Mixed Integer Nonlinear Programming

IMA New Directions Short Course on Mathematical Optimization
Jeff Linderoth and Jim Luedtke

Department of Industrial and Systems Engineering
University of Wisconsin-Madison

August 12, 2016
Mixed-Integer Nonlinear Optimization

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

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

- \( f : \mathbb{R}^n \to \mathbb{R}, \ g : \mathbb{R}^n \to \mathbb{R}^m \) smooth, sometimes convex functions.
- \( X \in \mathbb{R}^n \) bounded, polyhedral set, e.g. \( X = \{x : l \leq A^T x \leq u\} \)
- \( I \subseteq \{1, \ldots, n\} \) subset of integer variables
- We may assume that \( f \) is a *linear* function.

**NP-Super-Hard**

- Combines challenges of handling nonlinearities with combinatorial explosion of integer variables
The MINLP Family

The great watershed in optimization isn’t between linearity and nonlinearity, but convexity and nonconvexity.

- R. Tyrrell Rockafellar

- If $f$ and $g$ are convex functions, then we have a convex MINLP
- If $f$ and $g$ are not convex, then we have a nonconvex MINLP

Optimization 101

- Know the convexity properties of your instance!
- Algorithms for two different classes of problems are different
- We can solve significantly larger convex MINLP instances than nonconvex MINLP instances
- Algorithms designed for convex-MINLP are often used as heuristics for nonconvex MINLPs
Specializations

**Functional Form**

<table>
<thead>
<tr>
<th>$g_i(x)$</th>
<th><strong>Problem Type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_i(x) = |A_i x + b_i|_2 - p_i^T x + q_i$</td>
<td>MISOCP</td>
</tr>
<tr>
<td>$g_i(x) = \text{polynomial}$</td>
<td>MIPP</td>
</tr>
<tr>
<td>$g_i(x) = x^T Q_i x + p_i^T + q_i$</td>
<td>MIQP</td>
</tr>
<tr>
<td>$g_i(x) = a_i^T x - b_i$</td>
<td>MILP</td>
</tr>
</tbody>
</table>

- **MISOCMP**: Mixed Integer Second Order Cone Program
  - Are convex-MINLP
- **MIPP**: Mixed Integer Polynomial Program
- **MIQP**: Mixed Integer Quadratic Program
  - May be convex or nonconvex
  - Convex MIQP is a special case of MISOCMP
  - If $f$ is convex quadratic and $c$ is an affine mapping, then there are specialized algorithms for convex-MIQP
- **MILP**: Mixed Integer Linear Program
Aside – MISOCPP

- The feasible regions of surprisingly many MINLP problems can be expressed as MISOCPP:

- Hyperbolic constraints: Product of (non-negative) variables $\geq$ a norm$^2$.
  \[
  w^T w \leq xy, \ x \geq 0, \ y \geq 0 \iff \left\| \begin{bmatrix} 2w \\ x - y \end{bmatrix} \right\| \leq x + y
  \]

- Convex Quadratic: $c_i(x) = x^T Q_i x + p_i^T x + q_i \leq t$
  - $Q_i$ is positive semidefinite, so it has Cholesky factorization $Q_i = L^T L$
  
  \[
  y = Lx \Rightarrow g(x) = y^T y + p_i^T x + q_i \leq t
  \]

\[
\{(x, t) \in \mathbb{R}^{n+1} \mid x^T Qx + p_i^T x + q_i \leq t\} = 
\]

\[
\text{Proj}_{(x,t)}\{(x, y, t, p) \in \mathbb{R}^{2n+2} \mid y = Lx, \ p + q^T x + r \leq t, \ p \geq \|y\|^2\}
\]
# How Big Is It?

The Leyffer-Linderoth-Luedtke (LLL) Measure of Complexity

You have a problem of class \( X \) that has \( Y \) decision variables? What is the largest value of \( Y \) for which one of Sven, Jim, and Jeff would be willing to bet $50 that a “state-of-the-art” solver could solve the problem?

<table>
<thead>
<tr>
<th>Convex Problem Class ((X))</th>
<th># Var ((Y))</th>
<th>Nonconvex Problem Class ((X))</th>
<th># Var ((Y))</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINLP</td>
<td>500</td>
<td>MINLP</td>
<td>100</td>
</tr>
<tr>
<td>NLP</td>
<td>(5 \times 10^4)</td>
<td>NLP</td>
<td>100</td>
</tr>
<tr>
<td>MISOCp</td>
<td>1000</td>
<td>MIPPP</td>
<td>150</td>
</tr>
<tr>
<td>SOCP</td>
<td>(10^5)</td>
<td>PP</td>
<td>150</td>
</tr>
<tr>
<td>MIQP</td>
<td>1000</td>
<td>MIQP</td>
<td>300</td>
</tr>
<tr>
<td>QP</td>
<td>(5 \times 10^5)</td>
<td>QP</td>
<td>300</td>
</tr>
<tr>
<td>MILP</td>
<td>(2 \times 10^4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>(5 \times 10^7)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Solvers for MINLP

<table>
<thead>
<tr>
<th>Convex</th>
<th>Nonconvex</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$-ECP</td>
<td>$\alpha$-BB</td>
</tr>
<tr>
<td>BONMIN</td>
<td>ANTIGONE</td>
</tr>
<tr>
<td>DICOPT</td>
<td>BARON</td>
</tr>
<tr>
<td>FilmMINT</td>
<td>COCONUT</td>
</tr>
<tr>
<td>GUROBI†</td>
<td>COUENNE</td>
</tr>
<tr>
<td>KNITRO</td>
<td>CPLEX‡</td>
</tr>
<tr>
<td>MILANO</td>
<td>LGO</td>
</tr>
<tr>
<td>MINLPBB</td>
<td>LindoGlobal</td>
</tr>
<tr>
<td>MINOPT</td>
<td>SCIP</td>
</tr>
<tr>
<td>MINOTAUR</td>
<td></td>
</tr>
<tr>
<td>SBB</td>
<td></td>
</tr>
<tr>
<td>Xpress†</td>
<td></td>
</tr>
<tr>
<td>†MIQP</td>
<td>‡MIQP/convex MIQP</td>
</tr>
</tbody>
</table>

**Open-source and commercial solvers**
# 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
**Simple Example**

\[
(x_1 - 2)^2 + (x_2 + 1)^2 \leq 36
\]

\[
3x_1 - x_2 \leq 6
\]

\[
-2.5(x_1 - 1)^2 - x_2 \leq -4
\]

\[
0 \leq x_1, x_2 \leq 5
\]

\[
x_2 \in \mathbb{Z}
\]

**Doubly NP-Hard!**

- Feasible region is not convex for two reasons—Integrality of \(x_2\) and nonconvexity of constraint.
Q: What to Do?!

Relax!

- Remove some constraints.
- Solution of relaxed problem provide a **lower bound** on optimal solution value.
“Natural” relaxation is to remove constraints $x_i \in \mathbb{Z}$ $\forall i \in I$

But this still leaves a nonconvex NLP relaxation
We need a mechanism for relaxing the nonconvex constraints.

Secant underestimator for concave functions

“Easy” to get polyhedral outerapproximation of nonconvex constraints

Ideally, we would like to get the convex hull of the feasible points
Linear Objective Is Important Here!

\[
\min(x_1 - 1/2)^2 + (x_2 - 1/2)^2
\]
\[
\text{s.t. } x_1 \in \{0, 1\}, x_2 \in \{0, 1\}
\]
\[
\eta \geq (x_1 - 1/2)^2 + (x_2 - 1/2)^2
\]

- Without linear objective, optimal solution may be \textit{interior} to the convex hull \Rightarrow convexifying may do you no good!
Algorithm Ingredients

1. Relaxation
   - Used to compute a lower bound on the optimum value
   - Obtained by enlarging feasible set; e.g. ignore constraints
   - Should be “tractable” and “tight”

2. Constraint Enforcement
   - Exclude solutions from relaxations not feasible to MINLP
   - Refine or tighten of relaxation—Branching and Cutting

3. Upper Bounds
   - Obtained from any feasible point; e.g. solve NLP for fixed $x_i$. ($\text{NLP}(x_i)$).
   - Other heuristic techniques have been developed
Models are Important!

- Writing “good” models is extremely important in MILP
- Probably more important for MINLP

More Complexity

- Tradeoff between physical accuracy and mathematical tractability
- Convex reformulations? Linearize expressions?

MINLP Modeling Preference

- We want strong formulations—Relaxations that can be relaxed to something that closely approximates the convex hull
- We prefer linear over convex over nonconvex formulations
Some MINLP Applications

1. Supply Chain—Convex MINLP
   - Discrete: Fixed charges for opening facilities
   - Nonlinear: Nonlinear transportation costs

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

3. Pooling/Petrochemical—Nonconvex MINLP
   - Discrete: Which process to use?
   - Nonlinear: Product Blending (among others)

Luedtke Theorem

- 97.5% of all practical MINLPs have integer variables for one of two purposes...
  1. Binary variables to make a “multiple choice” selection
  2. Binary indicator variables that turn on/off continuous variables and/or constraints.
Uncapacitated Facility Location

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

- $M$: Facility
- $N$: Customer
- $x_{ij}$: percentage of customer $j \in N$ demand met by facility $i \in M$
- $z_i = 1 \iff$ facility $i \in M$ is built
- Fixed cost for opening facility $i \in M$
- **Quadratic** cost for meeting demand $j \in N$ from facility $i \in M$
A very simple MIQP

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

subject to

\[ x_{ij} \leq z_i \quad \forall i \in M, \forall j \in N \]
\[ \sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \]
\[ x_{ij} \geq 0 \quad \forall i \in M, \forall j \in N \]
\[ z_i \in \{0, 1\} \quad \forall i \in M \]
Base Case of Induction to “Luedtke Theorem”

- Note that binary variables are used as indicators: $z_i = 0 \Rightarrow x_{ij} = 0$
- If $z = 1$, then we need to model the epigraph of $x_{ij}^2$

Intager Programmerz ’r’ Smrt

- 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
- Important to understand the structure of a special MINLP with indicator variables
A Very Simple Structure

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

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

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

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

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

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

Second Order Cone Programming

- There are effective, robust algorithms for optimizing over conv(R)
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

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

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

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.
Perspectivizing Quadratic Uncapacitated Facility Location

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

\[ x_{ij} \leq z_i \quad \forall i \in M, \forall j \in N \]

\[ \sum_{i \in M} x_{ij} = 1 \quad \forall j \in N \]

\[ x_{ij} \geq 0 \quad \forall i \in M, \forall j \in N \]

\[ z_i \in \{0, 1\} \quad \forall i \in M \]

\[ x_{ij}^2 - z_i y_{ij} \leq 0 \quad \forall i \in M, \forall j \in N \]

Steps

1. Make Objective Linear
2. Apply Perspective
3. Make Giant Profit
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>326.4</td>
<td>346.5</td>
<td>348.7</td>
</tr>
<tr>
<td>15</td>
<td>50</td>
<td>141.3</td>
<td>312.2</td>
<td>380.0</td>
<td>384.1</td>
</tr>
<tr>
<td>20</td>
<td>65</td>
<td>122.5</td>
<td>248.7</td>
<td>288.9</td>
<td>289.3</td>
</tr>
<tr>
<td>25</td>
<td>80</td>
<td>121.3</td>
<td>260.1</td>
<td>314.8</td>
<td>315.8</td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>128.0</td>
<td>327.0</td>
<td>391.7</td>
<td>393.2</td>
</tr>
</tbody>
</table>

Cheers!
Impact of SOCP

**$m = 30, n = 100$**

- Convex MINLP, Original: 16697 CPU seconds, 45901 nodes
- Convex MINLP, w/GLW Ineq.: 21206 CPU seconds, 29277 nodes
- MISOCO, 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>
Design of Water Distribution Networks

**Goal:** Design minimum cost network from discrete pipe diameters to meet water demand

**Sets**
- \( \mathcal{N} \): nodes in network
- \( \mathcal{S} \): source nodes
- \( \mathcal{A} \): arcs in the network
- \( \mathcal{K} \): Pipe diameters
### Design of Water Distribution Networks

#### Variables

- $q_{ij}$: flow pipe $(i, j) \in A$
- $h_i$: hydraulic head at node $i \in N$
- $d_{ij}$: diameter of pipe $(i, j) \in A$, where $d_{ij} \in \{P_1, \ldots, P_r\}$
- $a_{ij}$: area of cross section $(i, j) \in A$
- $y_{ijk}$: Binary variable $= 1$ if pipe of size $k \in K$ chosen for $(i, j) \in A$

#### Constraints

- **Conservation of flow**

  $$\sum_{(i,j) \in A} q_{ij} - \sum_{(j,i) \in A} q_{ji} = D_i, \ \forall i \in N \setminus S.$$

- **Flow bounds:** (linear in $a$, but since $a_{ij} = 0.25\pi d_{ij}^2$, nonlinear in $d$)

  $$-V_{\text{max}} a_{ij} \leq q_{ij} \leq V_{\text{max}} a_{ij}, \ \forall (i, j) \in A.$$
Design of Water Distribution Networks

- Discrete pipe sizes $d_{ij} \in \{P_1, \ldots, P_r\}$
- Use “multiple choice” integrality to remove nonlinearity $a_{ij} = \pi d_{ij}^2 / 4$
- This is in general a good modeling practice

1. Binary variables $y_{ijk} \in \{0, 1\}$ for $k \in \mathcal{K}$
2. Model discrete choice as

$$\sum_{k \in \mathcal{K}} y_{ijk} = 1, \text{ and } \sum_{k \in \mathcal{K}} P_k y_{ijk} = d_{ij}. \forall (i, j) \in \mathcal{A}$$

3. Model $a_{ij} = \pi d_{ij}^2 / 4$ (nonlinear) as

$$a_{ij} = \sum_{k \in \mathcal{K}} \left( \frac{\pi}{4} P_k \right) y_{ijk} \forall (i, j) \in \mathcal{A}$$
Design of Water Distribution Networks

- Nonsmooth, nonlinear pressure loss as a function of flow $q_{ij}$ along arc $(i, j) \in \mathcal{A}$ and diameter $d_{ij}$:

$$h_i - h_j = \kappa_{ij}d_{ij}^{-c_1} \text{sgn}(q_{ij})|q_{ij}|^{c_2} \quad \forall (i, j) \in \mathcal{A}$$

- Another trick: disaggregate flow into flow variables for each pipe size:

$$q_{ij} = \sum_{k \in \mathcal{K}} q_{ijk}$$

$$(-V_{\max}a_{ij})y_{ijk} \leq q_{ijk} \leq (V_{\max}a_{ij})y_{ijk}$$
Why On Earth?

- Doing this allows us to write the (nonlinear) headloss as a function of a single variable $q_{ijk}$
- Often in applications, the nonlinearity is low-dimensional ($\text{low} \in \{1, 2\}$)
- Therefore a piecewise linear approximation (or relaxation) may be sufficiently accurate

**Luedtke’s Theorem!**

- Note also that $y_{ijk} = 0 \Rightarrow q_{ijk} = 0$
Modeling Piecewise-Linear Functions

Take Your Pick

- Incremental Model [Markowitz and Manne, 1957]
- Multiple Choice Model [Jeroslow and Lowe, 1984]
- Convex Combination Model [Dantzig, 1960, Padberg, 2000]
- Disaggregated Convex Combination Model [Meyer, 1976]
- Logarithmic Model [Vielma and Nemhauser, 2011]
Multiple Choice Model

Want to model ($\forall i$)

$$f(q) = m_i q + c_i, \quad q \in [B_{i-1}, B_i]$$

**Introduce New Variables**

- $w_i$: $= q$ if $q \in [B_{i-1}, B_i]$
- $b_i$: $= 1$ if $q \in [B_{i-1}, B_i]$
- $t \approx f(q)$

$$q = \sum_{i=1}^{n} w_i,$$

$$t = \sum_{i=1}^{n} (m_i w_i + c_i b_i)$$

$$B_{i-1} b_i \leq w_i \leq B_i b_i, \quad \forall i = 1, \ldots, n$$

$$1 = \sum_{i=1}^{n} b_i$$

$$B_0 y \leq q \leq B_n y$$
A Simple Trick

- Instead of modeling the piecewise-linear indicator in the standard way:
  \[ B_0 y \leq q \leq B_n y, \quad 1 = \sum_{i=1}^{n} b_i \]

- Model it as
  \[ y = \sum_{i=1}^{n} b_i \]

- Resulting formulation is provably stronger (locally-ideal)
- All piecewise-linear functions turned on/off by an indicator have a similar modeling trick [Sridhar et al., 2013]
- It is the exact extension of the “perspective reformulation” to this nonconvex case
The Pooling Problem

- A canonical problem in the petrochemical industry
- Decide how to blend (pool) feeds together to make products meeting “quality requirements” at minimum cost

Feeds $I$  Pools $J$  Products $K$

- Generally each product has *multiple* attributes
Pooling Problem

**Basic variables**
- $x_{ij}$: Flow of input $i$ to pool $j$
- $x_{ik}$: Flow of input $i$ to product $k$
- $x_{jk}$: Flow from pool $j$ to product $k$

**Key Parameters**
- $\lambda_i$: Concentration of attribute in feed $i$
- $\ell_k, \upsilon_k$: Bounds on attribute concentration in product $k$
- $C_i, C_j, C_k$: Capacities on feeds, pools, and outputs
- $U_{ij}, U_{ik}, U_{kj}$: Flow capacities (some may be zero)

**Simple (Linear) Constraints**
- Linear Constraints: Flow balance, bounds on flows, feed/pool/output capacity
P-formulation

(For example) Input capacity:

\[
\sum_{\ell \in L} x_{i\ell} + \sum_{j \in J} x_{ij} \leq C_i \quad \forall i \in I
\]

Flow Balance:

\[
\sum_{i \in I} x_{i\ell} = \sum_{j \in J} x_{\ell j} \quad \forall \ell \in L
\]

Pool Quality:

\[
\sum_{i \in I} \lambda_i x_{i\ell} = p_{\ell} \sum_{i \in I} x_{i\ell} \quad \forall \ell \in L
\]

Product Quality:

\[
\sum_{\ell \in L} p_{\ell} x_{\ell j} + \sum_{i \in I} \lambda_i x_{ij} = p_j \left( \sum_{\ell \in L} x_{\ell j} + \sum_{i \in I} x_{ij} \right) \quad \forall j \in J
\]

\[\ell_j \leq p_j \leq v_j \quad \forall j \in J\]
Modeling the Quality Constraints

Q-formulation [Ben-Tal et al., 1994]

- New variables $q_{ij}$: proportion of flow to pool $j$ coming from feed $i$:
  \[
  \sum_{i \in I} q_{ij} = 1
  \]

- $q_{ij}$ satisfy the relations:
  \[
  q_{ij} = \frac{x_{ij}}{\sum_{t \in I} x_{tj}} \iff x_{ij} = q_{ij} \sum_{t \in I} x_{tj}
  \]

- Reformulate using flow balance at pool $j$:
  \[
  \sum_{t \in I} x_{tj} = \sum_{k \in K} x_{jk}
  \]
  \[
  x_{ij} = q_{ij} \sum_{k \in K} x_{jk}
  \]

- Concentration of attribute at pool $i$ is then:
  \[
  \sum_{i \in I} \lambda_i q_{ij}
  \]
Modeling the Quality Constraints (2)

- Concentration of attribute at pool $i$: $\sum_{i \in I} \lambda_i q_{ij}$
- Quality constraint on product $k$:

$$\ell_k \leq \frac{\sum_{i \in I} \lambda_i x_{ik} + \sum_{j \in J} \sum_{i \in I} \lambda_i q_{ij} x_{jk}}{\sum_{i \in I} x_{ik} + \sum_{j \in J} x_{jk}} \leq \upsilon_k$$

- Multiply and gather terms (e.g., for upper bound):

$$\sum_{i \in I} (\lambda_i - \upsilon_k) x_{ik} + \sum_{j \in J} \sum_{i \in I} (\lambda_i - \upsilon_k) q_{ij} x_{jk} \leq 0$$

- Introduce variables $w_{ijk}$ to represent bilinear terms $q_{ij} x_{jk}$:

$$x_{ij} = q_{ij} \sum_{k \in K} x_{jk} = \sum_{k \in K} w_{ijk}$$
Bringing it all together

\[
\sum_{i \in I} q_{ij} = 1, \quad j \in J
\]

\[
x_{ij} = \sum_{k \in K} w_{ijk}, \quad i \in I, j \in J
\]

\[
\sum_{i \in I} (\lambda_i - v_k) x_{ik} + \sum_{j \in J} \sum_{i \in I} (\lambda_i - v_k) w_{ijk} \leq 0, \quad k \in K
\]

\[
\sum_{i \in I} (\ell_k - \lambda_i) x_{ik} + \sum_{j \in J} \sum_{i \in I} (\ell_k - \lambda_i) w_{ijk} \geq 0 \quad k \in K
\]

\[
w_{ijk} = q_{ij} x_{jk}, \quad i \in I, j \in J, k \in K
\]
### PQ-Formulation [Tawarmalani and Sahinidis, 2002]

1. For each $j \in J$, multiply the constraint $\sum_{i \in I} q_{ij} = 1$ with $x_{jk}$:
   \[ \sum_{i \in I} q_{ij} x_{jk} = x_{jk} \]

2. For each $j \in J$, multiply the constraint: $\sum_{k \in K} x_{jk} \leq C_j$ with $q_{ij}$:
   \[ \sum_{k \in K} q_{ij} x_{jk} \leq C_j q_{ij} \]

3. Replace $q_{ij} x_{jk}$ with $w_{ijk}$:
   \[ \sum_{i \in I} w_{ijk} = x_{jk}, \quad \sum_{k \in K} w_{ijk} \leq C_j q_{ij} \]

**Much better relaxations!**

Takeaway: Adding redundant constraints can significantly improve relaxations of nonconvex MINLPs.
The Pooling Problem with Binary Variables

Arcs connecting the input streams to other nodes may be used only by paying a fixed cost.

- Network Design
- Ship assignment

- Add binary variables $z_{ij}$ for these arcs:

$$x_{ij} \leq U_{ij} z_{ij}$$

- A real nonconvex, integer problem.
Practical Advice About the Luedtke Theorem

- We have seen many cases where we have continuous variable \(0 \leq x \leq U\) and associated binary variable \(z \in \{0, 1\}\).
- We see many models that write the expression \(w = zx\).

**On The Horror!**

- Never multiply a binary variable by a bounded continuous variable
- You can linearize this:

\[
0 \leq w \leq Uz \\
-U(1 - z) \leq x - w \leq U(1 - z)
\]
Other MINLP Applications

1. Portfolio Management—**Convex MIQP**
   - Discrete: Trading Strategy
   - Nonlinear: Utility

2. Reactor Core Reloading—**Nonconvex MINLP**
   - Discrete: Bundle Reordering
   - Nonlinear: Neutron Transport

3. Power Grid Operation—**Nonconvex MINLP**
   - Discrete: Unit Commitment
   - Nonlinear: AC Power Flow

4. Building Co-Generation—**Nonconvex MINLP**
   - Discrete: Unit Commitment
   - Nonlinear: Ramping Constraints

5. Oil-Spill Response—**Nonconvex MINLP**
   - Discrete: Resource Allocation
   - Nonlinear: Fluid Flow & Chemistry
Algorithms for convex MINLP are very similar to those for MILP.

NLP-based branch-and-bound:
- Solve NLP relaxation at each node

Cutting plane based algorithms:
- Use linear inequalities to “outer approximate” continuous relaxation
- Solve LP relaxations instead of NLP relaxations
- Update LP relaxations by adding cuts
- Benders decomposition [Geoffrion, 1972], Outer approximation [Duran and Grossmann, 1986], LP/NLP branch-and-bound [Quesada and Grossmann, 1992], Extended cutting plane [Westerlund and Pettersson, 1995], Algorithmic refinements, e.g. [Abhishek et al., 2010, Bonami et al., 2008]
Valid Inequalities for Convex MINLP

Many classes of inequalities for MILP have been extended/studied for convex MINLP

- **Disjunctive/split cuts**

- Chvátal-Gomory rounding and mixed-integer rounding for conic MINLP [Çezik and Iyengar, 2005, Drewes, 2009]

- Conic mixed-integer rounding [Atamtürk and Narayanan, 2010]

- Intersection cuts [Andersen and Jensen, 2013, Modaresi et al., 2014a]

- Minimal valid inequalities and cut generating functions [Kılınç-Karzan, 2013, Moran R. et al., 2012]

- Closures [Dadush et al., 2011b, Dey and Moran R., 2012, Dadush et al., 2011a]
General Approach to Nonconvex MINLP

\[
\begin{align*}
\text{minimize } & \quad f(x) \\
\text{subject to } & \quad g(x) \leq 0, \; x \in X, \; x_i \in \mathbb{Z} \; \forall \; i \in I
\end{align*}
\]

Use our old trick: convex (often polyhedral) relaxation!

- Relax integrality as before: \( x_i \in \mathbb{R} \; \forall \; i \in I \)
- **New:** \( \mathcal{R}(\Theta) \supseteq \Theta = \{ x \in X : g(x) \leq 0 \} \)
- Ensure relaxation is tractable: e.g. \( \mathcal{R}(\Theta) \) is convex

\[
\begin{align*}
\text{minimize } f(x) \\
\text{subject to } g(x) \leq 0, \; x \in X, \; x_i \in \mathbb{Z} \; \forall \; i \in I
\end{align*}
\]
Consider MINLP with nonconvex, factorable $f(x)$ and $g(x)$

$$\min_{x} f(x) \quad \text{subject to} \quad g(x) \leq 0, \ x \in X, \ x_i \in \mathbb{Z} \ \forall \ i \in I$$

**Factorable Functions**

$g(x)$ is **factorable** if it can be expressed as a combination of functions from finite set of operators $O = \{+, \times, /, ^, \sin, \cos, \exp, \log, | \cdot | \}$ whose arguments are variables, constants, or other factorable functions.

- Excludes integrals $\int_{x_0}^{x} h(\xi) d\xi$ and black-box functions
- Represented as expression trees
Expression tree of $f(x_1, x_2) = x_1 \log(x_2) + x_2^3$

Modeling languages (e.g. AMPL, GAMS) have expression tree “API”
Reformulation of Factorable MINLP

Reformulate factorable MINLP as

\[
\begin{align*}
\text{minimize} \quad & x_{n+q} \\
\text{subject to} \quad & x_k = \vartheta_k(x) \quad k = n + 1, n + 2, \ldots, n + q \\
& l_i \leq x_i \leq u_i \quad i = 1, 2, \ldots, n + q \\
& x \in X, \\
& x_i \in \mathbb{Z}, \quad \forall i \in I,
\end{align*}
\]

see e.g. [Smith and Pantelides, 1997]

- \( q \) new auxiliary variables, \( x_{n+1}, \ldots, x_{n+q} \)
- \( \vartheta_k \) is operator from \( O\{+, \times, /, ^\wedge, \sin, \cos, \exp, \log\} \)
Example of Reformulation of Factorable MINLP

\[ \begin{align*}
\text{min } & \quad x_1 + x_1^2 x_2^2 \\
\text{s.t. } & \quad x_1 x_2 + \sin x_2 \leq 4, \\
& \quad x_1 \in [-4, 4] \cap \mathbb{Z}, \\
& \quad x_2 \in [0, 10] \cap \mathbb{Z}.
\end{align*} \]

Reformulation

\[ \begin{align*}
\text{min } & \quad x_7 \\
\text{s.t. } & \quad x_3 = \sin x_2, \\
& \quad x_4 = x_1 x_2, \\
& \quad x_5 = 4 - x_3 - x_4, \\
& \quad x_6 = x_2 x_4, \\
& \quad x_7 = x_1 + x_6, \\
& \quad -4 \leq x_1 \leq 4, \quad 0 \leq x_5 \leq 45, \\
& \quad 0 \leq x_2 \leq 10, \quad -400 \leq x_6 \leq 400, \\
& \quad -1 \leq x_3 \leq 1, \quad -404 \leq x_7 \leq 404, \\
& \quad -40 \leq x_4 \leq 40, \\
\end{align*} \]

- Factorization is not unique!
- Integrality inherited from function and integrality on arguments
- Bounds inherited from function and argument bounds
Reformulation of Factorable MINLP

Factorable form allows systematic construction of convex relaxation:

- Nonconvex sets, \( k = n + 1, n + 2, \ldots, n + q \)

\[
\Theta_k = \{ x \in \mathbb{R}^{n+q} : x_k = \vartheta_k(x), x \in X, l \leq x \leq u, x_i \in \mathbb{Z}, i \in I \}
\]

- Let \( \mathcal{R}(\Theta_k) \supset \Theta_k \) be a convex relaxation

\[
\begin{aligned}
\text{minimize} & \quad x_{n+q} \\
\text{subject to} & \quad x \in \mathcal{R}(\Theta_k) \quad k = n + 1, n + 2, \ldots, n + q \\
& \quad l_i \leq x_i \leq u_i \quad i = 1, 2, \ldots, n + q \\
& \quad x \in X.
\end{aligned}
\]

... convex relaxation ... only look at simple sets!
Examples of Relaxations

Construct relaxation for each operator $\in \mathcal{O}\{+\,\times\,\div\,\wedge\,\sin\,\cos\,\exp\,\log\}$

- Odd-degree monomials, $x_k = x_i^{2p+1}$, see [Liberti and Pantelides, 2003]
- Bilinear functions $x_k = x_i x_j$, [McCormick, 1976]

Let $x = (x_i, x_j)$, $L = (l_i, l_j)$, $U = (u_i, u_j)$

get convex hull of $\Theta_k = \{(x, x_k) : x_k = x_i x_j, L \leq x \leq U\}$:

$$
\begin{align*}
x_k & \geq l_j x_i + l_i x_j - l_i l_j \\
x_k & \geq u_j x_i + u_i x_j - u_i u_j
\end{align*}
\begin{align*}
x_k & \leq l_j x_i + u_i x_j - u_i l_j \\
x_k & \leq u_j x_i + l_i x_j - l_i u_j
\end{align*}
$$

Important!
Tightness of convex hull depends on bounds $l_i, l_j, u_i, u_j$
Examples of Relaxations

Polyhedral relaxation, $\mathcal{R}(\Theta_k)$, of $x_k = x_i^3$ and $x_k = x_ix_j$
Examples of Relaxations

Polyhedral relaxation, $\mathcal{R}(\Theta_k)$, of $x_k = x_i^2$ with $x_i$ continuous/integer

... if $x_i \in \mathbb{Z}$ then add inequalities violated at $x_i' \notin \mathbb{Z}$
Spatial Branch-and-Bound (BnB)

To enforce constraints violated by relaxation solution use spatial BnB

- Implicit enumeration technique like integer BnB
- Recursively define partitions of feasible set into two sets by changing continuous and integer variable bounds
- Solve LP (or convex NLP) relaxations (⇒ lower bounds) ... and nonconvex NLPs (⇒ upper bound if feasible)

Lower bounding problem at node with bounds \((\bar{l}, \bar{u})\), e.g. \(\text{LP}(\bar{l}, \bar{u})\)

\[
\begin{align*}
\text{minimize} \quad & \quad x_{n+q} \\
\text{subject to} \quad & \quad x \in \mathcal{R}(\Theta_k) \quad k = n+1, n+2, \ldots, n+q \\
& \quad \bar{l}_i \leq x_i \leq \bar{u}_i \quad i = 1, 2, \ldots, n+q \\
& \quad x \in X.
\end{align*}
\]
Spatial Branch-and-Bound (BnB)

If $\text{LP} (\bar{l}, \bar{u})$ infeasible, then prune node.

Otherwise, $\hat{x}$ optimal solution of $\text{LP} (\bar{l}, \bar{u})$:

- If $\hat{x}$ integer feasible and feasible in $\text{NLP} (\bar{l}, \bar{u})$ (hence MINLP), then fathom node (new incumbent)
- Otherwise ... branch ...

Two possible ways to branch (integer / nonlinear):

1. $\hat{x}$ not integral: $x_i \leq \lfloor \hat{x}_i \rfloor \lor x_i \geq \lceil \hat{x}_i \rceil$ like integer BnB
2. $\exists k : \hat{x}_k \neq \vartheta_k (\hat{x})$ nonlinear infeasible:
   - Choose branching variable $x_i$ from arguments of $\vartheta_k(x)$
   - Branch $x_i \leq \hat{x}_i \lor x_i \geq \hat{x}_i$ ... two subproblems
   - Refine convex relaxation in each branch ... tighter bounds
Branching example $x_k = \vartheta_k(x_i) = (x_i)^2$ violated
Branching for Spatial Branch-and-Bound

Branching based on $x_i \leq b \lor x_i \geq b$

- Good performance depends on good choice of $i$ and $b$
- Goal: Increase both bounds LP($l^-, u^-$) and LP($l^+, u^+$)
- Strong branching, pseudocost branching, and reliability branching generalized from convex MINLP
- Other techniques (e.g., violation transfer) specific to nonconvex MINLP

Finding continuous branching candidates $x_i$:

- $x_i$ not fixed in parent problem
- $x_i$ is argument of violated function $\hat{x}_k \neq \vartheta_k(x)$
Tightening Bounds and Relaxations

Bound tightening to reduce range of bounds \( x_i \in [l_i, u_i] \)
... because tighter bounds \( \Rightarrow \) tighter relaxations \( \Rightarrow \) smaller trees

- Spatial BnB + Bound tightening = Branch-and-reduce
  [Ryoo and Sahinidis, 1996, Belotti et al., 2009]

Conceptual bound-tightening procedure:
Feasible set \( \mathcal{F} = \{ x \in [l, u] : g(x) \leq 0, \ x \in X, x_i \in \mathbb{Z}^p \} \)
Solve 2n (global) optimization problems, given upper bound \( U \):

\[
l_i' = \min \{ x_i : x \in \mathcal{F}, f(x) \leq U \}; \quad u_i' = \max \{ x_i : x \in \mathcal{F}, f(x) \leq U \}.
\]

... nonconvex MINLPs just as hard \( \Rightarrow \) use relaxations:
1. FBBT: feasibility-based bound tightening
2. OBBT: optimality-based bound tightening
FBBT: Feasibility-Based Bound Tightening

FBBT broadly used:
- Artificial intelligence community & constraint programming
- NLP solvers [Messine, 2004]
- MILP solvers [Savelsbergh, 1994]

**Basic Principle of FBBT**

Infer bounds on $x_i$ from tighter bounds on $x_j$ for $j \neq i$.

Example: $x_j = x_i^3$ and $x_i \in [l_i, u_i]$
- Tighten interval of $x_j$ to $[l_j, u_j] \cap [l_i^3, u_i^3]$
- Tightened $l'_j$ on $x_j \Rightarrow$ tighter $l'_i = \sqrt[3]{l_j}$ for $x_i$

FBBT is efficient and fast, but convergence can be slow
Solving \( \min / \max x_i \) s.t. \( x \in \mathcal{F} \) (nonconvex MINLP) not practical

Instead, use current relaxation

\[
\mathcal{F}(l, u) = \left\{ x \in \mathbb{R}^{n+q} : \begin{array}{c}
l_i \leq x_i \leq u_i \\
x \in X
\end{array}, \begin{array}{c}
k = n + 1, n + 2, \ldots, n + q \\
i = 1, 2, \ldots, n + q
\end{array} \right\}.
\]

Now get bounds on \( x_i \) for \( i = 1, \ldots, n \) by solving \( 2n \) LPs:

\[
\begin{align*}
l'_i &= \min \{ x_i : x \in \mathcal{F}(l, u) \} \\
u'_i &= \max \{ x_i : x \in \mathcal{F}(l, u) \}
\end{align*}
\]

... only apply at root node, or small number of nodes
The Trouble with Factorable Relaxations

Factorable relaxations are primarily based on *local* structure ⇒ Potentially weak bounds

**Example:** \( \Theta = \{ (x_1, x_2) : x_1^2 + x_2^2 \geq 1, \ 0 \leq x_i \leq 2 \} \)

Convex hull: \( x_1 + x_2 \geq 1 \)

Best possible factorable relaxation:

\[
\begin{align*}
    x_3 &= x_1^2 \Rightarrow (1/2)x_3 \leq x_1 \\
    x_4 &= x_2^2 \Rightarrow (1/2)x_4 \leq x_2 \\
    x_3 + x_4 &\geq 1 \Rightarrow \\
    x_1 + x_2 &\geq (1/2)(x_3 + x_4) \geq 1/2
\end{align*}
\]

Remedy: Study relaxations of more global structures
A Special Structure: Nonconvex Quadratic Functions

Quadratically constrained quadratic program (QCQP)

\[
\begin{align*}
\text{minimize} & \quad x^T Q_0 x + c_0^T x, \\
\text{subject to} & \quad x^T Q_k x + c_k^T x \leq b_k, \quad k = 1, \ldots, q \\\n& \quad A x \leq b, \quad 0 \leq x \leq u, \quad x_i \in \mathbb{Z}, \quad \forall i \in I,
\end{align*}
\]

- \( Q_k \) not necessarily convex \( \Rightarrow \) nonconvex problem
- Problem is \( \mathcal{NP} \)-hard even if all variables are continuous
- Many important special cases: Bilinear programming, pooling problem, integer linear programming, max cut, …
- Extensively studied: E.g.,
General Nonconvex Quadratic Functions

Equivalent reformulation of QCQP: introduce $X_{ij}$ for all $i, j$ pairs

\[
\begin{align*}
\text{minimize} & \quad Q_0 \bullet X + c_0^T x, \\
\text{subject to} & \quad Q_k \bullet X + c_k^T x \leq b_k, \quad k = 1, \ldots, q \\
& \quad A x \leq b, \\
& \quad 0 \leq x \leq u, \quad x_i \in \mathbb{Z}, \quad \forall i \in I, \\
& \quad X = xx^T,
\end{align*}
\]

where $X = [X_{ij}]$ matrix, $Q_k \bullet X = \sum_{ij} [Q_k]_{ij} X_{ij} = \sum_{ij} [Q_k]_{ij} x_i x_j$.

- $X = xx^T$ represents nonconvex constraint $X_{ij} = x_i x_j$
  ... otherwise problem is linear!
- Relaxing equality $X = xx^T$ gives convex (linear) relaxation

We show two approaches: RLT and SDP
Reformulation-Linearization Technique (RLT)
[Adams and Sherali, 1986]

Begin with standard polyhedral relaxation of $X_{ij} = x_i x_j$:

$$X_{ij} \geq 0, \quad X_{ij} \geq u_i x_j + u_j x_i - u_i u_j, \quad X_{ij} \leq u_j x_i, \quad X_{ij} \leq u_i x_j$$

Strengthen using more structure

1. Exploiting binary variables: $x_i \in \{0, 1\} \Rightarrow x_i^2 = x_i$
   - Add linear constraints $X_{ii} = x_i$

2. Multiply linear constraints (e.g., PQ formulation for pooling)
   - Multiplying $x_i \geq 0$ and $b_t - \sum_j a_{tj} x_j \geq 0$ gives
     $$b_t x_i - \sum_j a_{tj} x_i x_j \geq 0$$
   - Replace $x_i x_j$ with $X_{ij}$ yields linear inequality
     $$b_t x_i - \sum_j a_{tj} X_{ij} \geq 0$$
Semidefinite Programming (SDP) relaxations of QCQPs

1. Relax \( X - xx^T = 0 \) to \( X - xx^T \succeq 0 \)

\[
X - xx^T \succeq 0 \iff \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0
\]

2. Improve by adding \( X_{ii} = x_i \) for binary \( x_i \in \{0, 1\} \)

3. Additional constraints by squaring & linearizing constraints

... here \( H \succeq 0 \) means \( H \) positive semidefinite \( (x^T H x \geq 0, \forall x) \)

Both RLT and SDP provide good relaxations ... can also combine them
Some Open Research Question in MINLP

A very biased view of open research in MINLP
- Strong relaxations for *nonlinear structures*
- Implementation & software design issues
- Open questions from previous slide

Going beyond traditional MINLP
- Black-box functions or simulations ... *unrelaxable variables*
- Partial-differential equation constraints
- Bilevel problems, e.g., leader-follower games
- Stochastic/robust optimization

Make impact in important applications
The Future of MINLP?

Take Home Messages

- Modeling is extremely important for MINLP
  ... strong relaxations are key
- Effective techniques from MILP useful for MINLP
  ... more work can be done!
- MINLP software maturing,
  ... but lags behind commercial MILP software
- Linear better than convex better than nonconvex
- Exploit structure whenever possible


Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables.


Global minimization by reducing the duality gap.

Lift-and-project cuts for mixed integer convex programs.

An algorithmic framework for convex mixed integer nonlinear programs.

Semidefinite relaxations for non-convex quadratic mixed-integer programming.

*Mathematical Programming*, pages 1–18. 10.1007/s10107-012-0534-y.

On the copositive representation of binary and continuous nonconvex quadratic programs.

*Mathematical Programming*, 120:479–495.

On nonconvex quadratic programming with box constraints.


Unbounded convex sets for non-convex mixed-integer quadratic programming.


Generalized Benders decomposition.  

Modelling with integer variables.  

Effective separation of disjunctive cuts for convex mixed integer nonlinear programs.  
Technical Report 1681, Computer Sciences Department, University of Wisconsin-Madison.

On minimal valid inequalities for mixed integer conic programs.

Two-term disjunctions on the second-order cone.

Convex envelopes of monomials of odd degree.


Disjunctive cuts for non-convex mixed integer quadratically constrained programs.
Manuscript, available on Optimization Online, (www.optimization-online.org).

A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems. 
Kluwer, Dordrecht.

Global optimization of nonconvex MINLPs. 

Locally ideal formulations for piecewise linear functions with indicator variables. 


A cutting plane method for solving convex MINLP problems.  
*Computers & Chemical Engineering, 19:*s131–s136.

Polyhedral approach for nonconvex quadratic programming problems with box constraints.  