Programmation Mathématique Avancée: Mixed Integer Nonlinear Programming

Claudia D'Ambrosio

LIX, CNRS & École Polytechnique

MPRO – PMA

http://www.lix.polytechnique.fr/~dambrosio/teaching/MPRO/PMA-2024/pma-2024.php

General Information about PMA

Lecturers:

- **Amélie Lambert** (CNAM): Quadratic Programming (3 lectures)
- **Claudia D'Ambrosio, responsable du cours** (CNRS & École Polytechnique): convex and nonconvex Mixed Integer Nonlinear Programming (2 lectures)

Webpage: `http://www.lix.polytechnique.fr/~dambrosio/teaching/MPRO/PMA-2024/pma-2024.php`

General Information about PMA

Webpage: `http://www.lix.polytechnique.fr/~dambrosio/teaching/MPRO/PMA-2024/pma-2024.php`

Exam:

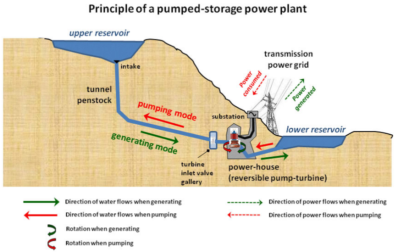
- documents/material allowed
- 2 parts, one per lecturer
- 2 hours (**Starts at 9:30 am**)

Outline

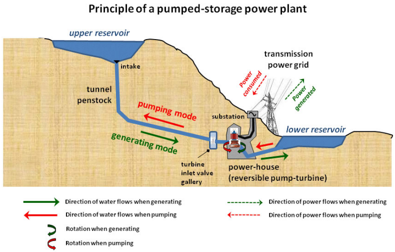
Outline

Motivating Applications

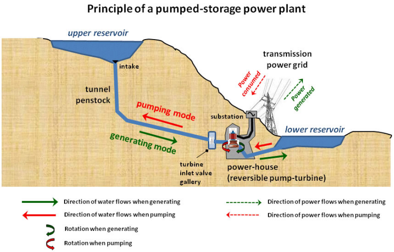
Motivating Applications

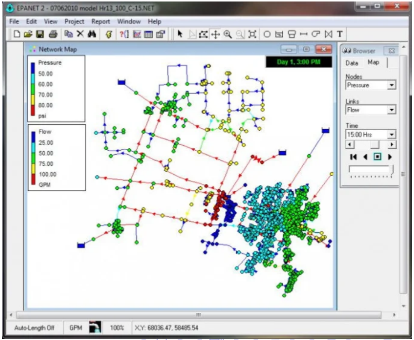


Motivating Applications



Motivating Applications





Robust Portfolio Selection

Robust Portfolio Selection

- n possibly risky assets
- mean return vector $\overline{\mu} \in \mathbb{R}^n$
- Variables: $x \in \mathbb{R}^n_+$: fraction of the porftolio value invested in each of the n assets

$$\begin{aligned} \min\ & x^\mathsf{T}\overline{\Sigma}x \\ \overline{\mu}^\mathsf{T}x &\geq R \\ \mathbf{e}^\mathsf{T}x &= 1 \\ x &\geq 0 \end{aligned}$$

where $\overline{\Sigma} \in \mathbb{R}^{n \times n}$ is the covariance return matrix, $R > 0$ is the minimum portfolio return, $\mathbf{e} \in \mathbb{R}^n$ is the all-one vector.

Robust Portfolio Selection

- n possibly risky assets
- mean return vector $\overline{\mu} \in \mathbb{R}^n$
- Variables: $x \in \mathbb{R}^n_+$: fraction of the porftolio value invested in each of the n assets

$$\min x^\mathsf{T} \overline{\Sigma} x$$

$$\begin{aligned} \overline{\mu}^\mathsf{T} x &\geq R \\ \mathbf{e}^\mathsf{T} x &= 1 \\ x &\geq 0 \end{aligned}$$

where $\overline{\Sigma} \in \mathbb{R}^{n \times n}$ is the covariance return matrix, $R > 0$ is the minimum portfolio return, $\mathbf{e} \in \mathbb{R}^n$ is the all-one vector.

- H. Markowitz, Portfolio Selection, **The Journal of Finance**, 7 (1), pp. 77–91, 1952.
- L. Mencarelli, C. D'Ambrosio. Complex Portfolio Selection via Convex Mixed-Integer Quadratic Programming: A Survey, **International Transactions in Operational Research** 26, pp. 389–414, 2019.

Outline

(MINLP)

$$
\begin{aligned}
\min f(x,y) & & \\
g_i(x,y) & \leq 0 \quad \forall i = 1, \ldots, m \\
x \in X & \\
y \in Y &
\end{aligned}
$$

where $f(x,y) : \mathbb{R}^n \to \mathbb{R}$, $g_i(x,y) : \mathbb{R}^n \to \mathbb{R}\ \forall i = 1, \ldots, m$, $X \subseteq \mathbb{R}^{n_1}$ $Y \subseteq \mathbb{N}^{n_2}$ and $n = n_1 + n_2$.

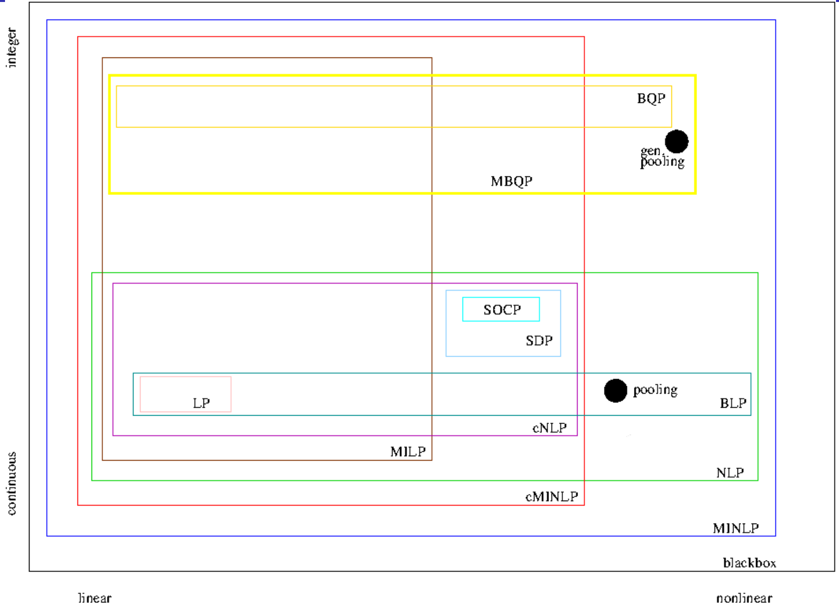
(MINLP)

$$\begin{aligned} \min f(x,y) & \\ g_i(x,y) &\leq 0 \quad \forall i = 1,\ldots,m \\ x &\in X \\ y &\in Y \end{aligned}$$

where $f(x,y) : \mathbb{R}^n \rightarrow \mathbb{R}$, $g_i(x,y) : \mathbb{R}^n \rightarrow \mathbb{R}\ \forall i = 1,\ldots,m$, $X \subseteq \mathbb{R}^{n_1}$ $Y \subseteq \mathbb{N}^{n_2}$ and $n = n_1 + n_2$.

Hypothesis: f and g are twice continuously differentiable functions.

Main optimization problem classes



Outline

Theorem [Jeroslow, 1973]
The problem of minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function.

Theorem [Jeroslow, 1973]
The problem of minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function.

Theorem [De Loera et al., 2006]
The problem of minimizing a linear function over polynomial constraints in at most 10 integer variables is not computable by a recursive function.

Complexity

Theorem [Jeroslow, 1973]
The problem of minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function.

Theorem [De Loera et al., 2006]
The problem of minimizing a linear function over polynomial constraints in at most 10 integer variables is not computable by a recursive function.

Solvable if we add

- $x_j^L \leq x_j \leq x_j^U \; \forall j = 1, \ldots, n_1$ and
- $y_j^L \leq y_j \leq y_j^U \; \forall j = 1, \ldots, n_2$

to (MINLP).

References

- R.G. Jeroslow, There Cannot be any Algorithm for Integer Programming with Quadratic Constraints, **Journal Operations Research**, 21 (1), pp. 221–224, 1973.
- J. A. De Loera, R. Hemmecke, M. Köppe, R. Weismantel, Integer polynomial optimization in fixed dimension, **Mathematics of Operations Research**, 31 (1), pp. 147–153, 2006.
- A. Del Pia, S.S. Dey, M. Molinaro, Mixed-integer quadratic programming is in NP, **Mathematical Programming A**, 162(1), pp. 225–240, 2017.

Outline

Exact reformulations

(MINLP')

$$\min h(w,z) \tag{1}$$
$$p_i(w,z) \leq 0 \quad \forall i = 1,\ldots,r \tag{2}$$
$$w \in W \tag{3}$$
$$z \in Z \tag{4}$$

where $h(w,z) : \mathbb{R}^q \to \mathbb{R}$, $p_i(w,z) : \mathbb{R}^q \to \mathbb{R}\ \forall i = 1,\ldots,r$, $W \subseteq \mathbb{R}^{q_1}$, $Z \subseteq \mathbb{N}^{q_2}$ and $q = q_1 + q_2$.

Exact reformulations

(MINLP')

$$\min h(w,z) \tag{1}$$
$$p_i(w,z) \leq 0 \quad \forall i = 1,\ldots,r \tag{2}$$
$$w \in W \tag{3}$$
$$z \in Z \tag{4}$$

where $h(w,z): \mathbb{R}^q \to \mathbb{R}$, $p_i(w,z): \mathbb{R}^q \to \mathbb{R}\ \forall i = 1,\ldots,r$, $W \subseteq \mathbb{R}^{q_1}$, $Z \subseteq \mathbb{N}^{q_2}$ and $q = q_1 + q_2$.

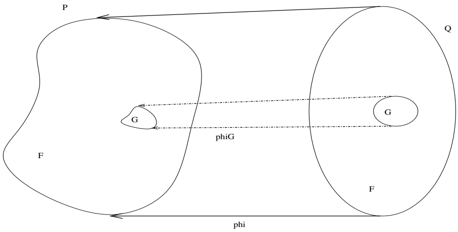
The formulation (MINLP') is an **exact reformulation** of (MINLP) if

- $\forall (w',z')$ satisfying (2)-(4), $\exists (x',y')$ feasible solution of (MINLP) s.t. $\phi(w',z') = (x',y')$
- ϕ is efficiently computable
- $\forall (w',z')$ global solution of (MINLP'), then $\phi(w',z')$ is a global solution of (MINLP)
- $\forall (x',y')$ global solution of (MINLP), there is a (w',z') global solution of (MINLP')

(MINLP')

$$\min h(w,z) \quad (1)$$
$$p_i(w,z) \leq 0 \quad \forall i = 1,\ldots,r \quad (2)$$
$$w \in W \quad (3)$$
$$z \in Z \quad (4)$$

where $h(w,z) : \mathbb{R}^q \to \mathbb{R}$, $p_i(w,z) : \mathbb{R}^q \to \mathbb{R}$ $\forall i = 1,\ldots,r$, $W \subseteq \mathbb{R}^{q_1}$, $Z \subseteq \mathbb{N}^{q_2}$ and $q = q_1 + q_2$.



Exact reformulations: example 1

$$\begin{aligned} \min y_1^2 + y_2^2 & \\ 10y_1 + 5y_2 &\leq 11 \\ y_1 &\in \{0,1\} \\ y_2 &\in \{0,1\} \end{aligned}$$

is equivalent to

$$\begin{aligned} \min y_1 + y_2 & \\ 10y_1 + 5y_2 &\leq 11 \\ y_1 &\in \{0,1\} \\ y_2 &\in \{0,1\} \end{aligned}$$

or

$$\begin{aligned} \min w_1 + w_2 & \\ w_1(=y_1^2) &= y_1 \\ w_2(=y_2^2) &= y_2 \\ 10y_1 + 5y_2 &\leq 11 \\ y_1 &\in \{0,1\} \\ y_2 &\in \{0,1\} \end{aligned}$$

Exact reformulations: example 2

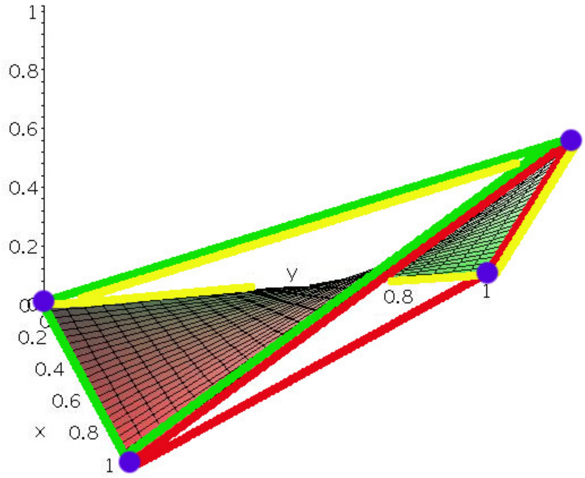
xy when *y* is binary

- If $\exists$ bilinear term xy where $x \in [0,1]$, $y \in \{0,1\}$
- We can construct an **exact reformulation**:
  - Replace each term xy by an added variable w
  - Adjoin Fortet's reformulation constraints:

$$
\begin{aligned}
w &\geq 0 \\
w &\geq x + y - 1 \\
w &\leq x \\
w &\leq y
\end{aligned}
$$

  - Get a MILP reformulation
  - Solve reformulation using MILP solver: more effective than solving MINLP

“Proof”



“Proof”

$$
\begin{aligned}
w &\geq 0\\
w &\geq x + y - 1\\
w &\leq x\\
w &\leq y
\end{aligned}
$$

$$
\begin{aligned}
y &= 0\\
\hline
w &\geq 0\\
w &\geq x - 1\\
w &\leq 0\\
w &\leq x
\end{aligned}
$$

$w = 0$

$$
\begin{aligned}
y &= 1\\
\hline
w &\geq 0\\
w &\geq x\\
w &\leq 1\\
w &\leq x
\end{aligned}
$$

$w = x$

Relaxations

(rMINLP)

$$\begin{aligned} \min \underline{f(w,z)} & \\ \underline{g_i(w,z)} &\leq 0 \quad \forall i = 1,\ldots,r \\ w &\in W \\ z &\in Z \end{aligned}$$

where $X \subseteq W \subseteq \mathbb{R}^{q_1}$, $Y \subseteq Z \subseteq \mathbb{Z}^{q_2}$, $q_1 \geq n_1$, $q_2 \geq n_2$, $\underline{f(w,z)} \leq f(x,y)$ $\forall (x,y) \subseteq (w,z)$, and $\{(x,y) | g(x,y) \leq 0\} \subseteq \text{Proj}_{(x,y)}\{(w,z) | \underline{g(w,z)} \leq 0\}$.

Examples:

- continuous relaxation: when $(w,z) \in \mathbb{R}^n$, $W = X$, $Z = Y$, $\underline{f(x,y)} = f(x,y)$, $\underline{g(x,y)} = g(x,y)$
- linear relaxation: when $q = n$, $W = X$, $Z = Y$, $\underline{f(w,z)}$ and $\underline{g(w,z)}$ are linear
- convex relaxation: when $q = n$, $W = X$, $Z = Y$, $\underline{f(w,z)}$ and $\underline{g(w,z)}$ are convex

Relaxations: example

$x_1 x_2$ when x_1, x_2 continuous

- Get bilinear term $x_1 x_2$ where $x_1 \in [x_1^L, x_1^U]$, $x_2 \in [x_2^L, x_2^U]$
- We can construct a **relaxation**:
  - Replace each term $x_1 x_2$ by an added variable w
  - Adjoin following constraints:

$$
\begin{aligned}
w &\geq x_1^L x_2 + x_2^L x_1 - x_1^L x_2^L \\
w &\geq x_1^U x_2 + x_2^U x_1 - x_1^U x_2^U \\
w &\leq x_1^U x_2 + x_2^L x_1 - x_1^U x_2^L \\
w &\leq x_1^L x_2 + x_2^U x_1 - x_1^L x_2^U
\end{aligned}
$$

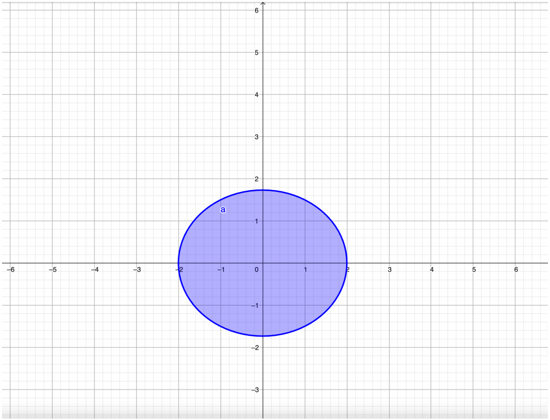
- These are called **McCormick's envelopes**
- Get an LP relaxation (solvable in polynomial time)

References & Software

- Fortet, *Applications de l'algèbre de Boole en recherche opérationelle*, **Revue Française de Recherche Opérationelle**, 4, pp. 251–259, 1960.
- McCormick, *Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems*, **Mathematical Programming**, 1976.
- Liberti, *Reformulations in Mathematical Programming: definitions and systematics*, **RAIRO-RO**, 2009.
- Liberti, Cafieri, Tarissan, *Reformulations in Mathematical Programming: a computational approach*, in Abraham et al. (eds.), **Foundations of Comput. Intel.**, 2009
- ROSE (`https://projects.coin-or.org/ROSE`)

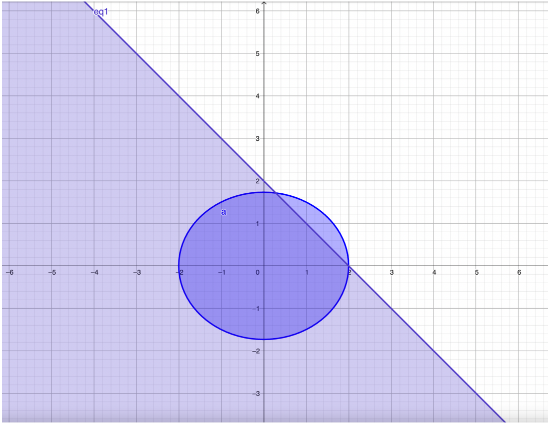
Outline

Numerical example: Convex NLP



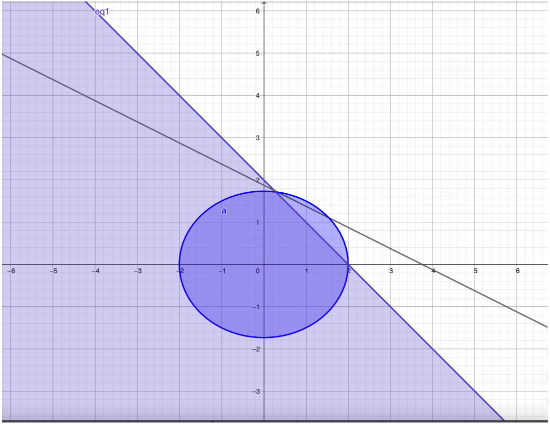
$$3x_1^2 + 4x_2^2 \leq 12$$

Numerical example: Convex NLP



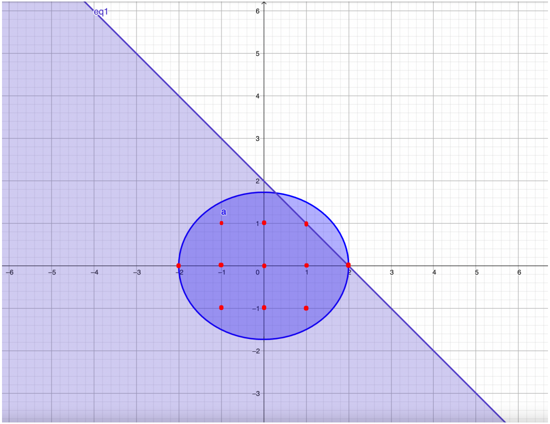
$$3x_1^2 + 4x_2^2 \leq 12$$
$$x_1 + x_2 \leq 2$$

Numerical example: Convex NLP



$$\max x_1 + 2x_2$$
$$3x_1^2 + 4x_2^2 \leq 12$$
$$x_1 + x_2 \leq 2$$

Numerical example: Convex MINLP



$$\max y_1 + 2y_2$$
$$3y_1^2 + 4y_2^2 \leq 12$$
$$y_1 + y_2 \leq 2$$
$$y_1, y_2 \in \mathbb{Z}$$

What is a convex MINLP?

Convex Mixed Integer NonLinear Programming (MINLP).

$$\begin{array}{rcl} \min f(x,y) & & \\ g(x,y) & \leq & 0 \\ x & \in & X = \{x \mid x \in \mathbb{R}^{n_1}, Dx \leq d, x^L \leq x \leq x^U\} \\ y & \in & Y = \{y \mid y \in \mathbb{Z}^{n_2}, Ay \leq a, y^L \leq y \leq y^U\} \end{array}$$

with $f(x,y) : \mathbb{R}^{n_1+n_2} \to \mathbb{R}$ and $g(x,y) : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^m$ are

- continuous

What is a convex MINLP?

Convex Mixed Integer NonLinear Programming (MINLP).

$$
\begin{array}{rcl}
\min f(x,y) & & \\
g(x,y) & \leq & 0 \\
x & \in & X = \{x \mid x \in \mathbb{R}^{n_1}, Dx \leq d, x^L \leq x \leq x^U\} \\
y & \in & Y = \{y \mid y \in \mathbb{Z}^{n_2}, Ay \leq a, y^L \leq y \leq y^U\}
\end{array}
$$

with $f(x,y) : \mathbb{R}^{n_1+n_2} \to \mathbb{R}$ and $g(x,y) : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^m$ are

* continuous
* twice differentiable

What is a convex MINLP?

Convex Mixed Integer NonLinear Programming (MINLP).

$$
\begin{aligned}
\min f(x,y) & & \\
g(x,y) &\leq& 0 \\
x &\in& X = \{x \mid x \in \mathbb{R}^{n_1}, Dx \leq d, x^L \leq x \leq x^U\} \\
y &\in& Y = \{y \mid y \in \mathbb{Z}^{n_2}, Ay \leq a, y^L \leq y \leq y^U\}
\end{aligned}
$$

with $f(x,y) : \mathbb{R}^{n_1+n_2} \to \mathbb{R}$ and $g(x,y) : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^m$ are

* continuous
* twice differentiable
* convex

functions.

What is a convex MINLP?

Convex Mixed Integer NonLinear Programming (MINLP).

$$
\begin{aligned}
\min f(x,y) & & \\
g(x,y) &\leq& 0 \\
x &\in& X = \{x \mid x \in \mathbb{R}^{n_1}, Dx \leq d, x^L \leq x \leq x^U\} \\
y &\in& Y = \{y \mid y \in \mathbb{Z}^{n_2}, Ay \leq a, y^L \leq y \leq y^U\}
\end{aligned}
$$

with $f(x,y) : \mathbb{R}^{n_1+n_2} \rightarrow \mathbb{R}$ and $g(x,y) : \mathbb{R}^{n_1+n_2} \rightarrow \mathbb{R}^m$ are

* continuous
* twice differentiable
* convex

functions.

- Local optima are also global optima .

Convex MINLP Algorithms

- Branch-and-Bound (BB).

Convex MINLP Algorithms

- Branch-and-Bound (BB).
- Outer-Approximation (OA).

Convex MINLP Algorithms

- Branch-and-Bound (BB).
- Outer-Approximation (OA).
- Generalized Benders Decomposition (GBD).

Convex MINLP Algorithms

- Branch-and-Bound (BB).
- Outer-Approximation (OA).
- Generalized Benders Decomposition (GBD).
- Extended Cutting Plane (ECP).

Convex MINLP Algorithms

- Branch-and-Bound (BB).
- Outer-Approximation (OA).
- Generalized Benders Decomposition (GBD).
- Extended Cutting Plane (ECP).
- LP/NLP-based Branch-and-Bound (QG).

Convex MINLP Algorithms

- Branch-and-Bound (BB).
- Outer-Approximation (OA).
- Generalized Benders Decomposition (GBD).
- Extended Cutting Plane (ECP).
- LP/NLP-based Branch-and-Bound (QG).
- Hybrid Algorithms (Hyb).

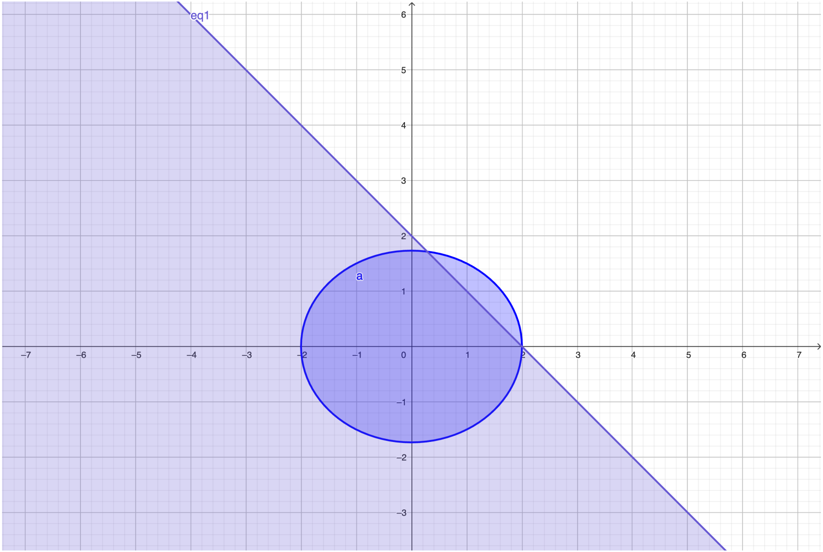
Outline

NLP relaxation

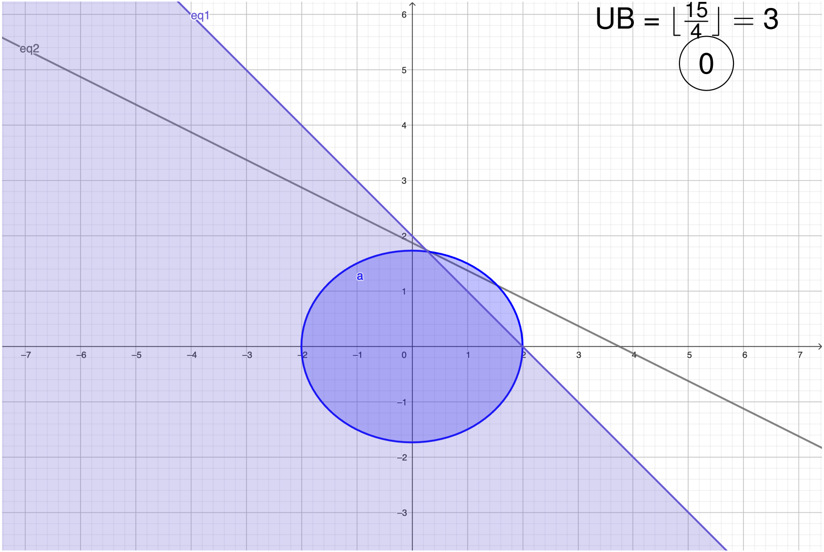
$$
\begin{aligned}
\min f(x,y) & & & \\
g(x,y) &\leq 0 & & \\
x &\in X & & \\
y &\in \{y \mid Ay \leq a\} & & \\
y_j &\leq \alpha_j^k & & j \in \{1,2,\ldots,n_2\} \\
y_j &\geq \beta_j^k & & j \in \{1,2,\ldots,n_2\}
\end{aligned}
$$

k: current step of a Branch-and-Bound procedure;
α^k: current lower bound on y ($\alpha^k \geq y^L$);
β^k: current upper bound on y ($\beta^k \leq y^U$).

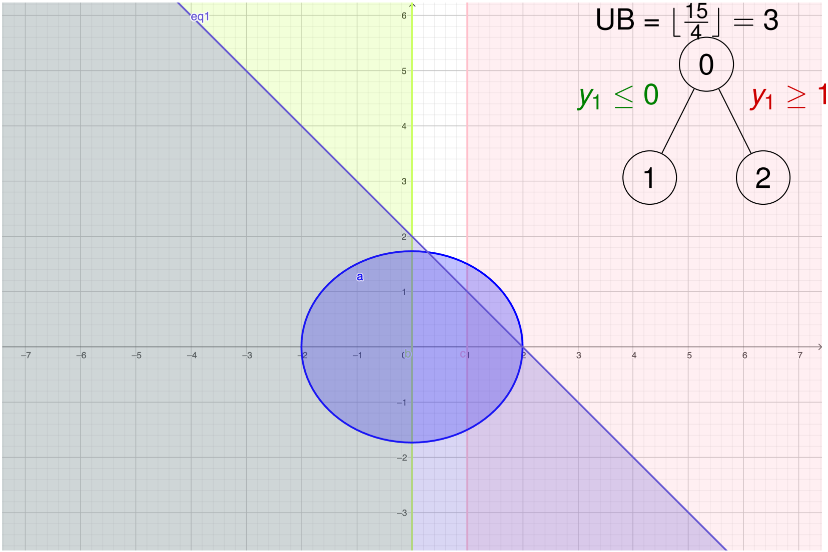
Branch-and-Bound (BB)



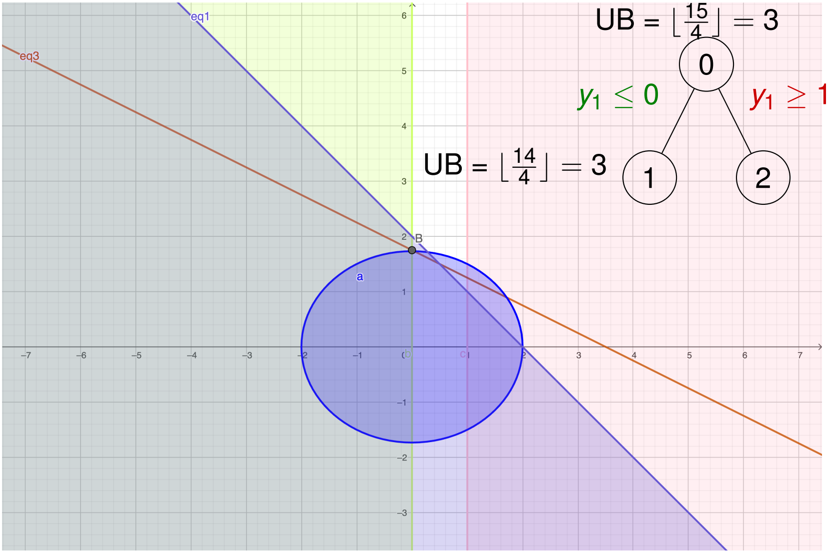
Branch-and-Bound (BB)



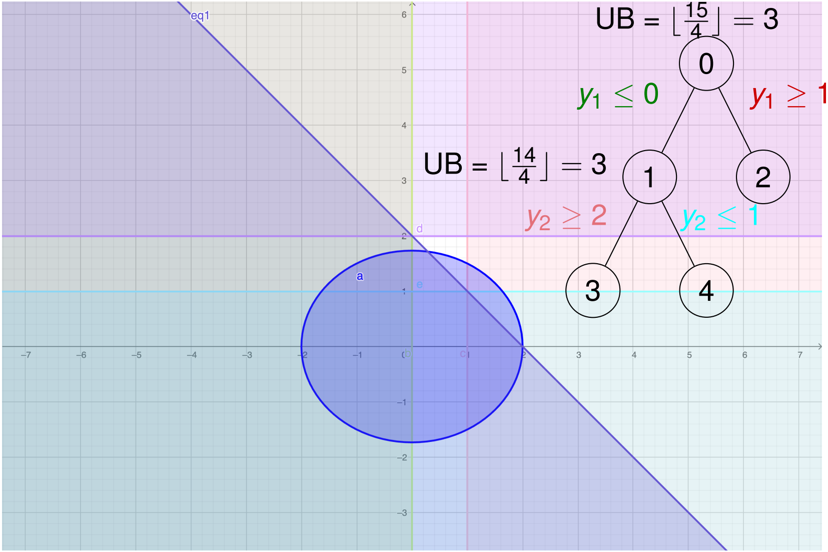
Branch-and-Bound (BB)



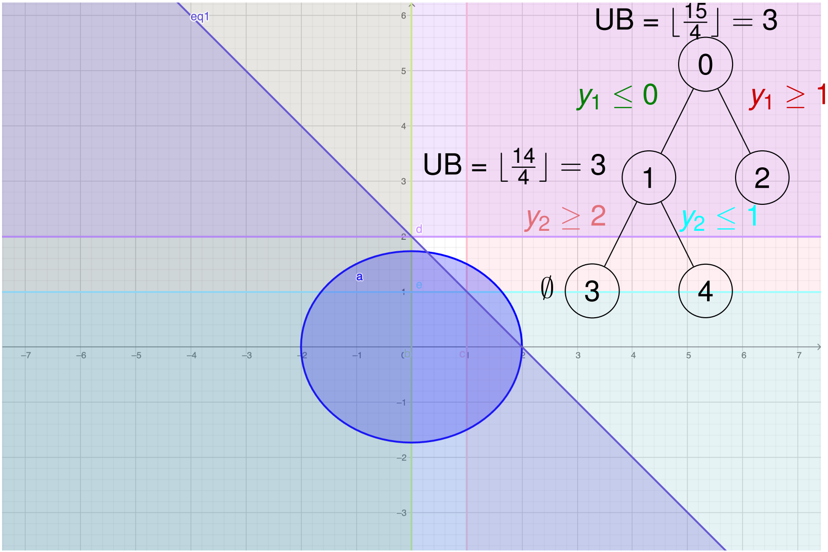
Branch-and-Bound (BB)

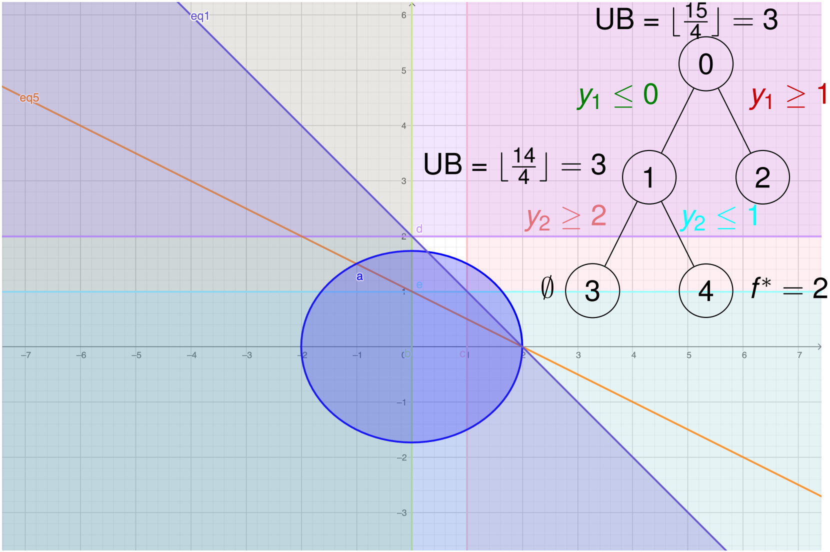


Branch-and-Bound (BB)



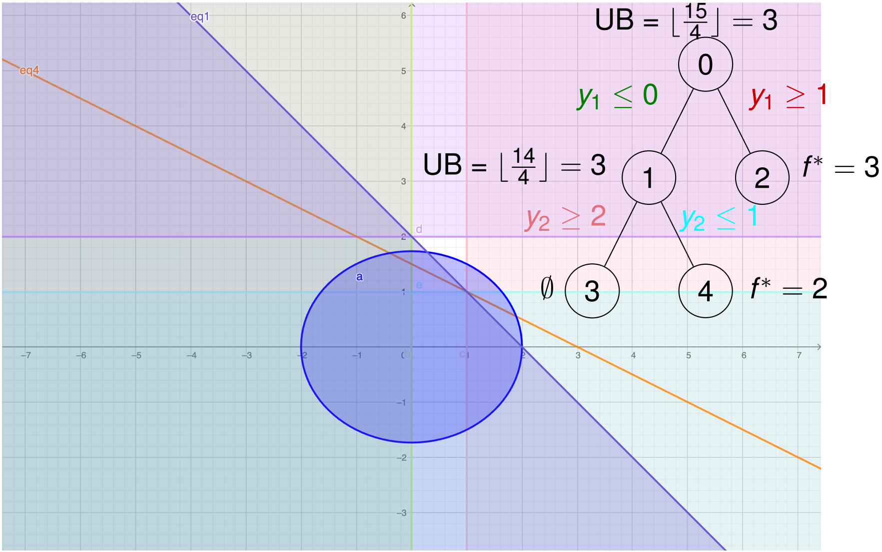
Branch-and-Bound (BB)





eq1
eq5
a
d
e
b
c
UB = $\lfloor \frac{15}{4} \rfloor = 3$
0
$y_1 \leq 0$
$y_1 \geq 1$
UB = $\lfloor \frac{14}{4} \rfloor = 3$
1
2
$y_2 \geq 2$
$y_2 \leq 1$
$\emptyset$
3
4
$f^* = 2$

Branch-and-Bound (BB)



Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

1: $f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

1: $f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.
2: **while** $\Pi \neq \emptyset$ **do**

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

1: $f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where $P^0 =$ NLP relaxation.
2: **while** $\Pi \neq \emptyset$ **do**
3: Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.

Gupta and Ravindran, 1985. Link BB for MILPs.

1: $f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where $P^0 =$ NLP relaxation.
2: **while** $\Pi \neq \emptyset$ **do**
3: Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
4: Solve P obtaining $(\bar{x}, \bar{y})$.

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.
while $\Pi \neq \emptyset$ **do**
    Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
    Solve P obtaining $(\bar{x}, \bar{y})$.
    if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
        continue
    end if

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

```
f* = +∞, Π = {P^0}, LB(P^0) = −∞ where P^0 = NLP relaxation.
while Π ≠ ∅ do
    Choose the current subproblem P ∈ Π, Π = Π \ {P}.
    Solve P obtaining (x̄, ȳ).
    if P infeasible ∨ f(x̄, ȳ) ≥ f* then
        continue
    end if
    if ȳ ∈ Z^{n_2} then
```

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where $P^0 =$ NLP relaxation.
while $\Pi \neq \emptyset$ **do**
    Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
    Solve P obtaining $(\bar{x}, \bar{y})$.
    if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
        continue
    end if
    if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
        $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where $P^0 =$ NLP relaxation.
while $\Pi \neq \emptyset$ **do**
    Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
    Solve P obtaining $(\bar{x}, \bar{y})$.
    if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
        continue
    end if
    if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
        $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
        Update Π potentially fathoming subproblems.

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.
while $\Pi \neq \emptyset$ **do**
    Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
    Solve P obtaining $(\bar{x}, \bar{y})$.
    if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
        continue
    end if
    if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
        $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
        Update Π potentially fathoming subproblems.
    else
        Take a fractional value $\bar{y}_j$ and obtain two subproblems $P^1 = P$ with $\alpha_j^1 = \lfloor \bar{y}_j \rfloor$ and $P^2 = P$ with $\beta_j^2 = \lfloor \bar{y}_j \rfloor + 1$.

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.
while $\Pi \neq \emptyset$ **do**
  Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
  Solve P obtaining $(\bar{x}, \bar{y})$.
  if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
    continue
  end if
  if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
    $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
    Update Π potentially fathoming subproblems.
  else
    Take a fractional value $\bar{y}_j$ and obtain two subproblems $P^1 = P$ with $\alpha_j^1 = \lfloor \bar{y}_j \rfloor$ and $P^2 = P$ with $\beta_j^2 = \lfloor \bar{y}_j \rfloor + 1$.
    $LB(P^1) = LB(P^2) = f(\bar{x}, \bar{y})$.
    $\Pi = \Pi \bigcup \{P^1, P^2\}$.

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.
while $\Pi \neq \emptyset$ **do**
  Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
  Solve P obtaining $(\bar{x}, \bar{y})$.
  if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
    continue
  end if
  if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
    $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
    Update Π potentially fathoming subproblems.
  else
    Take a fractional value $\bar{y}_j$ and obtain two subproblems $P^1 = P$ with $\alpha_j^1 = \lfloor \bar{y}_j \rfloor$ and $P^2 = P$ with $\beta_j^2 = \lfloor \bar{y}_j \rfloor + 1$.
    $LB(P^1) = LB(P^2) = f(\bar{x}, \bar{y})$.
    $\Pi = \Pi \bigcup \{P^1, P^2\}$.
  end if
end while
return (x^*, y^*).

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.
while $\Pi \neq \emptyset$ **do**
    Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
    Solve P obtaining $(\bar{x}, \bar{y})$.
    if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
        continue
    end if
    if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
        $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
        Update Π potentially fathoming subproblems.
    else
        Take a fractional value $\bar{y}_j$ and obtain two subproblems $P^1 = P$ with $\alpha_j^1 = \lfloor \bar{y}_j \rfloor$ and $P^2 = P$ with $\beta_j^2 = \lfloor \bar{y}_j \rfloor + 1$.
        $LB(P^1) = LB(P^2) = f(\bar{x}, \bar{y})$.
        $\Pi = \Pi \bigcup \{P^1, P^2\}$.
    end if
end while
return (x^*, y^*).

Fathoming is performed when:

- The subproblem solution is MINLP feasible (f^*).

Branch-and-Bound (BB)

Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.
while $\Pi \neq \emptyset$ **do**
    Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
    Solve P obtaining $(\bar{x}, \bar{y})$.
    if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
        continue
    end if
    if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
        $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
        Update Π potentially fathoming subproblems.
    else
        Take a fractional value $\bar{y}_j$ and obtain two subproblems $P^1 = P$ with $\alpha_j^1 = \lfloor \bar{y}_j \rfloor$ and $P^2 = P$ with $\beta_j^2 = \lfloor \bar{y}_j \rfloor + 1$.
        $LB(P^1) = LB(P^2) = f(\bar{x}, \bar{y})$.
        $\Pi = \Pi \bigcup \{P^1, P^2\}$.
    end if
end while
return (x^*, y^*).

Fathoming is performed when:

- The subproblem solution is MINLP feasible (f^*).
- The subproblem is infeasible.

Branch-and-Bound (BB)

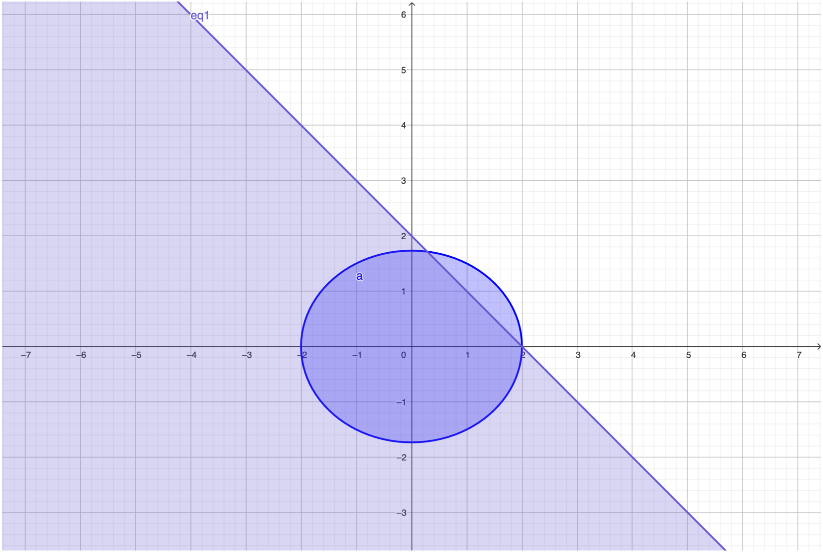
Gupta and Ravindran, 1985. Link BB for MILPs.

$f^* = +\infty$, $\Pi = \{P^0\}$, $LB(P^0) = -\infty$ where P^0 = NLP relaxation.
while $\Pi \neq \emptyset$ **do**
  Choose the current subproblem $P \in \Pi$, $\Pi = \Pi \setminus \{P\}$.
  Solve P obtaining $(\bar{x}, \bar{y})$.
  if P infeasible $\vee$ $f(\bar{x}, \bar{y}) \geq f^*$ **then**
    continue
  end if
  if $\bar{y} \in \mathbb{Z}^{n_2}$ **then**
    $f^* = f(\bar{x}, \bar{y})$, $(x^*, y^*) = (\bar{x}, \bar{y})$.
    Update Π potentially fathoming subproblems.
  else
    Take a fractional value $\bar{y}_j$ and obtain two subproblems $P^1 = P$ with $\alpha_j^1 = \lfloor \bar{y}_j \rfloor$ and $P^2 = P$ with $\beta_j^2 = \lfloor \bar{y}_j \rfloor + 1$.
    $LB(P^1) = LB(P^2) = f(\bar{x}, \bar{y})$.
    $\Pi = \Pi \bigcup \{P^1, P^2\}$.
  end if
end while
return (x^*, y^*).

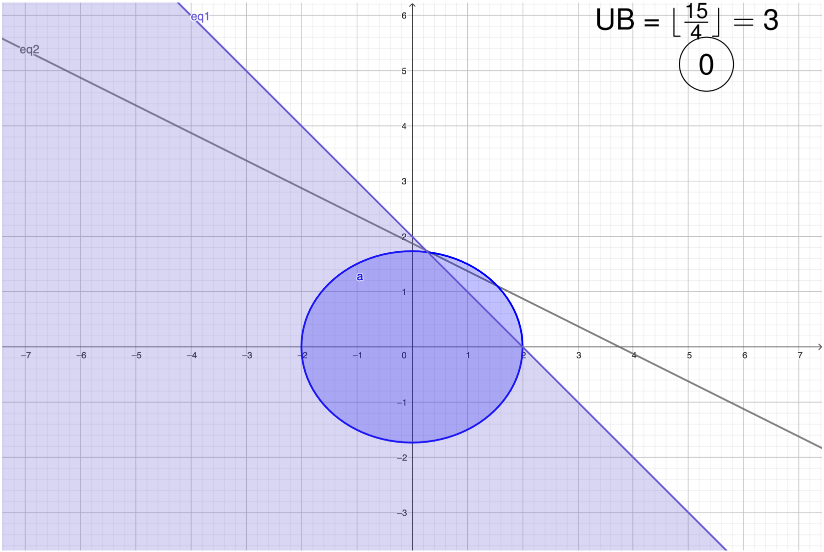
Fathoming is performed when:

- The subproblem solution is MINLP feasible (f^*).
- The subproblem is infeasible.
- The subproblem P^k has $LB(P^k) \geq f^*$.

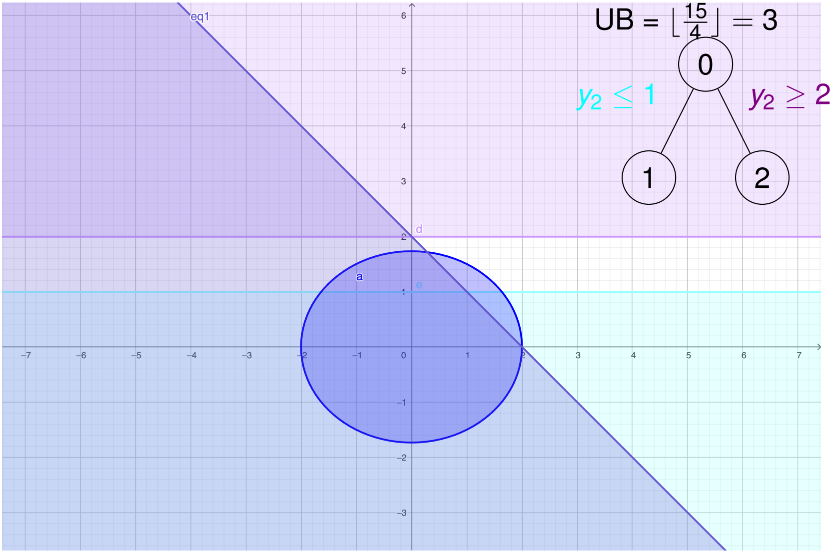
How to choose the branching variables?



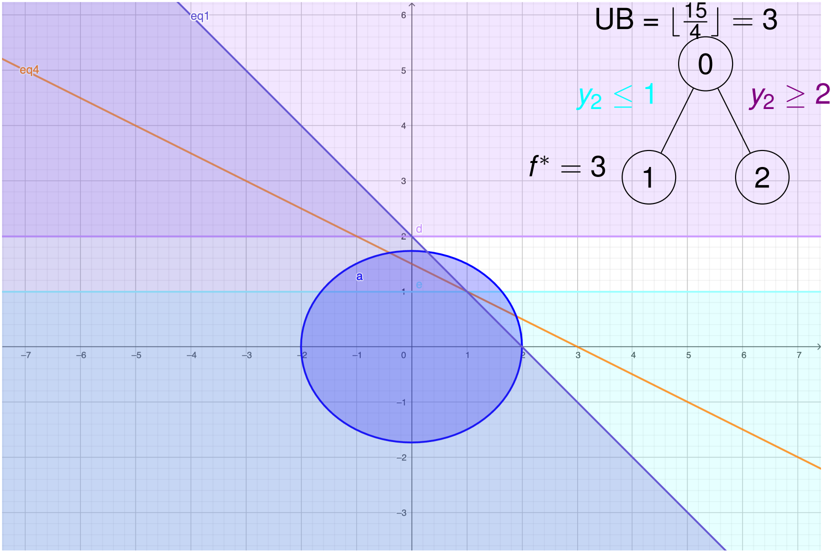
How to choose the branching variables?



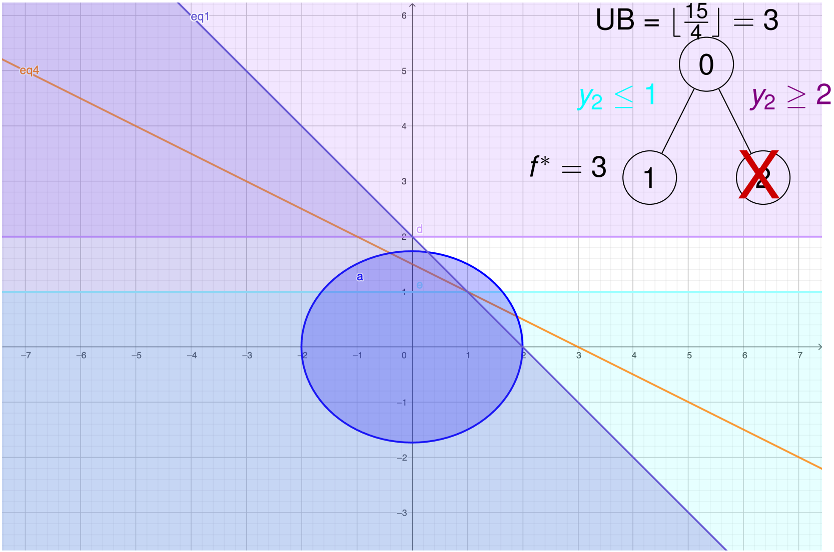
How to choose the branching variables?



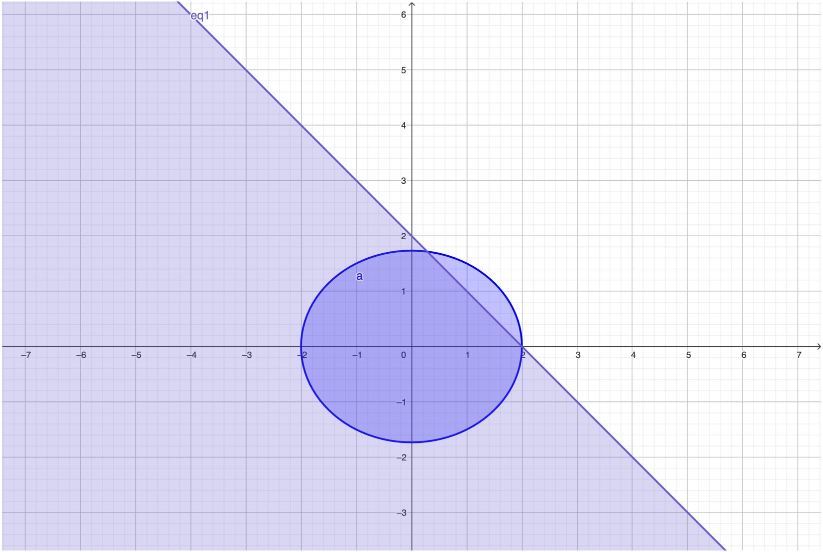
How to choose the branching variables?



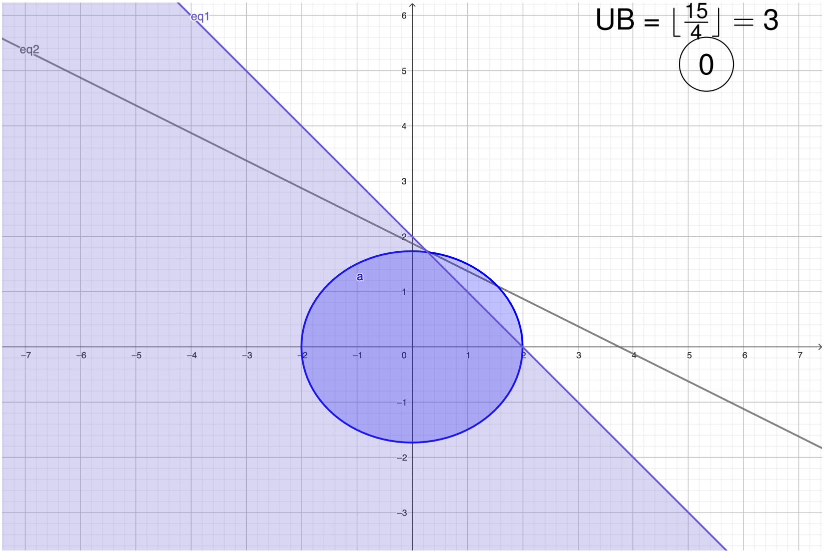
How to choose the branching variables?



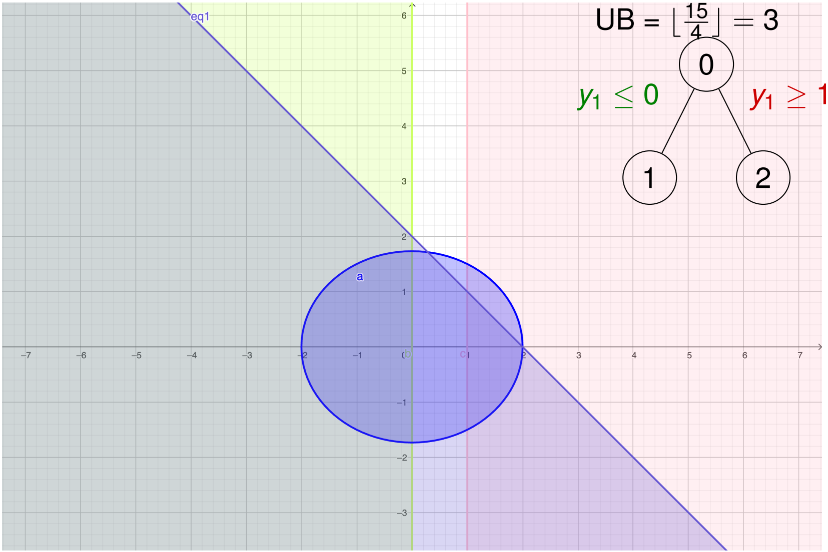
How to choose the next subproblem to explore?



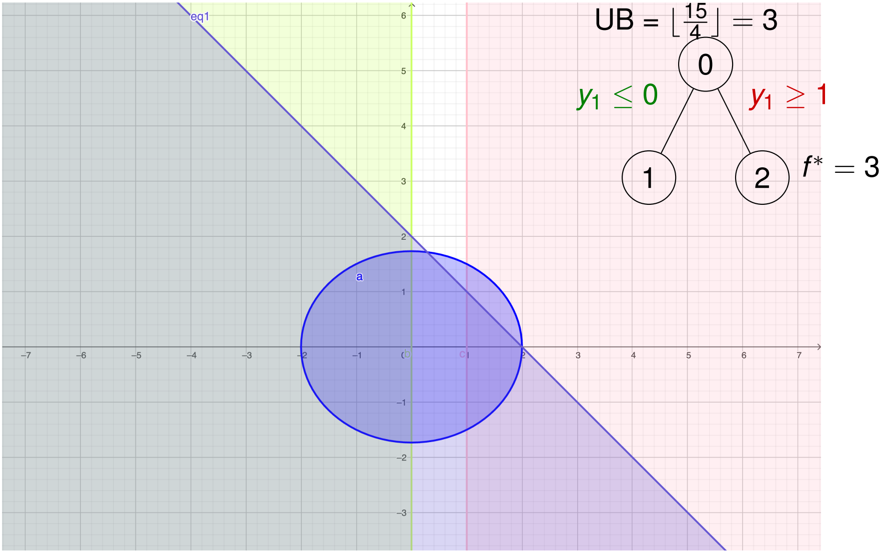
How to choose the next subproblem to explore?



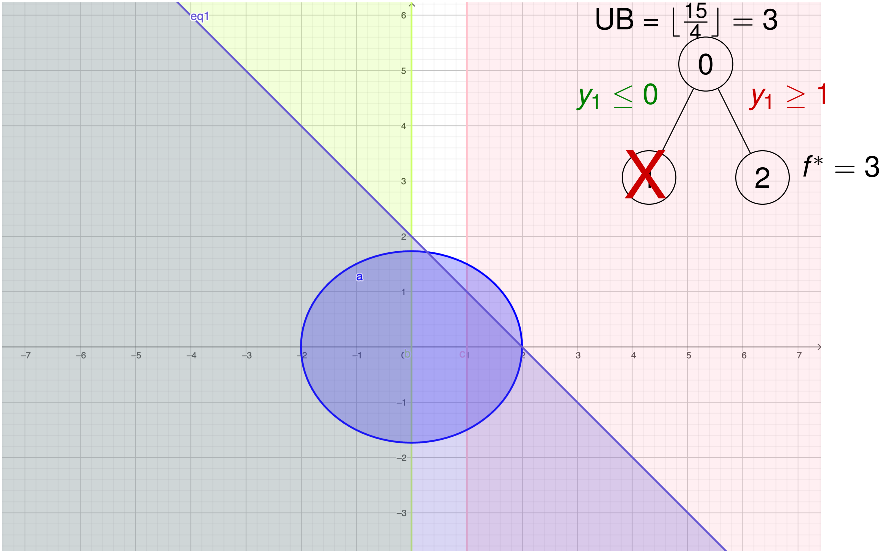
How to choose the next subproblem to explore?



How to choose the next subproblem to explore?



How to choose the next subproblem to explore?



Branch-and-Bound (BB)

Proposition

If the functions f and g are convex and twice continuously differentiable, X and Y are bounded, it follows that branch-and-bound terminates at an optimal solution after searching a finite number of nodes (or that the instance is infeasible).

Proof.

- Every NLP node can be solved to global optimality
- As X and Y are bounded, the B&B tree is finite
- Thus, similar proof for MILP B&B (see Th. 24.1 of Schrijver (1986)).

□

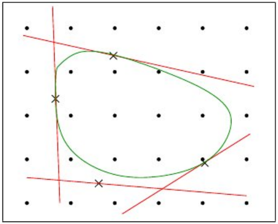
Outline

Outer-Approximation (OA)

Duran and Grossmann, 1986.

Outer-Approximation (OA)

Duran and Grossmann, 1986.



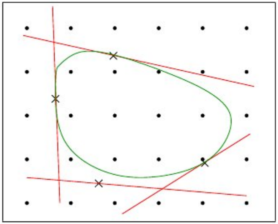
Epigraph form

$$
\begin{array}{rcll}
 & \min \gamma & & \\
f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq & \gamma & \forall k \\
g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq & 0 & \forall k \; \forall i \in I^k \\
x & \in & X & \\
y & \in & Y. &
\end{array}
$$

Outer-Approximation (OA)

Duran and Grossmann, 1986.

Epigraph form



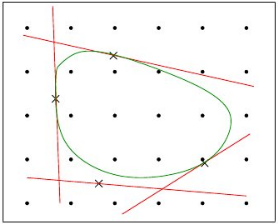
$$\min \gamma$$
$$f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq \gamma \quad \forall k$$
$$g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq 0 \quad \forall k \ \forall i \in I^k$$
$$x \in X$$
$$y \in Y.$$

$I^k = \{1, 2, \dots, m\} \ \forall k = 1, \dots, K.$

Outer-Approximation (OA)

Duran and Grossmann, 1986.

Epigraph form



$$\begin{array}{rcll} & \min \gamma & & \\ f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq & \gamma & \forall k \\ g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq & 0 & \forall k \; \forall i \in I^k \\ x & \in & X & \\ y & \in & Y. & \end{array}$$

$I^k = \{1, 2, \ldots, m\} \; \forall k = 1, \ldots, K.$

NB. The linearization constraints of MILP relaxation are not valid for non-convex MINLPs.

Outer-Approximation (OA)

MILP relaxation

$$
\begin{array}{rcll}
 & \min \gamma & & \\
f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq & \gamma & \forall k \\
g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq & 0 & \forall k \; \forall i \in I^k \\
x & \in & X & \\
y & \in & Y. &
\end{array}
$$

where $I^k \subseteq \{1, 2, \ldots, m\}$.

MILP relaxation

$$
\begin{array}{rcll}
 & \min \gamma & & \\
f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq & \gamma & \forall k \\
g_i(x^k, y^k) + \nabla g_i(x^k, y^k)^T \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} & \leq & 0 & \forall k \, \forall i \in I^k \\
x & \in & X & \\
y & \in & Y. &
\end{array}
$$

where $I^k \subseteq \{1, 2, \ldots, m\}$. Two "classical" choices:

- $I^k = \{1, 2, \ldots, m\}$

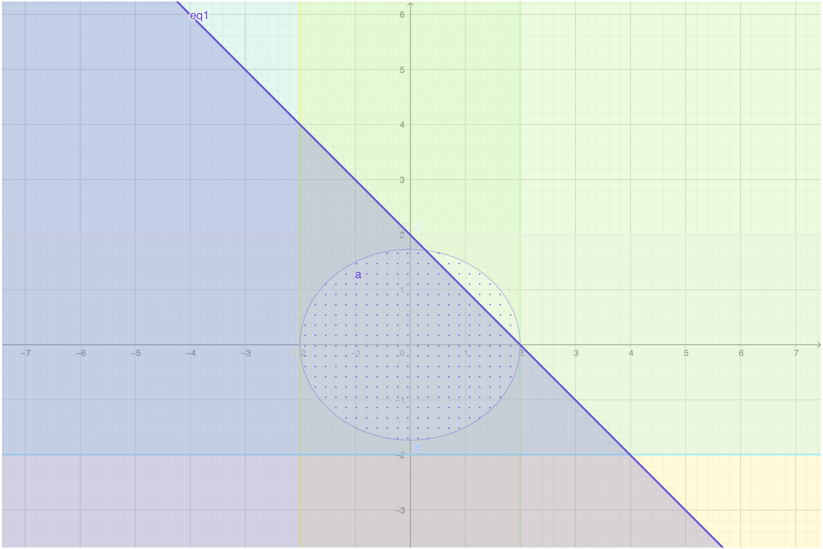
MILP relaxation

$$\begin{array}{rcll} & \min \gamma & & \\ f(x^k,y^k)+\nabla f(x^k,y^k)^T \begin{pmatrix} x-x^k \\ y-y^k \end{pmatrix} & \leq & \gamma & \forall k \\ g_i(x^k,y^k)+\nabla g_i(x^k,y^k)^T \begin{pmatrix} x-x^k \\ y-y^k \end{pmatrix} & \leq & 0 & \forall k\ \forall i \in I^k \\ x & \in & X & \\ y & \in & Y. & \end{array}$$

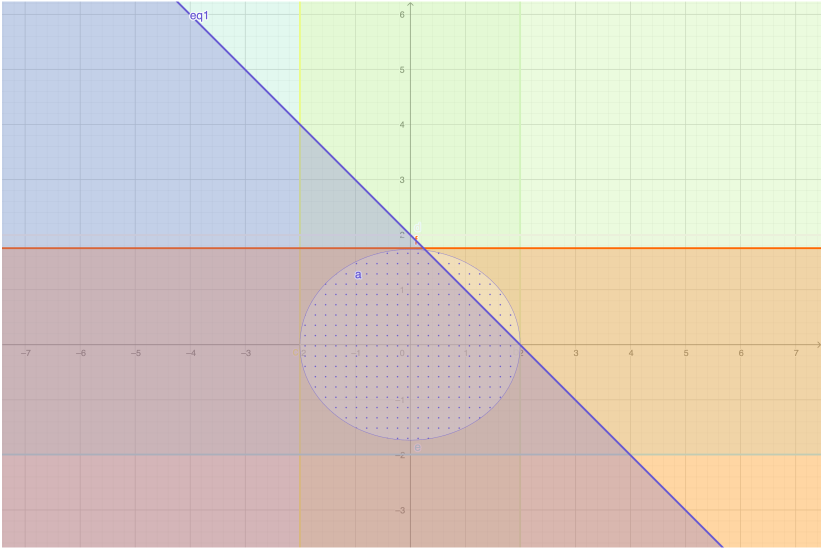
where $I^k \subseteq \{1,2,\ldots,m\}$. Two “classical” choices:

- $I^k = \{1,2,\ldots,m\}$
- $I^k = \{i \mid g(x^k,y^k) > 0, 1 \leq i \leq m\}$

Outer-Approximation (OA)



Outer-Approximation (OA)



Outer-Approximation (OA)

1: $K = 1$,

Outer-Approximation (OA)

1: $K = 1$, define an initial MILP relaxation,

Outer-Approximation (OA)

1: $K = 1$, define an initial MILP relaxation, $f^* = +\infty$,

Outer-Approximation (OA)

1: $K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.

Outer-Approximation (OA)

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
  Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.

Outer-Approximation (OA)

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
    Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.
    Solve the current NLP restriction for y^K.

Outer-Approximation (OA)

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
    Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.
    Solve the current NLP restriction for y^K.
    if NLP restriction for y^K infeasible **then**

Outer-Approximation (OA)

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
    Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.
    Solve the current NLP restriction for y^K.
    if NLP restriction for y^K infeasible **then**
        Solve the infeasibility subproblem for y^K.

Outer-Approximation (OA)

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
  Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.
  Solve the current NLP restriction for y^K.
  if NLP restriction for y^K infeasible **then**
    Solve the infeasibility subproblem for y^K.
  else
    if $f(x^K, y^K) < f^*$ **then**
      $f^* = f(x^K, y^K)$, $(x^*, y^*) = (x^K, y^K)$.

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
  Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.
  Solve the current NLP restriction for y^K.
  if NLP restriction for y^K infeasible **then**
    Solve the infeasibility subproblem for y^K.
  else
    if $f(x^K, y^K) < f^*$ **then**
      $f^* = f(x^K, y^K)$, $(x^*, y^*) = (x^K, y^K)$.
    end if
  end if
  Generate linearization cuts, update MILP relax.

Outer-Approximation (OA)

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
  Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.
  Solve the current NLP restriction for y^K.
  if NLP restriction for y^K infeasible **then**
    Solve the infeasibility subproblem for y^K.
  else
    if $f(x^K, y^K) < f^*$ **then**
      $f^* = f(x^K, y^K)$, $(x^*, y^*) = (x^K, y^K)$.
    end if
  end if
  Generate linearization cuts, update MILP relax.
  $K = K + 1$.

Outer-Approximation (OA)

$K = 1$, define an initial MILP relaxation, $f^* = +\infty$, LB$= -\infty$.
while $f^* \neq$ LB **do**
  Solve the current MILP relaxation (obtaining (x^K, y^K)) and update LB.
  Solve the current NLP restriction for y^K.
  if NLP restriction for y^K infeasible **then**
    Solve the infeasibility subproblem for y^K.
  else
    if $f(x^K, y^K) < f^*$ **then**
      $f^* = f(x^K, y^K)$, $(x^*, y^*) = (x^K, y^K)$.
    end if
  end if
  Generate linearization cuts, update MILP relax.
  $K = K + 1$.
end while
return (x^*, y^*)

NLP restriction and Feasibility subproblem

NLP restriction for a fixed y^k:

$$\begin{aligned} \min f(x, y^k) & & \\ g(x, y^k) & \leq & 0 \\ x & \in & X. \end{aligned}$$

NLP restriction and Feasibility subproblem

NLP restriction for a fixed y^k:

$$\begin{aligned} \min f(x, y^k) & & \\ g(x, y^k) &\leq 0 \\ x &\in X. \end{aligned}$$

Feasibility subproblem for a fixed y^k:

$$\begin{aligned} \min u & & \\ g(x, y^k) &\leq u \\ x &\in X \\ u &\in \mathbb{R}_+. \end{aligned}$$

Worst-case complexity of outer approximation

Hijazi, Bonami, Ouorou. An Outer-Inner Approximation for separable MINLPs, INFORMS Journal on Computing (2014)

$$
\begin{aligned}
\min 0 & \\
\sum_{i=1}^{n} \left(x_i - \frac{1}{2}\right)^2 &\leq \frac{n-1}{4} \\
x &\in \{0,1\}^n
\end{aligned}
$$

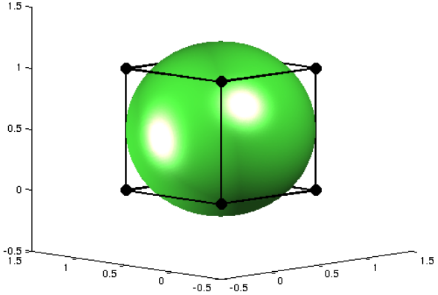


Figure: Source Belotti et al. (2013)

Outline

Generalized Benders Decomposition (GBD)

Geoffrion, 1972.

Generalized Benders Decomposition (GBD)

Geoffrion, 1972.
Similar to OA, but with a different MILP relaxation, i.e.,

Generalized Benders Decomposition (GBD)

Geoffrion, 1972.
Similar to OA, but with a different MILP relaxation, i.e.,

- $x \in X$ is relaxed.

Generalized Benders Decomposition (GBD)

Geoffrion, 1972.
Similar to OA, but with a different MILP relaxation, i.e.,

- $x \in X$ is relaxed.
- $I^k = \{i \mid g(x^k, y^k) = 0, 1 \leq i \leq m\} \; \forall k = 1, \dots, K$.

Generalized Benders Decomposition (GBD)

Geoffrion, 1972.
Similar to OA, but with a different MILP relaxation, i.e.,

- $x \in X$ is relaxed.
- $I^k = \{i \mid g(x^k, y^k) = 0, 1 \leq i \leq m\} \; \forall k = 1, \ldots, K$.

Proposition

Given the same set of K subproblems, the LB provided by the MILP relaxation of OA is $\geq$ of the one provided by the MILP relaxation of GDB.

Generalized Benders Decomposition (GBD)

Geoffrion, 1972.
Similar to OA, but with a different MILP relaxation, i.e.,

- $x \in X$ is relaxed.
- $I^k = \{i \mid g(x^k, y^k) = 0, 1 \leq i \leq m\} \; \forall k = 1, \ldots, K$.

Proposition

Given the same set of K subproblems, the LB provided by the MILP relaxation of OA is $\geq$ of the one provided by the MILP relaxation of GDB.

Proof.

(Sketch of) It can be shown that the constraints of GDB MILP relaxation are surrogate of the ones of OA MILP relaxation (see, Quesada and Grossmann, 1992). □

Outline

Extended Cutting Plane (ECP)

Westerlund and Pettersson, 1995.

Extended Cutting Plane (ECP)

Westerlund and Pettersson, 1995.

1: $K = 1$, obtain an initial MILP relaxation.

Westerlund and Pettersson, 1995.

1: $K = 1$, obtain an initial MILP relaxation.
2: **while do**
3: Solve the MILP relaxation obtaining (x^K, y^K).

Extended Cutting Plane (ECP)

Westerlund and Pettersson, 1995.

$K = 1$, obtain an initial MILP relaxation.
while do
    Solve the MILP relaxation obtaining (x^K, y^K).
    if no constraint is violated by (x^K, y^K) **then**

Extended Cutting Plane (ECP)

Westerlund and Pettersson, 1995.

1: $K = 1$, obtain an initial MILP relaxation.
2: **while do**
3: Solve the MILP relaxation obtaining (x^K, y^K).
4: **if** no constraint is violated by (x^K, y^K) **then**
5: **return** (x^K, y^K) (optimal solution).

Extended Cutting Plane (ECP)

Westerlund and Pettersson, 1995.

$K = 1$, obtain an initial MILP relaxation.
while do
  Solve the MILP relaxation obtaining (x^K, y^K).
  if no constraint is violated by (x^K, y^K) **then**
    return (x^K, y^K) (optimal solution).
  else
    Generate (some) new linearization constraints from (x^K, y^K) and update MILP relaxation.

Extended Cutting Plane (ECP)

Westerlund and Pettersson, 1995.

```
K = 1, obtain an initial MILP relaxation.
while do
    Solve the MILP relaxation obtaining (x^K, y^K).
    if no constraint is violated by (x^K, y^K) then
        return (x^K, y^K) (optimal solution).
    else
        Generate (some) new linearization constraints from (x^K, y^K)
        and update MILP relaxation.
    end if
    K = K + 1.
end while
```

More iterations needed wrt OA.

Outline

LP/NLP-based Branch-and-Bound (QG)

Quesada and Grossmann, 1992.

1: Obtain an initial MILP relaxation.

LP/NLP-based Branch-and-Bound (QG)

Quesada and Grossmann, 1992.

1: Obtain an initial MILP relaxation.
2: Solve the MILP relaxation through BB for MILP, but, anytime a MILP feasible solution is found

LP/NLP-based Branch-and-Bound (QG)

Quesada and Grossmann, 1992.

1: Obtain an initial MILP relaxation.
2: Solve the MILP relaxation through BB for MILP, but, anytime a MILP feasible solution is found
   - Solve NLP restriction.

LP/NLP-based Branch-and-Bound (QG)

Quesada and Grossmann, 1992.

1: Obtain an initial MILP relaxation.
2: Solve the MILP relaxation through BB for MILP, but, anytime a MILP feasible solution is found
   - Solve NLP restriction.
   - Generate new linearization constraints.

LP/NLP-based Branch-and-Bound (QG)

Quesada and Grossmann, 1992.

1: Obtain an initial MILP relaxation.
2: Solve the MILP relaxation through BB for MILP, but, anytime a MILP feasible solution is found
   - Solve NLP restriction.
   - Generate new linearization constraints.
   - Update open MILP relaxation subproblems.

LP/NLP-based Branch-and-Bound (QG)

Quesada and Grossmann, 1992.

1: Obtain an initial MILP relaxation.
2: Solve the MILP relaxation through BB for MILP, but, anytime a MILP feasible solution is found
   - Solve NLP restriction.
   - Generate new linearization constraints.
   - Update open MILP relaxation subproblems.

Link OA, but only 1 MILP relaxation is solved, and updated in the tree search.

LP/NLP-based Branch-and-Bound (QG)

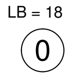
Quesada and Grossmann, 1992.

1: Obtain an initial MILP relaxation.
2: Solve the MILP relaxation through BB for MILP, but, anytime a MILP feasible solution is found
   - Solve NLP restriction.
   - Generate new linearization constraints.
   - Update open MILP relaxation subproblems.

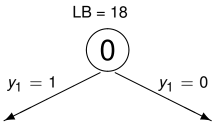
Link OA, but only 1 MILP relaxation is solved, and updated in the tree search.

Finite convergence as for BB.

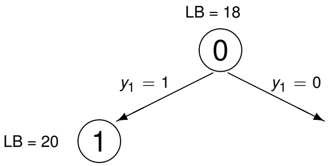
LP/NLP-based Branch-and-Bound (QG)



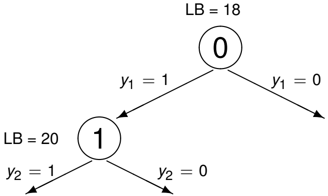
LP/NLP-based Branch-and-Bound (QG)



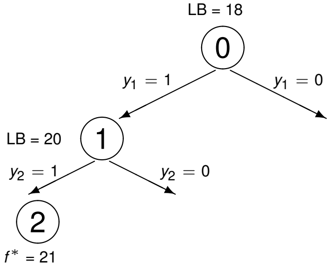
LP/NLP-based Branch-and-Bound (QG)



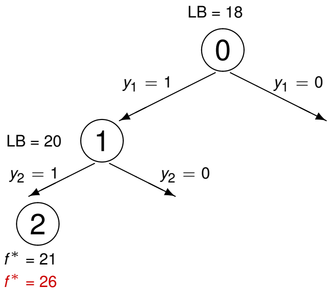
LP/NLP-based Branch-and-Bound (QG)



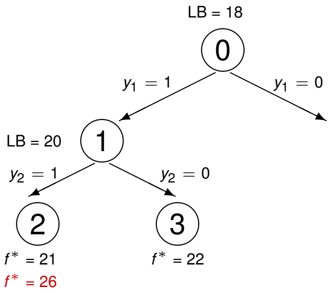
LP/NLP-based Branch-and-Bound (QG)



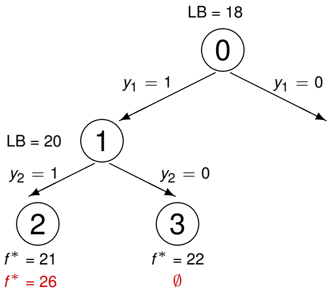
LP/NLP-based Branch-and-Bound (QG)



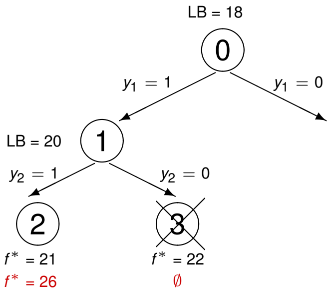
LP/NLP-based Branch-and-Bound (QG)



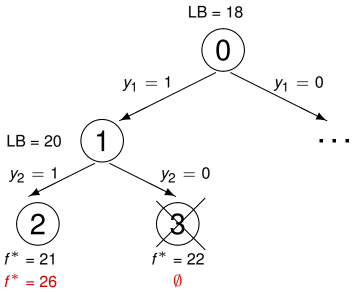
LP/NLP-based Branch-and-Bound (QG)



LP/NLP-based Branch-and-Bound (QG)



LP/NLP-based Branch-and-Bound (QG)



Outline

Hybrid Algorithms (Hyb)

For example, Bonami et al., 2008 (**BONMIN**).

Very similar to Quesada and Grossmann, 1992, but NLP solved not only when the node is integer feasible but also, for example, any 10 nodes.

Hybrid Algorithms (Hyb)

For example, Bonami et al., 2008 (**BONMIN**).

Very similar to Quesada and Grossmann, 1992, but NLP solved not only when the node is integer feasible but also, for example, any 10 nodes.

Pros : more “nonlinear” information added to the MILP relaxation.

Hybrid Algorithms (Hyb)

For example, Bonami et al., 2008 (**BONMIN**).

Very similar to Quesada and Grossmann, 1992, but NLP solved not only when the node is integer feasible but also, for example, any 10 nodes.

Pros : more "nonlinear" information added to the MILP relaxation.
Cons : More NLP solved.

Alternative,

Abhishek et al., 2010 (**FILMINT**).
Very similar to Quesada and Grossmann, 1992, but add linearization cuts not only when the node is integer feasible (different strategies).

Hybrid Algorithms (Hyb)

For example, Bonami et al., 2008 (**BONMIN**).

Very similar to Quesada and Grossmann, 1992, but NLP solved not only when the node is integer feasible but also, for example, any 10 nodes.

Pros : more “nonlinear” information added to the MILP relaxation.
Cons : More NLP solved.

Alternative,

Abhishek et al., 2010 (**FILMINT**).
Very similar to Quesada and Grossmann, 1992, but add linearization cuts not only when the node is integer feasible (different strategies).
Pros : more “nonlinear” information added to the MILP relaxation.

Hybrid Algorithms (Hyb)

For example, Bonami et al., 2008 (**BONMIN**).

Very similar to Quesada and Grossmann, 1992, but NLP solved not only when the node is integer feasible but also, for example, any 10 nodes.

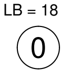
Pros : more "nonlinear" information added to the MILP relaxation.
Cons : More NLP solved.

Alternative,

Abhishek et al., 2010 (**FILMINT**).
Very similar to Quesada and Grossmann, 1992, but add linearization cuts not only when the node is integer feasible (different strategies).
Pros : more "nonlinear" information added to the MILP relaxation.
Cons : MILP relaxation more difficult to solve.

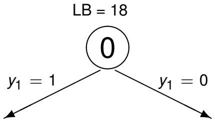
LP/NLP-based Branch-and-Bound (QG)

E.g., Bonami et al., 2008 with NLP every 2 nodes.



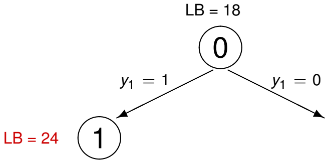
LP/NLP-based Branch-and-Bound (QG)

E.g., Bonami et al., 2008 with NLP every 2 nodes.

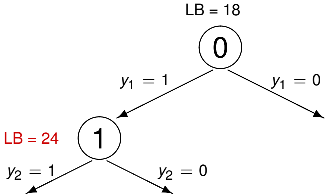


LP/NLP-based Branch-and-Bound (QG)

E.g., Bonami et al., 2008 with NLP every 2 nodes.

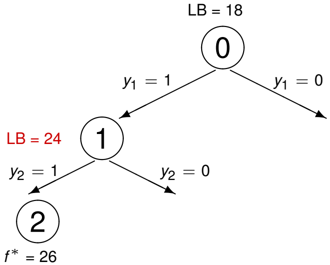


E.g., Bonami et al., 2008 with NLP every 2 nodes.



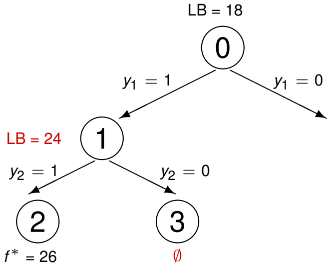
LP/NLP-based Branch-and-Bound (QG)

E.g., Bonami et al., 2008 with NLP every 2 nodes.



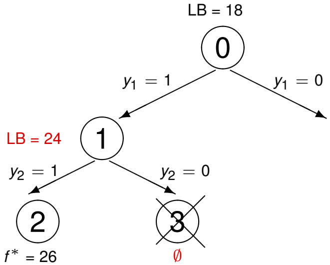
LP/NLP-based Branch-and-Bound (QG)

E.g., Bonami et al., 2008 with NLP every 2 nodes.



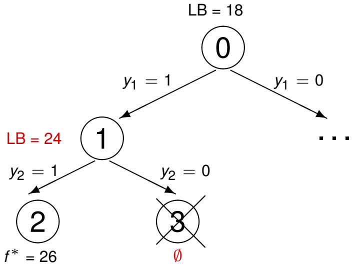
LP/NLP-based Branch-and-Bound (QG)

E.g., Bonami et al., 2008 with NLP every 2 nodes.



LP/NLP-based Branch-and-Bound (QG)

E.g., Bonami et al., 2008 with NLP every 2 nodes.



Number of subproblems solved

	# MILP	# NLP	note
BB	0	# nodes	
OA	# iterations	# iterations	
GBD	# iterations	# iterations	[1]
ECP	# iterations	0	
QG	1	1 + # explored MILP solutions	
Hyb ALL10	1	1 + # explored MILP solutions	[2]
Hyb CMUIBM	1	[# explored MILP solutions,# nodes]	

Table: Number of MILP and NLP subproblems solved by each algorithm.

[1] weaker lower bound w.r.t. OA, MILP with less constraints than the one of OA
[2] stronger lower bound w.r.t. QG ,MILP with more constraints than the one of QG

References

- C. D'Ambrosio, A. Lodi. Mixed Integer Non-Linear Programming Tools: a Practical Overview, **4OR**: A Quarterly Journal of Operations Research, 9 (4), pp. 329-349, 2011.
- P. Bonami, M. Kilinç, J. Linderoth, Algorithms and software for convex mixed integer nonlinear programs. In: Lee J, Leyffer S (eds) **Mixed integer nonlinear programming**. Springer, pp. 1–39, 2012.
- C. D'Ambrosio, A. Lodi. Mixed integer nonlinear programming tools: an updated practical overview, **Annals of Operations Research**, 204, pp. 301–320, 2013.
- P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, A. Mahajan, Mixed-integer nonlinear optimization. **Acta Numerica**, 22, pp. 1–131, 2013.
- J. Kronqvist, D. E. Bernal, A. Lundell, I. E. Grossmann, A review and comparison of solvers for convex MINLP, **Optimization and Engineering**, 20 (2), pp. 397–455, 2019.

Outline

Reminder: Convex functions and some properties

Properties:

- $\forall x_1, x_2 \in X, \forall t \in [0,1]: \qquad f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$.

Reminder: Convex functions and some properties

Properties:

- $\forall x_1, x_2 \in X, \forall t \in [0,1]: \qquad f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$.
- A differentiable function of one variable is convex on an interval if and only if the function lies above all of its tangents $f(x) \geq f(y) + f'(y)(x-y)$

Reminder: Convex functions and some properties

Properties:

- $\forall x_1, x_2 \in X, \forall t \in [0,1]: \qquad f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$
- A differentiable function of one variable is convex on an interval if and only if the function lies above all of its tangents $f(x) \geq f(y) + f'(y)(x-y)$
- A twice continuously differentiable function of several variables is convex on a convex set if and only if its Hessian matrix of second partial derivatives is positive semidefinite on the interior of the convex set

Reminder: Convex functions and some properties

Properties:

- $\forall x_1, x_2 \in X, \forall t \in [0,1]: \qquad f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$
- A differentiable function of one variable is convex on an interval if and only if the function lies above all of its tangents $f(x) \geq f(y) + f'(y)(x-y)$
- A twice continuously differentiable function of several variables is convex on a convex set if and only if its Hessian matrix of second partial derivatives is positive semidefinite on the interior of the convex set
- Any local minimum of a convex function is also a global minimum. A strictly convex function will have at most one global minimum

Reminder: Convex functions and some properties

Properties:

- $\forall x_1, x_2 \in X, \forall t \in [0,1]: \qquad f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$.
- A differentiable function of one variable is convex on an interval if and only if the function lies above all of its tangents $f(x) \geq f(y) + f'(y)(x-y)$
- A twice continuously differentiable function of several variables is convex on a convex set if and only if its Hessian matrix of second partial derivatives is positive semidefinite on the interior of the convex set
- Any local minimum of a convex function is also a global minimum. A strictly convex function will have at most one global minimum
- If f is concave, $-f$ is convex

Evidently convex terms

Simple convex functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (concave too)

Evidently convex terms

Simple convex functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (concave too)
- Exponential: e^{ax} over $\mathbb{R}$ for any $a \in \mathbb{R}$

Evidently convex terms

Simple convex functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (concave too)
- Exponential: e^{ax} over $\mathbb{R}$ for any $a \in \mathbb{R}$
- Power: x^p over $(0, +\infty)$ for $p \geq 1$ or $p \leq 0$

Evidently convex terms

Simple convex functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (concave too)
- Exponential: e^{ax} over $\mathbb{R}$ for any $a \in \mathbb{R}$
- Power: x^p over $(0, +\infty)$ for $p \geq 1$ or $p \leq 0$
- Powers of absolute value: $|x|^p$ over $\mathbb{R}$ for $p \geq 1$

Evidently convex terms

Simple convex functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (concave too)
- Exponential: e^{ax} over $\mathbb{R}$ for any $a \in \mathbb{R}$
- Power: x^p over $(0, +\infty)$ for $p \geq 1$ or $p \leq 0$
- Powers of absolute value: $|x|^p$ over $\mathbb{R}$ for $p \geq 1$
- Negative entropy: $x \ln x$ over $(0, +\infty)$

Evidently convex terms

Simple convex functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (concave too)
- Exponential: e^{ax} over $\mathbb{R}$ for any $a \in \mathbb{R}$
- Power: x^p over $(0, +\infty)$ for $p \geq 1$ or $p \leq 0$
- Powers of absolute value: $|x|^p$ over $\mathbb{R}$ for $p \geq 1$
- Negative entropy: $x \ln x$ over $(0, +\infty)$

Simple concave functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (convex too)

Evidently convex terms

Simple convex functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (concave too)
- Exponential: e^{ax} over $\mathbb{R}$ for any $a \in \mathbb{R}$
- Power: x^p over $(0, +\infty)$ for $p \geq 1$ or $p \leq 0$
- Powers of absolute value: $|x|^p$ over $\mathbb{R}$ for $p \geq 1$
- Negative entropy: $x \ln x$ over $(0, +\infty)$

Simple concave functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (convex too)
- Power: x^p over $(0, +\infty)$ for $0 \leq p \leq 1$

Evidently convex terms

Simple convex functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (concave too)
- Exponential: e^{ax} over $\mathbb{R}$ for any $a \in \mathbb{R}$
- Power: x^p over $(0, +\infty)$ for $p \geq 1$ or $p \leq 0$
- Powers of absolute value: $|x|^p$ over $\mathbb{R}$ for $p \geq 1$
- Negative entropy: $x \ln x$ over $(0, +\infty)$

Simple concave functions:

- Affine: $ax + b$ over $\mathbb{R}$ for any $a, b \in \mathbb{R}$ (convex too)
- Power: x^p over $(0, +\infty)$ for $0 \leq p \leq 1$
- Logarithm: $\ln x$ over $(0, +\infty)$

Operations Preserving Convexity

- Positive Scaling, e.g., λf

Operations Preserving Convexity

- Positive Scaling, e.g., λf
- Sum, e.g., $f_1 + f_2$

Operations Preserving Convexity

- Positive Scaling, e.g., λf
- Sum, e.g., $f_1 + f_2$
- Composition with affine function, e.g., $f(Ax + b)$

Operations Preserving Convexity

- Positive Scaling, e.g., λf
- Sum, e.g., $f_1 + f_2$
- Composition with affine function, e.g., $f(Ax + b)$
- Pointwise maximum and supremum, e.g., $\max\{f_1(x), f_2(x), \ldots, f_m(x)\}$

Operations Preserving Convexity

- Positive Scaling, e.g., λf
- Sum, e.g., $f_1 + f_2$
- Composition with affine function, e.g., $f(Ax + b)$
- Pointwise maximum and supremum, e.g., $\max\{f_1(x), f_2(x), \ldots, f_m(x)\}$
- Composition, e.g., $f_2(f_1(x))$ if f_1 convex and f_2 nondecreasing and convex or f_1 concave and f_2 nonincreasing and convex

Operations Preserving Convexity

- Positive Scaling, e.g., λf
- Sum, e.g., $f_1 + f_2$
- Composition with affine function, e.g., $f(Ax + b)$
- Pointwise maximum and supremum, e.g., $\max\{f_1(x), f_2(x), \ldots, f_m(x)\}$
- Composition, e.g., $f_2(f_1(x))$ if f_1 convex and f_2 nondecreasing and convex or f_1 concave and f_2 nonincreasing and convex
- Minimization, e.g., $\inf_{z \in C} f(x, z)$

Outline

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf
- Juniper: https://www.github.com/lanl-ansi/juniper.jl

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf
- Juniper: https://www.github.com/lanl-ansi/juniper.jl
- LAGO: https://projects.coin-or.org/LaGO

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf
- Juniper: https://www.github.com/lanl-ansi/juniper.jl
- LAGO: https://projects.coin-or.org/LaGO
- MINLPBB: https://www-unix.mcs.anl.gov/~leyffer/solvers.htm

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf
- Juniper: https://www.github.com/lanl-ansi/juniper.jl
- LAGO: https://projects.coin-or.org/LaGO
- MINLPBB: https://www-unix.mcs.anl.gov/~leyffer/solvers.htm
- MINOTAUR: https://wiki.mcs.anl.gov/minotaur

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf
- Juniper: https://www.github.com/lanl-ansi/juniper.jl
- LAGO: https://projects.coin-or.org/LaGO
- MINLPBB: https://www-unix.mcs.anl.gov/~leyffer/solvers.htm
- MINOTAUR: https://wiki.mcs.anl.gov/minotaur
- Muriqui: http://www.wendelmelo.net/software

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf
- Juniper: https://www.github.com/lanl-ansi/juniper.jl
- LAGO: https://projects.coin-or.org/LaGO
- MINLPBB: https://www-unix.mcs.anl.gov/~leyffer/solvers.htm
- MINOTAUR: https://wiki.mcs.anl.gov/minotaur
- Muriqui: http://www.wendelmelo.net/software
- Pavito: https://www.github.com/juliaopt/pavito.jl

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf
- Juniper: https://www.github.com/lanl-ansi/juniper.jl
- LAGO: https://projects.coin-or.org/LaGO
- MINLPBB: https://www-unix.mcs.anl.gov/~leyffer/solvers.htm
- MINOTAUR: https://wiki.mcs.anl.gov/minotaur
- Muriqui: http://www.wendelmelo.net/software
- Pavito: https://www.github.com/juliaopt/pavito.jl
- SBB: https://www.gams.com/latest/docs/S_SBB.html

Convex MINLP Solvers

- ALPHA-ECP: `https://www.gams.com/latest/docs/S_ALPHAECP.html`
- AOA: `https://www.aimms.com/english/developers/resources/solvers/aoa`
- BONMIN: `https://projects.coin-or.org/Bonmin`
- DICOPT: `https://www.gams.com/24.8/docs/solvers/dicopt/index.html`
- FilMINT: `https://www.mcs.anl.gov/~leyffer/papers/fm.pdf`
- Juniper: `https://www.github.com/lanl-ansi/juniper.jl`
- LAGO: `https://projects.coin-or.org/LaGO`
- MINLPBB: `https://www-unix.mcs.anl.gov/~leyffer/solvers.htm`
- MINOTAUR: `https://wiki.mcs.anl.gov/minotaur`
- Muriqui: `http://www.wendelmelo.net/software`
- Pavito: `https://www.github.com/juliaopt/pavito.jl`
- SBB: `https://www.gams.com/latest/docs/S_SBB.html`
- SHOT: `https://github.com/coin-or/shot`
- ...

Need value of the function, its first and its second derivative at a given point (x^*, y^*).

Convex MINLP Solvers

- ALPHA-ECP: `https://www.gams.com/latest/docs/S_ALPHAECP.html`
- AOA: `https://www.aimms.com/english/developers/resources/solvers/aoa`
- BONMIN: `https://projects.coin-or.org/Bonmin`
- DICOPT: `https://www.gams.com/24.8/docs/solvers/dicopt/index.html`
- FilMINT: `https://www.mcs.anl.gov/~leyffer/papers/fm.pdf`
- Juniper: `https://www.github.com/lanl-ansi/juniper.jl`
- LAGO: `https://projects.coin-or.org/LaGO`
- MINLPBB: `https://www-unix.mcs.anl.gov/~leyffer/solvers.htm`
- MINOTAUR: `https://wiki.mcs.anl.gov/minotaur`
- Muriqui: `http://www.wendelmelo.net/software`
- Pavito: `https://www.github.com/juliaopt/pavito.jl`
- SBB: `https://www.gams.com/latest/docs/S_SBB.html`
- SHOT: `https://github.com/coin-or/shot`
- ...

Need value of the function, its first and its second derivative at a given point (x^*, y^*). Possible source of errors!

Convex MINLP Solvers

- ALPHA-ECP: https://www.gams.com/latest/docs/S_ALPHAECP.html
- AOA: https://www.aimms.com/english/developers/resources/solvers/aoa
- BONMIN: https://projects.coin-or.org/Bonmin
- DICOPT: https://www.gams.com/24.8/docs/solvers/dicopt/index.html
- FilMINT: https://www.mcs.anl.gov/~leyffer/papers/fm.pdf
- Juniper: https://www.github.com/lanl-ansi/juniper.jl
- LAGO: https://projects.coin-or.org/LaGO
- MINLPBB: https://www-unix.mcs.anl.gov/~leyffer/solvers.htm
- MINOTAUR: https://wiki.mcs.anl.gov/minotaur
- Muriqui: http://www.wendelmelo.net/software
- Pavito: https://www.github.com/juliaopt/pavito.jl
- SBB: https://www.gams.com/latest/docs/S_SBB.html
- SHOT: https://github.com/coin-or/shot
- ...

Need value of the function, its first and its second derivative at a given point (x^*, y^*). Possible source of errors! $\rightarrow$ **Modeling Languages** !

Modeling Languages

Modeling languages, e.g., AMPL, GAMS, JUMP.

Modeling Languages

Modeling languages, e.g., AMPL, GAMS, JUMP.
Example:

```
param pi := 3.142;
param N;
set VARS ordered := {1..N};
param Umax default 100;
param U {j in VARS};
param a {j in VARS};
param b {j in VARS};
param c {j in VARS};
param d {j in VARS};
param w{VARS};
param C;
```

Modeling languages, e.g., AMPL, GAMS, JUMP.
Example:

```
param pi := 3.142;
param N;
set VARS ordered := {1..N};
param Umax default 100;
param U {j in VARS};
param a {j in VARS};
param b {j in VARS};
param c {j in VARS};
param d {j in VARS};
param w{VARS};
param C;
var x {j in VARS} >= 0, <= U[j], integer;
```

Modeling Languages

Modeling languages, e.g., AMPL, GAMS, JUMP.
Example:

```
param pi := 3.142;
param N;
set VARS ordered := {1..N};
param Umax default 100;
param U {j in VARS};
param a {j in VARS};
param b {j in VARS};
param c {j in VARS};
param d {j in VARS};
param w{VARS};
param C;
var x {j in VARS} >= 0, <= U[j], integer;

maximize Total_Profit:
     sum {j in VARS} c[j]/(1+b[j]*exp(-a[j]*(x[j]+d[j])));
```

Modeling languages, e.g., AMPL, GAMS, JUMP.
Example:

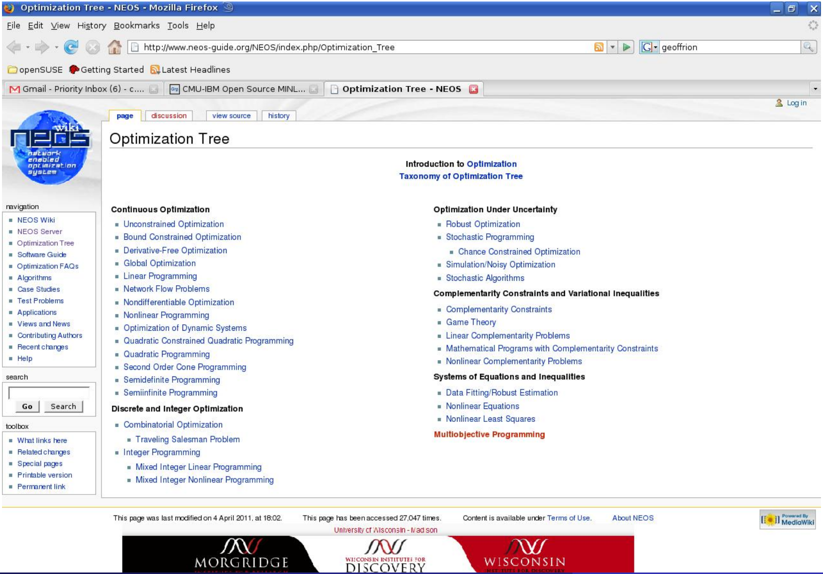
```
param pi := 3.142;
param N;
set VARS ordered := {1..N};
param Umax default 100;
param U {j in VARS};
param a {j in VARS};
param b {j in VARS};
param c {j in VARS};
param d {j in VARS};
param w{VARS};
param C;
var x {j in VARS} >= 0, <= U[j], integer;

maximize Total_Profit:
     sum {j in VARS} c[j]/(1+b[j]*exp(-a[j]*(x[j]+d[j])));
subject to KP_constraint:  sum{j in VARS} w[j]*x[j] <= C;
```

Neos

NEOS: http://www.neos-server.org/neos/.

Neos

NEOS: `http://www.neos-server.org/neos/`.



MINLP Libraries

- CMU/IBM: 23 different kind of MINLP problems

  http://www.minlp.org

MINLP Libraries

- CMU/IBM: 23 different kind of MINLP problems

  http://www.minlp.org

- MacMINLP: 53 instances

  http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP

MINLP Libraries

- CMU/IBM: 23 different kind of MINLP problems
  http://www.minlp.org
- MacMINLP: 53 instances
  http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP
- MINLPlib: 1626 instances
  http://www.minlplib.org/

MINLP Libraries

- CMU/IBM: 23 different kind of MINLP problems

  http://www.minlp.org
- MacMINLP: 53 instances

  http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP
- MINLPlib: 1626 instances

  http://www.minlplib.org/
- QPLIB: 367 instances

  http://qplib.zib.de/
- ...

Recall...

- MINLP can be theoretically an “undecidable” problem and it is in practice much more difficult than MILP

Recall...

- MINLP can be theoretically an “undecidable” problem and it is in practice much more difficult than MILP
- When modeling a problem, do not forget to define simple bounds on each variable

Recall...

- MINLP can be theoretically an “undecidable” problem and it is in practice much more difficult than MILP
- When modeling a problem, do not forget to define simple bounds on each variable
- Exactly reformulate nonlinear term, if possible

Recall...

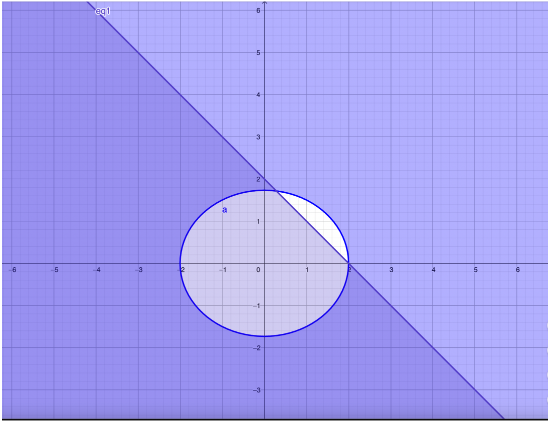
- MINLP can be theoretically an “undecidable” problem and it is in practice much more difficult than MILP
- When modeling a problem, do not forget to define simple bounds on each variable
- Exactly reformulate nonlinear term, if possible
- Several tailored methods for convex MINLPs (not exact for nonconvex MINLPs)

Recall...

- MINLP can be theoretically an “undecidable” problem and it is in practice much more difficult than MILP
- When modeling a problem, do not forget to define simple bounds on each variable
- Exactly reformulate nonlinear term, if possible
- Several tailored methods for convex MINLPs (not exact for nonconvex MINLPs)
- Identifying convexity is, in general, very difficult

Outline

Numerical example: Convex NLP



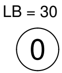
$$3x_1^2 + 4x_2^2 \geq 12$$
$$x_1 + x_2 \leq 2$$

MINLP branch-and-bound with local NLP solver

Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.

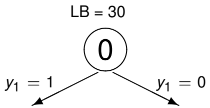
MINLP branch-and-bound with local NLP solver

Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.



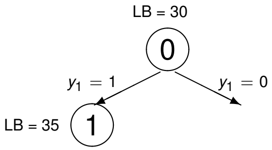
MINLP branch-and-bound with local NLP solver

Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.



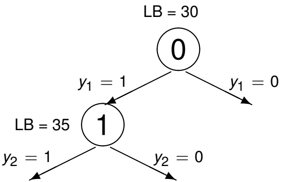
MINLP branch-and-bound with local NLP solver

Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.



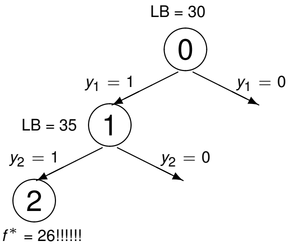
MINLP branch-and-bound with local NLP solver

Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.

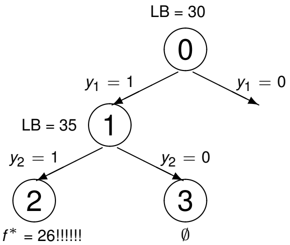


MINLP branch-and-bound with local NLP solver

Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.

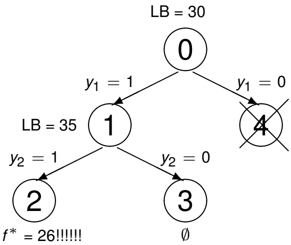


<u>Branch-and-bound</u> algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.

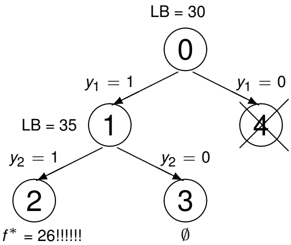


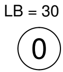
MINLP branch-and-bound with local NLP solver

<u>Branch-and-bound</u> algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.



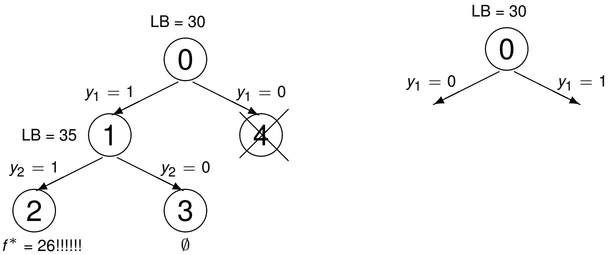
Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers → local optimum
No valid bound for nonconvex MINLPs.





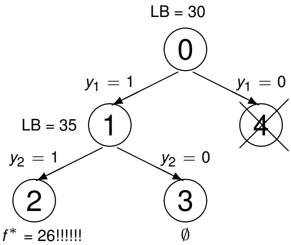
MINLP branch-and-bound with local NLP solver

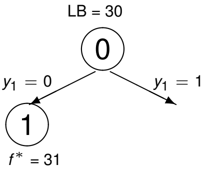
Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.



MINLP branch-and-bound with local NLP solver

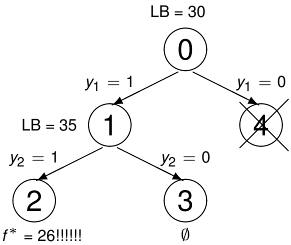
Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.

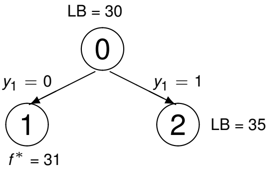




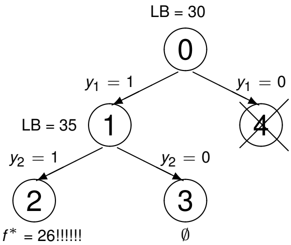
MINLP branch-and-bound with local NLP solver

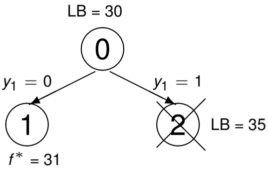
Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers $\rightarrow$ local optimum
No valid bound for nonconvex MINLPs.





Branch-and-bound algorithm: solve continuous (NLP) relaxation at each node of the search tree and branch on variables.
NLP solver used:
Local NLP solvers → local optimum
No valid bound for nonconvex MINLPs.





Outer Approximation and nonconvex MINLPs

Several methods for convex MINLPs use **Outer Approximation** cuts (Duran and Grossman, 1986) which are not exact for nonconvex MINLPs.

$$g_i(x) \leq 0 \quad \rightarrow \quad g_i(x^k) + \nabla g_i(x^k)^T \left(\; x - x^k \; \right) \leq 0$$

where $\nabla g(x^k)$ is the Jacobian of $g(x)$ evaluated at point (x^k).