Basic notions of Mixed Integer Non-Linear Programming

Claudia D’Ambrosio

CNRS & LIX, École Polytechnique

5th Porto Meeting on Mathematics for Industry, April 10, 2014
Outline

1. Intro to MINLP
2. What is a convex MINLP?
3. “Basic” subproblems
4. Convex MINLP Algorithms
   - Branch-and-Bound
   - Outer-Approximation
   - Generalized Benders Decomposition
   - Extended Cutting Plane
   - LP/NLP-based Branch-and-Bound
   - Hybrid Algorithms
5. Practical Tools
   - Convex MINLP Solvers
   - Modeling Languages
   - Neos
   - MINLP Libraries
   - Solvers’ performance
Mixed Integer NonLinear Programming (MINLP).

\[
\begin{align*}
\text{min } f(x, y) \\
g(x, y) & \leq 0 \\
\end{align*}
\]

\[
\begin{align*}
x & \in X = \{ x \mid x \in \mathbb{R}^p, Dx \leq d, x^L \leq x \leq x^U \} \\
y & \in Y = \{ y \mid y \in \mathbb{Z}^q, Ay \leq a, y^L \leq y \leq y^U \}
\end{align*}
\]

with \( f(x) : \mathbb{R}^{p+q} \to \mathbb{R} \) and \( g(x) : \mathbb{R}^{p+q} \to \mathbb{R}^m \) are

* continuous
* twice differentiable

functions.

- Undecidable/NP-hard problem.
- Local optima might not be also global optima.
Sources of non-convexities:

i. non-convex functions

ii. integrality requirement $\rightarrow$ worst-case: exponential number of possible values of integer variables, e.g., if binary, $2^I$ cases.
Outline

1. Intro to MINLP
2. What is a convex MINLP?
3. “Basic” subproblems
4. Convex MINLP Algorithms
   - Branch-and-Bound
   - Outer-Approximation
   - Generalized Benders Decomposition
   - Extended Cutting Plane
   - LP/NLP-based Branch-and-Bound
   - Hybrid Algorithms
5. Practical Tools
   - Convex MINLP Solvers
   - Modeling Languages
   - Neos
   - MINLP Libraries
   - Solvers’ performance
What is a convex MINLP?

Convex Mixed Integer NonLinear Programming (MINLP).

\[
\begin{align*}
\min \ f(x, y) \\
g(x, y) &\leq 0 \\
x &\in X = \{x \mid x \in \mathbb{R}^p, Dx \leq d, x^L \leq x \leq x^U\} \\
y &\in Y = \{y \mid y \in \mathbb{Z}^q, Ay \leq a, y^L \leq y \leq y^U\}
\end{align*}
\]

with \( f(x) : \mathbb{R}^{p+q} \to \mathbb{R} \) and \( g(x) : \mathbb{R}^{p+q} \to \mathbb{R}^m \) are

* continuous
* twice differentiable
* convex functions.

- Local optima are also global optima.
- Continuous relaxation: convex NLP.
Convex vs. Non-Convex MINLP

Convex

- **Convex** function involved
- **Convex** continuous relaxation $\rightarrow$ **polynomially** solvable
- Local optima $=$ Global optima

Non-Convex

- **Non-Convex** function involved
- **Non-Convex** continuous relaxation $\rightarrow$ **NP-hard** problem
- Local optima $\neq$ Global optima

Focus on

**Convex MINLPs**

Non-convex MINLPs $\rightarrow$ A. Gleixner
min \ f(x, y) \\
g(x, y) \leq 0 \\
x \in X \\
y \in \{y \mid Ay \leq a\} \\
y_j \leq \alpha_j^k \quad j \in \{1, 2, \ldots, q\} \\
y_j \geq \beta_j^k \quad j \in \{1, 2, \ldots, q\}

k: current step of a Branch-and-Bound procedure; 
\alpha^k: current lower bound on \ y (\alpha^k \geq y^L); 
\beta^k: current upper bound on \ y (\beta^k \leq y^U).
NLP restriction for a fixed $y^k$:

$$\min f(x, y^k)$$
$$g(x, y^k) \leq 0$$
$$x \in X.$$

Feasibility subproblem for a fixed $y^k$:

$$\min u$$
$$g(x, y^k) \leq u$$
$$x \in X$$
$$u \in \mathbb{R}.$$
MILP relaxation

\[
\begin{align*}
\min & \quad \gamma \\
\text{s.t.} & \quad f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq \gamma \quad \forall k \\
g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq 0 \quad \forall k \quad \forall i \in I^k
\end{align*}
\]

where \( I^k \subseteq \{1, 2, \ldots, m\} \). Two “classical” choices:

- \( I^k = \{1, 2, \ldots, m\} \)
- \( I^k = \{i \mid g(x^k, y^k) > 0, 1 \leq i \leq m\} \)
Outline

1. Intro to MINLP
2. What is a convex MINLP?
3. "Basic" subproblems
4. Convex MINLP Algorithms
   - Branch-and-Bound
   - Outer-Approximation
   - Generalized Benders Decomposition
   - Extended Cutting Plane
   - LP/NLP-based Branch-and-Bound
   - Hybrid Algorithms
5. Practical Tools
   - Convex MINLP Solvers
   - Modeling Languages
   - Neos
   - MINLP Libraries
   - Solvers’ performance
Convex MINLP Algorithms

- Branch-and-Bound (BB).
- Outer-Approximation (OA).
- Generalized Benders Decomposition (GBD).
- Extended Cutting Plane (ECP).
- LP/NLP-based Branch-and-Bound (QG).
- Hybrid Algorithms (Hyb).
Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

1: $f^* = +\infty$, $\Pi = \{ P^0 \}$ where $P^0 = \text{NLP relaxation}$.

2: while $\Pi \neq \emptyset$ do
3:  Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{ P \}$.
4:  Solve $P$ obtaining $(\bar{x}, \bar{y})$.
5:  if $P$ infeasible or $f(\bar{x}, \bar{y}) \geq f^*$ then
6:      break
7:  end if
8:  if $\bar{y} \in \mathbb{Z}^q$ then
9:      $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
10:     Update $\Pi$ potentially fathoming subproblems.
11:    else
12:      Take a fractional value $\bar{y}_j$ and obtain two subproblems $P^1 = P$ with $\alpha^1_j = \lfloor \bar{y}_j \rfloor$ and $P^2 = P$ with $\beta^2_j = \lfloor \bar{y}_j \rfloor + 1$.
13:      $LB(P^1) = LB(P^2) = f(\bar{x}, \bar{y})$.
14:      $\Pi = \Pi \cup \{ P^1, P^2 \}$.
15:  end if
16: end while
RETURN $(x^*, y^*)$.

Fathoming is performed when:

- The subproblem solution is MINLP feasible ($f^*$).
- The subproblem is infeasible.
- The subproblem $P^k$ has $LB(P^k) \geq f^*$. 
Branch-and-Bound (BB)

```
0
x_1 = 1
LB = 21

1
x_1 = 0
LB = 24
x_2 = 1
LB = 24

f^* = 26

2

3
LB = 30

4
x_2 = 1
LB = 25

5
x_2 = 0

6
x_3 \leq 2
LB = 26

f^* = 25

7
x_3 \geq 3

8
LB = 25
```

C. D'Ambrosio (CNRS)
Outer-Approximation (OA)

Duran and Grossmann, 1986.

\[
\begin{align*}
\min_{\gamma} & \quad f(x^k, y^k) + \nabla f(x^k, y^k)^T \left( \begin{array}{c} x - x^k \\ y - y^k \end{array} \right) \\ g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \left( \begin{array}{c} x - x^k \\ y - y^k \end{array} \right) & \leq 0 \quad \forall k \forall i \in I^k \\
\end{align*}
\]

\[
I^k = \{1, 2, \ldots, m\} \quad \forall k = 1, \ldots, K.
\]

NB. The linearization constraints of MILP relaxation are not valid for non-convex MINLPs.
Outer-Approximation (OA)

Initialization: $K = 1$, define an initial MILP relaxation, $f^* = +\infty$, $LB = -\infty$.

1: while $f^* \neq LB$ do
2: Solve the current MILP relaxation (obtaining $(\hat{x}^K, \hat{y}^K)$) and update LB.
3: Solve the current NLP restriction for $\hat{y}^K$.
4: if NLP restriction for $\hat{y}^K$ infeasible then
5: Solve the infeasibility subproblem for $\hat{y}^K$.
6: else
7: Update $f^*$.
8: end if
9: Generate new linearization cuts and update MILP relaxation.
10: $K = K + 1$.
11: end while
Generalized Benders Decomposition (GBD)

Geoffrion, 1972.
Similar to OA, but with a different MILP relaxation, i.e.,

- $x \in X$ is relaxed.
- $I^k = \{i \mid g(x^k, y^k) = 0, 1 \leq i \leq m\} \quad \forall k = 1, \ldots, K.$

Proposition

Given the same set of $K$ subproblems, the LB provided by the MILP relaxation of OA is $\geq$ of the one provided by the MILP relaxation of GDB.

Proof.

(Sketch of) It can be shown that the constraints of GDB MILP relaxation are surrogate of the ones of OA MILP relaxation (see, Quesada and Grossmann, 1992).
Extended Cutting Plane (ECP)


1: $k = 1$, obtain an initial MILP relaxation.
2: Solve the MILP relaxation obtaining $(x^k, y^k)$.
3: if no constraint is violated by $(x^k, y^k)$ then
4: Return $(x^k, y^k)$ (optimal solution).
5: else
6: Generate (some) new linearization constraints from $(x^k, y^k)$ and update MILP relaxation.
7: end if
8: $k = k + 1$.
9: Go to step 1.

More iterations needed wrt OA.

1. Obtain an initial MILP relaxation.

2. Solve the MILP relaxation through BB for MILP, but, anytime a MILP feasible solution is found
   - Solve NLP restriction.
   - Generate new linearization constraints.
   - Update open MILP relaxation subproblems.

Link OA, but only 1 MILP relaxation is solved, and updated in the tree search.
Hybrid Algorithms (Hyb)

For example, Bonami et al., 2008 *(BONMIN)*.

Very similar to Quesada and Grossmann, 1992, but NLP solved not only when the node is integer feasible but also, for example, any 10 nodes.

**Pros**: more “nonlinear” information added to the MILP relaxation.

**Cons**: More NLP solved.

Alternative,

Abhishek et al., 2010 *(FILMINT)*.

Very similar to Quesada and Grossmann, 1992, but add linearization cuts not only when the node is integer feasible (different strategies).

**Pros**: more “nonlinear” information added to the MILP relaxation.

**Cons**: MILP relaxation more difficult to solve.
<table>
<thead>
<tr>
<th></th>
<th># MILP</th>
<th># NLP</th>
<th>note</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>0</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>OA</td>
<td>1</td>
<td>1</td>
<td>weaker lower bound w.r.t. OA</td>
</tr>
<tr>
<td>GBD</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ECP</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>QG</td>
<td>1</td>
<td>1 + E</td>
<td>stronger lower bound w.r.t. QG</td>
</tr>
<tr>
<td>Hyb (Abhishek et al.)</td>
<td>1</td>
<td>1 + E</td>
<td></td>
</tr>
<tr>
<td>Hyb (Bonami et al.)</td>
<td>1</td>
<td>[E,N]</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Number of MILP and NLP subproblems solved by each algorithm.

where I is the # of iterations, N is the # of nodes, E is the # of explored MILP solutions
1. Intro to MINLP
2. What is a convex MINLP?
3. “Basic” subproblems
4. Convex MINLP Algorithms
   - Branch-and-Bound
   - Outer-Approximation
   - Generalized Benders Decomposition
   - Extended Cutting Plane
   - LP/NLP-based Branch-and-Bound
   - Hybrid Algorithms
5. Practical Tools
   - Convex MINLP Solvers
   - Modeling Languages
   - Neos
   - MINLP Libraries
   - Solvers’ performance
Convex MINLP Solvers

- **BONMIN**: [https://projects.coin-or.org/Bonmin](https://projects.coin-or.org/Bonmin)
- **DICOPT**: [www.gams.com/dd/docs/solvers/dicopt.pdf](http://www.gams.com/dd/docs/solvers/dicopt.pdf)
- **KNITRO**: [http://www.ziena.com/knitro.htm](http://www.ziena.com/knitro.htm)
- **MINLPBB**: [www-unix.mcs.anl.gov/~leyffer/solvers.htm](http://www-unix.mcs.anl.gov/~leyffer/solvers.htm)
- **SBB**: [http://www.gams.com/solvers/solvers.htm#SBB](http://www.gams.com/solvers/solvers.htm#SBB)
- **SCIP**: [http://scip.zib.de/](http://scip.zib.de/)

Problem: need value of the function, its first and its second derivative at a given point \((x^*, y^*)\).

Possible source of errors!

Solution? **Modeling Languages!**
Modeling Languages

Modeling languages, e.g., AMPL and GAMS.

Example:

```plaintext
param pi := 3.142;
param N;
set VARS ordered := {1..N};
param U {j in VARS};
param a {j in VARS};
param b {j in VARS};
param c {j in VARS};
param d {j in VARS};
param w{VARS};
param C;
var x {j in VARS} >= 0, <= U[j], integer;

maximize Total_Profit:
    sum {j in VARS} c[j]/(1+b[j]*exp(-a[j]*(x[j]+d[j])));

subject to KP_constraint: sum{j in VARS} w[j]*x[j] <= C;
```
NEOS: http://www.neos-server.org/neos/.
MINLP Libraries

- CMU/IBM: 42 different kind of MINLP problems
  http://www.minlp.org

- MacMINLP: ≃50 instances
  http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP

- MINLPlib: 274 instances
  http://www.gamsworld.org/minlp/minlplib.htm

The MINLPlib 2.0 is under construction: call for contribution (S. Vigerske)!
Solvers’ performance

Figure: Comparison of solvers in GAMS [Vigerske, 2012]

Best Algorithms are OA based!
Modeling Phase Importance

Thanks to P. Bonami for the picture

**Figure:** OA on 2 different formulations

Example:

\[
\begin{align*}
\text{min } & \quad c^T x \\
\sum_{i=1}^{n} (x_i - 0.5)^2 & \leq (n - 1)/4 \\
& \quad x \in \mathbb{Z}^n
\end{align*}
\]

\[
\begin{align*}
\text{min } & \quad c^T x \\
\sum_{i=1}^{n} y_i & \leq (n - 1)/4 \\
(x_i - 0.5)^2 & \leq y_i \quad \forall i = 1, \ldots, n \\
x \in \mathbb{Z}^n
\end{align*}
\]
MINLP modeling and real-world applications!!