Pierre Bonami

IBM ILOG CPLEX

MINO/COST Spring School – Paris – April 7 2016

Algorithms and Solvers for Mixed Integer Nonlinear (Convex) Optimization

©2016 IBM Corporation

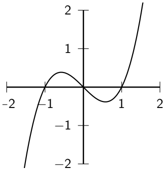
Introduction

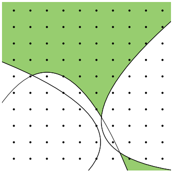
©2016 IBM Corporation

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

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

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



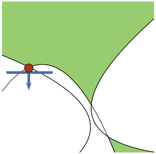


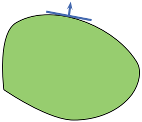
©2016 IBM Corporation

"Well solved" subproblems

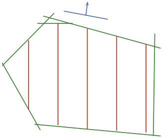
Nonlinear Programming (NLP)

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





Mixed-Integer linear programming (MILP)



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

©2016 IBM Corporation

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.

©2016 IBM Corporation

The complexity issue

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

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.

©2016 IBM Corporation

The complexity issue

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

Theo...

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

... even with 10 variables.

Theorem ([De Loera ...

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

©2016 IBM Corporation

More complexity results

In any dimension

- Mixed Integer Quadratic Programming in in NP [Del Pia et al., 2015].

In fixed dimension

- There is an FPTAS for minimizing a non-negative polynomial over the intersection of a polyhedron and integers [De Loera et al., 2006].
- Cubic and homogeneous polynomials can be minimized over integers in the plane in polynomial time [Del Pia et al., 2014].
- Minimizing polynomials of degrees four in the plane is NP-hard [De Loera et al., 2006].

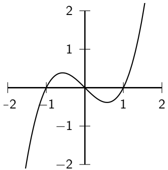
Many more results...

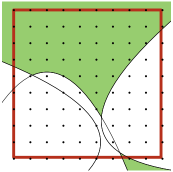
©2016 IBM Corporation

MINLP

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

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





©2016 IBM Corporation

Two main classes of MINLP

Mixed Integer Convex Program

Assume that the continuous relaxation is a convex optimization problem.

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

Mixed Integer Nonlinear Program (or Global 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 integers, variable x_j can be seen as a continuous variable satisfying:

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

©2016 IBM Corporation

A special class of convex MINLP: MISOCP

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

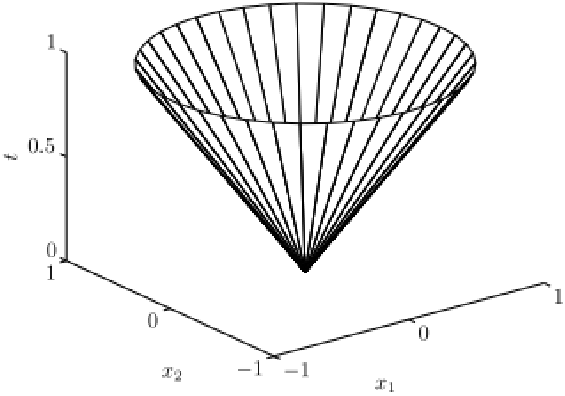
Where all quadratic constraints can be represented as second order cones (or Lorentz cone):

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

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

©2016 IBM Corporation

A Lorentz cone



It is convex!

©2016 IBM Corporation

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.

©2016 IBM Corporation

MISOCP

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

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 can think of).

©2016 IBM Corporation

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.		

©2016 IBM Corporation

Mixed Integer Convex Programming Applications (not MISOCP)

Application	nonlinear	discrete
Chemical plant design [Duran and Grossmann, 1986, Flores-Tlacuahuac and Biegler, 2007]	Chemical reactions	what to install
Block Layout Design [Castillo et al., 2005]	Spatial constraints	what to layout

©2016 IBM Corporation

Mixed Integer Nonlinear Programming Applications

Application	nonlinear	discrete
Petrochemical [Haverly, 1978]	Blending, pooling	–
Power industry [Carpentier, 1962]	AC power flow	–
Gaz/Water networks	gaz/liquid flow equations	choice of pipes
[Koch et al., 2015, Bragalli et al., 2011]		
Nuclear Reactor reloading [Quist et al., 1999]	reactions	What to reload
Airplane trajectories	aerodynamics	waypoints, colisions
[Cafieri and Durand, 2013, Soler et al., 2013]		
Mixed Integer Optimal control [Sager, 2005, 2012]	DE	discrete controls
Countless more	. . .	. . .
see for example [Belotti et al., 2013b]		

©2016 IBM Corporation

Agenda

- Part I: The Basic Algorithms for MICP.
  - Main Algorithmic Approaches
  - Glimpse of Computations
  - Glimpse of MISOCP
- Part II: Selected Advanced (or not) Topics.
  - Disaggregatin of nonlinear constraints
  - Lift-and-project cuts for MICP

©2016 IBM Corporation

Part I

The Basic Algorithms

©2016 IBM Corporation

The Basic Algorithms

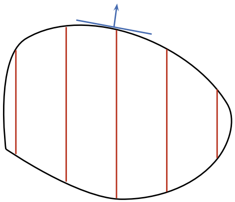
Section 1

The convex case

©2016 IBM Corporation

The mixed integer convex program

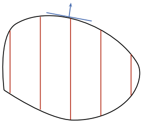
$$
\begin{array}{lll}
\min & c^T x & \\
\text{s.t.} & g_i(x) \leq 0 & i = 1, \ldots, m \\
& x \in X & \\
& x_j \in \mathbb{Z} & j = 1, \ldots, p
\end{array}
\qquad \text{(MICP)}
$$

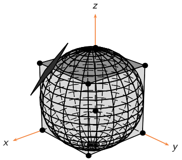


- $g_i : X \to \mathbb{R}$, $i = 1, \ldots, m$, *convex, differentiable.*
- Assume linear objective. If necessary, add var $\alpha \in \mathbb{R}$ and $\min \alpha$ with $f(x) \leq \alpha$ a constraint.

©2016 IBM Corporation

Main Algorithms for solving (MICP)





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].

©2016 IBM Corporation

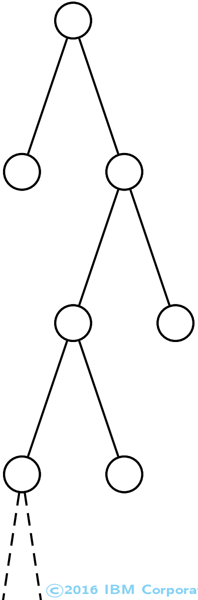
NLP based branch-and-bound

Straightforward generalization of main MILP algorithm:

- solve an NLP at each node of the tree.

©2016 IBM Corporation

NLP based branch-and-bound



Straightforward generalization of main MILP algorithm:

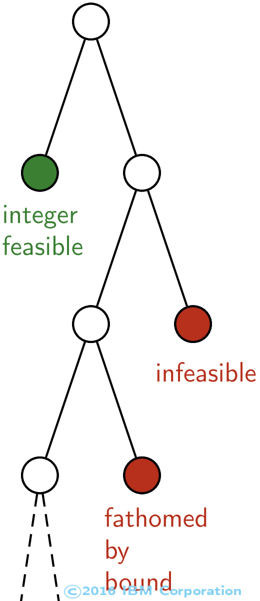
- solve an NLP at each node of the tree.
- Branch on variables with fractional value.

©2016 IBM Corporation

NLP based branch-and-bound

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.



©2016 IBM Corporation

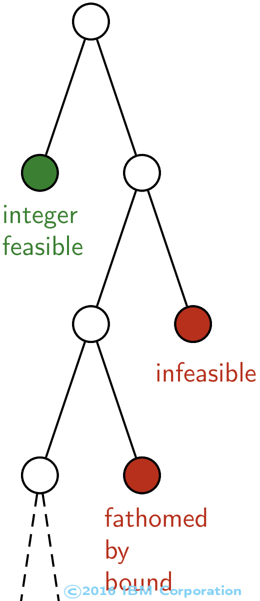
NLP based branch-and-bound

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.

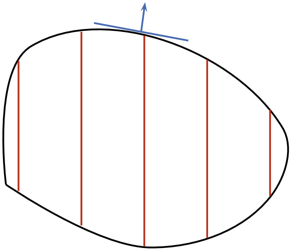
Main issues

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



©2016 IBM Corporation

Outer Approximation [Duran and Grossmann, 1986]

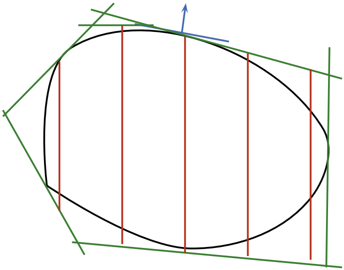


$$
\begin{aligned}
&\min \quad c^T x \\
&\text{s.t.} \\
&g_i(x) \leq 0 \qquad i = 1, \ldots, m, \\
&x_j \in \mathbb{Z}, \qquad j = 1, \ldots, p.
\end{aligned}
$$

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

©2016 IBM Corporation

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.

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

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

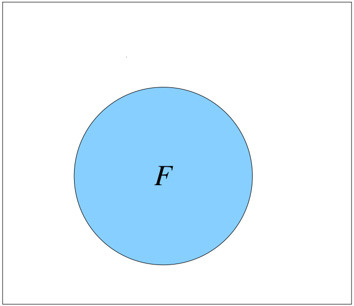
©2016 IBM Corporation

Outer approximation constraints

Let $F := \{x : x \in X : g_i(x) \leq 0\}$
$(g_i : \mathbb{R}^n \to \mathbb{R}$ convex. $)$
Outer approximation constraint in $\bar{x}$:

$$\nabla g_j(\bar{x})^T (x - \bar{x}) + g_j(\bar{x}) \leq g_j(x) \leq 0.$$

(valid for F by convexity of g_j and definition of F.)



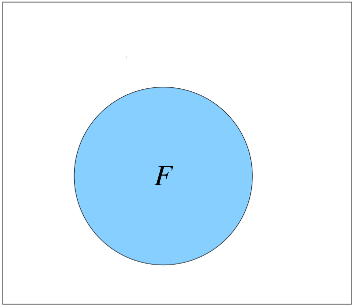
©2016 IBM Corporation

Outer approximation constraints

Let $F := \{x : x \in X : g_i(x) \leq 0\}$
($g_i : \mathbb{R}^n \to \mathbb{R}$ convex.)
Outer approximation constraint in $\bar{x}$:

$$\nabla g_j(\bar{x})^T (x - \bar{x}) + g_j(\bar{x}) \leq g_j(x) \leq 0.$$

(valid for F by convexity of g_j and definition of F.)



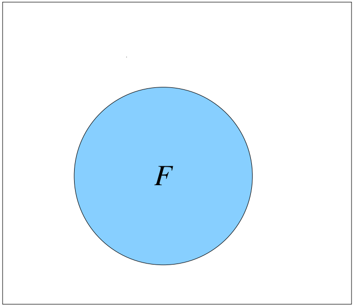
©2016 IBM Corporation

Outer approximation constraints

Let $F := \{x : x \in X : g_i(x) \leq 0\}$
($g_i : \mathbb{R}^n \to \mathbb{R}$ convex.)
Outer approximation constraint in $\bar{x}$:

$$\nabla g_j(\bar{x})^T (x - \bar{x}) + g_j(\bar{x}) \leq g_j(x) \leq 0.$$

(valid for F by convexity of g_j and definition of F.)



©2016 IBM Corporation

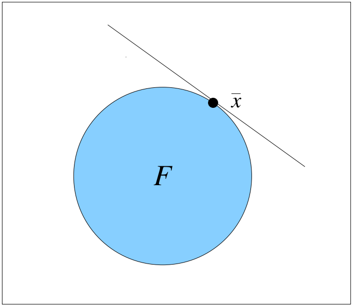
Outer approximation constraints

Let $F := \{x : x \in X : g_i(x) \leq 0\}$
($g_i : \mathbb{R}^n \to \mathbb{R}$ convex.)
Outer approximation constraint in $\bar{x}$:

$$\nabla g_j(\bar{x})^T (x - \bar{x}) + g_j(\bar{x}) \leq g_j(x) \leq 0.$$

(valid for F by convexity of g_j and definition of F.)

- If $g(\bar{x}) = 0$ tangent to feasible region.
- If $g(\bar{x}) < 0$ non-tight constraint.
- If $g(\bar{x}) > 0$ non-tight constraint cutting off $\bar{x}$.



©2016 IBM Corporation

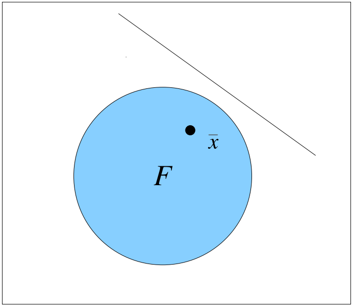
Outer approximation constraints

Let $F := \{x : x \in X : g_i(x) \leq 0\}$
($g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ convex.)
Outer approximation constraint in $\bar{x}$:

$$\nabla g_j(\bar{x})^T (x - \bar{x}) + g_j(\bar{x}) \leq g_j(x) \leq 0.$$

(valid for F by convexity of g_j and definition of F.)

- If $g(\bar{x}) = 0$ tangent to feasible region.
- If $g(\bar{x}) < 0$ non-tight constraint.
- If $g(\bar{x}) > 0$ non-tight constraint cutting off $\bar{x}$.



©2016 IBM Corporation

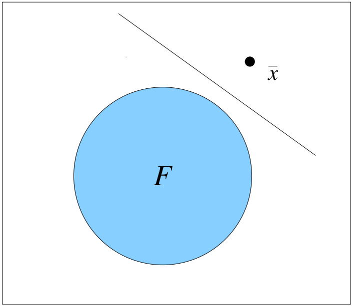
Outer approximation constraints

Let $F := \{x : x \in X : g_i(x) \leq 0\}$
$(g_i : \mathbb{R}^n \to \mathbb{R}$ convex. $)$
Outer approximation constraint in $\bar{x}$:

$$\nabla g_j(\bar{x})^T (x - \bar{x}) + g_j(\bar{x}) \leq g_j(x) \leq 0.$$

(valid for F by convexity of g_j and definition of F.)

- If $g(\bar{x}) = 0$ tangent to feasible region.
- If $g(\bar{x}) < 0$ non-tight constraint.
- If $g(\bar{x}) > 0$ non-tight constraint cutting off $\bar{x}$.



©2016 IBM Corporation

Subproblems

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

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

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

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

fixed feasibility subproblem

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

©2016 IBM Corporation

Subproblems

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

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

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

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

fixed feasibility subproblem

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

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($\hat{x}$))

©2016 IBM Corporation

Equivalent MILP formulation of convex MINLP

For each $\hat{x}^k \in K = \mathrm{Proj}_{1,\ldots,p}(X) \cap \mathbb{Z}^p$, 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

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

has the same optimal value as (MICP).

©2016 IBM Corporation

OA decomposition

Generate MILP equivalent by constraint generation.

- Initialize with one set of linearizations.

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

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

©2016 IBM Corporation

OA decomposition

Generate MILP equivalent by constraint generation.

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

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

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

©2016 IBM Corporation

OA decomposition

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 OA Theorem guarantees that the two bounds converge in finite # of iterations.

©2016 IBM Corporation

Outer-Approximation Decomposition Algorithm

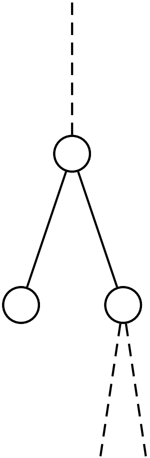
0. **Initialize.**
   $z_U \leftarrow +\infty$. $z_L \leftarrow -\infty$. Let $\overline{x}^0$ be the optimal solution of continuous relaxation.
   $\mathcal{K} \leftarrow \left\{\overline{x}^0\right\}$. Choose a convergence tolerance ϵ.
1. **Terminate?**
   Is $z_U - z_L < \epsilon$ or $(\mathrm{OA}(\mathcal{K}))$ infeasible? If so z_U is $\epsilon-$optimal.
2. **Lower Bound**
   Let $z_{\mathrm{MP}(\mathcal{K})}$ be the optimal value of $\mathrm{OA}(\mathcal{K})$ and $(\hat{x})$ its optimal solution.
   $z_L \leftarrow z_{\mathrm{MP}(\mathcal{K})}$
3. **NLP Solve**
   Solve $(\mathrm{NLP}(\hat{x}))$.
   Let $\overline{x}^i$ be the optimal (or minimally infeasible) solution.
4. **Upper Bound?**
   Is $\overline{x}^i$ feasible for (MINLP)? If so, $z_U \leftarrow \min(z_U, f(\overline{x}^i))$.
5. **Refine**
   $\mathcal{K} \leftarrow \mathcal{K} \cup \{\overline{x}^i\}$ and $i \leftarrow i + 1$.
   Go to 1.

©2016 IBM Corporation

LP/NLP Branch-and-bound

OA can be embedded in a single tree search.

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

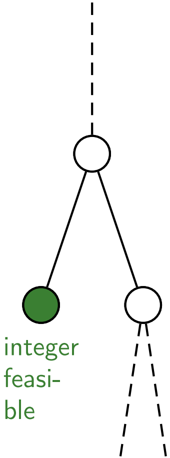


©2016 IBM Corporation

LP/NLP Branch-and-bound

OA can be embedded in a single tree search.

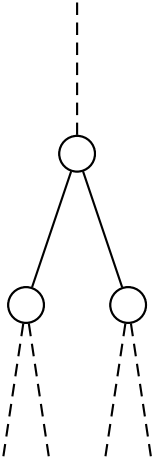
- Start solving the same initial MILP by branch-and-bound.
- At each integer feasible node:
  1. solve (NLP($\hat{x}$)), and enrich the set of linearizations.
  2. Resolve the LP relaxation of the node with the new cuts.
  3. Repeat as long as node is integer feasible.



©2016 IBM Corporation

LP/NLP Branch-and-bound

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.

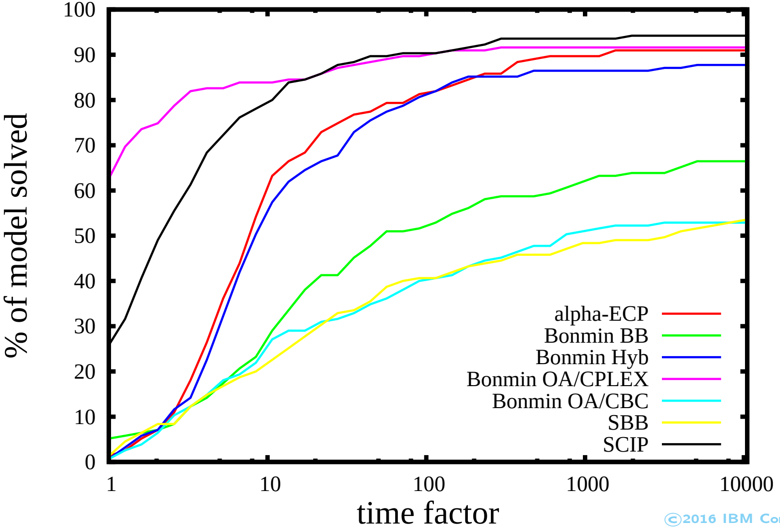
©2016 IBM Corporation

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, 2012]	LP/NLP

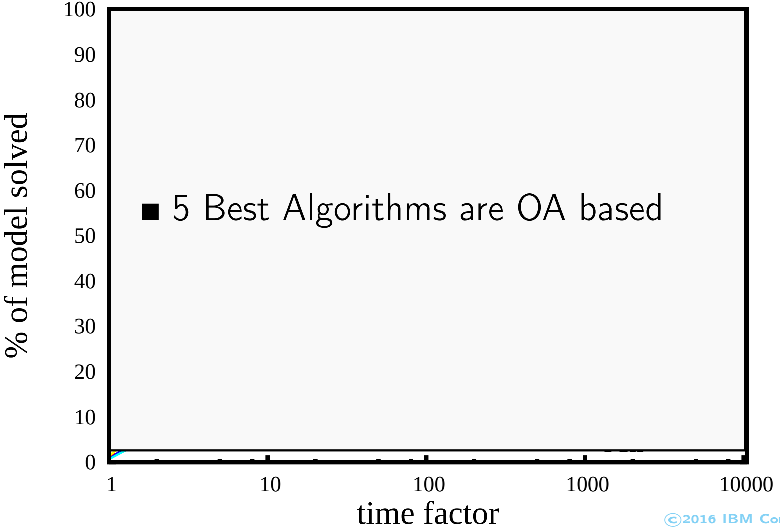
©2016 IBM Corporation

Comparison of solvers in GAMS [Vigerske, 2013]



 ©2016 IBM Corporation

Comparison of solvers in GAMS [Vigerske, 2013]



- 5 Best Algorithms are OA based

©2016 IBM Corporation

Notes on results with Bonmin

- 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. Doesn't reuse much from Cbc. Everything specifically tailored.
  - Better implementation exists that should be on par with Hyb.
- Bonmin's OA using CBC seems the worst algorithm overall.

©2016 IBM Corporation

Notes on results with Bonmin

- 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. Doesn't reuse much from Cbc. Everything specifically tailored.
  - Better implementation exists that should be on par with Hyb.
- Bonmin's OA using CBC seems the worst algorithm overall.

©2016 IBM Corporation

The MIQCP/MISOCP solver in CPLEX

Implements the two main algorithms:

- A branch-and-bound based on the continuous SOCP solver (barrier).
- An outer approximation branch-and-cut algorithm.

Choice is controled by the parameter `CPXPARAM_MIP_Strategy_MIQCPStrat`.
Default is trying to decide which of the two algorithms to run in a "clever" way.

History of MIQCP with CPLEX

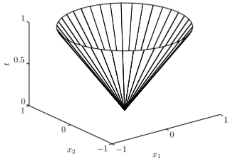
class	p	algorithm	V. (Year)
Convex QCP	0	barrier	9.0 (2003)
convex MIQCP	> 0	barrier based B&B	9.0 (2003)
–	–	Outer approximation B&C	11.0 (2007)

©2016 IBM Corporation

MISOCP specifics

Remember our assumption for MICP algorithms and the Lorentz cone:

Assumption:
$g_i : X \to \mathbb{R}$, $i = 1, \ldots, m$, convex, differentiable.
We have:



$$g_i(x) = \sum_{i=1}^{d} x_i^2 - x_0^2, \text{ with } x_0 \geq 0$$

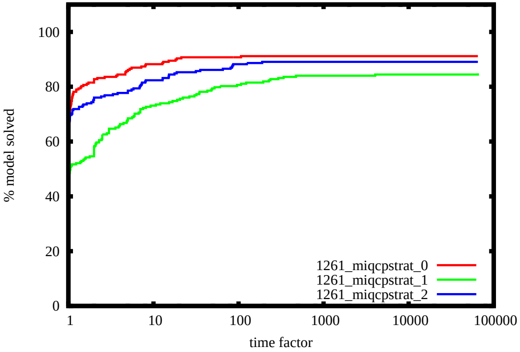
The assumption is not verified at the head of the cone (when $x_0 = 0$), in particular Outer Approximation can't be applied as is.

In practice

[Drewes, 2009, Drewes and Ulbrich, 2012] recommend a hybrid OA approach where the SOCP relaxation is applied and used at integer feasible nodes where we are at the head of a cone.

©2016 IBM Corporation

A comparison of OA and SOCP-BB in CPLEX 12.6.1 [1]



Default strategy picked

- OA 113 times
- SOCP-BB 46 times
- 56 models identical with both

To be perfect should have picked

- 14 more models with OA
- 36 more models with SOCP-BB

[1] 225 models solved by at least one method and failed by none.

©2016 IBM Corporation

Advanced algorithms for convex case (non exhaustive references)

- Preprocessing/Modeling:
  - separability [Hijazi et al., 14]
  - perspective formulations [Frangioni and Gentile, 2006, Günlük and Linderoth, 2008]
  - propagation [Vigerske, 2012]
- Node relaxations/Branching:
  - QP Delaxations in strong-branching [Bonami et al., 2013]
  - QP 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].
  - Conic Cuts for Conic Crogramming [Andersen and Jensen, 2013, Belotti et al., 2013a, Kılınç-Karzan and Yıldız, 2015, Modaresi et al., 2015] (among others)

©2016 IBM Corporation

Part II

Selected Advanced (or not) Topics

©2016 IBM Corporation

Selected Advanced (or not) Topics

Section 2

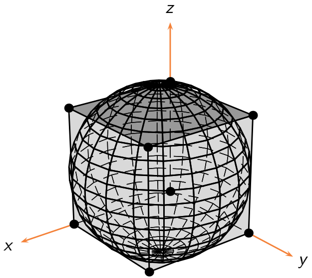
Disaggregation of nonlinear constraint

©2016 IBM Corporation

A most simple MINLP

Consider the following convex MINLP:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} i \times x_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} \left(x_i - \tfrac{1}{2}\right)^2 \leq \tfrac{n-1}{4} \qquad (1) \\
& x \in \mathbb{Z}^n
\end{aligned}
$$



Exercise

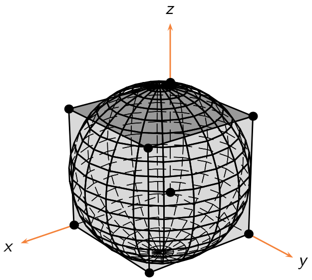
- Find the optimum or prove that (1) is infeasible or unbounded.
- How many nodes, would a simple branch-and-bound take to solve (1)?
- How many linear approximations would an Outer Approximation approach need?

 ©2016 IBM Corporation

Answers

Consider the following convex MINLP:

$$
\begin{array}{ll}
\min & \sum_{i=1}^{n} i \times x_i \\
\text{s.t.} & \sum_{i=1}^{n} \left(x_i - \frac{1}{2}\right)^2 \leq \frac{n-1}{4} \\
& -10 \leq x \leq 10, x \in \mathbb{Z}^n
\end{array}
\qquad (1)
$$

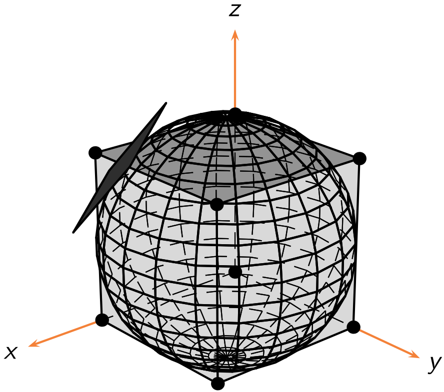


- (1) is infeasible:
  - The ball is too small to contain integer points.
  - It is large enough to touch every edge of the hypercube.
- A basic branch-and-bound would take at least 2^{n+1} nodes.
- We need at least 2^n linear outer approximations to prove infeasibility.

©2016 IBM Corporation

Solving (1) with OA cuts

- No OA constraint can cut 2 vertices of the hypercube.
  - If an inequality cuts two points, it cuts the segment joining them.
  - The ball has a non-empty intersection with every segment joining two vertices.
  - Remember that an outer approximation is only a tangent to the ball.

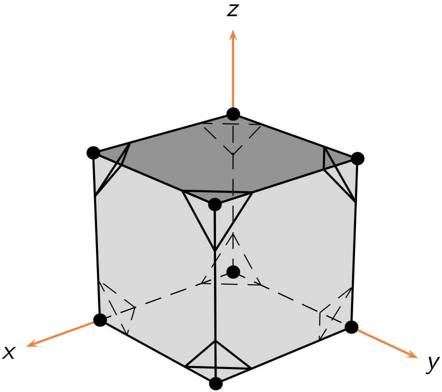


What do solvers tell?

©2016 IBM Corporation

Solving (1) with OA cuts

- No OA constraint can cut 2 vertices of the hypercube.
  - If an inequality cuts two points, it cuts the segment joining them.
  - The ball has a non-empty intersection with every segment joining two vertices.
  - Remember that an outer approximation is only a tangent to the ball.



What do solvers tell?

©2016 IBM Corporation

Four years ago

		CPLEX 12.4	SCIP 2.0.1	B-OA	B-Hyb
n	2^n	nodes	nodes	OA it.	nodes

Four years ago

		CPLEX 12.4	SCIP 2.0.1	B-OA	B-Hyb
n	2^n	nodes	nodes	OA it.	nodes
10	1,024	2,047	720	1,105	11,156

Four years ago

		CPLEX 12.4	SCIP 2.0.1	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

Four years ago

		CPLEX 12.4	SCIP 2.0.1	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	...	...

©2016 IBM Corporation

Four years ago

		CPLEX 12.4	SCIP 2.0.1	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 trivial if variables are $0-1$: replace x_i^2 by x_i, the contradiction $\frac{n}{4} \leq \frac{n-1}{4}$ follows.

©2016 IBM Corporation

Four years ago

		CPLEX 12.4	SCIP 2.0.1	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 trivial if variables are $0-1$: replace x_i^2 by x_i, the contradiction $\frac{n}{4} \leq \frac{n-1}{4}$ follows.
- SCIP $\geq$ 2.1 and CPLEX $\geq$ 12.6.1 solve it in a blink.

©2016 IBM Corporation

Solving the problem by presolve/propagation

An easy way to deduce infeasibility is to compute the component-wise maximum of the left-hand-side of the constraint:

$$\sum_{i=1}^{n} \min \left\{ \left(x_i - \frac{1}{2} \right)^2 : x_i \in \mathbb{Z} \right\}$$

Each optimization problem is one dimensional and can be easily solved:

$$\min \left\{ \left(x_i - \frac{1}{2} \right)^2 : x_i \in \mathbb{Z} \right\} = \frac{1}{4}$$

Summing up we get that:

$$\sum_{i=1}^{n} \min \left\{ \left(x_i - \frac{1}{2} \right)^2 : x_i \in \mathbb{Z} \right\} = \frac{n}{4} > \frac{n-1}{4}.$$

A contradiction, therefore (1) is infeasible.

©2016 IBM Corporation

Twisting our example

The following model should be complicated enough to pass presolve untouched:

$$\begin{aligned} &\min \sum_{i=1}^{2n} i * x_i \\ &\sum_{i=1}^{n} (100x_{2i}^2 + 100x_{2i-1}^2 - 4x_{2i}x_{2i-1} - 98x_{2i} - 98x_{2i-1}) \leq -1 \\ &-10 \leq x \leq 10, x \in \mathbb{Z}^{2n} \end{aligned} \tag{2}$$

©2016 IBM Corporation

A recipe for solving (2) better with OA

We consider a specific class of MINLPs:

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

- For $i = 1, \dots, m$, $g_i : X \to R$ 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.

©2016 IBM Corporation

Disaggregated formulation

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

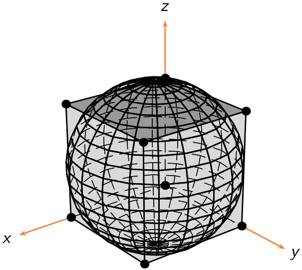
$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & \sum_{j=1}^{n} y_{ij} \leq 0 && i = 1, \dots, m, \\
& g_{ij}(x_j) \leq y_{ij} && \begin{array}{l} i = 1, \dots, m, \\ j = 1, \dots, n, \end{array} \\
& x \in X, \\
& x_i \in \mathbb{Z} && i = 1, \dots, p, \\
& l \leq x \leq u.
\end{aligned}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

©2016 IBM Corporation

Application to (1)

Extended formulation of (1)

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i \le (n-1)/4 \\ & (x_i - 0.5)^2 \le y_i \qquad i = 1, \dots, n \\ & x \in \mathbb{Z}^n. \end{aligned} \tag{3}$$



Its outer approximation

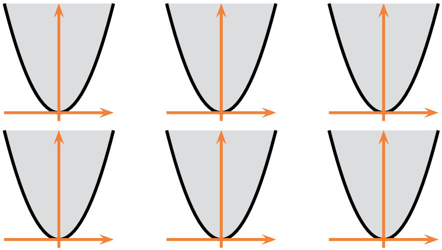
$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i \le (n-1)/4 \\ & 2\left(\overline{x}_i^k - 0.5\right)\left(x_i - \overline{x}_i^k\right) + \left(\overline{x}_i^k - 0.5\right)^2 \le y_i \qquad \begin{matrix} i = 1, \dots, n \\ k = 1, \dots, K \end{matrix} \\ & x \in \mathbb{Z}^n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

©2016 IBM Corporation

Application to (1)

Extended formulation of (1)

$$
\begin{aligned}
\min\ & c^T x \\
\text{s.t.} & \sum_{i=1}^{n} y_i \le (n-1)/4 \\
& (x_i - 0.5)^2 \le y_i \qquad i = 1, \ldots, n \\
& x \in \mathbb{Z}^n.
\end{aligned} \tag{3}
$$



Its outer approximation

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

©2016 IBM Corporation

Application to (1)

Extended formulation of (1)

$$\min \; c^T x$$

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

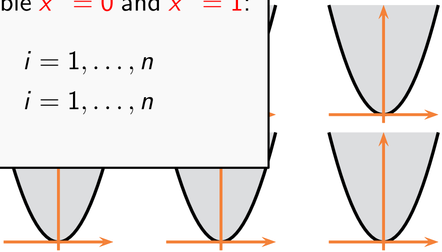
$$(x_i - 0.5) \le y_i \qquad i = 1, \dots, n$$

$$x \in \mathbb{Z}^n. \qquad (3)$$

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

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

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



Its outer approximation

$$\min \; c^T x$$

$$\text{s.t.} \sum_{i=1}^{n} y_i \le (n-1)/4$$

$$2\left(\overline{x}_i^k - 0.5\right)\left(x_i - \overline{x}_i^k\right) + \left(\overline{x}_i^k - 0.5\right)^2 \le y_i \qquad \begin{matrix} i = 1, \dots, n \\ k = 1, \dots, K \end{matrix}$$

$$x \in \mathbb{Z}^n$$

©2016 IBM Corporation

Application to (2): Solution

We need to get from this

$$\sum_{i=1}^{n}(100x_{2i}^2 + 100x_{2i-1}^2 - 4x_{2i}x_{2i-1} - 98x_{2i} - 98x_{2i-1}) \leq -1$$

to something of the form:

$$\begin{aligned} &\textstyle\sum_{i=1}^{n}(\alpha z_{2i} + \beta z_{2i-1} - 98x_{2i} - 98x_{2i-1}) \leq -1 \\ &y_i^2 \leq z_i \\ &y_i = \gamma_i^T x. \end{aligned}$$

How do we find $\alpha, \beta and \gamma$?
Spectral decompostion:

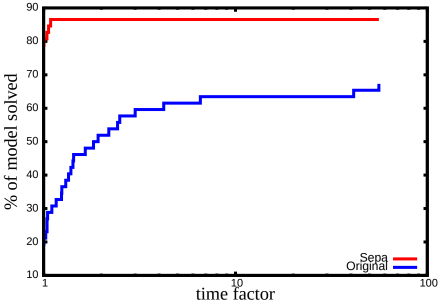
$$\begin{pmatrix} 100 & -2 \\ -2 & 100 \end{pmatrix} = \begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} 51 & 0 \\ 0 & 49 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix}$$

$$\alpha = 51, \beta = 49, \gamma_{2i} = (-1, 1), \gamma_{2i-1} = (-1, -1)$$

©2016 IBM Corporation

Experimental Illustration

- 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].



Similar technique developped in Baron for compositions of convex functions [Tawarmalani and Sahinidis, 2004].

©2016 IBM Corporation

Disaggregation of Second Order cones [Vielma et al., 2015]

In standard form the nonlinear constraint describing the second order cone is not convex separable:

$$\sum_{i=1}^{n} x_i^2 \leq x_0^2$$

Trick , divide the constraint by $x_0 \geq 0$ to get a convex separable constraint:

$$\sum_{i=1}^{n} \frac{x_i^2}{x_0} \leq x_0.$$

Now introduce $y_1, \ldots, y_n$ and rewrite as:

$$\sum_{i=1}^{n} y_i \leq x_0$$

$$x_i^2 \leq x_0 y_i$$

©2016 IBM Corporation

General cones and Disciplined Convex Programming [Grant et al., 2006]

More general convex cones can be used to model convex problems:

$$EXP := cl\{(x, y, z) \in R^3 : ye^{\frac{x}{y}} \leq z, y \geq 0\},$$
$$POW_\alpha := \{(x, y, z) \in \mathbb{R}^3 : |z| \leq x^\alpha y^{1-\alpha}, x, y \geq 0\}$$

Adding these to MISOCP, convex problem can be modeled in a *disciplined* way by using composition of functions such that convexity is verifiable leading to DCP (see CVX [Grant and Boyd, 2014]) and MIDCP.
Out of 333 convex models in MINLPLIB2 [Vigerske, 2015] 333 can be reformulated in a disciplined manner [Lubin et al., 2015]!

©2016 IBM Corporation

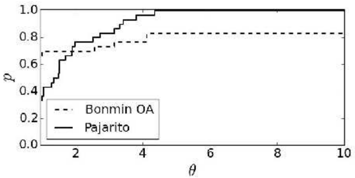
Disaggregation and MIDCP [Lubin et al., 2015]

The compositions used in DCP have a 1-1 correspondence with the existence of extended formulation disaggregating the cones and generalize the notion of separability.

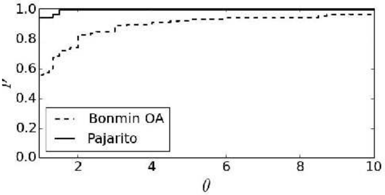
Pajarito sovler

- Solver for MIDCP implemented in Julia/JUMP [?Dunning et al., 2015].
- Claimed less than 1000 lines of codes.
- Dominates Bonmin OA on MINLPLIB2

Solution Time



Iterations



©2016 IBM Corporation

Selected Advanced (or not) Topics

Section 3

Lift-and-project cuts for MICP

©2016 IBM Corporation

Lift-and-project cuts for MISOCP

- Cuts are an essential component of MILP solvers
- Can always apply MILP cuts to a linear OA (and most people do)
- 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 [Andersen and Jensen, 2013, Belotti et al., 2013a, Kılınç-Karzan and Yıldız, 2015, Modaresi et al., 2015] (among others)

Our goal

- Derive linear cutting planes
- Fast
- Find an appropriate OA from which to derive a cut

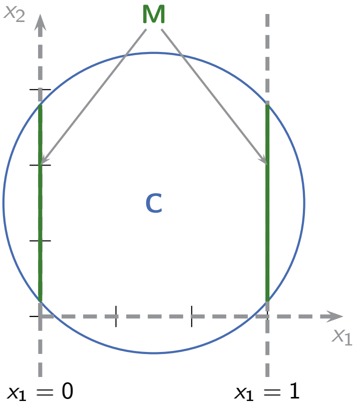
©2016 IBM Corporation

Split Relaxation

Consider $\mathbf{C}$ and $\mathbf{M} := \mathbf{C} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})$.
Let $\pi \in \mathbb{Z}^p \times \{0\}^{n-p}$, $\pi_0 \in \mathbb{Z}$ and

$$\mathbf{C}^{(\pi,\pi_0)} := \text{conv}\Big(\mathbf{C} \cap \big(\{x : \pi^T x \le \pi_0\} \cup \{x : \pi^T x \ge \pi_0 + 1\}\big)\Big).$$

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



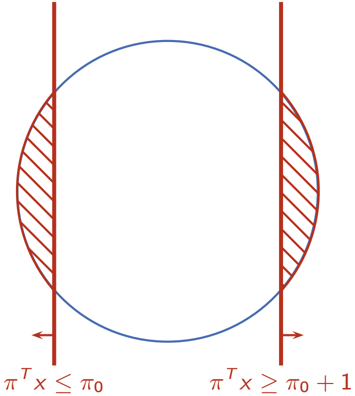
©2016 IBM Corporation

Split Relaxation

Consider **C** and **M** $:= \mathbf{C} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})$.
Let $\pi \in \mathbb{Z}^p \times \{0\}^{n-p}$, $\pi_0 \in \mathbb{Z}$ and

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

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



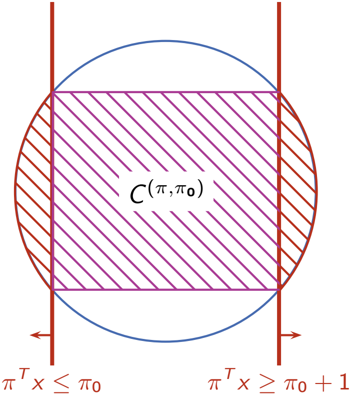
©2016 IBM Corporation

Split Relaxation

Consider $\mathbf{C}$ and $\mathbf{M} := \mathbf{C} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})$.
Let $\pi \in \mathbb{Z}^p \times \{0\}^{n-p}$, $\pi_0 \in \mathbb{Z}$ and

$$\mathbf{C}^{(\pi,\pi_0)} := \text{conv}\Big(\mathbf{C} \cap \big(\{x : \pi^T x \le \pi_0\} \cup \{x : \pi^T x \ge \pi_0 + 1\}\big)\Big).$$

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



©2016 IBM Corporation

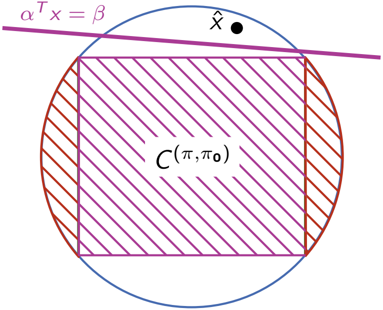
Split Relaxation

Consider $\mathbf{C}$ and $\mathbf{M} := \mathbf{C} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})$.
Let $\pi \in \mathbb{Z}^p \times \{0\}^{n-p}$, $\pi_0 \in \mathbb{Z}$ and

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

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

In the remainder, $\hat{x}$ is the point to separate,
$\pi = e_k$, $\hat{x}_k \in]0,1[$ $(k \leq p)$, and $\pi_0 = 0$



©2016 IBM Corporation

MILP case

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

Cut Generation LP [Balas, 1979, Balas et al., 1993]

$\hat{x} \in P$ is separated from $P^{(e_k,0)}$ using the LP:

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

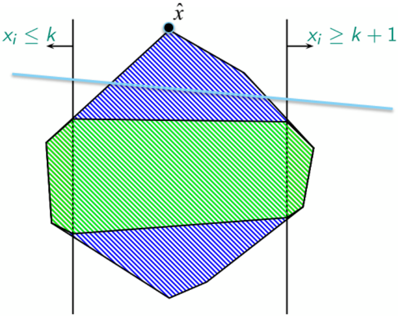
If $\hat{x} \notin P^{(e_k,0)}$, $\alpha^T x \geq \beta$ cuts $\hat{x}$; otherwise produces certificate that $\hat{x} \in P^{(e_k,0)}$ with $x^0 \in P \cap \{x_k = 0\}$, $x^1 \in P \cap \{x_k = 1\}$ such that $\hat{x} = \hat{x}_k x^1 + (1 - \hat{x}_k) x^0$.

©2016 IBM Corporation

Statement in picture

If $\hat{x}$ is not in the split relaxation we get a cut.

$$\alpha = u^T A - u_0 e_k, \quad \beta = u^T b.$$

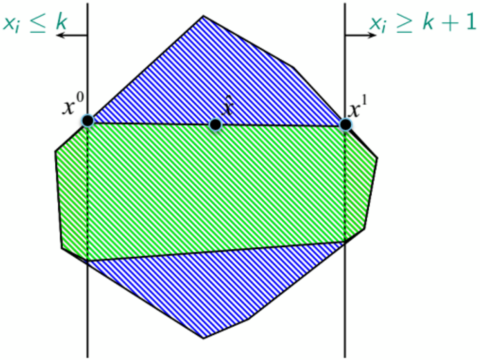


$$\alpha = v^T A + v_0 e_k, \quad \beta = v^T b + v_0.$$

©2016 IBM Corporation

Statement in picture

If $\hat{x}$ is in the split relaxation we get a certificate in the form of two points x^0, x^1.



©2016 IBM Corporation

MILP case (primal view)

Membership LP [Bonami, 2012]

$\hat{x} \in P$ wih $0 < \hat{x}_k < 1$ also in $P^{(e_k,0)}$ if $\exists\ x^0 \in P \cap \{x_k = 0\}$ and $x^1 \in P \cap \{x_k = 1\}$ with $\hat{x} = \hat{x}_k x^1 + (1 - \hat{x}_k)x^0$, or if

$$\begin{aligned} \max\ & y_k \\ \text{s.t.} & \\ & Ay = b\hat{x}_k \\ & 0 \leq y \leq \hat{x}, \\ & y \in \mathbb{R}^n. \end{aligned} \tag{MLP}$$

has a solution with $y_k = \hat{x}_k$ otherwise can deduce a cut from dual optimal solution.
(Hint $\frac{y}{\hat{x}_k}$ is x^1, $\frac{\hat{x}-y}{1-\hat{x}_k}$ is x^0).

©2016 IBM Corporation

Generalization to MICPs

Using the primal view

- Generalizing (MLP) to nonlinear convex constraints is relatively simple [Bonami, 2011].
- But Nonlinear programming duality is not the same as LP!

Using the dual view

- Generalizing CGLP is possible but poses many numerical/technical challenges[Ceria and Soares, 1999, Stubbs and Mehrotra, 1999].
- As long as we generate a linear cut, it can be obtained from linear outer approximations[Bonami et al., 2012].
- The linear case can be used within a cut generation framework [Kılınc et al., 2011].

©2016 IBM Corporation

Generalization to MICP: Two competing approaches

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 a cut is found STOP otherwise
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 there is no cut, STOP.

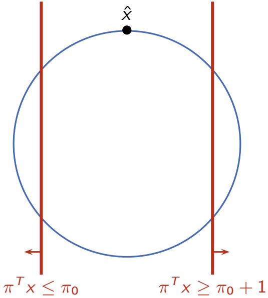
Using NLP [Bonami, 2011]

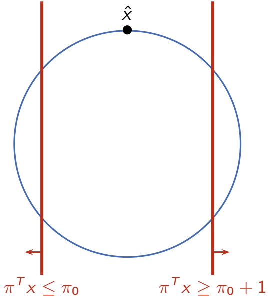
1. Solve a single NMLP that tells if $\hat{x}$ is in the split relaxation.
2. If not, deduce from solution two points such that $\hat{x} = \lambda x^1 + (1-\lambda)x^0$ and closest to the disjunction.
3. Build OA around these two points.
4. Solve MLP and get the cut.

©2016 IBM Corporation

Generalization: Two competing approaches in pictures

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

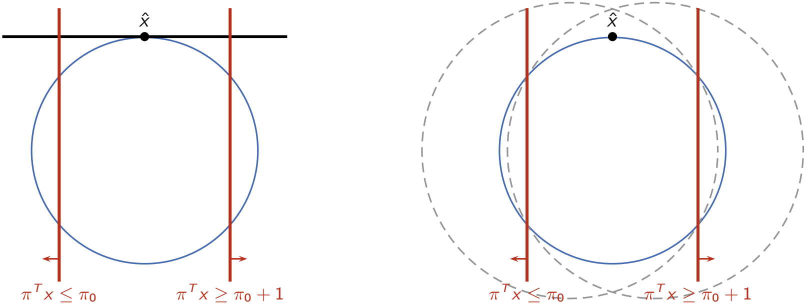




©2016 IBM Corporation

Generalization: Two competing approaches in pictures

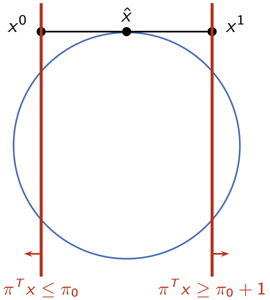
Goal: build a linear OA from which a "best" cut can be deduced using CGLP.

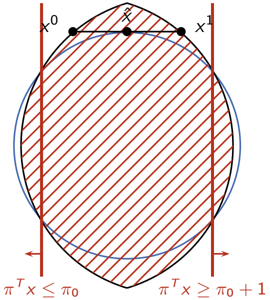


©2016 IBM Corporation

Generalization: Two competing approaches in pictures

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

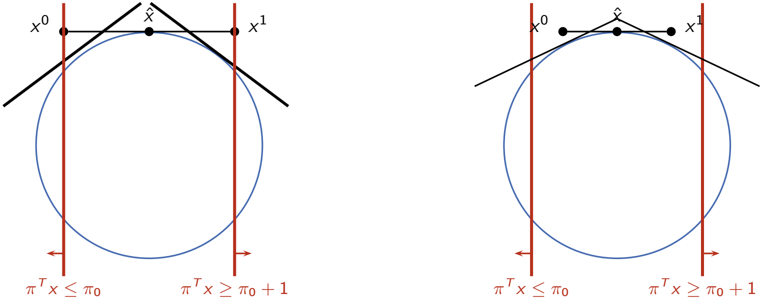




©2016 IBM Corporation

Generalization: Two competing approaches in pictures

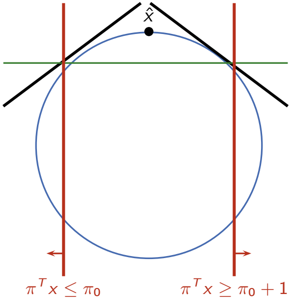
Goal: build a linear OA from which a "best" cut can be deduced using CGLP.

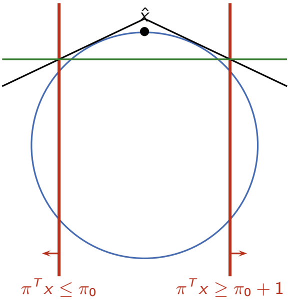


©2016 IBM Corporation

Generalization: Two competing approaches in pictures

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

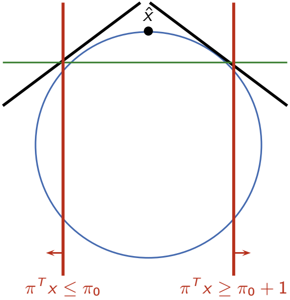


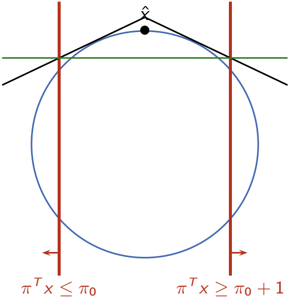


©2016 IBM Corporation

Generalization: Two competing approaches in pictures

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





©2016 IBM Corporation

Snapshot of results

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

Combination with disaggregation[Kılınç, 2011]

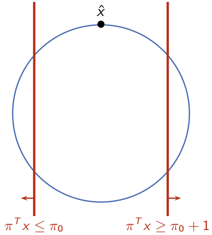
Even better results are obtained by combining the extended formulation trick for separability and these cuts.

		Original		Extended	
	n	root gap	sol time	root gap	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	5.0
uflquad	15	10.85	784	96.25	145

©2016 IBM Corporation

Sketch of an algorithm for MISOCP in CPLEX 12.6.2

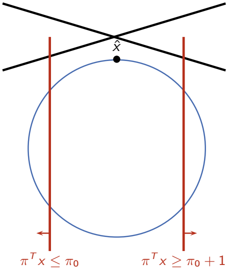
- Only solve LPs,
- Dynamic generation of additional OA constraints.
- compact formulation using MLP,



©2016 IBM Corporation

Sketch of an algorithm for MISOCP in CPLEX 12.6.2

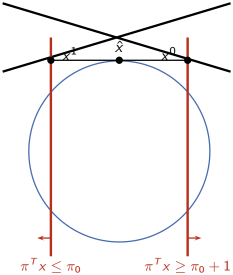
- Only solve LPs,
- Dynamic generation of additional OA constraints.
- compact formulation using MLP,



©2016 IBM Corporation

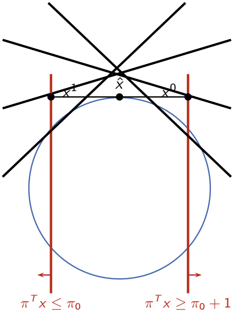
Sketch of an algorithm for MISOCP in CPLEX 12.6.2

- Only solve LPs,
- Dynamic generation of additional OA constraints.
- compact formulation using MLP,



©2016 IBM Corporation

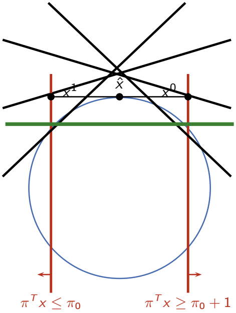
Sketch of an algorithm for MISOCP in CPLEX 12.6.2



- Only solve LPs,
- Dynamic generation of additional OA constraints.
- compact formulation using MLP,

©2016 IBM Corporation

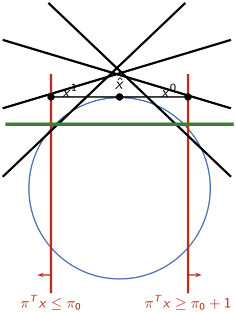
Sketch of an algorithm for MISOCP in CPLEX 12.6.2



- Only solve LPs,
- Dynamic generation of additional OA constraints.
- compact formulation using MLP,

©2016 IBM Corporation

Sketch of an algorithm for MISOCP in CPLEX 12.6.2



- Only solve LPs,
- Dynamic generation of additional OA constraints.
- compact formulation using MLP,

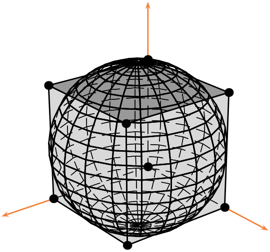
Plus new recent features

- Propagation of conic constraints (12.6.1).
- Cone disaggregation for MISOCP (12.6.2).
- Redesigned heuristic choice of most promising algorithm (12.6.2).
- Improved OA Cuts using perspective reformulation (12.6.2).

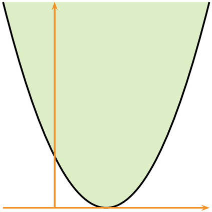
©2016 IBM Corporation

Putting it all together

[Cornuéjols and Li, 2001] showed that the empty ball in dimension n has *split rank n*
$\Rightarrow$ Practically unsolvable using any known cutting plane technique.



Instead the disaggregated formulation has (simple) split rank 1.

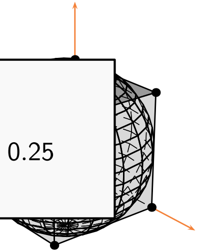


©2016 IBM Corporation

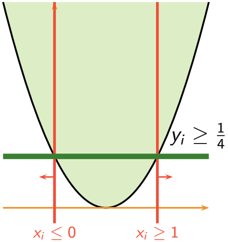
Putting it all together

[Cornuéjol
the empty
rank n
⇒ Practic
known cut

$$\left.\begin{array}{ll} -x_i + 0.25 \leq y_i & i = 1, \ldots, n \\ x_i - 0.75 + \leq y_i & i = 1, \ldots, n \\ x_i \leq 0 \text{ OR } x_i \geq 1 & \end{array}\right\} \Rightarrow y_i \geq 0.25$$



Instead the disaggregated formulation has (simple) split rank 1.



©2016 IBM Corporation

The effect on our ellipse

$$\min \sum_{i=1}^{2n} i * x_i$$
$$\sum_{i=1}^{n}(100x_{2i}^2 + 100x_{2i-1}^2 - 4x_{2i}x_{2i-1} - 98x_{2i} - 98x_{2i-1}) \leq -1 \qquad (2)$$
$$x \in \mathbb{Z}^{2n}$$

- results on 12 threads with 12.6.1, 12.6.3[2], 12.6.3-- (no lift-and-project cuts) and 12.6.3++ (aggressive lift-and-project cuts), 3 hours time limit

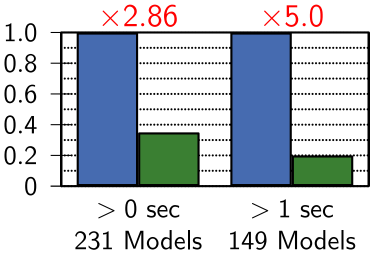
	12.6.1	12.6.2--	12.6.2	12.6.2++
n	nodes	nodes	nodes	nodes
5	2,261	2,045	2,045	1,825
10	2,097,151	1,914,797	29	1
15	>23,125,426	>146,604,478	7,769	1

(Largest model solved in 2.2 sec by 12.6.3, in 5.5 sec by 12.6.2++.)

[2]Default results can be very sensitive to objective function

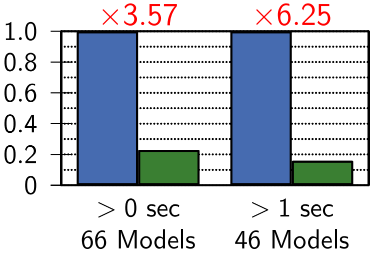
©2016 IBM Corporation

CPLEX 12.6.1 vs 12.6.3



CPLEX test bed

- CPLEX 12.6.1: 62 time limits
- CPLEX 12.6.3: 38 time limits

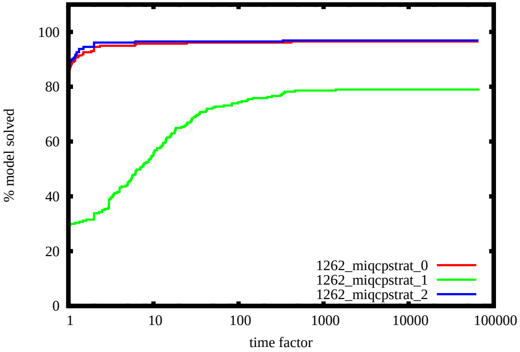


CBLIB

- CPLEX 12.6.1: 17 time limits
- CPLEX 12.6.3: 8 time limits

©2016 IBM Corporation

A comparison of OA and SOCP-BB in CPLEX 12.6.3 [3]



Default strategy picked

- OA 186 times
- SOCP-BB 4 times
- 55 models identical with both

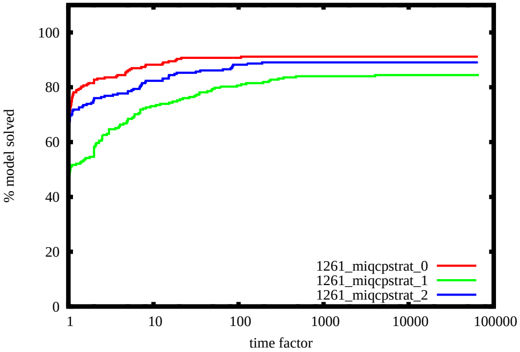
To be perfect should have picked

- 2 more models with OA
- 9 more models with SOCP-BB

[3] 245 models solved by at least one method and failed by none.

©2016 IBM Corporation

Reminder of CPLEX 12.6.1 [4]



Default strategy picked

- OA 113 times
- SOCP-BB 46 times
- 56 models identical with both

To be perfect should have picked

- 14 more models with OA
- 36 more models with SOCP-BB

[4] 225 models solved by at least one method and failed by none.

©2016 IBM Corporation

Conclusion

- MINLP is still very challenging.
- Some significant applications solved but many still out of reach.
- According to S. Vigerske MINLP accelerates at a rate of 1.96/*year* (more than MILP's 1.8 claimed by B. Bixby).
- Conjecture: 25 years from now MINLPs will be solved 2.024×10^7 faster than today.
- Disciplined approach for MICP seems very promising.
- To get there we need:
  - more applications:
    `www.minlp.org`,
  - more benchmark instances:
    `www.gamsworld.org/minlp/minlplib2/html/`
  - more clever people!

©2016 IBM Corporation

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.

 ©2016 IBM Corporation

References I

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. 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. Andersen and A. Jensen. Intersection cuts for mixed integer conic quadratic sets. In M. Goemans and J. Correa, editors, *Integer Programming and Combinatorial Optimization*, volume 7801 of *Lecture Notes in Computer Science*, pages 37–48. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36693-2. doi: 10.1007/978-3-642-36694-9_4. URL http://dx.doi.org/10.1007/978-3-642-36694-9_4.

E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.

E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Programming*, 58:295–324, 1993.

P. Belotti, J. C. Góez, I. Pólik, T. K. Ralphs, and T. Terlaky. On families of quadratic surfaces having fixed intersections with two hyperplanes. *Discrete Applied Mathematics*, 161(16–17):2778 – 2793, 2013a. ISSN 0166-218X. doi: http://dx.doi.org/10.1016/j.dam.2013.05.017. URL http://www.sciencedirect.com/science/article/pii/S0166218X13002461.

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

Bonami. On optimizing over lift-and-project closures. *Mathematical Programming Computation*, 4(2):151–179, 2012. ISSN 1867-2949. doi: 10.1007/s12532-012-0037-0. URL http://dx.doi.org/10.1007/s12532-012-0037-0.

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

 ©2016 IBM Corporation

References II

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. Linderoth, and A. Lodi. *Chapter 18*, chapter Disjunctive Cuts for Mixed Integer Nonlinear Programming Problems, pages 521–541. Progress in Combinatorial Optimization. ISTe-Wiley, 2012.

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

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.

J. Carpentier. Contributions to the economic dispatch problem. *Bulletin Societé Française des Électriciens*, 8:431–447, 1962.

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.

S. Ceria and J. Soares. Convex programming for disjunctive optimization. *Mathematical Programming*, 86:595–614, 1999.

©2016 IBM Corporation

References III

G. Cornuéjols and Y. Li. On the rank of mixed 0,1 polyhedra. In K. Aardal and B. Gerards, editors, *Integer Programming and Combinatorial Optimization*, volume 2081 of *Lecture Notes in Computer Science*, pages 71–77. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-42225-9. doi: 10.1007/3-540-45535-3_6. URL http://dx.doi.org/10.1007/3-540-45535-3_6.

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.

A. Del Pia, R. Hildebrand, R. Weismantel, and K. Zemmer. Minimizing Cubic and Homogeneous Polynomials over Integers in the Plane. *ArXiv e-prints*, Aug. 2014.

A. Del Pia, S. S. Dey, and M. Molinaro. Mixed-integer quadratic programming is in np. Optimization Online: http://www.optimization-online.org/DB_HTML/2014/07/4446.html, 2015.

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.

I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *arXiv:1508.01982 [math.OC]*, 2015. URL http://arxiv.org/abs/1508.01982.

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.

©2016 IBM Corporation

References IV

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

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, Mar. 2014.

M. Grant, S. Boyd, and Y. Ye. *Chapter in Global Optimization: From Theory to Implementation*, chapter Disciplined Convex Programming, pages 115–210. Nonconvex Optimization and its Applications. Springer, 2006.

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.

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.

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

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.

F. Kılınç-Karzan and S. Yıldız. Two term disjunctions on the second-order cone. *Mathematical Programming*, April 2015. http://link.springer.com/article/10.1007/s10107-015-0903-4.

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.

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.

 ©2016 IBM Corporation

References V

M. Lubin, E. Yamangil, R. Bent, and J. P. Vielma. Extended Formulations in Mixed-integer Convex Programming. *ArXiv e-prints*, Nov. 2015.

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.

S. Modaresi, M. Kılınç, and J. Vielma. Intersection cuts for nonlinear integer programming: convexification techniques for structured sets. *Mathematical Programming*, pages 1–37, 2015. ISSN 0025-5610. doi: 10.1007/s10107-015-0866-5. URL http://dx.doi.org/10.1007/s10107-015-0866-5.

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. Íllles, 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.

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.

R. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.

M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

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.

J. P. Vielma, I. Dunning, J. Huchette, and M. Lubin. Extended Formulations in Mixed Integer Conic Quadratic Programming. *Research Report*, 2015.

S. Vigerske. *Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming*. PhD thesis, 2012.

S. Vigerske. Gams minlp benchmarks, 2013. URL http://bit.ly/1ftcys8.

S. Vigerske. Minlplib2. http://www.gamsworld.org/minlp/minlplib2/html/, 2015.

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.

 ©2016 IBM Corporation

IBM®