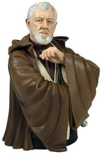
MINLP WARS

Mixed Integer Nonlinear Programming

Jeff "Obi-Wan" Linderoth

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



NGB/LNMB Workshop · Lunteren, the Netherlands · January 17, 2013

Our Quest

Mixed Integer Nonlinear Program: (MINLP)

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

- $X \stackrel{\text{def}}{=} \{x \mid x \in \mathbb{R}^n_+, Dx \leq d\}$
- f, g_j are continuously differentiable functions.
  - f, g_j linear $\Rightarrow$ MILP
- MINLPs combine challenge of nonlinearities with discrete choice

Our Quest

Mixed Integer Nonlinear Program: (MINLP)

$$\begin{aligned} z_{\mathrm{MINLP}} = \min \quad & \eta \\ \text{subject to} \quad & g_j(x) \le 0, \quad j = 1, \ldots, m \\ & f(x) \le \eta \\ & x \in X, \quad x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

- $X \stackrel{\text{def}}{=} \{x \mid x \in \mathbb{R}^n_+, Dx \le d\}$
- f, g_j are continuously differentiable functions.
  - f, g_j linear $\Rightarrow$ MILP
- MINLPs combine challenge of nonlinearities with discrete choice
- WLOG: We sometimes want $f(x)$ linear

Apology accepted

- The talk is a bit of a star-blaster approach to MINLP.
- Not many technical details.
- Hopefully my subsequent colleagues will fill them in!

Apology accepted

- The talk is a bit of a star-blaster approach to MINLP.
- Not many technical details.
- Hopefully my subsequent colleagues will fill them in!

Overview

- Applications/Models
- Basic algorithms: NLP-Based Branch-and-Bound, Linearization-Based methods, Spatial branch and bound for global optimization
- An important modeling trick

Solving MINLP – The Talk Theme

USE THE MILP

Solving MINLP – The Talk Theme

USE THE MILP

The MILP Force

- MILP has become a commodity technology
- Instances unsolvable a decade ago have now become routine

Solving MINLP – The Talk Theme

USE THE MILP

The MILP Force

- MILP has become a commodity technology
- Instances unsolvable a decade ago have now become routine

Use the MILP

- Strategies that have been effective for MILP should also be effective for MINLP

Important Special Cases

Mixed Integer Nonlinear Program: (MINLP)

$$\begin{aligned} z_{\mathrm{MINLP}} = \min \quad & f(x) \\ \text{subject to} \quad & g_j(x) \leq 0, \quad j = 1, \ldots, m \\ & x \in X, \quad x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

Important Special Cases

Mixed Integer Nonlinear Program: (MINLP)

$$\begin{aligned} z_{\text{MINLP}} = \min \quad & f(x) \\ \text{subject to} \quad & g_j(x) \le 0, \quad j = 1, \ldots, m \\ & x \in X, \quad x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

What's in a name?

- If f, g_j are convex, this is called convex MINLP
  - The set $\{x \in X \mid g_j(x) \le 0 \ \forall j\}$ is a convex set, and minimizing a convex function over a convex set is easy

Important Special Cases

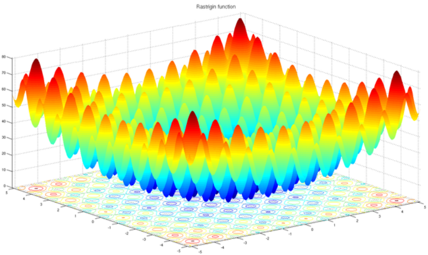
Mixed Integer Nonlinear Program: (MINLP)

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

What's in a name?

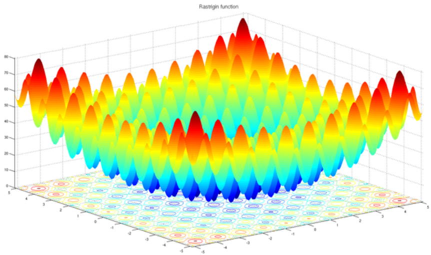
- If f, g_j are convex, this is called convex MINLP
  - The set $\{x \in X \mid g_j(x) \leq 0 \ \forall j\}$ is a convex set, and minimizing a convex function over a convex set is easy
- If f, g_j quadratic, e.g. $f(x) = x^T Q x + b^T x + c$, the problem is called a mixed integer quadratic program (MIQP)
  - Convex MIQP ($Q \succeq 0$)
  - Nonconvex MIQP ($Q \not\succeq 0$)

Why Important?



- Without convexity, many solvers only guarantee solution to a local optimum

Why Important?



- Without convexity, many solvers only guarantee solution to a local optimum

---

- Non-convex instances may require significant computational effort
- We will (attempt) to explain why later

Software for MINLP

Convex MINLP

- ALPHA-ECP, Bonmin, DICOPT, FilMINT, KNITRO, MINLP-BB, SBB

Software for MINLP

Convex MINLP

- ALPHA-ECP, Bonmin, DICOPT, FilMINT, KNITRO, MINLP-BB, SBB

(Convex) MIQP

- CPLEX, GUROBI, MOSEK, XPRESS

Software for MINLP

Convex MINLP

- ALPHA-ECP, Bonmin, DICOPT, FilMINT, KNITRO, MINLP-BB, SBB

(Convex) MIQP

- CPLEX, GUROBI, MOSEK, XPRESS

(Nonconvex) MIQP

- GLOMIQO

Software for MINLP

Convex MINLP

- ALPHA-ECP, Bonmin, DICOPT, FilMINT, KNITRO, MINLP-BB, SBB

(Convex) MIQP

- CPLEX, GUROBI, MOSEK, XPRESS

(Nonconvex) MIQP

- GLOMIQO

General Global Optimization

- BARON, Couenne, LINDOGlobal, SCIP

Software for MINLP

Convex MINLP

- ALPHA-ECP, Bonmin, DICOPT, FilMINT, KNITRO, MINLP-BB, SBB

(Convex) MIQP

- CPLEX, GUROBI, MOSEK, XPRESS

(Nonconvex) MIQP

- GLOMIQO

General Global Optimization

- BARON, Couenne, LINDOGlobal, SCIP

Try 'em on NEOS

- http://www.neos-server.org/neos/solvers/index.html

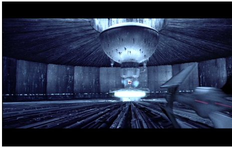
The Rebel Alliance



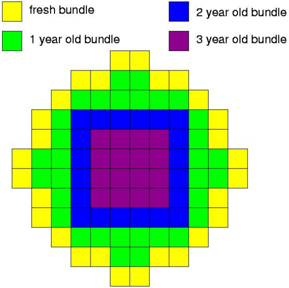
- While most of this material is at an introductory level, the "new" material I am presenting is joint with many talented members of the rebel alliance



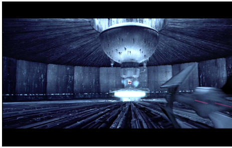
Application: Death Star Core Reload



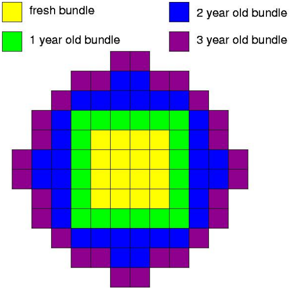
- Maximize reactor efficiency after reload subject to diffusion PDE
- **Discrete:** Placement (and age) of bundles.
- **Nonlinear:** Diffusion PDE
  $\Rightarrow$ A MINLP
- Avoid reactor becoming sub-critical



Application: Death Star Core Reload



- Maximize reactor efficiency after reload subject to diffusion PDE
- **Discrete:** Placement (and age) of bundles.
- **Nonlinear:** Diffusion PDE
  $\Rightarrow$ A MINLP
- Avoid reactor becoming overheated



Other MINLPs

1. Gas/Water Network Design—Nonconvex MINLP
   - Discrete: Pipe connections/sizes
   - Nonlinear: Pressure Loss

Other MINLPs

1. Gas/Water Network Design—Nonconvex MINLP
   - Discrete: Pipe connections/sizes
   - Nonlinear: Pressure Loss
2. Sparse Approximation—Convex MIQP
   - Discrete: Selection of elements
   - Nonlinear: (2-norm) model error

Other MINLPs

1. Gas/Water Network Design—Nonconvex MINLP
   - Discrete: Pipe connections/sizes
   - Nonlinear: Pressure Loss
2. Sparse Approximation—Convex MIQP
   - Discrete: Selection of elements
   - Nonlinear: (2-norm) model error
3. Petrochemical—Nonconvex MINLP
   - Discrete: Which process to use?
   - Nonlinear: Product Blending (among others)

Other MINLPs

1. Gas/Water Network Design—Nonconvex MINLP
   - Discrete: Pipe connections/sizes
   - Nonlinear: Pressure Loss
2. Sparse Approximation—Convex MIQP
   - Discrete: Selection of elements
   - Nonlinear: (2-norm) model error
3. Petrochemical—Nonconvex MINLP
   - Discrete: Which process to use?
   - Nonlinear: Product Blending (among others)
4. Portfolio Management—Convex MIQP
   - Discrete: Trading Strategy
   - Utility: Utility

Other MINLPs

1. Gas/Water Network Design—Nonconvex MINLP
   - Discrete: Pipe connections/sizes
   - Nonlinear: Pressure Loss
2. Sparse Approximation—Convex MIQP
   - Discrete: Selection of elements
   - Nonlinear: (2-norm) model error
3. Petrochemical—Nonconvex MINLP
   - Discrete: Which process to use?
   - Nonlinear: Product Blending (among others)
4. Portfolio Management—Convex MIQP
   - Discrete: Trading Strategy
   - Utility: Utility
5. Supply Chain—Convex MINLP
   - Discrete: Fixed charges for opening facilities
   - Nonlinear: Nonlinear transportation costs

Portfolio Management

- N: Universe of asset to purchase
- x_i: % investment in asset i
- α_i: Expected return of asset i
- R: Minimum desired expected return

$$\min_{x \in \mathbb{R}_+^{|N|}} \left\{ u(x) \mid \sum_{i \in N} x_i = 1, \sum_{i \in N} \alpha_i x_i \geq R \right\}$$

Portfolio Management

- N: Universe of asset to purchase
- x_i: % investment in asset i
- α_i: Expected return of asset i
- R: Minimum desired expected return

$$\min_{x \in \mathbb{R}_+^{|N|}} \left\{ u(x) \mid \sum_{i \in N} x_i = 1, \sum_{i \in N} \alpha_i x_i \geq R \right\}$$

- "Markowitz": $u(x) \stackrel{\text{def}}{=} x^T Q x$
  - $u(x)$: Variance of return if hold portfolio x
  - Q: Variance-covariance matrix of expected returns

Portfolio Management

Limit Names: $|i \in N \ : \ x_i > 0| \leq K$

- Use binary indicator variables to model the implication $x_i > 0 \Rightarrow z_i = 1$
- Implication modeled with variable upper bounds:

$$x_i \leq Bz_i \qquad \forall i \in N$$

- Then add cardinality: $\sum_{i \in N} z_i \leq K$

Portfolio Management

Limit Names: $|i \in N \ : \ x_i > 0| \leq K$

- Use binary indicator variables to model the implication $x_i > 0 \Rightarrow z_i = 1$
- Implication modeled with variable upper bounds:

$$x_i \leq Bz_i \qquad \forall i \in N$$

- Then add cardinality: $\sum_{i \in N} z_i \leq K$

Min Holdings: $(x_i = 0) \vee (x_i \geq m)$

- Model implication: $x_i > 0 \Rightarrow x_i \geq m$
- $x_i > 0 \Rightarrow y_i = 1 \Rightarrow x_i \geq m$
- $x_i \leq By_i, x_i \geq my_i \ \forall i \in N$

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: Death Stars
- N: Rebel Bases
- x_{ij}: percentage of rebel base $j \in N$ blown up by death star $i \in M$
- $z_i = 1 \Leftrightarrow$ death star $i \in M$ is built
- Fixed cost for opening death star $i \in M$
- Quadratic cost for blowing up base $j \in N$ from death star $i \in M$

Death Star Location Formulation

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

subject to

$$\begin{aligned} x_{ij} &\leq z_i && \forall i \in M, \forall j \in N \\ \sum_{i \in M} x_{ij} &= 1 && \forall j \in N \\ x_{ij} &\geq 0 && \forall i \in M, \forall j \in N \\ z_i &\in \{0, 1\} && \forall i \in M \end{aligned}$$

Death Star Location Formulation

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

subject to

$$\begin{aligned} x_{ij} &\leq z_i && \forall i \in M, \forall j \in N \\ \sum_{i \in M} x_{ij} &= 1 && \forall j \in N \\ x_{ij} &\geq 0 && \forall i \in M, \forall j \in N \\ z_i &\in \{0, 1\} && \forall i \in M \end{aligned}$$

Remember this!

- We'll show a better formulation at the end

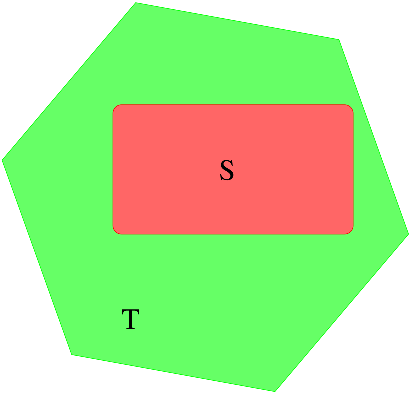
Focus Today – How to Solve MINLPs

Focus Today – How to Solve MINLPs

- Good relaxations are key

Relaxations

- Let $z_S = \min f(x) : x \in S$
- Let $z_T = \min f(x) : x \in T$
- We say that T is a relaxation of S if $S \subseteq T$



Question Time

- What can we say about the relationship between z_S and z_T?

Question Time

- What can we say about the relationship between z_S and z_T?

The "Ice Cream Theorem" (According to my 10-year old)

- More is better!

Question Time

- What can we say about the relationship between z_S and z_T?

The "Ice Cream Theorem" (According to my 10-year old)

- More is better!
- Since we have more to choose from in T, the best point in T must be at least as good as the best point in S: $z_T \leq z_S$

Question Time

- What can we say about the relationship between z_S and z_T?

The "Ice Cream Theorem" (According to my 10-year old)

- More is better!
- Since we have more to choose from in T, the best point in T must be at least as good as the best point in S: $z_T \leq z_S$

The Upshot

- We get lower bounds on z_{MINLP} in MINLP algorithms by solving a relaxation of the problem

More Simple Stuff

- If $\hat{x} \in S$, then the value $f(\hat{x}) \geq z_S$
  - There may be better solutions, but here is one...
- We get upper bounds on z_{MINLP} from feasible solutions

More Simple Stuff

- If $\hat{x} \in S$, then the value $f(\hat{x}) \geq z_S$
  - There may be better solutions, but here is one...
- We get upper bounds on z_{MINLP} from feasible solutions

Back to picture: $S \subseteq T$

- If x_T^* is an optimal solution to $\min f(x) : x \in T$
- And $x_T^* \in S$, then
- x_T^* is an optimal solution to $\min f(x) : x \in S$

Algorithms ↔ Relaxations

Feasible Region "S"

$$\{x \in X \mid g_j(x) \le 0 \ \forall j = 1, \ldots, m, x_I \in \mathbb{Z}^{|I|}\}$$

Algorithms $\leftrightarrow$ Relaxations

Feasible Region "S"

$$\{x \in X \mid g_j(x) \leq 0 \ \forall j = 1, \ldots, m, x_I \in \mathbb{Z}^{|I|}\}$$

NLP-Based Branch and Bound Relaxation

$$\text{“}T\text{''} = \{x \in X \mid g_j(x) \leq 0 \ \forall j = 1, \ldots, m\}$$

Algorithms $\leftrightarrow$ Relaxations

Feasible Region "S"

$$\{x \in X \mid g_j(x) \leq 0 \ \forall j = 1, \ldots, m, x_I \in \mathbb{Z}^{|I|}\}$$

NLP-Based Branch and Bound Relaxation

$$\text{“}T\text{''} = \{x \in X \mid g_j(x) \leq 0 \ \forall j = 1, \ldots, m\}$$

Linearization-based methods

- Outer-Approximation, Extended Cutting Plane, LP/NLP-Branch-and-Bound
- Assume objective function is linear
- Create polyhedral relaxation P such that $S \subseteq P$

Nonconvex Instances

- Must create a tractable (convex) relaxation both the integrality, and the functional non-convexities

---

Two Steps

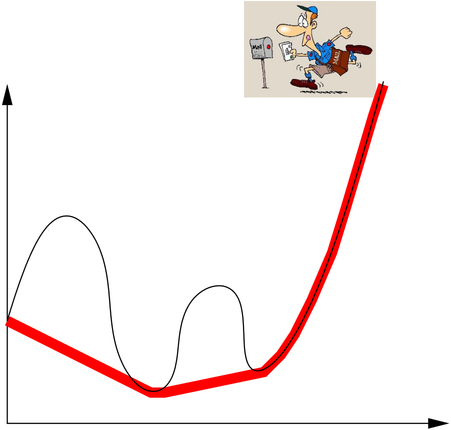
- Break the (factorable) non-convexities into "simple pieces"
- Individually convexify simple pieces using convex/concave envelopes

Envelopes

- Convex and concave envelopes.
  $f : \Omega \rightarrow \mathbb{R}$

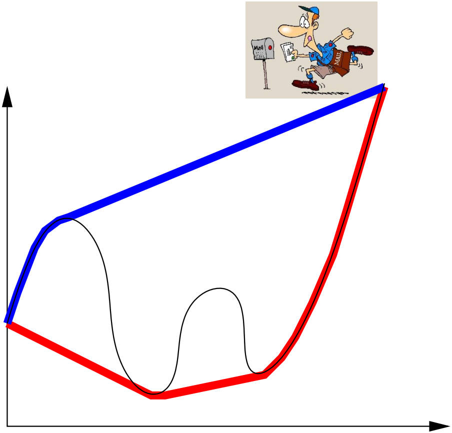
Envelopes

- Convex and concave envelopes. $f : \Omega \rightarrow \mathbb{R}$
- Convex Envelope ($\text{vex}_\Omega(f)$): Pointwise supremum of convex underestimators of f over Ω.



Envelopes

- Convex and concave envelopes. $f : \Omega \to \mathbb{R}$
- Convex Envelope ($\text{vex}_\Omega(f)$): Pointwise supremum of convex underestimators of f over Ω.
- Concave Envelope ($\text{cav}_\Omega(f)$): Pointwise infimum of concave overestimators of f over Ω.



Example

Nonconvex Instance

$$\min c^T x$$
$$\text{s.t. } x_1 x_2^2 + x_2^3 x_3 x_4^2 + x_4^2 x_5 \le 1$$
$$0 \le x \le 1$$

Reformulation

$$\min c^T x$$
$$\text{s.t. } y_1 + y_2 + y_3 \le 1$$
$$0 \le x \le 1$$
$$z_1 = x_2^2 \quad z_2 = x_2^3 \quad z_3 = x_4^2$$
$$y_1 = x_1 z_1 \quad y_2 = z_2 x_3 z_3$$
$$y_3 = z_3 x_5$$

Example

Nonconvex Instance

$$\min c^T x$$
$$\text{s.t. } x_1 x_2^2 + x_2^3 x_3 x_4^2 + x_4^2 x_5 \leq 1$$
$$0 \leq x \leq 1$$

Reformulation

$$\min c^T x$$
$$\text{s.t. } y_1 + y_2 + y_3 \leq 1$$
$$0 \leq x \leq 1$$
$$z_1 = x_2^2 \quad z_2 = x_2^3 \quad z_3 = x_4^2$$
$$y_1 = x_1 z_1 \quad y_2 = z_2 x_3 z_3$$
$$y_3 = z_3 x_5$$

- Global Optimization Solvers perform this reformulation

Example

Nonconvex Instance

$$\min c^T x$$
$$\text{s.t.} \quad x_1 x_2^2 + x_2^3 x_3 x_4^2 + x_4^2 x_5 \leq 1$$
$$0 \leq x \leq 1$$

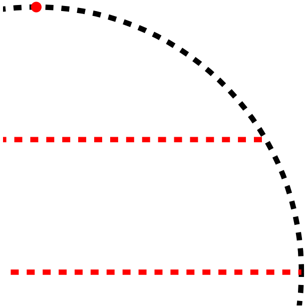
Reformulation

$$\min c^T x$$
$$\text{s.t.} \quad y_1 + y_2 + y_3 \leq 1$$
$$0 \leq x \leq 1$$
$$z_1 = x_2^2 \quad z_2 = x_2^3 \quad z_3 = x_4^2$$
$$y_1 = x_1 z_1 \quad y_2 = z_2 x_3 z_3$$
$$y_3 = z_3 x_5$$

- Global Optimization Solvers perform this reformulation
- They have handlers that enforce equality of the univariate functions via relaxations and branching.
- I will show an example later on

Convex MINLP: "Natural" Relaxation

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

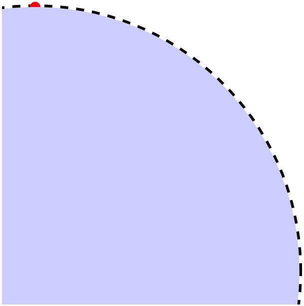


$$\min_{x,z \in \mathbb{R}_+ \times \mathbb{Z}_+} \{-x - z \mid x^2 + z^2 \leq 4\}$$

$$\begin{aligned} z_{\text{MINLP}} = \min \quad & f(x) \\ \text{subject to} \quad & g_j(x) \leq 0, \; j = 1, \ldots, m \\ & x \in X, \quad x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

Convex MINLP: "Natural" Relaxation

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

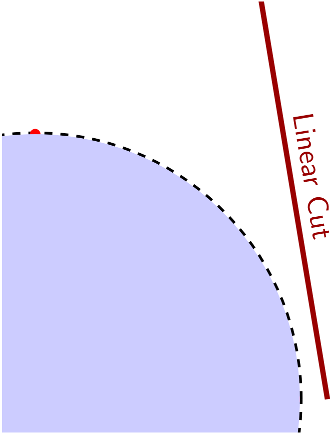


$$\min_{x,z \in \mathbb{R}_+ \times \mathbb{R}_+} \{-x - z \mid x^2 + z^2 \leq 4\}$$

$$\begin{aligned} z_{\text{MINLP}} = \min \quad & f(x) \\ \text{subject to} \quad & g_j(x) \leq 0, \; j = 1, \ldots, m \\ & x \in X, \quad x_I \in \mathbb{R}^{|I|} \end{aligned}$$

Convex MINLP: "Natural" Relaxation

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

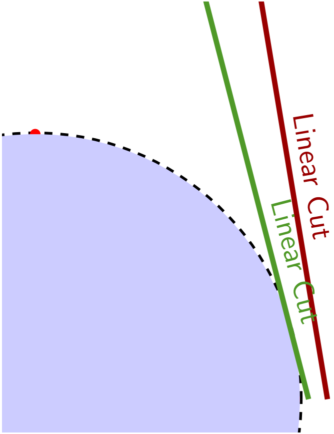


$$\min_{x,z \in \mathbb{R}_+ \times \mathbb{R}_+} \{-x - z \mid x^2 + z^2 \leq 4\}$$

$$\begin{aligned} z_{\text{MINLP}} = \min \quad & f(x) \\ \text{subject to} \quad & g_j(x) \leq 0, \; j = 1, \ldots, m \\ & x \in X, \quad x_I \in \mathbb{R}^{|I|} \end{aligned}$$

Convex MINLP: "Natural" Relaxation

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

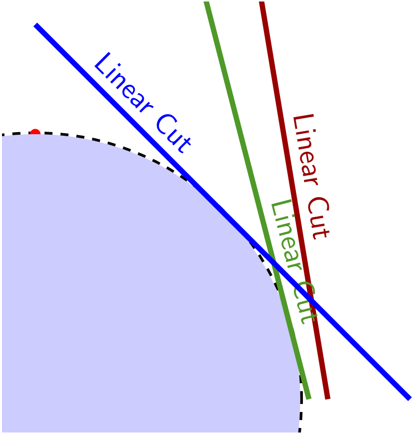


$$\min_{x,z \in \mathbb{R}_+ \times \mathbb{R}_+} \{-x - z \mid x^2 + z^2 \leq 4\}$$

$$\begin{aligned} z_{\text{MINLP}} = \min \quad & f(x) \\ \text{subject to} \quad & g_j(x) \leq 0, \; j = 1, \ldots, m \\ & x \in X, \quad x_I \in \mathbb{R}^{|I|} \end{aligned}$$

Convex MINLP: “Natural” Relaxation

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



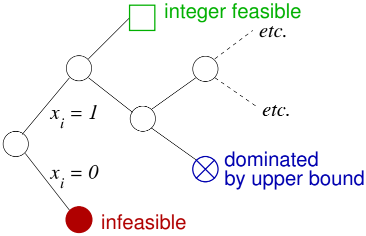
$$\min_{x,z \in \mathbb{R}_+ \times \mathbb{R}_+} \{-x - z \mid x^2 + z^2 \leq 4\}$$

$$\begin{aligned} z_{\text{MINLP}} = \min \quad & f(x) \\ \text{subject to} \quad & g_j(x) \leq 0, \; j = 1, \ldots, m \\ & x \in X, \quad x_I \in \mathbb{R}^{|I|} \end{aligned}$$

NLP-Based Branch and Bound

Solve relaxed NLP ($0 \leq x_i \leq 1$ continuous relaxation)
...solution value provides lower bound

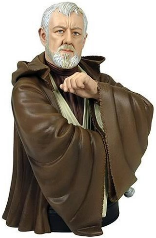
- Branch on $x_i, i \in I$ non-integral
- Solve NLPs & branch until
  1. Node infeasible ... ●
  2. Node integer feasible ... □
     $\Rightarrow$ get upper bound (U)
  3. Lower bound $\geq U$ ... ⊗



Search until no unexplored nodes on tree

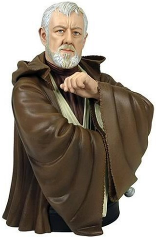
A Long Time Ago, At An Argonne Far, Far Away

"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"

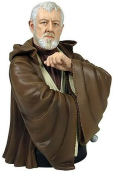
- Of the four algorithms I implemented for my thesis, best this was
- Good implementation exists it does not

ALGORITHMS

Kumar "Luke" Abhishek

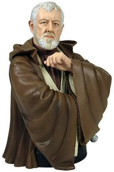
Sven "Yodi" Leyffer

Jeff "Obi-Wan" Linderoth



Mustafa "Anakin" Kılınç

ALGORITHMS

Kumar "Luke" Abhishek	Sven "Yodi" Leyffer	Jeff "Obi-Wan" Linderoth	Mustafa "Anakin" Kılınç
			

- I'll explain a bit of our background in building a linearization-based solver for convex MINLP

Fixed NLP Subproblem

- The QG algorithm solves nonlinear programs with the integer variables x_I fixed to specific values

Fixed NLP Subproblem

- The QG algorithm solves nonlinear programs with the integer variables x_I fixed to specific values

$\mathrm{NLP}(x^k)$

$$
\begin{aligned}
z_{\mathrm{NLP}(x_I^k)} = \min \quad & f(x) \\
\text{subject to} \quad & g_j(x) \le 0 \quad \forall j \\
& x_I = x_I^k \\
& x \in X
\end{aligned}
$$

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

$$
\begin{cases}
f(x^k) + \nabla f(x^k)^T (x - x^k) \le \eta \\
g_j(x^k) + \nabla g_j(x^k)^T (x - x^k) \le 0
\end{cases}
$$

Fixed NLP Subproblem

- The QG algorithm solves nonlinear programs with the integer variables x_I fixed to specific values

NLP(x^k)

$$
\begin{aligned}
z_{\mathrm{NLP}(x_I^k)} = \min \quad & f(x) \\
\text{subject to} \quad & g_j(x) \le 0 \quad \forall j \\
& x_I = x_I^k \\
& x \in X
\end{aligned}
$$

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

$$
\begin{cases}
f(x^k) + \nabla f(x^k)^T (x - x^k) \le \eta \\
g_j(x^k) + \nabla g_j(x^k)^T (x - x^k) \le 0
\end{cases}
$$

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

The Master

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

$$
\begin{aligned}
z_{\mathrm{MP}(\mathcal{K})} = \min \quad & \eta \\
\text{subject to} \quad & f(x^k) + \nabla f(x^k)^T (x - x^k) \leq \eta \;\; \forall (x^k) \in \mathcal{K} \qquad (\mathrm{MP}(\mathcal{K})) \\
& g_j(x^k) + \nabla g_j(x^k)^T (x - x^k) \leq 0 \;\; \forall (x^k) \in \mathcal{K} \; \forall j \\
& x \in X, \quad x_I \in \mathbb{Z}^{|I|}
\end{aligned}
$$

The Master

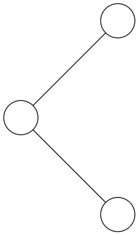
MP($\mathcal{K}$): Outer-Approximation MILP Master Problem

$$
\begin{aligned}
z_{\mathrm{MP}(\mathcal{K})} = \min \quad & \eta \\
\text{subject to} \quad & f(x^k) + \nabla f(x^k)^T (x - x^k) \le \eta \;\; \forall (x^k) \in \mathcal{K} \qquad (\mathrm{MP}(\mathcal{K})) \\
& g_j(x^k) + \nabla g_j(x^k)^T (x - x^k) \le 0 \;\forall (x^k) \in \mathcal{K} \;\forall j \\
& x \in X, \quad x_I \in \mathbb{Z}^{|I|}
\end{aligned}
$$

- **Thm:** $z_{\mathrm{MP}(\mathcal{K})} \le z_{\mathrm{MINLP}}$
- **Thm:** If $\mathcal{K}$ contains the “right” points, then $z_{\mathrm{MINLP}} = z_{\mathrm{MP}(\mathcal{K})}$

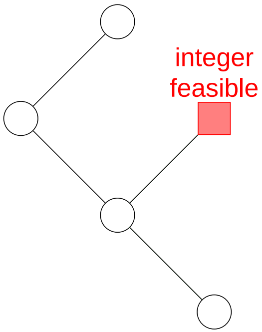
LP/NLP-BB (Quesada-Grossmann)

- Start solving Master MILP ($\mathcal{MP}(\mathcal{K})$) ... using MILP branch-and-cut.



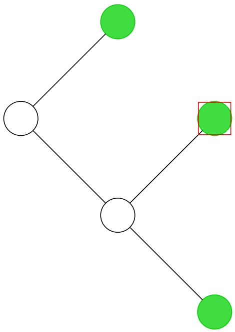
LP/NLP-BB (Quesada-Grossmann)

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



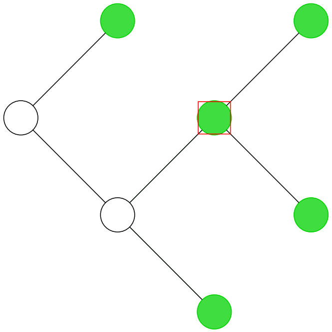
LP/NLP-BB (Quesada-Grossmann)

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



LP/NLP-BB (Quesada-Grossmann)

- Start solving Master MILP ($MP(\mathcal{K})$) ... using MILP branch-and-cut.
- If $x_I^k \in \mathbb{Z}_+^{|I|}$, then interrupt MILP. Solve NLP(x_I^k) get x^k
- linearize f, g_j about (x^k)
  $\Rightarrow$ 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.

RBA–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.

RBA–Rebels are Bad at Acronyms

FilMINT = Filter + MINTO

Why FilMINT Could Be Good

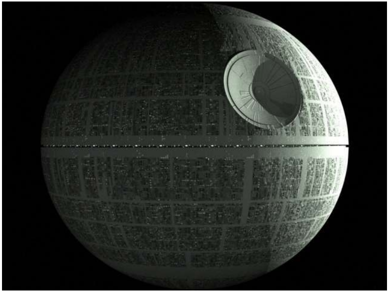
1. Use MINTO's advanced MIP Features "for free."
2. Really the only LP/NLP-BB algorithm available

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."

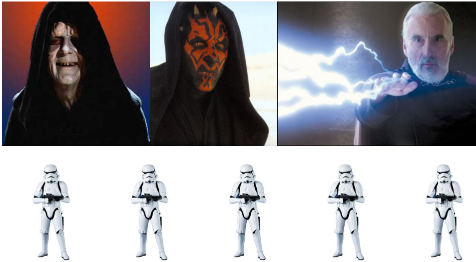


The Galactic Empire

Pierre Bonami, Larry Biegler, Andy Conn, Gérard Cornuéjols, Ignacio Grossmann, Carl Laird, Jon Lee, Andrea Lodi, François Margot, Nick Sawaya and Andreas Wächter, "An algorithmic framework for convex mixed integer nonlinear programs," *Discrete Optimization*, Volume 5, May 2008, Pages 186-204, 2008.

Battling the Empire

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



Variations on a Theme

- Instead of doing a "one-tree" approach, we could solve a sequence of integer programs and nonlinear programs
- This algorithm is known as outer-approximation

Variations on a Theme

- Instead of doing a "one-tree" approach, we could solve a sequence of integer programs and nonlinear programs
- This algorithm is known as outer-approximation

Outer-Approximation

1. Solve (MILP) MP($\mathcal{K}$), giving solution x^k. Solution gives lower bound.
2. Solve (NLP) NLP(x^k), giving solution y^k. If feasible, $f(y^k)$ gives an upper bound
3. Add y^k to linearization points $\mathcal{K}$, and Go To 1 if not yet converged.

Variations on a Theme

- Instead of doing a "one-tree" approach, we could solve a sequence of integer programs and nonlinear programs
- This algorithm is known as outer-approximation

Outer-Approximation

1. Solve (MILP) MP($\mathcal{K}$), giving solution x^k. Solution gives lower bound.
2. Solve (NLP) NLP(x^k), giving solution y^k. If feasible, $f(y^k)$ gives an upper bound
3. Add y^k to linearization points $\mathcal{K}$, and Go To 1 if not yet converged.

Outer-Approximation Solvers

- AIMMS, Bonmin, DICOPT
- Just "MIP" it—Use MILP Solver as black box

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

Instance Family	NL Ob?	# of ins	Average Var	Bin	LC	NLC
Batch	√	10	334.6	123	1089.1	1
CLay		12	116.7	35.3	138.3	40
FLay		10	158.0	28	183	4
fo-m-o		9	112.2	41.6	194.3	13.6
nd		5	574.0	37.6	283.8	37.6
RSyn		48	922.3	251	1716.3	34.2
safetyLay		3	120.7	38	111	34.7
SLay	√	14	336.0	92	437	0
sssd		14	162.4	135.5	50	20.1
Syn		48	366.3	95	660	34.2
trimloss		12	279.2	227.5	133.3	6
uflquad	√	10	1571.0	23.5	1613	0
others	√	12	205.4	86.4	206	3.3
Total		207	487.5	127.6	767.9	22.3

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

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).

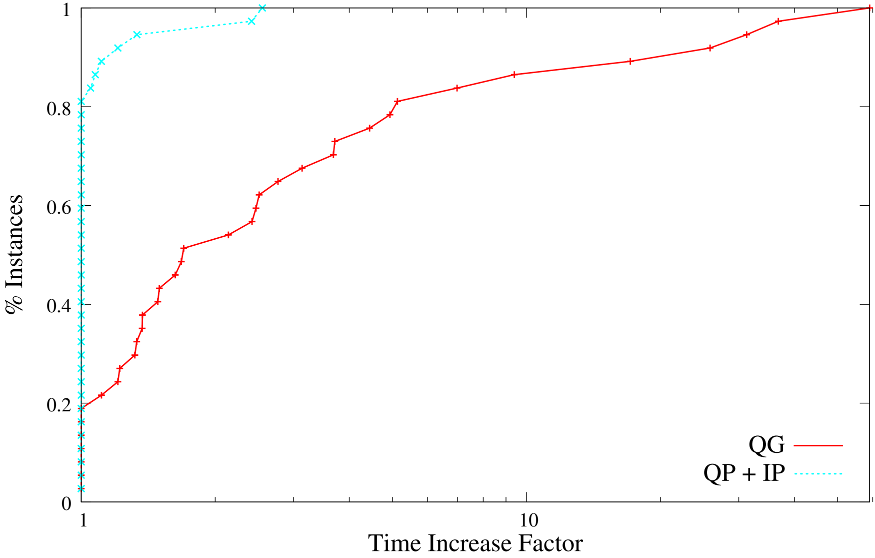
MINTO v3.1 MILP Features

- 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).

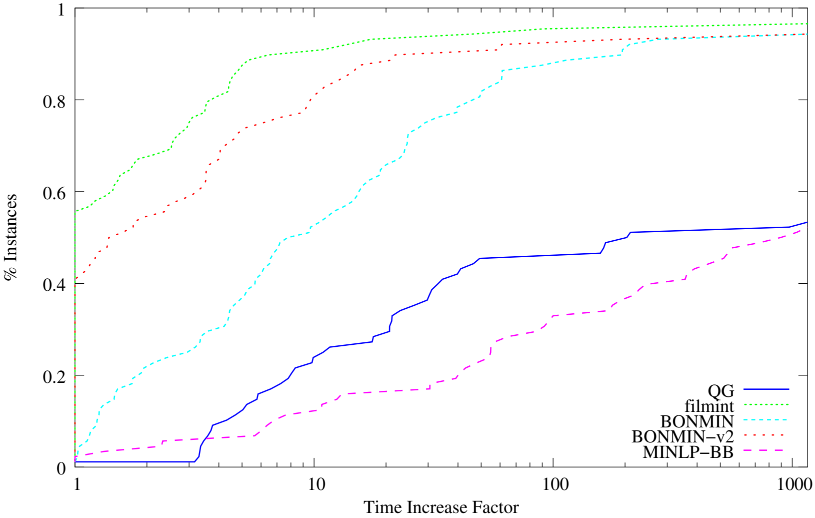
The MILP Force

- Do fancy-pants MILP techniques make a difference for the LP/NLP-BB (QG) Algorithm?

YES! Performance Profile: Moderate Instances



(Old) Comparison of Solvers



Nonconvex Instances

- I will explain the basic ideas using the nonconvex quadratic program (QCQP) as a specific example:

$$\text{QCQP}\begin{cases} \min_{x\in\mathbb{R}^n} & q_0(x) \\ \text{s.t.} & q_k(x) \le b_k \qquad \forall i = k \in \mathcal{M} \\ & l \le x \le u \end{cases}$$

- $q_k(x) = (c^k)^T x + x^T Q^k x \qquad \forall k \in \{0 \cup \ \mathcal{M}$
- $q_k(x)$ could be convex, concave, or nonconvex
- l and u are finite

Solving QCQP

- Convexify "simple" nonconvex term $x_i x_j$ over the region $(x_i, x_j) \in R \stackrel{\text{def}}{=} [l_i, u_i] \times [l_j, u_j]$.

$$\begin{aligned} x_i x_j &\geq \max\{l_i x_j + l_j x_i - l_i l_j, u_i x_j + u_j x_i - u_i u_j\} \\ x_i x_j &\leq \min\{l_i x_j + u_j x_i - l_i u_j, u_i x_j + l_j x_i - u_i l_j\} \end{aligned}$$

Solving QCQP

- Convexify "simple" nonconvex term $x_i x_j$ over the region $(x_i, x_j) \in R \stackrel{\text{def}}{=} [l_i, u_i] \times [l_j, u_j]$.

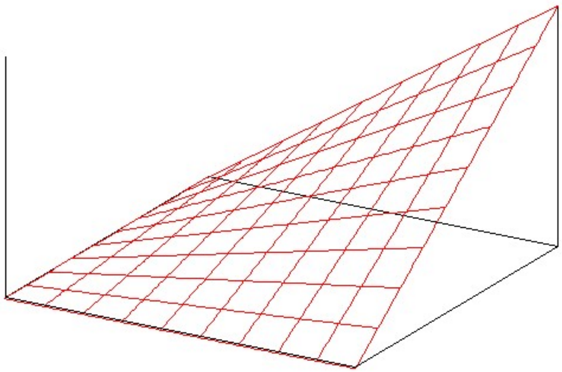
$$\begin{aligned} x_i x_j &\geq \max\{l_i x_j + l_j x_i - l_i l_j, u_i x_j + u_j x_i - u_i u_j\} \\ x_i x_j &\leq \min\{l_i x_j + u_j x_i - l_i u_j, u_i x_j + l_j x_i - u_i l_j\} \end{aligned}$$

- Thm: (McCormick '76, Al-Khayyal and Falk, '83)

$$\begin{aligned} \text{vex}_R(x_i x_j) &= \max\{l_i x_j + l_j x_i - l_i l_j, u_i x_j + u_j x_i - u_i u_j\} \\ \text{cav}_R(x_i x_j) &= \min\{l_i x_j + u_j x_i - l_i u_j, u_i x_j + l_j x_i - u_i l_j\} \end{aligned}$$

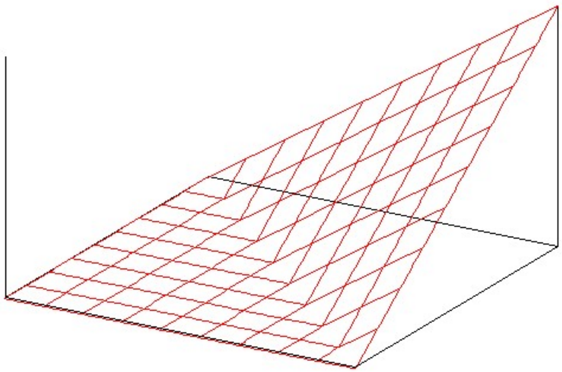
Worth 1000 Words?

$$x_i x_j$$



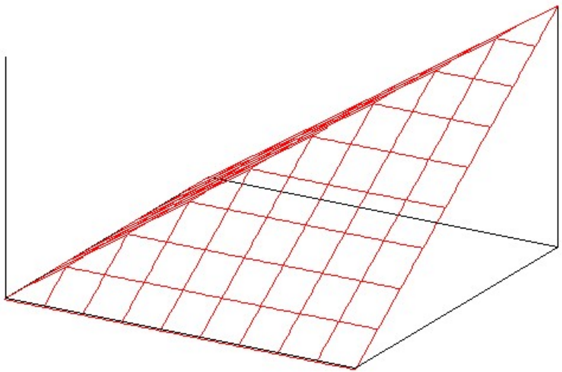
Worth 1000 Words?

$$\text{vex}_R(x_i x_j)$$



Worth 1000 Words?

$$\mathrm{cav}_R(x_i x_j)$$



(LP) Relaxation of QCQP

$$z_{LP} = \min \sum_{i=1}^{n} c_i^0 x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij}^0 z_{ij}$$

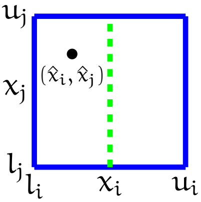
subject to

$$\begin{array}{rcll}
\sum_{i=1}^{n} c_i^k x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij}^k z_{ij} & \leq & b_k & \forall k \in \mathcal{M} \\
z_{ij} - l_i x_j - l_j x_i + l_i l_j & \geq & 0 & \forall i = 1, \dots, n, j = 1, \dots, n \\
z_{ij} - u_i x_j - u_j x_i + u_i u_j & \geq & 0 & \forall i = 1, \dots, n, j = 1, \dots, n \\
z_{ij} - l_i x_j - u_j x_i + l_i u_j & \leq & 0 & \forall i = 1, \dots, n, j = 1, \dots, n \\
z_{ij} - u_i x_j - l_j x_i + u_i l_j & \leq & 0 & \forall i = 1, \dots, n, j = 1, \dots, n \\
x_i & \in & [l_i, u_i] & \forall i = 1, \dots, n
\end{array}$$

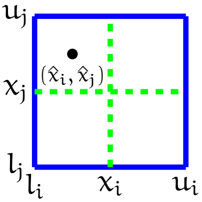
Branching

- In LP relaxation, $z_{ij} = x_i x_j \ \ \forall x_i, x_j$ on the boundary of the rectangular region R_{ij}
- If $z_{ij} \neq x_i x_j$, we branch. Two suggested branching schemes

Two Rectangles



Four Rectangles



Tight Bounds are Important!

Hock and Schittkowski

minimize

$$x_3 + x_1x_5 + x_2x_5 + x_3x_5$$

subject to

$$\begin{aligned} x_5 - x_1x_4 &= 0 \\ x_6 - x_2x_3 &= 0 \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 &= 40 \\ x_5x_6 &\geq 25 \\ 1 \leq x_k &\leq K \qquad k = 1, 2, 3, 4 \end{aligned}$$

Tight Bounds are Important!

Hock and Schittkowski

minimize

$$x_3 + x_1x_5 + x_2x_5 + x_3x_5$$

subject to

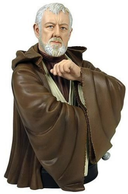
$$\begin{aligned} x_5 - x_1x_4 &= 0 \\ x_6 - x_2x_3 &= 0 \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 &= 40 \\ x_5x_6 &\geq 25 \\ 1 \leq x_k &\leq K \qquad k = 1, 2, 3, 4 \end{aligned}$$

K	**Nodes**
5	210
10	788
25	6834
50	19360
100	> 70000

PREPROCESSING

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

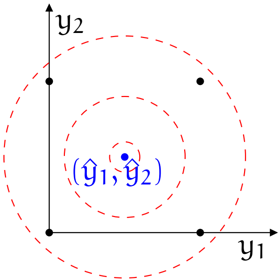
---

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

Linear Objective Is Important Here!

$$\min (y_1 - 1/2)^2 + (y_2 - 1/2)^2$$

$$\text{s.t. } y_1 \in \{0, 1\}, y_2 \in \{0, 1\}$$

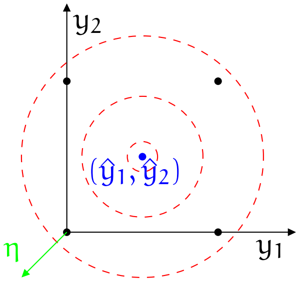


Linear Objective Is Important Here!

$$\min (y_1 - 1/2)^2 + (y_2 - 1/2)^2$$

$$\text{s.t. } y_1 \in \{0,1\}, y_2 \in \{0,1\}$$

$$\eta \geq (y_1 - 1/2)^2 + (y_2 - 1/2)^2$$



- Without linear objective, optimal solution may be interior to the convex hull $\Rightarrow$ convexifying may do you no good!

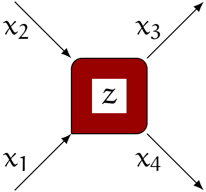
Indicator MINLPs

- 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

---

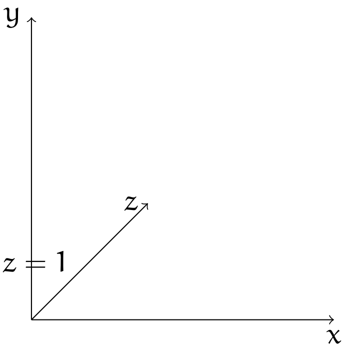
Process Flow Applications

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



A Very Simple Example

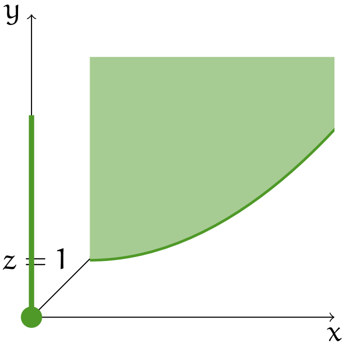
$$R \stackrel{\text{def}}{=} \left\{(x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz\right\}$$



A Very Simple Example

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

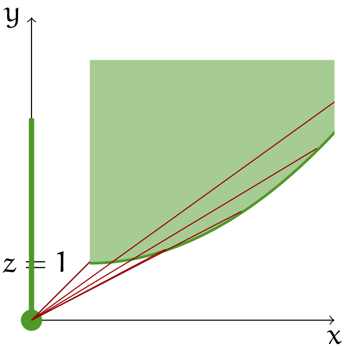
- $z = 0 \Rightarrow x = 0, y \geq 0$
- $z = 1 \Rightarrow x \leq u, y \geq x^2$



A Very Simple Example

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

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



Deep Insights

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

Characterization of Convex Hull

Deep Theorem #1

$$\begin{aligned} R &= \left\{(x, y, z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz\right\} \\ \mathrm{conv}(R) &= \left\{(x, y, z) \in \mathbb{R}^3 \mid yz \geq x^2, 0 \leq x \leq uz, 0 \leq z \leq 1, y \geq 0\right\} \end{aligned}$$

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

$$R = \left\{(x,y,z) \in \mathbb{R}^2 \times \mathbb{B} \mid y \geq x^2, 0 \leq x \leq uz\right\}$$
$$\operatorname{conv}(R) = \left\{(x,y,z) \in \mathbb{R}^3 \mid yz \geq x^2, 0 \leq x \leq uz, 0 \leq z \leq 1, y \geq 0\right\}$$

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

Second Order Cone Programming

- $x^2 - yz$ is not convex
- There are effective, robust algorithms for optimizing over $\operatorname{conv}(R)$

Giving You Some Perspective

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

$$\mathcal{P}(x, z) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } z = 0 \\ zf(x/z) & \text{if } z > 0 \end{cases}$$

- The epigraph of $\mathcal{P}(x, z)$ 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

$$\mathcal{P}(x, z) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } z = 0 \\ zf(x/z) & \text{if } z > 0 \end{cases}$$

- 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:

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

- 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: Death Stars
- N: Rebel Bases
- x_{ij}: percentage of rebel base $j \in N$ blown up by death star $i \in M$
- $z_i = 1 \Leftrightarrow$ death star $i \in M$ is built
- Fixed cost for opening death star $i \in M$
- Quadratic cost for blowing up base $j \in N$ from death star $i \in M$

Death Star Location Formulation

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

subject to

$$\begin{aligned} x_{ij} &\leq z_i & \forall i \in M, \forall j \in N \\ \sum_{i \in M} x_{ij} &= 1 & \forall j \in N \\ x_{ij} &\geq 0 & \forall i \in M, \forall j \in N \\ z_i &\in \{0, 1\} & \forall i \in M \end{aligned}$$

Death Star Location Formulation

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

subject to

$$\begin{aligned}
x_{ij} &\leq z_i && \forall i \in M, \forall j \in N \\
\sum_{i \in M} x_{ij} &= 1 && \forall j \in N \\
x_{ij} &\geq 0 && \forall i \in M, \forall j \in N \\
z_i &\in \{0, 1\} && \forall i \in M \\
x_{ij}^2 - y_{ij} &\leq 0 && \forall i \in M, \forall j \in N
\end{aligned}$$

Death Star Location Formulation

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

subject to

$$\begin{aligned} x_{ij} &\leq z_i && \forall i \in M, \forall j \in N \\ \sum_{i \in M} x_{ij} &= 1 && \forall j \in N \\ x_{ij} &\geq 0 && \forall i \in M, \forall j \in N \\ z_i &\in \{0, 1\} && \forall i \in M \\ x_{ij}^2 - z_i y_{ij} &\leq 0 && \forall i \in M, \forall j \in N \end{aligned}$$

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

$\|M\|$	$\|N\|$	z_R	z_{GLW}	z_P	z^*
10	30	140.6			348.7
15	50	141.3			384.1
20	65	122.5			289.3
25	80	121.3			315.8
30	100	128.0			393.2

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

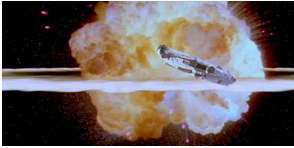
$\|M\|$	$\|N\|$	z_R	z_{GLW}	z_P	z^*
10	30	140.6	326.4		348.7
15	50	141.3	312.2		384.1
20	65	122.5	248.7		289.3
25	80	121.3	260.1		315.8
30	100	128.0	327.0		393.2

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

$\lvert M \rvert$	$\lvert N \rvert$	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

Woo Hoo!



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

m	n	T	N
30	200	141.9	63
40	100	76.4	54
40	200	101.3	45
50	100	61.6	49
50	200	140.4	47

"The Force is Strong with This One"

Conclusions

- My 10-year old likes Star Wars

Conclusions

- My 10-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 10-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
- http://wiki.mcs.anl.gov/minotaur/index.php/Main_Page
- Framework & toolbox for solving MINLPs
- Implemented algorithms in MINOTAUR:
  - branch-and-{bound|cut} for convex MINLPs
  - branch-and-bound for mixed polynomial programs
- Extensible: implement new MINLP algorithms, solvers

Bunch of refs

- I am including a number of classical and recent reference on MINLP
- The list is by no means comprehensive
- Don't yell at me if your favorite reference is not there!

- K. Abhishek, S. Leyffer, and J. T. Linderoth. FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs. *INFORMS Journal on Computing*, 22:555–567, 2010.
- Alper Atamtürk and Vishnu Narayanan. Conic mixed integer rounding cuts. *Mathematical Programming*, 122:1–20, 2010.
- P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24:597–634, 2009.
- P. Bonami. Lift-and-project cuts for mixed integer convex programs. In O. Günlük and G. Woeginger, editors, *Integer Programming and Combinatoral Optimization*.
- P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5:186–204, 2008.
- P. Bonami, M. Kılınç, and J. Linderoth. Algorithms and software for solving convex mixed integer nonlinear programs. In *IMA Volumes on MINLP*, 2011. To appear.
- S. Drewes. *Mixed Integer Second Order Cone Programming*. PhD thesis, Technische Universität Darmstadt, 2009.
- M. A. Duran and I. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.
- R. Fletcher and S. Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66:327–349, 1994.

I. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, pages 83–100, 2003.

I. E. Grossmann. Review of nonlinear mixed–integer and disjunctive programming techniques. *Optimization and Engineering*, 3:227–252, 2002.

O. Günlük and J. Linderoth. Perspective relaxation of mixed integer nonlinear programs with indicator variables. *Mathematical Programming Series B*, 104:186–203, 2010.

O. Günlük and J. T. Linderoth. Perspective reformulation and applications. In *IMA Volumes*, volume 154, pages 61–92, 2012.

M. Kılınç, J. Linderoth, and J. Luedtke. Effective separation of disjunctive cuts for convex mixed integer nonlinear programs. Technical Report 1681, Computer Sciences Department, University of Wisconsin-Madison, 2010.

M. Kılınç, J. Linderoth, J. Luedtke, and A. Miller. Strong branching inequalities for convex mixed integer nonlinear programs. Technical Report 1696, Computer Sciences Department, University of Wisconsin-Madison, 2011.

S. Leyffer. *Deterministic Methods for Mixed Integer Nonlinear Programming*. PhD thesis, University of Dundee, Dundee, Scotland, UK, 1993.

S. Leyffer. Generalized outer approximation. In C. Floudas and Pardalos P, editors, *Encyclopedia of Optimization*. Kluwer, 1997. to appear, `http://www.mcs.dundee.ac.uk:8080/ sleyffer/index.html`.

S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Comput. Optim. Appl.*, 18(3):295–309, 2001.

- R. Misener and C. Floudas. Glomiqo: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 2012. to appear.
- I. Quesada and I. E. Grossmann. An LP/NLP based branch–and–bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16:937–947, 1992.
- M. Tawarmalani and N. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.
- M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.
- J. P. Vielma, S. Ahmed, and G. L. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed integer conic quadratic programs. *INFORMS Journal on Computing*, 2008. To appear.