A short tutorial on methods for mixed integer non linear programming

Claudia D'Ambrosio

LIX, CNRS & École Polytechnique
Institut Polytechnique de Paris

12ème journée du GT OR

Outline

Outline

Mixed Integer Non Linear Programming

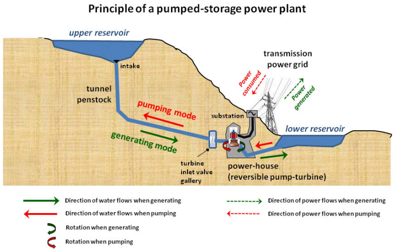
(MINLP)

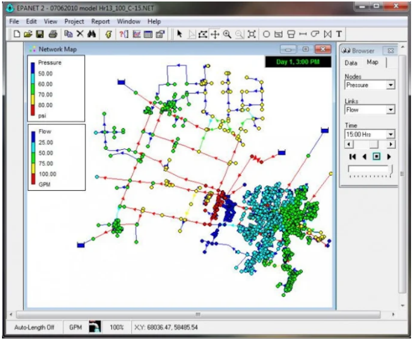
$$\begin{aligned} \min f(x,y) & \\ g_i(x,y) &\leq 0 \quad \forall i = 1,\dots,m \\ x &\in X \\ y &\in Y \end{aligned}$$

where $f(x,y) : \mathbb{R}^n \to \mathbb{R}$, $g_i(x,y) : \mathbb{R}^n \to \mathbb{R}\ \forall i = 1,\dots,m$, $X \subseteq \mathbb{R}^{n_1}$ $Y \subseteq \mathbb{N}^{n_2}$ and $n = n_1 + n_2$.

Hypothesis: f and g can be written in a **closed form** and are **twice continuously differentiable** functions.

Motivating Applications





Outline

What is a convex MINLP?

Convex Mixed Integer NonLinear Programming (MINLP).

$$
\begin{aligned}
\min f(x,y) & & \\
g(x,y) &\leq& 0 \\
x &\in& X = \{x \mid x \in \mathbb{R}^{n_1}, Dx \leq d, x^L \leq x \leq x^U\} \\
y &\in& Y = \{y \mid y \in \mathbb{Z}^{n_2}, Ay \leq a, y^L \leq y \leq y^U\}
\end{aligned}
$$

with $f(x,y) : \mathbb{R}^{n_1+n_2} \rightarrow \mathbb{R}$ and $g(x,y) : \mathbb{R}^{n_1+n_2} \rightarrow \mathbb{R}^m$ are

* continuous
* twice differentiable
* convex

functions.

- Local optima are also global optima.

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).

J. Kronqvist, D. E. Bernal, A. Lundell, I. E. Grossmann, A review and comparison of solvers for convex MINLP, **Optimization and Engineering**, 20 (2), pp. 397–455, 2019.

Outline

Branch-and-Bound (BB)

NLP relaxation

$$
\begin{aligned}
\min f(x,y) \\
g(x,y) &\leq 0 \\
x &\in X \\
y &\in \{y \mid Ay \leq a\} \\
y_j &\leq \alpha_j^k \qquad\qquad j \in \{1,2,\ldots,n_2\} \\
y_j &\geq \beta_j^k \qquad\qquad j \in \{1,2,\ldots,n_2\}
\end{aligned}
$$

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

Branch-and-Bound (BB)

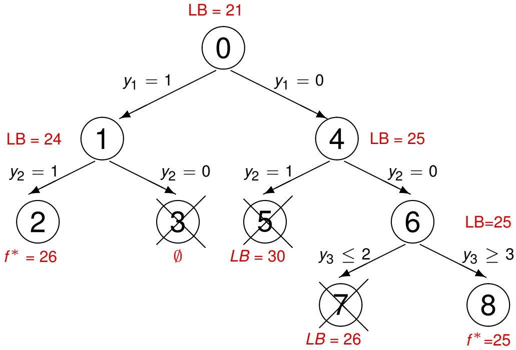
Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ 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**
        continue
    end if
    if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
        $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
        Update Π 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 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)



Branch-and-Bound (BB)

Proposition

If the functions f and g are convex and twice continuously differentiable, X and Y are bounded, it follows that branch-and-bound terminates at an optimal solution after searching a finite number of nodes (or that the instance is infeasible).

Proof.

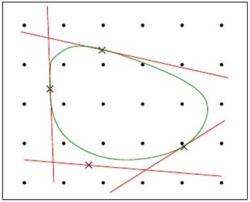
- Every NLP node can be solved to global optimality
- As X and Y are bounded, the B&B tree is finite
- Thus, similar proof for MILP B&B (see Th. 24.1 of Schrijver (1986)).

□

Outline

Outer-Approximation (OA)

Duran and Grossmann, 1986.



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

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

NB. The linearization constraints of MILP relaxation are not valid for non-convex MINLPs.

Outer-Approximation (OA)

MILP relaxation

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

where $I^k \subseteq \{1, 2, \ldots, m\}$. Two "classical" choices:

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

Outer-Approximation (OA)

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
  Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.
  Solve the current NLP restriction for y^K.
  if NLP restriction for y^K infeasible **then**
    Solve the infeasibility subproblem for y^K.
  else
    if $f(x^K, y^K) < f^*$ **then**
      $f^* = f(x^K, y^K)$, $(x^*, y^*) = (x^K, y^K)$.
    end if
  end if
  Generate linearization cuts, update MILP relax.
  $K = K + 1$.
end while
return (x^*, y^*)

NLP restriction and Feasibility subproblem

NLP restriction for a fixed y^k:

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

Infeasibility subproblem for a fixed y^k:

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

Worst-case complexity of outer approximation

Hijazi, Bonami, Ouorou. An Outer-Inner Approximation for separable MINLPs, INFORMS Journal on Computing (2014)

$$\min 0$$
$$\sum_{i=1}^{n} \left(x_i - \frac{1}{2}\right)^2 \leq \frac{n-1}{4}$$
$$x \in \{0,1\}^n$$

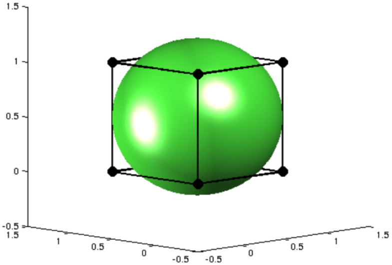


Figure: Source Belotti et al. (2013)

Outline

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). □

Outline

Extended Cutting Plane (ECP)

Westerlund and Pettersson, 1995.

$K = 1$, obtain an initial MILP relaxation.
while do
  Solve the MILP relaxation obtaining (x^K, y^K).
  if no constraint is violated by (x^K, y^K) **then**
    return (x^K, y^K) (optimal solution).
  else
    Generate (some) new linearization constraints from (x^K, y^K) and update MILP relaxation.
  end if
  $K = K + 1$.
end while

More iterations needed wrt OA.

Outline

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

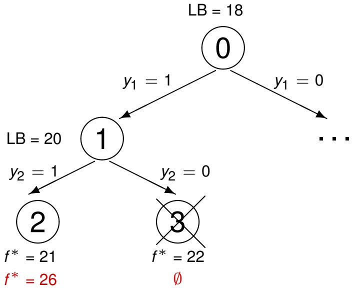
Quesada and Grossmann, 1992.

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.

Finite convergence as for BB.

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



Outline

Hybrid Algorithms (Hyb)

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

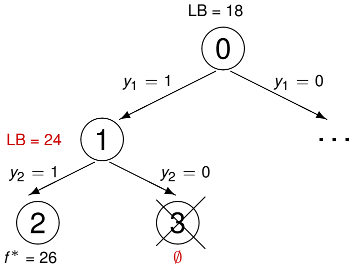
Very similar to Quesada and Grossmann, 1992, but NLP solved not only when the node is integer feasible but also, for example, any 10 nodes.

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.

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

E.g., Bonami et al., 2008 with NLP every 2 nodes.



Number of subproblems solved

	# MILP	# NLP	note
BB	0	# nodes	
OA	# iterations	# iterations	
GBD	# iterations	# iterations	1
ECP	# iterations	0	
QG	1	1 + # explored MILP solutions	
Hyb ALL10	1	1 + # explored MILP solutions	2
Hyb CMUIBM	1	[# explored MILP solutions,# nodes]	

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

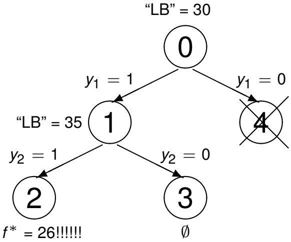
[1] weaker lower bound w.r.t. OA, MILP with less constraints than the one of OA

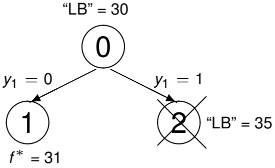
[2] stronger lower bound w.r.t. QG, MILP with more constraints than the one of QG

Outline

MINLP branch-and-bound with local NLP solver

Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers → local optimum
No valid bound for nonconvex MINLPs.



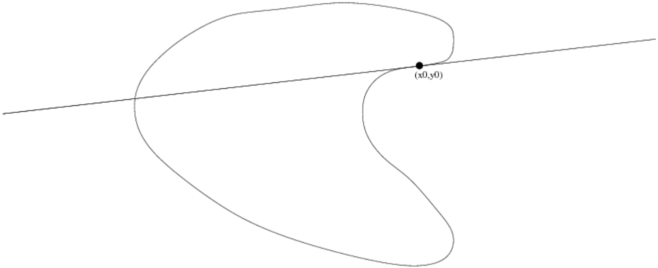


Outer Approximation and nonconvex MINLPs

Several methods for convex MINLPs use **Outer Approximation** cuts (Duran and Grossman, 1986) which are not exact for nonconvex MINLPs.

$$g_i(x) \leq 0 \quad \rightarrow \quad g_i(x^k) + \nabla g_i(x^k)^T \left(\ x - x^k \ \right) \leq 0$$

where $\nabla g(x^k)$ is the Jacobian of $g(x)$ evaluated at point (x^k).



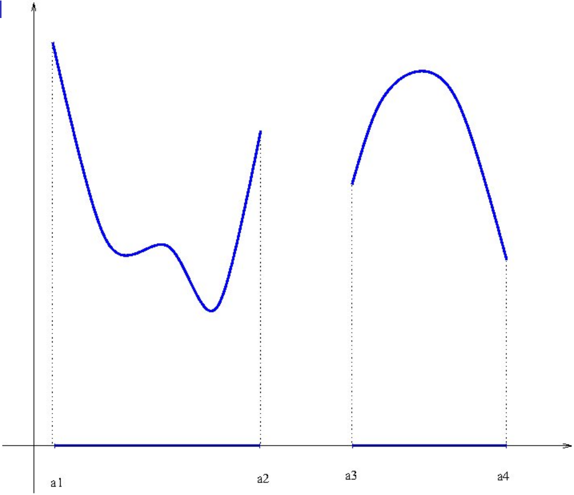
Outline

Spatial Branch-and-Bound

Falk and Soland (1969) "An algorithm for separable nonconvex programming problems".
25 years ago: first general-purpose "exact" algorithms for nonconvex MINLP.

- **Tree-like search**
- Explores search space exhaustively but **implicitly**
- Builds a sequence of **decreasing upper bounds** and **increasing lower bounds** to the global optimum
- Exponential worst-case
- Only general-purpose **"exact"** algorithm for MINLP
  Since continuous vars are involved, should say "**ε-approximate**"
- Like BB for MILP, but may branch on continuous vars
  Done whenever one is involved in a nonconvex term

Spatial B&B: Example



Original problem P

Spatial B&B: Pruning

1. P was branched into C_1, C_2
2. C_1 was branched into C_3, C_4
3. C_3 was **pruned by optimality**
   ($x^* \in \mathcal{G}(C_3)$ was found)
4. C_2, C_4 were **pruned by bound**
   (lower bound for C_2 worse than f^*)
5. No more nodes: whole space explored, $x^* \in \mathcal{G}(P)$

- Search generates a tree
- Suproblems are nodes
- Nodes can be pruned by optimality, bound or **infeasibility** (when subproblem is infeasible)
- Otherwise, they are branched

Spatial B&B: General idea

Aimed at solving "factorable functions", i.e., f and g of the form:

$$\sum_h \prod_k f_{hk}(x, y)$$

where $f_{hk}(x, y)$ are univariate functions $\forall h, k$.

- Exact reformulation of MINLP so as to have **"isolated basic nonlinear functions"** (additional variables and constraints).
- Relax (linear/convex) the basic nonlinear terms (**library of envelopes/underestimators**).
- Relaxation depends on variable bounds, thus **branching** potentially strengthen it.

Outline

Spatial B&B: exact reformulation to standard form

Consider a NLP for simplicity. Transform it in a **standard form** like:

$$
\begin{array}{rcll}
\min c^{\mathsf{T}}(x,w) & & & \\
A(x,w) & \leq & b & \\
w_{ij} & = & x_i \bigotimes x_j & \text{for suitable } i,j \\
x & \in & X & \\
w & \in & W &
\end{array}
$$

where, for example, $\bigotimes \in$ {**sum, product, quotient, power, exp, log, sin, cos, abs**} (Couenne).

Outline

Spatial B&B: convexification

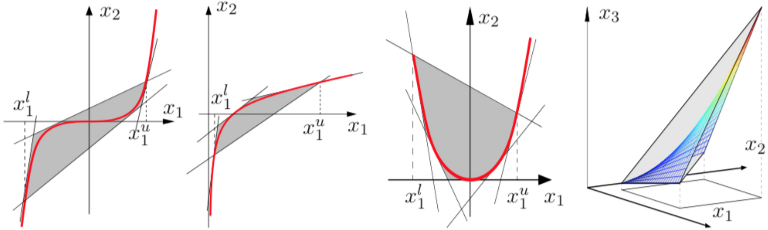
Relax $w_{ij} = x_i \bigotimes x_j$ $\forall$ suitable i, j where $\bigotimes \in$ {sum, product, quotient, power, exp, log, sin, cos, abs} such that:

$$\begin{aligned} w_{ij} &\leq \text{overestimator}(x_i \bigotimes x_j) \\ w_{ij} &\geq \text{underestimator}(x_i \bigotimes x_j) \end{aligned}$$

Convex relaxation is **not the tightest possible**, but **built automatically** .

- Underestimator/overestimator of convex/concave function: tangent cuts (OA)
- Odd powers or Trigonometric functions: separate intervals in which function is convex or concave and do as for convex/concave functions
- Product or Quotient: Mc Cormick relaxation

Spatial B&B: Examples of Convexifications



(a) $x_2 = x_1^3$ (b) $x_2 = \log x_1$ (c) $x_2 = x_1^2$ (d) $x_3 = x_1 x_2$

P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wächter, “Branching and bounds tightening techniques for non-convex MINLP”. Optimization Methods and Software 24(4-5): 597-634 (2009).

Example: Standard Form Reformulation

$$\begin{aligned} \min x_1^2 + x_1 x_2 & \\ x_1 + x_2 &\geq 1 \\ x_1 &\in [0, 1] \\ x_2 &\in [0, 1] \end{aligned}$$

becomes

$$\begin{aligned} \min w_1 + w_2 & \\ w_1 &= x_1^2 \\ w_2 &= x_1 x_2 \\ x_1 + x_2 &\geq 1 \\ x_1 &\in [0, 1] \\ x_2 &\in [0, 1] \end{aligned}$$

Convex hull of pieces weaker than the whole convex hull

Consider the following feasible set:

$$\begin{aligned} x_1^2 + x_2^2 &\geq 1 \\ x_1, x_2 &\in [0, 2] \end{aligned}$$

Convex hull: $x_1 + x_2 \geq 1$

Convex hull of standard form

$$\begin{aligned} x_3 + x_4 &\geq 1 \\ x_3 &\leq x_1^2 \\ x_4 &\leq x_1^2 \\ x_1, x_2 &\in [0, 2] \end{aligned}$$

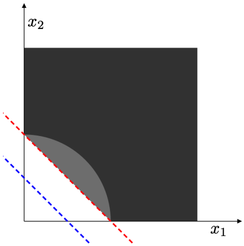


Figure: Source Belotti et al. (2013)

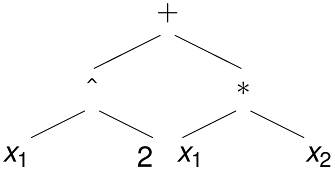
Outline

Expression trees

Representation of objective f and constraints g

Encode mathematical expressions in trees or DAGs

E.g. $x_1^2 + x_1 x_2$:

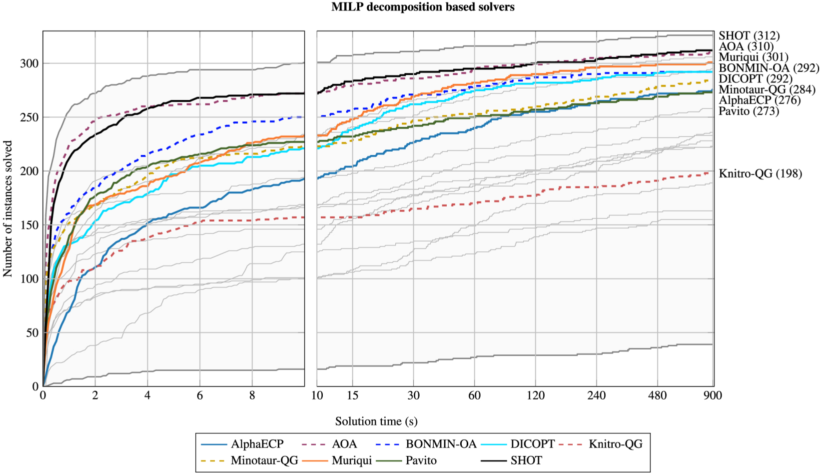


Outline

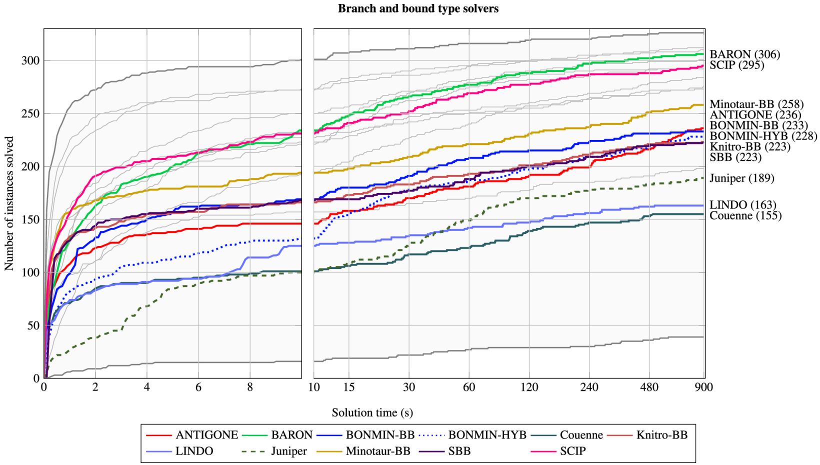
Convex MINLP Solvers

- ALPHA-ECP: `https://www.gams.com/latest/docs/S_ALPHAECP.html`
- AOA: `https://www.aimms.com/english/developers/resources/solvers/aoa`
- BONMIN: `https://projects.coin-or.org/Bonmin`
- DICOPT: `https://www.gams.com/24.8/docs/solvers/dicopt/index.html`
- FilMINT: `https://www.mcs.anl.gov/~leyffer/papers/fm.pdf`
- Juniper: `https://www.github.com/lanl-ansi/juniper.jl`
- LAGO: `https://projects.coin-or.org/LaGO`
- MINLPBB: `https://www-unix.mcs.anl.gov/~leyffer/solvers.htm`
- MINOTAUR: `https://wiki.mcs.anl.gov/minotaur`
- Muriqui: `http://www.wendelmelo.net/software`
- Pavito: `https://www.github.com/juliaopt/pavito.jl`
- SBB: `https://www.gams.com/latest/docs/S_SBB.html`
- SHOT: `https://github.com/coin-or/shot`
- ...

Convex MINLP solvers comparison



Convex MINLP solvers comparison



Nonconvex MINLP Solvers

```
10 Jul 2023 ========================================================
            Mixed Integer Nonlinear Programming Benchmark (MINLPLIB)
            ========================================================
                      H. Mittelmann (mittelmann@asu.edu)
```

The following codes were run through GAMS with a limit of 2 hours on these instances from MINLPLIB and with eight threads on an Intel i7-11700K, 64GB, 3.6GHz. All problems were solved GLOBALLY.

Description of selection process of benchmark instances. Statistics of the instances.

ANTIGONE, BARON, COUENNE, LINDO, OCTERACT, SCIP

Table for all solvers, Result files per solver, Log files per solver, Trace files per solver, Error files per solver

++
Unscaled and scaled shifted geometric means of run times

Feasibility tolerance set to 1e-6. All non-successes are counted as max-time.
The second line lists the number of problems (87 total) solved.

	ANTIGONE	BARON	COUENNE	LINDO	OCTERACT	SCIP
unscaled	1447.1	86.2	3304.4	1208.9	36.8	380.5
scaled	39.3	2.3	89.8	32.8	1.0	10.3
solved	53	77	24	42	87	64

Since Octeract will be removed from GAMS, it will be frozen at version 4.7.1

Modeling Languages

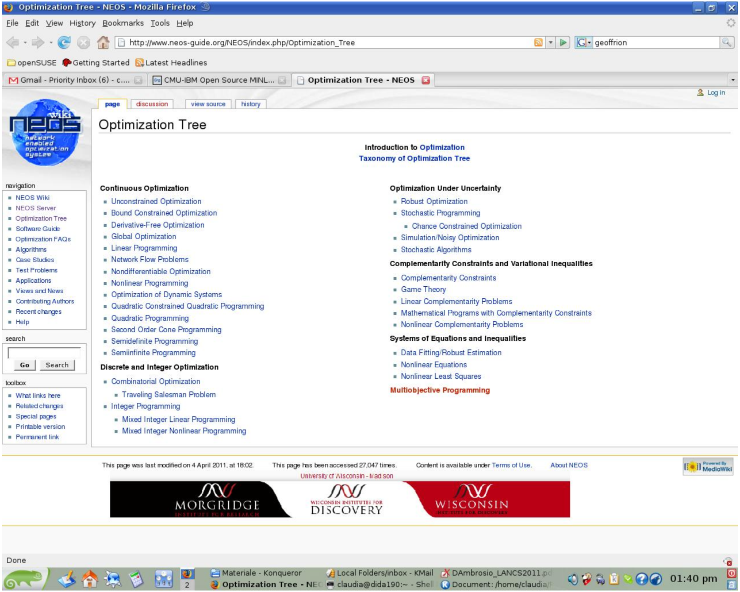
Modeling languages, e.g., AMPL, GAMS, JUMP, PYOMO, etc.
Example:

```
param N;
set VARS ordered := {1..N};
param Umax default 100;
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/`.



Outline

Conclusions

- Methods for **convex MINLP**
- Methods for **nonconvex MINLP**
- **Perspectives**
  - Best way to **reformulate** , then **convexify** ?
  - Tailored convexification techniques for **relevant classes of MINLP**
  - **Valid inequalities** to strengthen the convexification
  - **Branching strategies**
  - ...

References

- D'Ambrosio, Lodi. Mixed Integer Non-Linear Programming Tools: a Practical Overview, 4OR: A Quarterly Journal of Operations Research, 9 (4), pp. 329-349, 2011.
- Bonami, Kilinç, Linderoth, Algorithms and software for convex mixed integer nonlinear programs. In: Lee J, Leyffer S (eds) Mixed integer nonlinear programming. Springer, pp. 1–39, 2012.
- D'Ambrosio, Lodi. Mixed integer nonlinear programming tools: an updated practical overview, Annals of Operations Research, 204, pp. 301–320, 2013.
- Belotti, Kirches, Leyffer, Linderoth, Luedtke, Mahajan, Mixed-integer nonlinear optimization. Acta Numerica, 22, pp. 1–131, 2013.
- Kronqvist, Bernal, Lundell, Grossmann, A review and comparison of solvers for convex MINLP, Optimization and Engineering, 20 (2), pp. 397–455, 2019.
- Sherali, Alameddine, A new reformulation-linearization technique for bilinear programming problems, JOGO, 1991
- Falk, Soland, An algorithm for separable nonconvex programming problems, Manag. Sci. 1969.
- Horst, Tuy, Global Optimization, Springer 1990.
- Ryoo, Sahinidis, Global optimization of nonconvex NLPs and MINLPs with applications in process design, Comp. Chem. Eng. 1995.
- Adjiman, Floudas et al., A global optimization method, αBB, for general twice-differentiable nonconvex NLPs, Comp. Chem. Eng. 1998.
- Smith, Pantelides, A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs, Comp. Chem. Eng. 1999.
- Nowak, Relaxation and decomposition methods for Mixed Integer Nonlinear Programming, Birkhäuser, 2005.
- Belotti, et al., Branching and bounds tightening techniques for nonconvex MINLP, Opt. Meth. Softw., 2009.
- Vigerske, PhD Thesis: Decomposition of Multistage Stochastic Programs and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming, Humboldt-University Berlin, 2013.