# 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).

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

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

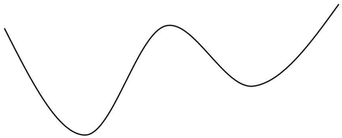
functions.

- Undecidable/NP-hard problem.
- Local optima might not be also global optima .

# Intro

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).

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

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

# NLP relaxation

$$\begin{aligned}
\min f(x, y) & & & \\
g(x, y) & \leq 0 & & \\
x & \in X & & \\
y & \in \{y \mid Ay \leq a\} & & \\
y_j & \leq \alpha_j^k & & j \in \{1, 2, \ldots, q\} \\
y_j & \geq \beta_j^k & & j \in \{1, 2, \ldots, q\}
\end{aligned}$$

$k$: 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 and Feasibility subproblem

NLP restriction for a fixed $y^k$:

$$\begin{aligned} \min f(x, y^k) & \\ g(x, y^k) &\leq 0 \\ x &\in X. \end{aligned}$$

Feasibility subproblem for a fixed $y^k$:

$$\begin{aligned} \min u & \\ g(x, y^k) &\leq u \\ x &\in X \\ u &\in \mathbb{R}. \end{aligned}$$

# MILP relaxation

$$\min \gamma$$

$$\begin{aligned} f(x^k,y^k) + \nabla f(x^k,y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} &\leq \gamma && \forall k \\ g_i(x^k,y^k) + \nabla g_i(x^k,y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} &\leq 0 && \forall k \; \forall i \in I^k \\ x &\in X \\ y &\in Y. \end{aligned}$$

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)

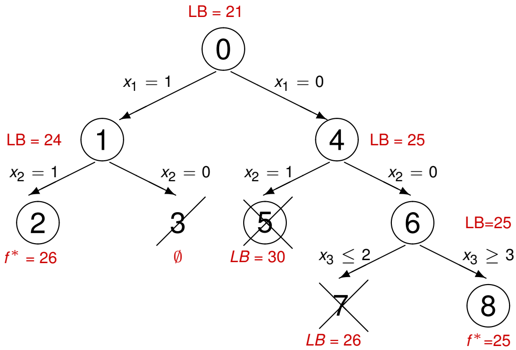
## Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$ where $P^0$ = NLP relaxation.
**while** $\Pi \neq \emptyset$ **do**
    Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
    Solve $P$ obtaining $(\bar{x}, \bar{y})$.
    **if** $P$ infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
        **break**
    **end if**
    **if** $\bar{y} \in \mathbb{Z}^q$ **then**
        $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
        Update $\Pi$ potentially fathoming subproblems.
    **else**
        Take a fractional value $\bar{y}_j$ and obtain two subproblems $P^1 = P$ with $\alpha_j^1 = \lfloor \bar{y}_j \rfloor$ and $P^2 = P$ with $\beta_j^2 = \lfloor \bar{y}_j \rfloor + 1$.
        $LB(P^1) = LB(P^2) = f(\bar{x}, \bar{y})$.
        $\Pi = \Pi \bigcup \{P^1, P^2\}$.
    **end if**
**end whileRETURN** $(x^*, y^*)$.

Fathoming is performed when:

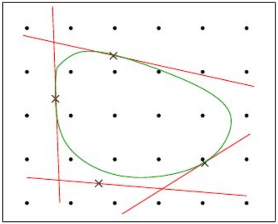
- The subproblem solution is MINLP feasible ($f^*$).
- The subproblem is infeasible.
- The subproblem $P^k$ has $LB(P^k) \geq f^*$.

# Branch-and-Bound (BB)



# Outer-Approximation (OA)

Duran and Grossmann, 1986.



$$
\begin{aligned}
& \min \gamma \\
f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq \gamma \quad \forall k \\
g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq 0 \quad \forall k\ \forall i \in I^k \\
x & \in X \\
y & \in Y.
\end{aligned}
$$

$I^k = \{1, 2, \ldots, m\}\ \forall k = 1, \ldots, K.$

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$= -\infty$.

```
while f* ≠ LB do
    Solve the current MILP relaxation (obtaining (x̂^K, ŷ^K)) and
    update LB.
    Solve the current NLP restriction for ŷ^K.
    if NLP restriction for ŷ^K infeasible then
        Solve the infeasibility subproblem for ŷ^K.
    else
        Update f*.
    end if
    Generate new linearization cuts and update MILP relaxation.
    K = K + 1.
end while
```

# Generalized Benders Decomposition (GBD)

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

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

**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)

Westerlund and Pettersson, 1995.

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.

# LP/NLP-based Branch-and-Bound (QG)

Quesada and Grossmann, 1992.

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.

# Summarizing...

| | # MILP | # NLP | note |
|---|---|---|---|
| BB | 0 | N | |
| OA | I | I | |
| GBD | I | I | weaker lower bound w.r.t. OA |
| ECP | I | 0 | |
| QG | 1 | 1 + E | |
| Hyb (Abhishek et al.) | 1 | 1 + E | stronger lower bound w.r.t. QG |
| Hyb (Bonami et al.) | 1 | [E,N] | |

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

# 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 Solvers

- ALPHA-ECP: http://www.gams.com/dd/docs/solvers/alphaecp.pdf
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: www.gams.com/dd/docs/solvers/dicopt.pdf
- FilMINT: www.mcs.anl.gov/~leyffer/papers/fm.pdf
- KNITRO: http://www.ziena.com/knitro.htm
- MINLPBB: www-unix.mcs.anl.gov/~leyffer/solvers.htm
- SBB: http://www.gams.com/solvers/solvers.htm#SBB
- SCIP: 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!

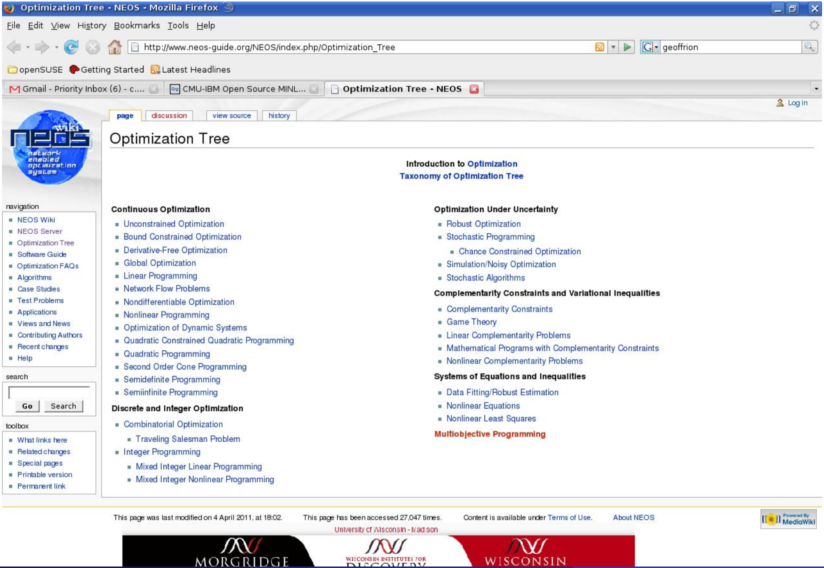
Modeling languages, e.g., AMPL and GAMS.
Example:

```
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

NEOS: `http://www.neos-server.org/neos/`.



# MINLP Libraries

- CMU/IBM: 42 different kind of MINLP problems
  `http://www.minlp.org`
- MacMINLP: $\simeq$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

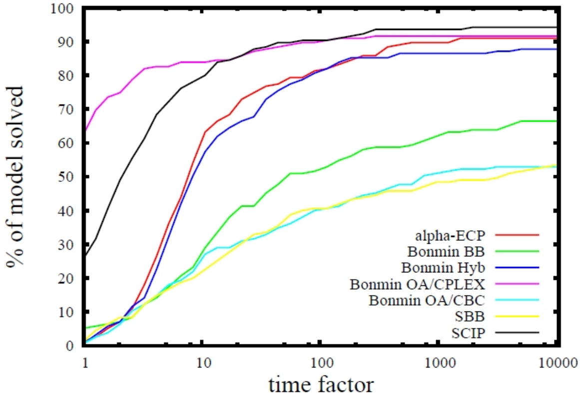



Figure: Comparison of solvers in GAMS [Vigerske, 2012]

Best Algorithms are OA based!

# Modeling Phase Importance

Thanks to P. Bonami for the picture

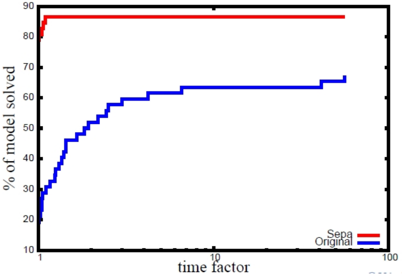



Figure: OA on 2 different formulations

Example:

$$\min c^T x$$
$$\sum_{i=1}^{n} (x_i - 0.5)^2 \leq (n-1)/4$$
$$x \in \mathbb{Z}^n$$

$$\min c^T x$$
$$\sum_{i=1}^{n} y_i \leq (n-1)/4$$
$$(x_i - 0.5)^2 \leq y_i \quad \forall i = 1, \ldots, n$$
$$x \in \mathbb{Z}^n$$

# MINLP modeling and real-world applications!!