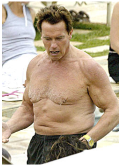
Symmetric Integer Linear Optimization (SILO!)

Jeff Linderoth

Dept. of Industrial and Systems Engineering
Univ. of Wisconsin-Madison
linderoth@wisc.edu

Jim Ostrowski
University of Tennessee

François Margot
Carnegie Mellon University



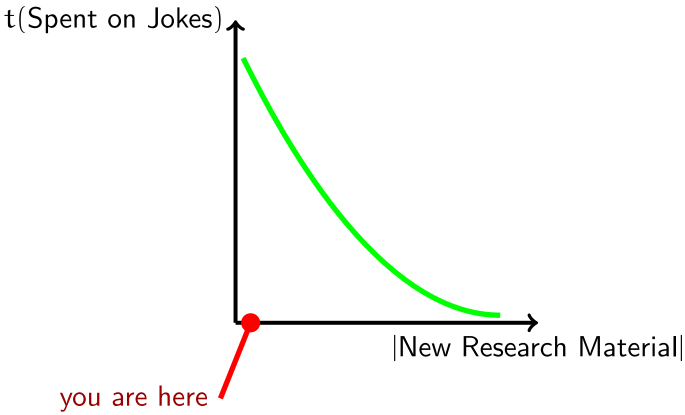
SILO UW-Madison March 5, 2014

Italian Arnolds

Fabrizio Rossi
Stefano Smriglio

Università di L'Aquila

What You Are In For



Integer Linear Optimization (ILO)

$$\min_{x \in \{0,1\}^n} \{c^T x \mid Ax \geq b\} \qquad \text{(ILO)}$$

- Today, we care about $A \in \{0,1\}^{m \times n}$ and

$$\text{SCP} \min_{x \in \{0,1\}^n} \{1^T x \mid Ax \geq 1\} \quad \text{or} \quad \text{SPP} \max_{x \in \{0,1\}^n} \{1^T x \mid Ax \leq 1\}$$

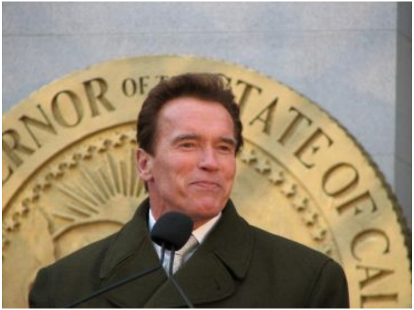
Outline

1. What is Symmetry in ILO?
2. Motivation - The Football Pool Problem
3. Orbital Branching
4. Isomorphism Pruning
5. Flexible Isomorphism Pruning
6. Computational Results

Warning! This All Happened Some Time Ago...

"This is really embarrassing. I just forgot our state governor's name, but I know that you will help me recall him."

—Arnold, speaking to a taxpayer advocacy group



- I hope I recall enough to give an informative talk.

Football Pool Problem

- We motivate our solve for SILO (SCP) via gambling

- Predict the outcome of v soccer matches
- Giant Prize if all v are correct
- Also win if $v - 1$ are correct: you mis-predict at most 1 game

The Football Pool Problem

What is the minimum number of tickets you must buy to assure yourself a win?

Football ILO

- Let N be the set of possible outcomes (also the set of tickets) $(|N| = 3^{\nu})$
- Binary variables: $x_j = 1$ if I purchase ticket $j \in N$
- Let $A \in \{0,1\}^{|N| \times |N|}$ with $a_{ij} = 1$ iff ticket $j \in N$ is a winner for outcome $i \in N$

ILO Formulation

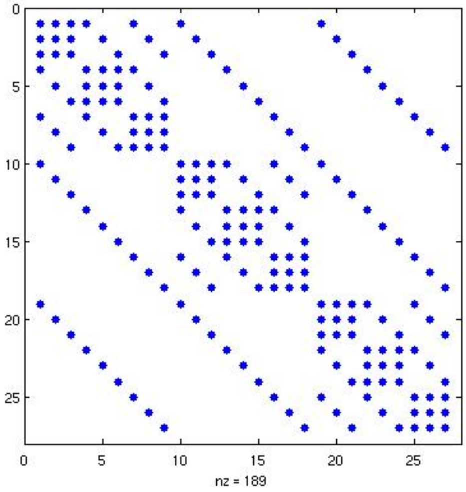
$$\min \mathbf{1}^{\top} x$$

$$\begin{aligned} \text{s.t.} \quad Ax &\geq \mathbf{1} \\ x &\in \{0,1\}^{|N|} \end{aligned}$$

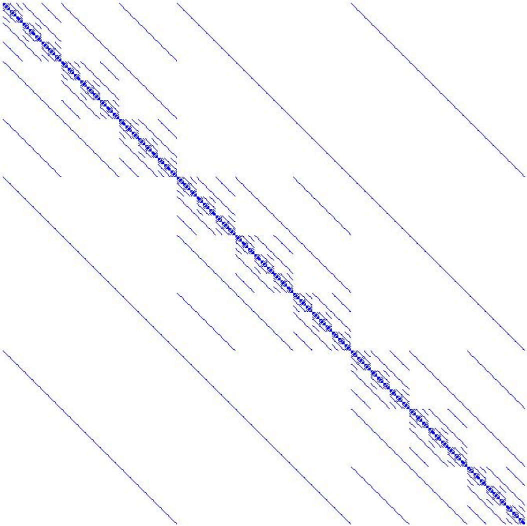
Football Matrix, $v = 3$

Playing Football

Chose columns to cover all rows



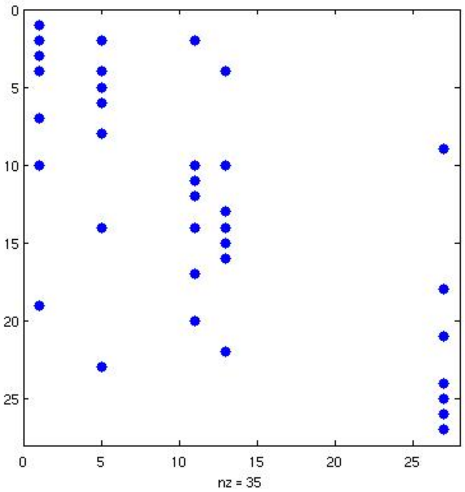
It's Pretty! The Football Matrix, $\nu = 6$



$\nu = 3$, Solution #1

Answer #1

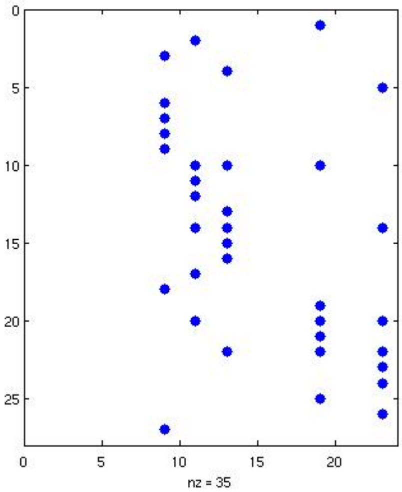
M1	M2	M3
W	W	W
L	L	W
L	W	L
W	L	L
D	D	D



$\nu = 3$, Solution #2

Answer #2

M1	M2	M3
D	W	W
L	L	W
L	W	L
D	L	L
W	D	D



Solutions for $v = 3$

Answer #1

M1	**M2**	**M3**
W	W	W
L	L	W
L	W	L
W	L	L
D	D	D

Answer #2

M1	**M2**	**M3**
D	W	W
L	L	W
L	W	L
D	L	L
W	D	D

- These solutions are isomorphic.
  - Swap W $\leftrightarrow$ D in the first ticket
- There are LOTS of isomorphic solutions:
  1. "Rename" W,L,D for any subset of the matches: $(3!)^v$
  2. Reorder the matches: $v!$
- There are $(3!)^3(3!) = 1296$ equivalent solutions for $v = 3$
- There are $(3!)^6(6!) = 33,592,320$ equivalent solutions for $v = 6$

How Many Must I Buy?

Known Optimal Values

v	1	2	3	4	5
$\lvert C_v^* \rvert$	1	3	5	9	27

The Football Pool Problem

What is $\lvert C_6^* \rvert$?

- Despite significant effort on this problem for > 40 years, it is only known that

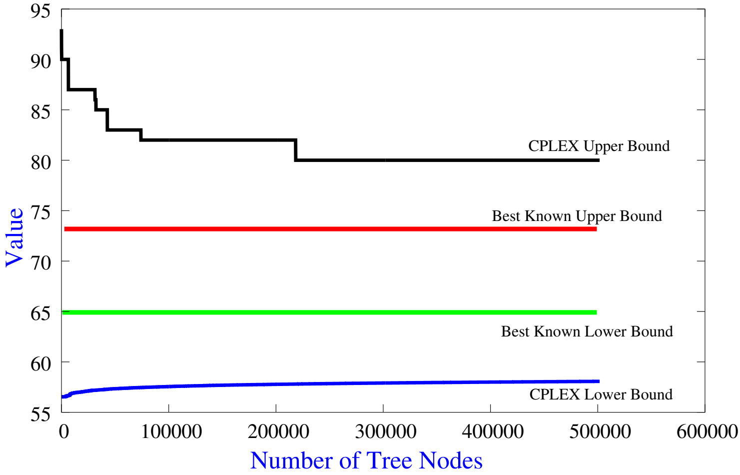
$$6571 \leq C_6^* \leq 73$$

Hooray For Us!

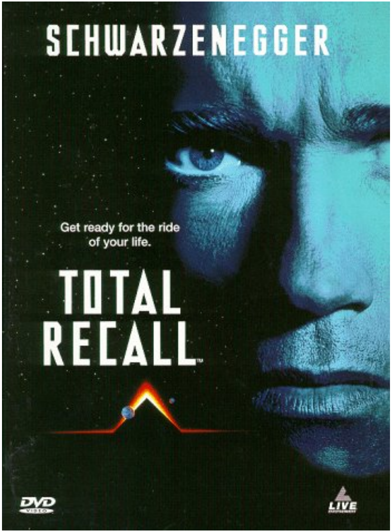
- We were able to improve the lower bound to 71
- It only took 140 CPU years!

CPLEX Can't Solve Every IP

- Roughly 10^8 universe lifetimes in order to establish that $|C_6^*| > 72$



A Review of Symmetry and Algebra



Symmetry

- Let Π^n be collection of permutations of $\{1, 2, \ldots, n\}$
- Given $\lambda \in \mathbb{R}^n$, $\pi \in \Pi^n$ acts on λ by permuting its coordinates: $\pi(\lambda) = (\lambda_{\pi_1}, \lambda_{\pi_2}, \ldots \lambda_{\pi_n})$.

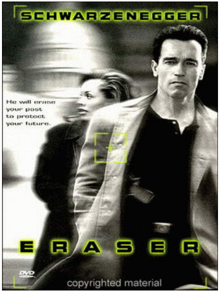
- $\pi \in \Pi^n$ is a symmetry of IP if...
  1. x feasible $\Leftrightarrow \pi(x)$ feasible
  2. $c^T x = c^T \pi(x)$

- The set of symmetries of IP (with composition of permutations) forms the symmetry group of IP

$$\mathcal{G}(\text{IP}) = \{\pi \in \Pi^n \mid \pi(x) \in \mathcal{F}, c^T x = c^T \pi(x) \quad \forall x \in \mathcal{F}\},$$

where $\mathcal{F} = \{x \in \{0,1\}^n \mid Ax \geq b\}$ is the set of feasible solutions

Symmetry and Branching

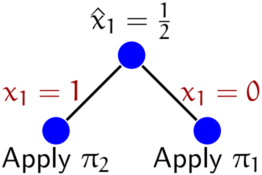


Symmetry "Erases" Branching Decision

- In the presence of symmetry, branching does not effectively change the solution to the LP relaxation

---

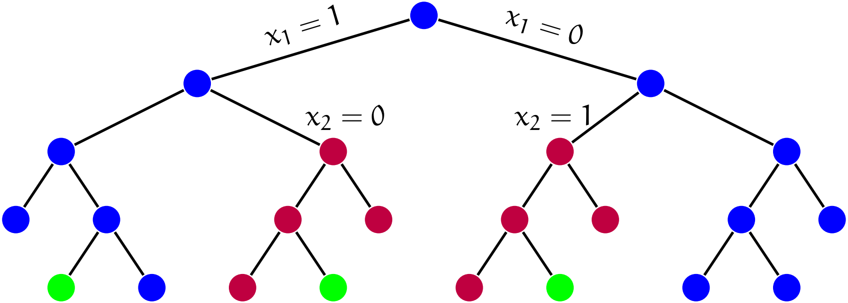
- Let $\hat{x} = (1/2, 0, 1/4, 1, \ldots)^T$ be a solution to the LP relaxation
- If $\pi_1 = (1,2) \in \mathcal{G}$, $\pi_2 = (1,4) \in \mathcal{G}$ then $z_i^- = z_i^+ = z_{LP}$.
- You are guaranteed to have a bad branch



Searching with Symmetry

"Big Mistake."
Jack Slater, Last Action Hero

- Suppose the permutation $(1, 2) \in \mathcal{G}$



- You evaluate many completely equivalent (isomorphic) subtrees

About Symmetry Groups

- $\mathcal{G}(\text{IP})$ is a property of the feasible region: $\mathcal{F} = \emptyset \Rightarrow \mathcal{G}(\text{IP}) = \Pi^n$
- For our methods, we can work with any subgroup $\Gamma \subset \mathcal{G}(\text{IP})$
- If $c = \mathbf{1}$, $b = \mathbf{1}$, we can use the symmetry group of the matrix A:

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

- Given A of "reasonable" size, there exist software packages (`nauty`, `saucy`) that can compute (generators of) $\mathcal{G}(A)$ "effectively"
  - Actual algorithm is exponential, but in general works quickly

Orbits

- For a point $z \in \mathcal{Z}$, the orbit of z under $\mathcal{G}$ is the set of all elements of $\mathcal{Z}$ to which z can be sent by permutations in $\mathcal{G}$:

$$\mathrm{orb}(\mathcal{G}, z) \stackrel{\text{def}}{=} \{\pi(z) \mid \pi \in \mathcal{G}\}.$$

- Consider the orbits of each of coordinate axes: $e_j, j \in N$
- By definition, if $e_j \in \mathrm{orb}(\mathcal{G}, e_k)$ then $e_k \in \mathrm{orb}(\mathcal{G}, e_j)$, i.e. the variables x_j and x_k share the same orbit. Therefore, the union of the orbits

$$\mathcal{O}(\mathcal{G}) \stackrel{\text{def}}{=} \bigcup_{j=1}^{n} \mathrm{orb}(\mathcal{G}, e_j)$$

forms a partition of $N = \{1, 2, \ldots, n\}$, which we refer to as the orbits of $\mathcal{G}$.

- The orbits encode which variables are "equivalent" (symmetric) with respect to the symmetry $\mathcal{G}$.

Ugh... More Notation

- Branch-and-bound node $a = (F_1^a, F_0^a)$,
  - F_1^a: Set of variables fixed to one
  - F_0^a: Set of variables fixed to zero
- $\mathcal{F}(a)$: The set of feasible solutions to the IP at node a

---

- The stabilizer of a set S in $\mathcal{G}$ is the set of permutations in $\mathcal{G}$ that send S to itself: $\mathrm{stab}(S, \mathcal{G}) = \{\pi \in \mathcal{G} \mid \pi(S) = S\}$.
- $\mathrm{stab}(S, \mathcal{G})$ is a subgroup of $\mathcal{G}$

The Upshot

- As we fix variables (to 1) at node a, the symmetry "remaining" in the problem becomes $\mathrm{stab}(\chi^a_{F_1}, \mathcal{G})$

Orbital Branching: A Simple Idea

Orbital Branching

- A way to exploit symmetry in your branching decision
- Let $O \in \mathcal{O}(\mathcal{G}(\text{IP}))$ be an orbit of the symmetry group of the IP.
- Surely we can branch as

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

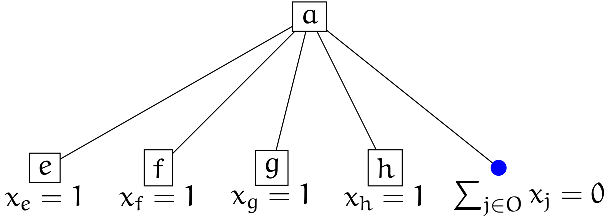
- If at least one variable $i \in O$ is going to be one, and they are all "equivalent", then you may as well pick (i^*) one arbitrarily.

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

- No, really. That's it. :-)

An Alternative View of 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

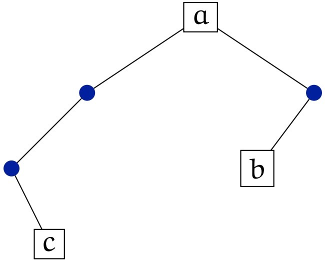
Orbital Branching Theorems

Theorem: OB is Valid

All optimal solutions are not eliminated.

Theorem: OB Reduces Symmetry

Let b and c be any two subproblems in the enumeration tree. Let a be the first common ancestor of b and c. If $x \in \mathcal{F}(b)$ and $y \in \mathcal{F}(c)$, then $\nexists \pi \in \mathcal{G}(A(F_0^a, F_1^a))$ with $\pi(x) = y$.



But Can We Do Better!?

Can we branch and prune such that $x \in \mathcal{F}(b)$ and $y \in \mathcal{F}(c)$ are not equivalent (isomorphic) with respect to the original symmetry group $\mathcal{G}$?

Isomorphism Pruning

The admitedly very bad **Joke**

Can we "terminate" the search without exploring equivalent solutions?

Search the Fundamental Domain!

It's Fundamental

- The (minimal) Fundamental Domain of a feasible region $\mathcal{F}$ with respect to a (permutation) group $\mathcal{G}$ is the *smallest subset* $F \subseteq \mathcal{F}$, such that if $x \in \mathcal{F}$, then $x = \pi(y)$ for some $\pi \in \mathcal{G}$, $y \in F$

Key Idea: Exploit Symmetry

- Restrict search to F, not $\mathcal{F}$!
- Put another way: for any feasible solution, x, we only need to consider one element in $\mathrm{orb}(\mathcal{G}, x)$.
- By definition, any method that restricts itself to a fundamental domain $\mathcal{F}$ will not encounter isomorphic solutions

Creating a Fundamental Domain

Order the solutions lexicographically

- Adding the following lexicographic ordering constraints to $\mathcal{F}$ creates a fundamental domain:

$$[2^n \ 2^{n-1} \ \ldots \ 4 \ 2]^T x \leq [2^n \ 2^{n-1} \ \ldots \ 4 \ 2]^T \pi(x) \quad \forall \pi \in \mathcal{G}$$

Not an Optimal Solution

- 2^n term creates numerical instability.
- $|\mathcal{G}|$ can be large

Solutions?

- We need to find clever ways to enforce lexicographic inequalities without adding them to the problem formulation.

Dr. Clever

- Isomorphism Pruning, developed by Margot (2002, 2003) in the context of IP, provides an algorithm for testing (at each node) if the set of variables fixed by branching violate a lexicographic inequality
- If a lexicographic inequality is violated, the node is pruned $\Rightarrow$ only a fundamental domain is searched

Isomorphism Pruning Theorem:

For node a and symmetry group $\mathcal{G}$, let F_1^a be the set of variables fixed to one (by branching decision) at node a. If F_1^a is not lexicographically minimal with respect to $\mathrm{orb}(\mathcal{G}, \chi_{F_1^a})$, node a can be pruned.

Isomorphism Pruning “Problems”

Problem #1

- Algorithm for testing if $\chi_{F_1^a}$ is lex min member of $\text{orb}(\mathcal{G}, \chi_{F_1^a})$ is exponential in the size of F_1^a, but is “reasonably fast” if the tree is not very deep.
- This can be done using computational algebra packages such as `GAP`.

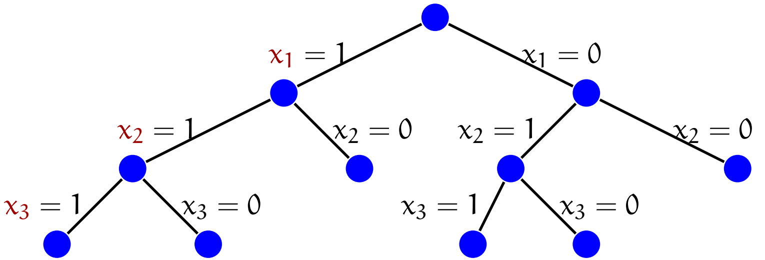
BIG Problem #2

- We need to ensure that the lex min member of $\text{orb}(\mathcal{G}, \chi_{F_1^a})$ occurs somewhere in the tree
  - (if the node would not otherwise be pruned by bound or infeasibility)

What to do?

- Branch on variables in lexicographic order

Isomorphism Pruning Tree



No Flexibility!

- You must branch on variable, regardless of impact on bound.
- Even if $\hat{x}_d$ is not fractional at level d
- This is typically a very bad idea for branch and bound

I am the Greatest Thesis Advisor Ever!

"Why can't we define a 'local ordering' of the variables to define lexicographic min?"

"Because if we could, then I am sure that François would have thought of it"

Jim's Reaction

- Thankfully, Jim rarely listens to me...

Flexible Isomorphism Pruning

François Does Not Think of Everything!

- Each node a has rank vector R^a.
- $R^a[i] = j$ implies x_i was branched on at the ancestor node of a at depth j. Variables not fixed by a branching at node a are assigned the rank (-1).

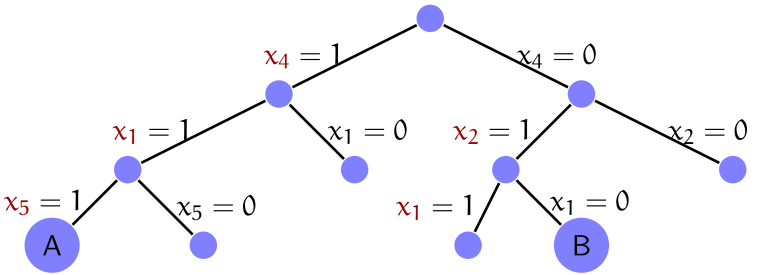
Flexible Isomorphism Theorem:

For node a and symmetry group $\mathcal{G}$, let F_1^a be the set of variables fixed to one (by branching decision) at node a. If $R^a(F_1^a)$ is not lexicographically minimal with respect to $R^a(\mathrm{orb}(\mathcal{G}, \chi_{F_1^a}))$, node a can be pruned.

The Good Part!

- This is true for any branching decisions.
- No additional (costly) computations are needed compared to regular isomorphism pruning

Most Flexible Isomorphism Pruning Tree



i	1	2	3	4	5
$R^A(i)$	2	-1	-1	1	3

i	1	2	3	4	5
$R^B(i)$	3	2	-1	1	-1

Flexible Isomorphism Pruning

Facts

- At every node a we still have an implicit set of lexicographic inequalities:

$$\sum_{i=1}^{n} 2^{n+1-R^{a}(i)} x_i \leq \sum_{i=1}^{n} 2^{n+1-R^{a}(\pi(i))} x_i \quad \forall \pi \in \mathcal{G}$$

- The fundamental domain searched is *not* a polyhedron
- The branching decisions impact the fundamental domain that is searched. Only when the problem is done solving is the exact fundamental domain known.

Proof Intuition

- Because these constraints are "local" they do not affect behavior at other nodes in the tree.

Branching: What To Do?

Combine "Strength" of Isomorphism Pruning

With "Flexibility" of Branching on Any Variable/Orbit

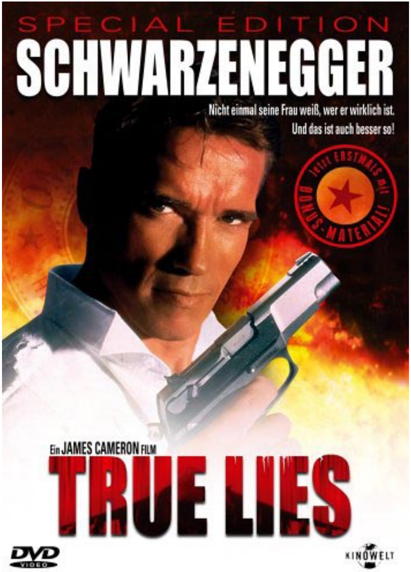
Which Branchable Orbit to Choose?

- Given $\hat{x}$ and (branchable) orbits $O_1, O_2, \ldots O_p$ at node a, which orbit should be choose?
- We investigated (so far) lots of different branching rules. I will only talk of 3

---

1. Branch Min Index: Branch on orbit that contains the smallest unfixed variable index. "Close" to Margot's original branching
2. Branch Largest LP Solution: Branch on Orbit with most LP solution
3. Strong Branching: For each orbit, create orbital branching dichotomy. Evaluate the two resulting children, and choose "the best"

Computational Results



Instance Families

- (Binary) Error Correcting Codes (`cod(n,d)`): Find maximum number of $(0, 1)$ $n-$vectors such that 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 is at most a (Hamming) distance 1 from a codeword.
- Steiner Triple System: (`sts(n)`): Find the "incidence width" of a Steiner Triple System of order n

Number of Nodes

Instance	CPLEX v11 w/Sym	Largest LP O.B.	Largest LP IsoP.	Min Index O.B.	Min Index IsoP.	Str. Branching O.B.	Str. Branching IsoP.
cod83	9338	35	35	23	23	23	23
cod93	287998*	3531	2933	711	269	129	127
cov954	1226	113	113	701	549	39	39
cov1053	262628	3535	2639	893	607	569	435
cov1054	94949*	52509	45577	567	417	426	311
cov1075	21076	107	105	471	367	71	63
codbt42	-	73	73	1059	893	37	37
codbt05	107816	303	285	1521	1245	103	103
sts45	19931	9373	4469	6037	1553	1861	1203
sts63	4805781	37243	6327	12365	4303	3221	2309
sts81	13361288*	2361	527	2995	585	991	509

Tale of the Tape—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
	"I'm the party pooper."			“Hasta la vista, baby”		

Tale of the Tape—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

Football Fail!

- Orbital Branching and Isomorphism Pruning solve `codbt05` super fast
- The football pool problem is `codbt06`
- These methods (by themself) fail to make progress on the football pool problem

Key Idea!

- Enumerate "necessary conditions" for there to exist an optimal solution (code) of value/cardinality M
- If for each "necessary" condition, no such code of value M exists...
- The smallest code must be of cardinality at least $M+1$

Necessary Conditions by Subcode Enumeration

- Partition 729 outcomes (or tickets) W by the outcome of the first match
- $W = W_0 \cup W_1 \cup W_2$
- $w \in W_0$ covers 11 outcomes in W_0
- Ticket $w \in W_1$ covers 1 outcome in W_0
- Ticket $w \in W_2$ covers 1 outcome in W_0
- An optimal "code" C^* (solution to the problem) has
  - $C_0^* \subset W_0$, $|C_0^*| \stackrel{\text{def}}{=} y_0$
  - $C_1^* \subset W_1$, $|C_1^*| \stackrel{\text{def}}{=} y_1$
  - $C_2^* \subset W_2$, $|C_2^*| \stackrel{\text{def}}{=} y_2$

- So if a code of size $|C^*| = M$ exists, then it must satisfy

Covering System

$$\begin{aligned} 11y_0 + y_1 + y_2 &\geq 243 \\ y_0 + 11y_1 + y_2 &\geq 243 \\ y_0 + y_1 + 11y_2 &\geq 243 \\ y_0 + y_1 + y_2 &= M \end{aligned}$$

Sequence IP (M, y_0, y_1, y_2)

- Enumerate *all* (non-isomorphic) integer solutions (y_0, y_1, y_2) to the covering system
- Then solve...

$$
\begin{aligned}
& \min \mathbf{1}^\top x \\
\text{s.t.} \quad & Ax \geq \mathbf{1} \\
& \sum_{i \in W_0} x_i = y_0 \\
& \sum_{i \in W_1} x_i = y_1 \\
& \sum_{i \in W_2} x_i = y_2 \\
& \mathbf{1}^\top x \leq M \\
& x \in \{0, 1\}^{|W|}
\end{aligned}
$$

Improving the Lower Bound

- Solve for every (enumerated) sequence: (y_0, y_1, y_2).
- If you find no solution, then $M + 1$ is a valid lower bound

Results of Preprocessing/Enuemration

- In the end, after even more tricks, we are left the following number of difficult, symmetric, integer programs to solve:

M	#sequences	Modified #Sequences
65	0	0
66	797	7
67	1,723	13
68	3,640	45
69	7,527	102
70	13,600	176
71	24,023	264
72	40,431	393
		1000

- Solving $M = 66, 67, 68$ IPs takes less than a week on a single CPU with isomorphism pruning.
- Other instances are (quite) difficult $\Rightarrow$ we need a BIG computer

Is This Big Enough For You?

Site	Access Method	Arch/OS	Machines
Wisconsin - CS	Flocking	x86_32/Linux	975
Wisconsin - CS	Flocking	Windows	126
Wisconsin - CAE	Remote submit	x86_32/Linux	89
Wisconsin - CAE	Remote submit	Windows	936
Lehigh - COR@L Lab	Flocking	x86_32/Linux	57
Lehigh - Campus desktops	Remote Submit	Windows	803
Lehigh - Beowulf	ssh + Remote Submit	x86_32	184
Lehigh - Beowulf	ssh + Remote Submit	x86_64	120
OSG - Wisconsin	Schedd-on-side	x86_32/Linux	1000
OSG - Nebraska	Schedd-on-side	x86_32/Linux	200
OSG - Caltech	Schedd-on-side	x86_32/Linux	500
OSG - Arkansas	Schedd-on-side	x86_32/Linux	8
OSG - BNL	Schedd-on-side	x86_32/Linux	250
OSG - MIT	Schedd-on-side	x86_32/Linux	200
OSG - Purdue	Schedd-on-side	x86_32/Linux	500
OSG - Florida	Schedd-on-side	x86_32/Linux	100

Computational Grid, cont.

Site	Access Method	Arch/OS	Machines
TG - NCSA	Flocking	x86_32/Linux	494
TG - NCSA	Flocking	x86_64/Linux	406
TG - NCSA	Hobble-in	ia64-linux	1732
TG - ANL/UC	Hobble-in	ia-32/Linux	192
TG - ANL/UC	Hobble-in	ia-64/Linux	128
TG - TACC	Hobble-in	x86_64/Linux	5100
TG - SDSC	Hobble-in	ia-64/Linux	524
TG - Purdue	Remote Submit	x86_32/Linux	1099
TG - Purdue	Remote Submit	x86_64/Linux	1529
TG - Purdue	Remote Submit	Windows	1460
			19,012

Grid 2.0

Grid technology has realy "taken off," since now I need two slides to list all of my resources

Jealous?

- 19,012 processors sounds great, but sadly they aren't all mine.
- I can only use them when other, more important people aren't using them.
- This is the whole notion behind a concept called the computational grid

---

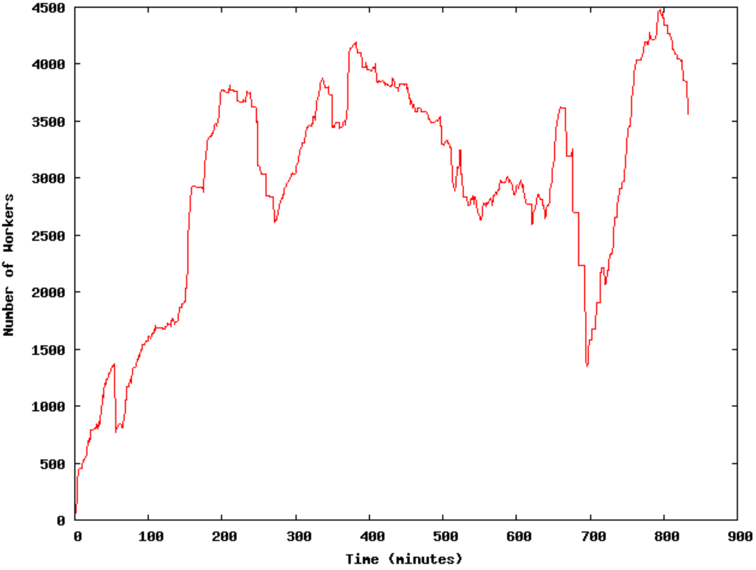
- Condor provides infrastructure for doing this type of computing.
- But still need to control the branch and bound algorithm.
- Computations must be flexible—fault tolerant and dynamic.
- Master-Worker: Isomorphism-Pruning enhanced branch and bound code was parallelized using software using MW.

Large Scale Computation

- We solved the (symmetric) IPs on this collection of machines over a period of a few months

	$M = 69$	$M = 70$
Avg. Workers	555.8	562.4
Max Workers	2038	1775
Worker Time (years)	110.1	30.3
Wall Time (days)	72.3	19.7
Worker Util.	90%	71%
Nodes	2.85×10^9	1.89×10^8
LP Pivots	2.65×10^{12}	1.82×10^{11}

Simultaneous Workers, $M = 71$ attempt



Why Did I Stop, You Ask?

Global Warming Is All My Fault

- 200 CPU Years = 1.752M CPU Hours. $\approx$ 500 W per CPU hour $\Rightarrow$ 876 MWH for the calculation.
- $\Rightarrow$ Roughly 1.1388 million pounds (569 tons) of CO_2 produced

Car Travel

- Prius produces around one ton of CO_2 for 5988 miles
- $\Rightarrow$ I could dive my Prius about 3.4 million miles.

"Don't worry about that."

—Arnold, on the environment

Conclusions

- Don't blame Jeff for the cold.
- Some Symmetric IPs are still very difficult
- One may branch on any variable and still do isomorphism pruning.
- Still more work to determine how to best exploit this flexibility

---

Thank you!

Any Questions?

Arnold Non Sequitur

"Milk is for babies. When you grow up you have to drink beer."

— Arnold Schwarzenegger, *Pumping Iron*

Publications

- J. Linderoth, F. Margot, and G. Thain, "Improving Bounds on the Football Pool Problem via Symmetry Reduction and High-Throughput Computing", *INFORMS Journal on Computing*, 21:445-457, 2009.
- J. Ostrowski, J. T. Linderoth, F. Rossi, and S. Smriglio, "Orbital Branching," *Mathematical Programming*, 126:147-178, 2011.
- J. Ostrowski, J. Linderoth, F. Rossi, and S. Smriglio, "Constraint Orbital Branching", *IPCO 2008: The Thirteenth Conference on Integer Programming and Combinatorial Optimization*, *Lecture Notes in Computer Science*, Vol. 5035, 225-239, 2008.
- J. Ostrowski, J. T. Linderoth, F. Rossi, and S. Smriglio, "Solving Large Steiner Triple Covering Problems," *Operations Research Letters*, 39:127-131, 2011.
- J. Ostrowski, J. Linderoth, F. Margot, "Flexible Isomorphism Pruning," Working paper.