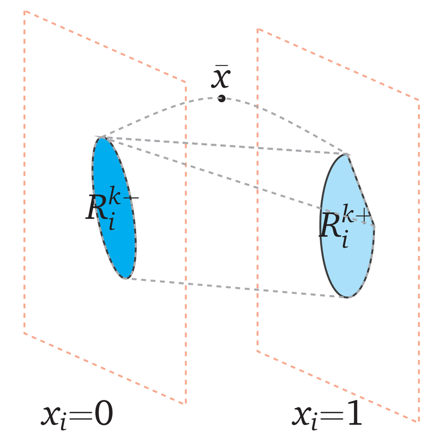
Effective Disjunctive Cuts for Convex Mixed Integer Nonlinear Programs

Mustafa Kılınç, Jeff Linderoth, and Jim Luedtke
University of Pittsburgh & University of Wisconsin-Madison



Convex MINLP

$$z_{\text{MINLP}} = \text{minimize } c^T x \quad \text{subject to } g(x) \le 0, \ \forall j \in J, \quad x \in X \overset{\text{def}}{=} \{x \mid Ax \le b\}, \quad x_I \in \mathbb{Z}^{|I|}$$

- $g : \mathbb{R}^n \to \mathbb{R}^m$ is a smooth, convex function

Disjunctive Cuts for MILP

- For Mixed Integer Linear Programs ($J = \emptyset$), disjunctive cutting planes (lift-and-project cuts) of Balas, Ceria, Cornuéjols are very effective computational tools
- Incorporated into all successful commercial MILP packages: CPLEX, Xpress-MP, Gurobi, etc.
- When used "aggressively" (to approximate rank 1 closure), lift-and-project cuts close significant optimality gap at the root node
  - MIPLIB 3.0: 63% on average
  - MIPLIB 2003: 44% on average

Key Research Quest

Can we effectively use disjunctive cuts in convex MINLP?

Disjunctive Cuts

- Continuous relaxation: $R = \{x \in X \mid g(x) \le 0\}$
- Disjunction on $x_i, i \in I$

$$R_i^{k-} = \{x \in R \mid x_i \le k\} \quad \text{and} \quad R_i^{k+} = \{x \in R \mid x_i \ge k+1\}$$

- Convex hull of union

$$\mathcal{R}_i^k = \text{conv}\,(R_i^{k-} \cup R_i^{k+})$$

- Extended Formulation

$$\mathcal{M}_i^k(R) = \left\{ (x, y, z, \lambda, \mu) \,\middle|\, \begin{array}{l} x = \lambda y + \mu z, \\ \lambda + \mu = 1, \quad \lambda \ge 0, \quad \mu \ge 0 \\ y \in R_i^{k-}, \quad z \in R_i^{k+} \end{array} \right\}$$

- Separation of $\bar{x} \notin \mathcal{R}_i^k$ [Stubbs and Mehrotra]

$$d_{\mathcal{M}_i^k}(\bar{x}) = \min_{(x,y,z,\lambda,\mu) \in \mathcal{M}_i^k(R)} d(x) \equiv \|x - \bar{x}\|$$

- Let $\hat{x}$ be an optimal solution of the separation problem
- Let $\xi \in \partial d(\hat{x})$
- $\xi^T(x - \hat{x}) \ge 0$ separates $\bar{x}$ from $\mathcal{R}_i^k$
- Bad News: Separation problem non-convex. ($x = \lambda y + \mu z$).

Convexification [Stubbs and Mehrotra]

- Homogenization; $\tilde{y} = \lambda y$ and $\tilde{z} = \mu z$
- Perspective function

$$\tilde{h}(\tilde{x}, \lambda) = \begin{cases} \lambda h(\tilde{x}/\lambda) & \text{if } \tilde{x}/\lambda \in C, \lambda > 0 \\ 0 & \text{if } \lambda = 0 \\ \infty & \text{otherwise} \end{cases} .$$

- Convex Reformulation

$$\tilde{\mathcal{M}}_i^k = \left\{ (x, \tilde{y}, \tilde{z}, \lambda, \mu) \,\middle|\, \begin{array}{ll} x = \tilde{y} + \tilde{z}, & \lambda + \mu = 1, \\ \tilde{g}(\tilde{y}, \lambda) \le 0, & \tilde{g}(\tilde{z}, \mu) \le 0 \\ A\tilde{y} \le \lambda b, & A\tilde{z} \le \mu b, \\ \tilde{y}_i \le \lambda k, & \tilde{z}_i \ge \mu k + \mu, \\ \lambda \ge 0, & \mu \ge 0 \end{array} \right\}$$

- Problem is non-differentiable (at $\lambda = 0$)
- It is twice as big a NLP to solve, just to generate a cut
- In practice: Only able to generate cuts for toy problems

Idea: Avoid solving NLPs

- Let $\mathcal{F} \supseteq \text{conv}\,(R_i^{k-} \cup R_i^{k+})$ be a polyhedral relaxation such that $\bar{x} \notin \mathcal{F}$ and try to separate $\bar{x}$ from $\mathcal{F}$
- Start with a polyhedral relaxation of $\text{conv}\,(R_i^{k-} \cup R_i^{k+})$ and iteratively improve relaxation to get stronger cuts.

A LP-Based MINLP Disjunctive Cut Framework

- Polyhedral relaxations of R_i^{k-} and R_i^{k+}

$$\mathcal{F}_-^t \equiv \{x \in X \mid x_i \le k, \ g(\bar{x}) + \nabla g(\bar{x})^T(x - \bar{x}) \le 0, \ \forall \bar{x} \in \mathcal{K}_-^t\}$$
$$\mathcal{F}_+^t \equiv \{x \in X \mid x_i \ge k+1, \ g(\bar{x}) + \nabla g(\bar{x})^T(x - \bar{x}) \le 0, \ \forall \bar{x} \in \mathcal{K}_+^t\}$$

- Extended Formulation

$$\tilde{\mathcal{M}}^t(\mathcal{F}) = \left\{ (x, \tilde{y}, \tilde{z}, \lambda, \mu) \,\middle|\, \begin{array}{ll} x = \tilde{y} + \tilde{z}, & \lambda + \mu = 1, \\ \lambda g(\bar{x}) + \nabla g(\bar{x})^T(\tilde{y} - \lambda\bar{x}) \le 0, & \forall \bar{x} \in \mathcal{K}_-^t \\ \lambda g(\bar{x}) + \nabla g(\bar{x})^T(\tilde{z} - \lambda\bar{x}) \le 0, & \forall \bar{x} \in \mathcal{K}_+^t \\ A\tilde{y} \le \lambda b, & A\tilde{z} \le \mu b, \\ \tilde{y}_i \le \lambda k, & \tilde{z}_i \ge \mu k + \mu, \\ \lambda \ge 0, & \mu \ge 0 \end{array} \right\}$$

- Separation problem (an LP with an appropriate norm)

$$d_{MDP(t)}(\bar{x}) = \min_{(x^t, \tilde{y}^t, \tilde{z}^t, \lambda^t, \mu^t) \in \widetilde{\mathcal{M}}^t(\mathcal{F})} d(x^t) = \|x^t - \bar{x}\|$$

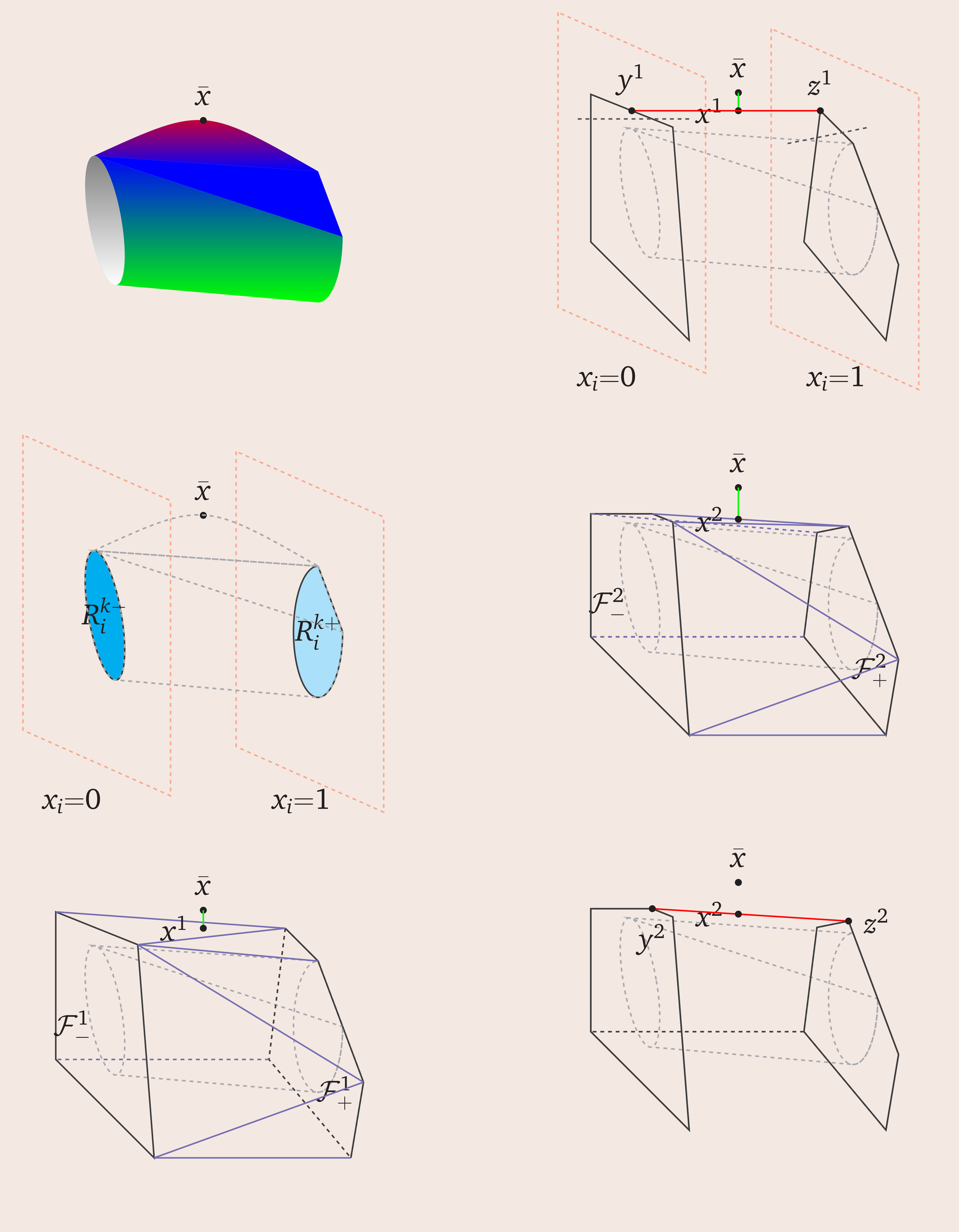
Theorem: Convergence of the algorithm

If $\mathcal{K}_-^t, \mathcal{K}_+^t$ are updated according to our algorithm, then

$$\lim_{t \to \infty} d_{MDP(t)}(\bar{x}) \to d_{\mathcal{M}_i^k}(\bar{x})$$

- Our cut is same strength as Stubbs and Mehrotra cut
- In practice, can stop iterating early

How to update $\mathcal{K}_-^t$ and $\mathcal{K}_+^t$?

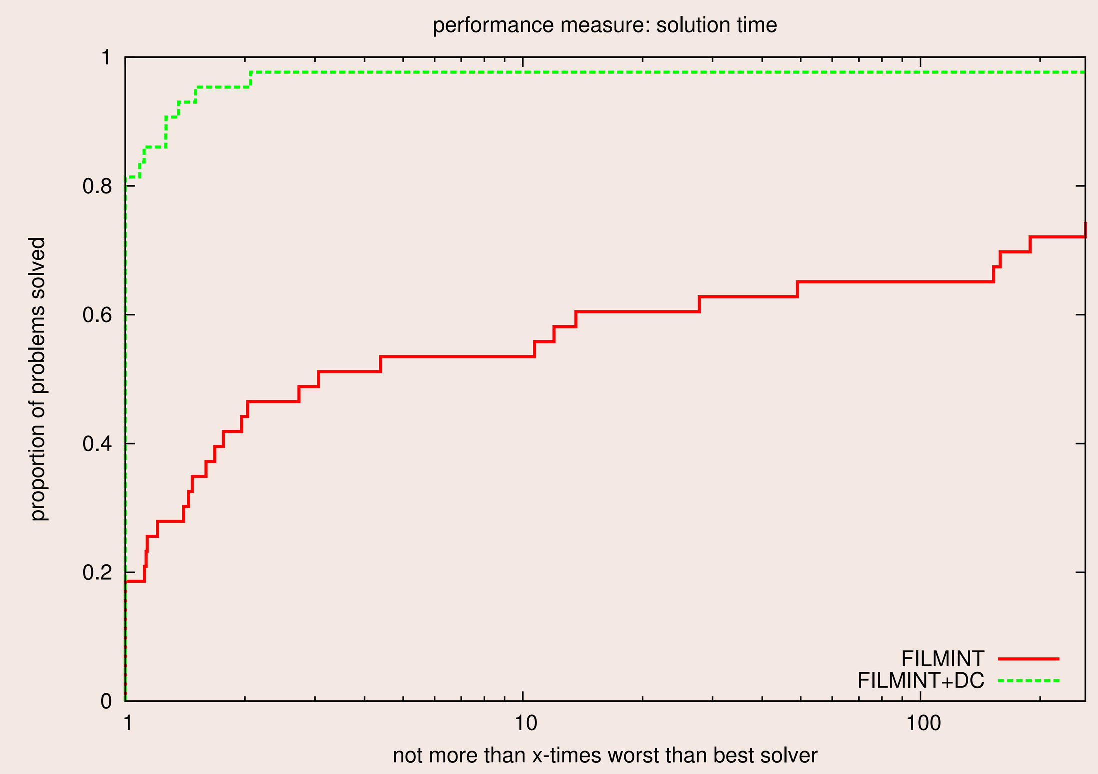


Root Gap Closed

Instance Family	# of ins	% Gap Closed min	ave	max	Hit limit
Batch	10	52.7	58.4	66.3	6
CLay	12	7.5	40.8	72.7	-
FLay	10	28.72	50.7	100.0	1
fo-m-o	9	0.0	2.2	19.6	1
MV	10	0.0	0.0	0.0	10
nd	5	73.0	85.0	93.0	-
RSyn	48	60.5	88.5	100.0	-
safetyLay	3	100.0	100.0	100.0	-
SLay	14	35.5	68.5	86.8	4
sssd	14	99.4	99.7	99.8	-
Syn	48	95.7	99.3	100.0	-
trimloss	12	0	6.4	14.4	4
uflquad	15	0.0	10.9	19.6	9
others	12	0.0	47.1	100.0	4
Total	222	0.0	65.3	100.0	39

It Works! Performance Profile (Time)

- On 59 "hard" instances (excluding MV)
- Solve using FilMINT: Linearization-based solver



Making $$$ — Exploiting Separability on MV

$$\begin{array}{l} \min \eta \\ x^T Q x \le \eta \\ e^T x = 1 \\ \ell_j z_j \le x_j \le u_j z_j \\ z_j \in \{0,1\} \end{array} \qquad \overset{Q = LL^T}{\longrightarrow} \qquad \begin{array}{l} \min \sum_{j=1}^n t_j \\ y_j^2 \le t_j \\ y = L^T x \\ e^T x = 1 \\ \ell_j z_j \le x_j \le u_j z_j \\ z_j \in \{0,1\} \end{array}$$

Separability and Disjunctive Cut Closure

Instance Family	# of ins	Original Form. Ave. gap closed	Original Form. Hit limit	Reformulation Ave. gap closed	Reformulation Hit limit
Batch	10	58.4	6	68.8	0
MV	10	0	10	98.1	0
Slay	14	68.5	4	86.1	0
uflquad	15	10.9	9	96.3	0

Separable Problems: Avg CPU Time

Instance Family	Original formulation FilMINT	Original formulation FilMINT+DC	Reformulation FilMINT	Reformulation FilMINT+DC
BatchS	20	376	25	59
MV	10800	10800	10800	1263
Slay	18	36	1	5
uflquad	639	785	502	145

Support from (DE-FG02-08ER25861): Next Generation Solvers for Mixed Integer Nonlinear Programs: Structure, Search, and Implementation