Cutting plane methods for Mixed Integer NonLinear Programming

Pierre Bonami and Andrea Lodi

LIF, CNRS/Aix-Marseille Université and LabOR, Universitá di Bologna

ISCO 2010 Spring School - Hammamet - March 2010

Basic Algorithmic ideas

Basic algorithmic ideas

Mixed Integer Non-Linear Programming

$$
\begin{aligned}
\min\ & f(x) \\
& s.t. \\
& g_j(x) \leq 0 \qquad\qquad \forall j = 1, \ldots, m \\
& x \in X \subseteq \mathbb{R}^n \\
& x_i \in \mathbb{Z} \qquad\qquad \forall i \in I \subseteq \{1, \ldots, n\}
\end{aligned}
\tag{1}
$$

- $f : \mathbb{R}^n \to \mathbb{R}$.
- $g_j : \mathbb{R}^n \to \mathbb{R}$.
- f and g_j sufficiently smooth ($\in C^1$ or better C^2).

Why should we study MINLP's?

Why should we study MINLP's?

Because it is not well solved and:

Why should we study MINLP's?

Because it is not well solved and:

- Rich source of applications for which solutions are needed.

Why should we study MINLP's?

Because it is not well solved and:

- Rich source of applications for which solutions are needed.
- Many problems which combine non-linear phenomenon and decision making ($0-1$ variables)
  - In particular problems in chemical engineering or which involve physical phenomenon

Why should we study MINLP's?

Because it is not well solved and:

- Rich source of applications for which solutions are needed.
- Many problems which combine non-linear phenomenon and decision making ($0-1$ variables)
  - In particular problems in chemical engineering or which involve physical phenomenon
- Some stochastic Integer Programs can be cast as deterministic MINLPs.

Why should we study MINLP's?

Because it is not well solved and:

- Rich source of applications for which solutions are needed.
- Many problems which combine non-linear phenomenon and decision making ($0-1$ variables)
  - In particular problems in chemical engineering or which involve physical phenomenon
- Some stochastic Integer Programs can be cast as deterministic MINLPs.
- More powerful modeling framework than MILP (and more natural for many engineers)

Why should we study MINLP's?

Because it is not well solved and:

- Rich source of applications for which solutions are needed.
- Many problems which combine non-linear phenomenon and decision making ($0-1$ variables)
  - In particular problems in chemical engineering or which involve physical phenomenon
- Some stochastic Integer Programs can be cast as deterministic MINLPs.
- More powerful modeling framework than MILP (and more natural for many engineers)
- We can rely on the progresses made in solving subproblems in the last decades.

Solving MINLP by fitting it into a "well solved" subproblem

From an MILP perspective

Solving MINLP by fitting it into a “well solved” subproblem

From an MILP perspective

- Remove/neglate non-linearities.

Solving MINLP by fitting it into a "well solved" subproblem

From an MILP perspective

- Remove/neglate non-linearities.
- Approximate non-linearities by linear functions.

Solving MINLP by fitting it into a "well solved" subproblem

From an MILP perspective

- Remove/neglate non-linearities.
- Approximate non-linearities by linear functions.
- Approximate unidimensional non-linearities by mixed-integer sets: SOS2[Beale1963, Forrest Tomlin 2007].

Solving MINLP by fitting it into a "well solved" subproblem

From an MILP perspective

- Remove/neglate non-linearities.
- Approximate non-linearities by linear functions.
- Approximate unidimensional non-linearities by mixed-integer sets: SOS2[Beale1963, Forrest Tomlin 2007].
- Most clever tricks handle arbitrary separable functions: linked-SOS [Beale Forrest 1976]

Solving MINLP by fitting it into a "well solved" subproblem

From an MILP perspective

- Remove/neglate non-linearities.
- Approximate non-linearities by linear functions.
- Approximate unidimensional non-linearities by mixed-integer sets: SOS2[Beale1963, Forrest Tomlin 2007].
- Most clever tricks handle arbitrary separable functions: linked-SOS [Beale Forrest 1976]

From an NLP perspective

Solving MINLP by fitting it into a "well solved" subproblem

From an MILP perspective

- Remove/neglate non-linearities.
- Approximate non-linearities by linear functions.
- Approximate unidimensional non-linearities by mixed-integer sets: SOS2[Beale1963, Forrest Tomlin 2007].
- Most clever tricks handle arbitrary separable functions: linked-SOS [Beale Forrest 1976]

From an NLP perspective

- Remove/neglate integrity.

Solving MINLP by fitting it into a "well solved" subproblem

From an MILP perspective

- Remove/neglate non-linearities.
- Approximate non-linearities by linear functions.
- Approximate unidimensional non-linearities by mixed-integer sets: SOS2 `[Beale1963, Forrest Tomlin 2007]`.
- Most clever tricks handle arbitrary separable functions: linked-SOS `[Beale Forrest 1976]`

From an NLP perspective

- Remove/neglate integrity.
- Round solutions to try to get "good" integer feasible solutions.

Two classes of MINLP

Mixed Integer Convex Program

If the functions $f(x)$ and $g_j(x)$ are convex.

- The continuous relaxation can be optimized to global optimality using NLP algorithms.
- Problem is NP-Hard.
- Important sub-class $f(x)$ and $g_j(x)$ are second-order cone representable.

Two classes of MINLP

Mixed Integer Convex Program

If the functions $f(x)$ and $g_j(x)$ are convex.

- The continuous relaxation can be optimized to global optimality using NLP algorithms.
- Problem is NP-Hard.
- Important sub-class $f(x)$ and $g_j(x)$ are second-order cone representable.

Mixed Integer NonLinear Program

- NLP techniques only give a local optima.
- Here we will always assume that integer variables are bounded:

Two classes of MINLP

Mixed Integer Convex Program

If the functions $f(x)$ and $g_j(x)$ are convex.

- The continuous relaxation can be optimized to global optimality using NLP algorithms.
- Problem is NP-Hard.
- Important sub-class $f(x)$ and $g_j(x)$ are second-order cone representable.

Mixed Integer NonLinear Program

- NLP techniques only give a local optima.
- Here we will always assume that integer variables are bounded:
- Problem is undecidable when integer variables are unbounded even if all functions are quadratic[Jeroslow 1973], or are polynomials and problem has 10 variables [De Lorea et. al. 2008].

Software for Mixed Integer Convex Programs

- α-ECP [Westerlund and Lundqvist 2005], academic, available in GAMS
- Bonmin [B. et. al. 2006], open source, available through COIN-OR, NEOS and in GAMS.
- Dicopt, commercial, available in GAMS.
- FilMINT [Abhishek, Leyffer and Linderoth 2010], available through NEOS.
- MINLP_BB [Leyffer 1998] recently upgraded [Leyffer 2010], academic.
- SBB [Bussieck and Drud], commercial, available in GAMS.

Second order cone constrained

- CPLEX, MOSEK, XPRESS,...

Software for Mixed Integer NonLinear Program

- Baron [Sahinidis, Tawarmalani 2003], commercial, stand-alone and in GAMS.
- Couenne [Belotti et. al. 2009], open source, available through COIN-OR.
- SCIP [Berhold et. al. 2010], academic, right now only quadratic functions, general MINLP upcoming.
- LindoGlobal, commercial, available in GAMS.

Basic algorithmic ideas

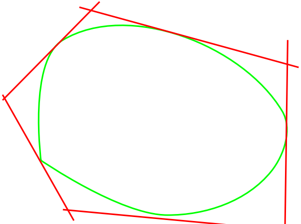
Problems can be solved to optimality by branch-and-bound using appropriate *convex relaxations*.

Convex relaxations

- for MICP: NLP relaxation obtained by dropping integrality requirements.
- for MICP: Outer approximation linear approximation of the problem.
- for MINLP: Sherali Adams relaxation obtained by convexifying each non-convex term separately.

Use the relaxations to setup a branch-and-cut algorithm.

Outer Approximation [Duran, Grossmann 1986]



(assume linear objective)

$$\min f(x)$$
$$\text{s.t.}$$
$$g(x) \leq 0,$$
$$x_i \in \mathbb{Z}\ \forall i \in \mathcal{I}.$$

Idea: linearize constraints at different points and build an equivalent MILP:

$\mathcal{T}$ contains suitably chosen linearization points.

$$(OA) \begin{cases} \min f(x) \\ J_g(x^k)^T \left(x - x^k\right) + g(x^k) \leq 0 \\ \qquad\qquad \forall (x^k) \in \mathcal{T} \\ x \in X,\ x_i \in \mathbb{Z}\ \forall i \in \mathcal{I}. \end{cases}$$

Separable MINLP

- f and g_i can be constructed as finite recursive composition of univariate and bivariate functions.
- Usually restricted to $\phi_1(x) = \ln(x)$, $\phi_2(x) = e^x$, $\phi_3(x, y) = x + y$, $\phi_4(x, y) = xy$, $\phi_5(x, \alpha) = x^\alpha$.
- Example:

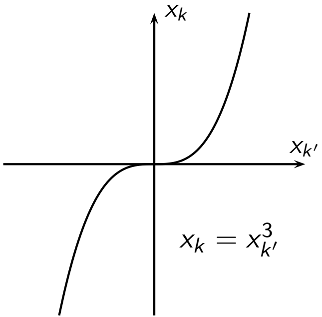
$$f(x) = \sqrt{x_1 x_2} + \ln(x_2)$$

is equivalent to:

$$\begin{aligned} f(x) &= x_3 + x_4 = \phi_3(x_3, x_4) \\ x_3 &= \sqrt{x_5} = \phi_5(x_5, \frac{1}{2}) \\ x_4 &= \ln(x_2) = \phi_1(x_2) \\ x_5 &= x_1 x_2 = \phi_4(x_1, x_2) \end{aligned}$$

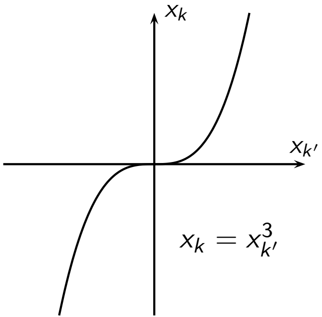
Building convex relaxation for separable MINLP

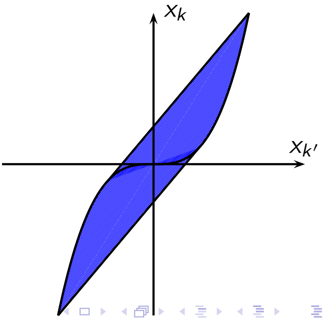
- Adding appropriate number of variables, decompose each separable function into atomic elements of the form: $x_{n+k} = \phi_j(x_{k'})$ or $x_{n+k} = \phi_j(x_{k'}, x_{k''})$
- Using bounds on original variables, infer bounds on new variables: $l_i \leq x_{n+k} \leq u_i$.
- For each x_{n+k} build an approximation of $\mathrm{conv}(\{x_{n+k} = \psi_i(x_{k'}), \, l_{k'} \leq x_{k'} \leq u_{k'}\})$



Building convex relaxation for separable MINLP

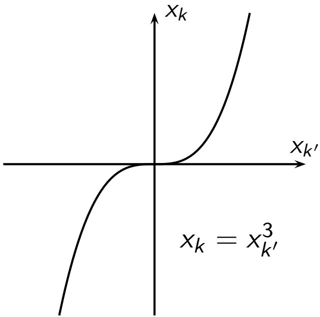
- Adding appropriate number of variables, decompose each separable function into atomic elements of the form: $x_{n+k} = \phi_j(x_{k'})$ or $x_{n+k} = \phi_j(x_{k'}, x_{k''})$
- Using bounds on original variables, infer bounds on new variables: $l_i \leq x_{n+k} \leq u_i$.
- For each x_{n+k} build an approximation of $\text{conv}(\{x_{n+k} = \psi_i(x_{k'}), \, l_{k'} \leq x_{k'} \leq u_{k'}\})$

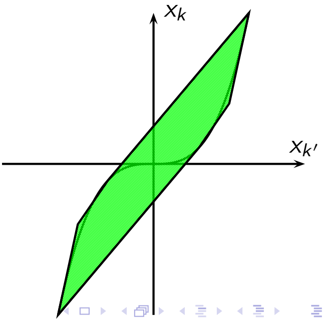




Building convex relaxation for separable MINLP

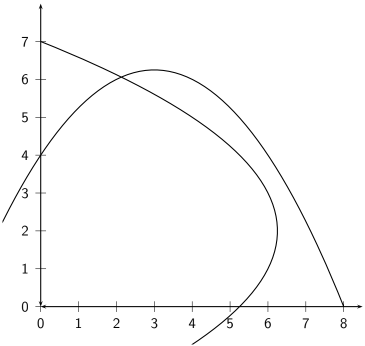
- Adding appropriate number of variables, decompose each separable function into atomic elements of the form: $x_{n+k} = \phi_j(x_{k'})$ or $x_{n+k} = \phi_j(x_{k'}, x_{k''})$
- Using bounds on original variables, infer bounds on new variables: $l_i \leq x_{n+k} \leq u_i$.
- For each x_{n+k} build an approximation of $\text{conv}(\{x_{n+k} = \psi_i(x_{k'}),\ l_{k'} \leq x_{k'} \leq u_{k'}\})$





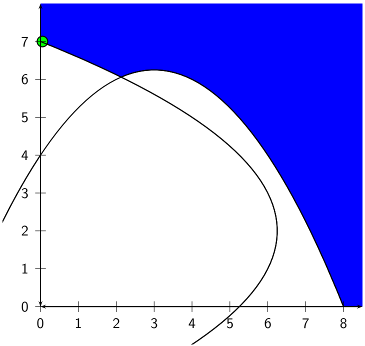
Spatial branch-and-bound (sketch)

- Consider the problem of finding $(x_1, x_2) \geq 0$ above the two parabolas which minimizes $x_1 + x_2$

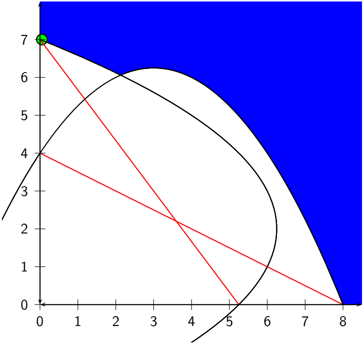


Spatial branch-and-bound (sketch)

- Consider the problem of finding $(x_1, x_2) \geq 0$ above the two parabolas which minimizes $x_1 + x_2$
- Convex relaxation for each of the constraints can be built by taking secant approximation

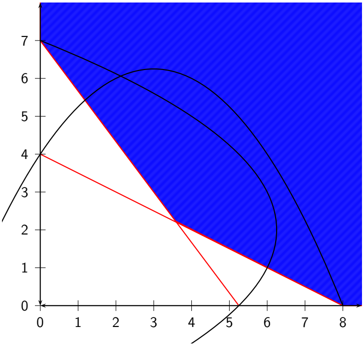


Spatial branch-and-bound (sketch)



- Consider the problem of finding $(x_1, x_2) \geq 0$ above the two parabolas which minimizes $x_1 + x_2$
- Convex relaxation for each of the constraints can be built by taking secant approximation

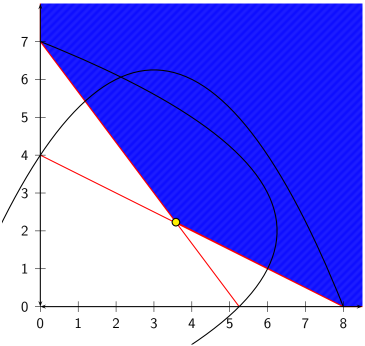
Spatial branch-and-bound (sketch)



- Consider the problem of finding $(x_1, x_2) \geq 0$ above the two parabolas which minimizes $x_1 + x_2$
- Convex relaxation for each of the constraints can be built by taking secant approximation

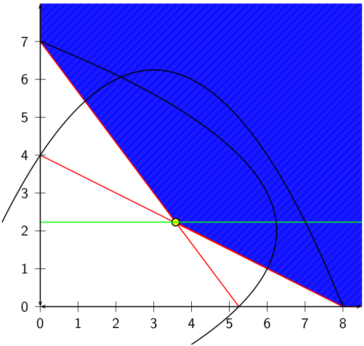
Spatial branch-and-bound (sketch)

- Solve convex relaxation.



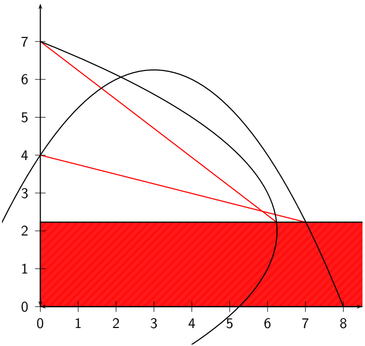
Spatial branch-and-bound (sketch)

- Solve convex relaxation.
- Solution is not feasible, branch on $x_2 \geq x_2^* \vee x_2 \leq x_2^*$

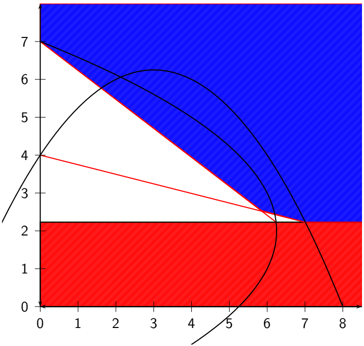


Spatial branch-and-bound (sketch)

- Solve convex relaxation.
- Solution is not feasible, branch on $x_2 \geq x_2^* \vee x_2 \leq x_2^*$
- First child add constraint $x_2 \geq x_2^*$ and update convex relaxation.

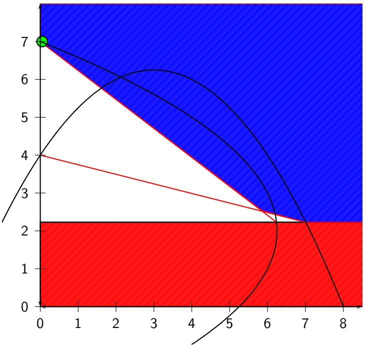


Spatial branch-and-bound (sketch)



- Solve convex relaxation.
- Solution is not feasible, branch on $x_2 \geq x_2^* \vee x_2 \leq x_2^*$
- First child add constraint $x_2 \geq x_2^*$ and update convex relaxation.

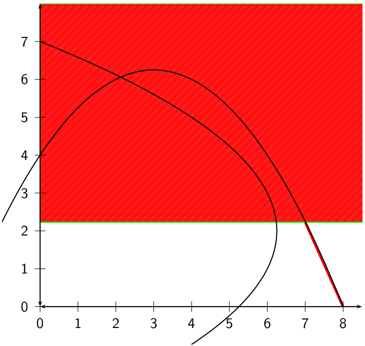
Spatial branch-and-bound (sketch)



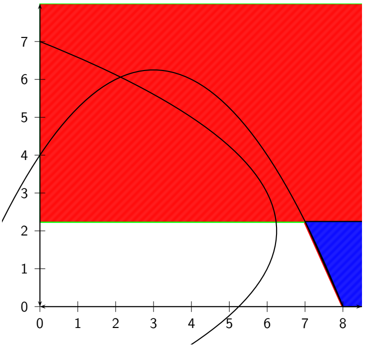
- Solve convex relaxation.
- Solution is not feasible, branch on $x_2 \geq x_2^* \vee x_2 \leq x_2^*$
- First child add constraint $x_2 \geq x_2^*$ and update convex relaxation.

Spatial branch-and-bound (sketch)

- Solve convex relaxation.
- Solution is not feasible, branch on $x_2 \geq x_2^* \vee x_2 \leq x_2^*$
- First child add constraint $x_2 \geq x_2^*$ and update convex relaxation.
- Second child add $x_2 \leq x_2^*$ and update.



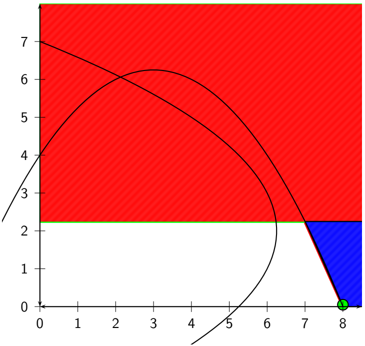
Spatial branch-and-bound (sketch)



- Solve convex relaxation.
- Solution is not feasible, branch on $x_2 \geq x_2^* \vee x_2 \leq x_2^*$
- First child add constraint $x_2 \geq x_2^*$ and update convex relaxation.
- Second child add $x_2 \leq x_2^*$ and update.

Spatial branch-and-bound (sketch)

- Solve convex relaxation.
- Solution is not feasible, branch on $x_2 \geq x_2^* \vee x_2 \leq x_2^*$
- First child add constraint $x_2 \geq x_2^*$ and update convex relaxation.
- Second child add $x_2 \leq x_2^*$ and update.



Part II

Disjunctive cuts for non-convex MINLPs

QCP

$$\min\ a_0^T x$$
$$s.t.$$
$$x^T A_k x + a_k^T x + b_k \leq 0 \quad k = 1 \ldots m$$
$$l \leq x \leq u$$

- A_k are symmetric but usualy not positive semi-definite.
- All variables appearing in quadratic expressions have finite bounds.

Reformulation of QCP

Introduce variables $y_{ij} = x_i x_j$:

$$\min \; a_0^T x$$

$$s.t.$$

$$\langle A_k, Y \rangle + a_k^T x + b_k \leq 0 \quad k = 1 \ldots m$$

$$Y = xx^T$$

$$l \leq x \leq u$$

- The only non-linearity of the problem is in the constraint $Y = xx^T$.

Reformulation of QCP

Introduce variables $y_{ij} = x_i x_j$:

$$\begin{aligned} \min\ & a_0^T x \\ s.t.\ & \\ & \langle A_k, Y \rangle + a_k^T x + b_k \leq 0 \quad k = 1 \ldots m \\ & Y = xx^T \\ & l \leq x \leq u \end{aligned}$$

- The only non-linearity of the problem is in the constraint $Y = xx^T$.
  - $Y - xx^T \succcurlyeq 0$ is convex (can be handled by conic optimization),

Reformulation of QCP

Introduce variables $y_{ij} = x_i x_j$:

$$\min\ a_0^T x$$
$$s.t.$$
$$\langle A_k, Y \rangle + a_k^T x + b_k \leq 0 \quad k = 1 \ldots m$$
$$Y = xx^T$$
$$l \leq x \leq u$$

- The only non-linearity of the problem is in the constraint $Y = xx^T$.
  - $Y - xx^T \succcurlyeq 0$ is convex (can be handled by conic optimization),
  - $xx^T - Y \succcurlyeq 0$ is non-convex.

RLT relaxation of QCP

Replace $y_{ij} = x_i x_j$ by its convex envelope given by RLT ineqalities

$$\begin{aligned}
&\min\ a_0^T x \\
&\ s.t. \\
&\qquad \langle A_k, Y \rangle + a_k^T x + b_k \le 0 \quad k = 1 \ldots m \\
&\qquad l \le x \le u
\end{aligned}$$

$$\max \left\{ \begin{array}{c} l_j x_i + l_i x_j - l_j l_i \\ u_j x_i + u_i x_j - u_j u_i \end{array} \right\} \le y_{ij} \le \min \left\{ \begin{array}{c} l_i x_j + u_j x_i - l_i u_j \\ l_j x_i + u_i x_j - l_j u_i \end{array} \right\}$$

RLT relaxation of QCP

Replace $y_{ij} = x_i x_j$ by its convex envelope given by RLT ineqalities

$$\min\ a_0^T x$$
$$s.t.$$
$$\langle A_k, Y \rangle + a_k^T x + b_k \leq 0 \quad k = 1 \dots m$$
$$l \leq x \leq u$$
$$\max \left\{ \begin{array}{c} l_j x_i + l_i x_j - l_j l_i \\ u_j x_i + u_i x_j - u_j u_i \end{array} \right\} \leq y_{ij} \leq \min \left\{ \begin{array}{c} l_i x_j + u_j x_i - l_i u_j \\ l_j x_i + u_i x_j - l_j u_i \end{array} \right\}$$

Our reference relaxation

RLT relaxation of QCP

Replace $y_{ij} = x_i x_j$ by its convex envelope given by RLT ineqalities

$$\begin{aligned}
&\min\ a_0^T x \\
&\ s.t. \\
&\qquad \langle A_k, Y\rangle + a_k^T x + b_k \le 0 \quad k = 1 \ldots m \\
&\qquad l \le x \le u \\
&\max\left\{\begin{array}{c} l_j x_i + l_i x_j - l_j l_i \\ u_j x_i + u_i x_j - u_j u_i \end{array}\right\} \le y_{ij} \le \min\left\{\begin{array}{c} l_i x_j + u_j x_i - l_i u_j \\ l_j x_i + u_i x_j - l_j u_i \end{array}\right\}
\end{aligned}$$

We know

Can be strengthened by the LMI inequality $Y - xx^T \succcurlyeq 0$

RLT relaxation of QCP

Replace $y_{ij} = x_i x_j$ by its convex envelope given by RLT ineqalities

$$\begin{aligned} &\min\ a_0^T x \\ &\ s.t. \\ &\qquad \langle A_k, Y \rangle + a_k^T x + b_k \le 0 \quad k = 1 \ldots m \\ &\qquad l \le x \le u \\ &\max \left\{ \begin{array}{c} l_j x_i + l_i x_j - l_j l_i \\ u_j x_i + u_i x_j - u_j u_i \end{array} \right\} \le y_{ij} \le \min \left\{ \begin{array}{c} l_i x_j + u_j x_i - l_i u_j \\ l_j x_i + u_i x_j - l_j u_i \end{array} \right\} \end{aligned}$$

We know

Can be strengthened by the LMI inequality $Y - xx^T \succcurlyeq 0$

Research Question?

Can we use the non-convex constraint $xx^T - Y \succcurlyeq 0$ to produce cuts for this relaxation?

Disjunctive Programming

Polyhedral Relaxation

$$P = \{x : Ax \geq b\}$$

Disjunction

$$\bigvee_{k=1}^{q} \left[D^k x \geq d^k \right]$$

$$P_D = clconv \left[\bigcup_{k=1}^{q} \left(P \cap \left\{ x : D^k x \geq d^k \right\} \right) \right]$$

Separation Problem

Given $\hat{x} \in P$ show that $\hat{x} \in P_D$ or find an inequality separating $\hat{x}$ from P_D.

Disjunctive Programming

Theorem

$\hat{x} \in P_D$ if and only if the optimal value of the following Cut-Generation Linear Program (CGLP) is non-negative.

$$
\begin{aligned}
\min\ & \alpha \hat{x} - \beta \\
s.t.\ & \\
& \alpha = u^k A + v^k D^k\ , \quad k = 1 \ldots q\ ; \\
& \beta \leq u^k b + v^k d^k\ , \quad k = 1 \ldots q\ ; \\
& u^k, v^k \geq 0\ , \quad k = 1 \ldots q\ ;
\end{aligned}
\tag{CGLP}
$$

Applying Disjunctive Programming to QCP?

Polyhedral Relaxation

$$P = \{x : Ax \geq b\}$$

Disjunction

$$\bigvee_{k=1}^{q} \left[D^k x \geq d^k \right]$$

What are the sources of non-convexity in QCP?

Applying Disjunctive Programming to QCP?

Polyhedral Relaxation

$$P = \{x : Ax \geq b\}$$

Disjunction

$$\bigvee_{k=1}^{q} \left[D^k x \geq d^k \right]$$

Integrality Constraints

- $x_j \in \mathbb{Z}$, $j \in N_{\mathcal{I}}$
- Elementary disjunctions

$$(x_j \leq k) \vee (x_j \geq k+1)$$

- GUB disjunctions
- Split disjunctions

Applying Disjunctive Programming to QCP?

Polyhedral Relaxation

$$P = \{x : Ax \geq b\}$$

Disjunction

$$\bigvee_{k=1}^{q} \left[D^k x \geq d^k \right]$$

Integrality Constraints

- $x_j \in \mathbb{Z}$, $j \in N_{\mathcal{I}}$
- Elementary disjunctions

$$(x_j \leq k) \vee (x_j \geq k+1)$$

- GUB disjunctions
- Split disjunctions

$$xx^T - Y \succcurlyeq 0$$

?

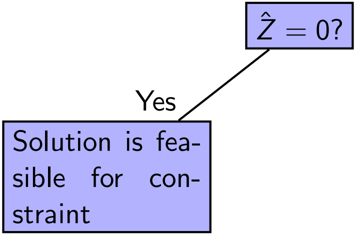
Cutting plane procedure

- Solve

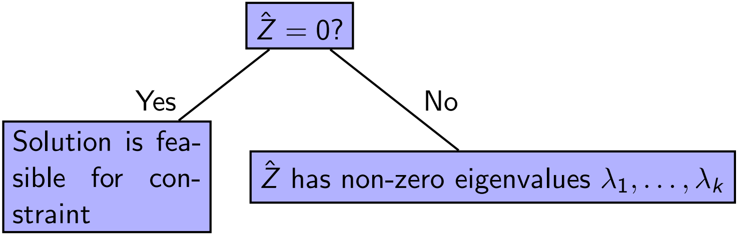
$$
\begin{aligned}
&\min\ a_0^T x \\
&\ s.t. \\
&\qquad \langle A_k, Y\rangle + a_k^T x + b_k \le 0 \quad l = 1 \dots m \\
&\qquad l \le x \le u \\
&\max \left\{ \begin{array}{c} l_j x_i + l_i x_j - l_j l_i \\ u_j x_i + u_i x_j - u_j u_i \end{array} \right\} \le y_{ij} \le \min \left\{ \begin{array}{c} l_i x_j + u_j x_i - l_i u_j \\ l_j x_i + u_i x_j - l_j u_i \end{array} \right\}
\end{aligned}
$$

- Let $(\hat{x}, \hat{Y})$ be the solution and $\hat{Z} = \hat{Y} - \hat{x}\hat{x}^T$.
- If $\hat{Z} = 0$ the solution satisfies $Y - xx^T = 0$.
- Otherwise can we find inequalities that cut off $(\hat{x}, \hat{Y})$?

Eigenvalues of $\hat{Z} = \hat{Y} - \hat{x}\hat{x}^T$



Eigenvalues of $\hat{Z} = \hat{Y} - \hat{x}\hat{x}^T$



Eigenvalues of $\hat{Z} = \hat{Y} - \hat{x}\hat{x}^T$

Let $c_1, \ldots, c_k$ be the corresponding eigenvectors

$\hat{Z} = 0$?

Yes

Solution is feasible for constraint

No

$\hat{Z}$ has non-zero eigenvalues $\lambda_1, \ldots, \lambda_k$

$\lambda_i < 0$

$c_i^T Y c_i - (c_i^T x)^2 \geq 0$ is a convex quadratic inequality which cuts off $(\hat{x}, \hat{Y})$

Eigenvalues of $\hat{Z} = \hat{Y} - \hat{x}\hat{x}^T$

Let $c_1, \ldots, c_k$ be the corresponding eigenvectors

$\hat{Z} = 0$?

Yes

Solution is feasible for constraint

No

$\hat{Z}$ has non-zero eigenvalues $\lambda_1, \ldots, \lambda_k$

$\lambda_i < 0$

$c_i^T Y c_i - (c_i^T x)^2 \geq 0$ is a convex quadratic inequality which cuts off $(\hat{x}, \hat{Y})$

$\lambda_i > 0$

$c_i^T Y c_i - (c_i^T x)^2 \leq 0$ is a non-convex cut.

Eigenvalues of $\hat{Z} = \hat{Y} - \hat{x}\hat{x}^T$

Let $c_1, \dots, c_k$ be the corresponding eigenvectors

$\hat{Z} = 0$?

Yes

Solution is feasible for constraint

No

$\hat{Z}$ has non-zero eigenvalues $\lambda_1, \dots, \lambda_k$

$\lambda_i < 0$

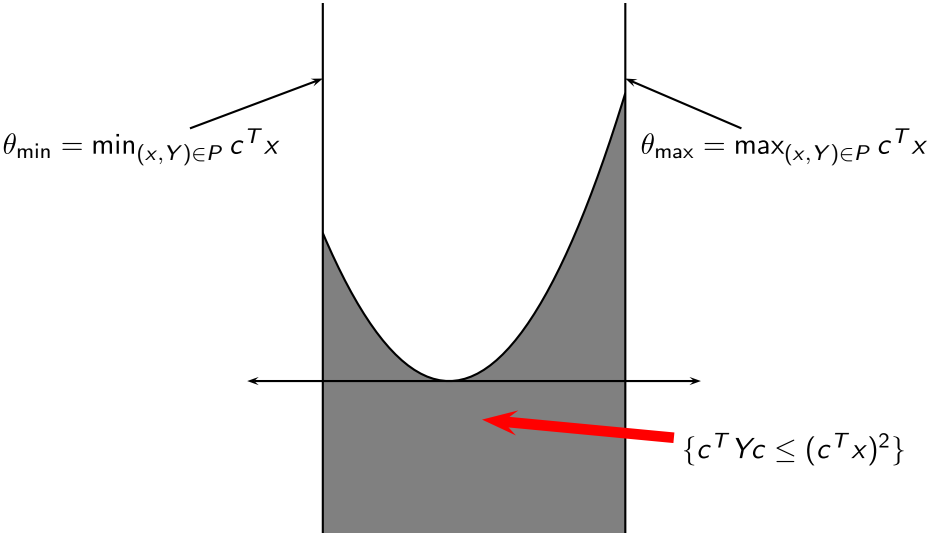
$c_i^T Y c_i - (c_i^T x)^2 \geq 0$ is a convex quadratic inequality which cuts off $(\hat{x}, \hat{Y})$

$\lambda_i > 0$

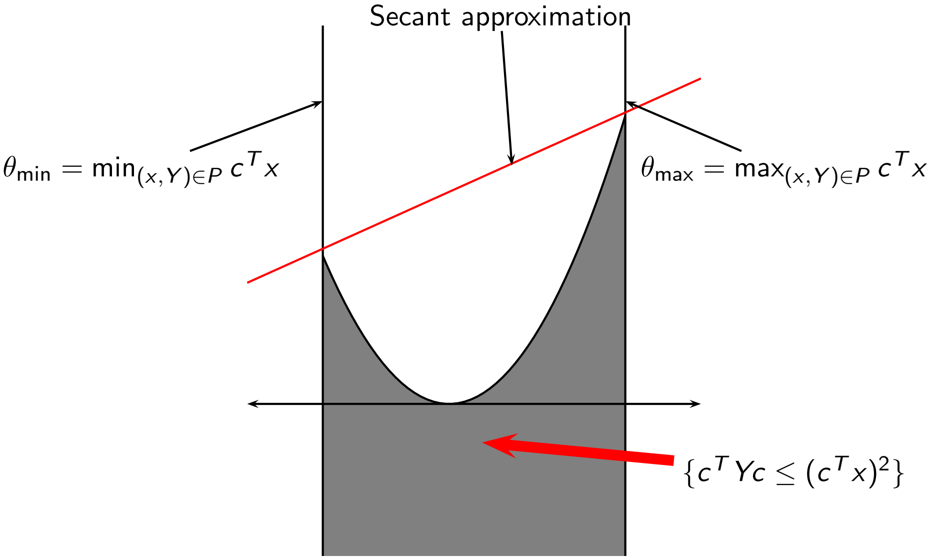
$c_i^T Y c_i - (c_i^T x)^2 \leq 0$ is a non-convex cut.

Imposes SDP condition $Y - xx^T \succcurlyeq 0$ by quadratic cuts

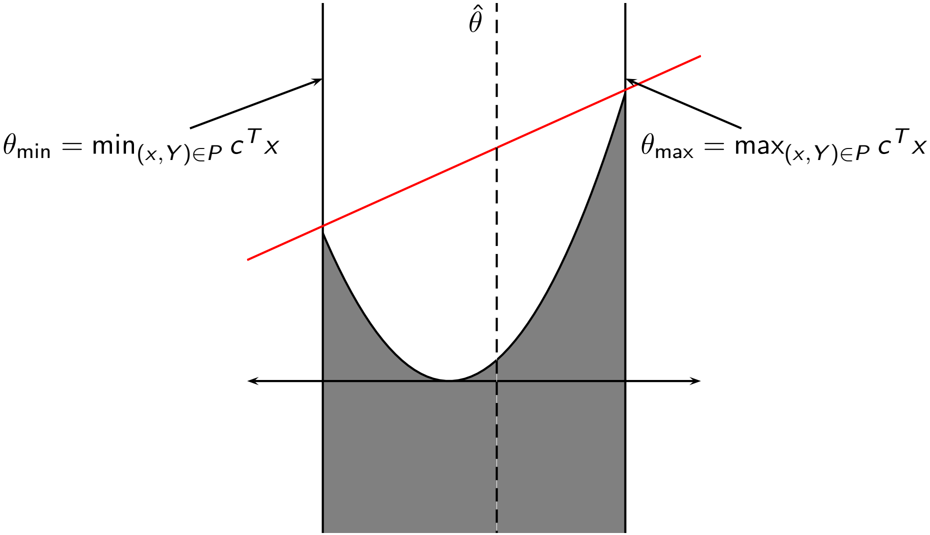
Cuts from > 0 eigenvalues?



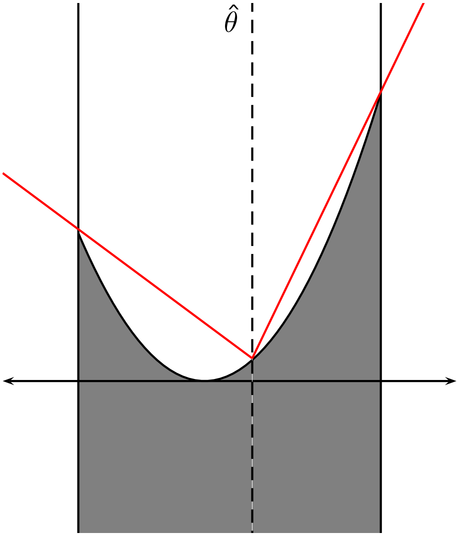
Cuts from > 0 eigenvalues?



Cuts from > 0 eigenvalues?

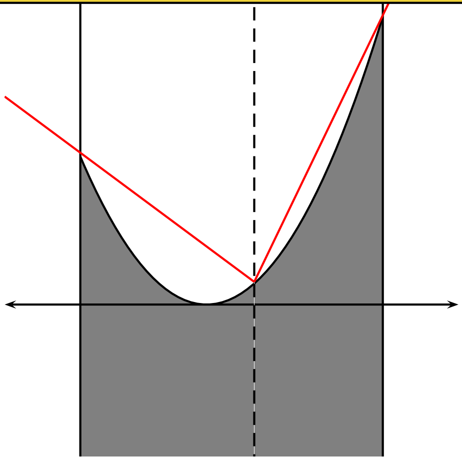


Cuts from > 0 eigenvalues?

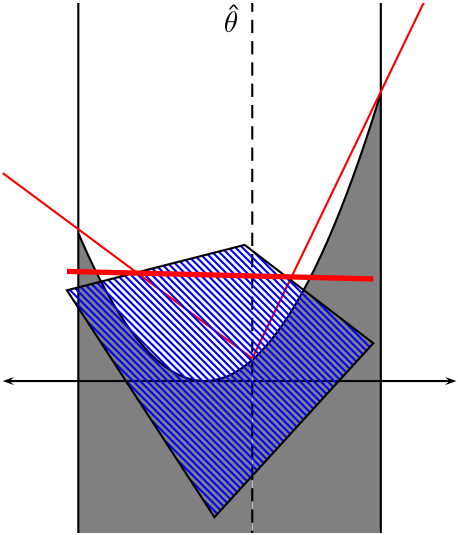


Cuts from > 0 eigenvalues?

$$\begin{pmatrix} c^T x \le \theta \\ c^T Y c \le p_1(c^T x) + q_1 \end{pmatrix} \vee \begin{pmatrix} c^T x \ge \theta \\ c^T Y c \le p_2(c^T x) + q_2 \end{pmatrix}$$



Cuts from > 0 eigenvalues?



Applying Disjunctive Programming to QCP

Polyhedral Relaxation

$$P = \{x : Ax \geq b\}$$

Disjunction

$$\bigvee_{k=1}^{q} \left[D^k x \geq d^k \right]$$

Linear Outer Approximation of QCP defined by incumbent solution.

Applying Disjunctive Programming to QCP

Polyhedral Relaxation

$$P = \{x : Ax \geq b\}$$

Disjunction

$$\bigvee_{k=1}^{q} \left[D^k x \geq d^k \right]$$

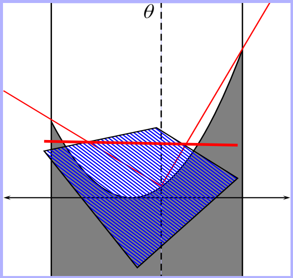
Integrality Constraints

- $x_j \in \mathbb{Z}$, $j \in N_I$
- Elementary disjunctions

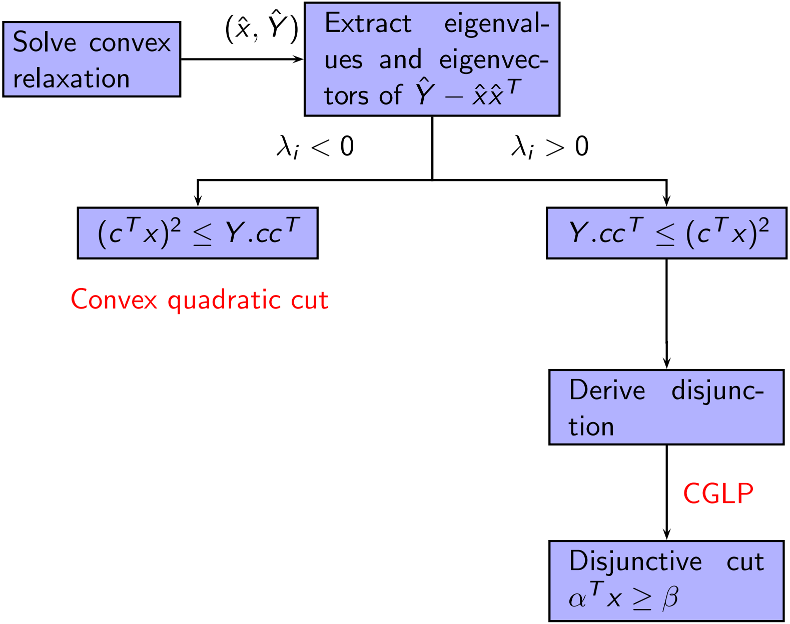
$$(x_j \leq 0) \vee (x_j \geq 1)$$

- Split disjunctions
- GUB disjunctions

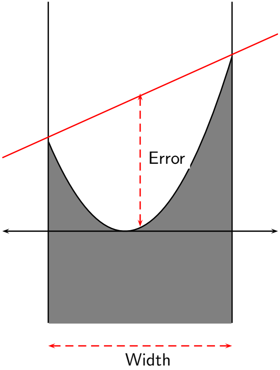
$$xx^T - Y \succcurlyeq 0$$



Cutting Plane Algorithm (I)



Error in secant approximation



Proposition

Let $f(t) = t^2$, and let $g(t) = t(a+b) + ab$ represent the secant approximation of $f(t)$ in the interval $[a, b]$; then $max_{t\in[a,b]}(f(t) - g(t)) = \frac{(a-b)^2}{4}$.

Secant approximation is better if the width of $[a, b]$ is smaller

Finding disjunctions in directions of small width

The width of outer approximation in direction c

$$\eta(c) = \max_{(x,Y)\in OA} c^T x - \min_{(x,Y)\in OA} c^T x$$

We want c such that:

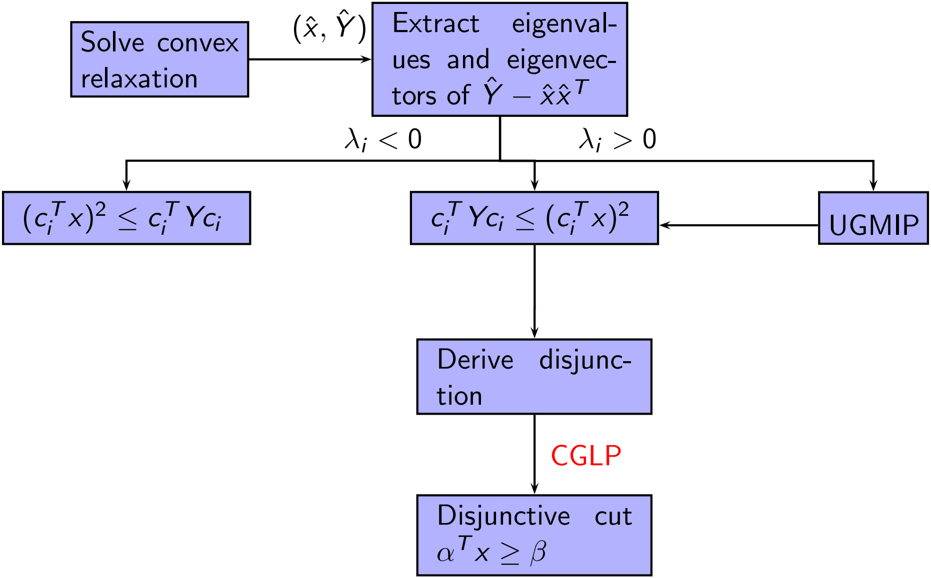
- $\eta(c)$ is as small as possible
- $c^T \hat{Y} c > (c^T \hat{x})^2$
  - Take c to be in the vector space spanned by the eigenvectors of $\hat{Y} - \hat{x}\hat{x}^T$ with positive eigenvalue.

UGMIP

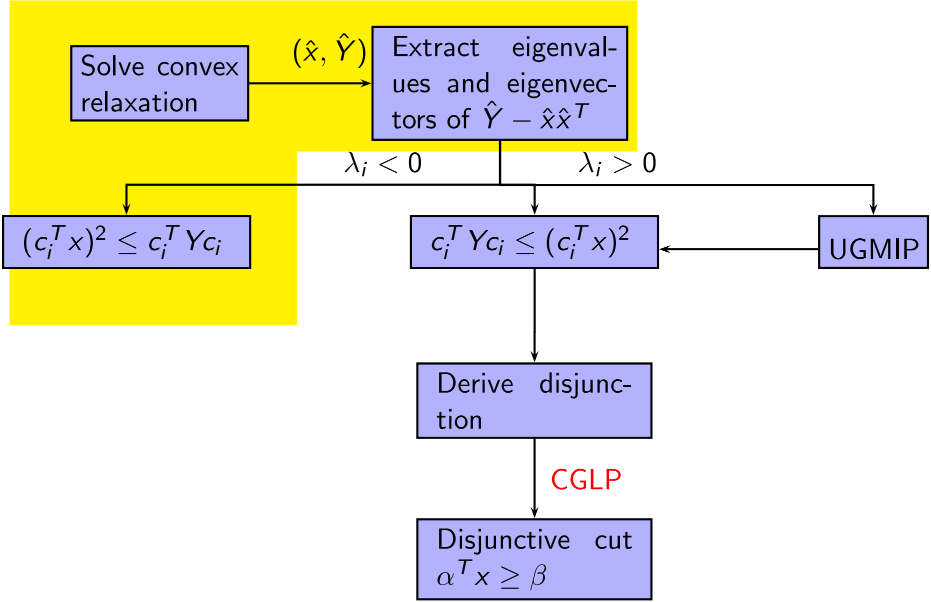
Problem can be formulated as an MILP

- Add a penalty term to favor solutions with large curvature.
- A diversification scheme is used to obtain several disjunctions.

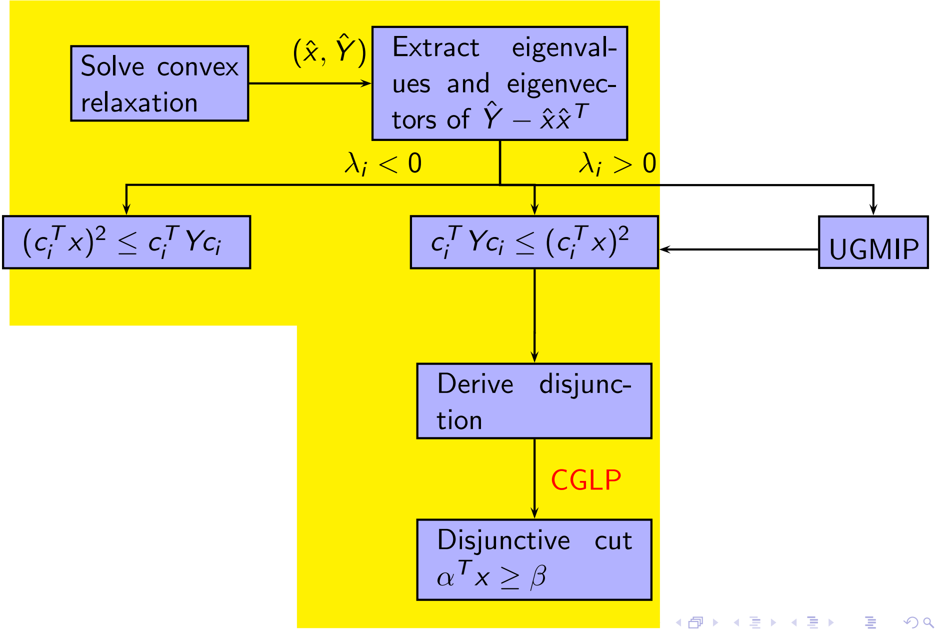
Cutting Plane Algorithm (II)



Cutting Plane Algorithm (II) Version 1

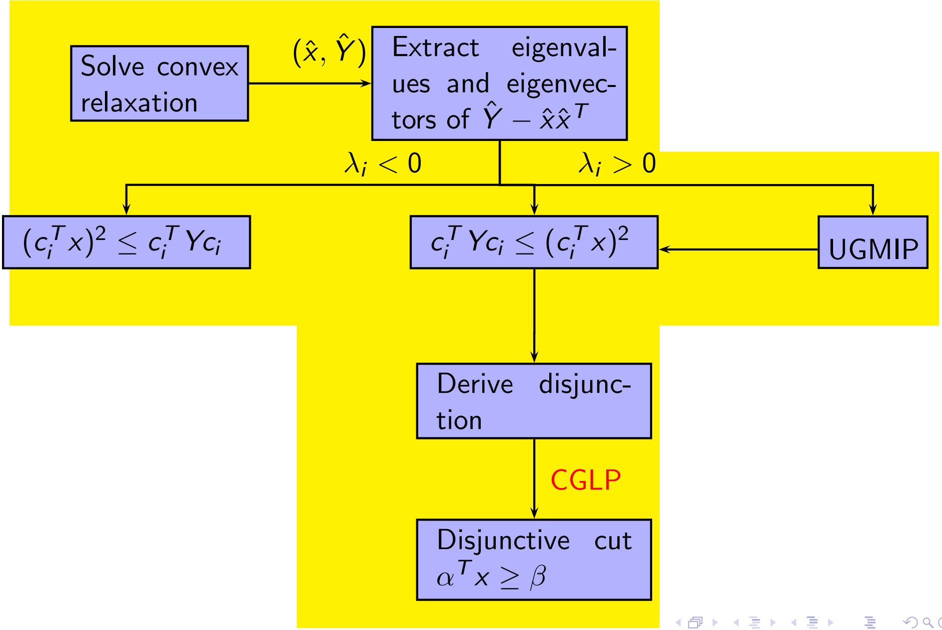


Cutting Plane Algorithm (II) Version 2



Cutting Plane Algorithm (II)

Version 3



Experimental testing

Solvers

- Convex relaxations IPOPT
- Eigenvalues computations LAPACK
- LP & MILP CPLEX10.1
- BONMIN based implementation.

Test bed

- 129 GlobalLIB instances ($\leq$ 50 vars) reformulated as QCP.
  - $x_1 x_2 x_3 x_4 x_5$
  - $(x_1 + x_2)/x_3 \geq 2x_1$

Experimental setup

- 1 hour time limit.
- report % gap closed: $\frac{\text{opt(final relaxation) - opt(initial)}}{\text{opt(QCP) - opt(initial)}} \times 100$

Gap closed for the 129 globallib instances with non-zero duality gap

	V1	V2	V3
$>$99.99 % gap closed	16	23	23
98-99.99 % gap closed	1	44	52
75-98 % gap closed	10	23	21
25-75 % gap closed	11	22	20
0-25 % gap closed	91	17	13
Average Gap Closed	24.80%	76.49%	80.86%

Gap closed for the 129 globallib instances with non-zero duality gap

	V1	V2	V3
>99.99 % gap closed	16	23	23
98-99.99 % gap closed	1	44	52
75-98 % gap closed	10	23	21
25-75 % gap closed	11	22	20
0-25 % gap closed	91	17	13
Average Gap Closed	24.80%	76.49%	80.86%

- V3 closes more than 98% gap on 75 instances.

Versions 2 and 3 vs. Version 1

	gap closed		
Instance	V1	V2	V3
ex2_1_1	0.00%	72.62%	99.92%
ex2_1_5	0.00%	99.98%	99.99%
ex2_1_6	0.00%	99.95%	99.97%
ex2_1_9	0.00%	98.79%	99.73%
ex3_1_3	0.00%	99.99%	99.99%
ex3_1_4	0.00%	86.31%	99.57%
ex5_2_4	0.00%	79.31%	99.92%
prob05	0.00%	99.78%	99.49%
st_bpv2	0.00%	99.99%	99.99%
st_bsj2	0.00%	99.98%	99.96%
st_bsj4	0.00%	99.86%	99.80%
st_e02	0.00%	99.88%	99.95%
st_e07	0.00%	99.97%	99.97%
st_e08	0.00%	99.81%	99.89%
st_e24	0.00%	99.81%	99.81%
st_e26	0.00%	99.96%	99.96%
st_e33	0.00%	99.94%	99.95%
st_fp1	0.00%	72.62%	99.92%
st_fp5	0.00%	99.98%	99.99%
st_fp6	0.00%	99.92%	99.97%
st_glmp_kk92	0.00%	99.98%	99.98%

	gap closed		
Instance	V1	V2	V3
st_glmp_kky	0.00%	99.80%	99.71%
st_ht	0.00%	99.81%	99.89%
st_jcbpaf2	0.00%	99.47%	99.61%
st_kr	0.00%	99.93%	99.95%
st_m1	0.00%	99.96%	99.96%
st_pan1	0.00%	99.72%	99.92%
st_pan2	0.00%	68.54%	99.91%
st_ph1	0.00%	99.98%	99.98%
st_ph11	0.00%	99.46%	98.19%
st_ph12	0.00%	99.49%	99.62%
st_ph13	0.00%	99.38%	98.80%
st_ph14	0.00%	99.85%	99.86%
st_ph15	0.00%	99.83%	99.81%
st_ph2	0.00%	99.98%	99.98%
st_ph20	0.00%	99.98%	99.98%
st_ph3	0.00%	99.98%	99.98%
st_phex	0.00%	99.96%	99.96%
st_qpc-m0	0.00%	99.96%	99.96%
st_qpc-m1	0.00%	99.99%	99.98%
st_qpc-m3a	0.00%	98.10%	99.16%
st_qpc-m3b	0.00%	100.00%	100.00%
st_qpk1	0.00%	99.98%	99.98%
st_rv1	0.00%	96.19%	98.44%
st_z	0.00%	99.96%	99.95%

Version 2 vs. Version 3

	gap closed		
Instance	V1	V2	V3
ex7_3_1	0.00%	0.00%	85.43%
ex9_2_3	0.00%	0.00%	47.17%
ex9_2_7	42.31%	51.47%	86.25%
st_fp7b	0.00%	22.06%	55.51%
st_rv3	0.00%	40.40%	72.68%
ex9_2_1	54.54%	60.04%	92.02%
st_pan2	0.00%	68.54%	99.91%
ex2_1_1	0.00%	72.62%	99.92%
st_fp1	0.00%	72.62%	99.92%
ex5_2_4	0.00%	79.31%	99.92%
ex8_1_7	77.43%	77.43%	95.79%
st_rv7	0.00%	45.43%	62.28%
st_qpk3	0.00%	33.53%	50.04%
st_rv8	0.00%	29.90%	45.80%
st_e20	0.00%	76.38%	90.88%
ex8_1_8	0.00%	76.49%	90.88%
ex5_3_2	0.00%	7.27%	21.00%
ex3_1_4	0.00%	86.31%	99.57%
st_fp7c	0.00%	44.26%	57.10%
st_qpk2	0.00%	71.34%	83.33%
st_rv9	0.00%	20.56%	31.64%
house	0.00%	86.93%	97.92%
ex7_3_2	0.00%	59.51%	70.26%

- On 23 instances it closes at least 10 % more gap.

Linear Complementarity Constraints

- Some GlobalLib problems have linear complementarity constraints

$$x_i x_j = 0$$

- These constraints can be reformulated as disjunctions

$$(x_i = 0) \vee (x_j = 0)$$

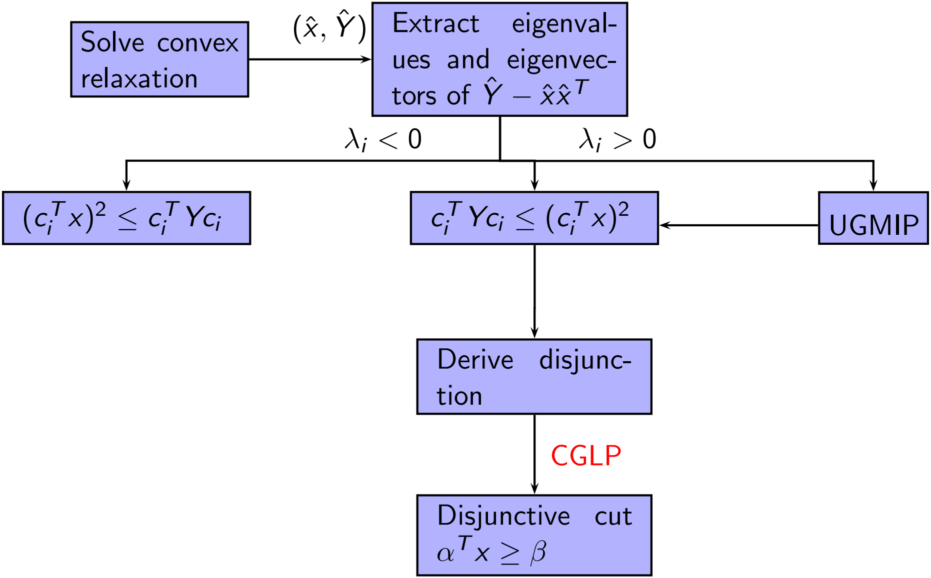
which can be added to our medley of disjunctions for deriving cuts.

Linear Complementarity Constraints

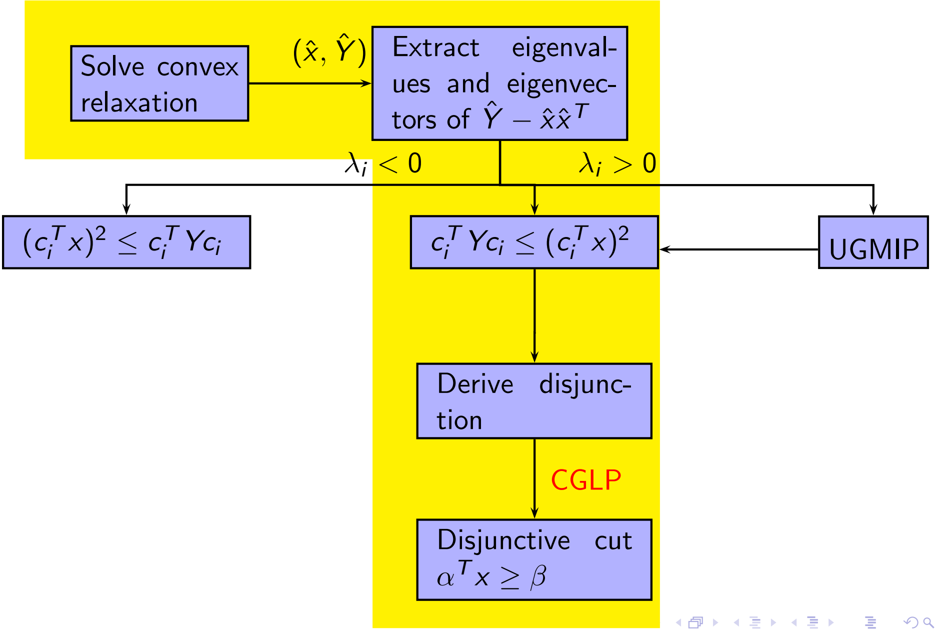
	Without Using LCD		Using LCD	
Instance	V2	V3	V2	V3
ex9_1_4	0.00%	1.55%	100.00%	99.97%
ex9_2_1	60.04%	92.02%	99.95%	99.95%
ex9_2_2	88.29%	98.06%	100.00%	100.00%
ex9_2_3	0.00%	47.17%	99.99%	99.99%
ex9_2_4	99.87%	99.89%	99.99%	100.00%
ex9_2_6	87.93%	62.00%	80.22%	92.09%
ex9_2_7	51.47%	86.25%	99.97%	99.95%

Linear complementarity constraints can be exploited effectively in a disjunctive framework.

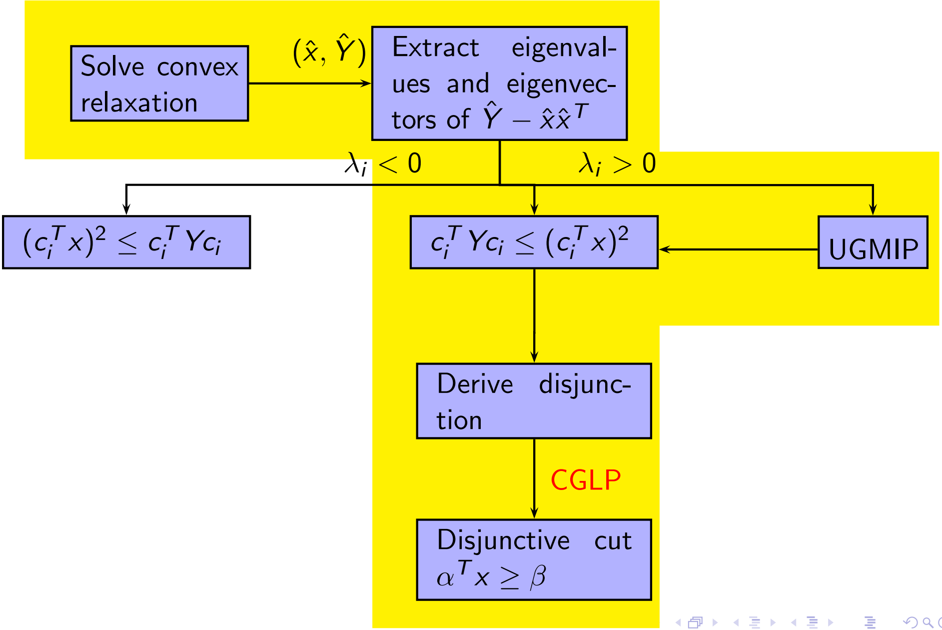
Marginal value of Quadratic Cuts



Marginal value of Quadratic Cuts V2-Dsj



Marginal value of Quadratic Cuts V3-Dsj



Marginal value of convex Quadratic constraints

	V1	V2	V3	V2-Dsj	V3-Dsj
>99.99 %	16	23	23	1	1
98-99.99 %	1	44	52	29	33
75-98 %	10	23	21	10	10
25-75 %	11	22	20	29	24
0-25 %	91	17	13	60	61
Average Gap Closed	24.80%	76.49%	80.86%	41.54 %	42.90%

- V2-Dsj closes 35% less gap than V2.
- V3-Dsj closes 38 % less gap than V3.

Marginal Value of Disjunctive Programming

Solve convex relaxation

$(\hat{x}, \hat{Y})$

Extract eigenvalues and eigenvectors of $\hat{Y} - \hat{x}\hat{x}^T$

$\lambda_i < 0$

$\lambda_i > 0$

$(c_i^T x)^2 \leq c_i^T Y c_i$

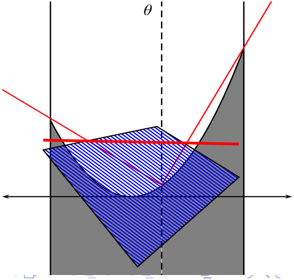
$c_i^T Y c_i \leq (c_i^T x)^2$

UGMIP

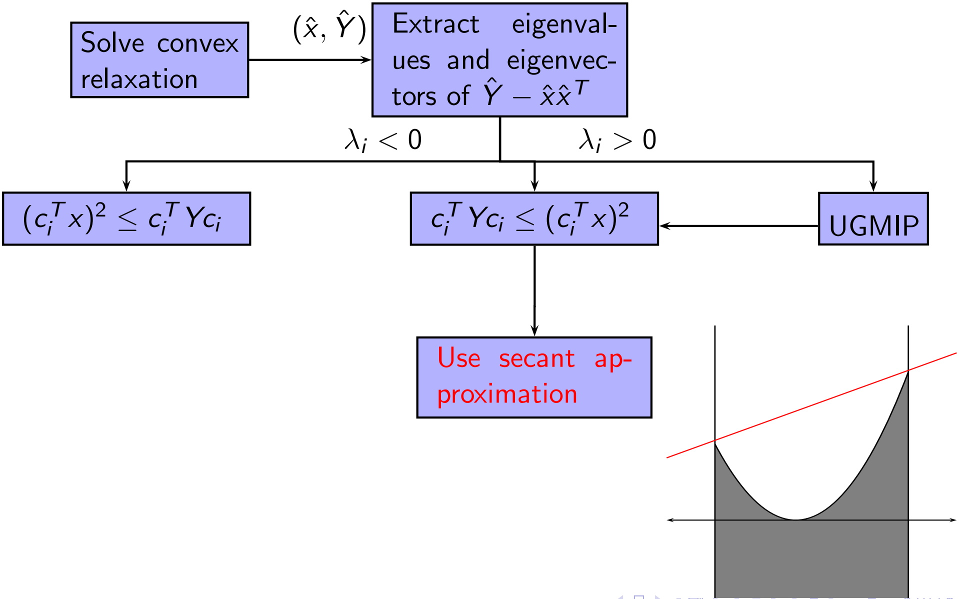
Derive disjunction

CGLP

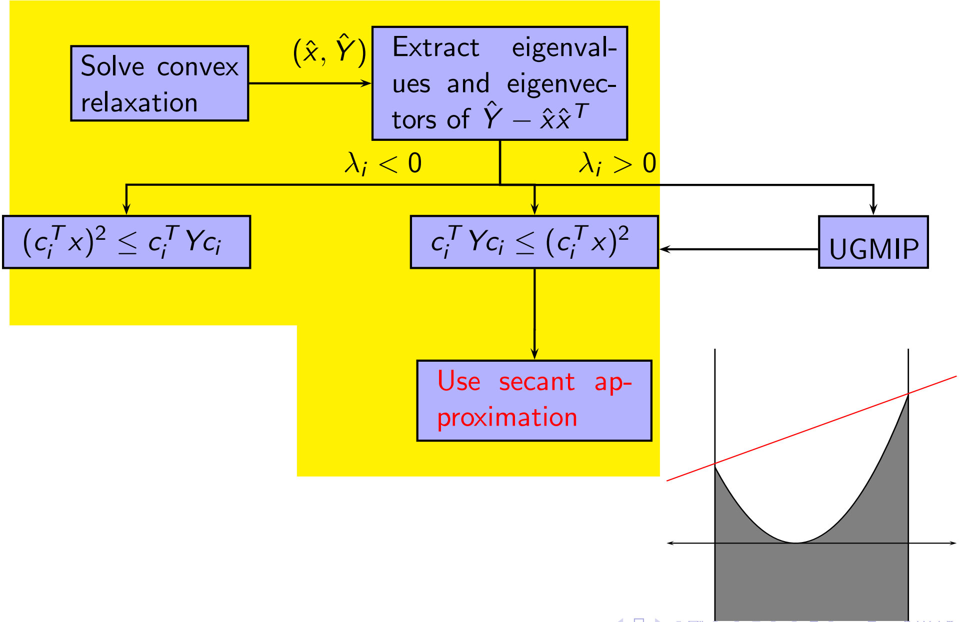
Disjunctive cut $\alpha^T x \geq \beta$



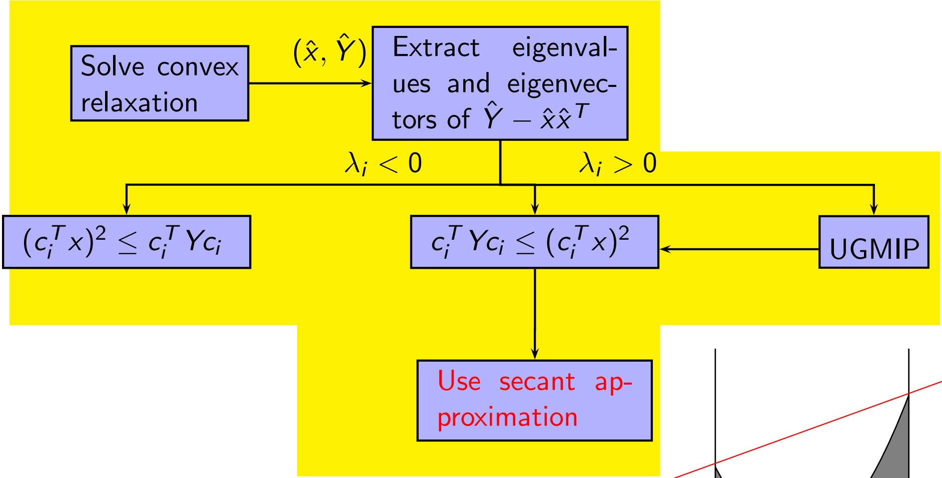
Marginal Value of Disjunctive Programming

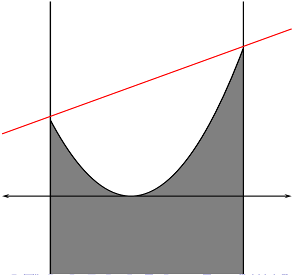


Marginal Value of Disjunctive Programming V2-SA



Marginal Value of Disjunctive Programming





Marginal value of Disjunctive Programming

	V1	V2	V3	V2-SA	V3-SA
>99.99 %	16	23	23	24	27
98-99.99 %	1	44	52	4	6
75-98 %	10	23	21	17	25
25-75 %	11	22	20	26	22
0-25 %	91	17	13	58	49
Average Gap Closed	24.80%	76.49%	80.86%	44.40%	52.56%

- V2-SA closes 32% less gap than V2.
- V3-SA closes 28 % less gap than V3.

Conclusion of experiments

- We are able to almost solve moderate size MIQCP to optimality by using only cutting planes.
- Both SDP constraints and Disjunctive Programming are necessary.
- The approach require lifting in a space of dimension $\mathcal{O}(n^2)$
- What about CPU times?

	V1	V2
Average Gap Closed	24.80%	76.49%
CPU Times (sec.)	198.043	978.140

Conclusion of experiments

- We are able to almost solve moderate size MIQCP to optimality by using only cutting planes.
- Both SDP constraints and Disjunctive Programming are necessary.
- The approach require lifting in a space of dimension $\mathcal{O}(n^2)$
- What about CPU times?

	V1	V2
Average Gap Closed	24.80%	76.49%
CPU Times (sec.)	198.043	978.140

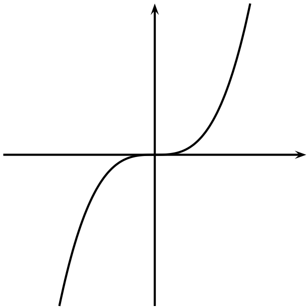
Questions

Can we get similar cuts for more general MINLP?
Can we try to obtain better CPU time?

- Key problem: can we try to get a final formulation which can be solved in a reasonable computing time (and in a small space).

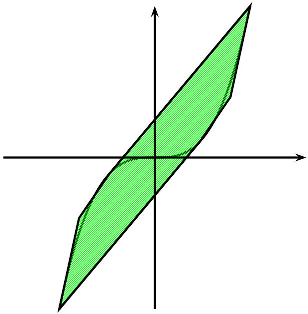
Disjunctive cuts for separable MINLP

- Adding appropriate number of variables, decompose each separable function into atomic elements of the form: $x_{n+k} = \phi_j(x_{k'})$ or $x_{n+k} = \phi_j(x_{k'}, x_{k''})$
- Using bounds on original variables, infer bounds on new variables: $l_i \leq x_{n+k} \leq u_i$.
- For each x_{n+k} build an approximation of $\text{conv}(\{x_{n+k} = \psi_i(x_{k'}), \, l_{k'} \leq x_{k'} \leq u_{k'}\})$



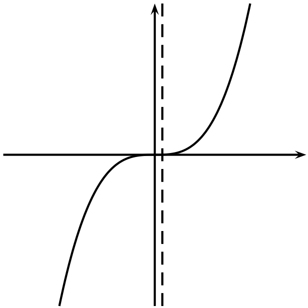
Disjunctive cuts for separable MINLP

- Adding appropriate number of variables, decompose each separable function into atomic elements of the form: $x_{n+k} = \phi_j(x_{k'})$ or $x_{n+k} = \phi_j(x_{k'}, x_{k''})$
- Using bounds on original variables, infer bounds on new variables: $l_i \leq x_{n+k} \leq u_i$.
- For each x_{n+k} build an approximation of $\text{conv}(\{x_{n+k} = \psi_i(x_{k'}), \ l_{k'} \leq x_{k'} \leq u_{k'}\})$



Disjunctive cuts for separable MINLP

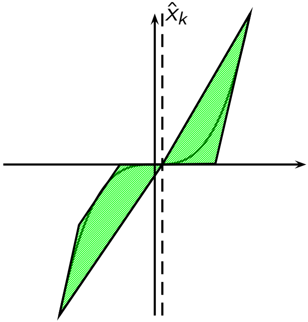
- Adding appropriate number of variables, decompose each separable function into atomic elements of the form: $x_{n+k} = \phi_j(x_{k'})$ or $x_{n+k} = \phi_j(x_{k'}, x_{k''})$
- Using bounds on original variables, infer bounds on new variables: $l_i \leq x_{n+k} \leq u_i$.
- For each x_{n+k} build an approximation of $\text{conv}(\{x_{n+k} = \psi_i(x_{k'}),\ l_{k'} \leq x_{k'} \leq u_{k'}\})$



- Apply disjunction $x_k \leq \hat{x}'_k \vee x_k \geq \hat{x}'_k$.

Disjunctive cuts for separable MINLP

- Adding appropriate number of variables, decompose each separable function into atomic elements of the form: $x_{n+k} = \phi_j(x_{k'})$ or $x_{n+k} = \phi_j(x_{k'}, x_{k''})$
- Using bounds on original variables, infer bounds on new variables: $l_i \leq x_{n+k} \leq u_i$.
- For each x_{n+k} build an approximation of $\text{conv}(\{x_{n+k} = \psi_i(x_{k'}),\ l_{k'} \leq x_{k'} \leq u_{k'}\})$



- Apply disjunction $x_k \leq \hat{x}'_k \vee x_k \geq \hat{x}'_k$.
- Strengthen the convexification: get disjunction

$$\begin{pmatrix} x_k \leq \hat{x}_k \\ A^1 x \leq b^1 \end{pmatrix} \bigvee \begin{pmatrix} x_k \geq \hat{x}_k \\ A^2 x \leq b^2 \end{pmatrix}$$

Procedure for applying disjuncive cuts [Belotti 2009]

- Apply a *spatial* disjunction $x_k \leq \hat{x}'_k \vee x_k \geq \hat{x}'_k \Rightarrow$ build sub-problems $LP^{\leq}$ and $LP^{\geq}$.
- Apply bound strengthening techniques to $LP^{\leq}$ and $LP^{\geq}$.
- Construct a CGLP and get a cut.

Implemented in Couenne.

Back to our RLT relaxation

$$
\begin{aligned}
\min\ & a_0^T x \\
s.t.\ & \\
& \langle A_k, Y \rangle + a_k^T x + b_k \leq 0 \quad k = 1 \dots m \\
& l \leq x \leq u \\
\max \left\{ \begin{array}{c} l_j x_i + l_i x_j - l_j l_i \\ u_j x_i + u_i x_j - u_j u_i \end{array} \right\} & \leq y_{ij} \leq \min \left\{ \begin{array}{c} l_i x_j + u_j x_i - l_i u_j \\ l_j x_i + u_i x_j - l_j u_i \end{array} \right\}
\end{aligned}
$$

Can we build a relaxation in the x-space which is:

- as strong as the RLT-relaxation ?

Back to our RLT relaxation

$$
\begin{aligned}
& \min \; a_0^T x \\
& \; s.t. \\
& \qquad \langle A_k, Y \rangle + a_k^T x + b_k \leq 0 \qquad k = 1 \ldots m \\
& \qquad l \leq x \leq u \\
& \max \left\{ \begin{array}{c} l_j x_i + l_i x_j - l_j l_i \\ u_j x_i + u_i x_j - u_j u_i \end{array} \right\} \leq y_{ij} \leq \min \left\{ \begin{array}{c} l_i x_j + u_j x_i - l_i u_j \\ l_j x_i + u_i x_j - l_j u_i \end{array} \right\} \\
& \qquad \; Y - xx^T \succeq 0
\end{aligned}
$$

Can we build a relaxation in the x-space which is:

- as strong as the RLT-relaxation ?
- as strong as SDP+RLT relaxation (V1)?

Back to our RLT relaxation

$$
\begin{aligned}
&\min \ a_0^T x \\
&\ s.t. \\
&\qquad \langle A_k, Y \rangle + a_k^T x + b_k \le 0 \qquad k = 1 \dots m \\
&\qquad l \le x \le u \\
&\max \left\{ \begin{array}{c} l_j x_i + l_i x_j - l_j l_i \\ u_j x_i + u_i x_j - u_j u_i \end{array} \right\} \le y_{ij} \le \min \left\{ \begin{array}{c} l_i x_j + u_j x_i - l_i u_j \\ l_j x_i + u_i x_j - l_j u_i \end{array} \right\} \\
&\qquad Y - x x^T \succeq 0
\end{aligned}
$$

Can we build a relaxation in the x-space which is:

- as strong as the RLT-relaxation ?
- as strong as SDP+RLT relaxation (V1)?
- as strong as V2, V3?

Projection theorem for RLT

Theorem

Suppose that $\hat{x} \in \mathbb{R}^N$ satisfies $l_j \leq \hat{x}_j \leq u_j \ \forall j$.
$\hat{x}$ feasible for RLT if and only if the following is non-positive.

$$
\begin{array}{l}
\min \ \eta \\
s.t. \\
-\langle A_k, Y \rangle + \eta \geq a_k^T \hat{x} + b_k \ , \quad \forall k \in M \ ; \\
y_{ij}^-(\hat{x}) \leq Y_{ij} \leq y_{ij}^+(\hat{x}) \ , \quad \forall i,j \in N \ .
\end{array}
\qquad \textbf{(DProjLP)}
$$

where

$$
\begin{aligned}
y_{ij}^-(x) &= \max \ \{u_i x_j + u_j x_i - u_i u_j, l_i x_j + l_j x_i - l_i l_j\} \quad \forall i,j \\
y_{ij}^+(x) &= \min \ \{l_i x_j + u_j x_i - l_i u_j, u_i x_j + l_j x_i - u_i l_j\} \quad \forall i,j.
\end{aligned}
$$

Projection theorem for RLT

Theorem

Suppose that $\hat{x} \in \mathbb{R}^N$ satisfies $l_j \leq \hat{x}_j \leq u_j \ \forall j$.
$\hat{x}$ feasible for RLT if and only if the following is non-positive.

$$
\begin{aligned}
&\min \ \eta \\
&s.t. \\
&-\langle A_k, Y\rangle + \eta \geq a_k^T \hat{x} + b_k \ , \quad \forall k \in M \ ; \\
&y_{ij}^-(\hat{x}) \leq Y_{ij} \leq y_{ij}^+(\hat{x}) \ , \quad \forall i,j \in N \ .
\end{aligned}
\qquad \textbf{(DProjLP)}
$$

If **DProjLP** has positive objective value and (μ, B, C) are optimal dual multipliers then

$$\sum_{i,j} \left(B_{ij} y_{ij}^-(x) - C_{ij} y_{ij}^+(x) \right) + \sum_{k \in M} \mu_k \left(a_k^T x + b_k \right) \leq 0 \qquad (2)$$

is a valid inequality that cuts off $\hat{x}$.

Projecting SDP+RLT relaxation

Theorem

Suppose that $\hat{x} \in \mathbb{R}^N$ *satisfies* $l_j \leq \hat{x}_j \leq u_j \; \forall j$.
$\hat{x}$ *feasible for RLT if and only if the following is non-positive.*

$$
\begin{aligned}
& \min \;\; \eta \\
& s.t. \\
& -\langle A_k, Y \rangle + \eta \geq a_k^T \hat{x} + b_k, \quad \forall k \in M \; ; \\
& Y + \eta I - \hat{x}\hat{x}^T \succcurlyeq 0 \; ; \\
& y_{ij}^-(\hat{x}) \leq Y_{ij} \leq y_{ij}^+(\hat{x}), \quad \forall i,j \in N \; .
\end{aligned}
\qquad (\textbf{DProjSDP})
$$

If **DProjSDP** *has positive objective value and* (μ, B, C, D) *are optimal dual multipliers then*

$$x^T B x + \sum_{i,j} \left(C_{ij} y_{ij}^-(x) - D_{ij} y_{ij}^+(x) \right) + \sum_{k \in M} u_k \left(a_k^T x + b_k \right) \leq 0 \quad (3)$$

is a valid inequality that cuts off $\hat{x}$.

Projecting SDP+RLT relaxation

Theorem

Suppose that $\hat{x} \in \mathbb{R}^N$ *satisfies* $l_j \leq \hat{x}_j \leq u_j \ \forall j$.
$\hat{x}$ *feasible for RLT if and only if the following is non-positive.*

$$
\begin{aligned}
& \min \ \eta \\
& s.t. \\
& -\langle A_k, Y \rangle + \eta \geq a_k^T \hat{x} + b_k, \quad \forall k \in M \ ; \\
& Y + \eta I - \hat{x}\hat{x}^T \succcurlyeq 0 \ ; \\
& y_{ij}^-(\hat{x}) \leq Y_{ij} \leq y_{ij}^+(\hat{x}), \quad \forall i,j \in N \ .
\end{aligned}
\qquad (\textbf{DProjSDP})
$$

If **DProjSDP** *has positive objective value and* (μ, B, C, D) *are optimal dual multipliers then*

$$x^T B x + \sum_{i,j} \left(C_{ij} y_{ij}^-(x) - D_{ij} y_{ij}^+(x) \right) + \sum_{k \in M} u_k \left(a_k^T x + b_k \right) \leq 0 \quad (3)$$

is a valid inequality that cuts off $\hat{x}$.

Low dimensional projections: Eigen Reformulation

Reformulate $x^T Ax + a^T x + b \leq 0$ using spectral decomposition of $A = \sum_k \lambda_k c_k c_k^T$:

$$\sum_{\lambda_k > 0} \lambda_k \left(c_k^T x \right)^2 + a^T x + b + \sum_{\lambda_k < 0} \lambda_k s_k \leq 0,$$
$$y_k = c_k^T x \ , \quad \forall \ k : \ \lambda_k < 0,$$
$$s_k = y_k^2 \ , \quad \forall \ k : \ \lambda_k < 0,$$

- Identifies directions of convexity.

Low dimensional projections: Eigen Reformulation

Reformulate $x^T A x + a^T x + b \leq 0$ using spectral decomposition of $A = \sum_k \lambda_k c_k c_k^T$:

$$\sum_{\lambda_k > 0} \lambda_k \left(c_k^T x \right)^2 + a^T x + b + \sum_{\lambda_k < 0} \lambda_k s_k \leq 0,$$

$$y_k = c_k^T x \ , \quad \forall\ k: \ \lambda_k < 0,$$

$$s_k = y_k^2 \ , \quad \forall\ k: \ \lambda_k < 0,$$

- Identifies directions of convexity.
- Identifies directions of maximal non-convexities.

Low dimensional projections: Eigen Reformulation

Reformulate $x^T Ax + a^T x + b \leq 0$ using spectral decomposition of $A = \sum_k \lambda_k c_k c_k^T$:

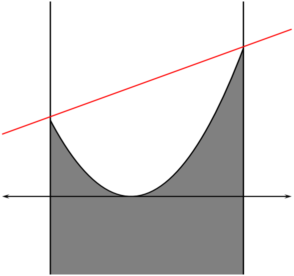
$$\sum_{\lambda_k > 0} \lambda_k \left(c_k^T x \right)^2 + a^T x + b + \sum_{\lambda_k < 0} \lambda_k s_k \leq 0,$$

$$y_k = c_k^T x \ , \quad \forall \ k : \ \lambda_k < 0,$$

$$s_k \geq y_k^2 \ , \quad \forall \ k : \ \lambda_k < 0,$$

$$L_k \ \leq y_k \leq U_k$$

$$s_k - (L_k + U_k)\, y_k + L_k U_k \geq 0.$$



- Identifies directions of convexity.
- Identifies directions of maximal non-convexities.
- Build convex envelopes in these direction.

Low dimensional projections: Eigen Reformulation

Reformulate $x^T A x + a^T x + b \leq 0$ using spectral decomposition of $A = \sum_k \lambda_k c_k c_k^T$:

$$
\begin{aligned}
& \sum_{\lambda_k > 0} \lambda_k \left(c_k^T x \right)^2 + a^T x + b + \sum_{\lambda_k < 0} \lambda_k s_k \leq 0, \\
& y_k = c_k^T x \ , \quad \forall\ k:\ \lambda_k < 0, \\
& s_k \geq y_k^2 \ , \quad \forall\ k:\ \lambda_k < 0, \\
& L_k \ \leq y_k \leq U_k \\
& s_k - (L_k + U_k)\, y_k + L_k U_k \geq 0.
\end{aligned}
$$

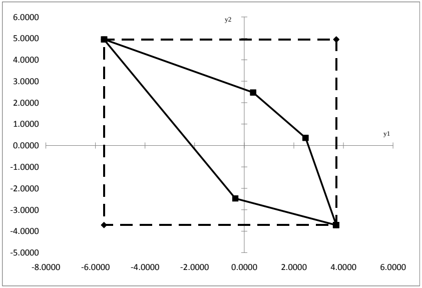
- Identifies directions of convexity.
- Identifies directions of maximal non-convexities.
- Build convex envelopes in these direction.
- Do that for all constraints and add to formulation: eigen reformulation.

Low dimensional projections: Polars.

Eigen-reformulation uses projections in d-1 for building convex envelopes.

Polarity Cuts

Do a 2-d projection and use polarity to generate cuts



- Compute extreme points in 2D space spanned by 2 ev
- Lift points to non-linear space.
- Compute cuts using polarity.

Computational results: projected formulations

ProjLP Eigen-value reformulation + Projected RLT.

ProjLP+ adds 2-dim polar cuts

For both we can separate additional disjunctive cuts by using spatial disjunctions:

$$\left(x_k \le \frac{l_k + u_k}{2}\right) \bigvee \left(x_k \ge \frac{l_k + u_k}{2}\right) .$$

Comparisons to gap closed by RLT relaxation.

Results

	ProjLP	ProjLP+	disjunctive cuts ProjLP	disjunctive cuts ProjLP+	V1	V2
>99.99	19	23	19	23	16	23
98-99.99	5	21	22	31	1	44
75-98	17	18	35	33	10	23
25-75	26	32	34	23	11	22
0-25	57	30	14	14	87	13
0	4	4	4	4	0	0
Average Gap Closed	40.92%	60.48 %	70.65%	76.06%	25.59%	79.34%
Average Time taken (sec)	0.893	0.814	4.616	19.462	198.043	978.140

Results

			disjunctive cuts			
	ProjLP	**ProjLP+**	**ProjLP**	**ProjLP+**	**V1**	**V2**
>99.99	19	23	19	23	16	23
98-99.99	5	21	22	31	1	44
75-98	17	18	35	33	10	23
25-75	26	32	34	23	11	22
0-25	57	30	14	14	87	13
0	4	4	4	4	0	0
Average Gap Closed	40.92%	60.48 %	70.65%	76.06%	25.59%	79.34%
Average Time taken (sec)	0.893	0.814	4.616	19.462	198.043	978.140

Results

			disjunctive cuts			
	ProjLP	**ProjLP+**	**ProjLP**	**ProjLP+**	**V1**	**V2**
>99.99	19	23	19	23	16	23
98-99.99	5	21	22	31	1	44
75-98	17	18	35	33	10	23
25-75	26	32	34	23	11	22
0-25	57	30	14	14	87	13
0	4	4	4	4	0	0
Average Gap Closed	40.92%	60.48 %	70.65%	76.06%	25.59%	79.34%
Average Time taken (sec)	0.893	0.814	4.616	19.462	198.043	978.140

Projecting SDP+RLT relaxation: box-QPs

Theorem

Suppose that $\hat{x} \in \mathbb{R}^N$ satisfies $l_j \le \hat{x}_j \le u_j \ \forall j$.
$\hat{x}$ feasible for RLT if and only if the following is non-positive.

$$
\begin{aligned}
&\min \ \eta \\
&s.t. \\
&-\langle A, Y\rangle + \eta \ge a^T \hat{x} + b; \\
&Y + \eta I - \hat{x}\hat{x}^T \succcurlyeq 0 \ ; \\
&y_{ij}^-(\hat{x}) \le Y_{ij} \le y_{ij}^+(\hat{x}), \quad \forall i,j \in N \ .
\end{aligned}
\qquad (\textbf{DProjSDP})
$$

If **DProjSDP** *has positive objective value and (μ, B, C, D) are optimal dual multipliers then*

$$
x^T B x + \sum_{i,j} \left(C_{ij} y_{ij}^-(x) - D_{ij} y_{ij}^+(x) \right) + \sum_{k \in M} u_k \left(a_k^T x + b_k \right) \le 0 \quad (4)
$$

is a valid inequality that cuts off $\hat{x}$.

Projected Sub-gradient heuristic

Theorem

DProjSDP *is equivalent to* $\max\{F(B) : B \succeq 0\}$*, where*

$$\begin{aligned} F(B) &= \sum_{i,j} (A_{ij} - B_{ij})^{+} \left(y_{ij}^{-}(\hat{x}) - \hat{x}_i \hat{x}_j \right) \\ &\quad + \sum_{i,j} (A_{ij} - B_{ij})^{-} \left(y_{ij}^{+}(\hat{x}) - \hat{x}_i \hat{x}_j \right) \\ &\quad + (\hat{x}^T A \hat{x}) + \left(a^T \hat{x} + b \right) \ . \end{aligned}$$

1. initialize $\hat{B}$ as the projection of A onto the SDP cone.
2. Compute a sub-gradient of $F(B)$ at $\hat{B}$.
3. Perform line search along sub-gradient direction.
4. Update $\hat{B}$ and go to 2.

Results

	% Gap Closed		Time Taken (sec)	
Instance	ProjLP-SDP	ProjLP	ProjLP-SDP	ProjLP
spar20*	94.60 - 99.97	91.54 - 99.91	2.49 - 408.36	0.84 - 2.46
spar30*	89.87 - 99.99	51.41 - 98.79	12.33 - 565.88	1.74 - 14.38
spar40*	87.85 - 99.60	21.78 - 89.63	35.77 - 134.8	4.16 - 65.28
spar50*	87.88 - 97.53	11.38 - 50.15	50.22 - 180.96	8.76 - 99.13
spar60*	85.78 - 90.99	0.00 - 0.00	121.83 - 226.11	111.07 - 127.47
spar70*	89.78 - 99.36	0.00 - 53.67	191.12 - 693.28	22.02 - 202.98
spar80*	88.13 - 97.49	2.94 - 56.23	257.62 - 892.96	34.77 - 67.66
spar90*	89.44 - 96.60	5.73 - 50.13	408.73 - 991.04	46.98 - 95.66
spar100*	92.15 - 96.46	8.17 - 51.79	538.03 - 1509.96	75.49 - 112.69
Average	**95.19%**	**50.01%**	**280.50**	**37.89**

Comparison with black-box SDP solvers

	% Duality Gap Closed			Time Taken (sec)			Time to solve last relaxation (sec)
Instance	SDPLR	SDPA	W3	SDPLR	SDPA	W3	W3
spar20*	99 - 100	99 - 100	94 - 100	0.97 - 56	2 - 3.39	2 - 408	0.05 - 0.32
spar30*	98 - 100	98 - 100	90 - 100	3.57 - 243	16 - 29	12 - 565	0.06 - 0.89
spar40*	97 - 100	97 - 100	88 - 99.6	10 - 515	105 - 157.83	35 - 134	0.16 - 1.18
spar50*	96 - 100	96 - 100	88 - 98	41 - 926	438 - 589	50 - 180	0.13 - 0.86
spar60*	99 - 100	99 - 100	85 - 90	88 - 532	1150 - 1408	121 - 226	0.53 - 1.55
spar70*	98 - 100	98 - 100	90 - 99	133 - 3600	2769 - 3721	191 - 693	0.48 - 1.1
spar80*	98 - 100	98 - 100	88 - 98	965 - 5413	6618 - 8285	257 - 892	0.56 - 2.02
spar90*	98 - 100	98 - 100	89 - 97	2403 - 7049	12838 - 17048	408 - 991	0.77 - 1.51
spar100*	98 - 99	98 - 99	92 - 97	5355 - 10295	23509 - 28604.	538 - 1509	0.82 - 2
Average	**99.40%**	**99.40%**	**95.19%**	**1741.20**	**5247.04**	**280.50**	**0.67**

Table: Summary Results: Comparison with SDP Solvers

	No. Variables		No. Constraints Linear		No. Constraints Convex (Non-Linear)		Computing Time (sec)		% Duality Gap Closed	
Instances	SDP	Proj	SDP	Proj	SDP	Proj (Quad)	SDP	Proj	SDP	Proj
spar100-025-1	5151	203	20201	156	1	119	5719.42	1.14	98.93%	92.36%
spar100-025-2	5151	201	20201	151	1	95	10185.65	1.52	99.09%	92.16%
spar100-025-3	5151	201	20201	150	1	114	5407.09	1.24	99.33%	93.26%
spar100-050-1	5151	201	20201	150	1	98	10139.57	1.07	98.17%	93.62%
spar100-050-2	5151	201	20201	150	1	113	5355.20	1.26	98.57%	94.13%
spar100-050-3	5151	201	20201	150	1	97	7281.26	0.82	99.39%	95.81%
spar100-075-1	5151	201	20201	150	1	131	9660.79	2.00	99.19%	95.84%
spar100-075-2	5151	201	20201	150	1	109	6576.10	1.23	99.18%	96.47%
spar100-075-3	5151	199	20201	147	1	90	10295.88	0.87	99.19%	96.06%

Part III

Algotithms for Convex MINLPs

Convex MINLP

We now consider:

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & g_j(x) \leq 0 \quad \forall j \in J, \\ & x \in X, \quad x_I \in \mathbb{Z}^{|I|}, \end{aligned} \tag{MINLP}$$

in the case when f and g are convex.
Given bounds $(l_I, u_I) = \{(\ell_i, u_i) \mid \forall i \in I\}$, the *NLP relaxation* of (MINLP) is

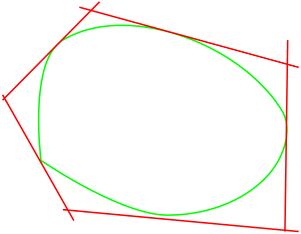
$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & g_j(x) \leq 0 \quad \forall j \in J, \\ & x \in X; \quad l_I \leq x_I \leq u_I. \end{aligned} \tag{NLPR(l_I, u_I)}$$

and can be solved to optimality by a standard NLP solver.

NLP Branch-and-Bound

0. **Initialize:**
   $\mathcal{L} \leftarrow \{(L_I, U_I)\}$. $z_U = \infty$. $x^* \leftarrow$ NONE.
1. **Terminate?**
   Is $\mathcal{L} = \emptyset$? If so, the solution x^* is optimal.
2. **Select.**
   Choose and delete a problem $N^i = (l_I^i, u_I^i)$ from $\mathcal{L}$.
3. **Evaluate.**
   Solve NLPR(l_I^i, u_I^i). If it is infeasible, go to step 1, else let $\hat{x}^i$ be its solution.
4. **Prune.**
   If $f(\hat{x}^i) \geq z_U$, go to step 1. If $\hat{x}^i$ is fractional, go to step 5, else let $z_U \leftarrow f(\hat{x}^i)$, $x^* \leftarrow \hat{x}^i$, and delete from $\mathcal{L}$ all problems with $z_L^j \geq z_U$. Go to step 1.
5. **Divide.**
   Divide the feasible region of N^i into a number of smaller feasible subregions, creating nodes $N^{i1}, N^{i2}, \ldots, N^{ik}$. For each $j = 1, 2, \ldots, k$, let $z_L^{ij} \leftarrow f(\hat{x}^i)$ and add the problem N^{ij} to $\mathcal{L}$. Go to 1.

Outer Approximation [Duran, Grossmann 1986]



$$\min f(x)$$
s.t.
$$g(x) \leq 0,$$
$$x \in X,\ x_i \in \mathbb{Z}\ \forall i \in \mathcal{I}.$$

Idea: linearize constraints at different points and build an equivalent MILP.

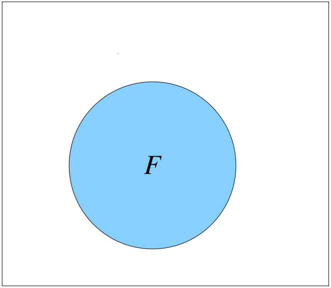
$$\begin{aligned} \min \quad & \eta \\ \text{s.t.} \quad & \eta \geq f(\overline{x}) + \nabla f(\overline{x})^T (x - \overline{x}) \\ & g_j(\overline{x}) + \nabla g_j(\overline{x})^T (x - \overline{x}) \leq 0 \quad j \in J, \\ & x \in X, \quad x_i \in \mathbb{Z}, \forall i \in \mathcal{I}. \end{aligned}$$

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.)

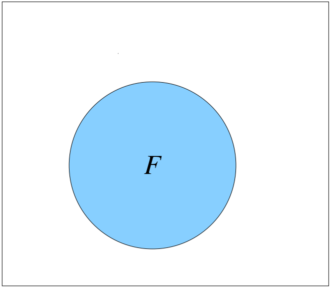


Outer approximation constraints

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.)

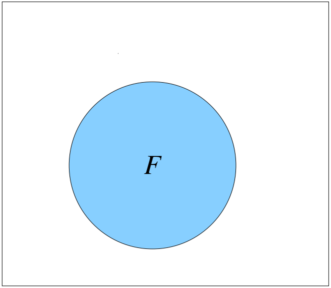


Outer approximation constraints

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.)



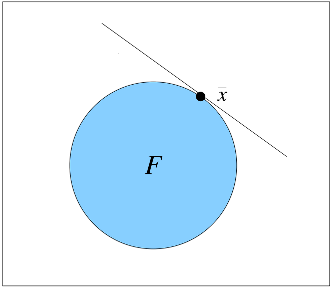
Outer approximation constraints

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 $\bar{x}$.



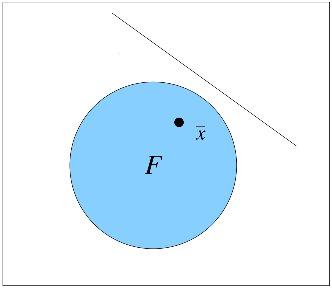
Outer approximation constraints

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 $\bar{x}$.



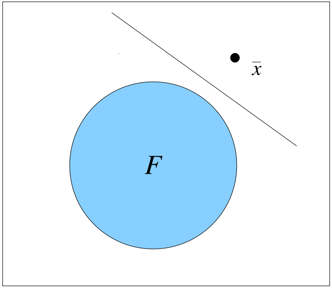
Outer approximation constraints

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 $\bar{x}$.



Subproblems

Given $\hat{x}_I$ *fixed NLP subproblem* is:

$$
\begin{aligned}
& \min f(x) \\
s.t. \quad & g_j(x) \leq 0, \ \forall j \in J \\
& x \in X; \quad x_i = \hat{x}_i, \ \forall i \in I.
\end{aligned}
\tag{NLP($\hat{x}_I$)}
$$

If $\hat{x}_i \in \mathbb{Z}$ for all $i \in I$, and feasible: gives an upper bound.
fixed NLP feasibility subproblem is:

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{m} w_j g_j(x)^+ \\
\text{s.t.} \ & x \in X, \quad x_I = \hat{x}_I,
\end{aligned}
\tag{NLPF($\hat{x}_I$)}
$$

where $g_j(x)^+ = \max\{0, g_j(x)\}$
If NLP($\hat{x}_I$) is infeasible, NLP software will typically solve NLPF($\hat{x}_I$).

Equivalent MILP formulation of convex MINLP

For each integer assignment $\hat{x}_I \in K = \mathrm{Proj}_{x_I}(X) \cap \mathbb{Z}^{|I|}$, let $\overline{k}$ be:

- An optimal solution to NLP($\hat{x}_I$) if it is feasible.
- An optimal solution to NLPF($\hat{x}_I$) otherwise.

Theorem (Duran Grossmann 86)

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

$$
\begin{aligned}
\min \quad & \eta \\
s.t. \quad & \eta \geq f(\overline{x}) + \nabla f(\overline{x})^T (x - \overline{x}) \quad \hat{x}_I \in K, \\
& g_j(\overline{x}) + \nabla g_j(\overline{x})^T (x - \overline{x}) \leq 0 \quad j \in J, \hat{x}_I \in K, \\
& x \in X, \quad x_I \in \mathbb{Z}^I.
\end{aligned}
\tag{MILP-OA}
$$

has the same optimal value as MINLP.

OA-based reduced master problem

Let $\mathcal{K} \subseteq K$

$$\begin{aligned} \min \quad & \eta \\ \text{s.t.} \quad & \eta \geq f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) \quad \hat{x} \in \mathcal{K}, \\ & g_j(\bar{x}) + \nabla g_j(\bar{x})^T (x - \bar{x}) \leq 0 \quad j \in J, \hat{x} \in \mathcal{K}, \\ & x \in X, \quad x_i \in \mathbb{Z}, \forall i \in \mathcal{I}. \end{aligned} \tag{MP($\mathcal{K}$)}$$

Reminder

Where for $\hat{x} \in \mathcal{K}$, $\bar{x}$ is the solution to

NLP($\hat{x}_I$)

$$\begin{aligned} & \min f(x) \\ s.t. \quad & g_j(x) \leq 0, \ \forall j \in J \\ & x \in X; \quad x_i = \hat{x}_i, \ \forall i \in I. \end{aligned}$$

NLPF($\hat{x}_I$):

$$\begin{aligned} \min \quad & \sum_{j=1}^{m} w_j g_j(x)^+ \\ \text{s.t.} \quad & x \in X, \quad x_I = \hat{x}_I, \end{aligned}$$

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 $(\mathrm{MP}(\mathcal{K}))$ infeasible? If so z_U is $\epsilon-$optimal.
2. **Lower Bound**
   Let $z_{\mathrm{MP}(\mathcal{K})}$ be the optimal value of $\mathrm{MP}(\mathcal{K})$ and $(\hat{\eta}, \hat{x})$ its optimal solution.
   $z_L \leftarrow z_{\mathrm{MP}(\mathcal{K})}$
3. **NLP Solve**
   Solve $(\mathrm{NLP}(\hat{x}_I))$.
   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.

Remarks on OA deconposition

- Solving many MIPs can be a high cost.
- One does not need to solve MP($\mathcal{K}$) to optimiality, a solution ϵ better than current incumbent is enough.
- It is possible to design a similar algorithm without solving NLPs.

Define a slightly different master problem:

$$
\begin{aligned}
\min \quad & \eta \\
\text{s.t.} \quad & \eta \geq f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) \quad \bar{x} \in \mathcal{K} & \text{(RM-ECP}(\mathcal{K})) \\
& g_j(\bar{x}) + \nabla g_j(\bar{x})^T (x - \bar{x}) \leq 0 \quad j \in J(\mathcal{K}) \quad \bar{x} \in \mathcal{K} \\
& x \in X, \quad x_I \in \mathbb{Z}^I
\end{aligned}
$$

where $J(\mathcal{K}) \stackrel{\text{def}}{=} \{j \in \arg\max_{j \in J} g_j(\bar{x})\}$

ECP Algorithm [Westerlund and Lunqvist 98]

0. **Initialize.**
   $z_U \leftarrow +\infty$. $z_L \leftarrow -\infty$. Choose convergence tolerances ϵ, κ.
   $\mathcal{K} \leftarrow \emptyset$.
1. **Terminate?**
   Is $z_U - z_L < \epsilon$ or is (RM-ECP($\mathcal{K}$)) infeasible? If so z_U is $\epsilon-$optimal, with an associated solution that is $\kappa-$feasible.
2. **Lower Bound**
   Let $z_{\text{RM-ECP}(\mathcal{K})}$ be the optimal value of (RM-ECP($\mathcal{K}$)) and $(\overline{\eta}^i, \overline{x}^i)$ be its optimal solution.
   $z_L \leftarrow z_{\text{RM-ECP}(\mathcal{K})}$
3. **Upper bound and refine**
   Is $g_j(\bar{x}^i) < \kappa \ \forall j \in J$?
   If so, $z_U \leftarrow \min(z_U, f(\overline{x}^i))$.
   If not, $\mathcal{K} \leftarrow \mathcal{K} \cup \{\overline{x}^i\}, t \in \arg\max_j g_j(\bar{x}^i)$, and $J(\mathcal{K}) \leftarrow J(\mathcal{K}) \cup \{t\}$
   $i \leftarrow i + 1$. Go to 1.

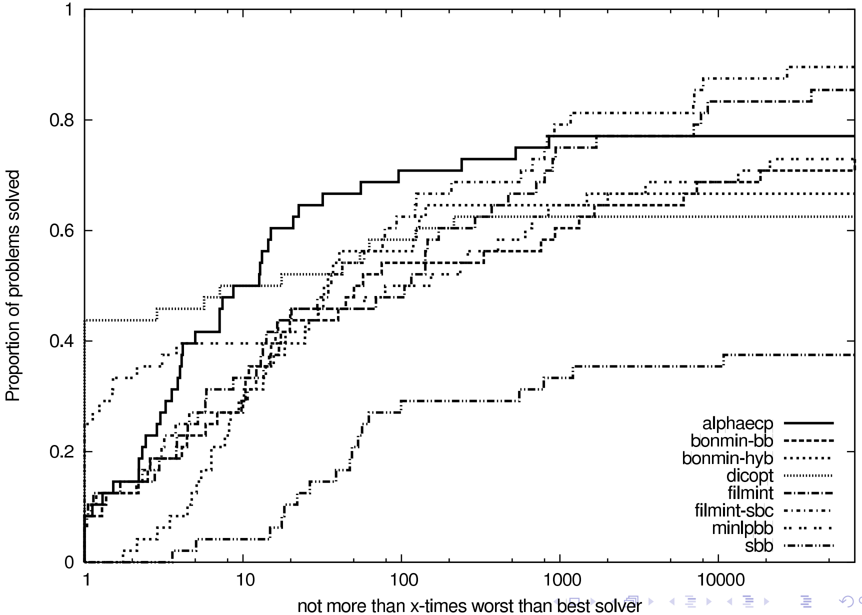
LP/NLP Based BB [Quesada, Grossmann 1993]

0. **Initialize.**
   $\mathcal{L} \leftarrow \{(L_I, U_I)\}$. $z_U \leftarrow +\infty$. $x^* \leftarrow$ NONE.
   Let $\overline{x}$ be the optimal solution of continuous relaxation.
   $\mathcal{K} \leftarrow \{\overline{x}\}$.
1. **Terminate?** Is $\mathcal{L} = \emptyset$? If so, the solution x^* is optimal.
2. **Select.** Choose and delete a problem $N^i = (l_I^i, u_I^i)$ from $\mathcal{L}$.
3. **Evaluate.** Solve LP($\mathcal{K}$, l_I^i, u_I^i). If infeasible, go to 1, else let $(\hat{\eta}^i, \hat{x}^i)$ be its solution.
4. **Prune.** If $\hat{\eta}^i \geq z_U$, go to 1.
5. **NLP Solve?** Is $\hat{x}_I^i$ integer? If so, solve (NLP($\hat{x}_I^i$)), otherwise go to 8. Let $\overline{x}^i$ be the optimal (or minimally infeasible) solution.
6. **Upper bound?** Is $\overline{x}^i$ feasible for (MINLP) and $f(\overline{x}^i) < z_U$? If so, $x^* \leftarrow \overline{x}^i$, $z_U \leftarrow f(\overline{x}^i)$.
7. **Refine.** Let $\mathcal{K} \leftarrow \mathcal{K} \cup (\overline{x})$. Go to 3 .
8. **Divide.** Divide the feasible region of N^i into a number of smaller feasible subregions, creating nodes $N^{i1}, N^{i2}, \ldots, N^{ik}$. For each $j = 1, 2, \ldots, k$, let $z_L^{ij} \leftarrow z_{\mathrm{MPR}(\mathcal{K}, l_I^i, u_I^i)}$ and add the problem N^{ij} to $\mathcal{L}$. Go to step 1.

Improvements to cut generation in LP/NLP based BB

- Add more cuts with three rules of thumb [Linderoth]
  1. Generate cuts early in the procedure.
  2. Measure effect in term of bound improvement.
  3. Stop when bound stalls.
- Three type of cuts proposed [Abishek et. al. 2010]:
  1. *ECP*: generate cuts at current fractional LP optimum $\hat{x}_I$.
  2. *FixFRAC*: Solve NLP($\hat{x}_I$).
  3. *NLP*: Solve NLPR(l_I^i, u_I^i) [also B. et. al. 2008].
- Also since relaxation is linear can use all cuts from MILP.
- Be careful not to overwhelm LP solver with too many cuts.

Comparison of state of the art solvers



Formulation of hulls of unions of convex sets

Let $C = \{g_j(x) \leq 0 \, \forall j = 1, \ldots, m, 0 \leq x_i \leq 1 \forall i \in I\}$ be a bounded convex set with g_j, $j = 1, \ldots, m$ convex functions bounded in C.
For $i \in I$ and $k = 0, 1$ define $C_i k = \{x \in C : x_i = k\}$

Proposition ([Stubbs, Merhotra 1999])

$$conv\left(C_0^i \cup C_1^i\right) = proj_x\left(\mathcal{M}_i\left(C\right)\right)$$

with

$$\mathcal{M}_i\left(C\right) = \left\{\begin{array}{r} x = \lambda_0 y + \lambda_1 z \\ 1 = \lambda_0 + \lambda_1 \\ y \in C_0^i, z \in C_1^i \\ \lambda_0,\ \lambda_1 \geq 0 \end{array}\right\}$$

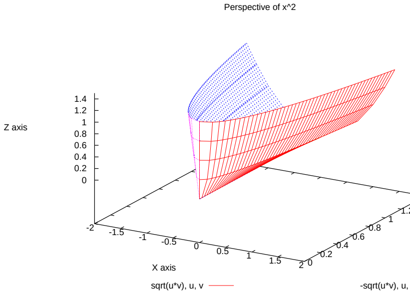
(By definition of convex hull)

Perspective function

For a given function $g_i(x) : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, let

$$\tilde{g}_i(x, z) \equiv \begin{cases} z g_i(x/z) & \text{if } z > 0, \\ \lim_{\lambda \to 0+} \lambda g(\tilde{x} - x + x/\lambda) & \text{if } z = 0 \end{cases}$$

- If g_i is a convex function $\tilde{g}_i$ is also convex.
- $\tilde{g}_i$ is positively homogeneous: $\lambda \tilde{g}_i(x) = \tilde{g}_i(\lambda x) \ \forall \lambda 0$.



Convex formulation of the convex-hull

Let $\tilde{C} = \{\tilde{g}_j(x,\lambda) \leq 0,\ \forall j = 1,\dots,m,\ 0 \leq x_i \leq \lambda\ \forall i \in I\}$.

Theorem ([Stubbs, Merhotra 1999, Ceria Soares 1999[1]])

$$conv\left(C_0^i \cup C_1^i\right) = proj_x\left(\tilde{\mathcal{M}}_i\left(C\right)\right)$$

with

$$\tilde{\mathcal{M}}_i\left(C\right) = \left\{ \begin{array}{r} x = \tilde{y} + \tilde{z} \\ 1 = \lambda_0 + \lambda_1 \\ (y,\lambda_0) \in \tilde{C}, (z,\lambda_1) \in \tilde{C} \\ y_i = 0,\ z_i = \lambda_1 \\ \lambda_0,\ \lambda_1 \geq 0 \end{array} \right\}$$

[1] More general statement for unbounded set

Separation problem in non-linear case

Let $\hat{x} \in \mathbb{R}^n$ be a point to cut define the problem:

$$\begin{aligned} \min ||x - \hat{x}|| \\ x = \tilde{y} + \tilde{z} \\ 1 = \lambda_0 + \lambda_1 \\ (y, \lambda_0) \in \tilde{C}, (z, \lambda_1) \in \tilde{C} \\ y_i = 0,\ z_i = \lambda_1 \\ \lambda_0,\ \lambda_1 \geq 0 \end{aligned} \tag{5}$$

Theorem

Let $\hat{x} \notin conv\left(C_0^i \cup C_1^i\right)$ and $\overline{x}$ be an optimal solution to 5. For any subgradient ξ of $||x - \hat{x}||$ at $\overline{x}$, $\xi^T(x - \overline{x}) \geq 0$ is a valid linear inequality that cuts off $\hat{x}$.

(If dual Lagrange multipliers ξ can be easily deduced from them).

Application to problems with indicator variables

We consider the simple mixed integer nonlinear set:

$$C = \{(x,z) \in \mathbb{R}\times\{0,1\} : x = 0, \text{ if } z = 0, g(x) \leq 0, l \leq x \leq u \text{ if } z = 1\}$$

By direct application of disjunctive programming:

$$\mathrm{conv}(C) = \{(x,z) \in \mathbb{R} \times [0,1] : zl \leq x \leq zu, \tilde{g}(x,z) \leq 0\}$$

This reformulation of C is called perspective reformulation and has been shown to be very effective [Frangioni, Gentile 2006 2007 2010, Günluk Linderoth 2008 2009]

m	n	# Sol.	CPU	# Sol.	CPU
20	100	7	12263	10	3.8
20	200	0	-	10	9.6

Table: Running times on Uncapacitated Quadratic Facility Location

Application to problems with indicator variables

We consider the simple mixed integer nonlinear set:

$$C = \{(x,z) \in \mathbb{R} \times \{0,1\} : x = 0, \text{ if } z = 0, g(x) \leq 0, l \leq x \leq u \text{ if } z = 1\}$$

By direct application of disjunctive programming:

$$\text{conv}(C) = \{(x,z) \in \mathbb{R} \times [0,1] : zl \leq x \leq zu, \tilde{g}(x,z) \leq 0\}$$

This reformulation of C is called perspective reformulation and has been shown to be very effective [Frangioni, Gentile 2006 2007 2010, Günluk Linderoth 2008 2009]

m	n	# Sol.	CPU	# Sol.	CPU
20	100	7	12263	10	3.8
20	200	0	-	10	9.6

Table: Running times on Uncapacitated Quadratic Facility Location

Commercial : Hassan Hijazi talk on Wednesday.

Second Order cone

Reminder

A cone $\mathcal{K}$ is a subset of $\mathbb{R}^m$ such that $x \in \mathcal{K}$ implies $\lambda x \in \mathcal{K}$ for all $\lambda \geq 0$.

Definition

The Lorentz cone is $\mathcal{Q}^{m+1} = \{(t, t_0) \in R^m \times R : ||t|| \leq t_0\}$.
We define the partial order:

$$x \preceq y \Leftrightarrow y - x \in \mathcal{Q}^{m+1}$$

Second Order Cone Constraint

$$(Ax - b, u) \preceq 0$$

which is equivalent to

$$||Ax - b|| \leq u$$

Mixed Integer Second Order Cone Programming

$$\min c^T x + h^T y$$
$$||A_i x + G_i y - b_i|| \leq (u_i^T x + v_i^T y - w_i) \quad i = 1, \ldots, k$$
$$x \in \mathbb{Z}^n, y \in \mathbb{R}^p$$

Families of cuts for MISOCP

- Generalization of Chvátal-Gomory [Cezik and Yiengar 2005].
- Generalization of MIR cuts [Atamtürk and Narayan 2010].
- Lift-and-project cuts [Drewes 2009].

Mixed Integer Second Order Cone Programming

$$
\begin{aligned}
\min\ & c^T x + h^T y \\
& ||A_i x + G_i y - b_i|| \le (u_i^T x + v_i^T y - w_i) \quad i = 1, \ldots, k \\
& x \in \mathbb{Z}^n, y \in \mathbb{R}^p
\end{aligned}
$$

Families of cuts for MISOCP

- Generalization of Chvátal-Gomory [Cezik and Yiengar 2005].
- Generalization of MIR cuts [Atamtürk and Narayan 2010].
- Lift-and-project cuts [Drewes 2009].

Commercial: Sarah Drewes talk.