Computational Mixed Integer Nonlinear Programming

Jeff “Obi-Wan” Linderoth
Dept. of Industrial and Systems Engineering
Univ. of Wisconsin-Madison
linderoth@wisc.edu
Our Quest

Mixed Integer Nonlinear Program: (MINLP)

\[ z_{\text{MINLP}} = \min \quad f(x) \]
\[ \text{subject to} \quad g_j(x) \leq 0, \quad j = 1, \ldots, m \]
\[ x \in X, \quad x_I \in \mathbb{Z}^{|I|} \]

- \( X \overset{\text{def}}{=} \{ x \mid x \in \mathbb{R}^n_+, Dx \leq d \} \)
- \( f, g_j \) are continuously differentiable (sometimes) convex functions.
  - \( f, g_j \) linear \( \Rightarrow \) MILP
Our Quest

**Mixed Integer Nonlinear Program: (MINLP)**

\[
\begin{align*}
    z_{\text{MINLP}} &= \min \eta \\
    \text{subject to } g_j(x, z) &\leq 0, \quad j = 1, \ldots, m \\
    f(x) &\leq \eta \\
    x &\in X, \quad x_I \in \mathbb{Z}^{|I|}
\end{align*}
\]

- \( X \overset{\text{def}}{=} \{ x \mid x \in \mathbb{R}^n_+, Dx \leq d \} \)
- \( f, g_j \) are continuously differentiable (sometimes) **convex** functions.
  - \( f, g_j \) **linear** \( \Rightarrow \) MILP
Apology accepted

- The talk is a bit of a shotgunstar-blaster approach to MINLP.
- Not many technical details.
- But I am happy to discuss in more detail
The Rebel Alliance

This was been joint work with many talented individuals. They will be introduced as the talk proceeds.
Application: Death Star Core Reload

- Maximize reactor efficiency after reload subject to diffusion PDE
- diffusion PDE ≈ nonlinear equation
  ⇒ integer & nonlinear model
- avoid reactor becoming sub-critical
Application: Death Star Core Reload

- Maximize reactor efficiency after reload subject to diffusion PDE
- diffusion PDE \(\simeq\) nonlinear equation
  \(\Rightarrow\) integer & nonlinear model
- avoid reactor becoming overheated
### Other MINLPs

<table>
<thead>
<tr>
<th>Application</th>
<th>Discrete</th>
<th>Nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water network design</td>
<td>Pipe connections/sizes</td>
<td>Pressure Loss</td>
</tr>
</tbody>
</table>

---

Jeff Linderoth (UW-Madison)
Other MINLPs

<table>
<thead>
<tr>
<th>Application</th>
<th>Discrete</th>
<th>Nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water network design</td>
<td>Pipe connections/sizes</td>
<td>Pressure Loss</td>
</tr>
<tr>
<td>Petrochemical</td>
<td>Which process?</td>
<td>Product Blending</td>
</tr>
</tbody>
</table>
## Other MINLPs

<table>
<thead>
<tr>
<th>Application</th>
<th>Discrete</th>
<th>Nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water network design</td>
<td>Pipe connections/sizes</td>
<td>Pressure Loss</td>
</tr>
<tr>
<td>Petrochemical</td>
<td>Which process?</td>
<td>Product Blending</td>
</tr>
<tr>
<td>Oilfield Planning</td>
<td>Fields to develop?</td>
<td>Reservoir Flow</td>
</tr>
</tbody>
</table>
# Other MINLPs

<table>
<thead>
<tr>
<th>Application</th>
<th>Discrete</th>
<th>Nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water network design</td>
<td>Pipe connections/sizes</td>
<td>Pressure Loss</td>
</tr>
<tr>
<td>Petrochemical</td>
<td>Which process?</td>
<td>Product Blending</td>
</tr>
<tr>
<td>Oilfield Planning</td>
<td>Fields to develop?</td>
<td>Reservoir Flow</td>
</tr>
<tr>
<td>Portfolio Planning</td>
<td>Trading Strategy</td>
<td>Utility</td>
</tr>
</tbody>
</table>
## Other MINLPs

<table>
<thead>
<tr>
<th>Application</th>
<th>Discrete</th>
<th>Nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water network design</td>
<td>Pipe connections/sizes</td>
<td>Pressure Loss</td>
</tr>
<tr>
<td>Petrochemical</td>
<td>Which process?</td>
<td>Product Blending</td>
</tr>
<tr>
<td>Oilfield Planning</td>
<td>Fields to develop?</td>
<td>Reservoir Flow</td>
</tr>
<tr>
<td>Portfolio Planning</td>
<td>Trading Strategy</td>
<td>Utility</td>
</tr>
<tr>
<td>Sparse Approximation</td>
<td>Selection of elements</td>
<td>(2-norm) model error</td>
</tr>
</tbody>
</table>
**Other MINLPs**

<table>
<thead>
<tr>
<th>Application</th>
<th>Discrete</th>
<th>Nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water network design</td>
<td>Pipe connections/sizes</td>
<td>Pressure Loss</td>
</tr>
<tr>
<td>Petrochemical</td>
<td>Which process?</td>
<td>Product Blending</td>
</tr>
<tr>
<td>Oilfield Planning</td>
<td>Fields to develop?</td>
<td>Reservoir Flow</td>
</tr>
<tr>
<td>Portfolio Planning</td>
<td>Trading Strategy</td>
<td>Utility</td>
</tr>
<tr>
<td>Sparse Approximation</td>
<td>Selection of elements</td>
<td>(2-norm) model error</td>
</tr>
</tbody>
</table>

**How to Solve MINLPs?**
Relax!

because sometimes a galactic war has to take a break.
A “Natural” Relaxations

- Relax integrality restriction
- Instead of searching nonconvex set of feasible solutions

\[
\min_{x, z \in \mathbb{R}_+ \times \mathbb{Z}_+} \{ -x - z \mid x^2 + z^2 \leq 4 \}
\]

\[
z_{\text{MINLP}} = \min f(x) \quad \text{subject to} \quad g_j(x) \leq 0, \quad j = 1, \ldots, m
\]

\[
x \in X, \quad x_I \in \mathbb{Z}^{[I]}
\]
A “Natural” Relaxations

- Relax integrality restriction
- Instead of searching nonconvex set of feasible solutions
- Develop convex relaxation of set

\[
\begin{align*}
\min_{x,z \in \mathbb{R}_+ \times \mathbb{R}_+} & \quad \{ -x - z \mid x^2 + z^2 \leq 4 \} \\
\end{align*}
\]

\[
\begin{align*}
z_{\text{MINLP}} &= \min f(x) \\
\text{subject to} & \quad g_j(x) \leq 0, \quad j = 1, \ldots, m \\
& \quad x \in X, \quad x_1 \in \mathbb{R}^{|I|}
\end{align*}
\]
A “Natural” Relaxations

- Relax integrality restriction
- Instead of searching nonconvex set of feasible solutions
- Develop convex relaxation of set

\[
\begin{align*}
\min_{x, z \in \mathbb{R}^n \times \mathbb{R}^m} & \quad \{ -x - z \mid x^2 + z^2 \leq 4 \} \\
\end{align*}
\]

\[
\begin{align*}
z_{\text{MINLP}} &= \min f(x) \\
\text{subject to} \quad & g_j(x) \leq 0, \quad j = 1, \ldots, m \\
& x \in \mathcal{X}, \quad x_1 \in \mathcal{X}_1 
\end{align*}
\]
A “Natural” Relaxations

- Relax integrality restriction
- Instead of searching nonconvex set of feasible solutions
- Develop convex relaxation of set

\[
\begin{align*}
\min_{x,z \in \mathbb{R}_+} & \quad \{ -x - z \mid x^2 + z^2 \leq 4 \} \\
\text{subject to} & \quad g_j(x) \leq 0, \quad j = 1, \ldots, m \\
x \in X, & \quad x_1 \in \mathbb{R}_+ 
\end{align*}
\]
Branch-and-Bound

Solve relaxed NLP \((0 \leq x_i \leq 1 \text{ continuous relaxation})\)

...solution value provides lower bound

- Branch on \(x_i, i \in I\) non-integral
- Solve NLPs & branch until
  1. Node infeasible ...
  2. Node integer feasible ...
     \(\Rightarrow\) get upper bound \((U)\)
  3. Lower bound \(\geq U\) ...

Search until no unexplored nodes on tree
Why I Don’t Fit In At Wisconsin

Jeff
Why I Don’t Fit In At Wisconsin

Jeff Hates

No Love
Why I Don’t Fit In At Wisconsin

Jeff Hates NLP

No Love
Why I Don’t Fit In At Wisconsin

Jeff Hates NLP

- So I sought a different path...
A Long Time Ago, At An Argonne Far, Far Away

“Oh wise Yoda Leyffer, how can one solve (convex) MINLPs?”
A Long Time Ago, At An Argonne Far, Far Away

“Oh wise Yoda Leyffer, how can one solve (convex) MINLPs?”

“I suggest the LP/NLP Algorithm by Quesada and Grossmann”

- Of the four algorithms I implemented for my thesis, best this was
- Good implementation exists it does not
Talk Theme

USE THE MILP
Talk Theme

The MILP Force

- How can we leverage MILP Technology to Build an Effective MINLP Solver
Talk Theme

USE THE MILP

The MILP Force

- How can we leverage MILP Technology to Build an Effective MINLP Solver

Use the MILP in Four Ways

1. Algorithm Engineering
2. (Disjunctive) Cutting Planes
3. Preprocessing
4. Extended Formulations
Fixed NLP Subproblem

- The QG algorithm still has to solve NLPs.
- But only those where the integer variables $x_I$ are fixed.
Fixed NLP Subproblem

- The QG algorithm still has to solve NLPs.
- But only those where the integer variables $x_I$ are fixed

\[ z_{NLP}(x^k_I) = \min f(x) \]
subject to \[ g_j(x) \leq 0 \quad \forall j \]
\[ x_I = x^k_I \]
\[ x \in X \]

- NLP($x^k$) feasible $\Rightarrow$ Upper Bound.
- Linearize $f$ and $g_j$ about $x^k$:

\[
\begin{align*}
& f(x^k) + \nabla f(x^k)^T(x - x^k) \leq \eta \\
& g_j(x^k) + \nabla g_j(x^k)^T(x - x^k) \leq 0
\end{align*}
\]
Fixed NLP Subproblem

- The QG algorithm still has to solve NLPs.
- But only those where the integer variables $x_I$ are fixed

\[
\begin{align*}
NLP(x^k) \\
\end{align*}
\]

- $NLP(x^k)$ feasible $\Rightarrow$ Upper Bound.
- Linearize $f$ and $g_j$ about $x^k$:

\[
\begin{align*}
\left\{ \begin{array}{l}
f(x^k) + \nabla f(x^k)^T(x - x^k) & \leq \eta \\
g_j(x^k) + \nabla g_j(x^k)^T(x - x^k) & \leq 0
\end{array} \right.
\end{align*}
\]

- By convexity, inequalities underapproximate objective function and outer-approximate feasible region
- Collect linearization points into a set $K$ and create a polyhedral relaxation of the problem
The Master

**MP(\(\mathcal{K}\))**: Outer-Approximation MILP Master Problem

\[
\begin{align*}
z_{\text{MP}(\mathcal{K})} &= \min \quad \eta \\
\text{subject to} \quad f(x^k) + \nabla f(x^k)^T(x - x^k) &\leq \eta \quad \forall (x^k) \in \mathcal{K} \quad (\text{MP}(\mathcal{K})) \\
g_j(x^k) + \nabla g_j(x^k)^T(x - x^k) &\leq 0 \quad \forall (x^k) \in \mathcal{K} \quad \forall j \\
x \in X, \quad x_I \in \mathbb{Z}^{|I|}
\end{align*}
\]
The Master

MP(\( \mathcal{K} \)): Outer-Approximation MILP Master Problem

\[
\begin{align*}
    z_{\text{MP}(\mathcal{K})} &= \min \eta \\
    \text{subject to} \quad f(x^k) + \nabla f(x^k)^T(x - x^k) &\leq \eta \quad \forall (x^k) \in \mathcal{K} \quad (\text{MP}(\mathcal{K})) \\
    g_j(x^k) + \nabla g_j(x^k)^T(x - x^k) &\leq 0 \quad \forall (x^k) \in \mathcal{K} \quad \forall j \\
    x &\in X, \quad x_I \in \mathbb{Z}^{\vert I \vert}
\end{align*}
\]

- If \( \mathcal{K} \) contains the “right” points, then \( z_{\text{MINLP}} = z_{\text{MP}(\mathcal{K})} \)
LP/NLP-BB (Quesada-Grossmann)

- Start solving Master MILP ($MP(\mathcal{K})$) ... using branch-and-cut.
LP/NLP-BB (Quesada-Grossmann)

- Start solving Master MILP ($\text{MP}(\mathcal{K})$) ... using branch-and-cut.
- If $x^k_I \in \mathbb{Z}^{|I|}_+$, then interrupt MILP. Solve NLP($x^k_I$) get $x^k$
LP/NLP-BB (Quesada-Grossmann)

- Start solving Master MILP ($\text{MP} (\mathcal{K})$) ... using branch-and-cut.
- If $x^k_I \in \mathbb{Z}_+$, then interrupt MILP. Solve NLP($x^k_I$) get $x^k$
- Linearize f, $g_j$ about ($x^k_I$) $\Rightarrow$ add linearization to tree
LP/NLP-BB (Quesada-Grossmann)

- Start solving Master MILP ($\text{MP}(\mathcal{K})$) ... using branch-and-cut.
- If $x^k_I \in \mathbb{Z}_+^{|I|}$, then interrupt MILP. Solve NLP($x^k_I$) get $x^k$
- linearize $f$, $g_j$ about $(x^k)$
  ⇒ add linearization to tree
- continue MILP tree-search

... until entire tree is fathomed
LP/NLP-BB = Branch and Cut for MINLP

- This really is just a branch-and-cut method for solving MINLP
- **One slight difference:** At integer feasible points, we must solve an NLP and also branch
- Branch-and-cut frameworks (like MINTO) have this functionality.
- We need an NLP solver:
  - **Filter-SQP.** Sven’s award-winning, filter-sequential quadratic programming (active set) code.
LP/NLP-BB = Branch and Cut for MINLP

- This really is just a branch-and-cut method for solving MINLP
- **One slight difference:** At integer feasible points, we must solve an NLP and also branch
- Branch-and-cut frameworks (like MINTO) have this functionality.
- We need an NLP solver:
  - **Filter-SQP.** Sven’s award-winning, filter-sequential quadratic programming (active set) code.

---

**Rebels are Bad at Acronyms**

FilMINT = **Filter** + **MINTO**
LP/NLP-BB = Branch and Cut for MINLP

This really is just a branch-and-cut method for solving MINLP

One slight difference: At integer feasible points, we must solve an NLP and also branch

Branch-and-cut frameworks (like MINTO) have this functionality.

We need an NLP solver:

Filter-SQP. Sven’s award-winning, filter-sequential quadratic programming (active set) code.

Rebels are Bad at Acronyms

FilMINT = Filter + MINTO

Why FilMINT Could Be Good

1. Use MINTO’s advanced MIP Features “for free.”
2. Really the only LP/NLP-BB algorithm available
Or So We Thought...

“That’s no moon. It’s a space station.”
Or So We Thought...

“That’s no moon. It’s a space station.”

The Galactic Empire

Battling the Empire

- This enormously talented team built open-source Bonmin, which (among other things), has an LP/NLP-BB implementation.
Battling the Empire

- This enormously talented team built open-source Bonmin, which (among other things), has an LP/NLP-BB implementation

Luke Abhishekk

“But I want to be a Jedi Master”
(read: get my thesis)
Battling the Empire

- This enormously talented team built open-source Bonmin, which (among other things), has an LP/NLP-BB implementation.

Luke Abhisheek

“But I want to be a Jedi Master” (read: get my thesis)

- So Rebel Forces pressed on...
MINLP Instances

- Multi-product batch plant design problems (Batch)
- Layout design problems (CLay, FLay, SLay, safetyLay, fo-m-o)
- Synthesis design problems (Syn)
- Retrofit planning (RSyn)
- Stochastic service system design problems (sssd)
- Cutting stock problems (trimloss)
- Quadratic uncapacitated facility location problems (uflquad)
- Network design problems (nd)
## Instance Families

<table>
<thead>
<tr>
<th>Instance Family</th>
<th>NL Ob?</th>
<th># of ins</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Var</td>
</tr>
<tr>
<td>Batch</td>
<td>√</td>
<td>10</td>
<td>334.6</td>
</tr>
<tr>
<td>CLay</td>
<td></td>
<td>12</td>
<td>116.7</td>
</tr>
<tr>
<td>FLay</td>
<td></td>
<td>10</td>
<td>158.0</td>
</tr>
<tr>
<td>fo-m-o</td>
<td></td>
<td>9</td>
<td>112.2</td>
</tr>
<tr>
<td>nd</td>
<td></td>
<td>5</td>
<td>574.0</td>
</tr>
<tr>
<td>RSyn</td>
<td></td>
<td>48</td>
<td>922.3</td>
</tr>
<tr>
<td>safetyLay</td>
<td></td>
<td>3</td>
<td>120.7</td>
</tr>
<tr>
<td>SLay</td>
<td>√</td>
<td>14</td>
<td>336.0</td>
</tr>
<tr>
<td>sssd</td>
<td></td>
<td>14</td>
<td>162.4</td>
</tr>
<tr>
<td>Syn</td>
<td></td>
<td>48</td>
<td>366.3</td>
</tr>
<tr>
<td>trimloss</td>
<td></td>
<td>12</td>
<td>279.2</td>
</tr>
<tr>
<td>uflquad</td>
<td>√</td>
<td>10</td>
<td>1571.0</td>
</tr>
<tr>
<td>others</td>
<td>√</td>
<td>12</td>
<td>205.4</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>207</td>
<td>487.5</td>
</tr>
</tbody>
</table>
Computational Experiments

- Convex MINLPs from variety of sources: GAMS MINLP World, MacMINLP, IBM-CMU Team
  - ≈ 50% easy: Solved by B&B solver in < 1 min. (Ignored)
  - 37 moderate: Solved by B&B solver in < 1 hour
  - 85 hard: Took > 1 hour
Computational Experiments

- Convex MINLPs from variety of sources: GAMS MINLP World, MacMINLP, IBM-CMU Team
  - $\approx 50\%$ easy: Solved by B&B solver in $< 1$ min. (Ignored)
  - 37 moderate: Solved by B&B solver in $< 1$ hour
  - 85 hard: Took $> 1$ hour

Performance Profile

- An empirical CDF of relative solver performance
- The “probability” that a solver will be at most $x$ times worse (slower) than the best solver for an instance
- “High” lines denote more effective solvers
MINTO v3.1 MILP Features

- Preprocessing
- Cutting planes
  - Knapsack covers, flow covers, clique inequalities, implication cuts
- Primal heuristic: Diving-based
- Branching
  - Pseudo-cost based branching.
- Node selection strategies
  - Adaptive (Depth first + best estimate).
MINITO v3.1 MILP Features

- Preprocessing
- Cutting planes
  - Knapsack covers, flow covers, clique inequalities, implication cuts
- Primal heuristic: Diving-based
- Branching
  - Pseudo-cost based branching.
- Node selection strategies
  - Adaptive (Depth first + best estimate).

The MILP Force

- Do fancy-pants MILP techniques make a difference for the LP/NLP-BB (QG) Algorithm?
YES! Performance Profile: Moderate Instances
A Simple Idea

- Since \( f, g_j \) are convex, we can linearize the functions about any point (not just \( x_I \in \mathbb{Z}_I^+ \)) and still obtain valid outer and under approximators.
A Simple Idea

- Since $f$, $g_j$ are convex, we can linearize the functions about any point (not just $x_I \in \mathbb{Z}_+^I$) and still obtain valid outer and under approximators.

Three Simple Implementations

1. **ECP Cut**: Generate linearizations about current solution of master problem.
   - Extended Cutting Plane (Kelley’s cutting plane) method

2. **FF Cut**: Fix integer variables $x_I = x_I^k$, solve NLP($x_I^k$), add linearizations at that solution point
   - “FixFrac”, some integer variables may be fixed at fractional values

3. **NLPR Cuts**: Solve (nonlinear) continuous relaxation, add linearizations at that point
   - Bound obtained in master problem after adding these cuts is same as you would get by solving nonlinear relaxation
Many Cuts

- We have a mechanism for generating many linearizations ("cuts")
Many Cuts

- We have a mechanism for generating many linearizations ("cuts")

**USE THE MILP, LUKE**

**MILP Cutting Plane Management**

- Use "known" ideas from MILP to manage these linearizations
- FilMINT scheme is simple—based on 3 parameters
Effect of Linearization Generation for Moderate Instances

% Instances vs. Time Increase Factor for different linearization methods:
- None
- ECP
- FF
- NLPR
- FilMint

Jeff Linderoth (UW-Madison)
Comparison of Solvers

![Graph showing comparison of solvers including QG, filmint, BONMIN, BONMIN-v2, and MINLP-BB across different time increase factors. The y-axis represents the percentage of instances, while the x-axis shows the time increase factor ranging from 1 to 1000.]
Question: What is an effective way for rebel force to compete against the empire?
**Question**

What is an effective way for rebel force to compete against the empire?
Answer: Steal Their Plans!

"Commander, tear this ship apart until you’ve found those plans. And bring me the passengers, I want them alive!"

Similar Ideas Appear in ...

- Bonami, Lift-and-Project Cuts for Mixed Integer Convex Programs. IPCO 2011
Disjunctive Cuts

Answer: Steal Their Plans!

“Commander, tear this ship apart until you’ve found those plans. And bring me the passengers, I want them alive!”

Similar Ideas Appear in ...
- Bonami, Lift-and-Project Cuts for Mixed Integer Convex Programs. IPCO 2011

Different Because
- We solve only linear programs to generate inequalities
Disjunctive Cuts

- Disjunction on $x_i (i \in I)$
  
  $$R_i^{k-} = \{ x \in \mathbb{R} \mid x_i \leq k \} \quad \text{and} \quad R_i^{k+} = \{ x \in \mathbb{R} \mid x_i \geq k + 1 \}$$

- Convex hull of union
  
  $$R_i^k = \text{conv} (R_i^{k-} \cup R_i^{k+})$$

**Goal:** Find valid inequalities for $R_i^k$
Disjunctive Cuts

- Disjunction on $x_i (i \in I)$
  
  \[ R_i^{k-} = \{ x \in \mathbb{R} \mid x_i \leq k \} \quad \text{and} \quad R_i^{k+} = \{ x \in \mathbb{R} \mid x_i \geq k + 1 \} \]

- Convex hull of union
  
  \[ \mathcal{R}_i^k = \text{conv} (R_i^{k-} \cup R_i^{k+}) \]

**Goal:** Find valid inequalities for $\mathcal{R}_i^k$

---

**Convex Hull (Lifted Space)**

\[ \mathcal{M}_i^k = \left\{ (x, y, z, \lambda, \mu) \mid \begin{aligned} x &= \lambda y + \mu z, \\ \lambda + \mu &= 1, \quad \lambda \geq 0, \quad \mu \geq 0 \\ y &\in R_i^{k-}, \quad z \in R_i^{k+} \end{aligned} \right\} \]
Separating from \( \text{conv} \left( R_i^{k-} \cup R_i^{k+} \right) \) [Stubbs and Mehrotra]

- Let \( \bar{x} \not\in \text{conv} \left( R_i^{k-} \cup R_i^{k+} \right) \)
Separating from $\text{conv} \left( R_{i}^{k_{-}} \cup R_{i}^{k_{+}} \right)$ [Stubbs and Mehrotra]

- Let $\bar{x} \not\in \text{conv} \left( R_{i}^{k_{-}} \cup R_{i}^{k_{+}} \right)$

**Separation Problem**

$$
\min_{(x,y,z,\lambda,\mu) \in M_{i}^{k_{i}}(R)} \quad d(x) \equiv \|x - \bar{x}\|
$$
Disjunctive Cuts

Separating from \( \text{conv} (R^k_i^- \cup R^k_i^+) \) [Stubbs and Mehrotra]

- Let \( \bar{x} \not\in \text{conv} (R^k_i^- \cup R^k_i^+) \)

Separation Problem

\[
\min_{(x,y,z,\lambda,\mu) \in \mathcal{M}_i^k(R)} d(x) \equiv \|x - \bar{x}\|
\]

- Let \( \hat{x} \) be an optimal solution of the separation problem
- Let \( \xi \in \partial d(\hat{x}) \)
- \( \xi^T(x - \hat{x}) \geq 0 \) separates \( \bar{x} \) from \( R^k_i \)
- **Bad News:** Nonlinear optimization problem!
Separating from \( \text{conv} \left( R_i^{k-} \cup R_i^{k+} \right) \) [Stubbs and Mehrotra]

- Let \( \bar{x} \not\in \text{conv} \left( R_i^{k-} \cup R_i^{k+} \right) \)

**Separation Problem**

\[
\min_{(x,y,z,\lambda,\mu) \in M_i^k(R)} d(x) \equiv \|x - \bar{x}\|
\]

- Let \( \hat{x} \) be an optimal solution of the separation problem
- Let \( \xi \in \partial d(\hat{x}) \)
- \( \xi^T(x - \hat{x}) \geq 0 \) separates \( \bar{x} \) from \( R_i^k \)
- **Bad News:** Nonlinear optimization problem!
- **Worse News:** Non-convex optimization problem, but we can convexify
A Story

“Your thesis is great, Rob. How come you didn’t try to solve any problems bigger than 25 (binary) variables using disjunctive cuts?
A Story

“Your thesis is great, Rob. How come you didn’t try to solve any problems bigger than 25 (binary) variables using disjunctive cuts?

“Because NLP Solvers are horrible. None of them could solve the separation problem for larger instances.”
A Story

“Your thesis is great, Rob. How come you didn’t try to solve any problems bigger than 25 (binary) variables using disjunctive cuts?”

“Because NLP Solvers are horrible. None of them could solve the separation problem for larger instances.”

To Be Fair

- Convexified problem is non-differentiable (at $\lambda = 0$)
- Still it is twice as big a NLP to solve, just to generate a cut
Jeff Hates NLP

- Can we avoid solving NLPs to generate cuts?
Jeff Hates NLP

- Can we avoid solving NLPs to generate cuts?
  - Yes, let $B \supseteq \text{conv}(R_i^- \cup R_i^+)$ be a polyhedral relaxation such that $\bar{x} \notin B$
  - We can just try to separate $\bar{x}$ from $B$
Can we avoid solving NLPs to generate cuts?

Yes, let $\mathcal{B} \supseteq \text{conv} (\mathcal{R}_i^{-} \cup \mathcal{R}_i^{+})$ be a polyhedral relaxation such that $\bar{x} \notin \mathcal{B}$

We can just try to separate $\bar{x}$ from $\mathcal{B}$

**Idea #1**

- Take linearizations at some points $\mathcal{L} = \{\bar{x}_1, \bar{x}_2, \ldots\}$
- Can iteratively augment $\mathcal{L}$
  - Add linearizations about iterates coming from solving the NLP relaxation?

$\left\{ x \in X \mid g(\bar{x}) + \nabla g(\bar{x})^T(x - \bar{x}) \leq 0 \ \forall \bar{x} \in \mathcal{L} \right\}$

These linear inequalities can be used to write a standard cut-generating linear program (CGLP)
**Jeff Hates NLP**

- Can we avoid solving NLPs to generate cuts?
  - Yes, let $\mathcal{B} \supseteq \text{conv} \left( R^k_i^- \cup R^k_i^+ \right)$ be a polyhedral relaxation such that $\bar{x} \notin \mathcal{B}$
  - We can just try to separate $\bar{x}$ from $\mathcal{B}$

---

**Idea #1**

- Take linearizations at some points $\mathcal{L} = \{\bar{x}_1, \bar{x}_2, \ldots\}$
- Can iteratively augment $\mathcal{L}$
  - Add linearizations about iterates coming from solving the NLP relaxation?

$$ \left\{ x \in X \mid g(\bar{x}) + \nabla g(\bar{x})^T (x - \bar{x}) \leq 0 \ \forall \bar{x} \in \mathcal{L} \right\} $$

- These linear inequalities can be used to write a standard cut-generating linear program (CGLP)
- This idea doesn’t work!
NLP Cut is Stronger!

- Does this mean we have to use NLP to get “strongest” disjunctive cut?
NLP Cut is Stronger!

- Does this mean we have to use NLP to get “strongest” disjunctive cut?
- Note: the cut you get from the previous procedure is a function of the linearization points $\mathcal{L}$
- The better the polyhedral outerapproximation of $\text{conv}(R_i^{k-} \cup R_i^{k+})$, the better the cut
NLP Cut is Stronger!

- Does this mean we have to use NLP to get “strongest” disjunctive cut?
- Note: the cut you get from the previous procedure is a function of the linearization points $L$.
- The better the polyhedral outerapproximation of $\text{conv} \left( R_i^{k-} \cup R_i^{k+} \right)$, the better the cut.

**Idea:** Take linearizations of points that are on both sides of the disjunction
  - $K_i^{-}:$ points with $x_i \leq k$
  - $K_i^{+}:$ points with $x_i \geq k + 1$
How to update $\mathcal{K}_-$ and $\mathcal{K}_+$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}^t_-$ and $\mathcal{K}^t_+$?
How to update $\mathcal{K}_-$ and $\mathcal{K}_+$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}_-$ and $\mathcal{K}_+$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}^-_t$ and $\mathcal{K}^+_t$?
How to update $\mathcal{K}_t^-$ and $\mathcal{K}_t^+$?
How to update $\kappa_t^-$ and $\kappa_t^+$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?
Disjunctive Cuts

Convergence of our algorithm

The Don’t Use NLP Theorem

\[ \lim_{t \to \infty} d_{L_P(t)}(\bar{x}) \to d_{NLP}(\bar{x}) \]

Yeah for LP!

The inequality generated

- cuts off \( \bar{x} \) at any iteration if \( d_{L_P(t)} \) is positive
- is supported by a point in \( \text{conv}(R_i^{-} \cup R_i^{+}) \) in the limit
- cuts off \( \bar{x} \) from simple disjunctive closure in finitely many steps if such cut exists.
Cut Generating Linear Program

- Put all linearizations (from $\mathcal{K}_-$ and $\mathcal{L}$) into matrix $A^-$
- Put all linearizations (from $\mathcal{K}_+$ and $\mathcal{L}$) into matrix $A^+$

Master Linear Program

$$\min \quad \theta$$

s.t.

$$-\theta \leq \tilde{y}_j + \tilde{z}_j - \bar{x}_j \leq \theta, \quad \forall j \in N$$

$$A^-\tilde{y} - \lambda b^- \geq 0,$$

$$A^+\tilde{z} - \mu b^+ \geq 0,$$

$$-\tilde{y}_i + \lambda k \geq 0,$$

$$\tilde{z}_i - \mu (k + 1) \geq 0,$$

$$\lambda + \mu = 1,$$

$$\lambda \geq 0, \quad \mu \geq 0,$$
Cut Generating Linear Program

- Put all linearizations (from $\mathcal{K}_-$ and $\mathcal{L}$) into matrix $A^-$
- Put all linearizations (from $\mathcal{K}_+$ and $\mathcal{L}$) into matrix $A^+$

Master Linear Program

\[
\begin{align*}
\text{min} & \quad \theta \\
\text{s.t.} & \quad -\theta \leq \tilde{y}_j + \tilde{z}_j - \tilde{x}_j \leq \theta, \quad \forall j \in \mathcal{N} \\
& \quad A^-\tilde{y} - \lambda b^- \geq 0, \\
& \quad A^+\tilde{z} - \mu b^+ \geq 0, \\
& \quad -\tilde{y}_i + \lambda k \geq 0, \\
& \quad \tilde{z}_i - \mu (k + 1) \geq 0, \\
& \quad \lambda + \mu = 1, \\
& \quad \lambda \geq 0, \quad \mu \geq 0,
\end{align*}
\]

Its Dual

\[
\begin{align*}
\text{min} & \quad \beta - \bar{x}^T(\alpha^+ - \alpha^-) \\
\text{s.t.} & \quad A^-T u - u_0 e_i - (\alpha^+ - \alpha^-) = 0, \\
& \quad A^+T v + v_0 e_i - (\alpha^+ - \alpha^-) = 0, \\
& \quad -u^T b^- + u_0 k + \beta \leq 0, \\
& \quad -v^T b^+ - v_0 (k + 1) + \beta \leq 0, \\
& \quad \sum_{j=1}^{n} (\alpha_j^+ + \alpha_j^-) = 1, \\
& \quad \alpha^+ \geq 0, \quad \alpha^- \geq 0, \quad u \geq 0, \\
& \quad v \geq 0, \quad u_0 \geq 0, \quad v_0 \geq 0,
\end{align*}
\]
Implementation Details

- Limit work for cut generation
  - Only generate inequalities for some fractional variables (ordered w.r.t. fractionality)
  - Stop rounds more quickly: If LP ’tails,’ or by a maximum number of rounds

Out of 207 instances, 148 of them are “easy” (solved in less than 300s)

FilMINT

FilMINT with DC

Arithmetic mean 24.4 25.4
Geometric mean 2.9 4.5

59 hard instances
3 hour time limit
Filmint can solve 30 of them
FilMINT with DC solve 42 of them
Implementation Details

- Limit work for cut generation
  - Only generate inequalities for some fractional variables (ordered w.r.t. fractionality)
  - Stop rounds more quickly: If LP 'tails,' or by a maximum number of rounds

- Out of 207 instances, 148 of them are “easy” (solved in less than 300s)
Implementation Details

- Limit work for cut generation
  - Only generate inequalities for some fractional variables (ordered w.r.t. fractionality)
  - Stop rounds more quickly: If LP 'tails,' or by a maximum number of rounds

- Out of 207 instances, 148 of them are “easy” (solved in less than 300s)

<table>
<thead>
<tr>
<th></th>
<th>FilMINT</th>
<th>FilMINT with DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic mean</td>
<td>24.4</td>
<td>25.4</td>
</tr>
<tr>
<td>Geometric mean</td>
<td>2.9</td>
<td>4.5</td>
</tr>
</tbody>
</table>

FilMINT with DC can solve 30 of them.

Jeff Linderoth (UW-Madison)
Implementation Details

- Limit work for cut generation
  - Only generate inequalities for some fractional variables (ordered w.r.t. fractionality)
  - Stop rounds more quickly: If LP ‘tails,’ or by a maximum number of rounds

- Out of 207 instances, 148 of them are “easy” (solved in less than 300s)

<table>
<thead>
<tr>
<th></th>
<th>FilMINT</th>
<th>FilMINT with DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic mean</td>
<td>24.4</td>
<td>25.4</td>
</tr>
<tr>
<td>Geometric mean</td>
<td>2.9</td>
<td>4.5</td>
</tr>
</tbody>
</table>

- 59 hard instances
  - 3 hour time limit
  - Filmint can solve 30 of them
  - FilMINT with DC solve 42 of them
PP with Disjunctive Cuts (Hard Instances)

![Graph showing performance measure: solution time]

- **Performance measure**: solution time
- **Proportion of problems solved**: not more than $x$-times worst than best solver
- **Comparison**: FILMINT vs. FILMINT+DC

Jeff Linderoth (UW-Madison)
## (Cherry-Picked) Results

<table>
<thead>
<tr>
<th>Problem</th>
<th>FilMINT</th>
<th>FilMINT with DC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Node</td>
<td>Time</td>
</tr>
<tr>
<td>FLay05H</td>
<td>106823</td>
<td>1771.5</td>
</tr>
<tr>
<td>fo9</td>
<td>4729736</td>
<td>10800.0</td>
</tr>
<tr>
<td>nd-14</td>
<td>113635</td>
<td>9939.8</td>
</tr>
<tr>
<td>o7</td>
<td>2433961</td>
<td>4284.1</td>
</tr>
<tr>
<td>RSync0820M03M</td>
<td>2369872</td>
<td>10800.0</td>
</tr>
<tr>
<td>SLay09H</td>
<td>53071</td>
<td>898.0</td>
</tr>
<tr>
<td>sssd-16-8-3</td>
<td>3421746</td>
<td>1691.8</td>
</tr>
<tr>
<td>Syn40M03M</td>
<td>3508363</td>
<td>10800.0</td>
</tr>
<tr>
<td>uflquad-15-80</td>
<td>3647</td>
<td>1249.9</td>
</tr>
</tbody>
</table>
Oktay “R2D2” Günlük

Jeff “Obi-Wan” Linderoth
Preprocessing for MINLP

MILP Force: Exploit The Structure!

- Mixed Integer Linear Programmers carefully study simple problem structures to come up with “good” formulations for problems.
- Good formulations closely approximate convex hull of feasible solutions.

- We study carefully the structure of a special MINLP with indicator variables.
Indicator MINLPs

- Binary variables $z$ are used as indicator variables.
- If $z_i = 0$, components of $x$ controlled by $z_i$ collapse to a point.
- If $z_i = 1$, components of $x$ controlled by $z_i$ belong to a convex set.

**Process Flow Applications**

- $z = 0 \Rightarrow x_1 = x_2 = x_3 = x_4 = 0$
- $z = 1 \Rightarrow f(x_1, x_2, x_3, x_4) \leq 0$
A Very Simple Example

$$\mathbb{R} \overset{\text{def}}{=} \left\{ (x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz \right\}$$
A Very Simple Example

\[ R \overset{\text{def}}{=} \left\{ (x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz \right\} \]

- \( z = 0 \Rightarrow x = 0, y \geq 0 \)
- \( z = 1 \Rightarrow x \leq u, y \geq x^2 \)
A Very Simple Example

\[
R \overset{\text{def}}{=} \left\{ (x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz \right\}
\]

- \( z = 0 \Rightarrow x = 0, y \geq 0 \)
- \( z = 1 \Rightarrow x \leq u, y \geq x^2 \)

Deep Insights

- \( \text{conv}(R) \equiv \text{line connecting } (0, 0, 0) \text{ to } y = x^2 \text{ in the } z = 1 \text{ plane} \)
Characterization of Convex Hull

Deep Theorem #1

\[ R = \left\{ (x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz \right\} \]

\[ \text{conv}(R) = \left\{ (x, y, z) \in \mathbb{R}^3 \mid yz \geq x^2, 0 \leq x \leq uz, 0 \leq z \leq 1, y \geq 0 \right\} \]
Characterization of Convex Hull

**Deep Theorem #1**

\[
R = \left\{ (x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz \right\}
\]

\[
\text{conv}(R) = \left\{ (x, y, z) \in \mathbb{R}^3 \mid yz \geq x^2, 0 \leq x \leq uz, 0 \leq z \leq 1, y \geq 0 \right\}
\]

\[
x^2 \leq yz, y, z \geq 0 \equiv
\]
Characterization of Convex Hull

Deep Theorem #1

\[
R = \left\{ (x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz \right\}
\]

\[
\text{conv}(R) = \left\{ (x, y, z) \in \mathbb{R}^3 \mid yz \geq x^2, 0 \leq x \leq uz, 0 \leq z \leq 1, y \geq 0 \right\}
\]

\[x^2 \leq yz, y, z \geq 0 \equiv\]
Characterization of Convex Hull

Deep Theorem #1

\[ R = \left\{ (x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz \right\} \]

\[ \text{conv}(R) = \left\{ (x, y, z) \in \mathbb{R}^3 \mid yz \geq x^2, 0 \leq x \leq uz, 0 \leq z \leq 1, y \geq 0 \right\} \]

\[ x^2 \leq yz, y, z \geq 0 \equiv \]

The Only NLP I like: Second Order Cone Programming

- \( x^2 - yz \) is not convex
- There are effective, robust algorithms for optimizing over \( \text{conv}(R) \)
General Results with Indicator Variables

- To deal with general convex sets, let $\mathcal{W} = \mathcal{W}^1 \cup \mathcal{W}^0$:

  $\mathcal{W}^0 = \{(x, z) \in \mathbb{R}^{n+1} \mid x = 0, z = 0\}$

  $\mathcal{W}^1 = \{(x, z) \in \mathbb{R}^{n+1} \mid f_k(x) \leq 0 \text{ for } k \in K, u \geq x \geq 0, z = 1\}$

---

**Theorem**

If $\mathcal{W}^1$ is convex, then $\text{conv}(\mathcal{W}) = \mathcal{W}^- \cup \mathcal{W}^0$, where

$\mathcal{W}^- = \{(x, z) \in \mathbb{R}^{n+1} \mid zf_k(x/z) \leq 0 \ \forall k \in K, uz \geq x \geq 0, 1 \geq z > 0\}$

General Results with Indicator Variables

To deal with general convex sets, let $W = W^1 \cup W^0$:

\[
W^0 = \{(x, z) \in \mathbb{R}^{n+1} \mid x = 0, z = 0\}
\]
\[
W^1 = \{(x, z) \in \mathbb{R}^{n+1} \mid f_k(x) \leq 0 \text{ for } k \in K, \ u \geq x \geq 0, \ z = 1\}
\]

**Theorem**

If $W^1$ is convex, then $\text{conv}(W) = W^- \cup W^0$, where

\[
W^- = \{(x, z) \in \mathbb{R}^{n+1} \mid zf_k(x/z) \leq 0 \ \forall k \in K, \ uz \geq x \geq 0, \ 1 \geq z > 0\}
\]

- Result also known by Stubbs, Mehrotra, Ceria, Soares, Grossmann, Lee, Frangioni, Gentile, Rockafellar, ...
Giving You Some Perspective

- For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, the perspective function $P : \mathbb{R}^{n+1} \to \mathbb{R}$ of $f$ is
  \[
P(x, z) \overset{\text{def}}{=} z f(x/z)
  \]

- The epigraph of $P(x, z)$ is a cone pointed at the origin whose lower shape is $f(x)$
Giving You Some Perspective

- For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, the perspective function $\mathcal{P} : \mathbb{R}^{n+1} \to \mathbb{R}$ of $f$ is

  $$\mathcal{P}(x, z) \overset{\text{def}}{=} zf(x/z)$$

- The epigraph of $\mathcal{P}(x, z)$ is a cone pointed at the origin whose lower shape is $f(x)$

Exploiting Your Perspective

- If $z_i$ is an indicator that the (nonlinear, convex) inequality $f(x) \leq 0$ must hold, (otherwise $x = 0$), replace the inequality with its perspective version:

  $$z_if(x/z_i) \leq 0$$

- The resulting (convex) inequality is a much tighter relaxation of the feasible region.
Death Star Location Problem

- Problem studied by Günlük, Lee, and Weismantel ('07) and classes of strong cutting planes derived
Death Star Location Problem

- Problem studied by Günlük, Lee, and Weismantel ('07) and classes of strong cutting planes derived

\[ M: \text{Death Stars} \]
\[ N: \text{Rebel Bases} \]
\[ x_{ij}: \text{percentage of rebel base } j \in N \text{ blown up by death star } i \in M \]
\[ z_i = 1 \iff \text{death star } i \in M \text{ is built} \]
\[ \text{Fixed cost for opening death star } i \in M \]
\[ \text{Quadratic cost for blowing up base } j \in N \text{ from death star } i \in M \]
Death Star Location Formulation

\[ z^* \overset{\text{def}}{=} \min \sum_{i \in M} c_i z_i + \sum_{i \in M} \sum_{j \in N} q_{ij} x_{ij}^2 \]

subject to

\[ \sum_{i \in M} x_{ij} \leq z_i \quad \forall i \in M, \forall j \in N \]
\[ \sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \]
\[ x_{ij} \geq 0 \quad \forall i \in M, \forall j \in N \]
\[ z_i \in \{0, 1\} \quad \forall i \in M \]
Death Star Location Formulation

\[ z^* \overset{\text{def}}{=} \min \sum_{i \in M} c_i z_i + \sum_{i \in M} \sum_{j \in N} q_{ij} y_{ij} \]

subject to

\[ x_{ij} \leq z_i \quad \forall i \in M, \forall j \in N \]
\[ \sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \]
\[ x_{ij} \geq 0 \quad \forall i \in M, \forall j \in N \]
\[ z_i \in \{0, 1\} \quad \forall i \in M \]
\[ x_{ij}^2 - y_{ij} \leq 0 \quad \forall i \in M, \forall j \in N \]
Death Star Location Formulation

\[ z^* \overset{\text{def}}{=} \min \sum_{i \in M} c_i z_i + \sum_{i \in M} \sum_{j \in N} q_{ij} y_{ij} \]

subject to

\[ x_{ij} \leq z_i \quad \forall i \in M, \forall j \in N \]
\[ \sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \]
\[ x_{ij} \geq 0 \quad \forall i \in M, \forall j \in N \]
\[ z_i \in \{0, 1\} \quad \forall i \in M \]
\[ x_{ij}^2 - z_i y_{ij} \leq 0 \quad \forall i \in M, \forall j \in N \]
Strength of Relaxations

- $z_R$: Value of NLP relaxation
- $z_{GLW}$: Value of NLP relaxation after GLW cuts
- $z_P$: Value of perspective relaxation
- $z^*$: Optimal solution value
Strength of Relaxations

- $z_R$: Value of NLP relaxation
- $z_{GLW}$: Value of NLP relaxation after GLW cuts
- $z_P$: Value of perspective relaxation
- $z^*$: Optimal solution value

<table>
<thead>
<tr>
<th>M</th>
<th>N</th>
<th>$z_R$</th>
<th>$z_{GLW}$</th>
<th>$z_P$</th>
<th>$z^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>30</td>
<td>140.6</td>
<td>346.5</td>
<td></td>
<td>348.7</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>141.3</td>
<td>348.7</td>
<td></td>
<td>384.1</td>
</tr>
<tr>
<td>20</td>
<td>65</td>
<td>122.5</td>
<td>289.3</td>
<td></td>
<td>289.3</td>
</tr>
<tr>
<td>25</td>
<td>80</td>
<td>121.3</td>
<td>315.8</td>
<td></td>
<td>315.8</td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>128.0</td>
<td>393.2</td>
<td></td>
<td>393.2</td>
</tr>
</tbody>
</table>
Strength of Relaxations

- \( z_R \): Value of NLP relaxation
- \( z_{GLW} \): Value of NLP relaxation after GLW cuts
- \( z_P \): Value of perspective relaxation
- \( z^* \): Optimal solution value

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>( z_R )</th>
<th>( z_{GLW} )</th>
<th>( z_P )</th>
<th>( z^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>30</td>
<td>140.6</td>
<td>326.4</td>
<td>348.7</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>141.3</td>
<td>312.2</td>
<td>384.1</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>65</td>
<td>122.5</td>
<td>248.7</td>
<td>289.3</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>80</td>
<td>121.3</td>
<td>260.1</td>
<td>315.8</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>128.0</td>
<td>327.0</td>
<td>393.2</td>
<td></td>
</tr>
</tbody>
</table>
Strength of Relaxations

- $z_R$: Value of NLP relaxation
- $z_{GLW}$: Value of NLP relaxation after GLW cuts
- $z_P$: Value of perspective relaxation
- $z^*$: Optimal solution value

| $|M|$ | $|N|$ | $z_R$  | $z_{GLW}$ | $z_P$  | $z^*$  |
|------|------|--------|-----------|--------|--------|
| 10   | 30   | 140.6  | 326.4     | 346.5  | 348.7  |
| 15   | 50   | 141.3  | 312.2     | 380.0  | 384.1  |
| 20   | 65   | 122.5  | 248.7     | 288.9  | 289.3  |
| 25   | 80   | 121.3  | 260.1     | 314.8  | 315.8  |
| 30   | 100  | 128.0  | 327.0     | 391.7  | 393.2  |
Impact of SOCP

$m = 30, n = 100$

- **Bonmin B&B**, GLW, Original: 16697 CPU seconds, 45901 nodes
- **Bonmin B&B**, GLW, w/ineq: 21206 CPU seconds, 29277 nodes
- **Bonmin B&B**, Perspective, 4201 CPU seconds, 39 B&B nodes
Impact of SOCP

$m = 30, n = 100$

- **Bonmin B&B**, GLW, Original: 16697 CPU seconds, 45901 nodes
- **Bonmin B&B**, GLW, w/ineq: 21206 CPU seconds, 29277 nodes
- **Bonmin B&B**, Perspective, 4201 CPU seconds, 39 B&B nodes
- **Mosek SOCP**, Perspective, 23 CPU seconds, 44 B&B nodes
Impact of SOCP

$m = 30, n = 100$

- Bonmin B&B, GLW, Original: 16697 CPU seconds, 45901 nodes
- Bonmin B&B, GLW, w/ineq: 21206 CPU seconds, 29277 nodes
- Bonmin B&B, Perspective, 4201 CPU seconds, 39 B&B nodes
- Mosek SOCP, Perspective, 23 CPU seconds, 44 B&B nodes

Larger Instances

<table>
<thead>
<tr>
<th>m</th>
<th>n</th>
<th>T</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>200</td>
<td>141.9</td>
<td>63</td>
</tr>
<tr>
<td>40</td>
<td>100</td>
<td>76.4</td>
<td>54</td>
</tr>
<tr>
<td>40</td>
<td>200</td>
<td>101.3</td>
<td>45</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>61.6</td>
<td>49</td>
</tr>
<tr>
<td>50</td>
<td>200</td>
<td>140.4</td>
<td>47</td>
</tr>
</tbody>
</table>

“The Force is Strong with This One”
Strong Relaxations for Multilinear Programs

EXTENDED FORMULATIONS

Jeff “Obi-Wan” Linderoth

Jim “Mace” Luedtke

Mahdi “C3P0” Namazifar
Reformulating a Nonconvex Problem

**Multilinear Program**

\[
\min_{x \in [l,u]} \left\{ c^T x \mid \sum_{t \in T_i} a_{it} \prod_{j \in J_t} x_j = b_i \ \forall i = 1, \ldots, m \right\} \quad (MLP)
\]
Reformulating a Nonconvex Problem

Multilinear Program

\[
\min_{x \in [l,u]} \{ c^T x \mid \sum_{t \in T_i} a_{it} \prod_{j \in J_t} x_j = b_i \ \forall i = 1, \ldots, m \} \quad (MLP)
\]

Let \( z_t = \prod_{j \in J_t} x_j \), \( T \overset{\text{def}}{=} \bigcup_i T_i \).

\[
X = \{ x \in [l,u], z \in \mathbb{R}^{|T|} \mid z_t = \prod_{j \in J_t} x_j, \ t \in T \}
\]
Reformulating a Nonconvex Problem

**Multilinear Program**

\[
\min_{x \in [l,u]} \{ c^T x \mid \sum_{t \in T_i} a_{it} \prod_{j \in J_t} x_j = b_i \ \forall i = 1, \ldots, m \} \quad (MLP)
\]

- Let \( z_t = \prod_{j \in J_t} x_j \), \( T \overset{\text{def}}{=} \bigcup_i T_i \).

\[
X = \{ x \in [l,u], z \in \mathbb{R}^{|T|} \mid z_t = \prod_{j \in J_t} x_j, \ t \in T \}
\]

- Now (MLP) is

\[
\min_{(x,z) \in X} \{ c^T x \mid \sum_{t \in T_i} a_{it} z_t = b_i \ \forall i = 1, \ldots, m \}
\]

- Since we have ignored the constraints \( \sum_{t \in T_i} a_{it} z_t = b_i \), getting \( \text{conv}(X) \) doesn’t solve the problem.
  - But it won’t hurt either!
conv(\(X\)) is a Polyhedron!

\[ X = \{ x \in [l, u], z \in \mathbb{R}^{|T|} \mid z_t = \prod_{j \in J_t} x_j, \; t \in T \} \]

- conv(\(X\)) is a polyhedron. (Rikun '97):

\[
\text{conv}(X) = \text{Proj}_{x,z} \left\{ (x, z, \lambda) \in [l, u] \times \mathbb{R}^{|T|} \times \Delta_{2^n} \mid x = \sum_{k=1}^{2^n} \lambda^k \chi^k, \right. \\
\left. z_t = \sum_{k=1}^{2^{|J_t|}} \lambda^k \prod_{j \in J_t} \chi^k_j, \; t = 1, 2, \ldots, |T| \right\},
\]

- For each of the \(2^n\) extreme points \( \chi^k \) of the hyper-rectangle \([l, u]\\): evaluate (vector-valued) multilinear function at \( \chi^k \)
**conv**(**X**) is a Polyhedron!

\[ X = \{ x \in [\ell, u], z \in \mathbb{R}^{|T|} \mid z_t = \prod_{j \in J_t} x_j, \; t \in T \} \]

- **conv**(**X**) is a polyhedron. (Rikun '97):

\[ \text{conv}(X) = \text{Proj}_{x,z} \left\{ (x, z, \lambda) \in [\ell, u] \times \mathbb{R}^{|T|} \times \Delta_{2^n} \mid x = \sum_{k=1}^{2^n} \lambda^k \chi^k, \right. \]

\[ z_t = \sum_{k=1}^{2^{|J_t|}} \lambda^k \prod_{j \in J_t} \chi_j^k, \; t = 1, 2, \ldots, |T| \} \}

- For each of the \(2^n\) extreme points \(\chi^k\) of the hyper-rectangle \([\ell, u]\): evaluate (vector-valued) multilinear function at \(\chi^k\)

---

**Rats!**

- **conv**(X) has \(2^n\) variables
\[
\text{conv}\left(\{(x, z) \in [0, 1]^2 \times \mathbb{R} \mid z = x_1 x_2\}\right)
\]
\[ \text{conv}\left( \{(x, z) \in [0, 1]^2 \times \mathbb{R} \mid z = x_1 x_2 \} \right) \]
\[ \text{conv}(\{(x, z) \in [0, 1]^2 \times \mathbb{R} \mid z = x_1 x_2\}) \]
Tradeoffs in Building Relaxations

- **Convex hull**: Tight but exponentially large
- **Traditional “McCormick”**: Loose but relatively small
Tradeoffs in Building Relaxations

- **Convex hull**: Tight but exponentially large
- Traditional **“McCormick”**: Loose but relatively small
- **IDEA**: Group the variables together and build convex hull for each group.
Tradeoffs in Building Relaxations

- **Convex hull**: Tight but exponentially large
- **Traditional “McCormick”**: Loose but relatively small
- **IDEA**: Group the variables together and build convex hull for each group.

**Not the Only Idea**
- Bao, Sahinidis and Tawarmalani (’11) propose a related method:
  - Rows are considered *independently*
  - For each row they use a cutting plane mechanism to *separate* for the facets of the convex hull.

**Our Method**
- Rows are considered *all at once*
- The extended formulation is used to build relaxations
**Term Cover**

**Definition**

A finite collection of subsets of \( \{1, \ldots, n\} \), \( C = \{C_g\}_{g \in G} \), is called a **term-cover** of \( T \) if for all \( t \in T \) there exists \( g \in G \) such that \( J_t \subseteq C_g \).
Term Cover

Definition

A finite collection of subsets of \{1, \ldots, n\}, \( C = \{C_g\}_{g \in G} \), is called a term-cover of \( T \) if for all \( t \in T \) there exists \( g \in G \) such that \( J_t \subseteq C_g \).

\[
T = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{1, 6\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}, \{4, 6\}, \{5, 7\}, \{6, 7\}\}
\]
\[
C = \{\{1, 2, 3, 4\}, \{2, 3, 4, 5\}, \{1, 5, 6, 7\}, \{4, 6\}\}
\]
Term Cover

Definition
A finite collection of subsets of \(\{1, \ldots, n\}\), \(C = \{C_g\}_{g \in G}\), is called a term-cover of \(T\) if for all \(t \in T\) there exists \(g \in G\) such that \(J_t \subseteq C_g\).

\[
T = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{1, 6\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}, \{4, 6\}, \{5, 7\}, \{6, 7\}\}
\]

\[
C = \{\{1, 2, 3, 4\}, \{2, 3, 4, 5\}, \{1, 5, 6, 7\}, \{4, 6\}\}
\]

Generalizes...
- Term-by-term (McCormick) relaxation: \(C_t = J_t, \ t \in T\)
- Row-by-row relaxation: \(C_i = \bigcup_{t \in T, j \in J_t} \{j\}\)
- Convex hull: \(C_1 = \{1, \ldots, n\}\)
Term Cover Relaxation

- **Idea:** Build the convex hull formulation for each element (group of variables) in the term cover. First enumerate extreme points of box:

$$\{x^{g,k}\}_{k=1}^{2^{|C_g|}} = \text{Vert}([\ell_j \leq x_j \leq u_j, j \in C_g]) \forall g \in G$$
Term Cover Relaxation

- Idea: Build the convex hull formulation for each element (group of variables) in the term cover. First enumerate extreme points of box:

\[ \{x^{g,k}_j \}_{k=1}^{2^{|C_g|}} = \text{Vert}(\{\ell_j \leq x_j \leq u_j, j \in C_g\}) \forall g \in G \]

- Term Cover Relaxation: \( TC_C(X_\phi) \) is the set of all \((x, z) \in [l, u] \times \mathbb{R}^{|T|}\) such that

\[
\chi_{C_g} = \sum_{k=1}^{2^{|C_g|}} \lambda_{g,k}^k \chi_{g,k}, \quad \sum_{k=1}^{2^{|C_g|}} \lambda_{g,k}^g = 1, \quad \forall g \in G
\]

\[
z_t = \sum_{k=1}^{2^{|C_g|}} \lambda_{g,k}^g \prod_{j \in J_t} \chi_{j,k}, \quad \forall g \in G, t \in T : J_t \subseteq C_g.
\]
Building Good Term Covers

Overarching Goal

Get strong bounds fast

Questions

1. How should the maximum size ($\sigma$) of elements of the term cover be?

   (A): Not too big! ($\sigma \leq 6$) — You need 2 $\sigma$ variables for that element.

2. Which (original) variables should appear in the same element of the term cover?

3. How many elements should the term cover contain?

We (read Mahdi) tried lots of different strategies in his thesis work, but none did as well as a simple greedy two-phase heuristic.
Building Good Term Covers

**Overarching Goal**
Get strong bounds fast

**Questions**

1. (Q): How should be the maximum size ($\sigma$) of elements of the term cover.
   - (A): Not too big! ($\sigma \leq 6$) — You need $2^\sigma$ variables for that element.

2. (Q): Which (original) variables should appear in the same element of the term cover?

3. (Q): How many elements should the term cover contain?
Building Good Term Covers

**Overarching Goal**
Get strong bounds fast

**Questions**

1. (Q): How should be the maximum size ($\sigma$) of elements of the term cover.
   - (A): Not too big! ($\sigma \leq 6$)—You need $2^\sigma$ variables for that element.

2. (Q): Which (original) variables should appear in the same element of the term cover?

3. (Q): How many elements should the term cover contain?

- We (read Mahdi) tried *lots of different strategies* in his thesis work, but none did as well as a simple greedy two-phase heuristic.
Is It Good?

- MANY different methods were tried for building good term-covers.
- This greedy two-phase heuristic was by far the best.
Is It Good?

- MANY different methods were tried for building good term-covers.
- This greedy two-phase heuristic was by far the best.

A Small Experiment

- Compared bounds to a 40 random term covers of the same size.
- Greedy heuristic always beats the best of the 40 random covers.
Is It Good?

- MANY different methods were tried for building good term-covers
- This greedy two-phase heuristic was by far the best

A Small Experiment

- Compared bounds to a 40 random term covers of the same size.
- Greedy heuristic always beats the best of the 40 random covers.

The REAL Test

- Is this new method comparable with the state-of-the art—Semidefinite Programming for Nonconvex Quadratic Programming?
SDP Relaxation for QCQPs

**QCQP**

\[
\begin{align*}
\text{min} & \quad x^T Q_0 x + a_0^T x \\
\text{s.t.} & \quad x^T Q_i x + a_i^T x \leq b_i, \quad \forall i \\
& \quad \ell \leq x \leq u
\end{align*}
\]

**Reformulate**

\[
\begin{align*}
\text{min} & \quad Q_0 \bullet X + a_0^T x \\
\text{s.t.} & \quad Q_i \bullet X + a_i^T x \leq b_i, \quad \forall i \\
& \quad X - xx^T \succeq 0 \\
& \quad \ell \leq x \leq u
\end{align*}
\]

This is the state of the art in getting bounds on nonconvex QPs.
SDP Relaxation for QCQPs

**QCQP**

\[
\begin{align*}
\text{min} & \quad x^T Q_0 x + a_0^T x \\
\text{s.t.} & \quad x^T Q_i x + a_i^T x \leq b_i, \quad \forall i \\
& \quad \ell \leq x \leq u
\end{align*}
\]

**Relax**

\[
\begin{align*}
\text{min} & \quad Q_0 \cdot X + a_0^T x \\
\text{s.t.} & \quad Q_i \cdot X + a_i^T x \leq b_i, \quad \forall i \\
& \quad X - xx^T \succeq 0 \\
& \quad \ell \leq x \leq u
\end{align*}
\]

This is the state of the art in getting bounds on nonconvex QPs.

**Reformulate**

\[
\begin{align*}
\text{min} & \quad Q_0 \cdot X + a_0^T x \\
\text{s.t.} & \quad Q_i \cdot X + a_i^T x \leq b_i, \quad \forall i \\
& \quad X - xx^T = 0 \\
& \quad \ell \leq x \leq u
\end{align*}
\]
Why I Don’t Fit In At Wisconsin

Jeff
Why I Don’t Fit In At Wisconsin

Jeff Hates

No Love
Why I Don’t Fit In At Wisconsin

Jeff Hates SDP

No Love
Why I Don’t Fit In At Wisconsin

Jeff Hates SDP

No Love

Computational Comparison

- Scatter Plot of root optimality gap for
  1. Term-by-Term vs Term-Cover
  2. SDP versus Term-Cover
Box QP – Term-Cover vs. McCormick

Red points correspond to large instances: 70 – 100 variables
Blue points correspond to small and medium instances: 20 – 60 variables
Box QP – Term-Cover vs. SDP

Surprising! Linear term-cover relaxation is competitive with strong SDP relaxations.

Jeff Linderoth (UW-Madison)
Surprising!

Linear term-cover relaxation is competitive with strong SDP relaxations.
### Quadratic With Linear Cons

<table>
<thead>
<tr>
<th>Problem</th>
<th>SDP</th>
<th>Term-Cover $\sigma = 6, \nu = 2$</th>
<th>McCormick</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G%</td>
<td>T</td>
<td>G%</td>
</tr>
<tr>
<td>qkp020-025-1</td>
<td>4.0</td>
<td>1.5</td>
<td>1.4</td>
</tr>
<tr>
<td>qkp020-025-2</td>
<td>2.2</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>qkp020-050-1</td>
<td>1.9</td>
<td>0.5</td>
<td>1.9</td>
</tr>
<tr>
<td>qkp020-050-2</td>
<td>2.8</td>
<td>0.4</td>
<td>0.0</td>
</tr>
<tr>
<td>qkp020-075-1</td>
<td>2.8</td>
<td>0.5</td>
<td>0.2</td>
</tr>
<tr>
<td>qkp020-075-2</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>qkp040-025-1</td>
<td>5.5</td>
<td>4.1</td>
<td>0.7</td>
</tr>
<tr>
<td>qkp040-025-2</td>
<td>8.4</td>
<td>4.1</td>
<td>1.4</td>
</tr>
<tr>
<td>qkp040-050-1</td>
<td>6.8</td>
<td>3.9</td>
<td>1.4</td>
</tr>
<tr>
<td>qkp040-050-2</td>
<td>13.3</td>
<td>3.6</td>
<td>11.9</td>
</tr>
<tr>
<td>qkp040-075-1</td>
<td>6.8</td>
<td>4.0</td>
<td>2.6</td>
</tr>
<tr>
<td>qkp040-075-2</td>
<td>4.3</td>
<td>3.7</td>
<td>0.6</td>
</tr>
<tr>
<td>qkp060-025-1</td>
<td>10.8</td>
<td>101.8</td>
<td>8.6</td>
</tr>
<tr>
<td>qkp060-025-2</td>
<td>6.6</td>
<td>101.8</td>
<td>1.5</td>
</tr>
<tr>
<td>qkp060-050-1</td>
<td>6.3</td>
<td>119.1</td>
<td>6.2</td>
</tr>
<tr>
<td>qkp060-050-2</td>
<td>9.4</td>
<td>96.0</td>
<td>12.1</td>
</tr>
<tr>
<td>qkp060-075-1</td>
<td>11.0</td>
<td>97.8</td>
<td>21.5</td>
</tr>
<tr>
<td>qkp060-075-2</td>
<td>11.5</td>
<td>103.6</td>
<td>20.6</td>
</tr>
<tr>
<td>qkp080-025-1</td>
<td>6.0</td>
<td>1695.7</td>
<td>4.4</td>
</tr>
<tr>
<td>qkp080-025-2</td>
<td>10.2</td>
<td>1606.3</td>
<td>10.5</td>
</tr>
<tr>
<td>qkp080-050-1</td>
<td>5.4</td>
<td>1671.1</td>
<td>9.9</td>
</tr>
<tr>
<td>qkp080-050-2</td>
<td>8.6</td>
<td>1423.5</td>
<td>15.1</td>
</tr>
<tr>
<td>qkp080-075-1</td>
<td>14.9</td>
<td>1396.3</td>
<td>34.4</td>
</tr>
<tr>
<td>qkp080-075-2</td>
<td>12.0</td>
<td>1445.1</td>
<td>34.5</td>
</tr>
<tr>
<td>Average</td>
<td>7.2</td>
<td>411.9</td>
<td>8.4</td>
</tr>
</tbody>
</table>
## Quadratic With Linear Cons

<table>
<thead>
<tr>
<th>Problem</th>
<th>SDP G%</th>
<th>T</th>
<th>Term-Cover $\sigma = 6$, $\nu = 2$ G%</th>
<th>T</th>
<th>McCormick G%</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>qkp020-025-1</td>
<td>4.0</td>
<td>1.5</td>
<td>1.4</td>
<td>0.0</td>
<td>1.8</td>
<td>0.0</td>
</tr>
<tr>
<td>qkp020-025-2</td>
<td>2.2</td>
<td>0.4</td>
<td>0.5</td>
<td>0.0</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>qkp020-050-1</td>
<td>1.9</td>
<td>0.5</td>
<td>1.9</td>
<td>0.1</td>
<td>8.2</td>
<td>0.0</td>
</tr>
<tr>
<td>qkp020-050-2</td>
<td>2.8</td>
<td>0.4</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>qkp020-075-1</td>
<td>2.8</td>
<td>0.5</td>
<td>0.2</td>
<td>0.2</td>
<td>27.4</td>
<td>0.1</td>
</tr>
<tr>
<td>qkp020-075-2</td>
<td>1.0</td>
<td>0.3</td>
<td>0.0</td>
<td>0.2</td>
<td>6.8</td>
<td>0.1</td>
</tr>
<tr>
<td>qkp040-025-1</td>
<td>5.5</td>
<td>4.1</td>
<td>0.7</td>
<td>0.4</td>
<td>7.3</td>
<td>0.1</td>
</tr>
<tr>
<td>qkp040-025-2</td>
<td>8.4</td>
<td>4.1</td>
<td>1.4</td>
<td>0.4</td>
<td>20.9</td>
<td>0.1</td>
</tr>
<tr>
<td>qkp040-050-1</td>
<td>6.8</td>
<td>3.9</td>
<td>1.4</td>
<td>1.5</td>
<td>51.0</td>
<td>0.2</td>
</tr>
<tr>
<td>qkp040-050-2</td>
<td>13.3</td>
<td>3.6</td>
<td>11.9</td>
<td>2.0</td>
<td>89.5</td>
<td>0.2</td>
</tr>
<tr>
<td>qkp040-075-1</td>
<td>6.8</td>
<td>4.0</td>
<td>2.6</td>
<td>5.3</td>
<td>80.7</td>
<td>0.4</td>
</tr>
<tr>
<td>qkp040-075-2</td>
<td>4.3</td>
<td>3.7</td>
<td>0.6</td>
<td>4.9</td>
<td>104.8</td>
<td>0.4</td>
</tr>
<tr>
<td>qkp060-025-1</td>
<td>10.8</td>
<td>101.8</td>
<td>8.6</td>
<td>3.1</td>
<td>52.5</td>
<td>0.3</td>
</tr>
<tr>
<td>qkp060-025-2</td>
<td>6.6</td>
<td>101.8</td>
<td>1.5</td>
<td>2.7</td>
<td>34.8</td>
<td>0.3</td>
</tr>
<tr>
<td>qkp060-050-1</td>
<td>6.3</td>
<td>119.1</td>
<td>6.2</td>
<td>14.9</td>
<td>91.6</td>
<td>0.8</td>
</tr>
<tr>
<td>qkp060-050-2</td>
<td>9.4</td>
<td>96.0</td>
<td>12.1</td>
<td>15.7</td>
<td>119.3</td>
<td>0.7</td>
</tr>
<tr>
<td>qkp060-075-1</td>
<td>11.0</td>
<td>97.8</td>
<td>21.5</td>
<td>55.0</td>
<td>173.6</td>
<td>1.5</td>
</tr>
<tr>
<td>qkp060-075-2</td>
<td>11.5</td>
<td>103.6</td>
<td>20.6</td>
<td>52.2</td>
<td>169.3</td>
<td>1.3</td>
</tr>
<tr>
<td>qkp080-025-1</td>
<td>6.0</td>
<td>1695.7</td>
<td>4.4</td>
<td>14.7</td>
<td>60.6</td>
<td>0.6</td>
</tr>
<tr>
<td>qkp080-025-2</td>
<td>10.2</td>
<td>1606.3</td>
<td>10.5</td>
<td>13.2</td>
<td>67.3</td>
<td>0.5</td>
</tr>
<tr>
<td>qkp080-050-1</td>
<td><strong>5.4</strong></td>
<td><strong>1671.1</strong></td>
<td><strong>9.9</strong></td>
<td><strong>80.0</strong></td>
<td><strong>115.2</strong></td>
<td><strong>2.0</strong></td>
</tr>
<tr>
<td>qkp080-050-2</td>
<td><strong>8.6</strong></td>
<td><strong>1423.5</strong></td>
<td><strong>15.1</strong></td>
<td><strong>72.8</strong></td>
<td><strong>145.5</strong></td>
<td><strong>1.6</strong></td>
</tr>
<tr>
<td>qkp080-075-1</td>
<td><strong>14.9</strong></td>
<td><strong>1396.3</strong></td>
<td><strong>34.4</strong></td>
<td><strong>348.1</strong></td>
<td><strong>231.8</strong></td>
<td><strong>4.3</strong></td>
</tr>
<tr>
<td>qkp080-075-2</td>
<td><strong>12.0</strong></td>
<td><strong>1445.1</strong></td>
<td><strong>34.5</strong></td>
<td><strong>364.2</strong></td>
<td><strong>236.3</strong></td>
<td><strong>3.5</strong></td>
</tr>
<tr>
<td>Average</td>
<td>7.2</td>
<td>411.9</td>
<td>8.4</td>
<td>43.8</td>
<td>79.0</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Work in Progress
Work in Progress

- We are still working on getting relaxations implemented.
Conclusions

- My 9-year old likes Star Wars
Conclusions

- My 9-year old likes Star Wars

The MILP Force is Powerful

- Applying “traditional” techniques from MILP in the domain of MINLP can lead to significant improvements in our ability to solve instances
Conclusions

- My 9-year old likes Star Wars

---

The MILP Force is Powerful

- Applying “traditional” techniques from MILP in the domain of MINLP can lead to significant improvements in our ability to solve instances

---

Final Frontiers

- Keep using the MILP force on MINLP
  - Strong formulations
  - Cutting planes
  - Branching rules
  - Heuristics
MINLP: A New Hope?!

- MINOTAUR: Mixed Integer Nonlinear Optimization Toolkit: Algorithms, Underestimators, Refinements
MINLP: A New Hope?!

- MINOTAUR: Mixed Integer Nonlinear Optimization Toolkit: Algorithms, Underestimators, Refinements
- Framework & toolbox for solving MINLPs
- Implemented algorithms in MINOTAUR:
  - branch-and-\{bound|cut\} for convex MINLPs
  - branch-and-bound for nonconvex quadratic constraints
  - branch-and-cut for multilinear constraints
- Extensible: implement new MINLP algorithms, solvers