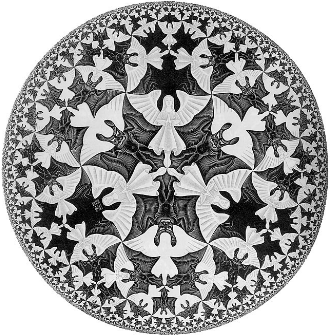


Symmetry in Integer Programming

Jim Ostrowski

Industrial and Information Engineering
University of Tennessee

jostrows@utk.edu

My Favorite Integer Program (Jeroslow Problem)

An Easy Problem

$$\min_{x \in \{0,1\}^{n+1}} \{x_{n+1} \mid 2x_1 + 2x_2 + \ldots + 2x_n + x_{n+1} = 2k + 1\}$$

- Yes, this problem is very easy!
- Let's try to solve it using branch and bound.
- (This problem comes from Bertsimas and Tsitsiklis's Introduction to Linear Programming)

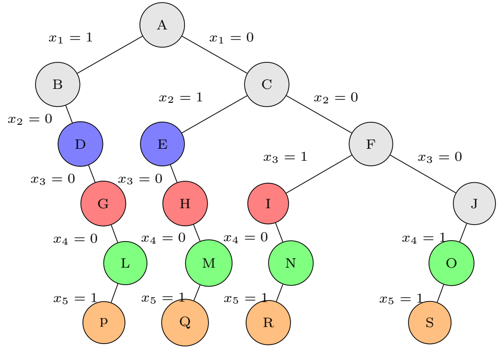
Solving an Easy Problem with Branch and Bound

n	k	Time (seconds)	Nodes
20	5	3.24	54,262
20	6	6.97	116,278
20	7	12.24	203,400
20	8	17.83	293,928
20	9	21.68	352,714
20	10	21.74	352,714
25	5	13.86	23,228
25	6	38.96	657,798
25	7	92.71	1,562,273
30	5	42.7	736,284
30	6	160.15	2,629,573

What happened!?!

The Branch-and-Bound Tree (N=4 K=1)

$$\min_{x \in \{0,1\}^5} \{x_5 \mid 2x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 = 3\}$$



- Subproblem at node D can be written as:

$$\min_{x \in \{0,1\}^5} \{x_5 \mid 2x_3 + 2x_4 + x_5 = 1\}$$

- Subproblem at node E can be written as:

$$\min_{x \in \{0,1\}^5} \{x_5 \mid 2x_3 + 2x_4 + x_5 = 1\}$$

Preliminaries

- $\Pi^n \stackrel{\text{def}}{=}$ the set of all permutations of $I^n = \{1, \ldots, n\}$.
- Given $\pi \in \Pi^n$, $\sigma \in \Pi^m$, let $A(\pi, \sigma) \stackrel{\text{def}}{=}$ the matrix obtained by permuting the columns of A by π and the rows of A by σ, i.e. $A(\pi, \sigma) = P_\sigma A P_\pi$,
- The symmetry group $\mathcal{G}$ of the matrix A is the set of permutations of the columns such that there is a corresponding permutation of the rows that when applied yields the original matrix

$$\mathcal{G}(A) \stackrel{\text{def}}{=} \{\pi \in \Pi^n \mid \exists \sigma \in \Pi^m \text{ such that } A(\pi, \sigma) = A\}$$

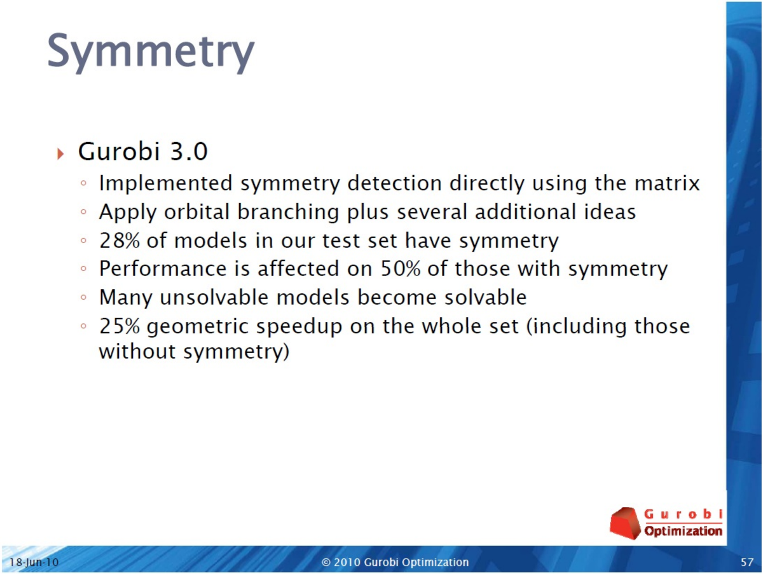
Facts About Symmetry

- $\pi(x) \stackrel{\text{def}}{=} (x_{\pi(1)}, x_{\pi(2)}, \ldots x_{\pi(n)})$ permutes the coordinates of x
- $\pi \in \mathcal{G}(A) \Rightarrow x$ feasible $\Leftrightarrow \pi(x)$ feasible .
- $e^T x = e^T \pi(x)$

Symmetry in the Real World

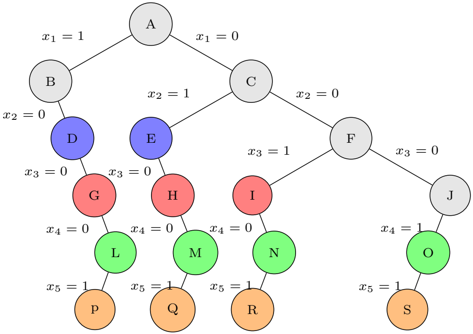
- Symmetry appears in
  - graph coloring problems,
  - cutting stock problems,
  - scheduling problems
  - and more!
- In a recent paper, Leo Liberti shows that many commonly used test problems for integer programming contain symmetry.
- CPLEX has recently implemented symmetry handling techniques.
- Gurobi implemented techniques based on an idea from my thesis, "Orbital Branching".

Don't Take My Word For It

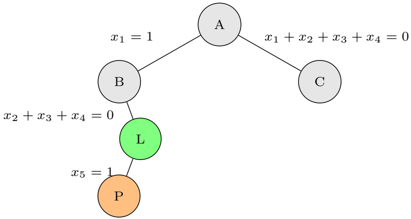


General Idea

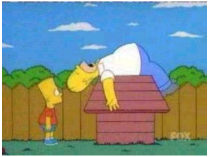
Turn this:



Into this:



More Preliminaries



- For a set $S \subseteq \mathrm{I}^n$, the orbit of S with respect to $\mathcal{G}(A)$ is the set of all subsets of I^n to which S can be sent by permutations in $\mathcal{G}(A)$:

$$\mathrm{orb}(S) \stackrel{\text{def}}{=} \{S' \subseteq \mathrm{I}^n \mid \exists \pi \in \mathcal{G}(A) \text{ such that } S' = \pi(S)\}.$$

Orbital Branching—The Whole Idea

- Let $O \in \Gamma^a$ be an orbit of the symmetry group of subproblem a.
- Surely we can branch as

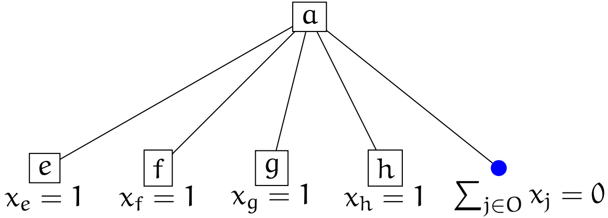
$$\sum_{i \in O} x_i \geq 1 \quad \text{or} \quad \sum_{i \in O} x_i \leq 0.$$

- If at least one variable $i \in O$ is going to be one, and they are all "equivalent", then you might as well pick (i^*) one arbitrarily.

$$x_{i^*} = 1 \quad \text{or} \quad \sum_{i \in O} x_i = 0$$

Another Way to View Orbital Branching

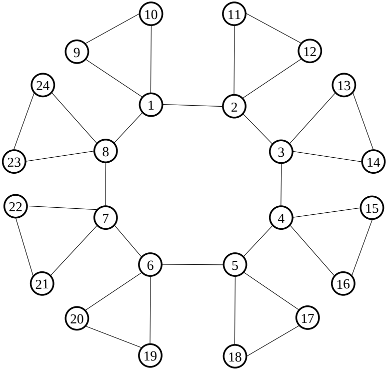
- Suppose that you have found that the variables x_e, x_f, x_g and x_h share an orbit at node a, $O = \{e, f, g, h\}$.
- Then you can surely branch as:



- But the best solution you can find from nodes f, g, and h will be the same as the best solution you can find from node e
- In fact, solutions will be isomorphic
- $\Rightarrow$ Prune nodes f, g, and h

Branching with Symmetry

$$\max_{x \in \{0,1\}^{|V|}} \left\{ \sum_{i \in V} x_i \mid x_i + x_j \leq 1 \ \forall (i,j) \in E \right\}.$$



Intuition

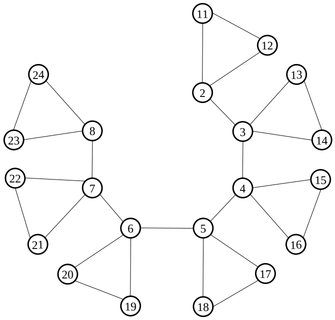
- Variables $\{x_1, x_2, \ldots x_8\}$ are "the same"
- Variables $\{x_9, x_{10}, \ldots x_{24}\}$ are "the same"

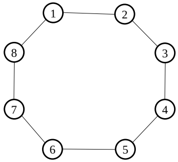
Example: Orbital Branching Subproblems

- Branching on orbit $\{9, 10, \ldots, 24\}$, gives subproblems:

$$x_9 = 1$$

$$\sum_{j=9}^{24} x_j = 0$$

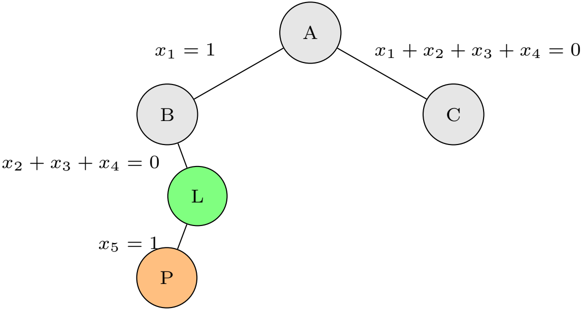




Orbital Branching and Jeroslow

Jeroslow Problem

$$\min_{x \in \{0,1\}^5} \{x_5 \mid 2x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 = 3\}$$



Computational Results – Some Symmetric BIPs

- (Binary) Error Correcting Codes (`cod(n,d)`): Find maximum number of $(0, 1)$ $n-$vectors such that the Hamming distance between each pair is $\geq d$
- Covering Design (`cov(v,k,t)`): $v > k > t$: Find minimum number of k-sets of $\{1, \ldots, v\}$ to "cover" all t-sets of $\{1, \ldots, v\}$.
- Covering Code (`codbt(b,t)`): Find minimum number of "codewords" such that *every* word in the alphabet is at most a (Hamming) distance 1 from a codeword. *Football Pool Problem*: `codbt(0,6)`
- Steiner Triple System: (`sts(n)`): Find the "incidence width" of a Steiner Triple System of order n

Measure for Measure – Gurobi v3.0

Instance	Symmetry = 0			Symmetry = 2		
	Time	Gap%	Nodes	Time	Gap%	Nodes
cod105	7200	50.0	150			
cod83	7200	15.0	724601			
cod93	7200	20.0	108572			
codbt05	7200	7.4	352025			
codbt33	8	0.0	604			
codbt42	159	0.0	75569			
codbt61	10	0.0	1485			
cov1053	7200	5.9	919836			
cov1054	7200	2.0	189645			
cov1075	7200	5.0	549355			
cov954	58	0.0	31950			
sts27	1	0.0	4044			
sts45	18	0.0	61194			
sts63	7200	4.4	8698168			
sts81	7200	16.4	3252747			

Measure for Measure – Gurobi v3.0

	Symmetry = 0			Symmetry = 2		
Instance	**Time**	**Gap%**	**Nodes**	**Time**	**Gap%**	**Nodes**
cod105	7200	50.0	150	173	0.0	7
cod83	7200	15.0	724601	6	0.0	372
cod93	7200	20.0	108572	905	0.0	54650
codbt05	7200	7.4	352025	7200	3.7	359268
codbt33	8	0.0	604	6	0.0	401
codbt42	159	0.0	75569	111	0.0	45912
codbt61	10	0.0	1485	7	0.0	950
cov1053	7200	5.9	919836	77	0.0	10958
cov1054	7200	2.0	189645	2330	0.0	103657
cov1075	7200	5.0	549355	17	0.0	665
cov954	58	0.0	31950	1	0.0	166
sts27	1	0.0	4044	0	0.0	78
sts45	18	0.0	61194	23	0.0	34839
sts63	7200	4.4	8698168	85	0.0	43135
sts81	7200	16.4	3252747	70	0.0	6317

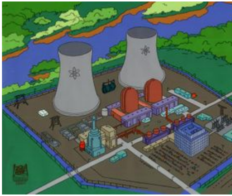
Measure for Measure – CPLEX v12.1

	Symmetry = 0			Symmetry = 5		
Instance	Time	Gap%	Nodes	Time	Gap%	Nodes
cod105	7200	52.4	13201	606	0.0	1120
cod83	7200	14.3	1418001	79	0.0	15452
cod93	7200	18.9	389028	7200	6.3	639001
codbt05	7200	5.6	1035046	150	0.0	23059
codbt33	8	0.0	1049	1	0.0	14
codbt42	89	0.0	84039	4	0.0	2141
codbt61	8	0.0	1833	1	0.0	61
cov1053	7200	5.9	1495461	2234	0.0	448008
cov1054	7200	2.0	191970	7200	2.0	169371
cov1075	7200	6.4	1505168	57	0.0	12227
cov954	64	0.0	36563	3	0.0	1351
sts27	0	0.0	3532	0	0.0	1307
sts45	10	0.0	59890	6	0.0	28775
sts63	1585	0.0	7692765	736	0.0	3607609
sts81	7200	13.1	23933498	7200	11.5	23415204

Symmetry in Scheduling Problems

- Scheduling problems can have a great deal of symmetry.
  - Identical machines.
  - Identical jobs.
- This symmetry is more structured than typical problems, allowing us to better exploit the symmetry.

The Unit Commitment Problem



- The Unit Commitment (UC) problem is a large scale MINLP that finds a low-cost generating schedule for power generators.
- These problems have quadratic objective functions, and transmission constraints can be highly nonlinear.
- These problems are typically solved as integer programs.
- I have been working on developing formulations and algorithms to solve the UC problem faster.

The Basic Problem

The UC Problem

$$\text{Minimize} \sum_{t \in T} \sum_{j \in J} c_j(p_j(t))$$

subject to

$$\sum_{j \in J} p_j(t) \geq D(t), \quad \forall\, t \in T$$

$$p_j \in \Pi_j, \quad \forall j \in J.$$

- $c(p(t))$ gives the cost of generator j producing $p_j(t)$ units of electricity at time t.
- In every time periods, demand $D(t)$ must be met.
- Each generator must work within its physical limits (ramping constraints, minimum shut down times, etc.).

Symmetry in The Unit Commitment Problem

$$\begin{array}{c c}
\text{Time} & \begin{array}{cc} \text{Gen 1} & \text{Gen 2} \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} &
\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}
\end{array}$$

On/Off status of Generators 1 & 2

- If generators 1 and 2 are identical, permuting their schedules will give an equivalent solution.
- Permutations schedules between like generators are symmetries.
- Orbital branching works, but we have developed more a tailored version for this problem.

Symmetry in The Unit Commitment Problem

$$
\begin{array}{c c}
\text{Time} & \begin{array}{cc}\text{Gen 1} & \text{Gen 2}\end{array} \\
\begin{array}{c}1\\2\\3\\4\\5\\6\\7\\8\\9\\10\end{array} &
\begin{pmatrix}
0 & 1 \\
1 & 1 \\
1 & 1 \\
1 & 1 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 1 \\
0 & 1 \\
0 & 1
\end{pmatrix}
\end{array}
$$

On/Off status of Generators 1 & 2

- If generators 1 and 2 are identical, permuting their schedules will give an equivalent solution.
- Permutations schedules between like generators are symmetries.
- Orbital branching works, but we have developed more a tailored version for this problem.

Symmetry in UC

- Multiple generators of the same type can introduce symmetry into the problem.
- Suppose we had J types of generators. We can think of the on/off status of an optimal solution to UC as J many $T \times n_j$ 0/1 matrices. All column permutations of each of these matrices are symmetries in the UC problem.

Finding Symmetry in Subproblems is Easy

$$x^i = \begin{pmatrix} 1 & 1 & 1 & ? & ? \\ ? & ? & ? & ? & ? \\ 0 & 0 & ? & ? & ? \\ ? & ? & ? & ? & ? \end{pmatrix}$$

- Suppose x^i represented a partial solution for the on/off status of generators of type i.
- All relabellings of the columns of x^i are in the symmetry group at the root node.
- After fixings, columns 4 and 5 are still symmetric (neither column contains variables fixed by branching).
- Even though columns 1 and 2 contain fixed variables, they are still symmetric (the fixings in column 1 are identical to the fixings in column 2).

Orbital Branching on UC

$$x^i_{LP} = \begin{pmatrix} .55 & .55 & .55 & .55 \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix}$$

- Suppose we chose to perform orbital branching, and branch on the orbit representing the first row of x^i.
- Branch is $x^i_{1,1} = 1 \vee \sum_{j=1}^4 x^i_{1,j} = 0$.
- It is likely that the right branch is strong, but how strong is the left branch? My current LP solution is already suggesting that more than one machine of type i should be on.
- This is not a very useful branch (but better than non-orbital branching!).

Modified Orbital Branching in UC

- Suppose you were branching on the first row of

$$x^i_{LP} = \begin{pmatrix} .55 & .55 & .55 & .55 \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix}.$$

- What about the disjunction "Either at least 3 generators are on or at least 2 are off"?
- Using symmetry, we can strengthen this to

$$\{x^i_{1,1} = x^i_{1,2} = x^i_{1,3} = 1\} \vee \{x^i_{1,3} = x^i_{1,4} = 0\}.$$

Computational Results

# of Generators	CPLEX Only				OB		Modified OB	
	Dynamic Search		B&B					
	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time
21	923	544.0	-	-	5498	482.8	190	57.9
23	499	386.5	82539	6415.5	3190	342.2	390	78.9
23	878	1227.6	-	-	-	-	715	308.6
24	3259	1169.3	-	-	6259	691.2	517	155.7
26	972	978.4	-	-	37150	5461.1	206	138.4
26	516	529.5	-	-	2574	366.2	180	68.4
26	558	558.4	-	-	3830	628.6	219	107.4
26	500	425.6	-	-	13790	1552.5	158	74.4
26	515	465.3	-	-	27890	2015.2	218	111.4
26	4369	1320.9	-	-	16805	1758.5	341	104.0
27	579	535.3	-	-	3323	494.5	187	105.4
27	545	594.3	-	-	-	-	7222	1339.8
28	522	679.5	-	-	-	-	720	307.7
28	532	444.0	-	-	5162	578.0	358	107.8
29	1182	975.6	-	-	-	-	-	-
30	1793	1514.6	-	-	-	-	2523	631.0
30	541	862.7	-	-	-	-	1252	381.8
31	1268	1210.3	-	-	4010	6553.9	6521	1197.4
31	538	783.5	-	-	13475	3660.6	113	107.3
31	537	712.5	-	-	-	-	842	296.1
31	1172	1360.8	-	-	19929	4599.6	570	220.7
34	544	739.0	-	-	-	-	4201	1401.9
35	600	1204.9	-	-	21612	4697.3	1190	404.5
37	4029	2808.2	-	-	-	-	946	447.0
42	994	1540.1	-	-	-	-	495	396.8

Thank You, Come Again

Results

- Symmetry is an important area in optimization.
- Ignoring the presence of symmetry can make problems unsolvable.
- Symmetry can be found in a variety of interesting real-world problems.
- Problem specific symmetry-breaking techniques can improve computation time considerably.