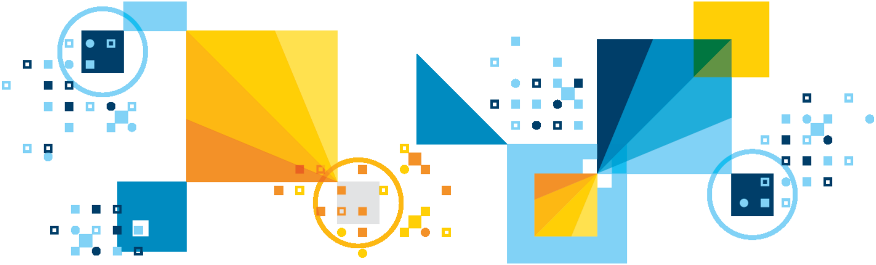
Pierre Bonami

IBM ILOG CPLEX

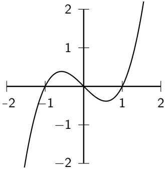
VIII International Workshop on Locational Analysis and Related Problems 2017, Segovia

Some Recent Advances in Mixed Integer Nonlinear Optimization

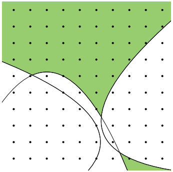


Mixed Integer Nonlinear Optimization

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



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

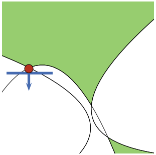


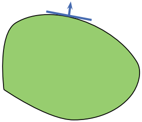
"Well solved" subproblems

Nonlinear Programming (NLP)

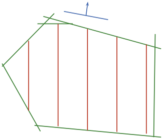
$p = 0$: local optima.

$+ f$ and g_i convex $\Rightarrow$ global optima.





Mixed-Integer linear programming (MILP)



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

The complexity issue

Theorem ([Jeroslow, 1973])

The problem of minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function.

Theorem ([De Loera et al., 2006])

The problem of minimizing a linear function over polynomial constraints in at most 10 integer variables is not computable by a recursive function.

The complexity issue

There is no algorithm to solve (MINO) ...

Theo

The p minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function.

Theorem ([De Loera et al., 2006])

The problem of minimizing a linear function over polynomial constraints in at most 10 integer variables is not computable by a recursive function.

The complexity issue

There is no algorithm to solve (MINO) ...

Theo

The p minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function.

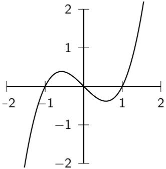
Theorem ([De Loera

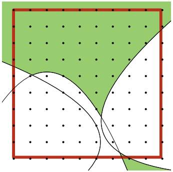
... even in small dimension.

The p over polynomial constraints in at most not computable by a recursive function.

MINO

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





- **To be solvable in general, l_j, u_j finite**.

Two main classes of MINO

Mixed Integer Convex Optimization

Assume that the continuous relaxation is a convex optimization problem.

- f is a convex function.
- g_i are convex functions.

Mixed Integer Nonlinear Optimization

Don't 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, an integer variable x_j can be seen as a continuous satisfying:

$$(x_j - l_j)(x_j - l_j - 1) \ldots .(x_j - u_j) = 0$$

A special class of convex MINLP: MISOCP

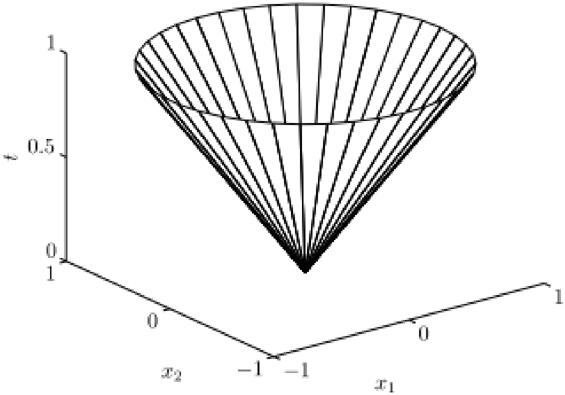
$$
\begin{array}{lll}
\min & c^T x & \\
 & x^T Q_k x + a_k^T x \leq a_k^0 & k = 1, \ldots, m, \\
 & Ax = b, & \\
 & x_j \in \mathbb{Z} & j = 1, \ldots, p.
\end{array}
\qquad \text{(MIQCP)}
$$

Where all quadratic constraints can be represented as second order cones (or Lorentz cone):

$$L^d := \{(x, x_0) \in \mathbb{R}^{d+1} : \sum_{i=1}^{d} x_i^2 \leq x_0^2, x_0 \geq 0\}.$$

(L^d defines the $(d+1)$-dimensional second order cone.)

A Lorentz cone



It is convex!

Second order cone representability

Through simple algebra can be represented as second order cones:

- Second order cones: $\sum_{i=1}^{d} x_i^2 \leq x_0^2,$ with $x_0 \geq 0$
- Rotated second order cones: $\sum_{i=2}^{d} x_i^2 \leq x_0 x_1,$ with $x_0, x_1 \geq 0$
- Simple convex quadratic constraints:

$$x^T Qx + a^T x \leq a^0, \text{ with } Q \succeq 0$$

- or more complicated...

$$||x^T Qx + a^T x|| \leq c^T x + b, \text{ with } Q \succeq 0$$

(the first three should be recognized by most solvers, the last one not.)

Many non-linear constraints can be formulated as second order cones but modeling may be very far from obvious.

MISOCP

$$\begin{array}{lll} \min & c^T x & \\ & (x_{J_i}, x_{h_i}) \in L^{d_i} & i = 1, \ldots, m \\ & Ax = b, & \\ & x_j \in \mathbb{Z} & j = 1, \ldots, p. \end{array} \qquad \text{(MISOCP)}$$

MINLP's where all nonlinear constraints are SOC

- Continuous relaxation solved efficiently by interior points.
- convex MINLP algorithms work with some added technicality due to non-differentiability [Drewes, 2009, Drewes and Ulbrich, 2012].
- Supported by most MIP solvers (all the ones you saw these 2 weeks).

MISOCP Applications

Application	SOC	Integer
Portfolio optimization	Risk, utility, robustness	number of assets, min investment
[Bienstock, 1996, Bonami and Lejeune, 2009, Vielma et al., 2008]		
Truss topology optimization	Physical forces	Cross section of bars
[Achtziger and Stolpe, 2006]		
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]		
Many more... see for eg. http://cblib.zib.de.		

Mixed Integer Convex Programming Applications (not MISOCP)

Application	nonlinear	discrete
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]		

Mixed Integer Nonlinear Optimization Applications

Application	nonlinear	discrete
Petrochemical [Haverly, 1978]	Blending, pooling	–
Gaz/Water networks [Koch et al., 2015, Bragalli et al., 2011]	you know from	last week
Nuclear Reactor reloading [Quist et al., 1999]	reactions	What to reload
Airplane trajectories [Cafieri and Durand, 2013, Soler et al., 2013]	aerodynamics	waypoints, colisions
Mixed Integer Optimal control [Sager, 2005, 2012]	DE	discrete controls
Countless more see for example [Belotti et al., 2013]	. . .	. . .

Agenda

- Non-convex MIQP
  - Basic Setup of a Spatial Branch-and-Bound.
  - Cuts from the Boolean Quadric Polytope
- Classification of Convex MIQP with Machine Learning

(MI)QP

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

(with Q symmetric),

(MI)QP

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

(with Q symmetric),

History of MIQP with CPLEX

class	p	Q	algorithm	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)

Example

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

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

$$s.t.$$

$$\sum x_j = 1$$

$$x \geq 0.$$

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].

Local solver of nonconvex QP

- 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 `optimality target` to 2 (or `CPX_OPTIMALITYTARGET_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 `optimality target` to 3 (or `CPX_OPTIMALITYTARGET_GLOBAL`).
- Note: previous versions could already solve some nonconvex MIQPs (pure 0-1 QPs, convex after presolve...)

Notes on complexity

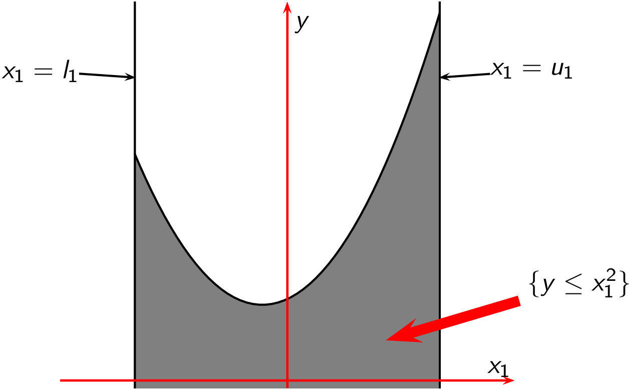
- Checking if a feasible solution is not a local minimum is coNP-Complete.
- Checking if a nonconvex QP is unbounded is NP-complete.

Spatial B&B

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

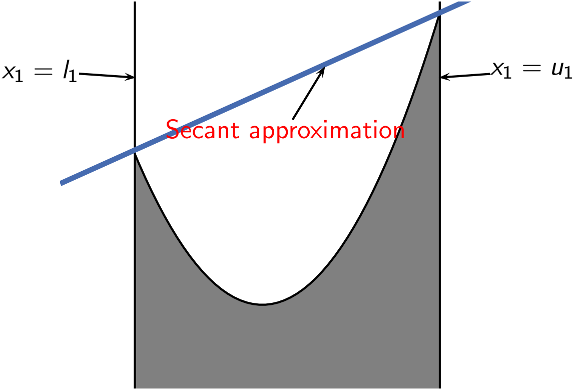
Elementary relaxations: Secant Approximation

The convex hull relaxations of a a square x_1^2



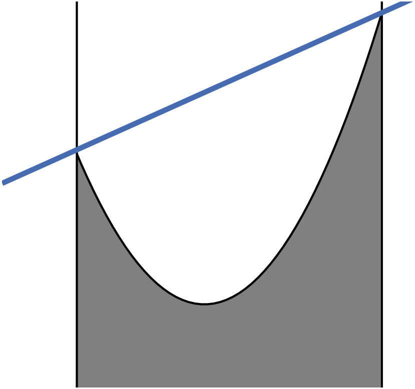
Elementary relaxations: Secant Approximation

The convex hull relaxations of a a square x_1^2



Elementary relaxations: Secant Approximation

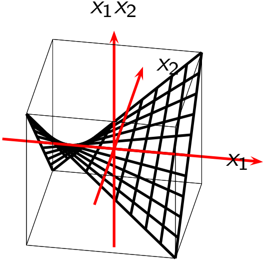
The convex hull relaxations of a a square x_1^2



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

Elementary relaxations: McCormick formulas

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

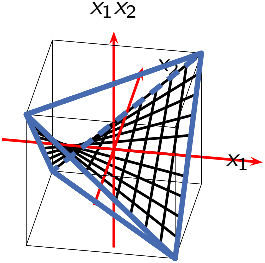


Elementary relaxations: McCormick formulas

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

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

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

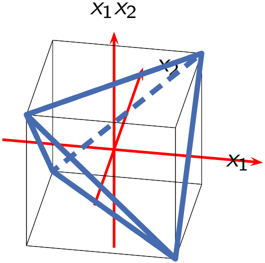


Elementary relaxations: McCormick formulas

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

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

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



- Depending on the sign of q_{ij} we only need y^{+} or y^{-}.

Q-space reformulation and relaxation

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

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

Q-space reformulation and relaxation

- 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 $\tilde{Q}$.

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

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

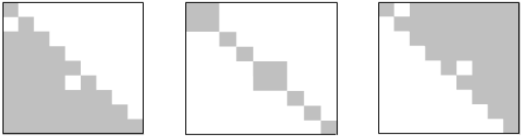
Q-space reformulation and relaxation

- 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 McCormick 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

- Our 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)

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)....

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

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.

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

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^-.

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

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^-.
- 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.

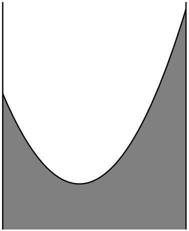
- If Q is diagonal the two relaxations are identical.
- In general they are not comparable.
- If $Q \succeq 0$, EV-space is better it preserves convexity.
- Q-space gives a surpisingly good approximation [Luedtke et al., 2012] show that, if Q has a 0 diagonal, for the box QP: $\min\{x^T Qx : 0 \leq x \leq 1\}$:
  - 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 different splittings of Q for eg. with SDP [Billionnet et al., 2012].

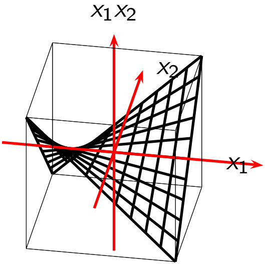
CPLEX current strategy

- By default, uses EV-space if problem looks almost convex.
- Can be controled with parameter.

Branching

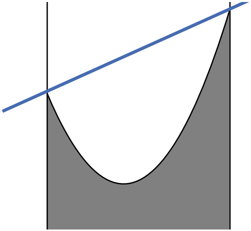
- Let $(\overline{x}, \overline{y})$ be the solution of the chosen QP relaxation after presolve/cutting. And assume $x_j \in \mathbb{Z}$, $j = 1, \ldots, p$.
- If $\exists \overline{y}_{ij} \neq \overline{x}_i \overline{x}_j$, $(\overline{x}, \overline{y})$ is not a solution of the problem and we need to branch.
- Pick such 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.

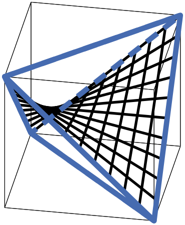




Branching

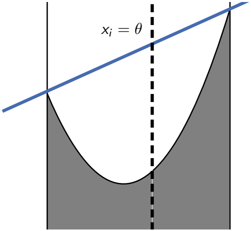
- Let $(\overline{x}, \overline{y})$ be the solution of the chosen QP relaxation after presolve/cutting. And assume $x_j \in \mathbb{Z}$, $j = 1, \dots, p$.
- If $\exists \overline{y}_{ij} \neq \overline{x}_i \overline{x}_j$, $(\overline{x}, \overline{y})$ is not a solution of the problem and we need to branch.
- Pick such 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.

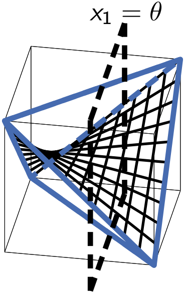




Branching

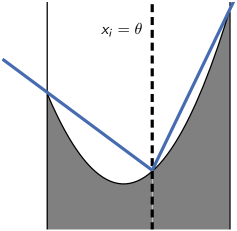
- Let $(\overline{x}, \overline{y})$ be the solution of the chosen QP relaxation after presolve/cutting. And assume $x_j \in \mathbb{Z}$, $j = 1, \ldots, p$.
- If $\exists \overline{y}_{ij} \neq \overline{x}_i \overline{x}_j$, $(\overline{x}, \overline{y})$ is not a solution of the problem and we need to branch.
- Pick such 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.

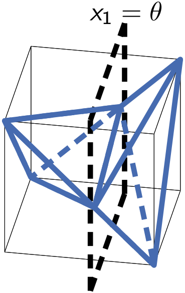




Branching

- 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 such 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.





Notes on unbounded problems

- 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 can't conclude on problem status, return `RELAXATION_UNBOUNDED`.
- (Very easy to construct examples where can't conclude).

[Hu et al., 2012]

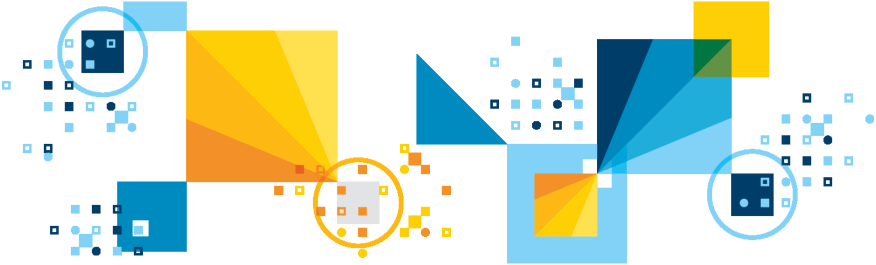
- Propose a KKT system that detects unbounded problems correctly.
- Use a combinatorial benders approach to solve it.

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.

Joint work with
Oktay Günlük - IBM Research
Jeff Linderoth - University of Wisconsin-Madison

Solving Box-Constrained Nonconvex Quadratic Programs via Integer Programming Methods



Box QP

We consider the box constrained QP:

$$\begin{aligned} \min\ & \frac{1}{2}x^T Qx + c^T x \\ s.t.\ & \\ & 0 \leq x \leq 1 \end{aligned} \qquad \text{(box-QP)}$$

- Bounds 0 and 1 are without loss of generality (every box QP can be scaled to those bounds).
- Still NP-Hard.
- Has some academic interest [Vandenbussche and Nemhauser, 2005, Burer and Vandenbussche, 2009, Chen and Burer, 2012]
- Also some applications [Moré and Toraldo, 1989] (usually huge size).

Box QP and Boolean Quadratic Optimization

Proposition ([Burer and Letchford, 2009])

Assume that Q is without diagonal term ($Q_{ii} = 0$, $i = 1, \ldots, n$).
Let Y^Q be the set where variables y represent the products in Q:

$$Y^Q = \{(x, Y) : y_{ij} = x_i x_j, \forall i, j \text{ such that } i \neq j \text{ and } q_{ij} \neq 0\}.$$

We have

$$\operatorname{conv}\left((x, Y) \in Y^Q : x \in [0, 1]^n\right) = \operatorname{conv}\left((x, Y) \in Y^Q : x \in \{0, 1\}^n\right).$$

Corollary

This set is the Boolean Quadratic Polytope (BQP) [Padberg, 1989]. Relaxing diagonal terms of Q using $0 \leq Y_{ii} \leq x_i$, we obtain a BQP (binary) relaxation of box-QP.

Box-QP and BQP relaxation/restriction

$$\min \ {}^1\!/_2 \sum_{i,j:q_{ij}\neq 0} q_{ij} y_{ij} + c^T x$$

$s.t.$ (Box-QP)

$$y_{ij} = x_i x_j$$

$$0 \le x \le 1$$

Box-QP and BQP relaxation/restriction

$$\min \ {}^1\!/_2 \sum_{i,j:q_{ij}\neq 0} q_{ij} y_{ij} + c^T x$$

$$s.t.$$

$$y_{ij} = x_i x_j$$

$$0 \leq x \leq 1$$

(Box-QP)

$$\min \ {}^1\!/_2 \Big(\sum_{i\neq j:q_{ij}\neq 0} q_{ij} y_{ij} + \sum_{i:q_{ii}\neq 0} q_{ii} y_{ii} \Big) + c^T x$$

$$s.t.$$

$$\max\left\{ \begin{array}{c} 0 \\ x_i + x_j - 1 \end{array} \right\} \leq y_{ij} \leq \min\left\{ \begin{array}{c} x_i \\ x_j \end{array} \right\}$$

$$0 \leq y_{ii} \leq x_i$$

$$0 \leq x \leq 1$$

$(\mathcal{M})$

Box-QP and BQP relaxation/restriction

$$\min \ {}^1\!/\!_2 \sum_{i,j:q_{ij}\neq 0} q_{ij}y_{ij} + c^T x$$

$s.t.$ (Box-QP)

$y_{ij} = x_i x_j$

$0 \leq x \leq 1$

$$\min \ {}^1\!/\!_2 \Big(\sum_{i\neq j:q_{ij}\neq 0} q_{ij}y_{ij} + \sum_{i:q_{ii}\neq 0} q_{ii}y_{ii} \Big) + c^T x$$

$s.t.$

$\max\left\{ \begin{matrix} 0 \\ x_i + x_j - 1 \end{matrix} \right\} \leq y_{ij} \leq \min\left\{ \begin{matrix} x_i \\ x_j \end{matrix} \right\}$

$0 \leq y_{ii} \leq x_i$

$0 \leq x \leq 1$

$(\mathcal{M})$

$$\min \ {}^1\!/\!_2 \Big(\sum_{i\neq j:q_{ij}\neq 0} q_{ij}y_{ij} + \sum_{i:q_{ii}<0} q_{ii}x_i \Big) + c^T x$$

$s.t.$

$y_{ij}^- \leq y_{ij} \leq y_{ij}^+$

$0 \leq y_{ii} \leq x_i$

$x \in \{0,1\}^n$

(BQP)

Box-QP and BQP relaxation/restriction

$$\min\ {}^1\!/_2 \sum_{i,j:q_{ij}\neq 0} q_{ij} y_{ij} + c^T x$$

$s.t.$ (Box-QP)

$$y_{ij} = x_i x_j$$

$$0 \leq x \leq 1$$

$$\min\ {}^1\!/_2 \Big(\sum_{i\neq j:q_{ij}\neq 0} q_{ij} y_{ij} + \sum_{i:q_{ii}\neq 0} q_{ii} y_{ii} \Big) + c^T x$$

$s.t.$

$$\max\left\{ \begin{array}{c} 0 \\ x_i + x_j - 1 \end{array} \right\} \leq y_{ij} \leq \min\left\{ \begin{array}{c} x_i \\ x_j \end{array} \right\}$$

$$0 \leq y_{ii} \leq x_i$$

$$0 \leq x \leq 1$$

$(\mathcal{M})$

$$\min\ {}^1\!/_2 \Big(\sum_{i\neq j:q_{ij}\neq 0} q_{ij} y_{ij} + \sum_{i:q_{ii}<0} q_{ii} x_i \Big) + c^T x$$

$s.t.$

$$y_{ij}^- \leq y_{ij} \leq y_{ij}^+$$

$$0 \leq y_{ii} \leq x_i$$

$$x \in \{0,1\}^n$$

(BQP)

$$\min\ \sum_{i\neq j:q_{ij}\neq 0} q_{ij} y_{ij} + \sum_{i:q_{ii}\neq 0} q_{ii} y_{ii} + c^T x$$

$s.t.$

$$y_{ij}^- \leq y_{ij} \leq y_{ij}^+$$

$$0 \leq y_{ii} \leq x_i$$

$$x \in \{0,1\}^n$$

(BQP-R)

Boolean Quadric Relaxation Bounds [1]

Size	Density	#	MC Gap	BQP Root	BQP	BQP-Restrict.
Small	Low	6	35.49	90.34	90.48	100.00
	Medium	9	59.93	90.12	90.24	100.08
	High	27	78.96	89.30	89.69	100.03
Medium	Low	12	47.37	94.88	94.88	100.03
	Medium	6	108.81	93.80	95.66	100.02
	High	3	163.47	91.55	96.74	100.02
Large	Low	6	68.65	95.60	96.92	100.06
	Medium	6	124.88	94.26	97.32	100.00
	High	6	180.85	89.10	96.22	100.00
Jumbo	Low	6	93.91	94.30	97.87	100.01
	Medium	6	170.78	89.89	94.36	100.07
	High	6	232.44	84.89	88.71	100.39

[1] Experiments on test set of [Vandenbussche and Nemhauser, 2005, Burer and Vandenbussche, 2009]

Chvátal Gomory cuts

Consider the feasible set of solutions to a generic integer program $P^I = P^{LP} \cap \mathbb{Z}^n$ where

$$P^{LP} = \{x \in \mathbb{R}^n \mid Ax \geq b\}$$

For any $\alpha \in \mathbb{R}^m_+$, $\alpha^T Ax \geq \alpha^T b$ is satisfied by all feasible solutions of P^{LP}. Furthermore, if $\alpha^T A \in \mathbb{Z}^n$

$$\alpha^T Ax \geq \lceil \alpha^T b \rceil \tag{1}$$

is satisfied by all feasible solutions of P^I.
This inequality is called a *Chvátal-Gomory cut* [Gomory, 1958, Chvátal, 1973].
In the special case when $\alpha \in \{0, 1/2\}^m$, Inequality (1) is called a 0-1/2 *cut* [Caprara and Fischetti, 1996].

CG cuts for Boolean Quadric Polytope

Theorem

All non-dominated Chvátal-Gomory cuts for the bqp *are* 0-$1/2$ *cuts.*

Proof idea

Every non-dominated 0-$1/2$ cut has a combinatorial form and is a odd-hole inequality [Padberg, 1989].

Use a result of Padberg on bases of BQP, that shows that multipliers are 0-$1/2$.

Related results for cut polytope [Barahona, 1993] and when Q is fully dense [Boros et al., 1992].

CG cuts for Boolean Quadric Polytope

Theorem

All non-dominated Chvátal-Gomory cuts for the bqp *are* 0-$^1/_2$ *cuts.*

Proof idea

Every non-dominated 0-$^1/_2$ cut has a combinatorial form and is a odd-hole inequality [Padberg, 1989].

Use a result of Padberg on bases of BQP, that shows that multipliers are 0-$^1/_2$.

Related results for cut polytope [Barahona, 1993] and when Q is fully dense [Boros et al., 1992].

Computational consequences

Separating CG or even 0-$^1/_2$cuts NP-hard in general [Caprara and Fischetti, 1996, Eisenbrand, 1999]

Instead, odd cycle inequalities can be separated in polynomial time [Barahona and Mahjoub, 1986, Barahona et al., 1989].

But MILP solvers have fast heuristics for finding 0-$^1/_2$cuts [Koster et al., 2009]...

Comparison of Bounds by cuts

Size	Density	#	MC Gap	Cplex 0-$^1/_2$	All 0-$^1/_2$	BQP Relax
Small	Low	6	35.49	90.34	90.48	90.48
	Medium	9	59.93	90.12	90.24	90.24
	High	27	78.96	89.30	89.45	89.69
Medium	Low	12	47.37	94.88	94.88	94.88
	Medium	6	108.81	93.80	94.52	95.66
	High	3	163.47	91.55	92.00	96.74
Large	Low	6	68.65	95.60	96.71	96.92
	Medium	6	124.88	94.26	95.64	97.32
	High	6	180.85	89.10	89.47	96.22
Jumbo	Low	6	93.91	94.30	95.84	97.87
	Medium	6	170.78	89.89	90.53	94.36
	High	6	232.44	84.89	84.95	88.71

Strengthened Convex Relaxation

In the BQP relaxation, we relaxed completely the diagonal of Q using $0 \leq Y_{ii} \leq x_i$

Instead, we can relax using $x_i^2 \leq Y_{ii} \leq x_i$ and keep some quadratic terms of Q

This leads to a convex MIQP relaxation, with a diagonal quadratic objective

We denote this strengthened relaxation $\mathcal{M}^2$

Strength of Convex Relaxation $\mathcal{M}^2$

			% Gap Closed			
Size	Density	#	$\mathcal{M}^2$	$\mathcal{M}$ + 0-$1/2$	$\mathcal{M}^2$ + 0-$1/2$	$\Delta(\mathcal{M}^2)$
Small	Low	6	4.68	90.34	99.29	8.95
	Medium	9	3.67	90.12	98.58	8.46
	High	27	3.55	89.30	98.64	9.34
Medium	Low	12	2.39	94.88	99.69	4.82
	Medium	6	1.72	93.80	96.83	3.03
	High	3	1.23	91.55	93.04	1.49
Large	Low	6	1.08	95.60	97.81	2.21
	Medium	6	1.11	94.26	95.99	1.73
	High	6	0.97	89.10	90.17	1.07
Jumbo	Low	6	0.96	94.30	95.80	1.50
	Medium	6	0.84	89.89	90.82	0.93
	High	6	0.66	84.89	85.64	0.75

Other implementation details

Using Implicit Integrality

A folklore property tells that if $q_{ii} < 0$ in a Box-QP, the corresponding variable x_i takes value 0 or 1 in an optimal solution.

Cuts at Branch and Bound Nodes

So far we always assumed bounds $0 \leq x \leq 1$, all results can be adapted to arbitrary finite bounds using shifting and scaling.
This can be used to generate locally valid cuts at nodes of the branch-and-bound tree (or strengthen existing one).

Comparison of CPLEX With and Without Cuts

Table: On BoxQP

		Without BQP cuts			With BQP cuts		Ratios	
			Av.	Av.	Av.	Av.	Ratio	Ratio
category	#	# t.o.	time	nodes	time	nodes	time	nodes
all	79	35	255.77	253301	5.38	23	40.24	7598.63
> 1 sec.	65	35	812.47	1062026	8.27	30	87.76	26274.28
> 10 sec.	56	35	1847.49	2079462	11.45	37	148.42	43925.40

Table: On instances that are not box QPs

		Without BQP cuts		With BQP cuts		Ratios	
		Av.	Av.	Av.	Av.	Ratio	Ratio
category	#	time	nodes	time	nodes	time	nodes
all	75	9.90	4008	8.84	2894	1.11	1.38
> 1 sec.	43	48.80	23397	40.15	13895	1.21	1.68
> 10 sec.	29	179.30	53092	134.60	27349	1.33	1.94

Related approaches

Separation of cuts for the box-QP (without BQP) have been developed in Global Optimization

- The McCormick formula gives the exact hull for 2 variable sets.
- [Meyer and Floudas, 2005] give closed form formula for 3 variables sets.
- Many approximation results on the McCormick Q-space formulation [Coppersmith et al., 1999, Meyer and Floudas, 2005, Luedtke et al., 2012]
- Exploit closed form formula for set with up to 6 variables [Misener and Floudas, 2013].

SDP Approaches

box-QP also admits a natural SDP relaxations:

$$z_{\mathcal{S}} = \{\min \langle {}^1\!/_2\ Q, Y\rangle + c^T x \mid Y - xx^T \succeq 0, 1 \geq x_i \geq 0 \ \ \forall i \in \mathbb{N}\}$$

This relaxation can in turn be strengthened using:

McCormick approximations: $x_i + x_j - 1 \leq Y_{ij} \leq \min\{x_i, x_j\}$ [Anstreicher, 2008].

The doubly non-negative relaxation of the copositive reformulation of box-QP [Burer, 2009].

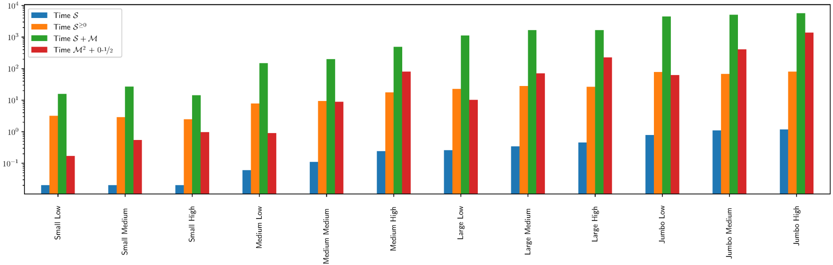
A line of exact approaches based on these relaxation and KKT formulations of QPs [Vandenbussche and Nemhauser, 2005, Burer and Vandenbussche, 2009, Chen and Burer, 2012].

Remark

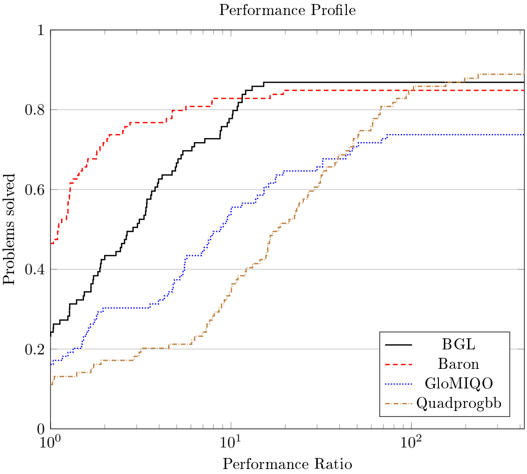
Contrary to what we do SDP relaxation work in the space of all entries of Q and not only non-zeroes.

SDP-based Bounds and BQP-based Bounds for BoxQP

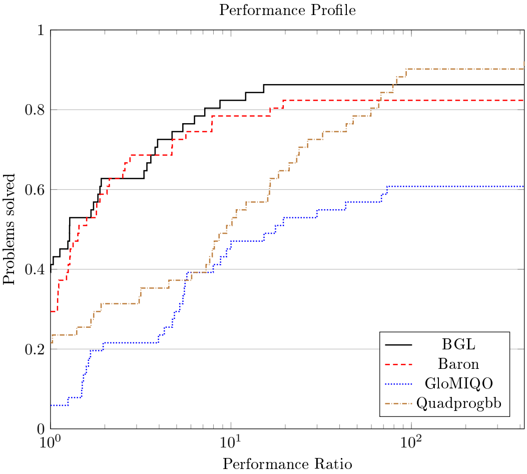
				% Gap closed			
Size	Density	#	$\mathcal{M}^2$ Gap	$\mathcal{S}$	$\mathcal{S}^{\geq 0}$	$\mathcal{S} + \mathcal{M}$	$\mathcal{M}^2$ + 0-1/2
Small	Low	6	35.49	80.65	99.11	99.29	99.51
	Medium	9	59.93	89.79	99.4	99.46	99.29
	High	27	78.97	94.15	99.76	99.8	99.13
Medium	Low	12	47.37	85.85	99.33	99.55	99.90
	Medium	6	108.81	93.0	98.77	98.86	98.01
	High	3	163.47	95.68	99.24	99.31	93.52
Large	Low	6	68.65	88.61	98.2	98.65	98.28
	Medium	6	124.89	94.96	99.05	99.25	97.48
	High	6	180.85	96.34	99.14	99.29	90.60
Jumbo	Low	6	93.91	92.9	98.35	98.84	96.28
	Medium	6	170.78	95.25	98.6	98.82	91.42
	High	6	232.44	96.67	98.96	99.16	85.68

10^4
10^3
10^2
10^1
10^0
10^{-1}
Time $\mathcal{S}$
Time $\mathcal{S}^{\geq 0}$
Time $\mathcal{S} + \mathcal{M}$
Time $\mathcal{M}^2$ + 0-1/2
Small Low
Small Medium
Small High
Medium Low
Medium Medium
Medium High
Large Low
Large Medium
Large High
Jumbo Low
Jumbo Medium
Jumbo High

Comparing Solvers on Box-QP Instances

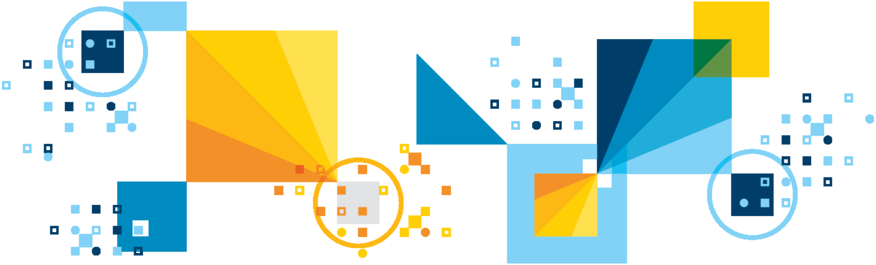


Comparing Solvers on Larger Solved Box-QP Instances



Joint work with
Andrea Lodi and Giulia Zarpellon
École Polytechnique de Montréal, Canada

Learning a Classification of Mixed-Integer Quadratic Programming Problems



What can ML do for (Integer) Optimization?

A fast growing literature has started to appear in the last 5 to 10 years on the use of Machine Learning techniques to help Optimization, especially MIP solvers. Among the first in these series, the papers on tuning MIP solvers.

[Hoos et al. (2010+)]

What can ML do for (Integer) Optimization?

A fast growing literature has started to appear in the last 5 to 10 years on the use of Machine Learning techniques to help Optimization, especially MIP solvers. Among the first in these series, the papers on tuning MIP solvers.

[Hoos et al. (2010+)]

ML has, of course, started to be used within Constraint Programming as well, including Neural Networks and Decision Trees.

[Lombardi & Milano (2015+)]

What can ML do for (Integer) Optimization?

A fast growing literature has started to appear in the last 5 to 10 years on the use of Machine Learning techniques to help Optimization, especially MIP solvers. Among the first in these series, the papers on tuning MIP solvers.

[Hoos et al. (2010+)]

ML has, of course, started to be used within Constraint Programming as well, including Neural Networks and Decision Trees.

[Lombardi & Milano (2015+)]

MIP solvers are complex software objects implementing a large variety of algorithmic approaches. Strategic decisions on how to combine those approaches in the most effective way have to be taken over and over.

What can ML do for (Integer) Optimization?

A fast growing literature has started to appear in the last 5 to 10 years on the use of Machine Learning techniques to help Optimization, especially MIP solvers. Among the first in these series, the papers on tuning MIP solvers.

[Hoos et al. (2010+)]

ML has, of course, started to be used within Constraint Programming as well, including Neural Networks and Decision Trees.

[Lombardi & Milano (2015+)]

MIP solvers are complex software objects implementing a large variety of algorithmic approaches. Strategic decisions on how to combine those approaches in the most effective way have to be taken over and over. Such decisions are taken heuristically, often breaking ties in architecture-dependent ways, thus showing the heuristic nature of MIP implementations.

[Lodi (2012)]

What can ML do for (Integer) Optimization?

A fast growing literature has started to appear in the last 5 to 10 years on the use of Machine Learning techniques to help Optimization, especially MIP solvers. Among the first in these series, the papers on tuning MIP solvers.

[Hoos et al. (2010+)]

ML has, of course, started to be used within Constraint Programming as well, including Neural Networks and Decision Trees.

[Lombardi & Milano (2015+)]

MIP solvers are complex software objects implementing a large variety of algorithmic approaches. Strategic decisions on how to combine those approaches in the most effective way have to be taken over and over. Such decisions are taken heuristically, often breaking ties in architecture-dependent ways, thus showing the heuristic nature of MIP implementations.

[Lodi (2012)]

ML can help systematize the process that leads to take these decisions, especially when a large quantity of data can be collected.

Variable selection in Branch and Bound

Branch-and-Bound algorithm (B&B):

- most widely used procedure for solving (Mixed-)Integer Programming problems
- implicit enumeration search, mapped into a decision tree
- leave (at least) two big choices:
  1. How to split a problem into subproblems (variable selection)
  2. Which node/subproblem to select for the next exploration

Variable selection in Branch and Bound

Branch-and-Bound algorithm (B&B):

- most widely used procedure for solving (Mixed-)Integer Programming problems
- implicit enumeration search, mapped into a decision tree
- leave (at least) two big choices:
  1. How to split a problem into subproblems (variable selection)
  2. Which node/subproblem to select for the next exploration

. . . decisions play a key role for the algorithm efficiency!

- as of today, decisions are made heuristically and empirically evaluated
- there are good branching strategies, but usually very costly

Variable selection in Branch and Bound

Branch-and-Bound algorithm (B&B):

- most widely used procedure for solving (Mixed-)Integer Programming problems
- implicit enumeration search, mapped into a decision tree
- leave (at least) two big choices:
  1. How to split a problem into subproblems (variable selection)
  2. Which node/subproblem to select for the next exploration

. . . decisions play a key role for the algorithm efficiency!

- as of today, decisions are made heuristically and empirically evaluated
- there are good branching strategies, but usually very costly

Ultimate goal

Use ML to learn an activation function that can be adopted as approximation / prediction of a good B&B strategy, ideally with a low computational cost.

[Alvarez, Wehenkel & Louveaux (2016), Khalil, Le Bodic, Song, Nemhauser & Dilkina (2016)]

MIQPs classification

We consider Mixed-Integer Quadratic Programming (MIQP)

$$
\begin{aligned}
\min \quad & \frac{1}{2}x^T Q x + c^T x \\
& Ax = b \\
& x_i \in \{0,1\} \quad \forall\, i \in I \\
& l \leq x \leq u
\end{aligned}
\tag{2}
$$

where $Q = \{q_{ij}\}_{i,j=1\dots n} \in \mathbb{R}^{n \times n}$ is a real symmetric matrix, either convex or nonconvex, and all integer variables are binary.

MIQPs classification

We consider Mixed-Integer Quadratic Programming (MIQP)

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + c^T x \\ & Ax = b \\ & x_i \in \{0,1\} \quad \forall\, i \in I \\ & l \leq x \leq u \end{aligned} \tag{2}$$

where $Q = \{q_{ij}\}_{i,j=1\dots n} \in \mathbb{R}^{n\times n}$ is a real symmetric matrix, either convex or nonconvex, and all integer variables are binary.

Depending on the problems' structure, we can tackle them in different ways:

- $Q \succeq 0$: perform NLP based B&B (or Outer Approximation algorithms)
- $Q \not\succeq 0$: depending on variables' type,
    - pure 0-1: transform into either a convex MIQP or into a MILP (i.e., linearize it)
    - mixed: perform $Q-$space reformulation/relaxation, run Global Optimization algorithms (Spatial B&B)

MIQPs classification (cont.d)

The linearization approach seems beneficial also for the convex case, both for pure 0-1 and mixed problems. However, is linearizing always the best choice?

"[. . .] when one looks at a broader variety of test problems the decision to linearize (vs. not linearize) does not appear so clear-cut.[2]"

[2] Fourer B. Quadratic Optimization Mysteries, Part 2: Two Formulations. `http://bob4er.blogspot.com/2015/03/quadratic-optimization-mysteries-part-2.html`

MIQPs classification (cont.d)

The linearization approach seems beneficial also for the convex case, both for pure 0-1 and mixed problems. However, is linearizing always the best choice?

"[. . .] when one looks at a broader variety of test problems the decision to linearize (vs. not linearize) does not appear so clear-cut.[2]"

Exploit ML predictive machinery to understand whether it is favorable to linearize the quadratic part of the MIQP or not

[2] Fourer B. Quadratic Optimization Mysteries, Part 2: Two Formulations. `http://bob4er.blogspot.com/2015/03/quadratic-optimization-mysteries-part-2.html`

MIQPs classification (cont.d)

The linearization approach seems beneficial also for the convex case, both for pure 0-1 and mixed problems. However, is linearizing always the best choice?

"[. . .] when one looks at a broader variety of test problems the decision to linearize (vs. not linearize) does not appear so clear-cut.[2]*"*

Exploit ML predictive machinery to understand whether it is favorable to linearize the quadratic part of the MIQP or not

- Learn an offline classifier predicting the most suited resolution approach within `IBM-CPLEX` framework (`qtolin` linearization switch parameter)
- Gain theoretical insights about which features of the MIQPs most affect the prediction

[Bonami, Lodi, Zarpellon (2017)]

[2] Fourer B. Quadratic Optimization Mysteries, Part 2: Two Formulations. `http://bob4er.blogspot.com/2015/03/quadratic-optimization-mysteries-part-2.html`

MIQPs classification - Dataset generation

. . . traditional benchmark sets are too small for learning!

MIQPs classification - Dataset generation

. . . traditional benchmark sets are too small for learning!

We define and implement a generator of MIQP instances, spanning a variety of structural parameters and optimization components.

(I) Objective function data generation: real symmetric matrices are generated via the `MATLAB` function

```
Q = sprandsym(size, density, eigenvalues)
```

(II) Variables' type definition: binary/continuous variables are added to the problems with respect to the sign of Q entries, in different proportions

(III) Constraints generation: different constraints sets are added accordingly to the type of variables of the problem (e.g., cardinality, simplex, multi-dimensional knapsack)

MIQPs classification - Dataset generation

. . . traditional benchmark sets are too small for learning!

We define and implement a generator of MIQP instances, spanning a variety of structural parameters and optimization components.

(I) Objective function data generation: real symmetric matrices are generated via the `MATLAB` function

```
Q = sprandsym(size, density, eigenvalues)
```

(II) Variables' type definition: binary/continuous variables are added to the problems with respect to the sign of Q entries, in different proportions

(III) Constraints generation: different constraints sets are added accordingly to the type of variables of the problem (e.g., cardinality, simplex, multi-dimensional knapsack)

- Dataset of 2300 instances, three types of MIQPs (0-1 convex, 0-1 nonconvex, mixed convex)
- Plan to compare with traditional benchmark libraries for test/extensions

MIQPs classification - Features design

We define a set of 23 features referring to an MIQP instance, and we divide them into two main blocks:

MIQPs classification - Features design

We define a set of 23 features referring to an MIQP instance, and we divide them into two main blocks:

- Static features describe the mathematical characteristics of the instance, in terms of
  - variables - e.g., number, types, presence in constraints and objective
  - constraints - e.g., coefficients and variables presence
  - quadratic objective function - e.g., coefficients, variables presence, sparsity, spectral properties

  They are extracted via `CPLEX/Python` before any solving (pre)process takes place.

MIQPs classification - Features design

We define a set of 23 features referring to an MIQP instance, and we divide them into two main blocks:

- Static features describe the mathematical characteristics of the instance, in terms of
  - variables - e.g., number, types, presence in constraints and objective
  - constraints - e.g., coefficients and variables presence
  - quadratic objective function - e.g., coefficients, variables presence, sparsity, spectral properties

  They are extracted via `CPLEX/Python` before any solving (pre)process takes place.

- Dynamic features describe the initial behavior with respect to different resolution methods.
  - e.g., bounds and solution times at the root node

  They are extracted from the early stages of the optimization, after the preprocessing and the resolution of the root node relaxation.

MIQPs classification - Labeling procedure

One of three different labels among $\{\mathsf{L}, \mathsf{NL}, \mathsf{T}\}$ can be assigned to an MIQP instance, describing the winner between *linearize*, *not-linearize* or the case of a *tie* of the two methods.

MIQPs classification - Labeling procedure

One of three different labels among $\{\mathsf{L}, \mathsf{NL}, \mathsf{T}\}$ can be assigned to an MIQP instance, describing the winner between *linearize*, *not-linearize* or the case of a *tie* of the two methods.

Each problem of the dataset is run with timelimit of 1h, for 5 different random seeds, with `qtolin` on and off.

To address solvability / consistency issues, we perform

- Solvability check, to discard never-solved instances
- Seed consistency check on each seed, to discard unstable instances w.r.t. the found upper and lower bounds
- Global consistency check on global best upper and lower bounds, to discard unstable instances

Running time is the ultimate compared measure, assessing the final label for each example.

MIQPs classification - Learning experiments

Instances, features and labels give a dataset ready for supervised learning:

$$\{(x^k, y^k)\}_{k=1..N} \quad \text{where } x^k \in \mathbb{R}^d,\ y \in \{\mathrm{L}, \mathrm{NL}, \mathrm{T}\} \quad \text{for } N \text{ MIQPs}$$

MIQPs classification - Learning experiments

Instances, features and labels give a dataset ready for supervised learning:

$$\{(x^k, y^k)\}_{k=1..N} \quad \text{where } x^k \in \mathbb{R}^d,\ y \in \{\mathsf{L}, \mathsf{NL}, \mathsf{T}\} \quad \text{for } N \text{ MIQPs}$$

Multiclass classifiers such as

- Support Vector Machines (nonlinear RBF kernel) (SVM)

and ensemble methods based on Decision Trees (more interpretable than Neural Networks), such as

- Random Forests (RF)
- Extremely Randomized Trees (EXT)
- Gradient Tree Boosting (GTB)

MIQPs classification - Learning experiments

Instances, features and labels give a dataset ready for supervised learning:

$$\{(x^k, y^k)\}_{k=1..N} \quad \text{where } x^k \in \mathbb{R}^d,\ y \in \{\mathsf{L}, \mathsf{NL}, \mathsf{T}\} \quad \text{for } N \text{ MIQPs}$$

Multiclass classifiers such as

- Support Vector Machines (nonlinear RBF kernel) (SVM)

and ensemble methods based on Decision Trees (more interpretable than Neural Networks), such as

- Random Forests (RF)
- Extremely Randomized Trees (EXT)
- Gradient Tree Boosting (GTB)

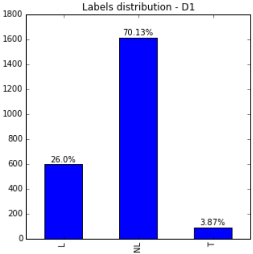
Methodology: follow ML best practices to avoid overfitting

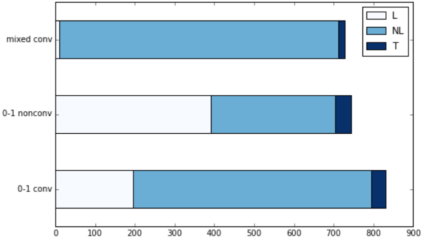
- Training phase to optimize parameters (1725 instances)
- *k*-fold cross validation and grid search for hyper-parameters selection
- Test phase to assess classifiers' performance (575 instances)

Main implementation tool: `scikit-learn` library.

MIQPs classification - Nutshell analysis

Before learning, look into the dataset! In a nutshell:





- Take care of unbalanced data in the learning procedure
- Can some trends already been recognized w.r.t. different problem types?
- More (statistical) analyses on features and data distribution

MIQPs classification - Some results

Classifiers perform well with respect to traditional classification measures

Multiclass - All features				
	SVM	RF	EXT	GTB
Accuracy	0.85	0.89	0.84	0.87
Precision	0.82	0.85	0.81	0.85
Recall	0.85	0.89	0.84	0.87
F1 score	0.83	0.87	0.82	0.86

MIQPs classification - Some results

Classifiers perform well with respect to traditional classification measures

Multiclass - All features				
	SVM	RF	EXT	GTB
Accuracy	0.85	0.89	0.84	0.87
Precision	0.82	0.85	0.81	0.85
Recall	0.85	0.89	0.84	0.87
F1 score	0.83	0.87	0.82	0.86

Top 5 - Features importance scores:

- difference of lower bounds found by L and NL at root node (dynamic ft.)
- difference of root node resolution times (dynamic ft.)
- value of smallest nonzero eigenvalue
- a measure of "diagonal dominance", computed as $\frac{1}{n}\sum_{i=1}^{n}(|q_{ii}| - \sum_{j\neq i}|q_{ij}|)$
- spectral norm of Q, i.e., $\|Q\| = \max_i |\lambda_i|$

MIQPs classification - Learning settings

Simplify the *Multiclass - All features* framework by considering

- Binary setting: remove all tie cases

  How relevant are ties with respect to the question L vs. NL?

  - Classification measures are overall improved, RF is still best performing.

- Static features setting: remove dynamic features

  How does the prediction change without information at root node?

  - Classification is slightly deteriorated, but overall coherent with the original one. The new best performing algorithm is SVM.

MIQPs classification - Learning settings

Simplify the *Multiclass - All features* framework by considering

- Binary setting: remove all tie cases

  How relevant are ties with respect to the question L vs. NL?

  - Classification measures are overall improved, RF is still best performing.

- Static features setting: remove dynamic features

  How does the prediction change without information at root node?

  - Classification is slightly deteriorated, but overall coherent with the original one. The new best performing algorithm is SVM.

- *Binary - Static features* setting, simplified in labels and features
  - Performance is balanced between improvement and deterioration, with SVM as best algorithm.
  - Static features about Q spectrum are the top ones.

What is the best learning setting to integrate predictions and solver?

MIQPs classification - Optimization scores

We need to evaluate classifiers' performance in optimization terms, and quantify the gain with respect to CPLEX default strategy (DEF)

- For each example, select the runtime corresponding to the predicted label (L, NL, T) to build a times vector t_{clf} for each classifier *clf* and DEF
- t_{best} (t_{worst}) contains times corresponding to the correct (wrong) labels

MIQPs classification - Optimization scores

We need to evaluate classifiers' performance in optimization terms, and quantify the gain with respect to CPLEX default strategy (DEF)

- For each example, select the runtime corresponding to the predicted label (L, NL, T) to build a times vector t_{clf} for each classifier *clf* and DEF
- t_{best} (t_{worst}) contains times corresponding to the correct (wrong) labels

σ_{clf} **Sum of predicted runtimes**: sum over times in t_{clf}

$N\sigma_{clf}$ **Normalized time score**: shifted geometric mean of times in t_{clf}, normalized between best and worst cases to get a score $\in [0, 1]$

MIQPs classification - Optimization scores

We need to evaluate classifiers' performance in optimization terms, and quantify the gain with respect to CPLEX default strategy (DEF)

- For each example, select the runtime corresponding to the predicted label (L, NL, T) to build a times vector t_{clf} for each classifier *clf* and DEF
- t_{best} (t_{worst}) contains times corresponding to the correct (wrong) labels

σ_{clf} **Sum of predicted runtimes**: sum over times in t_{clf}

$N\sigma_{clf}$ **Normalized time score**: shifted geometric mean of times in t_{clf}, normalized between best and worst cases to get a score $\in [0, 1]$

	SVM	RF	EXT	GTB	DEF
$\sigma_{clf}/\sigma_{best}$	1.49	1.31	1.43	1.35	5.77
$\sigma_{worst}/\sigma_{clf}$	7.48	8.49	7.81	8.23	1.93
$\sigma_{\text{DEF}}/\sigma_{clf}$	3.88	4.40	4.04	4.26	–
$N\sigma_{clf}$	0.98	0.99	0.98	0.99	0.42

MIQPs classification - CPLEX partial testbed

Preliminary experiments on partial CPLEX internal testbed (175 instances), used as new test set for classifiers trained on the synthetic data.

- Very different distribution of features, problem types and labels: T is the majority class, with very few NL
- All classifiers perform very poorly in terms of classification measures (and most often a T is predicted as NL), but . . .

MIQPs classification - CPLEX partial testbed

Preliminary experiments on partial CPLEX internal testbed (175 instances), used as new test set for classifiers trained on the synthetic data.

- Very different distribution of features, problem types and labels: T is the majority class, with very few NL
- All classifiers perform very poorly in terms of classification measures (and most often a T is predicted as NL), but ...

  ... performance is not bad in optimization terms:

	SVM	RF	EXT	GTB
$\sigma_{clf}/\sigma_{best}$	2.55	2.30	1.72	2.91
$\sigma_{worst}/\sigma_{clf}$	2.00	2.22	2.96	1.75
$N\sigma_{clf}$	0.75	0.90	0.91	0.74

Given the high presence of ties, runtimes for L and NL are most often comparable, so the loss in performance is not dramatic.

MIQPs classification - Going further

Directions for ongoing and future research:

- Analyze other benchmark datasets, e.g., QPLIB, to understand how representative the synthetic data is of commonly used instances, and enlarge the current training set
- Identify the best learning scenario in order to successfully integrate prediction and solver
- Define a custom loss function to train classifiers, to get a prediction tailored on the optimization aspects and the solver's performance as well

IBM®

Legal Disclaimer

- © IBM Corporation 2015. All Rights Reserved.
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- IBM, the IBM logo, CPLEX and SPSS are trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Intel, Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.

References I

W. Achtziger and M. Stolpe. Truss topology optimization with discrete design variables—Guaranteed global optimality and benchmark examples. *Structural and Multidisciplinary Optimization*, 34(1):1–20, 2006.

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

K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2):471–484, 2008.

F. Barahona. On cuts and matchings in planar graphs. *Mathematical Programming*, 60:53,58, 1993.

F. Barahona and A. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

F. Barahona, M. Grötschel, M. Jünger, and G. Reinelt. Experiments in quadratic 0–1 programming. *Mathematical Programming*, 44:127–137, 1989.

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.

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

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

E. Boros, Y. Crama, and P. L. Hammer. Chvátal cuts and odd cycle inequalities in quadratic 0-1 optimization. *SIAM Journal on Discrete Mathematics*, 5(2):163–177, 1992.

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.

S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120:479–495, 2009.

S. Burer and A. N. Letchford. On nonconvex quadratic programming with box constraints. *SIAM Journal on Optimization*, 20(2):1073–1089, 2009. doi: 10.1137/080729529.

References II

S. Burer and D. Vandenbussche. Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound. *Comput Optim Appl*, 43:181–195, 2009.

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.

A. Caprara and M. Fischetti. $\{0, \frac{1}{2}\}$ chvátal-gomory cuts. *Mathematical Programming*, 74:221–235, 1996.

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.

J. Chen and S. Burer. Globally solving nonconvex quadratic programming problems via completely positive programming. *Mathematical Programming Computation*, 4(1):33–52, 2012.

V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial optimization. *Discrete Mathematics*, 4:305–337, 1973.

D. Coppersmith, O. Günlük, J. Lee, and J. Leung. A polyhedron for products of linear functions in 0/1 variables. Technical Report RC21568, IBM Research Division, September 1999. Revised and published in: "Mixed Integer Nonlinear Programming", S. Leyffer and J. Lee, Eds., The IMA Volumes in Mathematics and its Applications, Vol. 154. 1st Edition, 2012, pp. 513-529.

J. A. De Loera, R. Hemmecke, M. Köppe, and R. Weismantel. Integer polynomial optimization in fixed dimension. *Mathematics of Operations Research*, 31(1):147–153, 2006.

S. Drewes. *Mixed Integer Second Order Cone Programming*. PhD thesis, Technische Universität Darmstadt, 2009.

S. Drewes and S. Ulbrich. Subgradient based outer approximation for mixed integer second order cone programming. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 41–59. Springer New York, 2012. ISBN 978-1-4614-1926-6. doi: 10.1007/978-1-4614-1927-3_2. URL http://dx.doi.org/10.1007/978-1-4614-1927-3_2.

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

F. Eisenbrand. On the membership problem for the elementary closure of a polyhedron. *Combinatorica*, 19(2), 1999.

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.

References III

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.

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

R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Soc.*, 64:275–278, 1958.

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

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

R. C. Jeroslow. There cannot be any algorithm for integer programming with quadratic constraints. *Operations Research*, 21(1):221–224, 1973.

T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe, editors. *Evaluating Gas Network Capacities*. SIAM-MOS series on Optimization. SIAM, 2015. ISBN 978-1-611973-68-6. doi: 10.1137/1.9781611973696.

A. Koster, A. Zymolka, and M. Kutschka. Algorithms to separate 0,1/2-Chvátal-Gomory cuts. *Algorithmica*, 55(2): 375–391, 2009.

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

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

C. A. Meyer and C. A. Floudas. Convex envelopes for edge-concave functions. *Math. Program.*, 103(2):207–224, 2005.

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

J. Moré and G. Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerische Mathematik*, 55 (4):377–400, 1989. ISSN 0029-599X. doi: 10.1007/BF01396045. URL http://dx.doi.org/10.1007/BF01396045.

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.

M. Padberg. The boolean quadric polytope: Some characteristics, facets and relatives. *Mathematical Programming*, 45 (1-3):139–172, 1989. ISSN 0025-5610. doi: 10.1007/BF01589101. URL http://dx.doi.org/10.1007/BF01589101.

References IV

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. 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.

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.

D. Vandenbussche and G. L. Nemhauser. A polyhedral study of nonconvex quadratic programs with box constraints. *Mathematical Programming*, 2005. to appear.

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.

IBM®