Condor and the Football Pool Problem
Or, How Condor Can Make you Rich Beyond Your Wildest Dreams

Jeff Linderoth
COR@L Lab
Lehigh University

The Annual Condor Clambake
University of Wisconsin-Madison
April 25, 2006

Thanks: NSF (OCI-0330607) and DOE (DE-FG02-05ER25694) for continuing support of this work
Application — Code Design

- $W(\nu, \alpha)$: Set of all “words” of length $\nu$ from alphabet $\{0, 1, \ldots \alpha - 1\}$.
- $|W(\alpha, \nu)| = \alpha^\nu$
- We will abbreviate $W(\nu, \alpha) = W$
- A code is a subset $C \subseteq W$
- Hamming distance: $a \in W, b \in W$, $\text{dist}(a, b) = |\{i \mid a_i \neq b_i\}|$
Code Applications

**Error Correcting Code**
- Find $C \subseteq W$ such that $a \in C, b \in C \Rightarrow \text{dist}(a, b) \geq 2d + 1$
- Maximize $|C|$
- **Application**: Words in $C$ submit over a “noisy” channel on which at most $d$ letters are changed can be “self-corrected.”

**Covering Code**
- Find a code $C \subseteq W$ such that every word $w \in W$ is at most a distance $d$ away from at least one word in $C$
  \[ \text{dist}(w, C) \leq d \ \forall w \in W \]
- Minimize $|C|$
- **Application**: Something far more practical
Are You Ready for Some Football!

- Predict the outcome of $\nu$ soccer matches
- $\alpha = 3$
  - 0: Team A wins
  - 1: Team B wins
  - 2: Draw
- You win if you miss at most $d = 1$ games

**The Football Pool Problem**

What is the minimum number of tickets you must buy to assure yourself a win?
How Many Must I Buy?

<table>
<thead>
<tr>
<th>Known Optimal Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
</tr>
<tr>
<td>$</td>
</tr>
</tbody>
</table>

The Football Pool Problem

What is $|C_6^*|$?

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

  $$65 \leq C_6^* \leq 73$$
But It’s Trivial!

- For each $j \in W$, let $x_j = 1$ iff word $j$ is in code $C$
- Let $A \in \{0, 1\}^{|W| \times |W|}$
- $a_{ij} = 1$ iff word $i \in W$ is distance $\leq d = 1$ from word $j \in W$

**IP Formulation**

\[
\begin{align*}
\min & \quad e^T x \\
\text{s.t.} & \quad Ax \geq e \\
& \quad x \in \{0, 1\}^{|W|}
\end{align*}
\]
Solving IPs in a Nutshell

- Problem is in general $\mathcal{NP}$-Hard
- Loads of theory and techniques going back $>40$ years
- Workhorse algorithm is a tree-search procedure known as branch-and-bound.
- But really, branch-and-bound or its souped-up cousin branch-and-cut have been replaced in the most part by the new technique: give-it-to-CPLEX
- **CPLEX**: A commercial IP package that is putting integer programmers out of business.
### CPLEX Can Solve Every IP

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Objective</th>
<th>IInf</th>
<th>Best Integer</th>
<th>Cuts/</th>
<th>Best Node</th>
<th>ItCnt</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>56.0769</td>
<td>729</td>
<td></td>
<td>56.0769</td>
<td>2200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>0+</td>
<td>0</td>
<td>243.0000</td>
<td>56.0769</td>
<td>2200</td>
<td>76.92%</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>0+</td>
<td>0</td>
<td>110.0000</td>
<td>56.0769</td>
<td>2200</td>
<td>49.02%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>56.5164</td>
<td>729</td>
<td>110.0000</td>
<td>Fract: 56</td>
<td>2542</td>
<td>48.62%</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>0+</td>
<td>0</td>
<td>107.0000</td>
<td>56.5164</td>
<td>2542</td>
<td>47.18%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>56.5279</td>
<td>729</td>
<td>107.0000</td>
<td>Fract: 6</td>
<td>2673</td>
<td>47.17%</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>0+</td>
<td>0</td>
<td>94.0000</td>
<td>56.5279</td>
<td>2673</td>
<td>39.86%</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>0+</td>
<td>0</td>
<td>93.0000</td>
<td>56.5279</td>
<td>2673</td>
<td>39.22%</td>
<td></td>
</tr>
</tbody>
</table>

Elapsed time = 90.03 sec. (tree size = 0.00 MB)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Objective</th>
<th>IInf</th>
<th>Best Integer</th>
<th>Cuts/</th>
<th>Best Node</th>
<th>ItCnt</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>50+</td>
<td>0</td>
<td>91.0000</td>
<td>56.5285</td>
<td>12242</td>
<td>37.88%</td>
<td></td>
</tr>
</tbody>
</table>

Elapsed time = 6841.16 sec. (tree size = 14.12 MB)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Objective</th>
<th>IInf</th>
<th>Best Integer</th>
<th>Cuts/</th>
<th>Best Node</th>
<th>ItCnt</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>31100</td>
<td>30002</td>
<td>60.1690</td>
<td>544</td>
<td>87.0000</td>
<td>57.1864</td>
<td>5467339</td>
<td>34.27%</td>
</tr>
<tr>
<td>31200</td>
<td>30102</td>
<td>77.7888</td>
<td>216</td>
<td>87.0000</td>
<td>57.1864</td>
<td>5499451</td>
<td>34.27%</td>
</tr>
<tr>
<td>*</td>
<td>31200+28950</td>
<td>0</td>
<td>86.0000</td>
<td>57.1864</td>
<td>5499451</td>
<td>33.50%</td>
<td></td>
</tr>
<tr>
<td>31300</td>
<td>29044</td>
<td>58.9809</td>
<td>611</td>
<td>86.0000</td>
<td>57.1870</td>
<td>5511005</td>
<td>33.50%</td>
</tr>
</tbody>
</table>

Elapsed time = 9500.15 sec. (tree size = 18.70 MB)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Objective</th>
<th>IInf</th>
<th>Best Integer</th>
<th>Cuts/</th>
<th>Best Node</th>
<th>ItCnt</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>42700</td>
<td>39098</td>
<td>78.3242</td>
<td>197</td>
<td>85.0000</td>
<td>57.2845</td>
<td>7623200</td>
<td>32.61%</td>
</tr>
<tr>
<td>*</td>
<td>42740+36552</td>
<td>0</td>
<td>83.0000</td>
<td>57.2845</td>
<td>7626440</td>
<td>30.98%</td>
<td></td>
</tr>
</tbody>
</table>

Elapsed time = 117349.90 sec. (tree size = 202.88 MB)

Nodefile size = 74.98 MB (61.52 MB after compression)
Bounds

- Roughly $10^{30}$ seconds $\approx 10^8$ universe lifetimes in order to establish that $|C^*_6| \geq 73$
Jeff’s Research Advice for the Day

How to solve problems that other people can’t...

1. Be smarter than other people
   - Sadly, this strategy doesn’t work for me
2. Get a more powerful computer than other people
   - Thank you Condor!!

The Optimal Solution

1. Collaborate with (mooch off of) smart people
2. Get a more powerful computer
The Brains of the Operation

François Margot
Carnegie Mellon

Greg Thain
Master Worker
University of Wisconsin-Madison
A Conversation With Miron

“What do you think I should talk about at the upcoming Condor Week, Miron?”

“Don’t talk about your research, Jeff. Nobody cares about your research.”
Miron Was Wrong!

“I love you, son. I think you should talk about your research if that makes you happy.”

A Rock and a Hard Place
Should I listen to Miron or my Mom?
Solutions for $v = 3$

- These solutions are isomorphic.
- For first component: $2 \leftrightarrow 0$
- There are LOTS of isomorphic solutions:
  - "Rename" W,L,D for any subset of the matches: $(x_1)^v$
  - Reorder the matches: $v!$
- There are $(x_1)^v v! = 1296$ equivalent solutions for $v = 3$

What's the Problem!?: Symmetry

- $\pi$: Permutation of $\{1, 2, \ldots n\}$
- $\pi(x) = \pi(x_1, x_2, \ldots x_n) = [\pi(x_1), \pi(x_2), \ldots \pi(x_n)]$
- $\pi$ is a symmetry of an IP if
  - $\pi$ feasible $\Rightarrow \pi(x)$ feasible
  - $\pi^T x = \pi^T(m(x))$
- $G(IP)$: Set of all symmetries of IP
- For covering design, $|G(IP)| = \nu(v)^v$
- $6^3 \times 3 = 52680$

Minimum Index Branching

- Butler, Ivanov, Lam, Magot, McKay, Read, Stinson, ...
- $C$ is a minimum index branchings of $S \subseteq \{1, 2, \ldots, n\}$
- Choose one representative for each potential set of variable fixings.
- For example, in Minimum Index Branching, A set $S$ is a representative of its equivalence class if $S = \text{lexmin}(g(S)) \mid g \in G(IP)$
- Isomorphism Pruning:
  - If $F_a \neq \emptyset$, then prune node $a$.

Results (All Thanks to François)

The Good

- For $d = 1, v = 5, \alpha = 3$, Isomorphism Pruning can establish $C_{\alpha} \geq 27$ in 1409 nodes, 82 seconds.
- CPLEX (v9.1) does not solve the problem in more than 4 hours.

The Bad and Ugly

- For $d = 1, v = 6, \alpha = 3$, Isomorphism Pruning gets nowhere...
- $C_2 \geq 61$ after long runtime.

Pick an $M = C^*$

- Enumerate all non-isomorphic solutions to covering system for $m = 1, 2, \ldots 6$
- If for some $m$, there are no solutions, then $M + 1$ is a valid lower bound on $C^*$
- Östergård and Weakley (2000): Able to show $M = 62$ is optimal code length for $d = 1, v = 9, \alpha = 2$
- Östergård and Wassermann (2002): Able to show $M \geq 65$ for $d = 1, v = 6, \alpha = 3$. Required over 1 CPU year!

Extending the Idea

A New Idea!

- $M$, and optimal code size $M$, enumerate all non-isomorphic solutions to the covering system
- This gives a list of possible $y$ values, e.g. for $m = 1$ you get a list of tuples $(y_0, y_1, y_2)$
- For each member of the list, solve the "Sequence IP"

Sequence IP $(M, y_0, y_1, y_2)$

\[
\min e^T x \\
\text{s.t. } Ax \geq e \\
\sum_{i \in W_0} x_i = y_0 \\
\sum_{i \in W_1} x_i = y_1 \\
\sum_{i \in W_2} x_i = y_2 \\
e^T x \leq M \\
x \in \{0, 1\}^M
\]

Extending the Idea:

- $\pi$: Permutation of $\{1, 2, \ldots n\}$
- $\pi(x) = \pi(x_1, x_2, \ldots x_n) = [\pi(x_1), \pi(x_2), \ldots \pi(x_n)]$
- $\pi$ is a symmetry of an IP if
  - $\pi$ feasible $\Rightarrow \pi(x)$ feasible
  - $\pi^T x = \pi^T(m(x))$
- $G(IP)$: Set of all symmetries of IP
- For covering design, $|G(IP)| = \nu(v)^v$
- $6^3 \times 3 = 52680$

Isomorphism Pruning

- For some permutation $\pi \in G(IP)$ and set of indices $S \subseteq \{1, 2, \ldots, n\}$, let
  - $g(S) = (\pi(i) \mid i \in S)$
- At a node $a$ of the branch-and-bound tree
  - $F^T_a = \{i \mid x_i = 1$ at $a\}$
  - $F^T_b = \{i \mid x_i = 0$ at $b\}$
- Nodes $a$ and $b$ are isomorphic if
  - $\exists g \in G(IP)$ with $g(F^T_a) = F^T_b, g(F^T_b) = F^T_a$
- You may prune one of $a$ or $b$. [Bazaraa, Kirca 83]
Great Research

The End Result

We need to perform many smaller branch-and-bound calculations

Grid Programmers Do It In Parallel

- Nodes in disjoint subtrees could be evaluated independently
- But this is not an embarrassingly pleasantly parallel operation
Master-Worker!

Important Points

- Master-Worker is a flexible, powerful framework for Grid Computing
- The MW software library can help you quickly write master-worker applications that run over Condor-provided computational grids
- Please come talk to Greg, Steve Wright, or myself about setting you up to run with MW
Mechanism #1—Flocking

Building My Grid

- Currently flock to UW-Madison CS Pool
- Large Condor Pool at NCSA
- Small pool in our lab at Lehigh: COR@L
Mechanism #2: Hobble-In

1. Log in to Teragrid site
2. Install condor_startd and condor_starter
3. Configure condor to report to your condor master
4. Write PBS or PSF scripts that run “preconfigured” condor startd’s for limited about of time.
5. Submit away. Machines will appear in your personal pool when local scheduler deigns to run them.
6. Currently working at the following Teragrid sites:
   - Argonne/UC, NCSA, SDSC, TACC

More Questions?
I’m happy to send anyone my hobble in scripts...
Building Your Grid: Flocking + Hobble-In

Jeff’s Personal Condor

Flocking

Hobble-In
Mechanism #3: sshIdle-In

- 300 Machines at Lehigh are on running Condor on a private network: And I wanted to use them!

**sshIdle-In**

- A GCB-free mechanism for running on private networks
- For use currently with MWSocket RMComm only

1. Look up port number where master is listening: default 8997.
2. Log into “landing point” – machine with connections to both private network and outside world – preferably condor_master
3. Transfer (or build) worker executables on landing point
4. Forward traffic:

   ```bash
   ssh -l fmargot -g -N -f -L 8997:0.0.0.0:8997 meisterbrau.cs.wisc.edu
   ```

5. Submit worker executables in private network
Sample sshIdle-In Script

Universe = Vanilla
Executable = mw_exec0.$(Opsys).$(Arch).exe
arguments = 1512 8997 8997 192.168.1.1
should_transfer_files = Yes
when_to_transfer_output = ON_EXIT
rank = Mips
on_exit_remove = false
nice_user = true
queue
arguments = 1513 8997 8997 192.168.1.1
queue
arguments = 1514 8997 8997 192.168.1.1
queue
arguments = 1515 8997 8997 192.168.1.1
queue
Flocking + Hobble-In + sshIdle-In

Jeff’s Personal Condor

Hobble-In

sshIdle-In

Flocking
Other Mechanisms You Should Probably Use

- **Condor-Glidein**: That’s what it was designed for!
- **Gridshell**: GridShell/Condor creates virtual Condor pools across TeraGrid clusters connected through Globus.
  - http://gridshell.net/
  - Talk to Ed Walker

---

It Begs The Question

- Before you ask, I **don’t** have a good excuse for using hobble-in other than it works for me.
  - I don’t need to rely on local admins to configure gateway properly
  - Scripts for various Teragrid sites (to get condor startd’s talking to condor master) were already configured/set-up before I knew about Gridshell.
Mechanism #3: IceCream-In

Stardate: Date: Thu, 20 Apr 2006 16:22:28 -0500

A Conversation Over Ice Cream...

“I really like all these mechanisms, Miron, but where and how can I get some more workers?”

“There’s this thing called the Open Science Grid. I’ll see what I can do…”
Jeff:
The Open Science Grid has asked us for some more jobs to demonstrate some of our Grid technologies, so I’m working on getting some of the football pool jobs routed over there. **Hang on for some Super Grid Power!** I need to reboot the personal condor first, so don’t worry if the ff3 output looks a little wonky.

-greg
Workers in Solving One Batch of Subproblems

This is the closest thing I’ve seen to a true “Grid” reality: I’m eating an ice cream cone, and I get 300 new workers.
Jeff’s Super-Duper Computational Grid

- Jeff’s Personal Condor
- Flocking
- sshIdle-In
- Hobble-In
- IceCream-In
To establish that $|C_6^*| \geq 70$ Final total: 346 sequence IPs

François and Jeff implemented these ideas using the old MW-FATCOP framework for MILP.

Many of the engineering/tuning ideas from the QAP experience were used in this implementation

Greg has made numerous improvements in MW’s robustness and has been great at scavenging cycles in an effort to solve this problem
Breaking News

Deal colleagues, we are pleased to announce that we have improved the lower bound on the cardinality of a covering code of radius one for the Hamming Space $\mathbb{Q} = F_3^6$ to 70.
Or Have We!?!?!?

Stardate: Tue, 11 Apr 2006 12:41:29 -0400 (EDT)

From: Francois Margot <fmargot@andrew.cmu.edu>
To: Jeff Linderoth <jtl3@Lehigh.EDU>
cc: gthain@cs.wisc.edu
Subject: Re: BooYah!

Now, remove all sharp objects from your desk. I noticed a mistake in the codbt06_fix2_reg01236_s.mylp.gz file (a permutation of the inequalities). I fixed the file, but I am afraid that we will have to rerun the *69s_* .seq problems. However, I can reduced the number of sequences in the files, taking into account the fact that every 5-subcode must have at least 22 words. I will let you steam a little bit before doing so ...
Doh!

```
jeff@pbr:  diff correct.lp incorrect.lp
1c1
< 738 729 1
---
> 739 729 1
```

Condor Allows You To Make Mistakes

- We (read François) wasted “only” 19.76 CPU Years.
- If we had to rely only on machines for which I had to write grants (i.e. Teragrid), we never would have been able to make progress
We’re Still Working

- I really, really, really wanted to announce that $C_6^* \geq 70$ today.
- But it hasn’t been for a lack of trying

---

**Statistics so far...**

- Wall Time: 47.15 days
- CPU Time: 57.31 years
- Avg Workers: 467.3
- Max Workers: 1253
- Total Nodes: $8.37 \times 10^8$
- Total LP Pivots: $6.53 \times 10^{11}$
- Parallel Performance: 94.9%
Conclusions

- There’s lots of CPU power out there for the taking.
- There are lots of mechanisms for you to build your Grid via Condor—Flocking, Glide-in, GridShell, Hobble-In, sshIdle-In, IceCream-In (or schedd on the side).
- Many computations can be put in a master-worker framework: Maybe yours can too!
- I’d be happy to help you in using MW.
The MWAgenda

The MW Present

- MW works on WinDoze (cygwin socket)
- MW has “Black Box” capabilities, so you need not write any code other than code for the MWDriver.

The MW Future

- Making ssh-idle in more automatic
- Easier Interfaces?:
- MWBOINC?

We wants you to help us steer the MWShip
Let’s Talk

- I’m lonely—Please come talk to me about MW or my research.
- MW: http://www.cs.wisc.edu/condor/mw
- COR@L: http://coral.ie.lehigh.edu
- mailto:jtl3@lehigh.edu

Thanks (again) to NSF (OCI-0330607) and DOE (DE-FG02-05ER25694) for continuing support of this work