$$\overbrace{\textbf{ESH}}^{\text{EXTENDED SUPPORTING HYPERPLANE}}$$

THE ESH ALGORITHM FOR CONVEX MINLP

INFORMS Annual Meeting 2014
San Francisco, USA

Andreas Lundell
Jan Kronqvist
Tapio Westerlund

Optimization and Systems Engineering
Åbo Akademi University, Finland

CONTENTS OF THE TALK

The extended cutting plane (ECP) algorithm.

CONTENTS OF THE TALK

The extended cutting plane (ECP) algorithm.

The extended supporting hyperplane (ESH) algorithm:

- Solves sequences of approximative LP and MILP problems.
- Cutting planes are replaced with supporting hyperplanes generated on the boundary of the feasible set.
- Two LP preprocessing steps are utilized to quickly get a tight linear relaxation of the feasible set.

CONTENTS OF THE TALK

The extended cutting plane (ECP) algorithm.

The extended supporting hyperplane (ESH) algorithm:

- Solves sequences of approximative LP and MILP problems.
- Cutting planes are replaced with supporting hyperplanes generated on the boundary of the feasible set.
- Two LP preprocessing steps are utilized to quickly get a tight linear relaxation of the feasible set.

Comparison of a COIN-OR ESH implementation to other solvers.

THE ECP ALGORITHM

THE EXTENDED CUTTING PLANE ALGORITHM

The ECP algorithm is applicable to generally convex mixed-integer nonlinear programming (MINLP) problems.

THE EXTENDED CUTTING PLANE ALGORITHM

The ECP algorithm is applicable to generally convex mixed-integer nonlinear programming (MINLP) problems.

Solves MILP relaxations of the MINLP problem where the nonlinear constraints are approximated using cutting planes.

THE EXTENDED CUTTING PLANE ALGORITHM

The ECP algorithm is applicable to generally convex mixed-integer nonlinear programming (MINLP) problems.

Solves MILP relaxations of the MINLP problem where the nonlinear constraints are approximated using cutting planes.

Implemented, *e.g.*, in the AlphaECP solver in GAMS and available on the NEOS server.

1960 Convex NLP problems

Kelley Jr. J. The cutting-plane method for solving convex programs, Journal of the SIAM, vol. 8(4), 703–712.

1995 Convex MINLP problems

Westerlund T. and Pettersson F., An extended cutting plane method for solving convex MINLP problems, Computers & Chemical Engineering 19, 131–136.

1960 Convex NLP problems

Kelley Jr. J. The cutting-plane method for solving convex programs, Journal of the SIAM, vol. 8(4), 703–712.

1995 Convex MINLP problems

Westerlund T. and Pettersson F., An extended cutting plane method for solving convex MINLP problems, Computers & Chemical Engineering 19, 131–136.

1998 Pseudoconvex constraints

Westerlund T., Skrifvars H., Harjunkoski I. and Pörn R. An extended cutting plane method for solving a class of non-convex MINLP problems. Computers and Chemical Engineering, 22, 357–365.

2000 Pseudoconvex objective function and constraints

Pörn R. and Westerlund T. A cutting plane method for minimizing pseudo-convex functions in the mixed-integer case. Computers and Chemical Engineering, 24, 2655–2665.

2013 Nonsmooth constraints

Eronen V.-P., Mäkelä M. M. and Westerlund T. On the generalization of ECP and OA methods to nonsmooth convex MINLP problems, Optimization, 63, 1057–1073.

EXTENSIONS

1998 Pseudoconvex constraints

Westerlund T., Skrifvars H., Harjunkoski I. and Pörn R. An extended cutting plane method for solving a class of non-convex MINLP problems. Computers and Chemical Engineering, 22, 357–365.

2000 Pseudoconvex objective function and constraints

Pörn R. and Westerlund T. A cutting plane method for minimizing pseudo-convex functions in the mixed-integer case. Computers and Chemical Engineering, 24, 2655–2665.

2013 Nonsmooth constraints

Eronen V.-P., Mäkelä M. M. and Westerlund T. On the generalization of ECP and OA methods to nonsmooth convex MINLP problems, Optimization, 63, 1057–1073.

EXTENSIONS

1998 Pseudoconvex constraints

Westerlund T., Skrifvars H., Harjunkoski I. and Pörn R. An extended cutting plane method for solving a class of non-convex MINLP problems. Computers and Chemical Engineering, 22, 357–365.

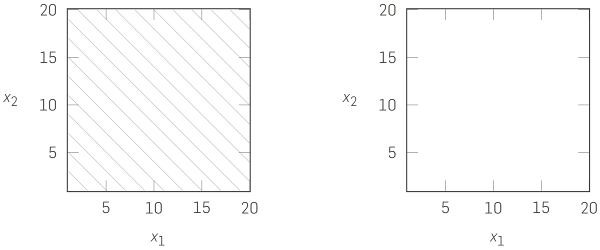
2000 Pseudoconvex objective function and constraints

Pörn R. and Westerlund T. A cutting plane method for minimizing pseudo-convex functions in the mixed-integer case. Computers and Chemical Engineering, 24, 2655–2665.

2013 Nonsmooth constraints

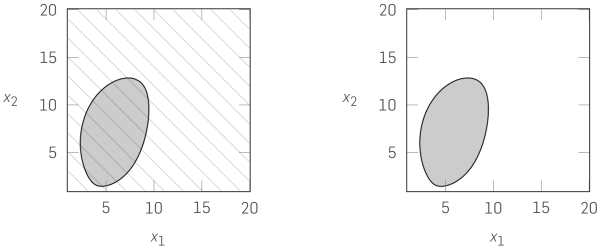
Eronen V.-P., Mäkelä M. M. and Westerlund T. On the generalization of ECP and OA methods to nonsmooth convex MINLP problems, Optimization, 63, 1057–1073.

AN ILLUSTRATIVE EXAMPLE



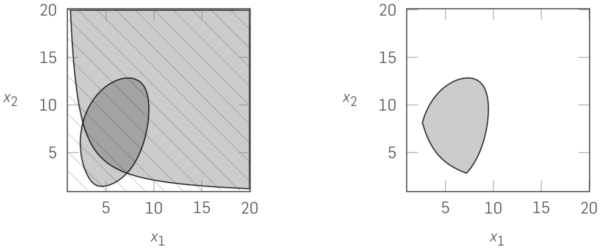
$$
\begin{aligned}
\text{minimize} \quad & c^T x = -x_1 - x_2 \\
\text{subject to} \quad & g_1(x_1, x_2) = 0.15(x_1 - 8)^2 + 0.1(x_2 - 6)^2 + 0.025 e^{x_1} x_2^{-2} - 5 \le 0 \\
& g_2(x_1, x_2) = 1/x_1 + 1/x2 - x_1^{0.5} x_2^{0.5} + 4 \le 0 \\
& 2x_1 - 3x_2 - 2 \le 0 \\
& 1 \le x_1 \le 20, \quad 1 \le x_2 \le 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}.
\end{aligned}
$$

AN ILLUSTRATIVE EXAMPLE



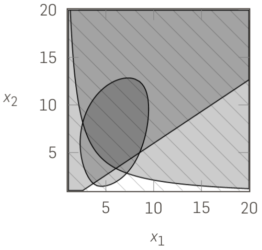
$$
\begin{aligned}
\text{minimize} \quad & c^T x = -x_1 - x_2 \\
\text{subject to} \quad & g_1(x_1, x_2) = 0.15(x_1 - 8)^2 + 0.1(x_2 - 6)^2 + 0.025 e^{x_1} x_2^{-2} - 5 \leq 0 \\
& g_2(x_1, x_2) = 1/x_1 + 1/x2 - x_1^{0.5} x_2^{0.5} + 4 \leq 0 \\
& 2x_1 - 3x_2 - 2 \leq 0 \\
& 1 \leq x_1 \leq 20, \quad 1 \leq x_2 \leq 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}.
\end{aligned}
$$

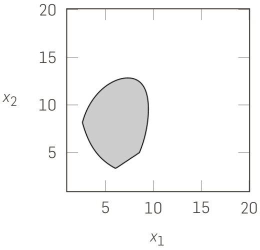
AN ILLUSTRATIVE EXAMPLE



$$
\begin{aligned}
\text{minimize} \quad & c^T x = -x_1 - x_2 \\
\text{subject to} \quad & g_1(x_1, x_2) = 0.15(x_1 - 8)^2 + 0.1(x_2 - 6)^2 + 0.025 e^{x_1} x_2^{-2} - 5 \leq 0 \\
& g_2(x_1, x_2) = 1/x_1 + 1/x2 - x_1^{0.5} x_2^{0.5} + 4 \leq 0 \\
& 2x_1 - 3x_2 - 2 \leq 0 \\
& 1 \leq x_1 \leq 20, \quad 1 \leq x_2 \leq 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}.
\end{aligned}
$$

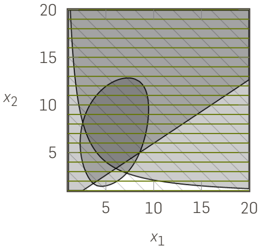
AN ILLUSTRATIVE EXAMPLE

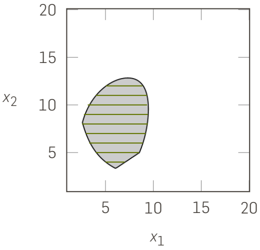




$$
\begin{aligned}
&\text{minimize} && c^T x = -x_1 - x_2 \\
&\text{subject to} && g_1(x_1, x_2) = 0.15(x_1 - 8)^2 + 0.1(x_2 - 6)^2 + 0.025 e^{x_1} x_2^{-2} - 5 \leq 0 \\
& && g_2(x_1, x_2) = 1/x_1 + 1/x2 - x_1^{0.5} x_2^{0.5} + 4 \leq 0 \\
& && 2x_1 - 3x_2 - 2 \leq 0 \\
& && 1 \leq x_1 \leq 20, \quad 1 \leq x_2 \leq 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}.
\end{aligned}
$$

AN ILLUSTRATIVE EXAMPLE

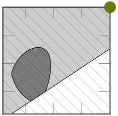




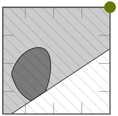
minimize	$c^T x = -x_1 - x_2$
subject to	$g_1(x_1, x_2) = 0.15(x_1 - 8)^2 + 0.1(x_2 - 6)^2 + 0.025e^{x_1} x_2^{-2} - 5 \leq 0$
	$g_2(x_1, x_2) = 1/x_1 + 1/x2 - x_1^{0.5} x_2^{0.5} + 4 \leq 0$
	$2x_1 - 3x_2 - 2 \leq 0$
	$1 \leq x_1 \leq 20, \quad 1 \leq x_2 \leq 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}.$

AN ILLUSTRATIVE EXAMPLE

In each iteration k of the ECP algorithm a MILP problem is solved to obtain the solution point (x_1^k, x_2^k).

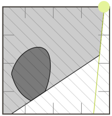


In each iteration k of the ECP algorithm a MILP problem is solved to obtain the solution point (x_1^k, x_2^k).



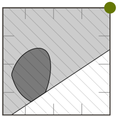
Select the constraint with the largest error $i = \arg\max g_i(x_1^k, x_2^k)$ and generate a cutting plane

$$g_i(x_1^k, x_2^k) + \nabla g_i(x_1^k, x_2^k)^T (x - x_1^k, x - x_2^k) \leq 0.$$



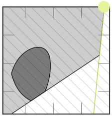
AN ILLUSTRATIVE EXAMPLE

In each iteration k of the ECP algorithm a MILP problem is solved to obtain the solution point (x_1^k, x_2^k).



Select the constraint with the largest error $i = \arg\max g_i(x_1^k, x_2^k)$ and generate a cutting plane

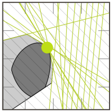
$$g_i(x_1^k, x_2^k) + \nabla g_i(x_1^k, x_2^k)^T (x - x_1^k, x - x_2^k) \leq 0.$$



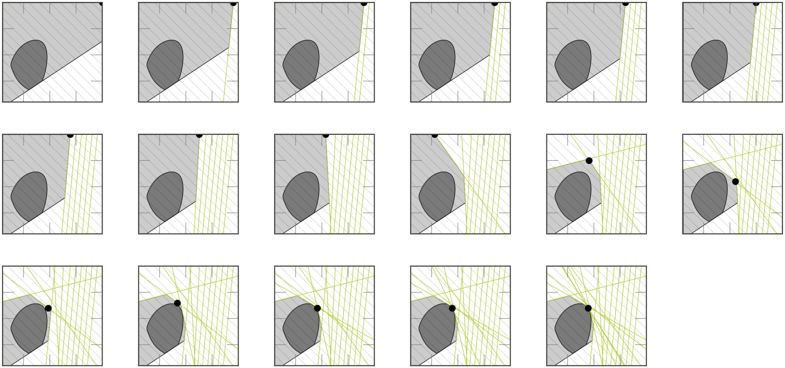
If the solution of the MILP problem is feasible for the MINLP problem *i.e.*,

$$g_i(x_1^k, x_2^k) \leq \epsilon \quad \forall i = 1, 2,$$

the optimal solution has been found.

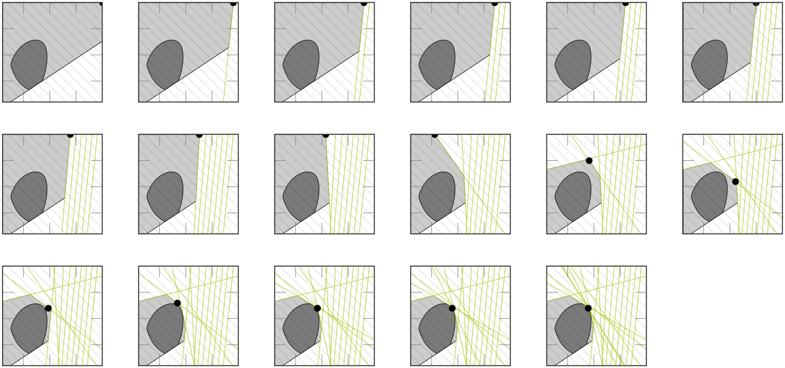


To solve the problem with the ECP algorithm ($\epsilon = 0.001$) it takes 17 iterations (17 MILP problems solved to optimality).



AN ILLUSTRATIVE EXAMPLE

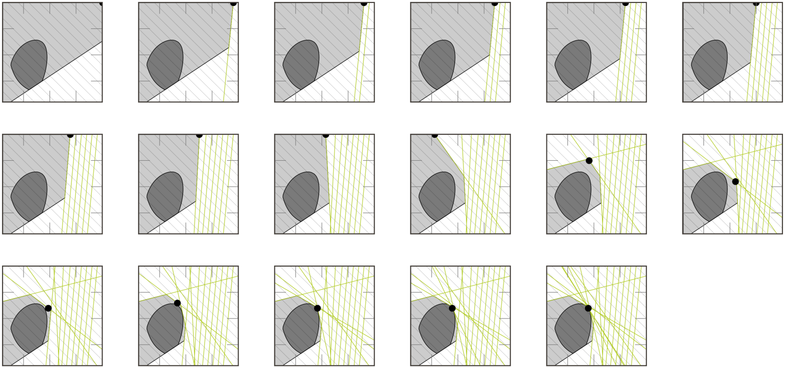
To solve the problem with the ECP algorithm ($\epsilon = 0.001$) it takes 17 iterations (17 MILP problems solved to optimality).



How can we improve performance?

AN ILLUSTRATIVE EXAMPLE

To solve the problem with the ECP algorithm ($\epsilon = 0.001$) it takes 17 iterations (17 MILP problems solved to optimality).



How can we improve performance?

Generate cutting planes on the boundary of the feasible set!

THE ESH ALGORITHM

THE EXTENDED SUPPORTING HYPERPLANE ALGORITHM

A new, stable and efficient algorithm for solving convex MINLP problems to global optimality.

THE EXTENDED SUPPORTING HYPERPLANE ALGORITHM

A new, stable and efficient algorithm for solving convex MINLP problems to global optimality.

Cutting planes replaced with supporting hyperplanes. Utilize a line search procedure to find the generation point. An interior point is required for the line search.

THE EXTENDED SUPPORTING HYPERPLANE ALGORITHM

A new, stable and efficient algorithm for solving convex MINLP problems to global optimality.

Cutting planes replaced with supporting hyperplanes. Utilize a line search procedure to find the generation point. An interior point is required for the line search.

Similar ideas as presented in:

The supporting hyperplane method for unimodal programming, Veinott Jr. A. F., Operations Research, Vol. 15, pp. 147–152, 1967

The ESH algorithm is briefly described in:

An extended supporting hyperplane algorithm for convex MINLP problems, Lundell A., Kronqvist J. and Westerlund T. Proceedings of XII Global Optimization Workshop, pp. 21–24, 2014

THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

$$x^* = \underset{x \in C \cap L \cap Y}{\arg\min}\, c^T x \tag{P}$$

where $x = [x_1, x_2, \dots, x_N]^T$ belongs to the compact set

$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \dots, N\} \subset \mathbb{R}^n,$$

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

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

and C is a convex set.

BREAKDOWN OF THE ESH ALGORITHM

NLP step

Obtain a feasible, relaxed interior point (satisfying C) by solving a NLP problem.

LP1 step (optional)

Solve simple LP problems (initially in X) to obtain supporting hyperplanes.

LP2 step (optional)

As in LP1 but also include the linear constraints in L.

MILP step

Solve MILP problems to find the optimal solution to (P).

BREAKDOWN OF THE ESH ALGORITHM

NLP step

Obtain a feasible, relaxed interior point (satisfying C) by solving a NLP problem.

LP1 step (optional)

Solve simple LP problems (initially in X) to obtain supporting hyperplanes.

LP2 step (optional)

As in LP1 but also include the linear constraints in L.

MILP step

Solve MILP problems to find the optimal solution to (P).

BREAKDOWN OF THE ESH ALGORITHM

NLP step

Obtain a feasible, relaxed interior point (satisfying C) by solving a NLP problem.

LP1 step (optional)

Solve simple LP problems (initially in X) to obtain supporting hyperplanes.

LP2 step (optional)

As in LP1 but also include the linear constraints in L.

MILP step

Solve MILP problems to find the optimal solution to (P).

BREAKDOWN OF THE ESH ALGORITHM

NLP step

Obtain a feasible, relaxed interior point (satisfying C) by solving a NLP problem.

LP1 step (optional)

Solve simple LP problems (initially in X) to obtain supporting hyperplanes.

LP2 step (optional)

As in LP1 but also include the linear constraints in L.

MILP step

Solve MILP problems to find the optimal solution to (P).

- A point in C is required as an endpoint for the line searches.

NLP STEP

- A point in C is required as an endpoint for the line searches.
- Assuming (P) has a solution, the internal point can be obtained from the NLP problem:

$$\begin{aligned} \tilde{x}_{\text{NLP}} &= \operatorname*{arg\,min}_{x \in X} F(x), \\ \text{where } \; F(x) &:= \max_{m=1,\dots,M} \{g_m(x)\}. \end{aligned} \tag{P-NLP}$$

NLP STEP

- A point in C is required as an endpoint for the line searches.
- Assuming (P) has a solution, the internal point can be obtained from the NLP problem:

$$\begin{aligned} \tilde{x}_{\mathrm{NLP}} &= \arg\min_{x \in X} F(x), \\ \text{where } F(x) &:= \max_{m=1,\dots,M} \{g_m(x)\}. \end{aligned} \tag{P-NLP}$$

- F is convex since it is the maximum of convex functions.

NLP STEP

- A point in C is required as an endpoint for the line searches.
- Assuming (P) has a solution, the internal point can be obtained from the NLP problem:

$$\begin{aligned} \tilde{x}_{\mathrm{NLP}} &= \arg\min_{x \in X} F(x), \\ \text{where } \; F(x) &:= \max_{m=1,\ldots,M} \{g_m(x)\}. \end{aligned} \tag{P-NLP}$$

- F is convex since it is the maximum of convex functions.
- (P-NLP) may be nonsmooth (if $M > 1$) even if g_m is smooth.

NLP STEP

- A point in C is required as an endpoint for the line searches.
- Assuming (P) has a solution, the internal point can be obtained from the NLP problem:

$$\begin{aligned} \tilde{x}_{\mathrm{NLP}} &= \arg\min_{x \in X} F(x), \\ \text{where } F(x) &:= \max_{m=1,\dots,M} \{g_m(x)\}. \end{aligned} \tag{P-NLP}$$

- F is convex since it is the maximum of convex functions.
- (P-NLP) may be nonsmooth (if $M > 1$) even if g_m is smooth.
- The point $\tilde{x}_{\mathrm{NLP}}$ need not be optimal but then fulfill $F(\tilde{x}_{\mathrm{NLP}}) < 0$.

LP1 STEP

- Starting from $k = 1,\ \Omega_0 = X$, the problem

$$\tilde{x}^k_{\mathrm{LP}} = \arg\min_{\Omega_{k-1}}\ c^T x \tag{P-LP1}$$

is repeatedly solved, and supporting hyperplanes (SHs)

$$l_k := F(x^k) + \nabla_F(x^k)^T(x - x^k) \leq 0$$

are generated and added to Ω_k.

LP1 STEP

- Starting from $k = 1,\ \Omega_0 = X$, the problem

$$\boxed{\tilde{x}^k_{\mathrm{LP}} = \arg\min_{\Omega_{k-1}}\ c^T x} \qquad \text{(P-LP1)}$$

  is repeatedly solved, and supporting hyperplanes (SHs)

$$l_k := F(x^k) + \nabla_F(x^k)^T(x - x^k) \leq 0$$

  are generated and added to Ω_k.
- A point x^k is obtained by a line search for $F(x^k) = 0$ between $\tilde{x}_{\mathrm{NLP}}$ and the solution to (P-LP1) $\tilde{x}^k_{\mathrm{LP}}$:

$$x^k = \lambda\tilde{x}_{\mathrm{NLP}} + (1 - \lambda)\tilde{x}^k_{\mathrm{LP}}, \quad \lambda \in [0, 1].$$

- If not $F(\tilde{x}^k_{\mathrm{LP}}) < \epsilon_{\mathrm{LP1}}$ or a maximum number of SHs have been generated, then k is increased and (P-LP1) resolved.

LP2 STEP

— This step is otherwise identical to LP1, with the exception that the linear constraints in L are now also included, *i.e.*,

$$\boxed{\tilde{x}_{\text{LP}}^{k} = \underset{\Omega_{k-1} \cap L}{\arg\min} \; c^T x} \qquad \text{(P-LP2)}$$

LP2 STEP

— This step is otherwise identical to LP1, with the exception that the linear constraints in L are now also included, *i.e.*,

$$\tilde{x}^k_{\mathrm{LP}} = \underset{\Omega_{k-1}\cap L}{\arg\min}\ c^T x \tag{P-LP2}$$

— (P-LP2) is repeatedly solved until $F(\tilde{x}^k_{\mathrm{LP}}) < \epsilon_{\mathrm{LP2}}$ or a maximum number of SHs have additionally been generated.

— In many cases the LP1 step may be omitted.

MILP STEP

- Finally, in order to also fulfill the integer requirements of problem (P), a MILP step is performed.

MILP STEP

- Finally, in order to also fulfill the integer requirements of problem (P), a MILP step is performed.
- This step is similar to LP2, but the integer requirements in Y are also included, *i.e.*,

$$\tilde{x}^k_{\mathrm{MILP}} = \underset{\Omega_{k-1} \cap L \cap Y}{\arg\min}\ c^T x. \tag{P-MILP}$$

MILP STEP

- Finally, in order to also fulfill the integer requirements of problem (P), a MILP step is performed.
- This step is similar to LP2, but the integer requirements in Y are also included, *i.e.*,

$$\boxed{\tilde{x}^k_{\mathrm{MILP}} = \underset{\Omega_{k-1} \cap L \cap Y}{\arg\min} \; c^T x.} \tag{P-MILP}$$

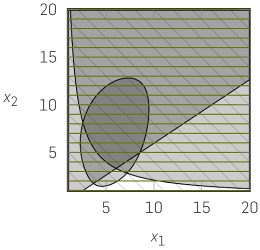
- (P-MILP) is repeatedly solved until $F(\tilde{x}^k_{\mathrm{MILP}}) < \epsilon_{\mathrm{MILP}}$.

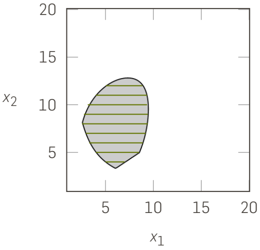
MILP STEP

- Finally, in order to also fulfill the integer requirements of problem (P), a MILP step is performed.
- This step is similar to LP2, but the integer requirements in Y are also included, *i.e.*,

$$\boxed{\tilde{x}^k_{\text{MILP}} = \underset{\Omega_{k-1} \cap L \cap Y}{\arg\min} \; c^T x.} \tag{P-MILP}$$

- (P-MILP) is repeatedly solved until $F(\tilde{x}^k_{\text{MILP}}) < \epsilon_{\text{MILP}}$.
- Intermediate (P-MILP) problems do not need to be solved to optimality, but in order to guarantee an optimal solution of (P), the final MILP solution must be optimal.





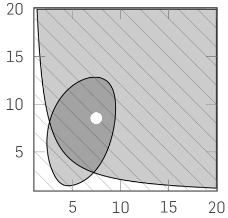
$$
\begin{aligned}
\text{minimize} \quad & c^T x = -x_1 - x_2 \\
\text{subject to} \quad & 0.15(x_1-8)^2 + 0.1(x_2-6)^2 + 0.025e^{x_1}x_2^{-2} - 5 \le 0 \\
& 1/x_1 + 1/x2 - x_1^{0.5}x_2^{0.5} + 4 \le 0 \\
& 2x_1 - 3x_2 - 2 \le 0 \\
& 1 \le x_1 \le 20, \quad 1 \le x_2 \le 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}
\end{aligned}
$$

NLP STEP – FIND AN INTERIOR POINT

$$\tilde{x}_{\mathrm{NLP}} = \operatorname*{arg\,min}_{(x_1,x_2)\in X} F(x_1, x_2),$$

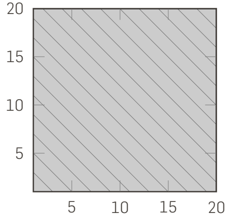
$$\text{where } F(x_1, x_2) := \max\{g_1(x_1, x_2),\ g_2(x_1, x_2)\}.$$

- The solution can be found using a suitable NLP solver
- Not required to be the optimal point
- The optimal point here is $(7.45, 8.54)$



LP1-STEP – ITERATION 1

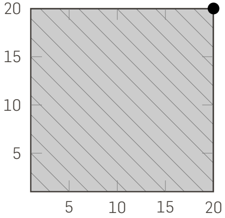
– Assume initially that $\Omega_0 = X$.



LP1-STEP – ITERATION 1

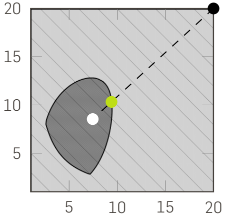
- Assume initially that $\Omega_0 = X$.
- $k = 1$, solve LP in Ω,

$$\tilde{x}^k_{\text{LP}} = \underset{\Omega_{k-1}}{\arg\min}\; c^T x.$$



- Assume initially that $\Omega_0 = X$.
- $k = 1$, solve LP in Ω,

$$\tilde{x}^k_{\text{LP}} = \arg\min_{\Omega_{k-1}} \; c^T x.$$

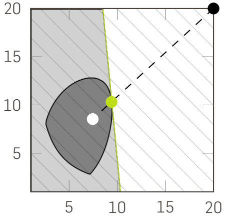


- Do line search

$$x^k = \lambda \tilde{x}_{\text{NLP}} + (1-\lambda)\tilde{x}^k_{\text{LP}}.$$

- Assume initially that $\Omega_0 = X$.
- $k = 1$, solve LP in Ω,

$$\tilde{x}_{\text{LP}}^k = \arg\min_{\Omega_{k-1}} \ c^T x.$$



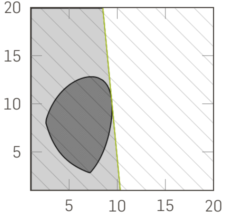
- Do line search

$$x^k = \lambda \tilde{x}_{\text{NLP}} + (1 - \lambda)\tilde{x}_{\text{LP}}^k.$$

- Generate supporting hyperplane in x^k and add to Ω.

– $\Omega_1 = \{x | l_1(x) \leq 0,\ x \in X\}$.

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$



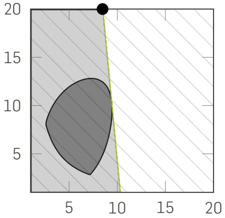
LP1-STEP – ITERATION 2

- $\Omega_1 = \{x | l_1(x) \leq 0,\ x \in X\}$.

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$

- $k = 2$, solve LP in Ω,

$$\tilde{x}^k_{\text{LP}} = \arg\min_{\Omega_{k-1}}\ c^T x.$$



LP1-STEP – ITERATION 2

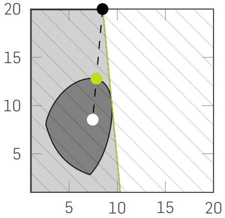
- $\Omega_1 = \{x | l_1(x) \leq 0,\ x \in X\}$.

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$

- $k = 2$, solve LP in Ω,

$$\tilde{x}^k_{\text{LP}} = \arg\min_{\Omega_{k-1}}\ c^T x.$$

- Do line search $x^k = \lambda \tilde{x}_{\text{NLP}} + (1-\lambda)\tilde{x}^k_{\text{LP}}$.

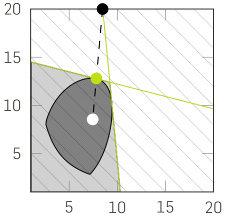


- $\Omega_1 = \{x | l_1(x) \leq 0,\ x \in X\}$.

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$

- $k = 2$, solve LP in Ω,

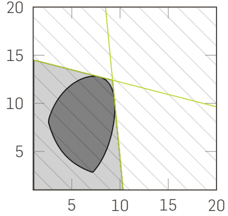
$$\tilde{x}^k_{\text{LP}} = \arg\min_{\Omega_{k-1}} \ c^T x.$$



- Do line search $x^k = \lambda \tilde{x}_{\text{NLP}} + (1 - \lambda)\tilde{x}^k_{\text{LP}}$.
- Generate supporting hyperplane in x^k and add to Ω.

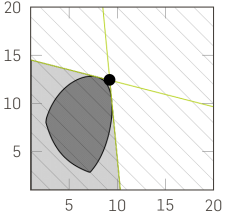
- $\Omega_2 = \{x | l_j(x) \leq 0,\ j \in \{1, 2\},\ x \in X\}$

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$
$$l_2(x) = 0.332x_1 + 1.30x_2 - 19.2$$



- $\Omega_2 = \{x | l_j(x) \leq 0,\ j \in \{1, 2\},\ x \in X\}$

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$
$$l_2(x) = 0.332x_1 + 1.30x_2 - 19.2$$



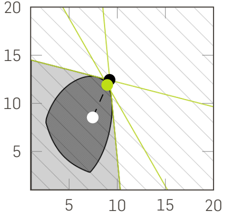
- $k = 3$, solve LP in Ω,

$$\tilde{x}_{\mathrm{LP}}^k = \arg\min_{\Omega_{k-1}}\ c^T x.$$

LP1-STEP – ITERATION 3

- $\Omega_2 = \{x | l_j(x) \leq 0,\ j \in \{1,2\},\ x \in X\}$

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$
$$l_2(x) = 0.332x_1 + 1.30x_2 - 19.2$$



- $k = 3$, solve LP in Ω,

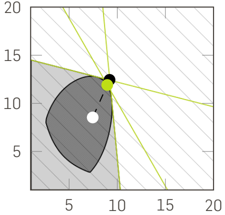
$$\tilde{x}^k_{\mathrm{LP}} = \arg\min_{\Omega_{k-1}}\ c^T x.$$

- Do line search, generate supporting hyperplane and add to Ω.

LP1-STEP – ITERATION 3

– $\Omega_2 = \{x | l_j(x) \leq 0,\ j \in \{1, 2\},\ x \in X\}$

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$
$$l_2(x) = 0.332x_1 + 1.30x_2 - 19.2$$



– $k = 3$, solve LP in Ω,

$$\tilde{x}^k_{\mathrm{LP}} = \arg\min_{\Omega_{k-1}} \ c^T x.$$

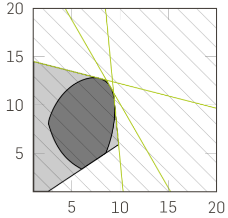
– Do line search, generate supporting hyperplane and add to Ω.

– Terminate LP1-step since $F(\tilde{x}^k_{\mathrm{LP}}) < \epsilon_{\mathrm{LP1}}$.

LP2-STEP – ITERATION 4

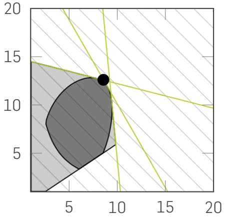
– $\Omega_3 = \{x | l_j(x) \leq 0,\ j \in \{1, 2, 3\},\ x \in X\}$

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$
$$l_2(x) = 0.332x_1 + 1.30x_2 - 19.2$$
$$l_3(x) = 1.66x_1 + 0.951x_2 - 26.2$$



- $\Omega_3 = \{x | l_j(x) \leq 0,\ j \in \{1, 2, 3\},\ x \in X\}$

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$
$$l_2(x) = 0.332x_1 + 1.30x_2 - 19.2$$
$$l_3(x) = 1.66x_1 + 0.951x_2 - 26.2$$

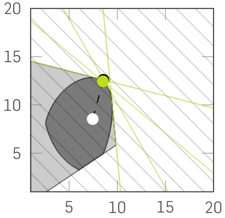


- $k = 4$, solve LP now in $\Omega \cap L$,

$$\tilde{x}^k_{\text{LP}} = \arg\min_{\Omega_{k-1} \cap L}\ c^T x.$$

- $\Omega_3 = \{x | l_j(x) \leq 0,\ j \in \{1, 2, 3\},\ x \in X\}$

$$\begin{aligned} l_1(x) &= 3.26x_1 + 0.313x_2 - 33.9 \\ l_2(x) &= 0.332x_1 + 1.30x_2 - 19.2 \\ l_3(x) &= 1.66x_1 + 0.951x_2 - 26.2 \end{aligned}$$



- $k = 4$, solve LP now in $\Omega \cap L$,

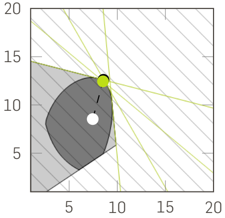
$$\tilde{x}^k_{\text{LP}} = \arg\min_{\Omega_{k-1} \cap L}\ c^T x.$$

- Do line search, generate supporting hyperplane and add to Ω.

LP2-STEP – ITERATION 4

– $\Omega_3 = \{x | l_j(x) \leq 0,\ j \in \{1, 2, 3\},\ x \in X\}$

$$l_1(x) = 3.26x_1 + 0.313x_2 - 33.9$$
$$l_2(x) = 0.332x_1 + 1.30x_2 - 19.2$$
$$l_3(x) = 1.66x_1 + 0.951x_2 - 26.2$$



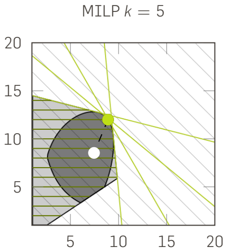
– $k = 4$, solve LP now in $\Omega \cap L$,

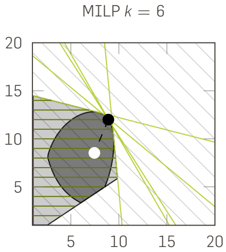
$$\tilde{x}^k_{\mathrm{LP}} = \arg\min_{\Omega_{k-1} \cap L}\ c^T x.$$

– Do line search, generate supporting hyperplane and add to Ω.

– Terminate LP2-step since $F(\tilde{x}^k_{\mathrm{LP}}) < \epsilon_{\mathrm{LP2}}$.

MILP STEP – ITERATIONS 5 AND 6





- In this step the integer requirements in Y are also considered, *i.e.*, initially $k = 5$, $\Omega = \Omega_{k-1} \cap L \cap Y$.
- The MILP steps are required to guarantee an integer-feasible solution.

— Solving the MINLP problem with the supporting hyperplane algorithm gives the following solution

Type	Iteration	Obj. funct.	x_1	x_2	$F(x_1, x_2)$
LP1	1	-40.0000	20.0000	20.0000	30 359
LP1	2	-28.4720	8.47199	20.0000	14.9321
LP1	3	-21.6378	9.19722	12.4406	0.957382
LP2	4	-21.1639	8.56022	12.6037	0.229455
MILP	5	-20.9065	8.90647	12	0.00442134
MILP	6	-20.9036	8.90362	12	$4.22619 \cdot 10^{-6}$

SOLUTION AND COMPARISONS TO OTHER SOLVERS

- Solving the MINLP problem with the supporting hyperplane algorithm gives the following solution

Type	Iteration	Obj. funct.	x_1	x_2	$F(x_1, x_2)$
LP1	1	-40.0000	20.0000	20.0000	30 359
LP1	2	-28.4720	8.47199	20.0000	14.9321
LP1	3	-21.6378	9.19722	12.4406	0.957382
LP2	4	-21.1639	8.56022	12.6037	0.229455
MILP	5	-20.9065	8.90647	12	0.00442134
MILP	6	-20.9036	8.90362	12	$4.22619 \cdot 10^{-6}$

- Solution times compared to some other MINLP solvers:

Solver	Subproblems solved	Time (s)	Implementation
ESH	4 LP + 2 MILP (2 OPT)	0.04	COIN-OR (IPOPT + CPLEX)
ECP (convex)	22 MILP (9 OPT) + 12 NLP	1.51	GAMS 24.2 (CONOPT + CPLEX)
DICOPT	11 NLP + 10 MILP	1.00	GAMS 24.2 (CONOPT + CPLEX)
SBB	4 NLP	0.41	GAMS 24.2 (CONOPT)

NUMERICAL COMPARISONS

SOME PRELIMINARY TESTS

A prototype ESH solver utilizing COIN-OR projects, IPOpt as NLP solver and CPLEX 12.6 as MILP solver was applied to some convex MINLP test problems from MINLPLib2.

It was compared to some MINLP solvers available in GAMS:

- AlphaECP (convex version), ANTIGONE, BARON, DICOPT, SBB and SCIP

The runs were terminated after 1,800 seconds.

QUADRATIC PROBLEMS

Solver	alan	fac3	netmod-dol2	netmod-kar1	slay05h
ESH	**0.01**	0.76	94.19	21.87	38.49
ECP	0.28	0.22	467.35	82.54	84.46
ANTIGONE	0.33	92.75	113.9	157.01	0.61
BARON	0.14	1.31	∗	#	1,067.14
DICOPT	0.14	0.53	#	#	**0.19**
SBB	**0.01**	**0.20**	#	23.26	6.13
SCIP	**0.01**	0.23	**43.57**	**4.04**	1.24
Variables	8	66	1,998	456	230
Binaries	4	12	462	136	40
Integers	0	0	0	0	0
Type	MBQP	MBQP	MBQP	MBQP	MBQP

Solver	du-opt	ex4
ESH	33.3	1.01
ECP	22.98	0.75
ANTIGONE	∗	**0.22**
BARON	13.74	2.62
DICOPT	#	0.44
SBB	**0.33**	1.06
SCIP	0.7	0.45
Variables	20	36
Binaries	0	25
Integers	13	0
Type	MIQP	MBQCQP

All times in seconds. Terminated problems after 1,800 s indicated with ∗. Nonoptimal solution indicated with #.

GENERAL NONLINEAR PROBLEMS

Solver	batchs101006m	enpro56pb	o7	rsyn0805m04h	rsyn0830m04h	sssd25-08
ESH	17.89	1.34	**461.75**	6.64	111.63	**58.71**
ECP	17.8	2.37	$*$	4.46	25.02	$*$
ANTIGONE	15.68	0.75	$*$	$*$	$*$	668.71
BARON	161.81	7.75	$*$	66.87	$*$	$*$
DICOPT	**1.75**	**0.34**	#	**2.31**	**4.57**	$*$
SBB	#	#	#	10.8	171.91	#
SCIP	11.68	1.48	$*$	13.63	$*$	$*$
Variables	278	127	114	1,400	1,956	256
Binaries	129	73	42	296	416	224
Integers	0	0	0	0	0	0
Type	MBNLP	MBNLP	MBNLP	MBNLP	MBNLP	MBNLP

Solver	fo7-ar4-1	fo9-ar3-1	jit1	m7-ar5-1
ESH	**32.51**	**169.15**	0.12	**1.06**
ECP	79.45	612.68	0.25	4.84
ANTIGONE	33.58	$*$	1.36	1.61
BARON	$*$	$*$	0.1	166.62
DICOPT	#	#	#	#
SBB	#	#	0.03	#
SCIP	34.12	393.22	**0.01**	13.15
Variables	112	180	25	112
Binaries	0	0	0	0
Integers	42	72	4	42
Type	MINLP	MINLP	MINLP	MINLP

All times in seconds. Terminated problems after 1,800 s indicated with $*$. Nonoptimal solution indicated with #.

CONCLUSIONS

FUTURE WORK

A journal paper will soon be submitted.

FUTURE WORK

A journal paper will soon be submitted.

Planned implementations of the algorithm

- COIN-OR
- Mathematica / Wolfram Language
- GAMS

FUTURE WORK

A journal paper will soon be submitted.

Planned implementations of the algorithm

- COIN-OR
- Mathematica / Wolfram Language
- GAMS

Algorithmic development

- selection (update) strategies of the interior point
- strategies for the LP1/LP2 steps
- MILP solution strategies

FUTURE WORK

A journal paper will soon be submitted.

Planned implementations of the algorithm

- COIN-OR
- Mathematica / Wolfram Language
- GAMS

Algorithmic development

- selection (update) strategies of the interior point
- strategies for the LP1/LP2 steps
- MILP solution strategies

Extensive comparisons to other solvers.

We are looking for additional reference test problems!

Thank you for your attention!

Thank you for your attention!

Any questions?