Modeling with Mixed-Integer Nonlinear Optimization
Summer School on Optimization of Dynamical Systems

Sven Leyffer and Jeff Linderoth

Argonne National Laboratory

September 3-7, 2018
Mixed-Integer Nonlinear Programming (MINLP)

1. Monday, September 3: Two-Part Lecture
   1. Modeling with Mixed-Integer Nonlinear Optimization

2. Tuesday, September 4: Two-Part Lecture & Tutorial
   1. Advanced Methods for Convex MINLPs

3. Tutorial: 10:30am – Noon (starting with short intro)

Time permitting: Short transition to dynamical MINLPs ...
Outline: Modeling with Mixed-Integer Nonlinear Optimization

1. Problem Definition and Assumptions
2. MINLP Modeling Practices
3. A Short Introduction to AMPL
4. Model: Quadratic Uncapacitated Facility Location
Mixed-Integer Nonlinear Optimization

Mixed-Integer Nonlinear Program (MINLP)

\[
\begin{align*}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & c(x) \leq 0 \\
& x \in \mathcal{X} \\
& x_i \in \mathbb{Z} \text{ for all } i \in \mathcal{I}
\end{align*}
\]

- $\mathcal{X}$ bounded polyhedral set, e.g. $\mathcal{X} = \{x : l \leq A^T x \leq u\}$
- $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ twice continuously differentiable (sometimes convex)
- $\mathcal{I} \subset \{1, \ldots, n\}$ subset of integer variables
- Relaxations satisfy a constraint qualification (technical)
Challenges of MINLP
Combines challenges of handling nonlinearities with combinatorial explosion of integer variables

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

- R. Tyrrell Rockafellar
Challenges of MINLP

Combines challenges of handling nonlinearities with combinatorial explosion of integer variables

\[ f \text{ and } c \text{ are convex functions, then we have a convex MINLP} \]
\[ f \text{ and } c \text{ are not convex, then we have a nonconvex MINLP} \]

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

- R. Tyrrell Rockafellar
The Importance of Being Convex

2D Rastrigin Test Function

Not All Solvers Are Equal
Without convexity, many solvers only guarantee local optimality
The Importance of Being Convex

2D Rastrigin Test Function

Impact of Convexity

- Nonconvex MINLPs are much harder to solve
- May not be able to “prove” global optimality
- Make sure you really need nonconvexity in your MINLP!

Not All Solvers Are Equal

Without convexity, many solvers only guarantee local optimality
Mixed-Integer Nonlinear Optimization

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

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad c(x) \leq 0 \\
& \quad x \in \mathcal{X} \\
& \quad x_i \in \mathbb{Z} \text{ for all } i \in \mathcal{I}
\end{align*}
\]

- MINLPs are NP-hard ... includes MILP, which are NP-hard, see [Kannan and Monma, 1978]
- Worse: MINLP are undecidable, see [Jeroslow, 1973]:
  \[\exists\] quadratically constrained IP for which no computing device can compute the optimum for all problems in this class
  ... but we’re OK if \(\mathcal{X}\) is compact!
MINLP Specializations

- **MISOCNP**: Mixed Integer Second Order Cone Program
  - Convex-MINLP with $c_i(x) = \|A_i x + b_i\|_2 - p_i^T x + q_i$

- **MIPP**: Mixed Integer Polynomial Program: $c_i(x)$ polynomial

- **MIQP**: Mixed Integer Quadratic Program
  - May be convex or nonconvex: $c_i(x) = x^T Q_i x + p_i^T x + q_i$
  - Convex MIQP is a special case of MISOCNP, if $Q_i \succeq 0$
  - If $f$ is convex quadratic and $c$ is an affine mapping, then there are specialized algorithms for convex-MIQP

- **MILP**: Mixed Integer Linear Program
  - Most efficient solvers: $c_i(x) = a_i^T x - b_i$
Practical Complexity ≃ Is There Hope to Solve it?

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

Given a problem of class $X$ with $Y$ decision variables, what is the largest value of $Y$ for which Jim, Jeff, or Sven would be willing to bet $50 that a “state-of-the-art” solver could solve the problem?
## Practical Complexity \( \sim \) Is There Hope to Solve it?

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

Given a problem of class \( X \) with \( Y \) decision variables, what is the largest value of \( Y \) for which Jim, Jeff, or Sven 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>MISOCM</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>MILP</td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>(5 \times 10^7)</td>
<td>LP</td>
<td></td>
</tr>
</tbody>
</table>
Outline

1. Problem Definition and Assumptions
2. MINLP Modeling Practices
3. A Short Introduction to AMPL
4. Model: Quadratic Uncapacitated Facility Location
MINLP Modeling Practices

Modeling plays a fundamental role in MILP see [Williams, 1999] ... even more important in MINLP

- MINLP combines integer and nonlinear formulations
- Reformulations of nonlinear relationships can be convex
- Interactions of nonlinear functions and binary variables
- Sometimes we can linearize expressions

MINLP Modeling Preference

We prefer linear over convex over nonconvex formulations.
Modeling with Integer Variables

Linderoth “Fundamental Theorem of Integer Variables”

97.238% of MINLPs have integer variables for two purposes only:

1. Binary variables to make a “multiple choice” selection
2. Binary indicator variables that turn on/off continuous variables and/or constraints.

1. Multiple Choice Selection

\[ y \in \{D_1, D_2, \ldots, D_k\} \]

where \( D_i \) are discrete parameters (e.g. pipe diameters)

2. Indicator Variables

\[
\text{if } y_i = 1 \text{ then } c_i(x) \leq 0, \quad \text{otherwise } c_i(x) \leq \infty
\]
Modeling Multiple Choice Selection

Discrete Choices

We can model \( y \in \{D_1, D_2, \ldots, D_k\} \), where \( D_i \) discrete parameters as special ordered set (SOS) [Beale and Tomlin, 1970]

\[
y = \sum_{i=1}^{k} z_i D_i, \quad 1 = \sum_{i=1}^{k} z_i, \quad z_i \in \{0, 1\}
\]

- Similarly linearize univariate functions \( f(z), \ z \in \mathbb{Z} \)
- Generalizes to higher dimensions, [Beale and Forrest, 1976]
- Solvers detect SOS structure and use special branching rules
Calculus of Logical Modeling: Indicator/Lookout Variables

1. **Indicator variable**: \( y_i \in \{0, 1\} \) to force a constraint to hold

\[
y_i = 1 \implies c_i(x) \leq 0
\]

can be modeled as

\[
y_i \in \{0, 1\} \quad \text{and} \quad c_i(x) \leq M(1 - y_i)
\]

where \( M > 0 \) is known upper bound on \( c(x) \) for \( x \in \mathcal{X} \).

2. **Lookout variable**: \( y_i \in \{0, 1\} \) is forced if a constraint holds

\[
a_i^T x + b_i \leq 0 \implies y_i = 1
\]

can be modeled as

\[
y_i \in \{0, 1\} \quad \text{and} \quad a_i^T x + b_i \geq (m - \epsilon)y_i
\]

where \( m > 0 \) lower bound on \( a_i^T x + b_i \) for \( x \in \mathcal{X} \), and \( \epsilon > 0 \)

... is tolerance tol (e.g. \( 10^{-4} \))
Some Useful Nonlinear Variable Transformations

Design of multiproduct batch plant includes nonconvex terms

$$\sum_{j \in M} \alpha_j N_j V_j^{\beta_j}; \quad C_i N_j \geq \tau_{ij}; \quad \sum_{i \in N} \frac{\psi_i}{B_i} C_i \leq \gamma$$

where variables are upper case, parameters are Greek letters.

Introduce log-transform variables:

$$v_j = \ln(V_j), \quad n_j = \ln(N_j), \quad b_i = \ln(B_i), \quad c_i = \ln(C_i)$$

Transformed expressions are convex:

$$\sum_{j \in M} \alpha_j e^{n_j + \beta_j v_j}, \quad c_i + n_j \geq \ln(\tau_{ij}), \quad \sum_{i \in N} \psi_i e^{c_i - b_i} \leq \gamma$$
Linearization of Constraints

Assume $x_2 \neq 0$. A simple transformation (a constant parameter):

$$\frac{x_1}{x_2} = a \iff x_1 = ax_2$$

Linearization of bilinear terms $x_1 x_2$ with:

- Binary indicator variable $x_2 \in \{0, 1\}$
- Variable upper bound: $0 \leq x_1 \leq Ux_2$

... new variable $x_{12}$ replaces $x_1 x_2$ ... using constraints

$$0 \leq x_{12} \leq x_2 U \quad \text{and} \quad 0 \leq x_1 - x_{12} \leq U(1 - x_2),$$

Proof: $x_{12} \in \{0, x_1\}$ follows from constraints.
Never Multiply a Nonlinear Function by a Binary

Previous example generalizes to nonlinear functions: $x_2 c(x_1) \leq 0$

Warning

Never model on/off constraints by multiplying by a binary variable.

Three alternative approaches

- Disjunctive programming, [Grossmann and Lee, 2003]
- Perspective formulations, [Günlük and Linderoth, 2012]
- Big-M formulation (weak relaxations)
Another Example of Bad Nonlinear Models

**Warning**

Never replace a binary variable by a nonlinear expression!

We can write a binary constraint, \( x \in \{0, 1\} \), equivalently as

\[
0 \leq x \leq 1 \quad \text{and} \quad x(1 - x) = 0
\]

or also as a complementarity constraint ...

\[
0 \leq x \perp x \leq 1
\]

... both are bad ... hide integrality from solvers!
Outline

1 Problem Definition and Assumptions

2 MINLP Modeling Practices

3 A Short Introduction to AMPL

4 Model: Quadratic Uncapacitated Facility Location
A Short Introduction to AMPL

AMPL: A Mathematical Programming Language

- Algebraic modeling language for optimization
- Three main model/instance components
  1. Model file (describes algebraic form of equations) *.mod
  2. Data file (describe data of the instance) *.dat
  3. Command file (optional: describe control sequence) *.ampl
- Link to solvers (binaries compiled with AMPL-solver interface)
  1. NLP Solvers: CONOPT, Knitro, LOQO, Minos, SNOPT
  2. MINLP Solvers: Baron, KNITRO
  3. MILP Solvers: GuRoBi, XPRESS

Other AMPL Solvers

- Minotaur (Argonne) download and binaries (Linux/MacOS)
  Minotaur_Download
- COIN-OR Project https://www.coin-or.org/projects/
Collections of MINLP Test Problems

AMPL Collections of MINLP Test Problems

2. IBM/CMU collection [egon.cheme.cmu.edu/ibm/page.htm](http://egon.cheme.cmu.edu/ibm/page.htm)

GAMS Collections of MINLP Test Problems

1. GAMS MINLP-world [www.gamsworld.org/minlp/](http://www.gamsworld.org/minlp/)

Solve MINLPs online on the NEOS server,
[www.neos-server.org/neos/](http://www.neos-server.org/neos/)

... and there are even a few CUTEr problems in SIF!
We have a full (temporary) license of AMPL for the course

**Optimization Problem**

\[
\min_x \exp(-x_1) + \sum_{i=2}^{3} x_i^2
\]

s.t.  \( x_1 \log(x_2) + x_2^3 \geq 1 \)

5 \( \geq x_1, x_2, x_3 \geq 0 \)

**AMPL Formulation**

\[
\text{var } x\{1..3\} \geq 0, \leq 5; \quad \# \ldots \text{ variables}
\]

\[
\text{minimize} \quad \# \ldots \text{ objective functn}
\]

\[
f: \exp(-x[1]) + \text{sum\{i in 2..3\} x[i]^2};
\]

\[
\text{subject to} \quad \# \ldots \text{ constraints}
\]

\[
\text{con: x[1]*log(x[2]) + x[2]^3} \geq 1;
\]

**Beware:** \( x_2 > 0 \ldots \log(x_2) \) undefined for \( x_2 \leq 0! \)
1. Create a *.mod model file (see file)

2. Start `ampl`; load model (e.g. `simple.mod`); select solver:
   ```bash
   ampl: reset; model simple.mod;
   ampl: option solver ipopt;
   ampl: solve;
   ```

3. Display the answer or trouble shoot
   ```bash
   ampl: display _varname, _var.lb, _var, _var.ub;
   ampl: display _conname, _con.lb, _con.body, _con.ub;
   ampl: expand;
   ```

   ... list variable/constraint name, lower bnd, body, upper bnd

   ... shows all constraints and objective functions

4. We forgot to ensure \( x^2 > 0 \) so that \( \log(x^2) \) defined:
   ```bash
   ampl: let x[2] := 1;
   ampl: solve;
   ```

   ... assigns an initial value to \( x^2 \) (different from default, 0).
Running & Trouble Shooting an AMPL Model

1. Create a *.mod model file (see file)

2. Start ampl; load model (e.g. simple.mod); select solver:
   - ampl: reset; model simple.mod;
   - ampl: option solver ipopt;
   - ampl: solve;

3. Display the answer or trouble shoot
   - ampl: display _varname, _var.lb, _var, _var.ub;
   - ampl: display _conname, _con.lb, _con.body, _con.ub;
   - ampl: expand;

   ... list variable/constraint name, lower bnd, body, upper bnd
   ... shows all constraints and objective functions

We forgot to ensure $x^2 > 0$ so that log($x^2$) defined:

- ampl: let x[2] := 1;
- ampl: solve;

... assigns an initial value to $x^2$ (different from default, 0).
Running & Trouble Shooting an AMPL Model

1. Create a *.mod model file (see file)

2. Start ampl; load model (e.g. simple.mod); select solver:
   - `ampl: reset; model simple.mod;`
   - `ampl: option solver ipopt;`
   - `ampl: solve;`

3. Display the answer or trouble shoot
   - `ampl: display _varname, _var.lb, _var, _var.ub;`
   - `ampl: display _conname, _con.lb, _con.body, _con.ub;`
   - `ampl: expand;`
   
   ... list variable/constraint name, lower bnd, body, upper bnd
   
   ... shows all constraints and objective functions

4. We forgot to ensure $x_2 > 0$ so that $\log(x_2)$ defined:
   - `ampl: let x[2] := 1;`
   - `ampl: solve;`
   
   ... assigns an initial value to $x_2$ (different from default, 0).
Short Quiz (see geartrain.mod in MacMINLP)

Consider the gear-train design problem for best matching gear ratio

\[
\text{minimize } \left( \frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2 \quad x \in \mathbb{Z}^4, \; 12 \leq x_i \leq 60
\]

- Is the problem a convex or nonconvex MINLP?
- Is there an equivalent but simpler formulation?
Outline

1. Problem Definition and Assumptions
2. MINLP Modeling Practices
3. A Short Introduction to AMPL
4. Model: Quadratic Uncapacitated Facility Location
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 $i \in M$
Quadratic Uncapacitated Facility Location Problem

- 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

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

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

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

\[ z_i \in \{0, 1\} \quad \forall i \in M \]
Partial AMPL Model for Quadratic facility Location

- **Declare variables**
  
  \[
  \text{var } x\{i,j\} \geq 0, \leq 1; \quad \text{# ... } \% \text{ of customer } i \text{ served by } j
  \]
  \[
  \text{var } z\{i\} \text{ binary}; \quad \text{# ... } z_i=1, \text{ iff facility } i \text{ built}
  \]

- **Declare objective function**
  
  \[
  \text{minimize quadCost: } \sum\{i \text{ in } I\} C[i]*z[i] \\
  + \sum\{i \text{ in } I, j \text{ in } J\} Q[i,j]*x[i,j]^2;
  \]

- **Declare constraints**
  
  subject to

  \[
  \text{# ... only serve customers from open facilities} \\
  \text{varUBD}\{i \text{ in } I, j \text{ in } J\}: x[i,j] \leq z[i];
  \]

  \[
  \text{# ... meet all customer’s demands} \\
  \text{meetDemand}\{j \text{ in } J\}: \sum\{i \text{ in } I\} x[i,j] = 1;
  \]

**Missing:** Definition of sets I, j, and data Q, C!
Base Case of Induction to “Linderoth Theorem”

- Binary variables used as indicators: $z_i = 0 \Rightarrow x_{ij} = 0$
- If $z = 1$, then we need to model the epigraph of $x_{ij}^2$
Base Case of Induction to “Linderoth Theorem”

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

Building on Years of MIP Expertise

- Mixed Integer Linear Programmers carefully study simple problem structures for “good” formulations of problems
- Goal: closely approximate convex hull of feasible points

... solve LP relaxation

- Study structure of a special MINLP with indicator variables
A Very Simple Structure

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

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

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

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

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

Deep Insights

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

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

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

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

Second Order Cone Programming

- \exists \text{ effective, robust algorithms for optimizing over } \text{conv}(R)
The Beauty of Cones

Remarkable Result for Convex MINLPs

All 333 convex problems in MINLPLIB2 can be represented with only four types of cones (for $x \in \mathbb{R}^n$) [Lubin et al., 2016]

1. Quadratic cones: $\|x\|_2^2 \leq x_0$, for $x \in \mathbb{R}^n$
2. Rotated quadratic cones: $2x_1x_2 \geq (x_3^2 + \ldots + x_n^2)^{1/2}$
3. Power cones: $|x|^p \leq x_0$ ... or $\ell_p$-norms $\|x\|_p \leq x_0$
4. Exponential cones: e.g. $e^x \leq x_0$

Advantages of Cones

- Convex cones give strong relaxations (more later)
- Build on “Disciplined Convex Modeling” [Grant et al., 2006] to assure convexity

Snag: Need interior-point solvers ... or use cutting planes!
Teaching Points: Modeling MINLPs

Modeling MINLP

\[
\text{minimize } f(x) \quad \text{subject to } c(x) \leq 0, \ x \in X, \ x_i \in \mathbb{Z} \ \forall \ i \in I
\]

- Most binary variables used as indicator or lookout variables
- Reformulation tricks are important ... linearization of \( x_1x_2 \) for \( x_2 \in \{0, 1\} \)
- AMPL modeling language: convenient & intuitive

Tighter relaxations/formulations from Conic Programming

- Use of perspective and conic formulations important


