An extended supporting hyperplane algorithm for convex MINLP problems

Jan Kronqvist, Andreas Lundell and Tapio Westerlund

Center of Excellence in Optimization and Systems Engineering
Åbo Akademi University, Finland

OSE annual seminar
Turku
November 14, 2014
Contents of the talk

- The extended cutting plane (ECP) algorithm is briefly introduced.
Contents of the talk

- The extended cutting plane (ECP) algorithm is briefly introduced.

- The extended supporting hyperplane (ESH) algorithm, a new algorithm for solving convex MINLP problems to global optimality, is introduced
  - Cutting planes are replaced with supporting hyperplanes using a line search procedure.
  - Two LP preprocessing steps are utilized to quickly get a tight linear relaxation of the part of the feasible region defined by the convex/quasiconvex constraints.
The ECP algorithm
The extended cutting plane algorithm

- The ECP algorithm is a solver for generally convex mixed-integer nonlinear programming (MINLP) problems.
The extended cutting plane algorithm

- The ECP algorithm is a solver for 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 a solver for 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.
The extended cutting plane algorithm

- First presented in internal report in 1992

- Inspired by Kelley’s cutting plane method

- Published in Computers & Chemical Engineering in 1995
Extensions

Extensions


Extensions


An example

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

\begin{align*}
  \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 + 0.025e^{x_1}x_2^{-2} - 5 \leq 0 \\
 & \quad g_2(x_1, x_2) = 1/x_1 + 1/x_2 - x_1^{0.5}x_2^{0.5} + 4 \leq 0 \\
 & \quad 2x_1 - 3x_2 - 2 \leq 0 \\
 & \quad 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{align*}
An example

minimize \( c^T x = -x_1 - x_2 \)

subject to

\[
\begin{align*}
g_1(x_1, x_2) &= 0.15(x_1 - 8)^2 + 0.1(x_2 - 6)^2 + 0.025e^{x_1 x_2^{-2}} - 5 \leq 0 \\
g_2(x_1, x_2) &= \frac{1}{x_1} + \frac{1}{x_2} - x_1^{0.5} x_2^{0.5} + 4 \leq 0 \\
2x_1 - 3x_2 - 2 &\leq 0 \\
1 \leq x_1 \leq 20, \quad 1 \leq x_2 \leq 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}.
\end{align*}
\]
An example

\[
\begin{align*}
\text{minimize} & \quad c^T x = -x_1 - x_2 \\
\text{subject to} & \quad g_1(x_1, x_2) = 0.15(x_1 - 8)^2 + 0.1(x_2 - 6)^2 + 0.025e^{x_1}x_2^{-2} - 5 \leq 0 \\
& \quad g_2(x_1, x_2) = 1/x_1 + 1/x_2 - x_1^{0.5}x_2^{0.5} + 4 \leq 0 \\
& \quad 2x_1 - 3x_2 - 2 \leq 0 \\
& \quad 1 \leq x_1 \leq 20, \quad 1 \leq x_2 \leq 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}.
\end{align*}
\]
In each iteration $k$ of the algorithm a MILP problem is solved to obtain the solution point $(x_1^k, x_2^k)$.

- First iteration gives $x_1^1 = 20, x_2^1 = 20$. 
In each iteration \( k \) of the algorithm a MILP problem is solved to obtain the solution point \((x^k_1, x^k_2)\).

- First iteration gives \(x^1_1 = 20, x^1_2 = 20\).
- \(g_1(x^1_1, x^1_2) = 30359.0, \quad g_2(x^1_1, x^1_2) = -15.9\).
In each iteration $k$ of the algorithm a MILP problem is solved to obtain the solution point $(x_1^k, x_2^k)$.

- First iteration gives $x_1^1 = 20, x_2^1 = 20$.
- $g_1(x_1^1, x_2^1) = 30359.0$, $g_2(x_1^1, x_2^1) = -15.9$.

A new cutting plane is generated for the violated nonlinear constraint $g_1$:

$$g_1(x_1^1, x_2^1) + \nabla g_1(x_1^1, x_2^1)^T (x - x_1^1, x - x_2^1) \leq 0$$
The nonlinear function $g_1(x_1, x_2)$
The nonlinear function $g_1(x_1, x_2)$

and the cutting plane generated at $x_1 = 20, x_2 = 20$
To solve the problem with the ECP algorithm ($\epsilon = 0.001$) it takes 17 iterations (17 MILP problems solved to optimality).
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?
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
A new interior point based algorithm for solving convex MINLP problems to global optimality.
A new interior point based algorithm for solving convex MINLP problems to global optimality.

Roots:

- The extended cutting plane algorithm 1995

---

A new interior point based algorithm for solving convex MINLP problems to global optimality.

Roots:
- The extended cutting plane algorithm 1995
- Kelley’s cutting plane algorithm 1960

A new interior point based algorithm for solving convex MINLP problems to global optimality.

Roots:

- The extended cutting plane algorithm 1995
- Kelley’s cutting plane algorithm 1960
- The supporting hyperplane method 1967

A new interior point based algorithm for solving convex MINLP problems to global optimality.

Roots:
- The extended cutting plane algorithm 1995
- Kelley’s cutting plane algorithm 1960
- The supporting hyperplane method 1967

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

---

A new interior point based algorithm for solving convex MINLP problems to global optimality.

Roots:
- The extended cutting plane algorithm 1995
- Kelley’s cutting plane algorithm 1960
- The supporting hyperplane method 1967

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

Two LP preprocessing steps are utilized to quickly get a tight linear relaxation of the part of the feasible region defined by the convex/quasiconvex constraints.

---

The MINLP problem

The algorithm finds the optimal solution \( x^* \) to the following convex MINLP problem:

\[
\begin{align*}
x^* &= \arg\min_{x \in C \cap L \cap Y} \; c^T x \\
\end{align*}
\]

(P)

where \( x = [x_1, x_2, \ldots, x_N]^T \) belongs to the compact set

\[
X = \left\{ x \mid \underline{x}_i \leq x_i \leq \bar{x}_i, \; i = 1, \ldots, N \right\} \subset \mathbb{R}^n,
\]

the feasible region is defined by \( C \cap L \cap Y \),

\[
\begin{align*}
C &= \{ x \mid g_m(x) \leq 0, \; m = 1, \ldots, M, \; x \in X \}, \\
L &= \{ x \mid Ax \leq a, \; Bx = b, \; x \in X \}, \\
Y &= \{ x \mid x_i \in \mathbb{Z}, \; i \in I_\mathbb{Z}, \; x \in X \},
\end{align*}
\]

and \( C \) is a convex set.
Steps in the ESH algorithm

**NLP:** Obtain a feasible, relaxed interior point (a point in the set $C$) by solving a NLP problem.
Steps in the ESH algorithm

**NLP:** Obtain a feasible, relaxed interior point (a point in the set $C$) by solving a NLP problem.

**LP1:** Solve simple LP problems (initially in $X$) to add additional supporting hyperplanes.
Steps in the ESH algorithm

**NLP:** Obtain a feasible, relaxed interior point (a point in the set \( C \)) by solving a NLP problem.

**LP1:** Solve simple LP problems (initially in \( X \)) to add additional supporting hyperplanes.

**LP2:** Solve simple LP problems (including the constraints in \( L \)) to add additional supporting hyperplanes.
Steps in the ESH algorithm

**NLP:** Obtain a feasible, relaxed interior point (a point in the set $C$) by solving a NLP problem.

**LP1:** Solve simple LP problems (initially in $X$) to add additional supporting hyperplanes.

**LP2:** Solve simple LP problems (including the constraints in $L$) to add additional supporting hyperplanes.

**MILP:** Solve MILP problems to find the optimal solution to (P).
The ESH algorithm

NLP step

- If an interior point is not given, obtain a feasible, relaxed interior point (satisfying all the nonlinear constraints in \( C \)) by solving a NLP problem.
LP1 step (optional)

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 
The ESH algorithm

**LP1 step (optional)**

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 

![Graph showing a shaded area and a point]
LP1 step (optional)

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 
LP1 step (optional)

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 
LP1 step (optional)

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 
**LP1 step (optional)**

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 
LP1 step (optional)

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 
LP1 step (optional)

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 

![Diagram showing a convex set C with supporting hyperplanes and a line search point.](Diagram.png)
LP1 step (optional)

- Solve simple LP problems (initially in $X$) and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set $C$. 

![Diagram of a convex set with supporting hyperplanes]
LP2 step (optional)

- Continue with a corresponding procedure as in LP1 but now also including the linear constraints in $L$ in the original problem.
LP2 step (optional)

- Continue with a corresponding procedure as in LP1 but now also including the linear constraints in $L$ in the original problem.
LP2 step (optional)

- Continue with a corresponding procedure as in LP1 but now also including the linear constraints in $L$ in the original problem.
Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).
MILP step

Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).
MILP step

Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).
Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).
NLP step

- A point in $C$ is required as an endpoint for the line searches to be conducted in the LP1-, LP2- and MILP-steps.
NLP step

- A point in $C$ is required as an endpoint for the line searches to be conducted in the LP1-, LP2- and MILP-steps.

- Assuming that (P) has a solution, the internal point can be obtained from the following NLP problem:

$$\tilde{x}_{\text{NLP}} = \arg\min_{x \in X} F(x),$$  \hspace{1cm} (P-NLP)

where $F(x) := \max_{m=1,\ldots,M} \{g_m(x)\}$. 

$F$ is convex/quasiconvex since it is the maximum of convex/quasiconvex functions.

- $(P\text{-NLP})$ may be nonsmooth (if $M > 1$) even if $g_m$ is smooth.

- The point $\tilde{x}_{\text{NLP}}$ need not be optimal but then fulfill $F(\tilde{x}_{\text{NLP}}) < 0$.

- The NLP step can also be formulated as a smooth NLP problem if all functions $g_m$ are smooth and convex.
NLP step

- A point in $C$ is required as an endpoint for the line searches to be conducted in the LP1-, LP2- and MILP-steps.

- Assuming that (P) has a solution, the internal point can be obtained from the following NLP problem:

$$\tilde{x}_{\text{NLP}} = \arg\min_{x \in X} F(x),$$

where $F(x) := \max_{m=1,\ldots,M} \{g_m(x)\}$. (P-NLP)

- $F$ is convex/quasiconvex since it is the maximum of convex/quasiconvex functions.
**NLP step**

- A point in $C$ is required as an endpoint for the line searches to be conducted in the LP1-, LP2- and MILP-steps.

- Assuming that (P) has a solution, the internal point can be obtained from the following NLP problem:

$$\tilde{x}_{\text{NLP}} = \arg\min_{x \in X} F(x), \quad (P\text{-NLP})$$

where $F(x) := \max_{m=1,\ldots,M} \{g_m(x)\}$.

- $F$ is convex/quasiconvex since it is the maximum of convex/quasiconvex 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 to be conducted in the LP1-, LP2- and MILP-steps.

Assuming that (P) has a solution, the internal point can be obtained from the following NLP problem:

\[
\tilde{x}_{\text{NLP}} = \arg \min_{x \in \mathcal{X}} F(x), \quad \text{(P-NLP)}
\]

where \( F(x) := \max_{m=1,\ldots,M} \{g_m(x)\} \).

\( F \) is convex/quasiconvex since it is the maximum of convex/quasiconvex functions.

(P-NLP) may be nonsmooth (if \( M > 1 \)) even if \( g_m \) is smooth.

The point \( \tilde{x}_{\text{NLP}} \) need not be optimal but then fulfill \( F(\tilde{x}_{\text{NLP}}) < 0 \).
NLP step

- A point in $C$ is required as an endpoint for the line searches to be conducted in the LP1-, LP2- and MILP-steps.

- Assuming that (P) has a solution, the internal point can be obtained from the following NLP problem:

$$
\tilde{x}_{NLP} = \arg\min_{x \in X} F(x),
$$

\begin{equation}
\text{(P-NLP)}
\end{equation}

where $F(x) := \max_{m=1,\ldots,M} \{g_m(x)\}$.

- $F$ is convex/quasiconvex since it is the maximum of convex/quasiconvex functions.

- (P-NLP) may be nonsmooth (if $M > 1$) even if $g_m$ is smooth.

- The point $\tilde{x}_{NLP}$ need not be optimal but then fulfill $F(\tilde{x}_{NLP}) < 0$.

- The NLP step can also be formulated as a smooth NLP problem if all functions $g_m$ are smooth and convex.
LP1 step

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

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

is solved, and the point $x^k$ is obtained by a line search for $F(x^k) = 0$ between the internal point $\tilde{x}_{NLP}$ and the solution point to (P-LP1) $\tilde{x}^k_{LP}$:

$$x^k = \lambda \tilde{x}_{NLP} + (1 - \lambda) \tilde{x}^k_{LP}, \quad \lambda \in [0, 1].$$
LP1 step

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

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

is solved, and the point $x^k$ is obtained by a line search for $F(x^k) = 0$ between the internal point $\tilde{x}_{NLP}$ and the solution point to (P-LP1) $\tilde{x}_{LP}^k$:

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

Supporting hyperplanes (SHs)

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

are generated and added to $\Omega_k$. $\xi_F(x^k)^T$ is a gradient or subgradient of $F$ at $x^k$. 
LP1 step

Starting from \( k = 1, \ Omega_0 = X \), the problem

\[
\begin{align*}
\tilde{x}^k_{\text{LP}} &= \arg\min_{\Omega_{k-1}} c^T x \\
\end{align*}
\tag{P-LP1}
\]

is solved, and the point \( x^k \) is obtained by a line search for \( F(x^k) = 0 \) between the internal point \( \tilde{x}_{\text{NLP}} \) and the solution point to (P-LP1) \( \tilde{x}^k_{\text{LP}} \):

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

Supporting hyperplanes (SHs)

\[
l_k := F(x^k) + \xi_F(x^k)^T(x - x^k) \leq 0
\]

are generated and added to \( \Omega_k \).
\( \xi_F(x^k)^T \) is a gradient or subgradient of \( F \) at \( x^k \).

Repeated until \( F(\tilde{x}^k_{\text{LP}}) < \epsilon_{\text{LP1}} \) or a maximum number of SHs
LP2 step

This step is otherwise identical to LP1, with the exception that the linear constraints in $L$ are now also included, \textit{i.e.},

$$\tilde{x}_{LP}^k = \arg \min_{\Omega_{k-1} \cap L} c^T x$$ \hspace{1cm} (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.,

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

(P-LP2)

> (P-LP2) is repeatedly solved until $F(\tilde{x}_{LP}^k) < \epsilon_{LP2}$ or a maximum number of SHs have additionally been generated.
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 otherwise identical to LP2, with the exception that the integer requirements in \( Y \) are now additionally considered, i.e.,

\[
\hat{x}_\text{MILP}^k = \arg \min_{\Omega_{k-1} \cap L \cap Y} c^T x. \quad (P-\text{MILP})
\]
MILP step

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

This step is otherwise identical to LP2, with the exception that the integer requirements in $Y$ are now additionally considered, i.e.,

$$\tilde{x}_{MILP}^k = \arg\min_{\Omega_k \cap L \cap Y} c^T x.$$  (P-MILP)

(P-MILP) is repeatedly solved until $F(\tilde{x}_{MILP}^k) < \epsilon_{MILP}$. 
Finally, in order to also fulfill the integer requirements of problem (P), a MILP step is performed.

This step is otherwise identical to LP2, with the exception that the integer requirements in $Y$ are now additionally considered, i.e.,

$$\tilde{x}_{k&MILP} = \arg\min_{\Omega_{k-1} \cap L \cap Y} c^T x.$$  \hspace{1cm} (P-MILP)

(P-MILP) is repeatedly solved until $F(\tilde{x}_{k&MILP}) < \epsilon_{MILP}$.

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.
Now, consider the same example as earlier

minimize \( c^T x = -x_1 - x_2 \)
subject to

\[
0.15(x_1 - 8)^2 + 0.1(x_2 - 6)^2 + 0.025e^{x_1 x_2^{-2}} - 5 \leq 0
\]

\[
1/x_1 + 1/x_2 - x_1^{0.5} x_2^{0.5} + 4 \leq 0
\]

\[
2x_1 - 3x_2 - 2 \leq 0
\]

\[
1 \leq x_1 \leq 20, \quad 1 \leq x_2 \leq 20, \quad x_1 \in \mathbb{R}, \quad x_2 \in \mathbb{Z}.
\]
NLP step – find an interior point

\[ \tilde{x}_{NLP} = \arg \min \, F(x_1, x_2), \quad (x_1, x_2) \in X \]

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

- The problem 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 $\Omega_0 = \mathcal{X}$.
LP1-step – Iteration 1

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

$$\tilde{x}^k_{LP} = \arg\min_{\Omega_{k-1}} c^T x.$$
LP1-step – Iteration 1

- Assume initially that $\Omega_0 = X$.
- $k = 1$, solve LP in $\Omega$,
  
  $$\tilde{x}_{LP}^k = \arg\min_{\Omega_{k-1}} c^T x.$$ 

- Do line search

  $$x^k = \lambda \tilde{x}_{NLP} + (1 - \lambda) \tilde{x}_{LP}^k.$$
LP1-step – Iteration 1

- **Assume initially that** \( \Omega_0 = X \).
- **k = 1**, solve LP in \( \Omega \),
  
  \[ \tilde{x}_{LP}^k = \arg\min_{\Omega_{k-1}} c^T x. \]

- **Do line search**
  
  \[ x^k = \lambda \tilde{x}_{NLP} + (1 - \lambda) \tilde{x}_{LP}^k. \]

- **Generate supporting hyperplane** in \( x^k \) and add to \( \Omega \).
LP1-step – Iteration 2

\[ \Omega_1 = \{ x | l_1(x) \leq 0, \ x \in X \} \]

\[ l_1(x) = 3.26x_1 + 0.313x_2 - 33.9 \]
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 $\Omega$, 
  \[ \tilde{x}_{LP}^k = \arg \min_{\Omega_{k-1}} c^T x. \]
LP1-step – Iteration 2

\[ \Omega_1 = \{ x | l_1(x) \leq 0, \ x \in X \} \]

\[ l_1(x) = 3.26x_1 + 0.313x_2 - 33.9 \]

\[ k = 2, \text{ solve LP in } \Omega, \]

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

\[ \text{Do line search } x^k = \lambda \tilde{x}_{\text{NLP}} + (1 - \lambda) \tilde{x}_{\text{LP}}^k. \]
The ESH algorithm

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 $\Omega$,
  \[
  \tilde{x}_{LP}^k = \arg\min_{\Omega_{k-1}} c^T x.
  \]
- Do line search $x^k = \lambda\tilde{x}_{NLP} + (1 - \lambda)\tilde{x}_{LP}^k$.
- Generate supporting hyperplane in $x^k$ and add to $\Omega$. 
\[ \Omega_2 = \{ x \mid l_j(x) \leq 0, \ j \in \{1, 2\}, \ x \in X \} \]

\[ l_1(x) = 3.26x_1 + 0.313x_2 - 33.9 \]

\[ l_2(x) = 0.332x_1 + 1.30x_2 - 19.2 \]
LP1-step – Iteration 3

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

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

\[ k = 3, \text{ solve LP in } \Omega, \]

\[ \tilde{x}_{LP}^k = \arg \min_{\Omega_{k-1}} c^T x. \]
LP1-step – Iteration 3

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

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

\[ k = 3, \ \text{solve LP in } \Omega, \]

\[ \tilde{x}_{LP}^k = \operatorname{argmin}_{\Omega_{k-1}} c^T x. \]

\[ \tilde{x}_{LP} = \operatorname{argmin}_{\Omega_{k-1}} c^T x. \]

\[ \tilde{x}_{LP} = \operatorname{argmin}_{\Omega_{k-1}} c^T x. \]

Do line search, generate supporting hyperplane and add to \( \Omega \).
The ESH algorithm

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 $\Omega$,

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

Do line search, generate supporting hyperplane and add to $\Omega$.

Terminate LP1-step since $F(\tilde{x}_{LP}^k) < \epsilon_{LP1}$. 
LP2-step – Iteration 4

\[ \Omega_3 = \{ x \mid l_j(x) \leq 0, \ j \in \{1, 2, 3\}, \ x \in X \} \]

\[ l_1(x) = 3.26x_1 + 0.313x_2 - 33.9 \]
\[ l_2(x) = 0.332x_1 + 1.30x_2 - 19.2 \]
\[ l_3(x) = 1.66x_1 + 0.951x_2 - 26.2 \]
LP2-step – Iteration 4

\[ \Omega_3 = \{ x \mid l_j(x) \leq 0, \ j \in \{1, 2, 3\}, \ x \in X \} \]

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

\[ k = 4, \text{ solve LP now in } \Omega \cap L, \]
\[ \tilde{x}_{LP}^k = \arg \min_{\Omega_{k-1} \cap L} c^T x. \]
LP2-step – Iteration 4

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

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

\[ k = 4, \text{ solve LP now in } \Omega \cap L, \]
\[ \tilde{x}^k_{\text{LP}} = \arg \min_{\Omega_{k-1} \cap L} c^T x. \]

\[ \text{Do line search, generate supporting hyperplane and add to } \Omega. \]
LP2-step – Iteration 4

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

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

\[ k = 4, \text{ solve LP now in } \Omega \cap L, \]
\[ \tilde{x}_{LP}^k = \arg \min_{\Omega_{k-1} \cap L} c^T x. \]

Do line search, generate supporting hyperplane and add to \( \Omega \).

Terminate LP2-step since \( F(\tilde{x}_{LP}^k) < \epsilon_{LP2} \).
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.
Solution and comparisons to other solvers

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Iteration</th>
<th>Obj. funct.</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$F(x_1, x_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP1</td>
<td>1</td>
<td>−40.0000</td>
<td>20.0000</td>
<td>20.0000</td>
<td>30 359</td>
</tr>
<tr>
<td>LP1</td>
<td>2</td>
<td>−28.4720</td>
<td>8.47199</td>
<td>20.0000</td>
<td>14.9321</td>
</tr>
<tr>
<td>LP1</td>
<td>3</td>
<td>−21.6378</td>
<td>9.19722</td>
<td>12.4406</td>
<td>0.957382</td>
</tr>
<tr>
<td>LP2</td>
<td>4</td>
<td>−21.1639</td>
<td>8.56022</td>
<td>12.6037</td>
<td>0.229455</td>
</tr>
<tr>
<td>MILP</td>
<td>5</td>
<td>−20.9065</td>
<td>8.90647</td>
<td>12</td>
<td>0.00442134</td>
</tr>
<tr>
<td>MILP</td>
<td>6</td>
<td>−20.9036</td>
<td>8.90362</td>
<td>12</td>
<td>$4.22619 \cdot 10^{-6}$</td>
</tr>
</tbody>
</table>
Solution and comparisons to other solvers

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

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

- Solution times compared to some other MINLP solvers:

<table>
<thead>
<tr>
<th>Solver</th>
<th>Subproblems solved</th>
<th>Time (s)</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESH</td>
<td>1 NLP + 4 LP + 2 MILP</td>
<td>0.04</td>
<td>Early prototype</td>
</tr>
<tr>
<td>ECP</td>
<td>21 MILP (10 OPT) + 1 NLP</td>
<td>1.4</td>
<td>GAMS 24.2 + CPLEX</td>
</tr>
<tr>
<td>DICOPT</td>
<td>10 NLP + 10 MILP</td>
<td>1.5</td>
<td>GAMS 24.2 + CONOPT + CPLEX</td>
</tr>
</tbody>
</table>
### Solver comparison

- The test problems are taken from MINLPlib2, which is a collection Mixed Integer Nonlinear Programming models.

<table>
<thead>
<tr>
<th>Solver</th>
<th>fo7-ar4-1</th>
<th>fo9-ar3-1</th>
<th>jit1</th>
<th>m7-ar5-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESH</td>
<td>32.51</td>
<td>169.15</td>
<td>0.12</td>
<td>1.06</td>
</tr>
<tr>
<td>ECP</td>
<td>79.45</td>
<td>612.68</td>
<td>0.25</td>
<td>4.84</td>
</tr>
<tr>
<td>ANTIGONE</td>
<td>33.58</td>
<td>*</td>
<td>1.36</td>
<td>1.61</td>
</tr>
<tr>
<td>BARON</td>
<td>*</td>
<td>*</td>
<td>0.1</td>
<td>166.62</td>
</tr>
<tr>
<td>DICOPT</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>SBB</td>
<td>#</td>
<td>#</td>
<td>0.03</td>
<td>#</td>
</tr>
<tr>
<td>SCIP</td>
<td>34.12</td>
<td>393.22</td>
<td>0.01</td>
<td>13.15</td>
</tr>
</tbody>
</table>

| Variables | 112 | 180 | 25   | 112 |
| Binaries  | 0   | 0   | 0    | 0   |
| Integers  | 42  | 72  | 4    | 42  |
| Type      | MINLP | MINLP | MINLP | MINLP |
## Solver comparison

<table>
<thead>
<tr>
<th>Solver</th>
<th>batches101006m</th>
<th>enpro56pb</th>
<th>o7</th>
<th>rsyn0805m04h</th>
<th>rsyn0830m04h</th>
<th>sssd25-08</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESH</td>
<td>17.89</td>
<td>1.34</td>
<td>461.75</td>
<td>6.64</td>
<td>111.63</td>
<td>58.71</td>
</tr>
<tr>
<td>ECP</td>
<td>17.8</td>
<td>2.37</td>
<td>*</td>
<td>4.46</td>
<td>25.02</td>
<td>*</td>
</tr>
<tr>
<td>ANTIGONE</td>
<td>15.68</td>
<td>0.75</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>668.71</td>
</tr>
<tr>
<td>BARON</td>
<td>161.81</td>
<td>7.75</td>
<td>*</td>
<td>66.87</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>DICOPT</td>
<td>1.75</td>
<td>0.34</td>
<td>#</td>
<td>2.31</td>
<td>4.57</td>
<td>*</td>
</tr>
<tr>
<td>SBB</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>10.8</td>
<td>171.91</td>
<td>#</td>
</tr>
<tr>
<td>SCIP</td>
<td>11.68</td>
<td>1.48</td>
<td>*</td>
<td>13.63</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Variables</td>
<td>278</td>
<td>127</td>
<td>114</td>
<td>1,400</td>
<td>1,956</td>
<td>256</td>
</tr>
<tr>
<td>Binaries</td>
<td>129</td>
<td>73</td>
<td>42</td>
<td>296</td>
<td>416</td>
<td>224</td>
</tr>
<tr>
<td>Integers</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Type</td>
<td>MBNLP</td>
<td>MBNLP</td>
<td>MBNLP</td>
<td>MBNLP</td>
<td>MBNLP</td>
<td>MBNLP</td>
</tr>
</tbody>
</table>
### Solver comparison

<table>
<thead>
<tr>
<th>Solver</th>
<th>alan</th>
<th>fac3</th>
<th>netmod-dol2</th>
<th>netmod-kar1</th>
<th>slay05h</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESH</td>
<td>0.01</td>
<td>0.76</td>
<td>94.19</td>
<td>21.87</td>
<td>38.49</td>
</tr>
<tr>
<td>ECP</td>
<td>0.28</td>
<td>0.22</td>
<td>467.35</td>
<td>82.54</td>
<td>84.46</td>
</tr>
<tr>
<td>ANTIGONE</td>
<td>0.33</td>
<td>92.75</td>
<td>113.9</td>
<td>157.01</td>
<td>0.61</td>
</tr>
<tr>
<td>BARON</td>
<td>0.14</td>
<td>1.31</td>
<td>*</td>
<td>#</td>
<td>1,067.14</td>
</tr>
<tr>
<td>DICOPT</td>
<td>0.14</td>
<td>0.53</td>
<td>#</td>
<td>#</td>
<td>0.19</td>
</tr>
<tr>
<td>SBB</td>
<td>0.01</td>
<td>0.20</td>
<td>#</td>
<td>23.26</td>
<td>6.13</td>
</tr>
<tr>
<td>SCIP</td>
<td>0.01</td>
<td>0.23</td>
<td>43.57</td>
<td>4.04</td>
<td>1.24</td>
</tr>
</tbody>
</table>

| 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 |

The ESH algorithm
## Solver comparison

<table>
<thead>
<tr>
<th>Solver</th>
<th>du-opt</th>
<th>ex4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESH</td>
<td>33.3</td>
<td>1.01</td>
</tr>
<tr>
<td>ECP</td>
<td>22.98</td>
<td>0.75</td>
</tr>
<tr>
<td>ANTIGONE</td>
<td>*</td>
<td>0.22</td>
</tr>
<tr>
<td>BARON</td>
<td>13.74</td>
<td>2.62</td>
</tr>
<tr>
<td>DICOPT</td>
<td>#</td>
<td>0.44</td>
</tr>
<tr>
<td>SBB</td>
<td>0.33</td>
<td>1.06</td>
</tr>
<tr>
<td>SCIP</td>
<td>0.7</td>
<td>0.45</td>
</tr>
</tbody>
</table>

| Variables | 20 | 36 |
| Binaries  | 0  | 25 |
| Integers  | 13 | 0  |
| Type      | MIQP | MBQCQP |
Future work

Implementations of the algorithm

- Mathematica / Wolfram Language. Early prototype “available”.

- COIN-OR: Utilize the Optimization Services and Open Solver Interface APIs.

- GAMS
Future work

Implementations of the algorithm

- Mathematica / Wolfram Language. Early prototype “available”.

- COIN-OR: Utilize the Optimization Services and Open Solver Interface APIs.

- GAMS

Development of the algorithm

- Pseudoconvex constraints and objective functions.
- Selection (update) strategies of the interior point.
- Strategies for the LP1/LP2 steps.
Thank you!