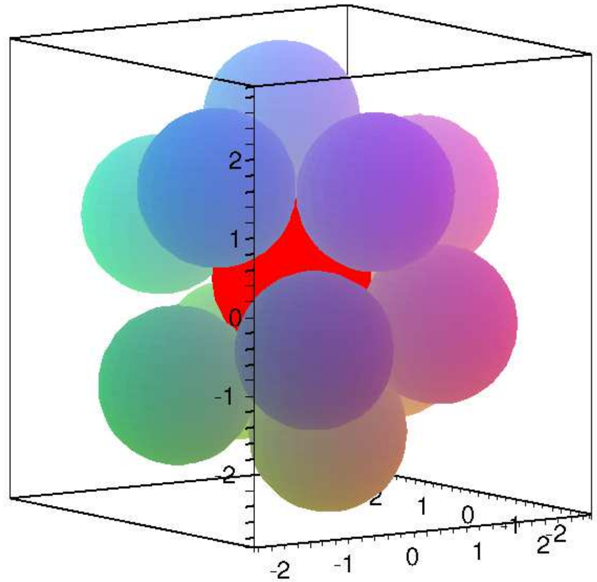
- **Reduction to graph isomorphism**: *per-instance*
- **Problem** $\equiv$ infinite set of instances parametrized on symbols taking infinitely many values
- How about finding symmetries of a problem?
  1. Generate many small instances
  2. Use method above to detect their formulation group
  3. Use GAP's `StructureDescription` to find their group name
  4. Conjecture a group name in function of problem parameter
  5. Prove your conjecture

Kissing Number Problem

- KISSING NUMBER PROBLEM (decision version): Given integers $D, N > 1$, can N unit spheres (with pairwise empty interior intersection) be placed adjacent to a given unit sphere in $\mathbb{R}^d$?
- Formulation:

$$\begin{array}{ll} \max_{x,\alpha} & \alpha \\ \forall i \le N & ||x_i||^2 = 1 \\ \forall i < j \le N & ||x_i - x_j||^2 \ge \alpha \\ \alpha \in [0,1], \forall i \le N \; x_i \in [-1,1]^D & \end{array}$$





KNP group properties

- If $\alpha \geq 1$, answer YES, otherwise NO
- *The solution group has infinite (uncountable) cardinality: each feasible solution can be rotated by any angle in* $\mathbb{R}^D$
- The formulation group, however, is necessarily finite

KNP group: wrong conjecture

```
$ ./knpsymm 2 4 knp.mod
./knpsymm:  group structure is C2
$ ./knpsymm 2 5 knp.mod
./knpsymm:  group structure is C2
$ ./knpsymm 2 6 knp.mod
./knpsymm:  group structure is C2
$ ./knpsymm 3 4 knp.mod
./knpsymm:  group structure is S3
$ ./knpsymm 3 5 knp.mod
./knpsymm:  group structure is S3
$ ./knpsymm 3 6 knp.mod
./knpsymm:  group structure is S3
```

Conjecture: $G_P \cong S_D$

KNP group: right conjecture

Rewrite constraint $\|x_i - x_j\|^2 \geq \alpha$ using $\|x_i\|^2 = 1$:

$$\begin{aligned}
\|x_i - x_j\|^2 &= \sum_{k \leq D} (x_{ik} - x_{jk})^2 \\
&= \sum_{k \leq D} (x_{ik}^2 + x_{jk}^2 - 2x_{ik}x_{jk}) \\
&= \sum_{k \leq D} x_{ik}^2 + \sum_{k \leq D} x_{ik}^2 - \sum_{k \leq D} x_{ik}x_{jk} \\
&= \|x_i\|^2 + \|x_j\|^2 - \sum_{k \leq D} x_{ik}x_{jk} \\
&= 2 - \sum_{k \leq D} x_{ik}x_{jk}
\end{aligned}$$

KNP group: right conjecture

```
$ ./knpsymm 2 4 knpref.mod
./knpsymm:  group structure is C2 x S4
$ ./knpsymm 2 5 knpref.mod
./knpsymm:  group structure is C2 x S5
$ ./knpsymm 2 6 knpref.mod
./knpsymm:  group structure is C2 x S6
$ ./knpsymm 3 4 knpref.mod
./knpsymm:  group structure is S4 x S3
$ ./knpsymm 3 5 knpref.mod
./knpsymm:  group structure is S5 x S3
$ ./knpsymm 3 6 knpref.mod
./knpsymm:  group structure is S6 x S3
```

Conjecture becomes: $G_P \cong S_D \times S_N$

(The reason: $-$ is not commutative, $\times$ is)

Exploiting symmetries

Improving CPU time

- If $G_P \neq 1$, algorithms may evaluate symmetric solutions: duplication of effort
- *Overall strategy*: “remove symmetries”
- You are using a heuristic algorithm:

  don’t bother

- You are using BB/sBB:

  can be useful

Removing symmetries

1. Add cuts leaving at least one solution feasible but making symmetric solutions infeasible
   - problem reformulation of **narrowing** type
   - symbolic algorithms acting on formulation
   - use standard solvers to solve the reformulation
   - marginal to good CPU improvement, worsening also possible

     but it's basically for free
2. Adapt standard solvers to natively deal with symmetry
   - best results
   - much more difficult to implement
   - CPU time worsening still occurs

Focus on reformulation methods *(at least it's free, right?)*

Symmetry Breaking Constraints

Recall the NLP example:

$$\min x_6 + x_7 + x_8 + x_9$$

$$\begin{array}{rcl} \mathsf{c}_0 : x_6 x_7 + x_8 x_9 & = & 1 \\ \mathsf{c}_1 : x_6 x_8 + x_7 x_9 & = & 1 \\ x_6, x_7, x_8, x_9 & \in & [-1, 1] \end{array}$$

```
gap> G := Group((6,7)(8,9), (6,8)(7,9), (7,8));; Orbits(G)
[ [6, 7, 8, 9] ]
```

- G is transitive on $\omega = \{6, 7, 8, 9\}$
- $\Rightarrow \forall i \neq j \in \omega \exists \pi \in G \; (\pi(i) = j)$
  Thm.
  Whatever the structure of G, and for all $i \in \omega$, there must be $\pi \in G$ such that $x_{\pi(i)} = \min\{x_j \mid j \in \omega\}$
- **Arbitrarily choose** $i = \min \omega$
- $\Rightarrow$ Cuts $\forall j > i \; (x_i \leq x_j)$ preserve at least one optimum

Generic SBCs

- By definition, all groups are transitive on their orbits
- The action of G_P *limited to a single orbit ω* subsumes a group $G_P[\omega]$ called a **transitive constituent** of G_P
  - $G_P[\omega]$ is isomorphic to a subgroup of G_P
  - $G_P[\omega]$ could be seen as the "projection of G_P on ω"
- Cuts for P guaranteed to keep at least one symmetric optimum feasible are called **symmetry breaking constraints** (SBC)
- SBCs depend on the structure of $G_P[\omega]$
- By transitivity of $G_P[\omega]$ on ω,

$$\forall j \in \omega \quad (j > \min \omega \to x_{\min \omega} \leq x_j)$$

are SBCs for any $G_P[\omega]$

In any orbit, one element must index a component with minimum value

Strong SBCs

Thm.

If x^* is an optimum of P and $G_P[\omega] = \mathsf{Sym}(\omega)$, any ordering of $\{x^*_j \mid j \in \omega\}$ yields an optimum of P

Hence, the following are valid SBCs:

$$\forall j \in \omega \quad (j < \max \omega \rightarrow x_j \leq x_{j^+}),$$

where j^+ is the $<$-successor of j in ω

In orbits where G_P has a full symmetric action, we can impose an order

The `sb3` reformulation

Automatic SBC generation

Strategy (`sb3`)

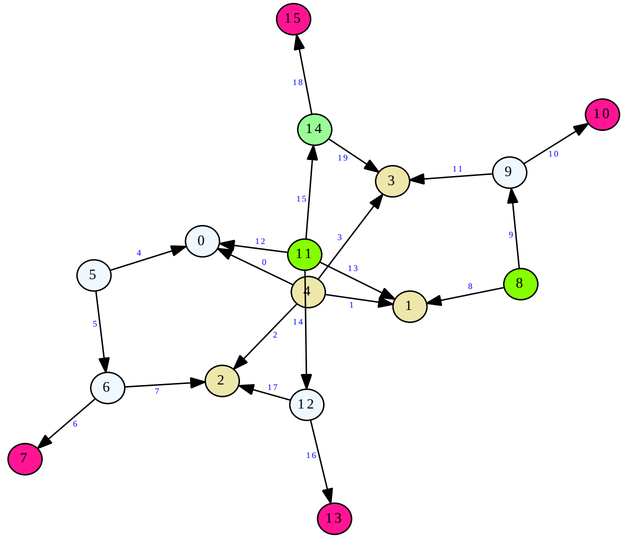
1. Automatically detect G_P
2. Let ω be the largest orbit of G_P
3. If $G_P[\omega] = \mathsf{Sym}(\omega)$, generate strong SBCs
4. Otherwise, generate generic SBCs
5. If *[very restrictive conditions]* let ω be another orbit of G_P and goto Step 3

Several orbits?

In general, we cannot carry out the previous procedure "for all orbits"

$$\min_{x\in\{0,1\}^4} x_1 + x_2 + x_3 + x_4$$

$$\begin{array}{rcl} x_1 + 2x_3 & \geq & 1 \\ x_2 + 2x_4 & \geq & 1 \\ x_1 + x_2 & = & x_3 + x_4 \end{array}$$



Group $G_P = \langle(1,2)(3,4)\rangle \cong C_2$, orbits $\Omega = \{\omega_1 = \{1,2\}, \omega_2 = \{3,4\}\}$

- $x^* = (0,1,1,0)$ is an optimum
- SBCs: $\omega_1 \Rightarrow \boxed{x_1 \leq x_2}$, $\omega_2 \Rightarrow \boxed{x_3 \leq x_4}$
- Both SBCs at the same time $\Rightarrow$ both x^*, $(1,2)(3,4)x^*$ infeasible $\Rightarrow$ problem becomes infeasible

Restrictive conditions

- Let $X \subseteq \{1, \ldots, n\}$ and $G \leq S_n$
- $G[X]$: product of generators g of G s.t. g moves elements of X only

---

- Consider distinct orbits ω, θ s.t. $\gcd(|\omega|, |\theta|) = 1$
- If $G_P[\omega \cup \theta]$ has a subgroup H s.t:

$$H[\omega] \cong C_{|\omega|} \ \wedge \ H[\theta] \cong C_{|\theta|}$$

then:

We can adjoin SBCs for ω, θ at the same time

Performance: small-scale

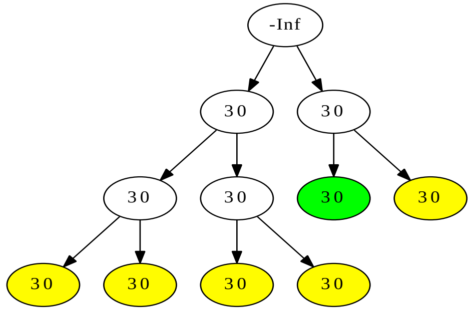
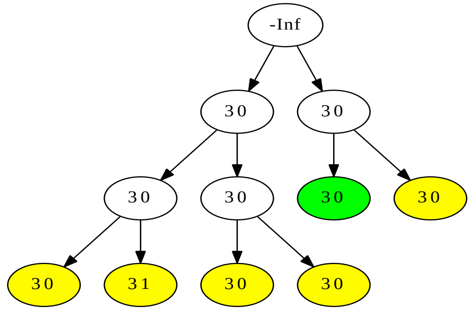
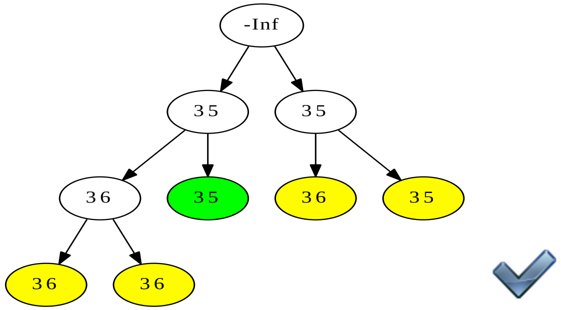
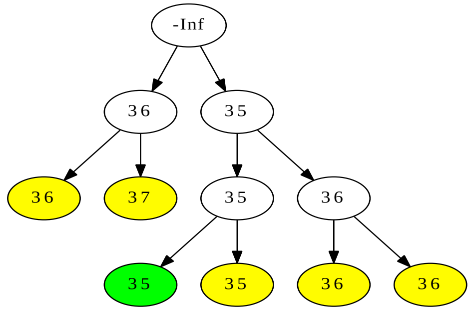
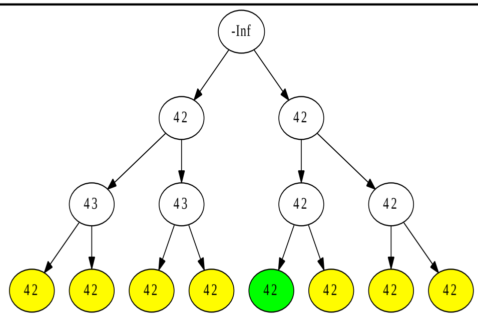
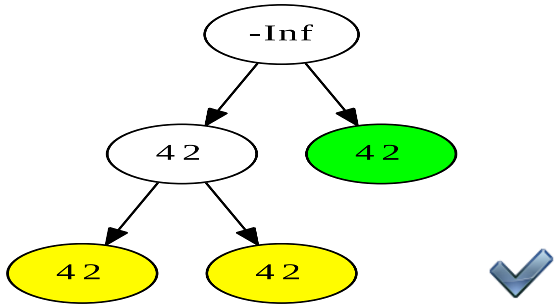
- Generate and test small-scale vertex cover instances
- Formulation: given $G = (V, E)$,

$$\min_{x \in \{0,1\}^{|V|}} \sum_{v \in V} x_v$$

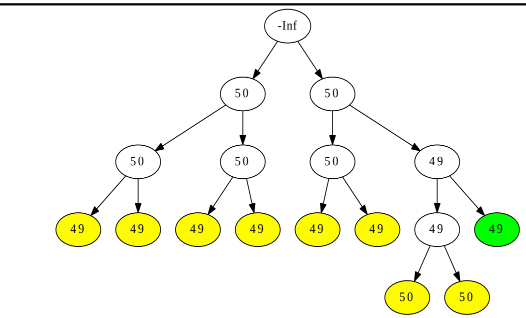
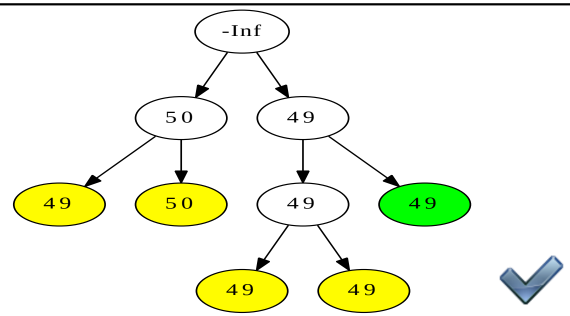
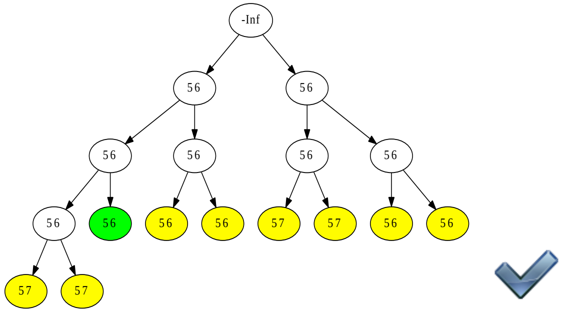
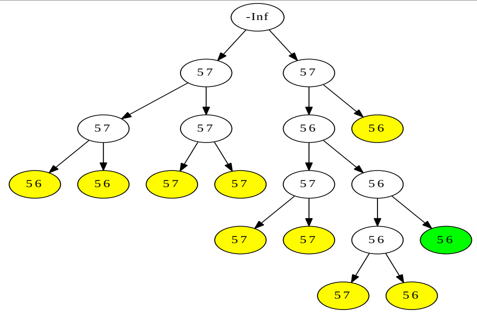
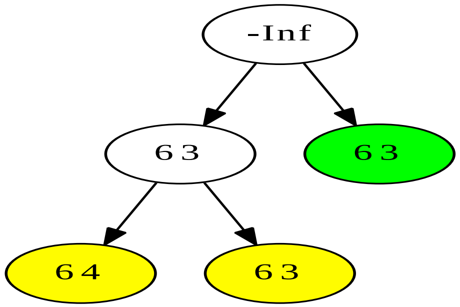
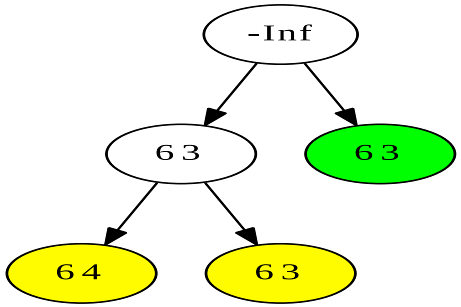
$$\forall v \in V \quad \sum_{\substack{u \in V \\ \{u,v\} \in E}} x_u \geq 1$$

- Formulation group $\cong$ graph automorphism group

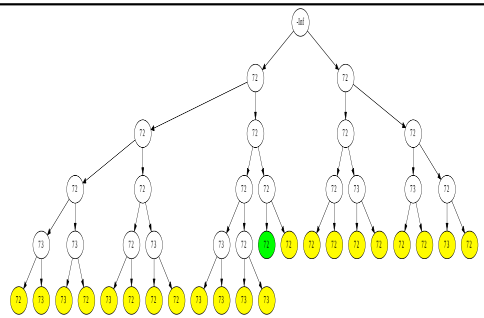
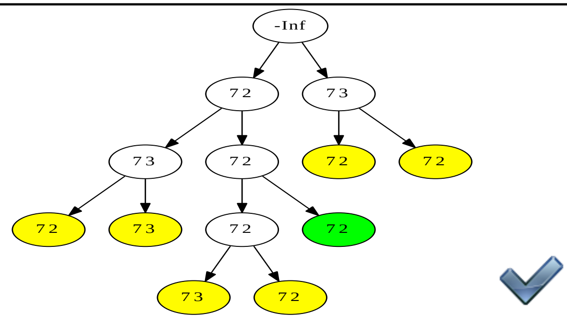
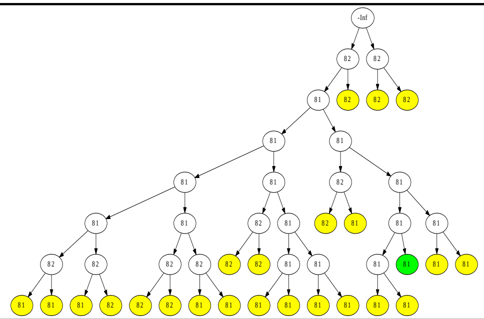
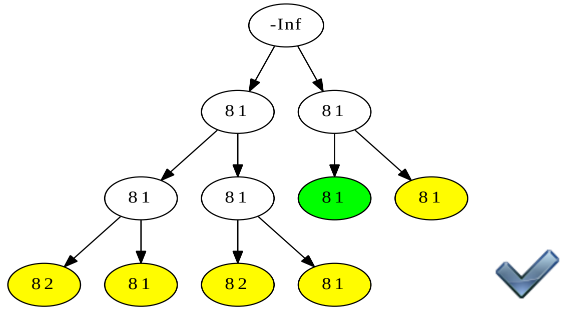
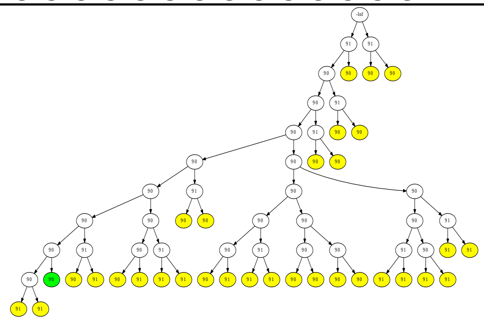
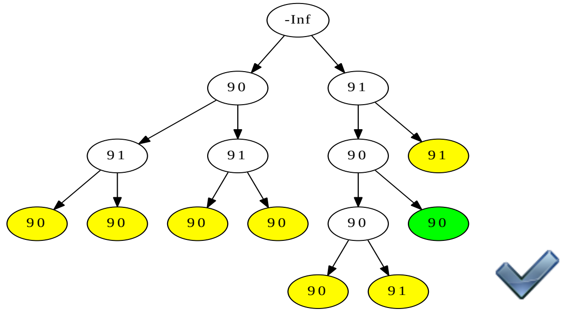
Mesh graphs 1

Size/group	BB tree: original	BB tree: `sb3`
10×10 $C_2^4 \ltimes C_2$		
11×11 D_8^2		
12×12 $C_2^4 \ltimes C_2$		

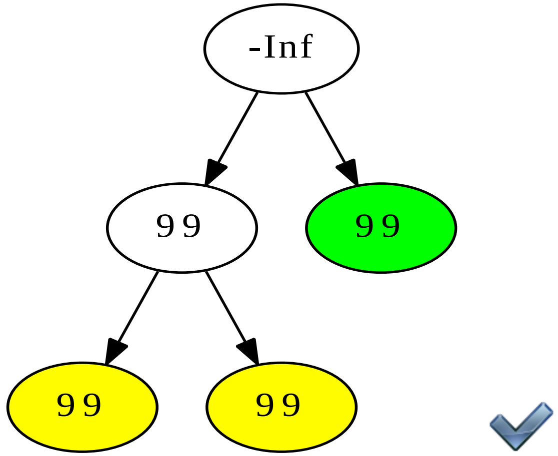
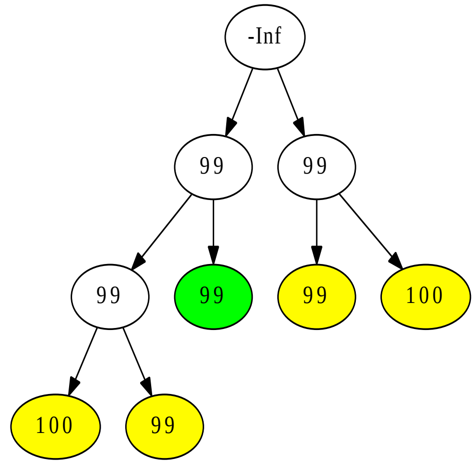
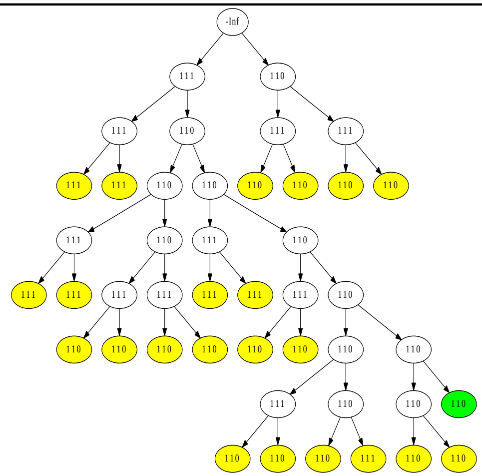
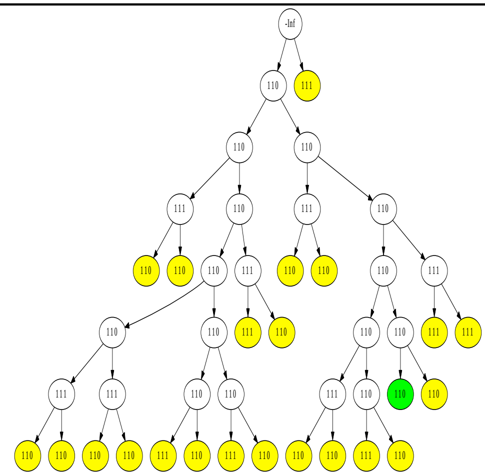
Mesh graphs 2

Mesh size	BB tree: original	BB tree: `sb3`
13×13 D_8^2		
14×14 $C_2^4 \ltimes C_2$		
15×15 D_8^2		

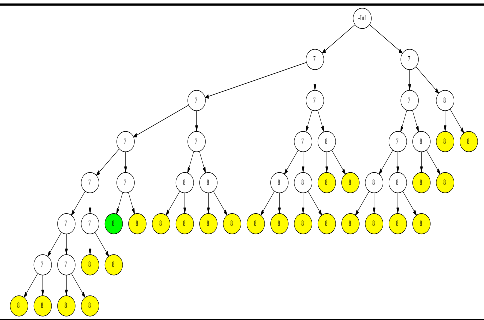
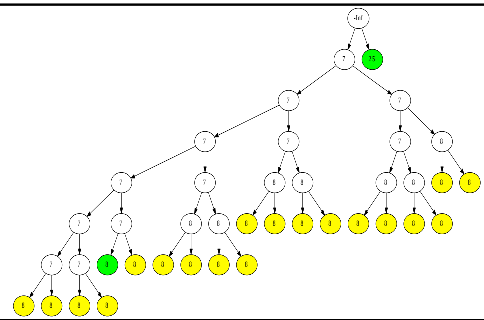
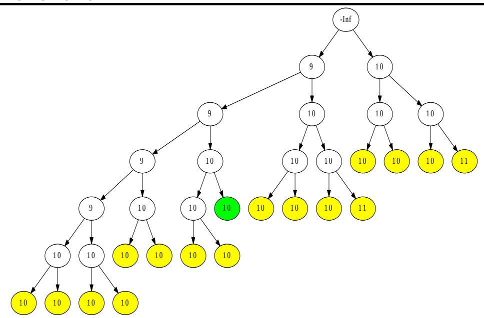
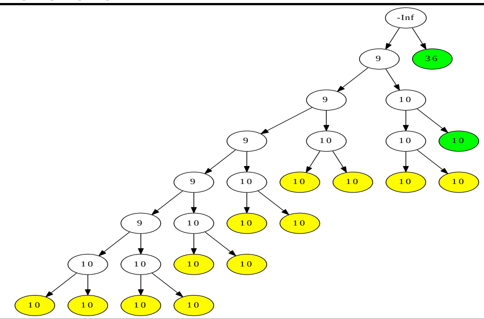
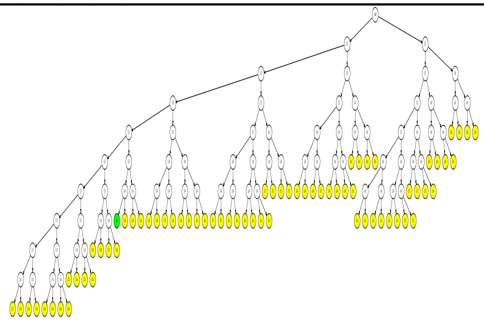
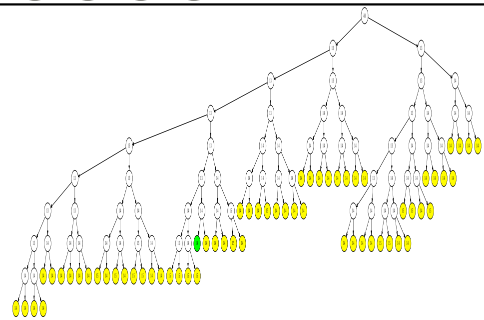
Mesh graphs 3

Mesh size	BB tree: original	BB tree: `sb3`
16×16 $C_2^4 \ltimes C_2$		
17×17 D_8^2		
18×18 $C_2^4 \ltimes C_2$		

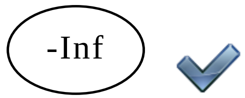
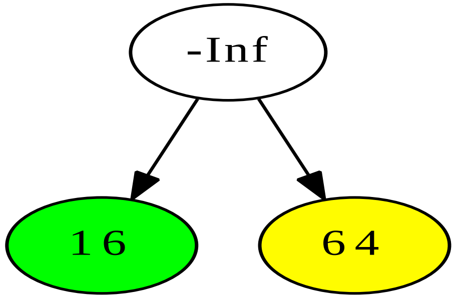
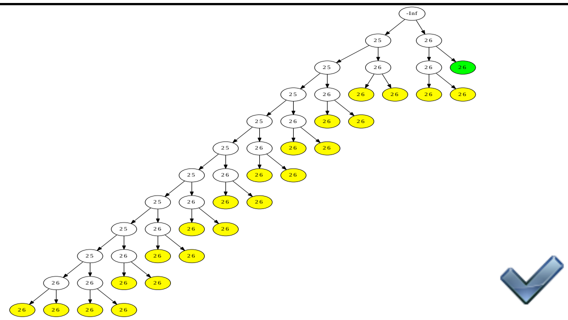
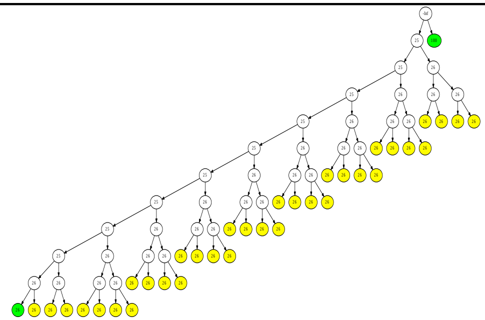
Mesh graphs 4

<table>
<tr><th>Mesh size</th><th>BB tree: original</th><th>BB tree: <code>sb3</code></th></tr>
<tr><td>19×19
D_8^2</td><td>
</td><td>
</td></tr>
<tr><td>20×20
$C_2^4 \ltimes C_2$</td><td></td><td></td></tr>
</table>

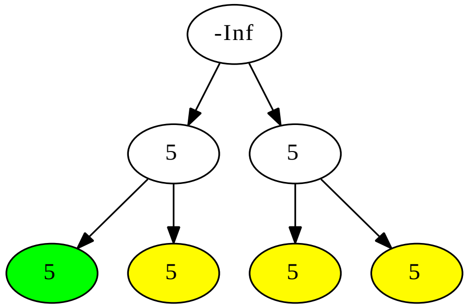
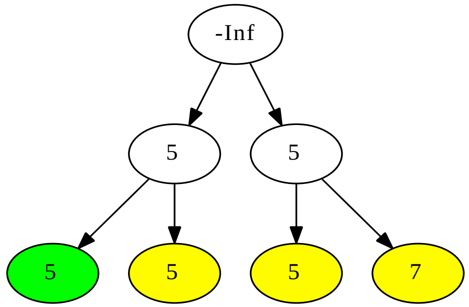
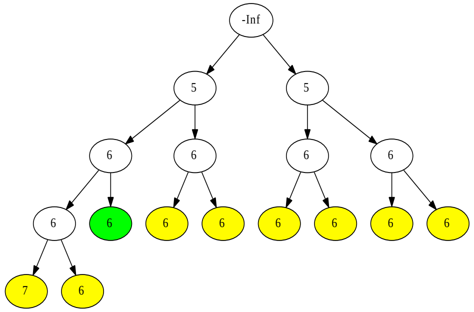
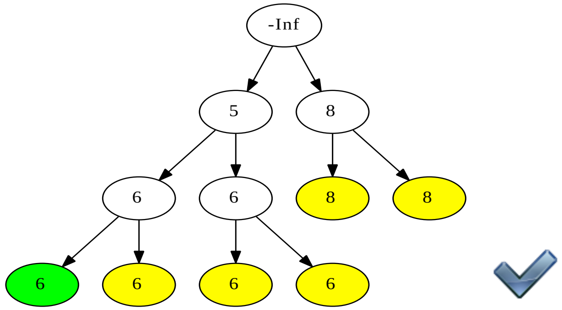
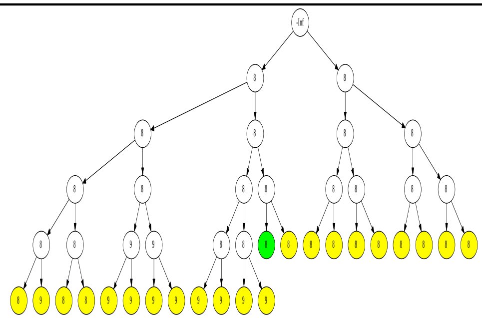
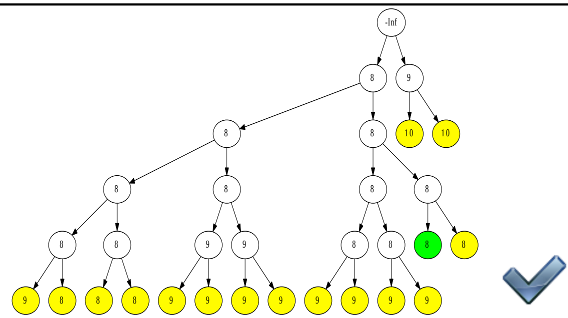
Torus graphs 1

Size/group	BB tree: original	BB tree: `sb3`
5×5 $D_{10}^2 \ltimes C_2$		
6×6 $(C_2 \times (S_3^2 \ltimes C_2)) \wr C_2$		
7×7 $D_{14}^2 \ltimes C_2$		

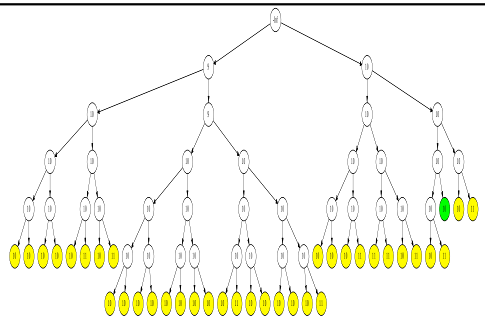
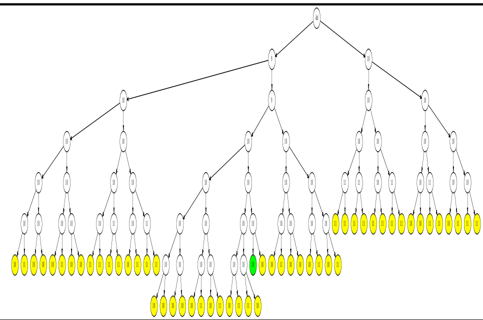
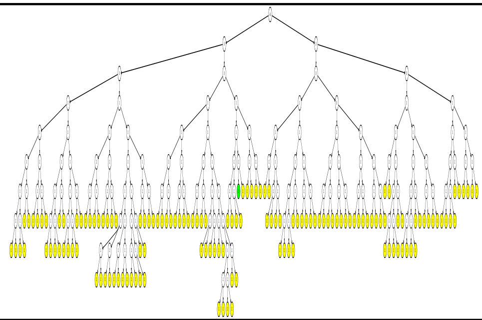
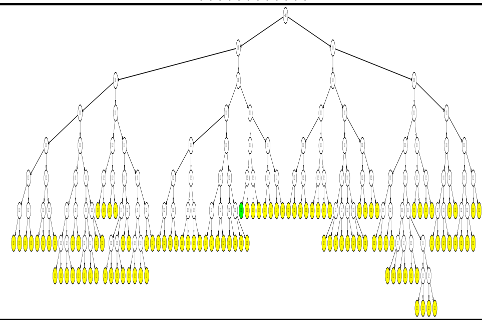
Torus graphs 2

Mesh size	BB tree: original	BB tree: `sb3`
8×8 $\lvert G \rvert = 131072$ 2 orbits		
9×9 $D_{18}^2 \ltimes C_2$	(1105 nodes)	(677 nodes)
10×10 $\lvert G \rvert = 320000$ 2 orbits		

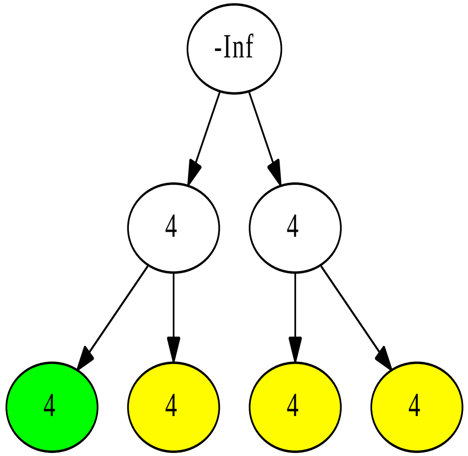
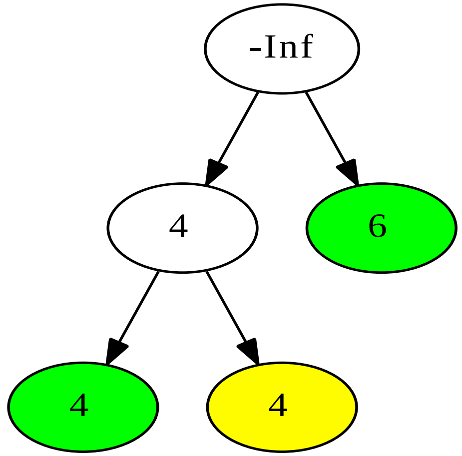
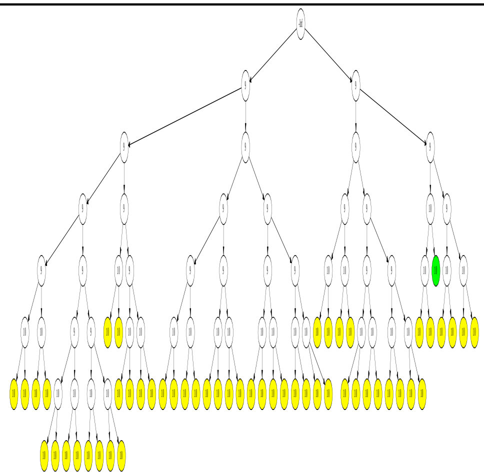
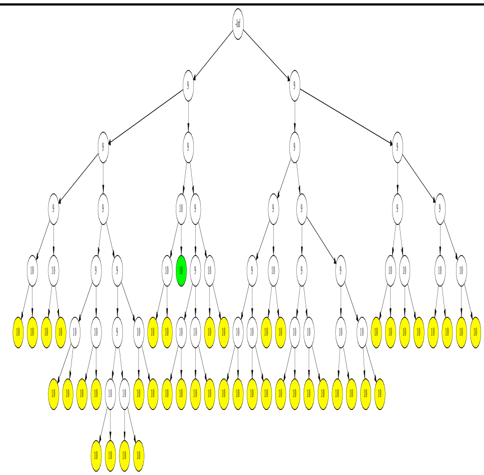
Triangle graphs 1

Size/group	BB tree: original	BB tree: `sb3`
5 S_3		
6 S_3		
7 S_3		

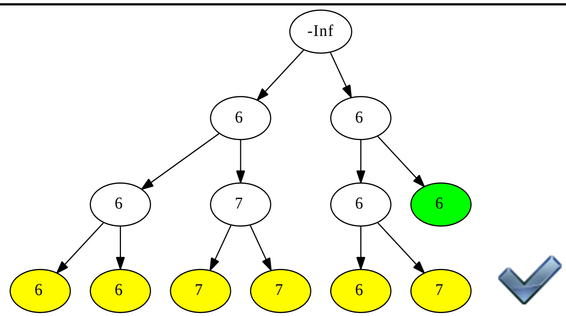
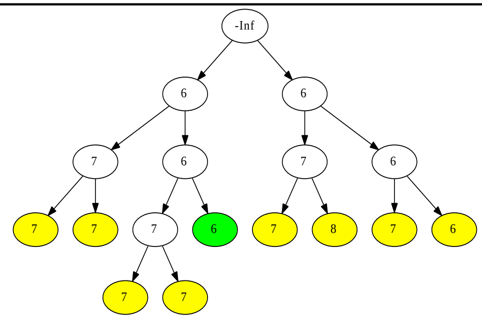
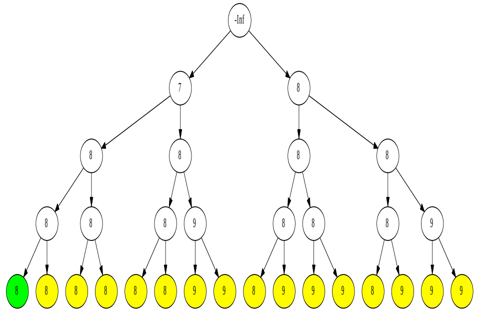
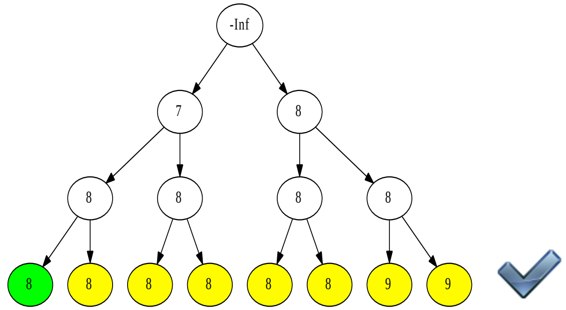
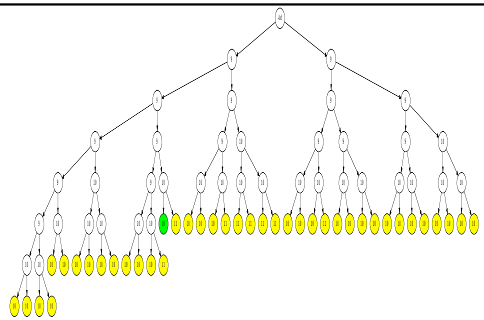
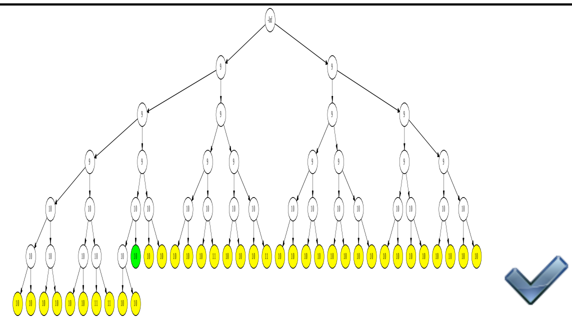
Triangle graphs 2

Mesh size	BB tree: original	BB tree: `sb3`
8 S_3	 ✓	
9 S_3		 ✓
10 S_3	(1593 nodes)	(1271 nodes) ✓

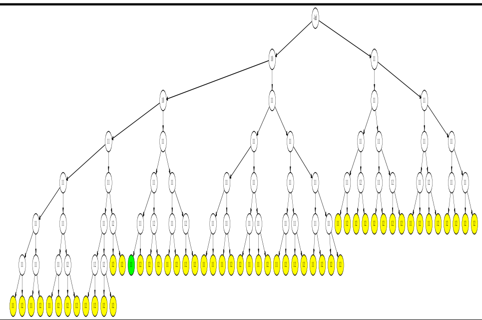
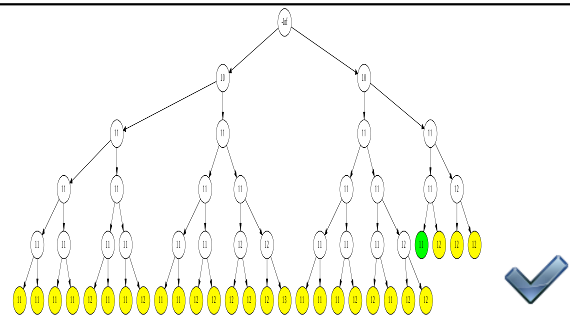
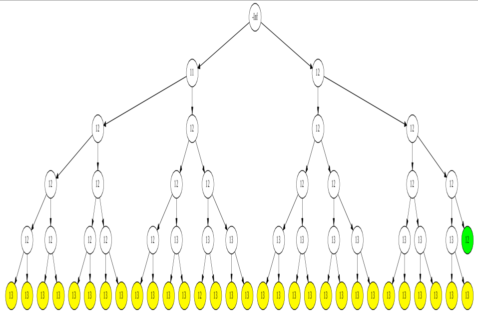
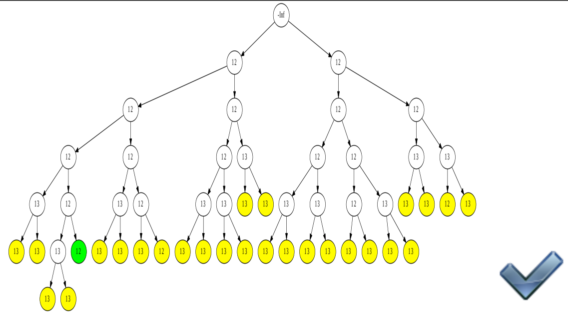
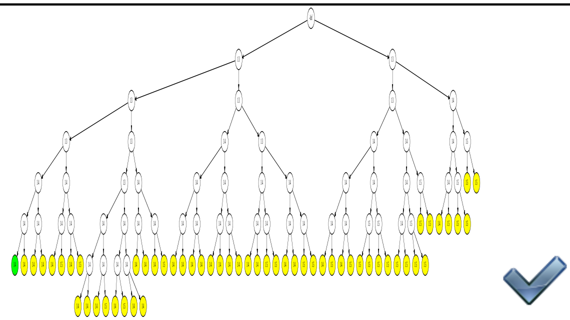
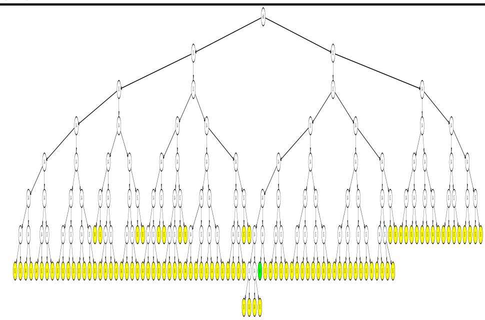
Permutahedron graphs

Order/group	BB tree: original	BB tree: `sb3`
3 $S_3^2 \ltimes C_2$		
4 $(C_2^2 \times A_4^2) \ltimes C_2$		

Flower snarks 1

Order/group	BB tree: original	BB tree: sb3
4 $D_8^2 \ltimes C_2$		
5 $S_3 \times D_{10}$		
6 $(C_2^2 \times S_3^2) \ltimes C_2$		

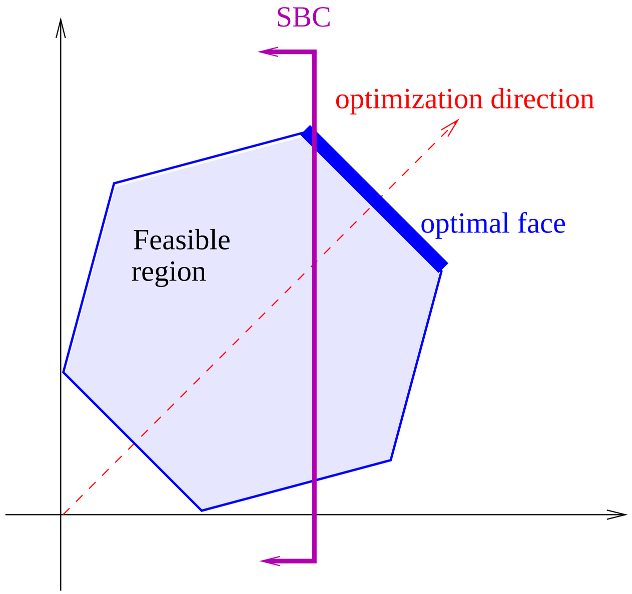
Flower snarks 2

Order	BB tree: original	BB tree: `sb3`
7 $S_3 \times D_{14}$		
8 $(D_{16}^2) \ltimes C_2$		
9 $S_3 \times D_{18}$		

Improvement?

Sometimes the original formulation solves faster than the SBC reformulation

- BB performance depends on several factors:
  LP solution, Branching policy, Cut generation, ...
- Adjoining SBCs keeps objective function value by changes LP solution



- Forcing **arbitrary choices** in SBCs prevents BB from making the correct ones

The `sbstab` reformulation

Leftover symmetry

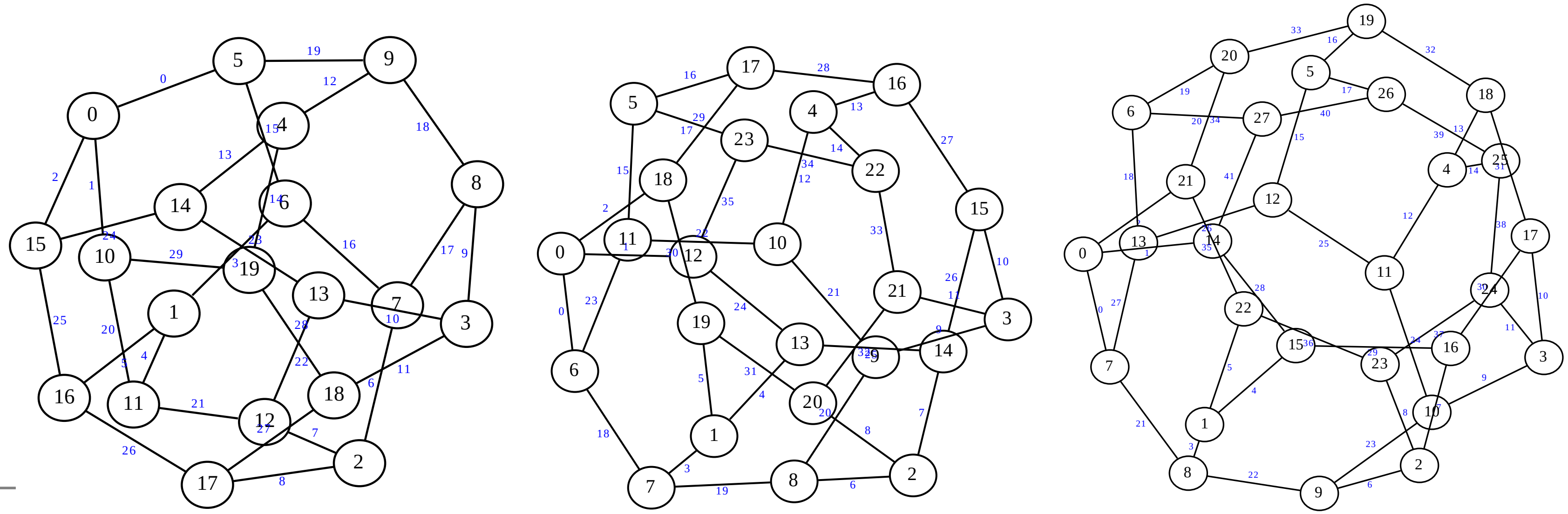
- Consider a covering problem C on the flower snark J_4
- $G_{C(J_4)} \cong (D_8 \times D_8) \ltimes C_2$
- Let $\mathcal{R}$ be the operator that applies the `sb3` reformulation to MPs
- $G_{\mathcal{R}(C(J_4))} \cong C_2 \times D_8$

Need to break more than just one orbit

By the way,

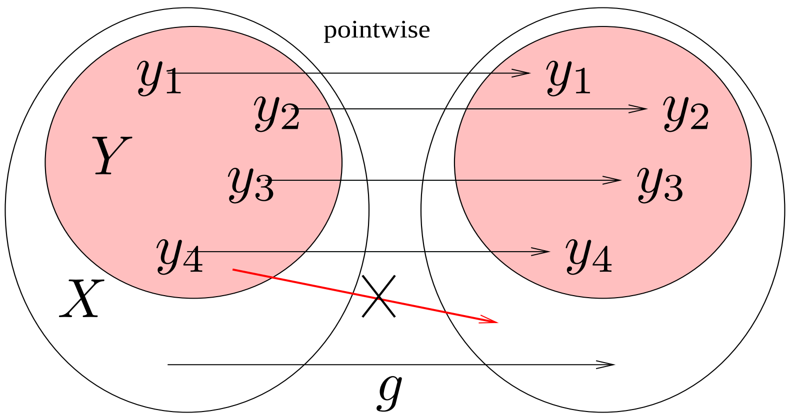
Flower snark J_k:

1. draw k copies of a 4-star
2. vertices with star size 3: $O_1, \ldots O_k$
3. other vertices: $A_1, \ldots, A_k$, $B_1, \ldots, B_k$, $C_1, \ldots, C_k$
4. draw n-cycle $A_1, \ldots, A_k$
5. draw $2n$-cycle $B_1, \ldots, B_k, C_1, \ldots, C_k$



A group "modulo an orbit"

- `sb3`: breaks symmetries of only one orbit ω
- Need the "residual group" of symmetries acting *outside* of ω (i.e. on $[n] \smallsetminus \omega$)
- Let G act on X and $Y \subseteq X$
- *Point-wise stabilizer* of Y w.r.t. G: subgroup $G^Y \leq G$ s.t. $gy = y$ for all $g \in G^Y, y \in Y$



- $G_P^\omega \leq G_P$: generate SBCs from the orbits of G_P^ω

Approach

```
# covering problem on flower snark J4
gap> G := Group([(9,13)(11,15)(25,29)(27,31),
     (5,7)(9,11)(13,15)(21,23)(25,27)(29,31),
     (2,4)(9,13)(18,20)(25,29),(10,14)(12,16)(26,30)(28,32),
     (6,8)(10,12)(14,16)(22,24)(26,28)(30,32),
     (1,2)(3,4)(5,6)(7,8)(9,14)(10,13)(11,12)(15,16)(17,18)
       (19,20)(21,22)(23,24)(25,30)(26,29)(27,28)(31,32)]);;
gap> Omg := Orbits(G);; List(Omg, function(v) return Size(v); end);
[ 4, 4, 8, 4, 4, 8 ]
gap> omega := Omg[1];
[ 1, 2, 4, 3 ];
# generate SBCs w.r.t. omega
gap> G1 := Stabilizer(G, omega, OnTuples);; Omg := Orbits(G1);;
gap> List(Omg, function(v) return Size(v); end);
[ 2, 2, 4, 4, 2, 2, 4, 4 ]
gap> omega := Omg[1];
[ 5, 7 ];
# generate SBCs w.r.t. omega
gap> G2 := Stabilizer(G1, omega, OnTuples);; Omg := Orbits(G2);;
omega := Omg[1];; # generate SBCs w.r.t. omega
gap> G3 := Stabilizer(G2, omega, OnTuples);; Omg := Orbits(G3);;
omega := Omg[1];; # generate SBCs w.r.t. omega
gap> G4 := Stabilizer(G3, omega, OnTuples);; Omg := Orbits(G4);;
omega := Omg[1];; # generate SBCs w.r.t. omega
gap> G5 := Stabilizer(G4, omega, OnTuples);
Group(()); # stop
```

The `sbstab` reformulation

1. Let $G = G_P$ and $\mathcal{C} = \varnothing$;
2. Let Ω be the set of orbits of the action of G on $[n]$;
3. Choose an orbit $\omega \in \Omega$;
4. Let $g[\omega](x) \leq 0$ be some SBCs for P and G w.r.t. ω;
5. Let $\mathcal{C} = \mathcal{C} \cup \{g[\omega](x) \leq 0\}$;
6. If the (pointwise) stabilizer G^ω is nontrivial:
   (a) replace G with G^ω
   (b) adjoin $g[\omega](x) \leq 0$ to P
   (c) go to Step 2;
7. Return $\mathcal{C}$.

Can add SBCs from all orbits chosen in Step 3 concurrently

Flower snarks

	BB tree nodes		
Order	original	`sb3`	`sbstab`
4	**13**	17	17
5	31	15	**11**
6	79	73	**51**
7	107	**55**	79
8	61	53	**25**
9	111	203	**97**

MILP results

MILP	sb3				sbstab			
Instance	*Incumbent*	*Time*	*Gap*	*Status*	*Incumbent*	*Time*	*Gap*	*Status*
air03	340160	1.66	0%	opt	340160	**1.06**	0%	opt
arki001	7.58e+06	30.35	0%	opt	7.58e+06	**13.73**	0%	opt
cod105r	-11	34.04	0%	opt	-11	**27.59**	0%	opt
cod83r	-19	53.14	0%	opt	-19	**27.04**	0%	opt
cod93r	-39	3315.69	9.31%	limit	-39	4699.36	0%	**opt**
flosn52	$+\infty$	19.3	NA	infeas	$+\infty$	**0**	NA	infeas
flosn60	$+\infty$	69.79	NA	infeas	$+\infty$	**0**	NA	infeas
jgt18	$+\infty$	2.09	NA	infeas	$+\infty$	**0.95**	NA	infeas
jgt30	$+\infty$	419.56	NA	infeas	$+\infty$	**131.7**	NA	infeas
mas74	11857.4	46.1	6.6876%	limit	11801.2	458.81	0%	**opt**
mas76	40005.4	62.29	0%	opt	40005.4	**48.73**	0%	opt
mered	$+\infty$	4.7	NA	infeas	$+\infty$	**0.99**	NA	infeas
misc06	12850.8	0.48	0%	opt	12850.8	**0.29**	0%	opt
mitre	115155	0.88	0%	opt	115155	**0.64**	0%	opt
oa66234	48	**0**	0%	opt	48	0.02	0%	opt
oa67233	48	**0**	0%	opt	48	0.01	0%	opt
oa68233	48	0.42	0%	opt	48	**0.2**	0%	opt
oa76234	56	0.02	0%	opt	56	0.02	0%	opt
oa77233	56	**0.03**	0%	opt	56	0.04	0%	opt
of5_14_7	$+\infty$	0.28	NA	infeas	$+\infty$	**0.19**	NA	infeas
of7_18_9	$+\infty$	0.17	NA	infeas	$+\infty$	**0.08**	NA	infeas
ofsub9	$+\infty$	2394.37	NA	infeas	$+\infty$	**288.64**	NA	infeas
p0201	7615	0.71	0%	opt	7615	**0.33**	0%	opt
p2756	3124	**0.87**	0%	opt	3124	1.16	0%	opt
protfold	-15	3313.45	147.778%	limit	-26	6753.95	**39.32%**	limit
qiu	-132.873	**46.35**	0%	opt	-132.873	64.89	0%	opt
rgn	82.2	0.34	0%	opt	82.2	**0.22**	0%	opt

Orbital shrinking

Orbital shrinking

- Don't *break* but *shrink* symmetry
- Aim to obtain a MILP relaxation "modulo symmetry"
- Replace:

$$\text{orbits}(x) \longrightarrow \text{added var.} z \qquad (1)$$

$$\sum_{j \in \omega} x_j \quad \rightarrow \quad z_\omega$$

- Obtain compact MILP (faster solution)
- Need reformulation where x's only appear as $\sum_{j \in \omega} x_j$

The original formulation

- ILP:

$$\left.\begin{array}{rcl} \min & cx & \\ Ax & \leq & b \\ x & \in & X^n \end{array}\right\} [P] \qquad (2)$$

where A is $m \times n$, $X = \{0, \ldots, q\}$

- Consider $G \leq G_P$
- Ω: set of orbits of G acting on $[n] = \{1, \ldots, n\}$

  E.g. $\langle(1,2)(3,4)\rangle$ acting on $[4]$ gives $\Omega = \{\{1,2\},\{3,4\}\}$

- Write i-th row of $Ax \leq b$ as:

$$\sum_{j\in[n]} A_{ij}x_j = \sum_{\omega\in\Omega}\sum_{j\in\omega} A_{ij}x_j \leq b_i \qquad (3)$$

The details 1/2

- Apply a given $\pi \in G$ to (3):

$$\sum_{\omega \in \Omega} \sum_{j \in \omega} A_{i\pi(j)} x_j \leq b_i \tag{4}$$

- $G \leq G_P \;\Rightarrow\; \exists \sigma \in S_m$ s.t. (4) is equal to:

$$\sum_{\omega \in \Omega} \sum_{j \in \omega} A_{\sigma(i)j} x_j \leq b_{\sigma(i)} \tag{5}$$

- (5) row of $Ax \leq b$ and equal to (4) $\Rightarrow$ (4) valid for P
- Sum (4) over all $\pi \in G$, get valid ineq.:

$$\sum_{\omega \in \Omega} \sum_{j \in \omega} \sum_{\pi \in G} A_{i\pi(j)} x_j \leq |G| b_i \tag{6}$$

The details 2/2

- Burnside's Lemma: ω orbit of G and f s.t. $\mathrm{dom}\, f = [n]$:

$$\forall j \in \omega \quad \sum_{\pi \in G} f(\pi(j)) = \frac{|G|}{|\omega|} \sum_{\ell \in \omega} f(\ell) \tag{7}$$

- Apply (7) to (6):

$$\sum_{\omega \in \Omega} \sum_{j \in \omega} \frac{|G|}{|\omega|} \sum_{\ell \in \omega} A_{i\ell} x_j \quad \leq \quad |G| b_i \tag{8}$$

$$\Rightarrow \quad \sum_{\omega \in \Omega} \sum_{j \in \omega} \left(\frac{1}{|\omega|} \sum_{\ell \in \omega} A_{i\ell} \right) x_j \quad \leq \quad b_i \tag{9}$$

$$\Rightarrow \quad \sum_{\omega \in \Omega} A_i^\omega \sum_{j \in \omega} x_j \quad \leq \quad b_i, \tag{10}$$

where $A_i^\omega = \frac{1}{|\omega|} (\sum_{\ell \in \omega} A_{i\ell})$

The relaxation

- For $\omega \in \Omega$, add new vars $z_\omega \in \{0, \ldots, q|\omega|\}$ to P
- Replace $\sum_{j\in\omega} x_j$ by z_ω and get:

$$\sum_{\omega\in\Omega} A_i^\omega z_\omega \leq b_i, \tag{11}$$

- Relaxation $\mathcal{A}(P)$:

  replace original system $Ax \leq b$ with (11) for each $i \leq m$

- $\mathcal{A}(P) \cup \{Ax \leq b\} \cup \{\forall \omega \in \Omega \; z_\omega = \sum_{j\in\omega} x_j\}$

  is an exact reformulation of P

- Shows $\mathcal{A}(P)$ is a relaxation

Remarks

- Extension to MINLPs s.t. only continuous vars. appear in nonlinear terms
- If $\omega = \{j\}$ for all $\omega \in \Omega$ and $j \in [n]$, then $z_\omega = x_j \Rightarrow$ relaxation same as original formulation

The relaxation is nontrivial only if $|\omega| > 1$

- If G is transitive on $[n]$ then $\Omega = \{\omega\} \Rightarrow$ relaxation consists of a single variable z_ω

The relaxation is nontrivial only if $|\Omega| > 1$

- Different choices of $G \leq G_P$ yield different orbit sets Ω and hence different relaxations

Use automated learning to pick a good G

CPLEX Results

Instance	opt	LP	Orb. Br.	CPU	BB:time2bnd
ca36243	49*	48	48	0.02	
clique9	∞*	36	**∞**	**0.06**	0.17
cod105	-12*	-18	**-12**	limit	
cod105r	-11*	-15	**-11**	**24.12**	28.36
cod83	-20*	-28	**-24**	16.78	**9.54**
cod83r	-19*	-25	**-22**	**4.44**	7.85
cod93	-40	-51	**-44**	limit	
cod93r	-38	-47	**-44**	**271.74**	446.48
cov1075	20*	18	**19**	**3.03**	79.79
cov1076	45	43	43	2.78	
cov954	30*	26	26	0.11	
mered	∞*	140	**∞**	**0.15**	3.37
O4_35	∞*	70	70	0.0	
oa36243	∞*	48	48	0.01	
oa77247	∞*	112	**∞**	**0.1**	265.92
of5_14_7	∞*	35	35	0.00	
of7_18_9	∞*	63	**∞**	**0.09**	0.15
pa36243	-44*	-48	-48	0.01	
sts27	18*	9	**12**	0.01	-
sts45	30*	15	15	0.00	
sts63	45*	21	**27**	**0.02**	1.99
sts81	61	27	**33**	**0.01**	3.92
sts135	106	45	**60**	**0.1**	109.81
esc16b	292	220	**222**	3.58	**1.84**

The end

1. L. Liberti, J. Ostrowski, **Stabilizer-based symmetry breaking constraints for mathematical programs**, JOGO, to appear
2. A. Costa, P. Hansen, L. Liberti, **On the impact of symmetry-breaking constraints on spatial Branch-and-Bound for circle packing in a square**, *Discrete Applied Mathematics*, **161**:96-106, 2013
3. M. Fischetti, L. Liberti, **Orbital Shrinking**, in ISCO12 Proceedings (LNCS), 2012
4. L. Liberti, **Reformulations in mathematical programming: Automatic symmetry detection and exploitation**, *Mathematical Programming A*, **131**:273-304, 2012
5. L. Liberti, **Symmetry in Mathematical Programming**, in S. Leyffer and J. Lee (eds.), Mixed Integer Nonlinear Programming, *IMA Series* **154**:263-286, Springer, New York, 2012
6. A. Costa, Pierre Hansen, Leo Liberti, **Static symmetry breaking in circle packing**, in U. Faigle, R. Schrader, D. Herrmann (eds.), *CTW10 Proceedings*, 47-50, Köln, 2010.
7. L. Liberti, **Automatic generation of symmetry-breaking constraints**, in B. Yang, D.-Z. Du and C.A. Wang (eds.) *COCOA08 Proceedings*, LNCS **5165**:328-338, Springer, New York, 2008