A DISJUNCTIVE CUT STRENGTHENING TECHNIQUE FOR CONVEX MINLP

Jan Kronqvist and Ruth Misener

Imperial College London

Acknowledgements

Research funded by

- Kronqvist, Newton International Fellowship by the Royal Society (Grant No. NIF\R1\182194).
- Kronqvist, Grant by the Swedish Cultural Foundation in Finland.
- Misener, Research Fellowship by the Engineering and Physical Sciences Research Council (Grant No. EP/P016871/1)

THE ROYAL SOCIETY

Svenska kulturfonden

UKRI Engineering and Physical Sciences Research Council

Motivation The performance of solvers greatly depends on the strength of the linear and continuous relaxations.
– Weak relaxations → huge number of subproblems.

A new framework for constructing tight linear relaxations by utilizing disjunctive structures in the problem.

1. Obtain a valid inequality (cut).
2. Strengthen the cut over the convex hull of a disjunction.

Presentation based on

Kronqvist J. and Misener R. A disjunctive cut strengthening technique for convex MINLP, Optimization and Engineering, 2020.

– Other disjunctive techniques to derive strong cuts for convex MINLP

- Lodi A, Tanneau M, Vielma JP (2019), Disjunctive cuts for mixed-integer conic optimization. ArXiv:191203166
- Kılınç MR, Linderoth J, Luedtke J (2017) Lift-and-project cuts for convex mixed integer nonlinear programs. Mathematical Programming Computation
- Trespalacios F, Grossmann IE (2016), Cutting plane algorithm for convex generalized disjunctive programs. INFORMS Journal on Computing.
- Bonami P (2011), Lift-and-project cuts for mixed integer convex programs. In: IPCO 2011.
- Zhu Y, Kuno T (2006), A disjunctive cutting-plane-based branch-and-cut al-gorithm for 0-1 mixed-integer convex nonlinear programs. Industrial Engineering Chemistry Research
- Stubbs RA, Mehrotra S (1999), A branch-and-cut method for 0-1 mixed convex programming. Mathematical Programming.

Main difference with our approach: we don't use the convex hull formulation of disjunctions.

Table of Contents

Background

The MINLP problem scope

Convex MINLP problems can be formulated as

$$\text{find } x^* \in \underset{x \in C \cap L \cap Y}{\arg\min}\; c^T x \tag{P}$$

where the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \in \mathbb{R}^n \mid g_m(x) \le 0,\ m = 1, \dots, M\} \\ L &= \{x \in \mathbb{R}^n \mid Ax \le a,\ Bx = b\} \\ Y &= \{x \in \mathbb{R}^n \mid x_i \in \mathbb{Z},\ i \in I_{\mathbb{Z}}\} \end{aligned}$$

and g_m are convex functions.

The MINLP problem scope

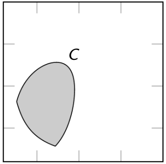
Convex MINLP problems can be formulated as

$$\text{find } x^* \in \underset{x \in C \cap L \cap Y}{\arg\min}\; c^T x \tag{P}$$

where the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \in \mathbb{R}^n \mid g_m(x) \le 0,\; m = 1, \ldots, M\} \\ L &= \{x \in \mathbb{R}^n \mid Ax \le a,\; Bx = b\} \\ Y &= \{x \in \mathbb{R}^n \mid x_i \in \mathbb{Z},\; i \in I_{\mathbb{Z}}\} \end{aligned}$$

and g_m are convex functions.



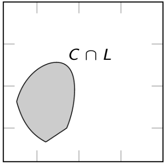
The MINLP problem scope

Convex MINLP problems can be formulated as

$$\text{find } x^* \in \underset{x \in C \cap L \cap Y}{\arg\min} \; c^T x \tag{P}$$

where the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \in \mathbb{R}^n \mid g_m(x) \leq 0, \; m = 1, \dots, M\} \\ L &= \{x \in \mathbb{R}^n \mid Ax \leq a, \; Bx = b\} \\ Y &= \{x \in \mathbb{R}^n \mid x_i \in \mathbb{Z}, \; i \in I_{\mathbb{Z}}\} \end{aligned}$$



and g_m are convex functions.

The MINLP problem scope

Convex MINLP problems can be formulated as

$$\text{find } x^* \in \underset{x \in C \cap L \cap Y}{\arg\min} \; c^T x \tag{P}$$

where the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \in \mathbb{R}^n \mid g_m(x) \leq 0, \; m = 1, \ldots, M\} \\ L &= \{x \in \mathbb{R}^n \mid Ax \leq a, \; Bx = b\} \\ Y &= \{x \in \mathbb{R}^n \mid x_i \in \mathbb{Z}, \; i \in I_{\mathbb{Z}}\} \end{aligned}$$

and g_m are convex functions.

The extended supporting hyperplane algorithm

We use cuts generated by the ESH algorithm.

The ESH algorithm is described in

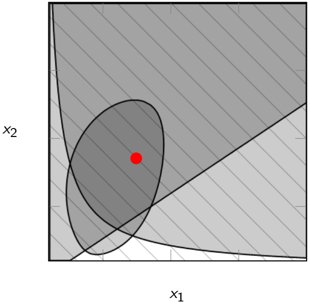
Kronqvist J., Lundell A. and Westerlund T. The extended supporting hyperplane algorithm for convex MINLP problems, Journal of Global Optimization, 2016.

Main idea

Construct an equivalent MILP representation of the MINLP by generating supporting hyperplanes to the feasible set.

First step (initialization)

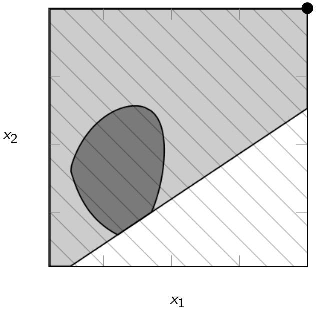
Find a strictly feasible solution to the continuous relaxation of the MINLP.



Can be formulated as a convex NLP.

ESH algorithm

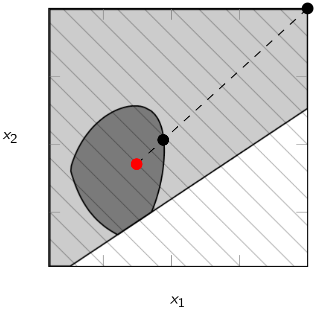
Solve linear relaxations to generate supporting hyperplanes.



Optimize the linear relaxation.

ESH algorithm

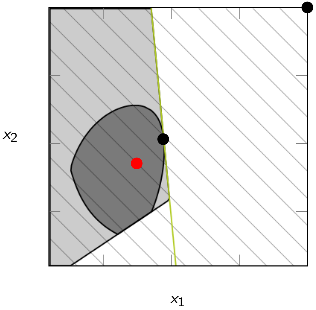
Solve linear relaxations to generate supporting hyperplanes.



The solution is approximately projected by a simple root search.

ESH algorithm

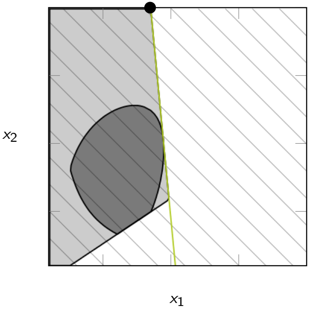
Solve linear relaxations to generate supporting hyperplanes.



Linearize the active nonlinear constraints.

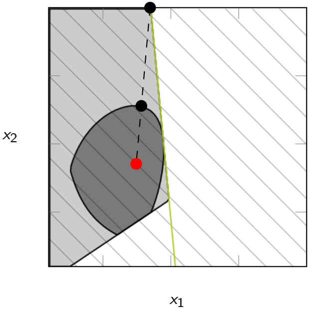
ESH algorithm

Solve linear relaxations to generate supporting hyperplanes.



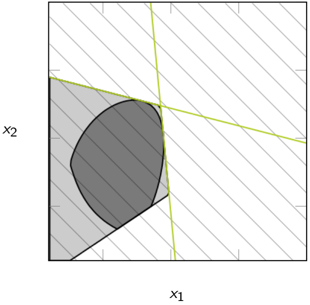
ESH algorithm

Solve linear relaxations to generate supporting hyperplanes.



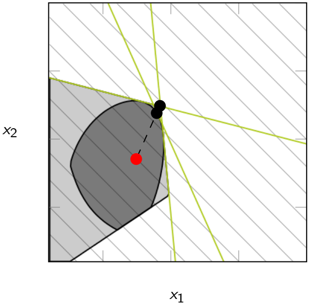
ESH algorithm

Solve linear relaxations to generate supporting hyperplanes.



ESH algorithm

Solve linear relaxations to generate supporting hyperplanes.



– Converges to a feasible and optimal solution.

J. Kronqvist 2020

Some remarks on the ESH algorithm

– The cuts generated by the ESH algorithm are
  - Tight with regards to the linear and nonlinear constraints.

Some remarks on the ESH algorithm

– The cuts generated by the ESH algorithm are

- Tight with regards to the linear and nonlinear constraints.
- Not tight when considering both the integrality restrictions and the linear/nonlinear constraints.
  - Possible to strengthen the cuts.

Illustrative example

Optimization task: find a point that minimizes the objective such that the point is in one of the three circles.

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & -x_1 - x_2 \\
\text{s.t.} \quad & (x_1-1)^2 + (x_2-2)^2 \leq 1 + 29.944(1-x_3), \\
& (x_1-2)^2 + (x_2-5)^2 \leq 1 + 29.944(1-x_4), \\
& (x_1-4)^2 + (x_2-1)^2 \leq 1 + 29.944(1-x_5), \\
& x_3 + x_4 + x_5 = 1, \\
& 0 \leq x_1 \leq 8, \; 0 \leq x_2 \leq 8, \\
& x_1, x_2 \in \mathbb{R}, \; x_3, x_4, x_5 \in \{0,1\}.
\end{aligned} \tag{EX1}
$$

Big-M formulation.

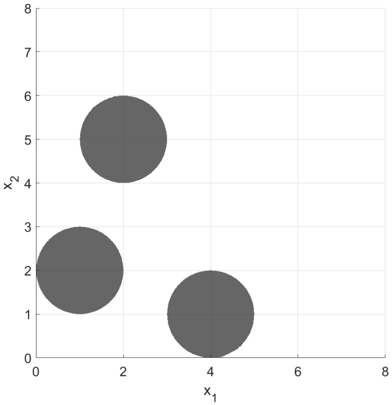


Figure: Feasible set of problem (EX1)

Illustrative example

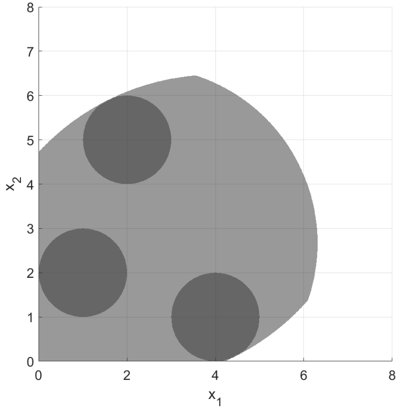


Figure: Feasible set of the continuous relaxation of problem (EX1)

Illustrative example

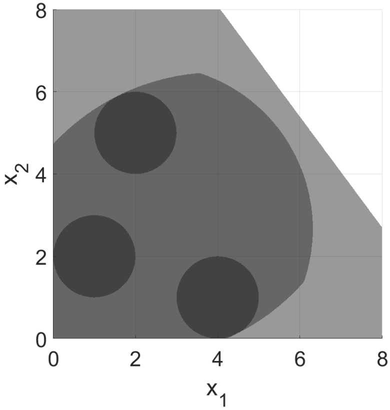


Figure: Feasible set of the continuous relaxation of problem (EX1)

Iteration 1: ESH gives the cut

$$5.920x_1 + 4.536x_2 + 29.944x_3 \leq 59.249.$$

Illustrative example

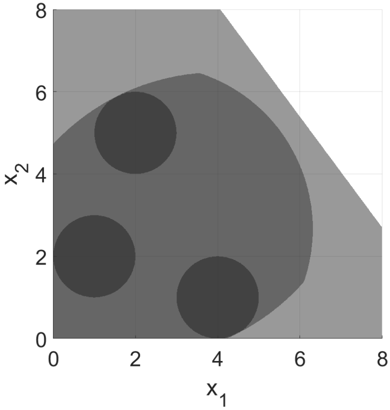


Figure: Feasible set of the continuous relaxation of problem (EX1)

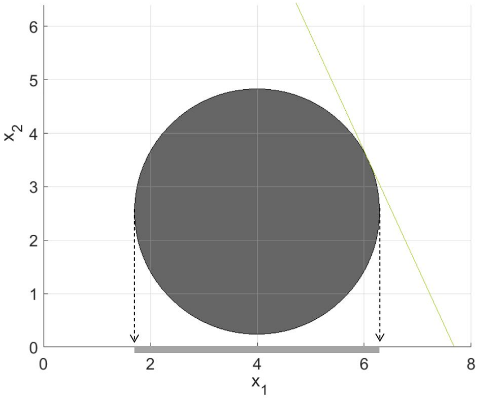


Figure: Note that the cut does not form a supporting hyperplane in the projected space

Illustrative example

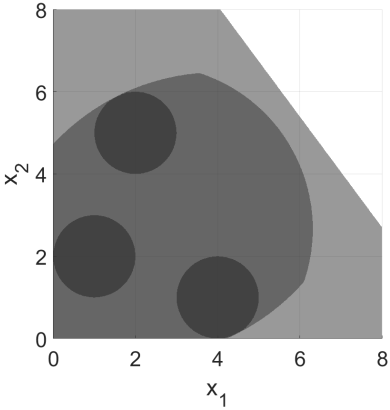


Figure: Feasible set of the continuous relaxation of problem (EX1)

Iteration 1: ESH gives the cut

$5.920x_1 + 4.536x_2 + 29.944x_3 \leq 59.249.$

We can strengthen the cut by reducing the right-hand side (RHS) value.

– How to determine the smallest valid RHS?

Illustrative example

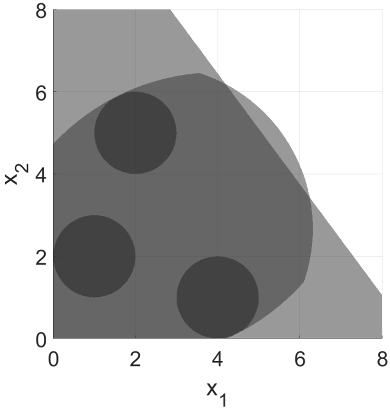


Figure: Feasible set of the continuous relaxation of problem (EX1)

Iteration 1: ESH gives the cut

$5.920x_1 + 4.536x_2 + 29.944x_3 \leq 59.249.$

We can strengthen the cut by reducing the right-hand side (RHS) value.

– How to determine the smallest valid RHS?

Definition 1: Tighter cut

We say that that the first cut **is tighter** than the second if all $\mathbf{x} \in X$ that satisfies inequality (1) also satisfies inequality (2) but not vice versa

$$\alpha_1^\top \mathbf{x} \leq \beta_1, \quad (1)$$

$$\alpha_2^\top \mathbf{x} \leq \beta_2. \quad (2)$$

Disjunctive cut strengthening

Problem structure

Assumption

The MINLP problem contains at least one **exclusive selection constraint** of binary variables, *i.e.*, $\exists\ I_D \subset I_{\mathbb{Z}} :\ x_i \in \{0,1\} \quad \forall i \in I_D$, and either one of the constraints

$$\sum_{i \in I_D} x_i = 1, \tag{3}$$

$$\sum_{i \in I_D} x_i \leq 1, \tag{4}$$

appears in the problem.

We only consider the first type of exclusive selection constraint, but the second type can easily be used by the same framework.

Cut strengthening

Given a valid cut $\alpha^\top \mathbf{x} \leq \beta$ and the index set I_D of an exclusive selection constraint,

Cut strengthening

Given a valid cut $\alpha^\top \mathbf{x} \leq \beta$ and the index set I_D of an exclusive selection constraint, we determine a reduced RHS-value for the cut by solving the disjunctive program

$$z^* = \max_{\mathbf{x}} \quad \alpha^\top \mathbf{x}$$
$$\text{s.t.} \quad \bigvee_{i \in I_D} \begin{bmatrix} \mathbf{x} \in N \cap L \\ x_i = 1 \\ x_j = 0 \quad \forall j \in I_D \setminus i \end{bmatrix}. \tag{5}$$

Cut strengthening

Given a valid cut $\alpha^\top \mathbf{x} \leq \beta$ and the index set I_D of an exclusive selection constraint, we determine a reduced RHS-value for the cut by solving the disjunctive program

$$z^* = \max_{\mathbf{x}} \quad \alpha^\top \mathbf{x} \qquad (5)$$

$$\text{s.t.} \quad \bigvee_{i \in I_D} \begin{bmatrix} \mathbf{x} \in N \cap L \\ x_i = 1 \\ x_j = 0 \quad \forall j \in I_D \setminus i \end{bmatrix}.$$

Theorem 1

The cut $\alpha^\top \mathbf{x} \leq z^*$ is a valid inequality for the MINLP problem and is at least as tight, or tighter, than the original cut.

Proof in: Kronqvist J. and Misener R. A disjunctive cut strengthening technique for convex MINLP, Optimization and Engineering, 2020.

We formulate the disjunctive program as $|I_D|$ convex NLP problems

$$
\begin{aligned}
z^* = \max_{i \in I_D} \quad b_i = \max_{\mathbf{x}} \alpha^\top \mathbf{x} & \\
\text{s.t.} \quad & \mathbf{x} \in N \cap L, \\
& x_i = 1, \\
& x_j = 0, \quad \forall j \in I_D \setminus i.
\end{aligned} \tag{6}
$$

$\iff$

Disjunctive formulation of the cut strengthening problem

$$
\begin{aligned}
z^* = \max_{\mathbf{x}} \quad & \alpha^\top \mathbf{x} \\
\text{s.t.} \quad & \bigvee_{i \in I_D} \begin{bmatrix} \mathbf{x} \in N \cap L \\ x_i = 1 \\ x_j = 0 \ \ \forall j \in I_D \setminus i \end{bmatrix}.
\end{aligned} \tag{5}
$$

We formulate the disjunctive program as $|I_D|$ convex NLP problems

$$z^* = \max_{i \in I_D} \quad b_i = \max_{\mathbf{x}} \alpha^\top \mathbf{x} \quad \text{s.t.} \quad \mathbf{x} \in N \cap L, \quad x_i = 1, \quad x_j = 0, \ \ \forall j \in I_D \setminus i. \tag{6}$$

$\iff$

Disjunctive formulation of the cut strengthening problem

$$z^* = \max_{\mathbf{x}} \quad \alpha^\top \mathbf{x} \quad \text{s.t.} \quad \bigvee_{i \in I_D} \begin{bmatrix} \mathbf{x} \in N \cap L \\ x_i = 1 \\ x_j = 0 \ \ \forall j \in I_D \setminus i \end{bmatrix}. \tag{5}$$

Further strengthen the cut

– Note that each b_i is a valid RHS-value for the corresponding partial integer assignment.

– A valid cut is given by

$$\alpha^\top \mathbf{x} \leq \sum_{i \in I_D} b_i x_i. \tag{7}$$

We formulate the disjunctive program as $|I_D|$ convex NLP problems

$$z^* = \max_{i \in I_D} \quad b_i = \max_{\mathbf{x}} \alpha^\top \mathbf{x} \quad \text{s.t.} \quad \mathbf{x} \in N \cap L, \quad x_i = 1, \quad x_j = 0, \quad \forall j \in I_D \setminus i. \tag{6}$$

$\iff$

Disjunctive formulation of the cut strengthening problem

$$z^* = \max_{\mathbf{x}} \quad \alpha^\top \mathbf{x} \quad \text{s.t.} \quad \bigvee_{i \in I_D} \begin{bmatrix} \mathbf{x} \in N \cap L \\ x_i = 1 \\ x_j = 0 \quad \forall j \in I_D \setminus i \end{bmatrix}. \tag{5}$$

Further strengthen the cut

– Note that each b_i is a valid RHS-value for the corresponding partial integer assignment.

– A valid cut is given by

$$\alpha^\top \mathbf{x} \leq \sum_{i \in I_D} b_i x_i. \tag{7}$$

Identify infeasible partial integer assignments

– If any of the NLP problems (6) are infeasible, then the variable fixed to one can be removed (permanently fixed to zero).

Two types of tightened cuts

By solving the convex NLP problems (6), we can determine two types of strengthened cuts.

Single tightening

$$\alpha^\top \mathbf{x} \leq z^* \tag{ST}$$

Multi tightening

$$\alpha^\top \mathbf{x} \leq \sum_{i \in I_D} b_i x_i \tag{MT}$$

Theorem 2

The cut given by (MT) is always as tight, or tighter, than the cut given by (ST).

Proof in: Kronqvist J. and Misener R. A disjunctive cut strengthening technique for convex MINLP, Optimization and Engineering, 2020.

Illustrative example

Back to our simple example:

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & -x_1 - x_2 \\
\text{s.t.} \quad & (x_1-1)^2 + (x_2-2)^2 \leq 1 + 29.944(1-x_3), \\
& (x_1-2)^2 + (x_2-5)^2 \leq 1 + 29.944(1-x_4), \\
& (x_1-4)^2 + (x_2-1)^2 \leq 1 + 29.944(1-x_5), \\
& x_3 + x_4 + x_5 = 1, \\
& 0 \leq x_1 \leq 8,\ 0 \leq x_2 \leq 8, \\
& x_1, x_2 \in \mathbb{R},\ x_3, x_4, x_5 \in \{0,1\}.
\end{aligned} \tag{EX1}
$$

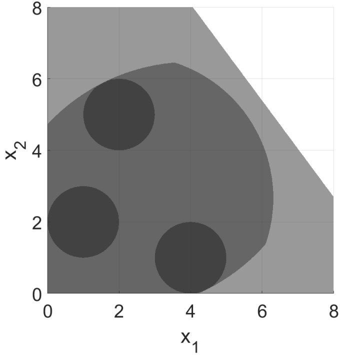
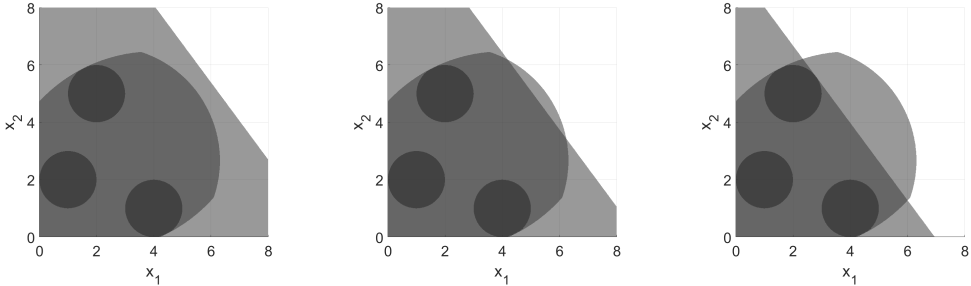


Figure: Original ESH cut.

Illustrative example

Illustration of the cuts



ESH cut

$5.92x_1 + 4.54x_2 + 29.94x_3 \leq 59.25$

ST cut

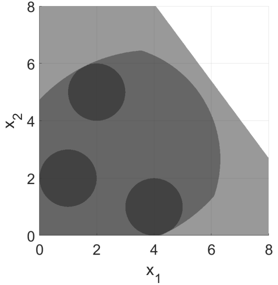
$5.92x_1 + 4.54x_2 + 29.94x_3 \leq 52.03$

MT cut

$5.92x_1 + 4.54x_2 \leq 22.09x_3 + 41.19x_4 + 35.45x_5$

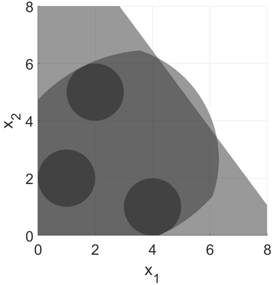
Illustrative example

Illustration of the cuts



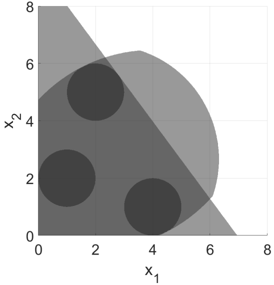
ESH cut

$5.92x_1 + 4.54x_2 + 29.94x_3 \leq 59.25$



ST cut

$5.92x_1 + 4.54x_2 + 29.94x_3 \leq 52.03$



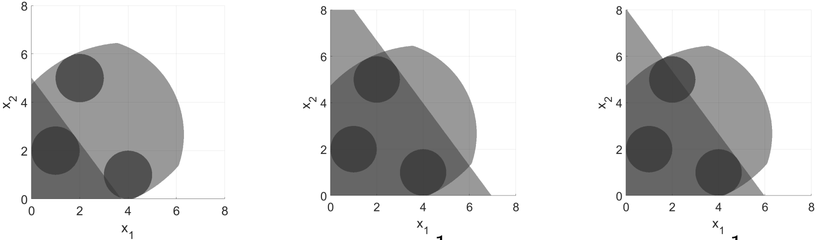
MT cut

$5.92x_1 + 4.54x_2 \leq 22.09x_3 + 41.19x_4 + 35.45x_5$

– The MT cut also improves the MILP relaxation.

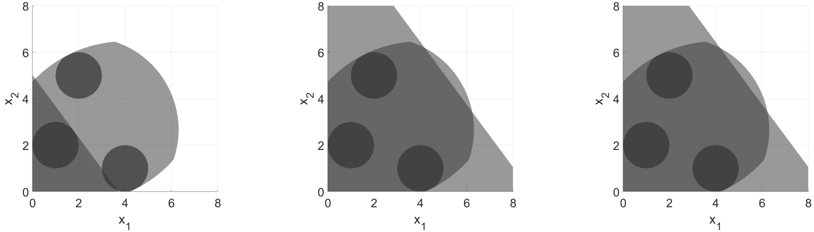
- For the MILP relaxation, the MT cut acts as a supporting hyperplane for the nonlinear constraints of each term of the disjunction.

MT cut for all integer assignments



$x_3 = 1$ $x_4 = 1$ $x_5 = 1$

ST cut for all integer assignments



$x_3 = 1$ $x_4 = 1$ $x_5 = 1$

Cut strengthening procedure

1. Obtain a cut by the ESH algorithm.

Cut strengthening procedure

1. Obtain a cut by the ESH algorithm.
2. Select an exclusive selection constraint.
   - Select the one with most variables interacting with the variables in the ESH cut, directly or through another constraint.

Cut strengthening procedure

1. Obtain a cut by the ESH algorithm.
2. Select an exclusive selection constraint.
   - Select the one with most variables interacting with the variables in the ESH cut, directly or through another constraint.
3. Solve problem (6) to obtain the RHS of the cut.
   - If any inner NLP problem is infeasible, then eliminate the corresponding binary variable form the MINLP problem.

Numerical results

Computational setup

Test set of 43 convex MINLP instances from MINLPLib[1].

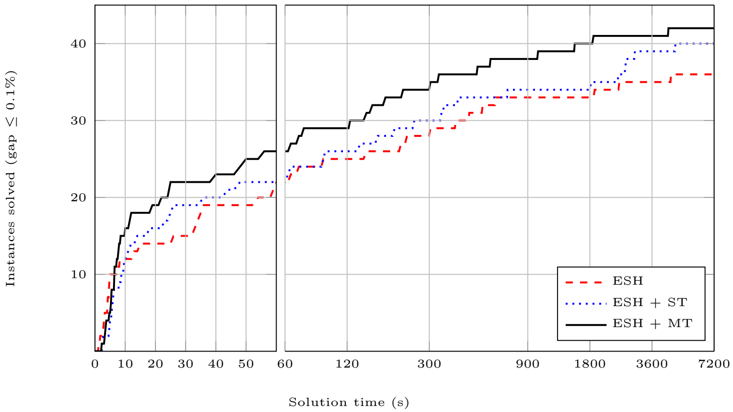
- 2 – 46 exclusive selection constraints.
- 2 – 40 disjunction size.
- 37 – 1060 variables (31 – 432 binary variables).
- 2 – 200 nonlinear constraints.

Basic implementation of the ESH algorithm with some primal heuristics[2].

- Basic ESH algorithm.
- ESH + ST cuts.
- ESH + MT cuts.

[1] http://www.minlplib.org

[2] Kronqvist J. and Misener R. A disjunctive cut strengthening technique for convex MINLP, Optimization and Engineering, 2020.



- Number of instances solved as a function of solution time.
- Solved in 2 hours: ESH 36, ESH + ST 40, and ESH + MT 42.
- ST cuts reduce time by 15% and number of iterations by 33%.
- MT cuts reduce time by 56% and number of iterations by 59%.

More detailed results

	ESH			ESH + ST			ESH + MT		
Instance	Iter.	Time	Time/iter.	Iter	Time	Time/iter.	Iter.	Time	Time/iter.
p_ball_15b_5p_2d	389	144.7s	**0.37s**	209	341.4s	**1.63s**	51	73.1s	**1.43s**

– The cut strengthening requires additional computations in each iteration.

More detailed results

	ESH			ESH + ST			ESH + MT		
Instance	Iter.	Time	Time/iter.	Iter	Time	Time/iter.	Iter.	Time	Time/iter.
p_ball_15b_5p_2d	389	144.7s	0.37s	209	341.4s	1.63s	51	73.1s	1.43s
p_ball_10b_5p_3d	**491**	543.4s	1.11s	**185**	168.7s	0.91s	**60**	48.0s	0.80s
p_ball_10b_5p_4d	**879**	2496.4s	2.84s	**265**	410.1s	1.55s	**115**	122.8s	1.07s

– The cut strengthening greatly reduces the number of iterations.

More detailed results

	ESH			ESH + ST			ESH + MT		
Instance	Iter.	Time	Time/iter.	Iter	Time	Time/iter.	Iter.	Time	Time/iter.
p_ball_15b_5p_2d	389	144.7s	0.37s	209	341.4s	1.63s	51	73.1s	1.43s
p_ball_10b_5p_3d	491	543.4s	1.11s	185	168.7s	0.91s	60	48.0s	0.80s
p_ball_10b_5p_4d	879	2496.4s	2.84s	265	410.1s	1.55s	115	122.8s	1.07s
slay10m	420	4432.5s	**10.55s**	105	203.1s	**1.93s**	109	219.2s	**2.01s**
stockcycle	>1205	>96h	**286.80s**	>3901	>96h	**88.59s**	2910	36.1h	**44.66s**

– The cut strengthening procedure can result in easier subproblems.

- In stockcycle we are able to eliminate 299 of the 432 binary variables.
- In slay10m we are able to eliminate 53 of the 180 binary variables.
- Smaller linear subproblems with a tighter continuous relaxation.

Summary

- Presented a new framework for strengthening cuts over disjunctive structures.
  - Based on the ESH algorithm, but can also be used with other techniques.
- Can greatly reduce both the number of iterations and time needed to solve convex MINLP problems.
- A new set of nonlinear disjunctive test problems github.com/jkronqvi/points_in_circles/.
- The paper "A disjunctive cut strengthening technique for convex MINLP" is available as open access doi.org/10.1007/s11081-020-09551-6.

Summary

- Presented a new framework for strengthening cuts over disjunctive structures.
  - Based on the ESH algorithm, but can also be used with other techniques.
- Can greatly reduce both the number of iterations and time needed to solve convex MINLP problems.
- A new set of nonlinear disjunctive test problems github.com/jkronqvi/points_in_circles/.
- The paper "A disjunctive cut strengthening technique for convex MINLP" is available as open access doi.org/10.1007/s11081-020-09551-6.

Thank you for your attention!