Nonlinear Mixed-Integer Optimization

MINO/COST SPRING SCHOOL ON OPTIMIZATION

Jon Lee — Andrea Lodi

University of Michigan — University of Bologna

10 April 2014

Outline for the Day

- 9:15-10:15. lecture (Lee): Introduction to MINLP // Complexity of MINLP: Hardness and polynomial tractability
- 10:15-11:00. coffee break
- 11:00-12:00. lecture (Lodi): General-purpose algorithms for convex and non-convex MINLP
- 12:00-14:15. lunch
- 14:15-15:00. lecture (Lee): Non-convex quadratic MINLP
- 15:00-15:30. coffee break
- 15:30-16:15. lecture (Lodi): Software and computational advances
- 16:15-16:30. short break
- 16:30-17:00. problem session

Mixed Integer Nonlinear Programming algorithms

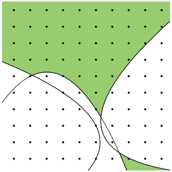
Andrea Lodi (and Pierre Bonami)

DEI, University of Bologna

MINO-COST Training School – Klagenfurt – April 10, 2014

"The mother of all deterministic optimization problems" [Lee, 2008]

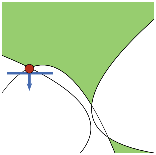
$$
\begin{array}{lll}
\min & f(x) & \\
\text{s.t.} & g_i(x) \leq 0 & i = 1, \dots, m \\
& x \in X & \\
& x_j \in \mathbb{Z} & j = 1, \dots, p \\
& l_j \leq x_j \leq u_j & j = 1, \dots, p
\end{array}
\qquad \text{(MINLP)}
$$

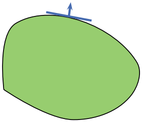


- $X \subseteq \mathbb{R}^n$ polyhedral.
- f and $g_i : X \rightarrow \mathbb{R}$, $i = 1, \dots, m$, continuous, differentiable.

Nonlinear Programming (NLP)

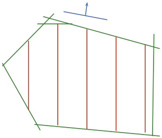
$p = 0$: local optima. $\quad + f$ and g_i convex $\Rightarrow$ global optima.



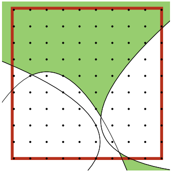


Mixed-Integer linear programming (MILP)

- f linear, $m = 0$, $p > 0$



$$
\begin{array}{lll}
\min & f(x) & \\
\text{s.t.} & g_i(x) \leq 0 & i = 1, \ldots, m \\
& x \in X & \\
& x_j \in \mathbb{Z} & j = 1, \ldots, p \\
& l_j \leq x_j \leq u_j & j = 1, \ldots, p
\end{array}
\qquad \text{(MINLP)}
$$



- **Solvable, in general,** l_j, u_j **finite**.

Mixed Integer Convex Programming

Assume that the continuous relaxation is a convex optimization problem.

- f is a convex function.
- g_i are either convex function or describe a convex feasible region (for example, second order cone constraints: $\sum x_j^2 \leq x_0^2$)

Mixed Integer Nonlinear Programming

Do not assume any convexity on f or g_i.

- Continuous relaxation is NP-hard to solve in general.
- Remark: if l_j and u_j are finite integer variable can be seen as a continuous satisfying

$$(x_j - l_j)(x_j - l_j - 1)\dots(x_j - u_j) = 0$$

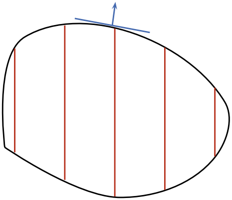
Application	nonlinear	discrete
Portfolio optimization	Risk, utility, robustness	number of assets, min investment
[Bienstock, 1996, Bonami and Lejeune, 2009, Vielma et al., 2008]		
Chemical plant design	Chemical reactions	what to install
[Duran and Grossmann, 1986, Flores-Tlacuahuac and Biegler, 2007]		
Block Layout Design	Spatial constraints	what to layout
[Castillo et al., 2005]		
Networks with delays	Delay as function of traffic	Path, flows
[Boorstyn and Frank, 1977, Ameur and Ouorou, 2006]		
Location with stochastic services	Demands	location model
[Elhedhli, 2006]		
TSP with neighborhoods (Robotics)	Definition of ngbh.	TSP
[Gentilini et al., 2013]		

Application	nonlinear	discrete
Petrochemical	Blending, pooling	Which process
[Haverly, 1978]		
Gas/Water networks	Pressure loss	Network topology
[Bragalli et al., 2011]		
Nuclear Reactor reloading	reactions	What to reload
[Quist et al., 1999]		
Airplane trajectory optimization	aerodynamics	waypoints, colision avoidance,...
[Cafieri and Durand, 2013, Soler et al., 2013]		
Mixed Integer Optimal control	DE	discrete controls
[Sager, 2005, 2012]		
Countless more	...	...
see for example [Belotti et al., 2013]		

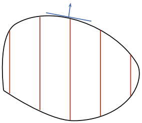
- The convex case
  - Main algorithmic approaches.
  - Glimpse at computation.
- A step into nonconvexity
  - MIQP with CPLEX 12.6.
  - Basic setup of a spatial branch-and-bound.
  - Computational results.
- Arbitrary selection of more advanced topics
  - Separability.
  - Disjunctive Cuts.
- Conclusions
- Problem Session
- Bibliography

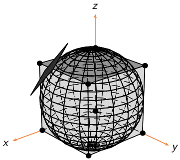
The mixed integer convex program

$$
\begin{array}{lll}
\textbf{min} & c^T x & \\
\text{s.t.} & g_i(x) \leq 0 & i = 1, \dots, m \\
& x \in X & \\
& x_j \in \mathbb{Z} & j = 1, \dots, p \\
& l_j \leq x_j \leq u_j & , j = 1, \dots, p
\end{array}
\quad \text{(MICP)}
$$



- $g_i : X \to \mathbb{R}$, $i = 1, \dots, m$, convex, differentiable.
- Assume linear objective. If necessary, add var $\alpha \in \mathbb{R}$ and $\min \alpha$ with $f(x) \leq \alpha$ as a constraint.





Fundamental property is convexity of the continuous relaxation, which can be efficiently solved.

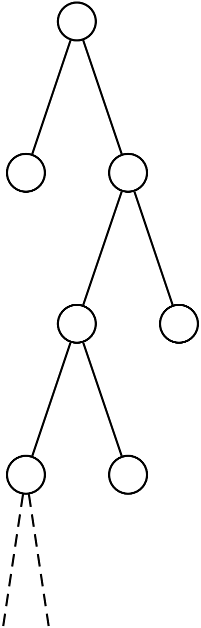
1. NLP Branch-and-bound [Gupta and Ravindran, 1985].
2. Outer Approximation Algorithm [Duran and Grossmann, 1986]. Builds an MILP equivalent of (MICP)
3. LP/NLP branch-and-cut [Quesada and Grossmann, 1992].

Straightforward generalization of main MILP algorithm:

- solve an NLP at each node of the tree.

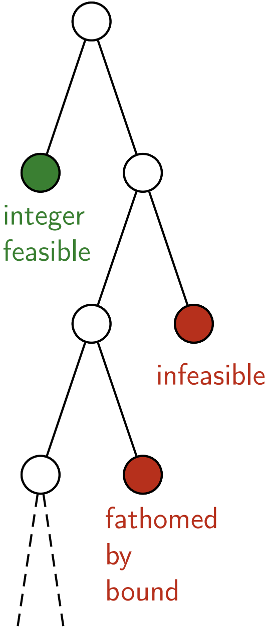
Straightforward generalization of main MILP algorithm:

- solve an NLP at each node of the tree.
- Branch on variables with fractional value.



Straightforward generalization of main MILP algorithm:

- solve an NLP at each node of the tree.
- Branch on variables with fractional value.
- Prune by infeasibility, bounds and integer feasibility.

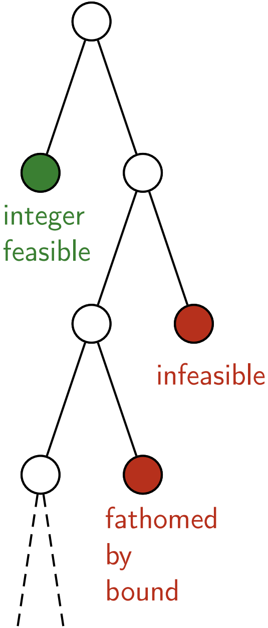


Straightforward generalization of main MILP algorithm:

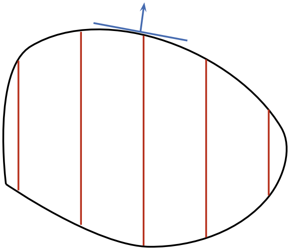
- solve an NLP at each node of the tree.
- Branch on variables with fractional value.
- Prune by infeasibility, bounds and integer feasibility.

Main issues

- Warm-starting of NLP solves.
- Difficulty of reusing MILP technologies.



Outer Approximation [Duran and Grossmann, 1986]



$$\min \quad c^T x$$

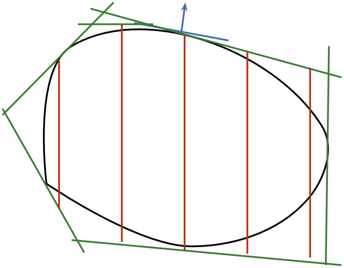
s.t.

$$g_i(x) \le 0 \qquad i = 1, \dots, m,$$

$$x_j \in \mathbb{Z} \qquad j = 1, \dots, p.$$

Idea: Take first-order approximations of constraints at different points and build an equivalent MILP.

Outer Approximation [Duran and Grossmann, 1986]



$$\min \quad c^T x$$

s.t.

$$g_i(x) \leq 0 \qquad i = 1, \ldots, m,$$

$$x_j \in \mathbb{Z} \qquad j = 1, \ldots, p.$$

Idea: Take first-order approximations of constraints at different points and build an equivalent MILP.

$$\min \quad c^T x$$

s.t.

$$g_i(\overline{x}^k) + \nabla g_i(\overline{x}^k)^T (x - \overline{x}^k) \leq 0 \qquad i = 1, \ldots, m,\ k = 1, \ldots, K$$

$$x_j \in \mathbb{Z} \qquad j = 1, \ldots, p.$$

Given $\hat{x} \in \mathbb{Z}^p$:

fixed NLP (NLP($\hat{x}$))

$$\begin{aligned} & \min\ c^T x \\ & s.t. \\ & g_i(x) \le 0 \qquad i = 1, \dots, m \\ & x \in X \qquad\qquad (\text{NLP}(\hat{x})) \\ & x_j = \hat{x}_j \qquad j = 1, \dots, p. \end{aligned}$$

If $\hat{x} \in \mathbb{Z}^p$, and feasible: gives an upper bound.

fixed feasibility subproblem

$$\begin{aligned} & \min \sum_{i=1}^{m} w_i \max\{0, g_i(x)\} \\ & \text{s.t.} \\ & x \in X, \qquad\qquad (\text{NLPF}(\hat{x})) \\ & x_j = \hat{x}_j,\ j = 1, \dots, p \end{aligned}$$

Given $\hat{x} \in \mathbb{Z}^p$:

fixed NLP (NLP($\hat{x}$))

$$\begin{aligned} & \min\ c^T x \\ & s.t. \\ & g_i(x) \leq 0 \qquad i = 1, \dots, m \\ & x \in X \qquad\qquad (\text{NLP}(\hat{x})) \\ & x_j = \hat{x}_j \qquad j = 1, \dots, p. \end{aligned}$$

If $\hat{x} \in \mathbb{Z}^p$, and feasible: gives an upper bound.

fixed feasibility subproblem

$$\begin{aligned} & \min \sum_{i=1}^{m} w_i \max\{0, g_i(x)\} \\ & \text{s.t.} \\ & x \in X, \qquad\qquad (\text{NLPF}(\hat{x})) \\ & x_j = \hat{x}_j,\ j = 1, \dots, p \end{aligned}$$

Remark: If (NLP($\hat{x}$)) is infeasible, NLP software will typically return a solution to (NLPF($\hat{x}$)). By abuse, always say solution to (NLP($\hat{x}$))

For each $\hat{x}^k \in K = \text{Proj}_{1,\dots,p}(X) \cap \mathbb{Z}^p$, let $\overline{x}^k$ be an optimal solution to (NLP($\hat{x}$)).

Theorem ([Duran and Grossmann, 1986])

If $X \neq \emptyset$, f and g are convex, continuously differentiable, and a constraint qualification holds for each $\overline{x}^k$ then

$$\begin{aligned} \min \quad & c^T x \\ & g_i(\overline{x}^k) + \nabla g_i(\overline{x}^k)^T (x - \overline{x}^k) \leq 0 \quad i = 1, \dots, m, \hat{x}^k \in K, \\ & x \in X,\ x_j \in \mathbb{Z},\ j = 1, \dots, p. \end{aligned}$$

has the same optimal value as (MICP).

Generate MILP equivalent by constraint generation.

- Initialize with one set of linearizations.

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & \\ & g_i(\overline{x}^0) + \nabla g_i(\overline{x}^0)^T (x - \overline{x}^0) \leq 0, \quad i = 1, \ldots, m, \quad , \\ & x \in X,\ x_j \in \mathbb{Z},\ j = 1, \ldots, p. \end{array} \qquad (\text{OA}(\mathcal{K}))$$

Where x^0 is the solution to the continuous relaxation:
$\min\{c^T x : x \in X,\ g_i(x) \leq 0, i = 1, \ldots, m\}$

Generate MILP equivalent by constraint generation.

- Initialize with one set of linearizations.
- Enrich iteratively the set of linearizations $\mathcal{K}$.

$$
\begin{aligned}
&\min \quad c^T x \\
&\text{s.t.} \\
&g_i(\overline{x}^k) + \nabla g_i(\overline{x}^k)^T (x - \overline{x}^k) \le 0, \qquad \begin{array}{c} i = 1, \dots, m, \\ \hat{x}^k \in \mathcal{K} \end{array}, \qquad (\text{OA}(\mathcal{K})) \\
&x \in X,\ x_j \in \mathbb{Z},\ j = 1, \dots, p.
\end{aligned}
$$

Where $\hat{x}^k$ is a solution to $(\text{OA}(\mathcal{K}))$ and, for $k = 1, \dots, |\mathcal{K}|$, $\overline{x}^k$ is the solution to $(\text{NLP}(\hat{x}))$.

Generate MILP equivalent by constraint generation.

- Initialize with one set of linearizations.
- Enrich iteratively the set of linearizations $\mathcal{K}$.

Convergence

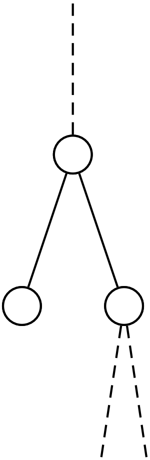
At each iteration:

- (OA($\mathcal{K}$)) gives a lower bound,
- If feasible, (NLP($\hat{x}$)) gives an upper bound.

The theorem guarantees that the two bounds converge in finite # of iterations.

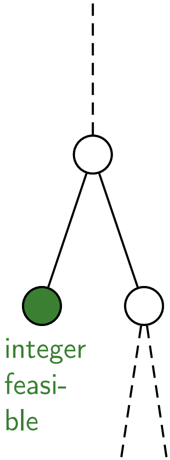
OA can be embedded in a single tree search.

- Start solving the same initial MILP by branch and bound.
- At each integer feasible node:



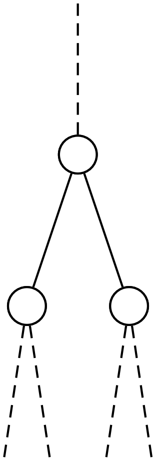
OA can be embedded in a single tree search.

- Start solving the same initial MILP by branch and bound.
- At each integer feasible node:
  1. solve (NLP($\hat{x}$)), and enrich the set of linearizations.
  2. Resolve the LP relaxation of the node with the new cuts.
  3. Repeat as long as node is integer feasible.



OA can be embedded in a single tree search.

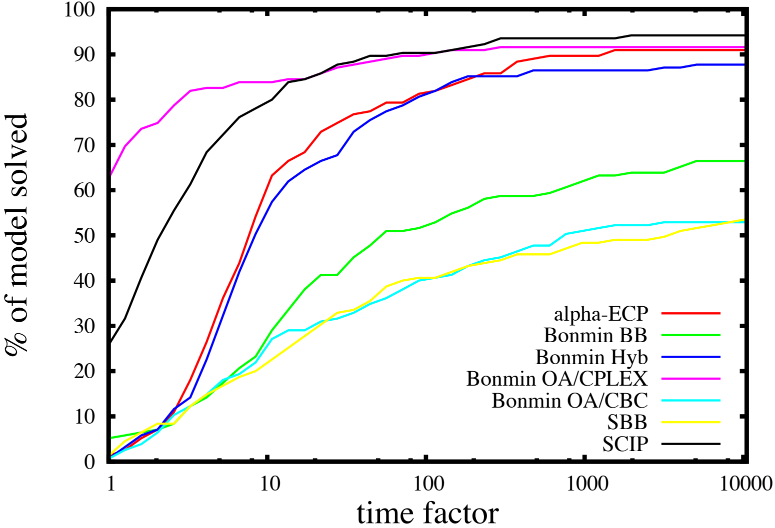
- Start solving the same initial MILP by branch and bound.
- At each integer feasible node:
  1. solve (NLP($\hat{x}$)), and enrich the set of linearizations.
  2. Resolve the LP relaxation of the node with the new cuts.
  3. Repeat as long as node is integer feasible.
- Never prune by integer feasibility.



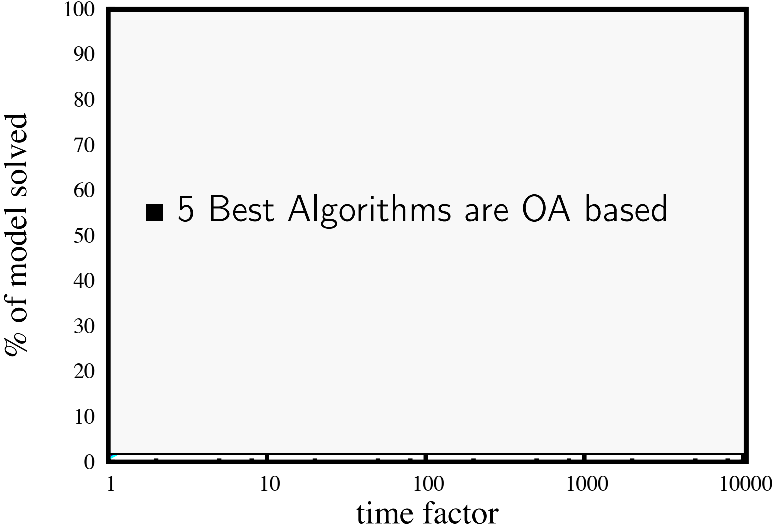
Solvers for Mixed Integer Convex Programs

Solver	Reference	Algorithm(s)
Dicopt		OA
MINLP_BB	[Leyffer, 1998]	NLP BB
SBB	[Bussieck and Drud, 2001]	NLP BB
α-ECP	[Westerlund and Lundqvist, 2005]	ECP (variant of OA)
Bonmin	[Bonami et al., 2008]	NLP BB, OA, LP/NLP
FilMINT	[Abhishek et al., 2010]	LP/NLP
KNITRO	[Byrd et al., 2006]	NLP BB, LP/NLP
SCIP	[Vigerske, 2013]	LP/NLP

Comparison of solvers in GAMS [Vigerske, 2012]



Comparison of solvers in GAMS [Vigerske, 2012]



- 5 Best Algorithms are OA based

- Bonmin's OA using CPLEX seems the best algorithm overall.
  - It is also the simplest: a loop calling CPLEX (MILP) and Ipopt (NLP) alternatively as black boxes.
  - Improves with CPLEX.
- Bonmin's Hyb is in the pack of relatively good solvers
  - own variant of LP/NLP BB.
  - Reuse CBC infrastructure, LP solver, Cuts, MIP presolve.
  - Improves at a slower pace.
- Bonmin's BB clearly behind.
  - pure NLP based branch-and-bound. Does not reuse much from Cbc. Everything specifically tailored.
  - Better implementation exists that should be on par with Hyb.

- The convex case
  - Main algorithmic approaches.
  - Glimpse at computation.
- A step into nonconvexity
  - MIQP with CPLEX 12.6.
  - Basic setup of a spatial branch-and-bound.
  - Computational results.
- Arbitrary selection of more advanced topics
  - Separability.
  - Disjunctive Cuts.
- Conclusions
- Problem Session
- Bibliography

(MI)QP

$$
\begin{aligned}
\min \quad & \frac{1}{2} x^T Q x + c^T x \\
s.t. \quad & \\
& Ax = b && \qquad \text{(MIQP)} \\
& x_j \in \mathbb{Z} \qquad j = 1, \ldots, p \\
& l \leq x \leq u
\end{aligned}
$$

(with Q symmetric),

Andrea Lodi, University of Bologna

$$\min \frac{1}{2}x^T Qx + c^T x$$

$$s.t.$$

$$Ax = b \qquad \text{(MIQP)}$$

$$x_j \in \mathbb{Z} \qquad j = 1, \ldots, p$$

$$l \leq x \leq u$$

(with Q symmetric),

History of MIQP with CPLEX

class	p	Q	algorithme	V. (Year)
Convex QP	0	$\succeq 0$	barrier	4.0 (1995)
–	–	–	QP simplex	8.0 (2002)
convex MIQP	> 0	$\succeq 0$	B&B	8.0 (2002)
nonconvex QP	0	$\not\succeq 0$	barrier (local)	12.3 (2011)
–	–	–	spatial B&B (global)	12.6 (2013)
nonconvex MIQP	> 0	$\not\succeq 0$	spatial B&B (global)	12.6 (2013)

Let $G = (N, E)$ be a graph and Q be the incidence matrix of G. The optimal value of:

$$\min \frac{1}{2} x^T Q x$$

$$s.t.$$

$$\sum_{j \in N} x_j = 1$$

$$x \geq 0.$$

is $\frac{1}{2}\left(1 - \frac{1}{\chi(G)}\right)$ where $\chi(G)$ is the clique number of G [Motzkin and Straus, 1965],

- $\Rightarrow$ QP is NP-hard
- More generally QPs on the simplex (general Q) can be solved by a nonlinear maximum clique algorithm [Scozzari and Tardella, 2008].

- Primal Dual Interior Point Algorithm.
- Available since IBM CPLEX 12.3.
- Not enabled by default, if Q is indefinite CPLEX will return `CPXERR_Q_NOT_POS_DEF`.
- Activated by setting the option `solution target` to 2 (or `CPX_SOLUTIONTARGET_FIRSTORDER`).
- Approach used by Ipopt but no need for
  - Feasibility restoration
  - Second order correction
  - Filter
- Own implementation of indefinite factorization.

Global (MI)QP

- Activated by setting `solution target` to 3 (or `CPX_SOLUTIONTARGET_OPTIMALGLOBAL`).
- Note: previous versions could already solve some nonconvex MIQPs (pure 0-1 QPs, convex after presolve...)

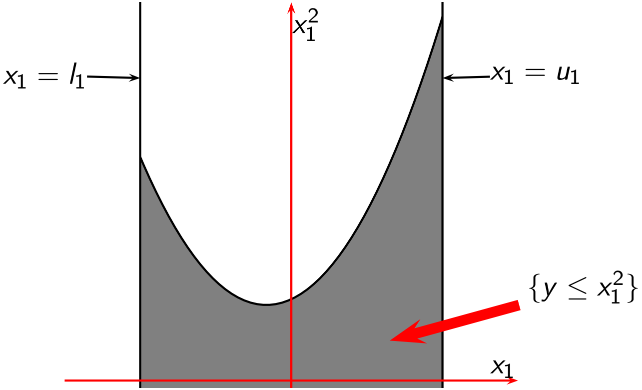
Notes on complexity

- Checking if a feasible solution is not a local minimum is coNP-Complete.
- Checking if a nonconvex QP is unbounded is NP-complete.

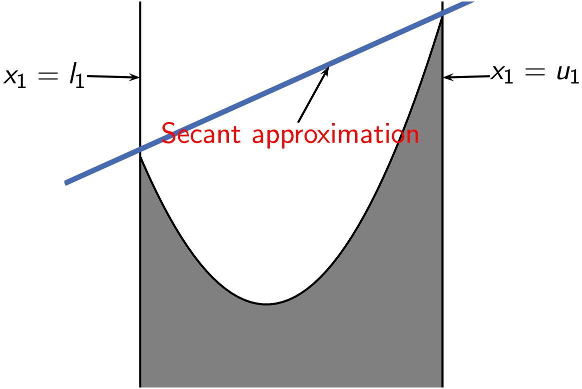
B&B spatial

- Establish a convex (easily solvable) relaxation.
- Establish branching rules on solutions of this relaxation.

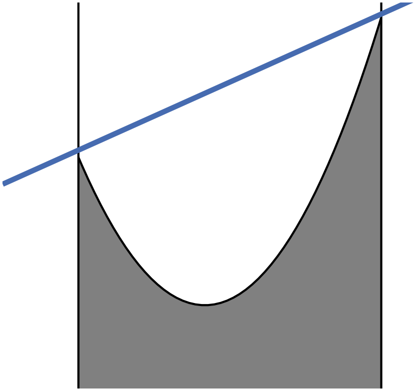
The convex hull relaxations of a square term x_1^2



The convex hull relaxations of a square term x_1^2

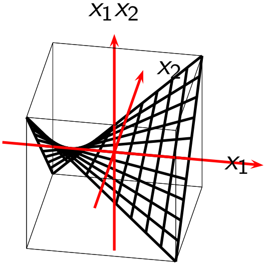


The convex hull relaxations of a square term x_1^2



$$x_1^2 \leq y_{ii}^+ := (l_1 + u_1)x_1 - l_1 u_1$$

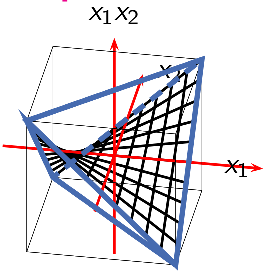
The convex hull relaxations of a single product $x_1 x_2$ [McCormick, 1976]



The convex hull relaxations of a single product $x_1 x_2$ [McCormick, 1976]

$$x_1 x_2 \geq y_{12}^{-} := \max \left\{ \begin{array}{r} u_2 x_1 + u_1 x_2 - u_1 u_2 \\ l_2 x_1 + l_1 x_2 - l_1 l_2 \end{array} \right\}$$

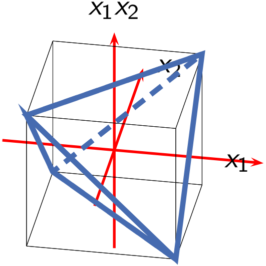
$$x_1 x_2 \leq y_{12}^{+} := \min \left\{ \begin{array}{l} u_2 x_1 + l_1 x_2 - l_1 u_2 \\ l_2 x_1 + u_1 x_2 - u_1 l_2 \end{array} \right\}$$



The convex hull relaxations of a single product x_1x_2 [McCormick, 1976]

$$x_1x_2 \geq y_{12}^- := \max \begin{Bmatrix} u_2x_1 + u_1x_2 - u_1u_2 \\ l_2x_1 + l_1x_2 - l_1l_2 \end{Bmatrix}$$

$$x_1x_2 \leq y_{12}^+ := \min \begin{Bmatrix} u_2x_1 + l_1x_2 - l_1u_2 \\ l_2x_1 + u_1x_2 - u_1l_2 \end{Bmatrix}$$



- Depending on the sign of q_{ij} we only need y^+ or y^-.
- For simplicity, we assume we put all in the remainder.

- Let $Q = P + \tilde{Q}$ with P the diagonal psd matrix containing $q_{ii} > 0$.

$$
\begin{aligned}
& \min \ \frac{1}{2}x^T P x + \frac{1}{2}x^T \tilde{Q} x + c^T x \\
& s.t. \\
& Ax = b \\
& x_j \in \mathbb{Z} \quad j = 1, \ldots, p \\
& l \leq x \leq u
\end{aligned}
\qquad \text{(MIQP)}
$$

- Let $Q = P + \tilde{Q}$ with P the diagonal psd matrix containing $q_{ii} > 0$.
- Add one $y_{ij} = x_i x_j$ variable for each non-zero entry q_{ij} of $\tilde{Q}$.

$$
\begin{aligned}
& \min \ \frac{1}{2} x^T P x + \frac{1}{2} \langle \tilde{Q}, Y \rangle + c^T x \\
& s.t. \\
& Ax = b \\
& x_j \in \mathbb{Z} \quad j = 1, \ldots, p \\
& Y = x x^T \\
& l \le x \le u
\end{aligned}
\qquad \text{(MIQP)}
$$

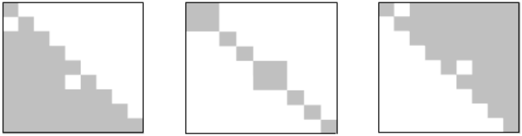
($\langle Q, Y \rangle = \sum_{i,j} q_{ij} y_{ij}$)

- Let $Q = P + \tilde{Q}$ with P the diagonal psd matrix containing $q_{ii} > 0$.
- Add one $y_{ij} = x_i x_j$ variable for each non-zero entry q_{ij} of $\tilde{Q}$.
- Relax $y_{ij} = x_i x_j$ using McCormik and Secant approximations.

$$
\begin{aligned}
& \min \ \frac{1}{2} x^T P x + \frac{1}{2} \langle \tilde{Q}, Y \rangle + c^T x \\
& s.t. \\
& Ax = b \\
& x_j \in \mathbb{Z} \quad j = 1, \ldots, p \\
& y_{ij}^- \le y_{ij} \le y_{ij}^+ \\
& y_{ii} \le y_{ii}^+ \\
& l \le x \le u
\end{aligned}
\tag{q-MIQP}
$$

Factorizations of Q

- CPLEX own block indefinite decomposition: M and B such that M 2-block triangular and B 2-blocks diagonal with $Q = M^T BM$



- Reformulate $x^T Qx$ using additional variables z so that $z^T Dz = x^T Bx$ and D diagonal. Let L, D give the spectral decomposition of B, $z = L\zeta$, $\zeta = Mx$.

(For simplicity assume $z = Lx$ gives the system we want)

Use a decomposition to get $z = Lx$ and $z^T Dz = x^T Qx$ and do the same steps as before (but more simple)....

$$\begin{aligned} &\min \ \frac{1}{2} z^T Dz + c^T x \\ &s.t. \\ &Ax = b, Lx = z \\ &x_j \in \mathbb{Z} \qquad\qquad j = 1, \ldots, p \\ &l \leq x \leq u \end{aligned} \tag{MIQP}$$

Use a decomposition to get $z = Lx$ and $z^T Dz = x^T Qx$ and do the same steps as before (but more simple)....

- Let $D = D^+ - D^-$ with $D^\pm$ diagonal psd matrices.

$$\begin{aligned}
& \min \ \frac{1}{2}(z^T D^+ z - z^T D^- z) + c^T x \\
& s.t. \\
& Ax = b, Lx = z \\
& x_j \in \mathbb{Z} \qquad\qquad j = 1, \ldots, p \\
& l \leq x \leq u
\end{aligned} \qquad \text{(MIQP)}$$

Factorized Eigenvector space reformulation and relaxation

Use a decomposition to get $z = Lx$ and $z^T Dz = x^T Qx$ and do the same steps as before (but more simple)....

- Let $D = D^+ - D^-$ with $D^\pm$ diagonal psd matrices.
- Add $y_{ii} \leq z^2$ variable for each non-zero of D^-.

$$
\begin{aligned}
\min\ & \frac{1}{2} z^T D^+ z - \sum_{i=1}^{n} \frac{d_{ii}}{2} y_{ii} + c^T x \\
s.t. & \\
& Ax = b, Lx = z \\
& x_j \in \mathbb{Z} \qquad j = 1, \ldots, p \\
& y_{ii} \leq z_i^2 \\
& l \leq x \leq u
\end{aligned}
\tag{MIQP}
$$

Use a decomposition to get $z = Lx$ and $z^T Dz = x^T Qx$ and do the same steps as before (but more simple)....

- Let $D = D^+ - D^-$ with $D^\pm$ diagonal psd matrices.
- Add $y_{ii} \leq z^2$ variable for each non-zero of D^-.
- Infer finite bounds, l^z, u^z for z and relax $y_{ii} \leq z_i^2$ using Secant approximations.

$$
\begin{aligned}
& \min \ \frac{1}{2} z^T D^+ z - \sum_{i=1}^{n} \frac{d_{ii}}{2} y_{ii} + c^T x \\
& s.t. \\
& Ax = b, Lx = z \\
& x_j \in \mathbb{Z} \qquad\qquad j = 1, \ldots, p \\
& y_{ii} \leq y_{ii}^+ \\
& l \leq x \leq u, l^z \leq z \leq u^z
\end{aligned}
\qquad \text{(ev-MIQP)}
$$

Notes on the two relaxations

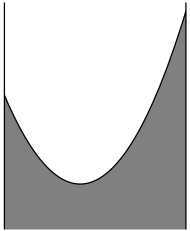
The steps are almost the same.

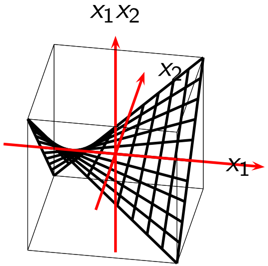
- If Q is diagonal the two relaxations are identical.
- In general they are not comparable.
- If $Q \succeq 0$, EV-space is better it preserves convexity.
- Q-space gives a surprisingly good approximation. Namely, [Luedtke et al., 2012] show that, if Q has a 0 diagonal, for the box QP: $\min\{x^T Qx : 0 \leq x \leq 1\}$:
  - if $Q \geq 0$ the approximation is within a factor 2:
  - if $Q \not\geq 0$ the approximation is within a factor of # nnz in Q (conjecture it is better)
- Many ways to do better splittings of Q, for example, with SDP [Billionnet et al., 2012].

CPLEX current strategy

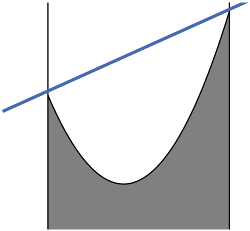
- Uses EV-space if problem looks almost convex.

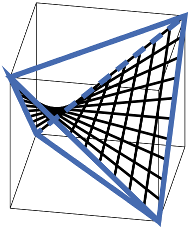
- Let $(\overline{x}, \overline{y})$ be the solution of the chosen QP relaxation after presolve/cutting. And assume $x_j \in \mathbb{Z}$, $j = 1, \ldots, p$.
- If $\exists \overline{y}_{ij} \neq \overline{x}_i \overline{x}_j$, $(\overline{x}, \overline{y})$ is not a solution of the problem and we need to branch.
- Pick an index i, choose a value θ between $\frac{l_i + u_i}{2}$ and $\overline{x}_i$.
- Branch by changing the bound to θ and updating all Secant and McCormick approximations involving this bound.



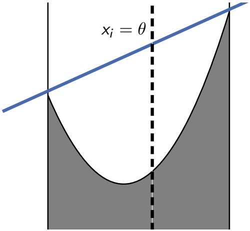


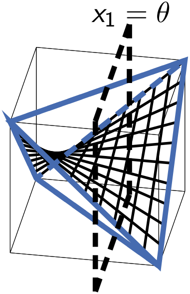
- Let $(\overline{x}, \overline{y})$ be the solution of the chosen QP relaxation after presolve/cutting. And assume $x_j \in \mathbb{Z}$, $j = 1, \dots, p$.
- If $\exists \overline{y}_{ij} \neq \overline{x}_i \overline{x}_j$, $(\overline{x}, \overline{y})$ is not a solution of the problem and we need to branch.
- Pick an index i, choose a value θ between $\frac{l_i + u_i}{2}$ and $\overline{x}_i$.
- Branch by changing the bound to θ and updating all Secant and McCormick approximations involving this bound.



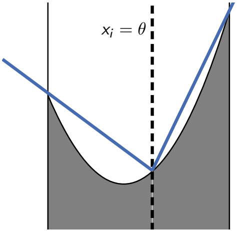


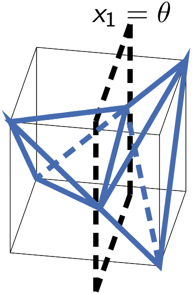
- Let $(\overline{x}, \overline{y})$ be the solution of the chosen QP relaxation after presolve/cutting. And assume $x_j \in \mathbb{Z}$, $j = 1, \ldots, p$.
- If $\exists \overline{y}_{ij} \neq \overline{x}_i \overline{x}_j$, $(\overline{x}, \overline{y})$ is not a solution of the problem and we need to branch.
- Pick an index i, choose a value θ between $\frac{l_i + u_i}{2}$ and $\overline{x}_i$.
- Branch by changing the bound to θ and updating all Secant and McCormick approximations involving this bound.





- Let $(\overline{x}, \overline{y})$ be the solution of the chosen QP relaxation after presolve/cutting. And assume $x_j \in \mathbb{Z}$, $j = 1, \ldots, p$.
- If $\exists \overline{y}_{ij} \neq \overline{x}_i \overline{x}_j$, $(\overline{x}, \overline{y})$ is not a solution of the problem and we need to branch.
- Pick an index i, choose a value θ between $\frac{l_i + u_i}{2}$ and $\overline{x}_i$.
- Branch by changing the bound to θ and updating all Secant and McCormick approximations involving this bound.





Other ingredients

- Convex QP relaxation solved by a QP simplex.
- Interior point solver for improving incumbents.
- Bound strengthening based on the KKT system.
- Linearize completely parts of the problem involving binary variables.
- Heuristic detection of unbounded problems.
- Multi-threaded.

- Try to bound all auxiliary variables with a basic presolve.
- If not possible, do it by solving LPs.
- If there is an unbounded direction r look at its cost $r^T Qr$:
  - If $r^T Qr < 0$: problem is unbounded,
  - If $r^T Qr \geq 0$: relaxation is unbounded but cannot conclude on problem status, return `RELAXATION_UNBOUNDED`.
- (Very easy to construct examples where can't conclude).

[Hu et al., 2012]

- Propose a KKT system that detects unbounded problems correctly.
- Use a combinatorial Benders approach to solve it.

Test set

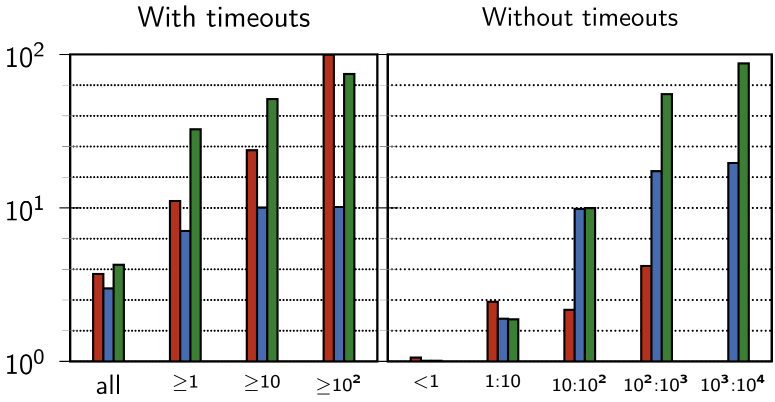
390 models

- Internal nonconvex MIQP (with three variants: original, 50% integer relaxed, 100 % relaxed).
- GAMS Globallib
- minlp.org, Box-QP, Tardella instances, ...
- CUTEr problems with flipped objective

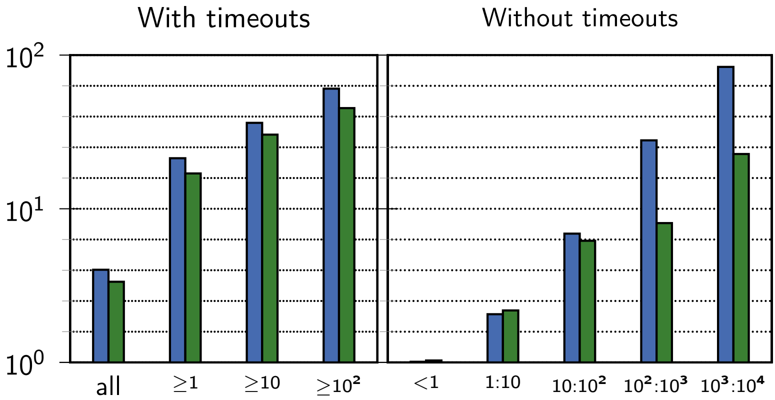
Experiments

- Not really any other solver aimed specifically at nonconvex MIQP.
- Compare with SCIP 3.0.1 [Vigerske, 2013] and Couenne 0.4 [Belotti et al., 2009] using 1 thread.
- Compare CPLEX with 1 and 4 threads.
- Time limit of 3 hours.

Comparison with SCIP on different test-sets



- Pure 0-1 models. Timeouts: SCIP 5.
- Mixed 0-1 models. Timeouts: CPLEX 2 , SCIP 2.
- Continuous and general integers. Timeouts: CPLEX 1, SCIP 29.

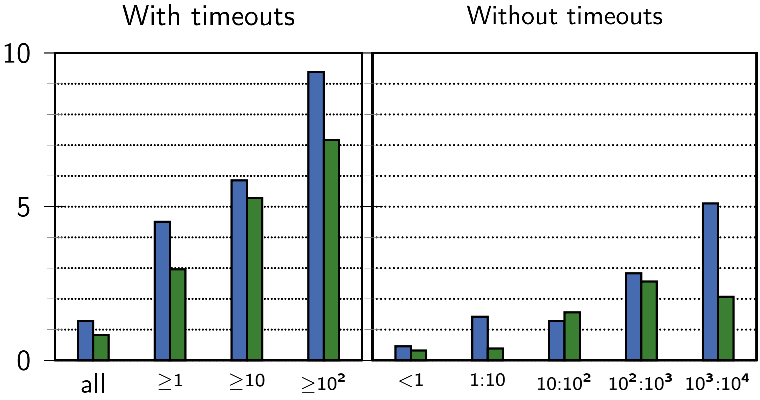


SCIP. 36 timeouts, 5 failures.

Couenne. 22 timeouts, 47 failures

CPLEX 3 timeouts and 7 failures.

CPLEX vs SCIP vs Couenne: Nodes

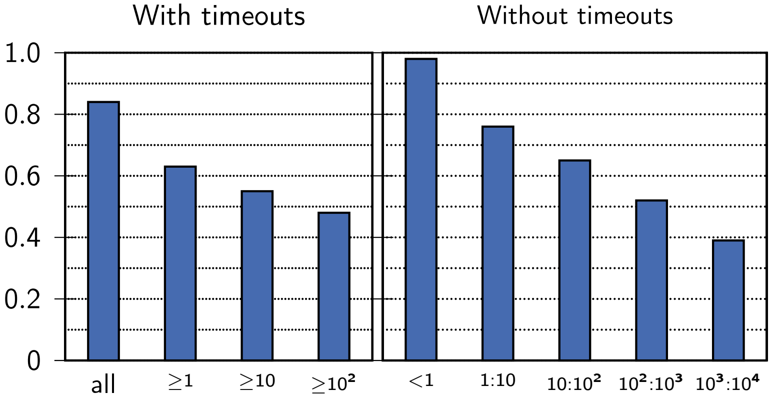


SCIP. 36 timeouts, 5 failures.

Couenne. 22 timeouts, 47 failures

CPLEX 3 timeouts and 7 failures.

CPLEX only, 1 versus 4 threads on computing time



4 models not solved with 1 threads solved with 4.

Agenda

- The convex case
  - Main algorithmic approaches.
  - Glimpse at computation.
- A step into nonconvexity
  - MIQP with CPLEX 12.6.
  - Basic setup of a spatial branch-and-bound.
  - Computational results.
- Arbitrary selection of more advanced topics
  - Separability.
  - Disjunctive Cuts.
- Conclusions
- Problem Session
- Bibliography

- Preprocessing/Modeling:
  - separability [Hijazi et al., 14]
  - perspective formulations [Frangioni and Gentile, 2006, Günlük and Linderoth, 2008]
  - propagation [Vigerske, 2013]
- Node relaxations/Branching:
  - exploiting QP relaxation in strong-branching [Bonami et al., 2013] item divings [Mahajan et al., 2012]
- Primal Heuristics:
  - Feasibility Pumps [Bonami et al., 2009],
  - Undercover [Berthold and Gleixner, 2013]
- Cuts:
  - disjunctive cuts [Kılınc et al., 2011, Bonami, 2011],

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & g_i(x) \leq 0 & i = 1, \ldots, m \\ & x \in X & \\ & x_j \in \mathbb{Z} & j = 1, \ldots, p \\ & l \leq x \leq u & \end{array} \qquad \text{(sMINLP)}$$

- For $i = 1, \ldots, m$, $g_i : X \to R$ are *convex separable*:

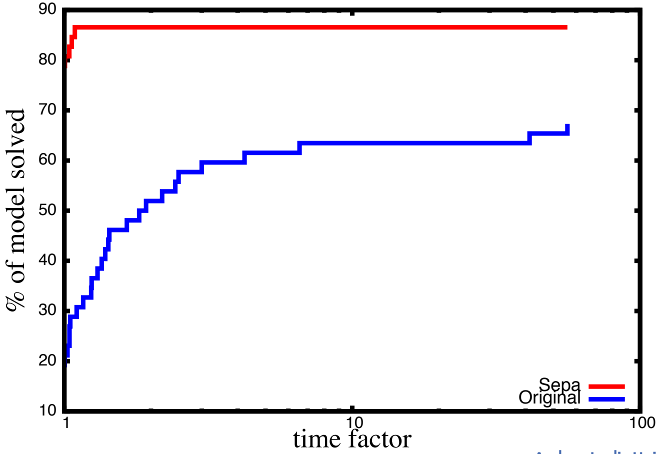
$$g_i(x) = \sum_{j=1}^{n} g_{ij}(x_j)$$

with $g_{ij} : [l_j, u_j] \to \mathbb{R}$ convex.

Introduce one variable y_{ij} for each elementary function:

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & \sum_{j=1}^{n} y_{ij} \le 0 && i = 1, \dots, m, \\
& g_{ij}(x_j) \le y_{ij} && \begin{array}{l} i = 1, \dots, m, \\ j = 1, \dots, n, \end{array} \\
& x \in X, \\
& x_i \in \mathbb{Z} && i = 1, \dots, p, \\
& l \le x \le u.
\end{aligned}
\qquad \text{(sMINLP}^*\text{)}
$$

- In the standard benchmark for MICP, out of 100+ instances, 8 are not directly separable.
- Constructing separated formulations on a subset of 47 instances gives a 3x speed up: [Hijazi et al., 14].



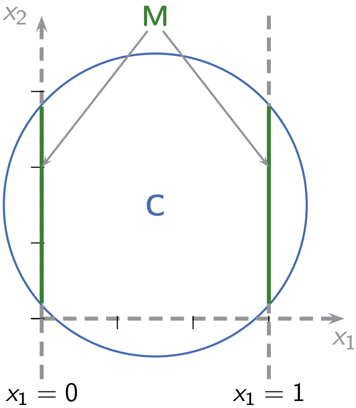
- Cuts are an essential component of MILP solvers.
- Of course one can always apply MILP cuts to a linear OA of MICP.
- How can we generate cuts that also exploit nonlinear constraints?
- Can we generate better cuts by looking directly at nonlinear functions?
- A partial answer: as long as the cut generated is linear it could also have been obtained from a linear outer approximation.
- In the past three years, tremendous activity towards conic cuts for conic programming but no general method yet, and no striking computational results.

- Cuts are an essential component of MILP solvers.
- Of course one can always apply MILP cuts to a linear OA of MICP.
- How can we generate cuts that also exploit nonlinear constraints?
- Can we generate better cuts by looking directly at nonlinear functions?
- A partial answer: as long as the cut generated is linear it could also have been obtained from a linear outer approximation.
- In the past three years, tremendous activity towards conic cuts for conic programming but no general method yet, and no striking computational results.

Consider $\mathbf{C}$ and $\mathbf{M} := \mathbf{C} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})$.
Let $k \leq p$, $\pi \in \mathbb{Z}$ and

$$\mathbf{C}^{(\pi,\pi_0)} := \text{conv}\Big(\mathbf{C} \cap (\{x : x_k \leq \pi\} \cup \{x : x \geq \pi + 1\})\Big).$$

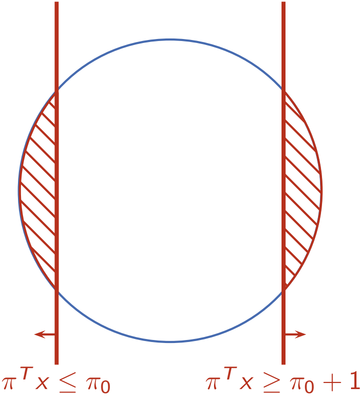
(clearly $\mathbf{M} \subseteq \mathbf{C}^{(\pi,\pi_0)} \subseteq \mathbf{C}$).



Consider $\mathbf{C}$ and $\mathbf{M} := \mathbf{C} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})$.
Let $k \leq p$, $\pi \in \mathbb{Z}$ and

$$\mathbf{C}^{(\pi,\pi_0)} := \text{conv}\Big(\mathbf{C} \cap (\{x : x_k \leq \pi\} \cup \{x : x \geq \pi + 1\})\Big).$$

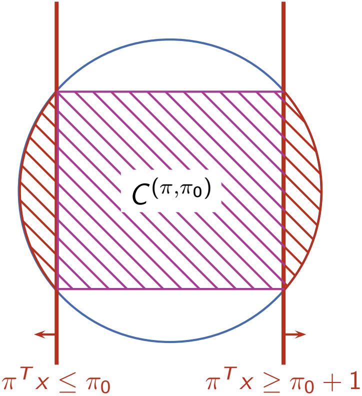
(clearly $\mathbf{M} \subseteq \mathbf{C}^{(\pi,\pi_0)} \subseteq \mathbf{C}$).



Consider $\mathbf{C}$ and $\mathbf{M} := \mathbf{C} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})$.
Let $k \leq p$, $\pi \in \mathbb{Z}$ and

$$\mathbf{C}^{(\pi,\pi_0)} := \text{conv}\Big(\mathbf{C} \cap (\{x : x_k \leq \pi\} \cup \{x : x \geq \pi + 1\})\Big).$$

(clearly $\mathbf{M} \subseteq \mathbf{C}^{(\pi,\pi_0)} \subseteq \mathbf{C}$).

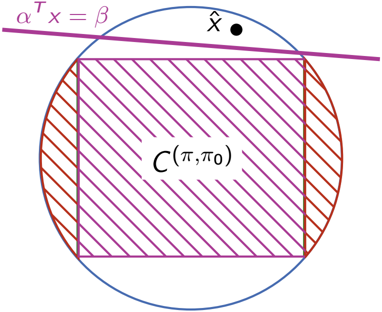


Consider $\mathbf{C}$ and $\mathbf{M} := \mathbf{C} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})$.
Let $k \leq p$, $\pi \in \mathbb{Z}$ and

$$\mathbf{C}^{(\pi,\pi_0)} := \text{conv}\Big(\mathbf{C} \cap (\{x : x_k \leq \pi\} \cup \{x : x \geq \pi + 1\})\Big).$$

(clearly $\mathbf{M} \subseteq \mathbf{C}^{(\pi,\pi_0)} \subseteq \mathbf{C}$).

In the remainder, $\hat{x}$ is the point to separate, $\hat{x}_k \in]0, 1[$ $(k \leq p)$, and $\pi = 0$



Consider C a polyhedron $\{x : Ax = b, x \geq 0\}$

Cut Generation LP

$\hat{x} \in C$ is separated using disjunctive programming:

$$
\begin{aligned}
& \min \alpha^T \hat{x} - \beta \\
& s.t. : \\
& \alpha = u^T A + s - u_0 e_k, \qquad \alpha = v^T A + t + v_0 e_k, \\
& \beta = u^T b, \qquad \beta = v^T b + v_0, \\
& \alpha \in \mathbb{R}^n, \beta \in \mathbb{R}, \; u, v \in \mathbb{R}^m, \; s, \; t \in \mathbb{R}^n_+, \; u_0, v_0 \in \mathbb{R}_+
\end{aligned}
\tag{CGLP}
$$

Goal: build a linear OA from which a "best" cut can be deduced using CGLP.

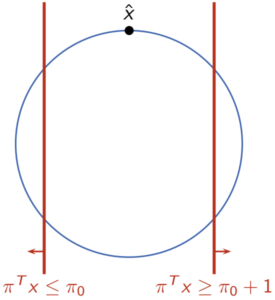
Using only LP [Kılınc et al., 2011].

1. Start with any linear OA of C
2. Solve CGLP. If no cut is found.
3. Deduce from dual of CGLP two points such that $\hat{x} = \lambda x^1 + (1 - \lambda)x^0$ and satisfying the disjunction.
4. If point(s) not in C generate new OA and goto 2, otherwise use the cut.

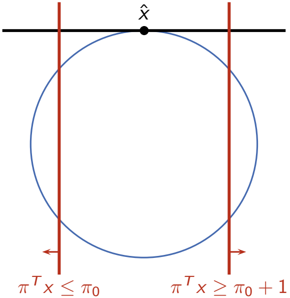
Using NLP [Bonami, 2011]

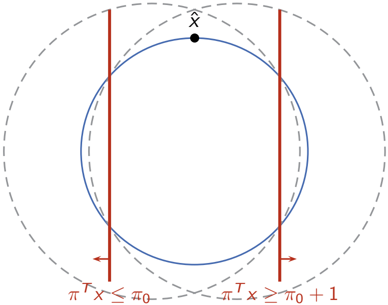
1. Solve a single NLP that tells if $\hat{x}$ is in the split relaxation.
2. If not, deduce from solution two points such that $\hat{x} = \lambda x^1 + (1 - \lambda)x^0$ and closest to satisfy the disjunction.
3. Build OA around these two points.
4. Solve CGLP and get the cut.

Goal: build a linear OA from which a "best" cut can be deduced using CGLP.

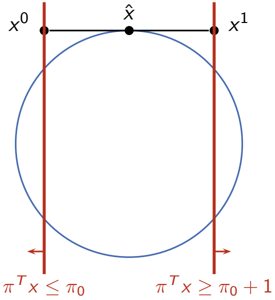


Goal: build a linear OA from which a "best" cut can be deduced using CGLP.

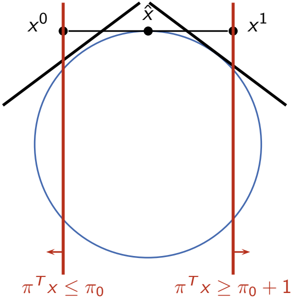


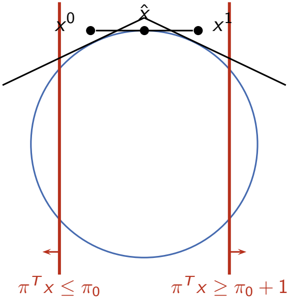


Goal: build a linear OA from which a "best" cut can be deduced using CGLP.

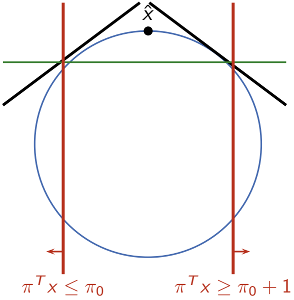


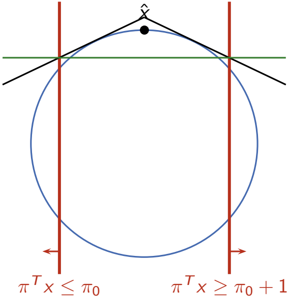
Goal: build a linear OA from which a "best" cut can be deduced using CGLP.



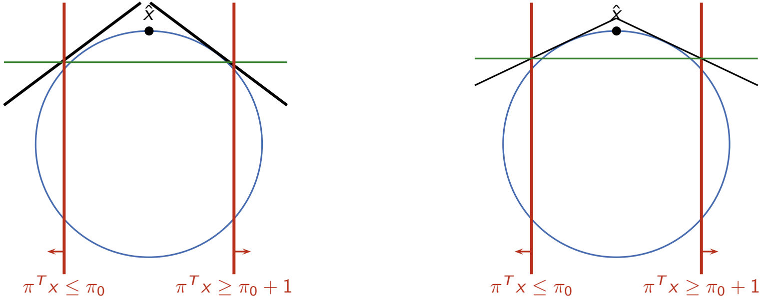


Goal: build a linear OA from which a "best" cut can be deduced using CGLP.





Goal: build a linear OA from which a "best" cut can be deduced using CGLP.



Andrea Lodi, University of Bologna

- [Kılınc et al., 2011] report a speedup of 3 on a set of "hard" instances with the NLP/LP FilMINT.
- [Bonami, 2011] report a speedup of 24 % on nontrivial instances with NLP B&B.
- In both cases, some instances not solved without these cuts are then solved within seconds.

Combination with separability[Kılınç, 2011]

Even better results are obtained by combining the extended formulation trick for separability and these cuts.

		Original		Extended	
	n	gap closed	sol time	gap closed	sol time
Batch	10	58.40	376.2	68.77	58.7
Markowitz	10	0.00	$>$ 10 800	98.07	1 262
SLay	14	68.50	36	86.08	5
uflquad	15	10.85	784	96.25	145

- MINLP is still very challenging and not well solved.
- In the last three years:
  - SCIP [Vigerske, 2013]
  - MINOTAUR [Leyffer et al., 2012]
  - GLOMIQO/ANTIGONE [Misener and Floudas, 2013]

  (each brought tremendous improvement over the state of the art).
- Commercial vendors are also moving.
- Good solvers need good test-sets:
  - www.minlp.org: repository of models.
  - more is needed.

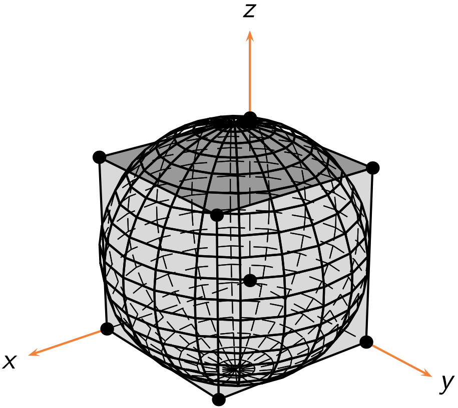
- The convex case
  - Main algorithmic approaches.
  - Glimpse at computation.
- A step into nonconvexity
  - MIQP with CPLEX 12.6.
  - Basic setup of a spatial branch-and-bound.
  - Computational results.
- Arbitrary selection of more advanced topics
  - Separability.
  - Disjunctive Cuts.
- Conclusions
- Problem Session
- Bibliography

Consider the following family of convex MINLPs:

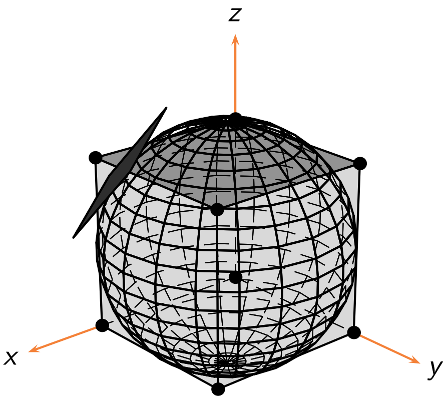
$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & \sum_{i=1}^{n} \left(x_i - \frac{1}{2}\right)^2 \leq \frac{n-1}{4} \\ & x \in \mathbb{Z}^n \end{array} \tag{1}$$

(1) is infeasible:

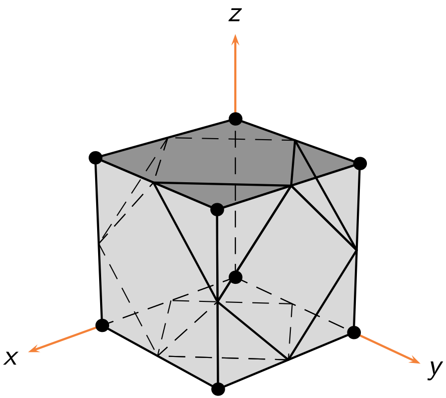
- The ball is too small to contain integer points.
- It is large enough to touch every edge of the hypercube.



- No OA constraint can cut 2 vertices of the hypercube.
  - If an inequality cuts two vertices, it cuts the segment joining them. This cannot be: the ball has non-empty intersection with any such segment.

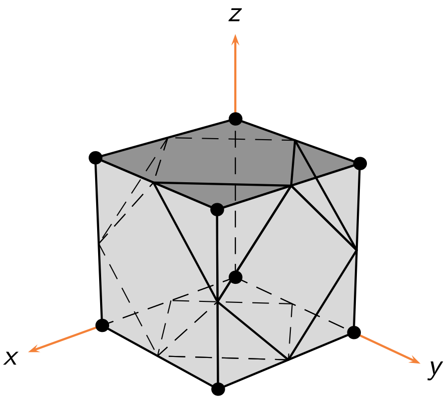


- No OA constraint can cut 2 vertices of the hypercube.
  - If an inequality cuts two vertices, it cuts the segment joining them. This cannot be: the ball has non-empty intersection with any such segment.
- OA decomposition takes at least 2^n iterations (each iteration requires solving a MILP).

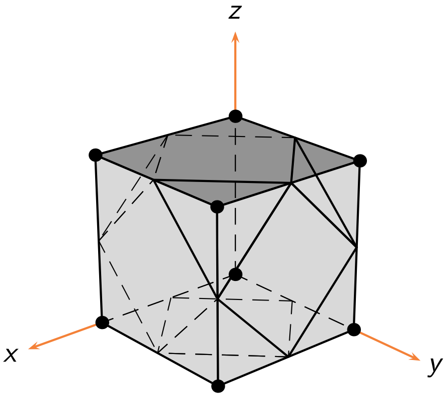


 Andrea Lodi, University of Bologna

- No OA constraint can cut 2 vertices of the hypercube.
  - If an inequality cuts two vertices, it cuts the segment joining them. This cannot be: the ball has non-empty intersection with any such segment.
- OA decomposition takes at least 2^n iterations (each iteration requires solving a MILP).
- An OA Based branch-and-cut would take at least 2^n nodes.



- No OA constraint can cut 2 vertices of the hypercube.
  - If an inequality cuts two vertices, it cuts the segment joining them. This cannot be: the ball has non-empty intersection with any such segment.
- OA decomposition takes at least 2^n iterations (each iteration requires solving a MILP).
- An OA Based branch-and-cut would take at least 2^n nodes.



- Note: NLP branch-and-bound also enumerates at least 2^n integer sols.

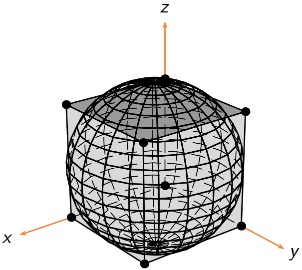
		CPLEX	SCIP 2.1	B-OA	B-Hyb
n	2^n	nodes	nodes	OA it.	nodes
10	1,024	2,047	720	1,105	11,156
15	32,768	65,535	31,993	...	947,014
20	1,048,576	2,097,151	1,216,354	...	...

		CPLEX	SCIP 2.1	B-OA	B-Hyb
n	2^n	nodes	nodes	OA it.	nodes
10	1,024	2,047	720	1,105	11,156
15	32,768	65,535	31,993	...	947,014
20	1,048,576	2,097,151	1,216,354	...	...

Remark

- Problem is simple for CPLEX/SCIP if variables are $0-1$: replace x_i^2 by x_i, the contradiction $\frac{n}{4} \leq \frac{n-1}{4}$ follows.
- SCIP > 3.0 applies tricks and solves it in a blink.

Extended formulation of (1)



$$\begin{aligned} \min\ & c^T x \\ \text{s.t.} & \sum_{i=1}^{n} y_i \le (n-1)/4 \\ & (x_i - 0.5)^2 \le y_i \qquad i = 1, \dots, n \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{2}$$

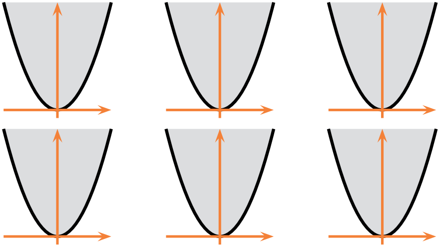
Its outer approximation

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.} & \sum_{i=1}^{n} y_i \le (n-1)/4 \\ & 2\left(\overline{x}_i^k - 0.5\right)(x_i - \overline{x}_i^k) + \left(\overline{x}_i^k - 0.5\right)^2 \le y_i \qquad \begin{matrix} i = 1, \dots, n \\ k = 1, \dots, K \end{matrix} \\ & x \in \mathbb{Z}^n \end{aligned}$$

Application to (1) [Hijazi et al., 14]

Extended formulation of (1)

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.} & \sum_{i=1}^{n} y_i \le (n-1)/4 \\ & (x_i - 0.5)^2 \le y_i \qquad i = 1, \dots, n \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{2}$$



Its outer approximation

$$\begin{aligned} \min\ & c^T x \\ \text{s.t.} & \sum_{i=1}^{n} y_i \le (n-1)/4 \\ & 2\left(\overline{x}_i^k - 0.5\right)(x_i - \overline{x}_i^k) + \left(\overline{x}_i^k - 0.5\right)^2 \le y_i \qquad \begin{array}{c} i = 1, \dots, n \\ k = 1, \dots, K \end{array} \\ & x \in \mathbb{Z}^n \end{aligned}$$

Extended formulation of (1)

min $c^T x$

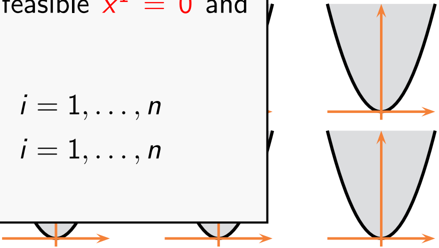
s.t. $\sum_{i=1}^{n}$

$(x_i$ -

$x \in$

2 points suffice to make it infeasible $\overline{x}^1 = 0$ and $\overline{x}^2 = 1$:

$$-x_i + 0.25 \leq y_i \qquad i = 1, \dots, n$$
$$x_i - 0.75 + \leq y_i \qquad i = 1, \dots, n$$



(2)

Its outer approximation

$$\min \; c^T x$$
$$\text{s.t.} \sum_{i=1}^{n} y_i \leq (n-1)/4$$
$$2\left(\overline{x}_i^k - 0.5\right)(x_i - \overline{x}_i^k) + \left(\overline{x}_i^k - 0.5\right)^2 \leq y_i \qquad i = 1, \dots, n, \; k = 1, \dots, K$$
$$x \in \mathbb{Z}^n$$

K. Abhishek, S. Leyffer, and J. T. Linderoth. FilMINT: An outer-approximation-based solver for convex mixed-integer nonlinear programs. *INFORMS Journal on Computing*, 2010. To appear, DOI: 10.1287/ijoc.1090.0373.

W. B. Ameur and A. Ouorou. Mathematical models of the delay constrained routing problem. *Algorithmic Operations Research*, 1(2):94–103, 2006.

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

P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 5 2013. ISSN 1474-0508.

T. Berthold and A. Gleixner. Undercover: a primal minlp heuristic exploring a largest sub-mip. *Mathematical Programming*, pages 1–32, 2013. doi: 10.1007/s10107-013-0635-2.

D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74:121–140, 1996.

A. Billionnet, S. Elloumi, and A. Lambert. Extending the qcr method to general mixed-integer programs. *Mathematical Programming*, 131(1-2):381–401, 2012. doi: 10.1007/s10107-010-0381-7.

P. Bonami. Lift-and-project cuts for mixed integer convex programs. In *IPCO*, pages 52–64, 2011.

P. Bonami and M. Lejeune. An Exact Solution Approach for Integer Constrained Portfolio Optimization Problems Under Stochastic Constraints. *Operations Research*, 57:650–670, 2009.

P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

P. Bonami, G. Cornuejols, A. Lodi, and F. Margot. A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming A*, 119:331–352, 2009.

P. Bonami, J. Lee, S. Leyffer, and A. Wächter. On branching rules for convex mixed-integer nonlinear optimization. *ACM JEA*, 5, 2013.

References II

R. Boorstyn and H. Frank. Large-scale network topological optimization. *IEEE Transactions on Communications*, 25:29–47, 1977.

C. Bragalli, C. D'Ambrosio, J. Lee, A. Lodi, and P. Toth. On the optimal design of water distribution networks: a practical minlp approach. *Optimization and Engineering*, pages 1–28, 2011.

M. R. Bussieck and A. Drud. Sbb: A new solver for mixed integer nonlinear programming. Talk, OR 2001, Section "Continuous Optimization", 2001.

R. H. Byrd, J. Nocedal, and R. A. Waltz. KNITRO: An integrated package for nonlinear optimization. In *Large Scale Nonlinear Optimization, 35–59, 2006*, pages 35–59. Springer Verlag, 2006.

S. Cafieri and N. Durand. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Journal of Global Optimization*, pages 1–17, 2013. ISSN 0925-5001.

I. Castillo, J. Westerlund, S. Emet, and T. Westerlund. Optimization of block layout deisgn problems with unequal areas: A comparison of MILP and MINLP optimization methods. *Computers and Chemical Engineering*, 30:54–69, 2005.

M. A. Duran and I. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

S. Elhedhli. Service System Design with Immobile Servers, Stochastic Demand, and Congestion. *Manufacturing & Service Operations Management*, 8(1):92–97, 2006. doi: 10.1287/msom.1050.0094. URL `http://msom.journal.informs.org/cgi/content/abstract/8/1/92`.

A. Flores-Tlacuahuac and L. T. Biegler. Simultaneous mixed-integer dynamic optimization for integrated design and control. *Computers and Chemical Engineering*, 31:648–656, 2007.

A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0-1 mixed integer programs. *Mathematical Programming*, 106:225–236, 2006.

I. Gentilini, F. Margot, and K. Shimada. The travelling salesman problem with neighborhoods: Minlp solution. *Optimization Methods and Software*, 28(2):364–378, 2013.

O. Günlük and J. Linderoth. Perspective relaxation of mixed integer nonlinear programs with indicator variables. In *IPCO 2008: The Thirteenth Conference on Integer Programming and Combinatorial Optimization*. Springer, 2008.

References III

O. K. Gupta and A. Ravindran. Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31:1533–1546, 1985.

C. A. Haverly. Studies of the behavior of the recursion for the pooling problem. *SIGMAP Bulletin*, 25:19–28, 1978.

H. Hijazi, P. Bonami, and A. Ouorou. An outer-inner approximation for separable mixed-integer nonlinear programs. *INFORMS Journal on Computing*, 26(1):null, 14. doi: 10.1287/ijoc.1120.0545.

J. Hu, J. E. Mitchell, and J.-S. Pang. An lpcc approach to nonconvex quadratic programs. *Math. Program.*, 133(1-2):243–277, 2012.

M. Kılınc, J. Linderoth, and J. Luedtke. Effective separation of disjunctive cuts for convex mixed integer nonlinear programs. Technical Report 1681, 2011.

M. R. Kılınç. *Disjunctive Cutting Planes ann Algorithms for Convex Mixed Integer Nonlinear Programming*. PhD thesis, University of Wisconsin-Madison, 2011.

J. Lee. How we participate in open source agreements, 2008. URL http://ibm.co/1nPMkAM.

S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. Technical report, University of Dundee, 1998.

S. Leyffer, J. Linderoth, J. Luedtke, A. Mahajan, and T. Munson. Minotaur, 2012. URL http://wiki.mcs.anl.gov/minotaur.

J. Luedtke, M. Namazifar, and J. Linderoth. Some results on the strength of relaxations of multilinear functions. *Math. Program.*, 136(2):325–351, 2012.

A. Mahajan, S. Leyffer, and C. Kirches. Solving mixed-integer nonlinear programs by qp-diving. Technical Report ANL/MCS-P2071-0312, Mathematics and Computer Science Division, Argonne National Laboratory, 2012.

G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

R. Misener and C. A. Floudas. GloMIQO: Global Mixed-Integer Quadratic Optimizer. *J. Glob. Optim.*, 57: 3–30, 2013.

References IV

T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of turán. *Canadian Journal of Mathematics*, 17:533–540, 1965.

I. Quesada and I. E. Grossmann. An LP/NLP based branch–and–bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16:937–947, 1992.

A. J. Quist, R. van Gemeert, J. E. Hoogenboom, T. Ílles, C. Roos, and T. Terlaky. Application of nonlinear optimization to reactor core fuel reloading. *Annals of Nuclear Energy*, 26:423–448, 1999.

S. Sager. *Numerical methods for mixed–integer optimal control problems*. Der andere Verlag, Tönning, Lübeck, Marburg, 2005. URL `http://mathopt.de/PUBLICATIONS/Sager2005.pdf`. ISBN 3-89959-416-9.

S. Sager. A benchmark library of mixed-integer optimal control problems. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 631–670. Springer, 2012. URL `http://mathopt.de/PUBLICATIONS/Sager2012b.pdf`.

A. Scozzari and F. Tardella. A clique algorithm for standard quadratic programming. *Discrete Applied Mathematics*, 156(13):2439–2448, 2008.

M. Soler, P. Bonami, A. Olivares, and E. Staffetti. Multiphase mixed-integer optimal control approach to aircraft trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 36(5):1267–1277, 2013.

J. P. Vielma, S. Ahmed, and G. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed integer conic quadratic programs. *INFORMS Journal on Computing*, 20:438–450, 2008.

S. Vigerske. MS Windows NT kernel description, 2012. URL `http://www.coin-or.org/GAMSlinks/benchmarks/MINLP/120913_convex_all/index.html`.

S. Vigerske. *Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming*. PhD thesis, 2013.

T. Westerlund and K. Lundqvist. Alpha-ECP, version 5.101. an interactive minlp-solver based on the extended cutting plane method. In *Updated version of Report 01-178-A, Process Design Laboratory, Abo Akademi Univeristy*, 2005.