Relaxation-Based Methods for Solving Nonconvex MINLPs

Ivo Nowak
Humboldt University Berlin

June 30, 2003

Overview

- The nonconvex MINLP problem
- Reformulations
- Relaxations
- Solution Algorithms

The nonconvex MINLP problem

MINLP

(P)
$$
\begin{array}{ll}
\min & h_0(x) \\
\text{s.t.} & h_E(x) = 0 \\
& h_I(x) \leq 0 \\
& x \in [\underline{x}, \overline{x}] \\
& x_j \in \{\underline{x}_j, \overline{x}_j\}, j \in B
\end{array}
$$

MINLP⇔GO
Piecewise C^2 models can be reformulated to be C^2

Applications: process engineering, communication, finance, marketing, and other areas.

Structural Properties

- (P) is called convex if all h_i are convex
- (P) is called block-separable if $h_i(x) = \sum_{k=1}^{p} h_i^k(x_{J_k})$
- (P) is called quadratic if $h_i(x) = x^T A_i x + 2b_i^T x + c_i$

Analysis of 150 problems of GAMS's MINLPLib:

- 85% problems are nonconvex
- 85% problems are block-separable
- 50% problems are quadratic

MINLP solution methods

- Relaxation-based methods: branch-and-cut, disjunctive programming, outer approximation, Benders decomposition, MILP approximation
- Sampling methods: clustering methods, evolutionary algorithms, simulated annealing, tabu-search

Acceleration tools

- Constraint programming for finding good constraints and box-reduction
- Heuristics for computing near global minimizers and finding regions of interest

MILP versus MINLP

MILP/MINLP branch-and-cut algorithms:

1. Get solution candidates by projecting solutions of a relaxation onto the feasible set.

2. Improve the relaxation and the solution candidate by partitioning and adding cuts.

Large gap between MINLP and MILP codes (CPLEX, XPRESS-MP)

Differences between MINLP and MILP:

- Continuous relaxation: convex underestimation of nonconvex NLP versus LP
- Cut generation: MINLP versus MILP sub-problems
- Local solutions: NLP versus LP

Block-Separable Reformulation

- The block-structure $h_i(x) = \sum_{k=1}^{p} h_i^k(x_{J_k})$ influences the quality and computation of a relaxation:
  - small blocks: fast computation of underestimators and cuts
  - large blocks: better relaxations (smaller duality gaps)
- Many problems have a natural block-structure (model components)

Splitting-schemes

Sparse MINLPs can be reformulated to be block-separable with almost arbitrary block-sizes by

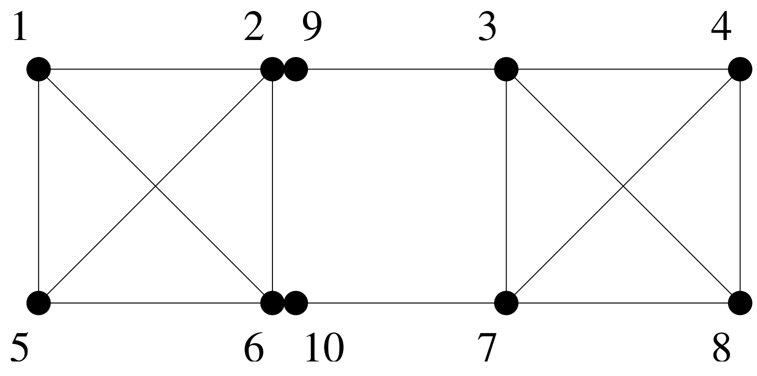
1. Partition the sparsity graph:

   $$E_{\text{sparse}} = \{(k,l) \mid \frac{\partial^2}{\partial x_k \partial x_l} h_i(x) \neq 0 \text{ for some } i \in \{0,\ldots,m\} \text{ and some } x \in [\underline{x}, \overline{x}]\}$$

   into blocks $J_1, \ldots, J_p$

2. For each adjacent node set $R_k = \{i \in \bigcup_{l=k+1}^{p} J_l \mid (i,j) \in E_{\text{sparse}},\ j \in J_k\}$, add new variables $y^k \in \mathbb{R}^{|R_k|}$ and copy-constraints $x_{R_k} = y^k$ where $k = 1, \ldots, p$.

Example



- Blocks: and $J_1 = \{3, 4, 7, 8\}$ and $J_2 = \{1, 2, 5, 6, \}$
- Adjacent nodes: $R_1 = \{2, 6\}$, $R_2 = \emptyset$
- New nodes: x_9 and x_{10}
- Copy constraints: $x_2 = x_9$ and $x_6 = x_{10}$
- New blocks: $J_1 = \{3, 4, 7, 8, 9, 10\}$ and $J_2 = \{1, 2, 5, 6, \}$

Extended blockseparable reformulation

By replacing block-separable constraints

$$\sum_{k=1}^{p} h_i^k(x_{J_k}) \leq 0$$

by

$$\sum_{k=1}^{p} t_{ik} \leq 0, \quad g_i^k(x_{J_k}, t_{ik}) = h_i^k(x_{J_k}) - t_{ik} \leq 0, \quad k = 1, \dots, p,$$

we obtain a problem with linear coupling constraints

$$(\mathrm{P}_{\text{ext}}) \qquad \begin{array}{ll} \min & c^T x + c_0 \\ \text{s.t.} & x \in H = \{x \mid Ax + b \leq 0\} \\ & x \in G = \times_{k=1}^{p} \{x_{I_k} \mid g^k(x_{I_k}) \leq 0\} \\ & x \in X = [\underline{x}, \overline{x}] \end{array}$$

(useful for generating cuts)

Convex relaxation and Lagrangian relaxation

Convex relaxation of (P$_{\text{ext}}$):

(C) $$\min\{c^T x + c_0 \mid x \in \text{conv}(G \cap X) \cap H\}$$

Lagrangian relaxation of (P$_{\text{ext}}$):

(D) $$\max_\mu \min_x \{c^T x + c_0 + \mu^T(Ax + b) \mid x \in G \cap X\}$$

- $\text{sol}(C) \neq \text{sol}(D)$, but $\text{val}(D) = \text{val}(C)$, since

$$\text{val}(D) = \max_\mu \min_x \{c^T x + c_0 + \mu^T(Ax + b) \mid x \in \text{conv}(G \cap X)\}$$

- Duality gap $\text{val}(\text{P}_{\text{ext}})$-$\text{val}(D)$ smaller if blocks larger

Decomposition methods for computing relaxations

1. Dual methods:
   solve (D) approximately by a subgradient method

2. Cutting plane methods:
   solve (C) approximately by generating supporting hyperplanes

3. Column generation:
   solve (C) approximately by generating extreme points and extreme rays

Decomposition:
Subgradients, supporting hyperplanes and extreme points (rays) are computed by solving several small MINLPs.

Semidefinite Relaxation

MIQQP

$$(Q) \qquad \begin{array}{ll} \min & q_0(x) \\ \text{s.t.} & q_i(x) \le 0, \quad i = 1, \dots, m \\ & x \in [\underline{x}, \overline{x}], \quad x_B \text{ binary} \end{array}$$

where $q_i(x) = x^T A_i x + 2b_i^T x + c_i, \quad i = 0, \dots, m$

- Quadratically constrained quadratic programming (QQP) reformulation by:

$$x_j \in [\underline{x}_j, \overline{x}_j] \Leftrightarrow (x_j - \underline{x}_j)(x_j - \overline{x}_j) \le 0, \qquad j \in \{1, \dots, n\} \setminus B$$
$$x_j \in \{\underline{x}_j, \overline{x}_j\} \Leftrightarrow (x_j - \underline{x}_j)(x_j - \overline{x}_j) = 0, \qquad j \in B$$

- (Q) $\Leftrightarrow$ polynomial programs

A spectral dual method

- The dual (D) of a QQP is equivalent to a semidefinite program
- Eigenvalue formulation of the dual function:

$$D(\mu) = c(\mu) + \sum_k \lambda_1(A^k(\mu))$$

- For each $\mu \in \operatorname{dom} D$, the Lagrangian is convex
- (D) is solved by the bundle method NOA (Kiwiel)
  (fast initial improvement versus accurate solution)

Numerical experiments

- Data: splitting schemes of sparse MIQQPs up to 1000 variables (No 02)
- Computing times for 100 iterations: 1-4 sec.
- Eigenvalue computation more stable (QL versus Lanczos)
- Evaluation of decomposed dual function is faster (factor 10-100)

Nonlinear Convex Relaxation of MINLP

Difficulties with dual approach:

- Lagrangian relaxation is a MINLP problem
- Lagrangian is usually not convex for $\mu \in \operatorname{dom} D$

The restricted dual problem

(D$_+$) $$\max_{\mu \in \mathcal{M}_+} \quad D(\mu),$$

with

$$\mathcal{M}_+ = \{\mu \in \mathcal{M} \mid \nabla^2 L(x;\mu) \succcurlyeq 0 \text{ for all feasible } x\}$$

is too difficult to solve !

Convex underestimator

Replacement of nonlinear functions h_i by a convex underestimator $\breve{h}_i$ in (P) yields a nonlinear convex relaxation:

$$(\mathrm{C_{nlp}}) \qquad \begin{array}{ll} \min & \breve{h}_i(x) \\ \text{s.t.} & \breve{h}_i(x) \leq 0, \quad i = 1, \ldots, m \\ & x \in [\underline{x}, \overline{x}] \end{array}$$

α-underestimators (Adjiman and Floudas 97)

$$\breve{f}(x) = f(x) + \langle \alpha, \mathrm{Diag}(x - \underline{x})(x - \overline{x}) \rangle$$

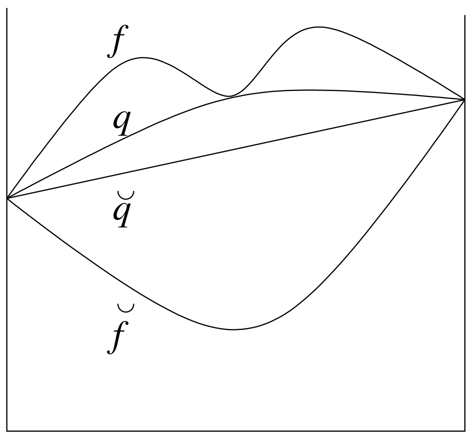
and $\alpha \geq 0$ such that $\nabla^2 \breve{f}(x) = \nabla^2 f(x) + 2\,\mathrm{Diag}(\alpha) \succcurlyeq 0, \quad \forall x \in [\underline{x}, \overline{x}]$

Convexified polynomial underestimator

1. Generate a polynomial underestimator $q(x) \leq f(x) \quad \forall x \in [\underline{x}, \overline{x}]$ by a sampling technique (using minimizers of f)

2. Set

$$\breve{q}(x) = q(x) + \langle \alpha, \mathrm{Diag}(x - \underline{x})(x - \overline{x}) \rangle$$



Polyhedral Relaxations

(C$_{\text{ext}}$)

$$\begin{array}{ll} \min & c^T x + c_0 \\ \text{s.t.} & Ax + b \le 0 \\ & \breve{g}_i^k(x_{J_k}) \le 0, \quad i \in M_k,\ k = 1, \ldots, p \\ & x \in [\underline{x}, \overline{x}] \end{array}$$

Replace the nonlinear convex functions $\breve{g}_i^k$ of (C$_{\text{ext}}$) by linearizations at sample points (minimizers)

(R)

$$\begin{array}{ll} \min & c^T x + c_0 \\ \text{s.t.} & Ax + b \le 0 \\ & c_{ki}^T x_{J_k} + d_{ki} \le 0, \quad i \in M_k, k = 1, \ldots, p \\ & x \in [\underline{x}, \overline{x}] \end{array}$$

Valid cuts

Solve separation (pricing) problem (small MINLP):

$$(\mathrm{S}_k) \qquad \begin{aligned} \delta_k = \quad & \min L_k(x_{J_k}; \hat{\mu}) \\ & \text{s.t. } g_i^k(x_{J_k}) \le 0,\ i \in \tilde{M}_k \\ & \qquad x_{J_k} \in [\underline{x}_{J_k}, \overline{x}_{J_k}],\ x_{B \cap J_k} \text{ binary} \\ & \qquad (x_{J_k} \in \textstyle\bigvee_j G_{kj}) \end{aligned}$$

where $\hat{\mu}$ is a dual solution point of (R)

Add to (R) the valid cut:

$$L_k(x; \hat{\mu}) \ge \delta_k$$

Lower bounds:

$$\underline{v}_1 = \text{val}(C_{\text{ext}}) \quad \leq \quad \underline{v}_2 = \text{val}(R) \quad \leq \quad \text{val}(P)$$

Box-reduction:

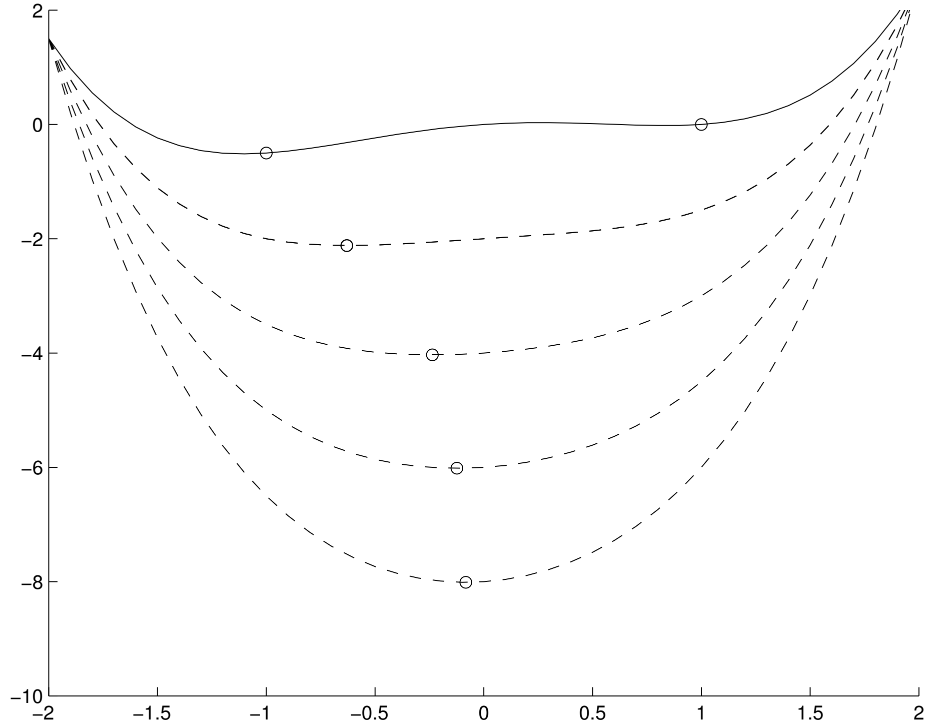
Let $\breve{S}$ be the feasible set of (C_{ext}) or (R) and set

$$X' = \Box(\breve{S}) = [\inf \breve{S}, \sup \breve{S}] \subset [\underline{x}, \overline{x}]$$

Better reduction if we include into $\breve{S}$ the level cut

$$c^T x + c_0 \leq \overline{v}$$

Deformation Heuristic



Deformation of a parametric problem (P_t) into (P), where (P_0) is a convex relaxation (C)

Assumption: (P_t) is easier to solve than (P), if t is small.

Box constraint parametric problem

Let

$$H(x;t) = (1-t)\breve{L}(x;\mu) + tP(x;t)$$

and $P(x;t)$ be a penalty function of (P) and $\breve{L}(x;\mu)$ is a Lagrangian of (C)

$$(P_t) \qquad \min_{x\in[\underline{x},\overline{x}]} H(x;t)$$

Then

$$H(x;0) = \breve{L}(x;\mu) \quad \text{and} \quad \lim_{t\to 1} \mathrm{val}(P_t) = \mathrm{val}(P)$$

The `multipath` algorithm

Input: $0 < t_1 < t_2 < \cdots < t_N < 1$ (discretization points)

1. initialize a sample set S

2. `for` $k = 1$ `to` N

   (a) `for` $x \in S$: trace a path of (P_t) from t_k to t_{k+1} starting from x

   (b) add sample points by neighbourhood search and delete sample points with high value of $P(x; \rho)$ or wich are close together

3. (local solutions)

   `for` $x \in S$: x_B=`round`(x_B), x_C=`loc_min`(x_C)

Quadratic binary programs (QBP) (MaxCut)

$$\min_{x \in \{0,1\}^n} x^T A x + 2b^T x \quad \Leftrightarrow \min_{x \in \{-1,1\}^n} x^T A x$$

Numerical experiments with a deformation heuristic, up to 3000 variables (Alperin, No 02)

- dual is an eigenvalue optimization problem
- performance guarantee (Goemans and Williamson 95)
- better than uniformely distributed multistart local optimization
- computing times: 2-20 sec.
- not necessary to solve the dual;
  the minimum eigenvalue convexification $\mu = -\lambda_1(A)e$ is sufficient

Partitioning Algorithms

Sub-Problems

$(P[U])$ $$\min\{c^T x + c_0 \mid x \in S \cap U\}$$

and

$(R[U])$ $$\min\{c^T x + c_0 \mid x \in \hat{S} \cap U\}$$

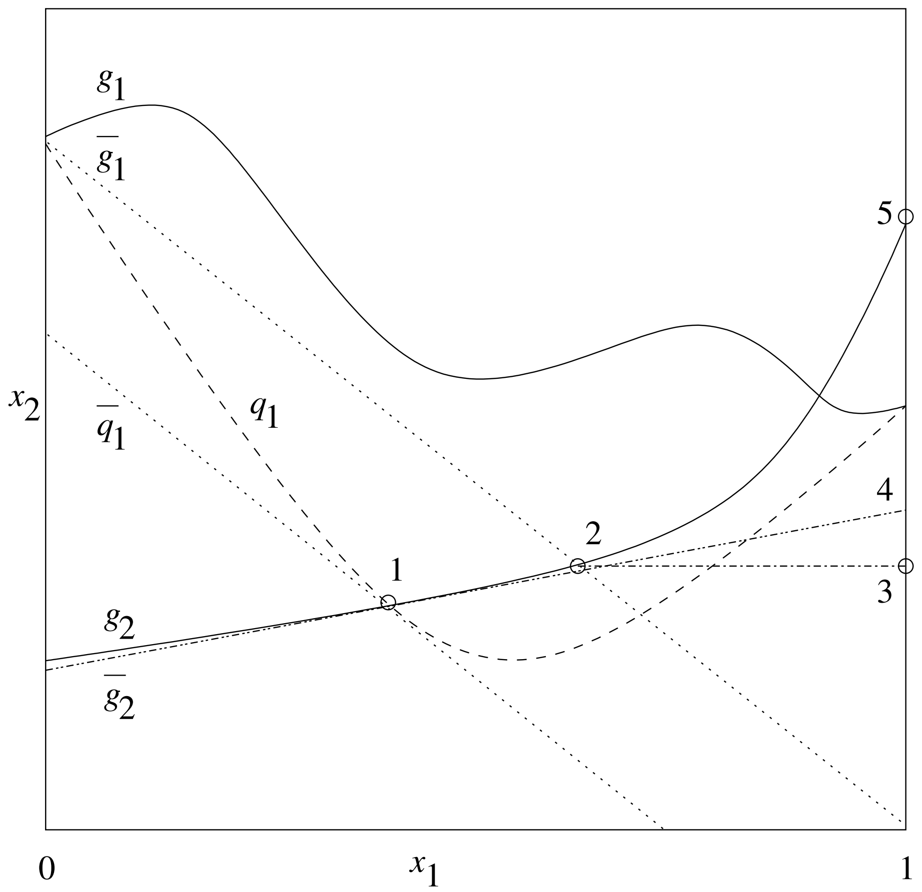
where $U \subset \mathbb{R}^n$ and S and $\hat{S}$ are the feasible sets of $(\mathrm{P}_{\mathrm{ext}})$ and (R) respectively.

Subset with fixed binary variables:

$$U_{y,K} = \{x \in \mathbb{R}^n \mid x_K = y_K\}$$

where $y \in [\underline{x}, \overline{x}]$ and $K \subseteq B$.

A rounding heuristic



1. Solve ($\mathsf{C}_{\mathrm{nlp}}$)
2. Add linearization and deep cuts, and solve (R)
3. Round
4. Solve the convex NLP subproblem ($\mathsf{C}_{\mathrm{nlp}}[U_{y,B}]$)
5. Solve the nonconvex NLP subproblem ($\mathsf{P}[U_{y,B}]$)
6. Switch some binary variables and repeat

Optimal design of complex energy conversion systems (DFG-project)

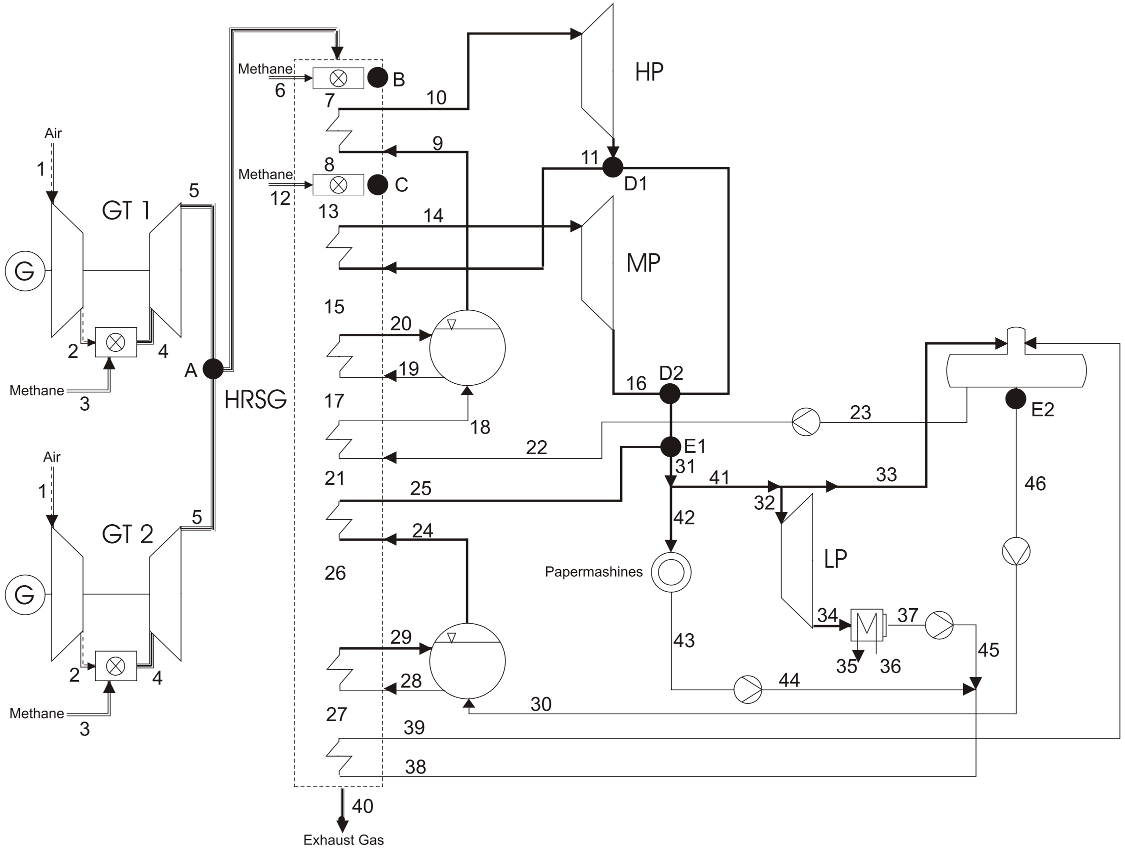


Figure 1: Simple superstructure of the cogeneration plant

Minimize: *Total levelized costs per time unit*
Subject to: constraints referring to *plant components, material properties, investment, operating and maintenance cost and economic analysis*

- Size: 508 variables and 461 constraints, $p = 172$ blocks with $\max |J_k| = 47$ (coded in `AMPL`)

- Difficulty: some functions have singularities in $[\underline{x}, \overline{x}]$ (contrained sampling)

- Lower bound: 5547.13 Euro/h

- Rounding heuristic: 6090.80 Euro/h,

- Best solution: 5995.83 Euro/h (difference of 1.6%)

Medium size MINLPs from MinlpLib (GAMS)

Data:

- 26 problems
- up to 57 variables 74 constraints
- stop if more than 50 solution candidates

Results:

- solved 24 problems
- computing time: 18 problems in less than 3 sec. and 6 problems between 15 sec. and 6 min.

Branch-and-Cut Algorithms

The convexification center

Let $\hat{S}$ be the feasible set of a linear relaxation (R)
(including the level cut $c^T x + c_0 \leq \overline{v}$)

We call the analytic center x^c of $\hat{S}$ the convexification center of (R).

Since $\hat{S}$ is an outer approximation of $\mathrm{conv}(\mathrm{sol}(P_{\mathrm{ext}}))$ we have

$$x^c \simeq \mathrm{center}(\mathrm{conv}(\mathrm{sol}(P_{\mathrm{ext}}))$$

Central cuts

- central binary cut:
  branch w.r.t to the most violated binary variable:

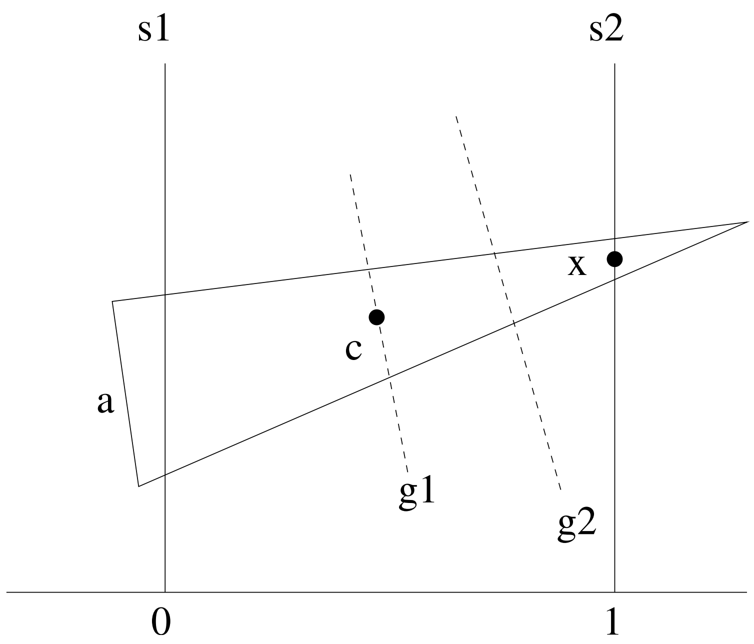
$$j = \operatorname*{argmin}_{i \in B} |x_i^c - 0.5(\underline{x}_i + \overline{x}_i)| = \operatorname*{argmax}_{i \in B} \operatorname{dist}(x_i^c, \{\underline{x}_i, \overline{x}_i\})$$

- central splitting cut
  separate x^e w.r.t the hyperplane

$$(x^c - x^e)^T((1-t)x^c + tx^e - x) = 0, \quad t \in (0,1)$$

- central diameter cut:
  subdivide w.r.t the hyperplane which goes through x^c and is parallel to the boundary-hyperplane with the largest distance to x^c

Illustration



- central binary cut: splitting into s_1 and s_2
- central splitting cut: subdivision at g_2
- central diameter cut: subdivision at g_1

A Branch-and-cut Algorithm

1. Get solution candidates obtained by a relaxation-based heuristic (deformation, rounding and partitioning) using the relaxations (R) and (C_{ext}).

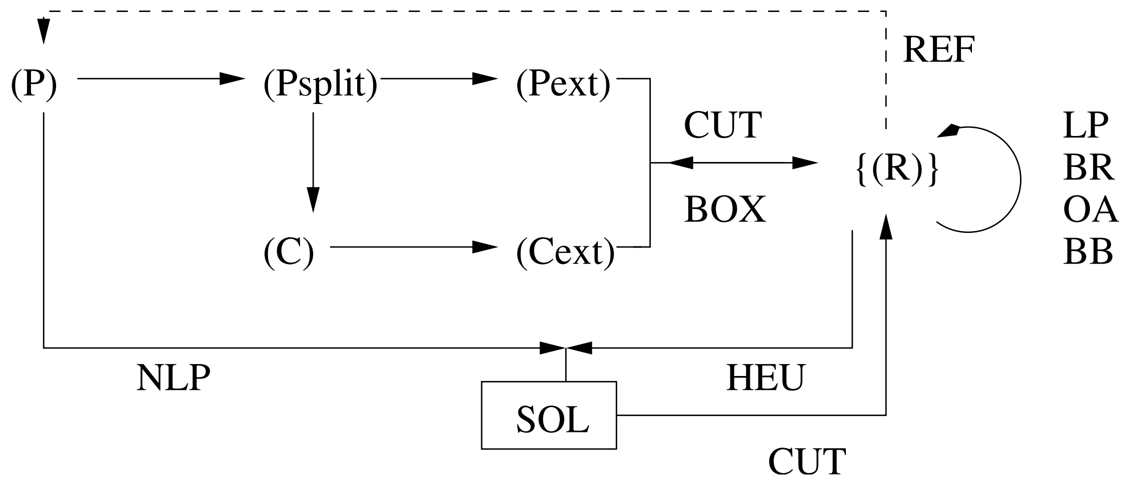
2. Improve the relaxation and the solution candidate by

   - Cuts:
     make linearization and valid cuts to improve (R) and (C_{ext})
   - Subdivision:
     make a central binary cut if a binary constraint is strongly violated
     else: make a central splitting cut if a local minimizer was found,
     else: make a central diameter cut
   - Lower bounds: take $\underline{v}(u) = \text{val}(C_{\text{ext}}[U])$ or $\underline{v}(U) = \text{val}(R[U])$

The C++ library LaGO (Lagrangian Global Optimizer)

- Input: AMPL, GAMS

- Basic components:
  (i) block-separable reformulation,
  (ii) convex relaxations (nonlinear, semidefinite and polyhedral),
  (iii) solution algorithms (deformation, rounding, partitioning, branch-and-cut)



Conclusion

- We presented a MINLP solution approach with the following features:
  - flexible decomposition through block-separable reformulations
  - convex relaxations of quadratic and black-box models
  - heuristics and a branch-and-cut method
- Preliminary results with LaGO
- Possible improvements through symbolic reformulations and interval arithmetic
- Future perspectives:
  - MINLP tends to be more important (Grossman/Biegler 02)
  - adaptive refinement of discretization of stochastic and optimal control programs via convex relaxations