# Mixed-Integer Nonlinear Optimisation: Branch & Bound

Benoit Chachuat
b.chachuat@imperial.ac.uk

Ruth Misener
r.misener@imperial.ac.uk

Computational Optimisation Group
Centre for Process Systems Engineering
Department of Computing


08 May 2017

# Arnold Neumaier's Classification[1]

**Incomplete Methods:**

- Use clever heuristics for searching
- No safeguard if search gets stuck in a local optimum

➥ Classical descent methods

**Asymptotically Complete Methods:**

- Reach a global optimum with probability one, given infinite run time and exact computations
- No way of knowing when a global optimizer has been found

➥ Many stochastic and deterministic search methods

**Complete and Rigorous Methods:**

- Reach a global optimum with certainty and within given tolerance in finite time
- Complete Methods: Assuming exact computations
- Rigorous Methods: Even in the presence of rounding errors

➥ Deterministic search methods only

[1] Neumaier, *Acta Numerica*, **13**:271-369, 2004

# Branch & Bound Methods

## Principle

**Divide and Conquer:** Partition a problem recursively into subproblems which are sooner or later eliminated, after showing that a solution superior to the best one found so far cannot be obtained for those subproblems

# Branch & Bound Methods

## Principle

**Divide and Conquer:** Partition a problem recursively into subproblems which are sooner or later eliminated, after showing that a solution superior to the best one found so far cannot be obtained for those subproblems

- Partitioning relies on the application of branching rules
- Elimination relies on the computation of upper/lower bounds for the subproblems

# Branch & Bound Methods

## Principle

**Divide and Conquer:** Partition a problem recursively into subproblems which are sooner or later eliminated, after showing that a solution superior to the best one found so far cannot be obtained for those subproblems

- Partitioning relies on the application of branching rules
- Elimination relies on the computation of upper/lower bounds for the subproblems

**Sanity Check:** Where does branch & bound fit in Neumaier's classification?

# Branch & Bound Methods

## Principle

**Divide and Conquer:** Partition a problem recursively into subproblems which are sooner or later eliminated, after showing that a solution superior to the best one found so far cannot be obtained for those subproblems

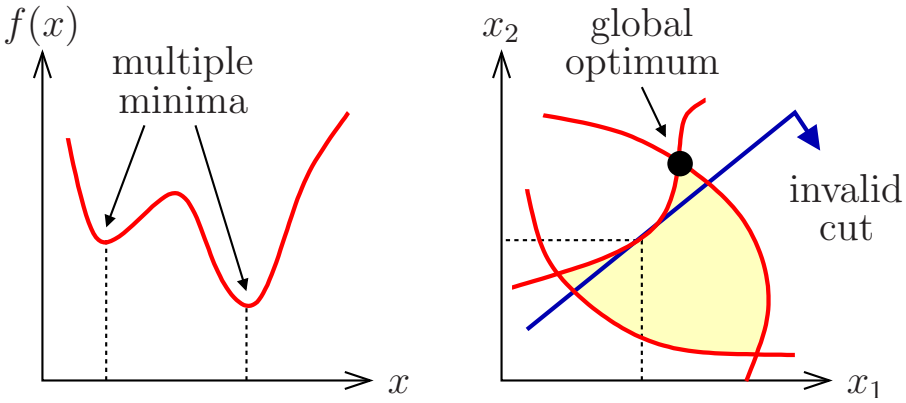
- Partitioning relies on the application of branching rules
- Elimination relies on the computation of upper/lower bounds for the subproblems

**Sanity Check:** Where does branch & bound fit in Neumaier's classification?

- Depends on the implementation;
- Using a convex MINLP method, *e.g.*, in DICOPT, SBB, or Bonmin: incomplete method for MINLP with nonconvex nonlinearities;
- Most global solver implementations, *e.g.*, ANTIGONE, Couenne, SCIP, are complete (not rigorous)

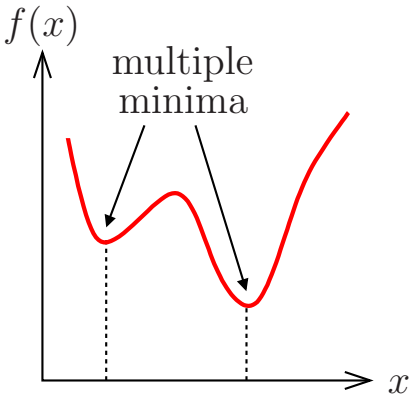
# Nonconvex MINLP Problems

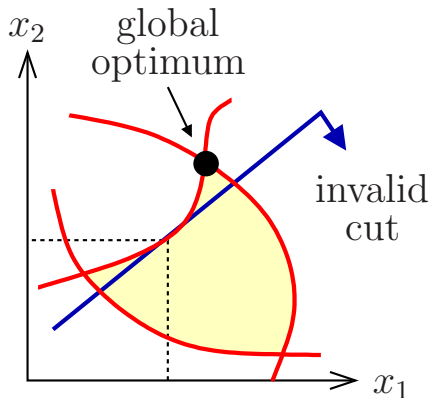
## Effects of Nonconvexity:



# Nonconvex MINLP Problems

## Effects of Nonconvexity:



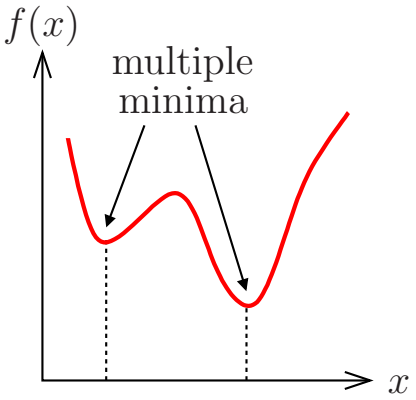



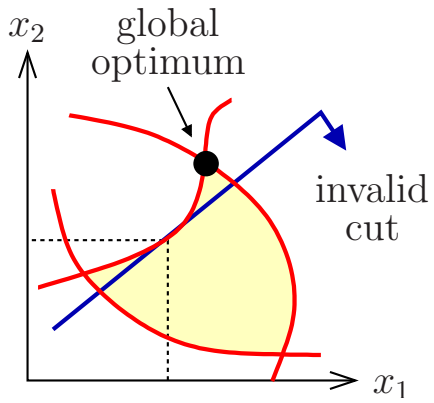

## Heuristic Approaches:

1. Use convex MINLP techniques, assuming valid lower bound
2. Add slacks to the relaxations/ linearizations

# Nonconvex MINLP Problems

## Effects of Nonconvexity:




## Heuristic Approaches:

1. Use convex MINLP techniques, assuming valid lower bound
2. Add slacks to the relaxations/ linearizations

**Complete Approaches:** Based mostly on spatial branch & bound

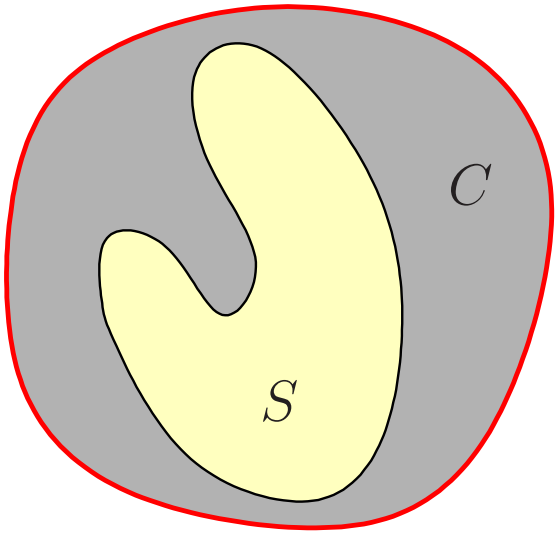
- Gmin-$\alpha$BB (Adjiman *et al*, 1997, 2000)
- Reformulation/relaxation (Smith & Pantelides, 1999; Zamora & Grossmann, 1999)
- Solvers: ANTIGONE (Misener & Floudas, 2014), BARON (Tawarmalani & Sahinidis, 2005), Couenne (Belotti et al., 2009), LINDO (Lin & Schrage, 2009), SCIP (Vigerske, 2012)

# Branch & Bound Building Blocks

## Convex Relaxation (Outer-Approximation)

Given a (nonconvex) set $S$, the set $C$ is a convex relaxation of $S$ if

1. $C$ is convex
2. $S \subseteq C$

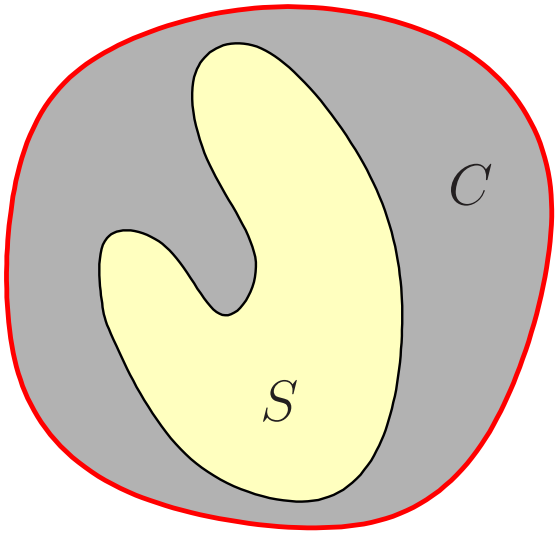


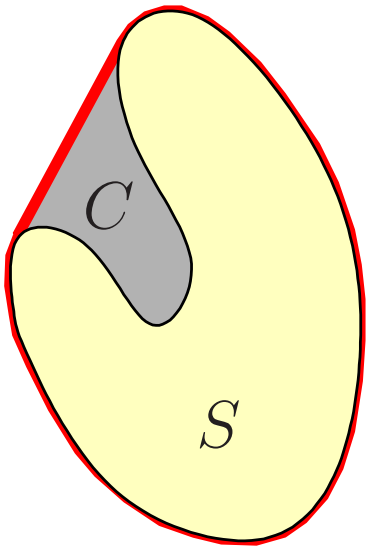

# Branch & Bound Building Blocks

## Convex Relaxation (Outer-Approximation)

Given a (nonconvex) set $S$, the set $C$ is a convex relaxation of $S$ if

1. $C$ is convex
2. $S \subseteq C$




## Convex Hull

Given a set $S$, $C$ is the convex hull of $S$ if

1. $C$ is a convex outer-approximation of $S$
2. Every convex outer-approximation $C'$ of $S$ is such that $C' \supseteq C$

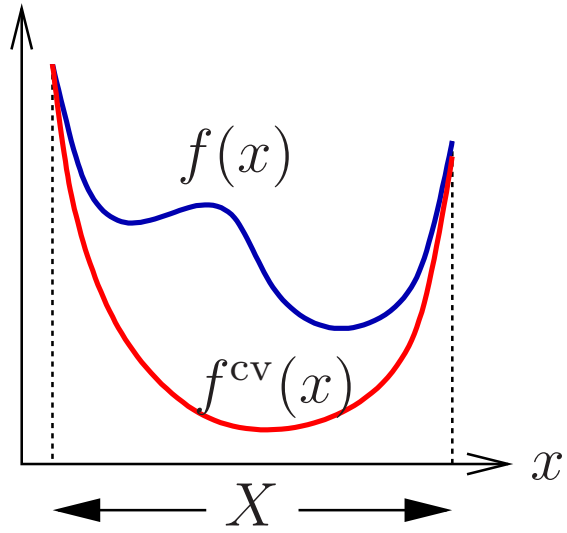
➥ The convex hull is the **tightest** possible convex outer-approximation of a set

# Branch & Bound Building Blocks [cont'd]

## Convex Relaxation

Given (nonconvex) $f : X \to \mathbb{R}$, with $X$ convex, $f^{\text{cv}} : X \to \mathbb{R}$ is a convex relaxation of $f$ on $X$ if

1. $f^{\text{cv}}$ is convex on $X$
2. $f^{\text{cv}}(\mathbf{x}) \leq f(\mathbf{x}),\ \forall \mathbf{x} \in X$

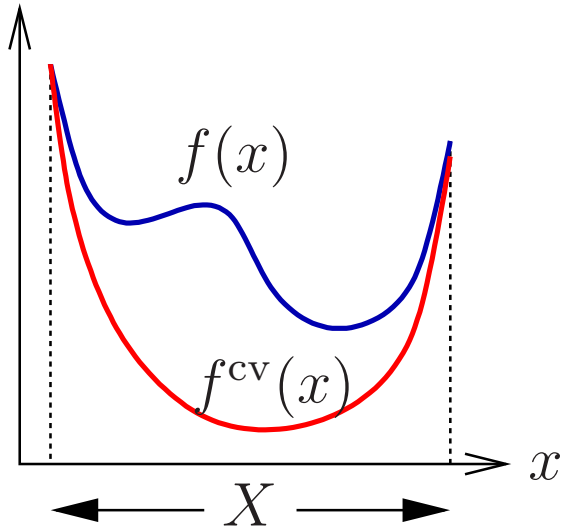


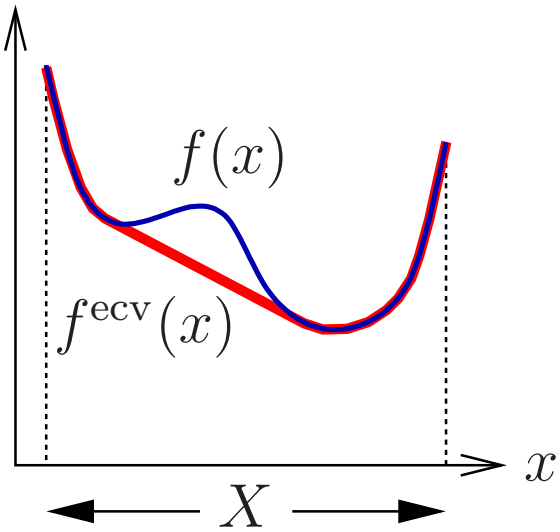

# Branch & Bound Building Blocks [cont'd]

## Convex Relaxation

Given (nonconvex) $f : X \to \mathbb{R}$, with $X$ convex, $f^{\text{cv}} : X \to \mathbb{R}$ is a convex relaxation of $f$ on $X$ if

1. $f^{\text{cv}}$ is convex on $X$
2. $f^{\text{cv}}(\mathbf{x}) \leq f(\mathbf{x}),\ \forall \mathbf{x} \in X$




## Convex Envelope

Given a function $f : X \to \mathbb{R}$, with $X$ convex, $f^{\text{ecv}} : X \to \mathbb{R}$ is the convex envelope of $f$ on $X$ if

1. $f^{\text{ecv}}$ is a convex underestimator of $f$ on $X$
2. Every convex underestimator $f^{\text{cv}}$ of $f$ on $X$ is such that $f^{\text{cv}}(\mathbf{x}) \leq f^{\text{ecv}}(\mathbf{x}),\ \forall \mathbf{x} \in X$

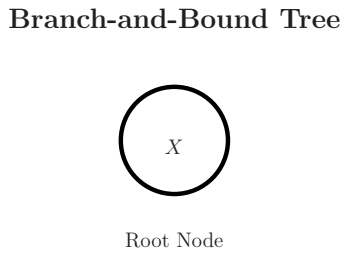
➥ The convex envelope is the **tightest** possible convex relaxation

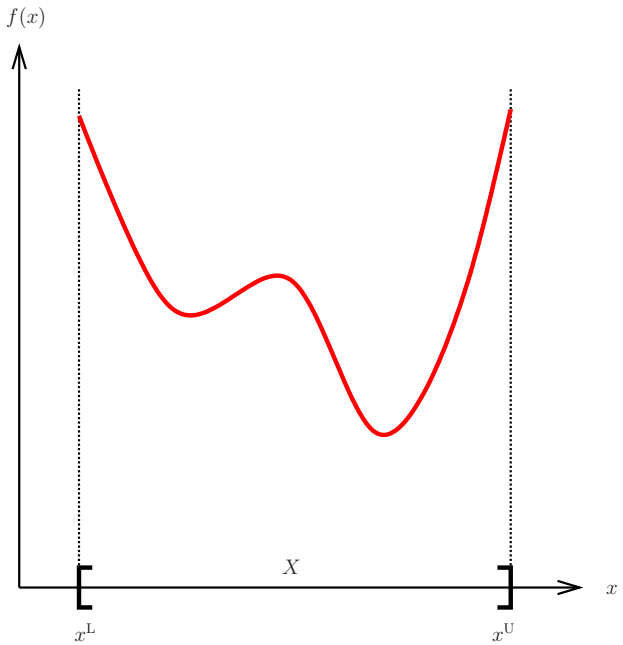
# Branch & Bound Procedure in a Nutshell

**Step 1 – Bounding Step:**

- Find upper and lower bounds:

$$\underline{\mathrm{LBD}} \leq \min_{\mathbf{x} \in S \subseteq X} f(\mathbf{x}) \leq \overline{\mathrm{UBD}}$$



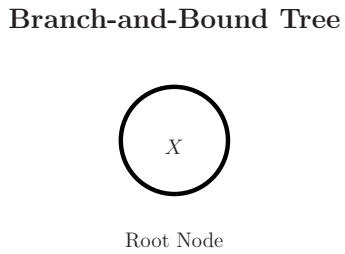
# Branch & Bound Procedure in a Nutshell

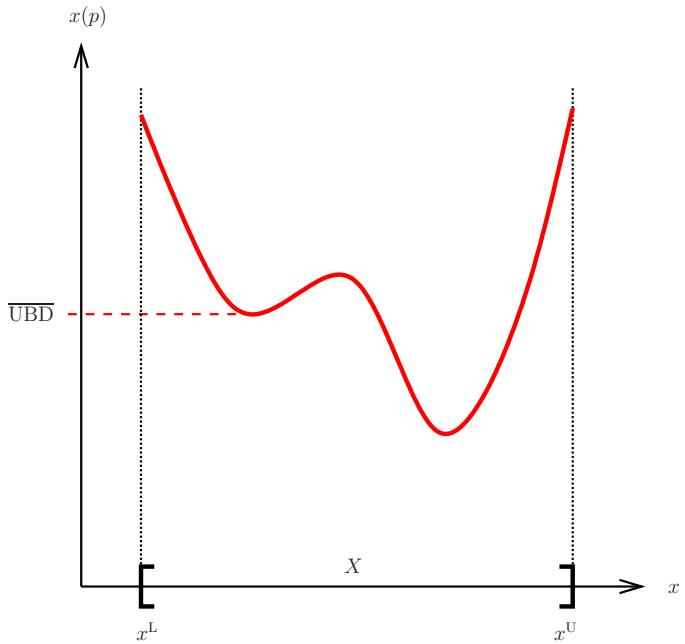
**Step 1 – Bounding Step:**

- Find upper and lower bounds:

$$\underline{\mathrm{LBD}} \leq \min_{\mathbf{x} \in S \subseteq X} f(\mathbf{x}) \leq \overline{\mathrm{UBD}}$$

- **Upper Bound**
  - A feasible point
  - A local optimum



# Branch & Bound Procedure in a Nutshell

**Step 1 – Bounding Step:**

- Find upper and lower bounds:

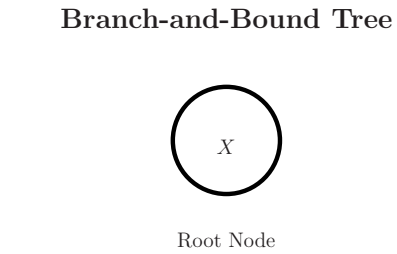
$$\underline{\text{LBD}} \leq \min_{\mathbf{x} \in S \subseteq X} f(\mathbf{x}) \leq \overline{\text{UBD}}$$

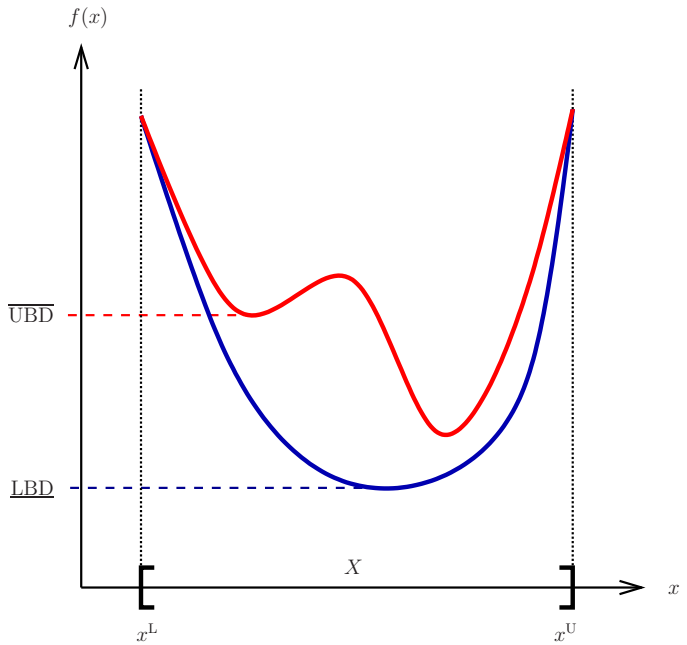
- **Upper Bound**
  - A feasible point
  - A local optimum
- **Lower Bound**
  - Solution of an LP relaxation
  - Solution of a convex NLP relax.

$$\underline{\text{LBD}} = \min_{\mathbf{x} \in C \cap X} \mathbf{u}_f(\mathbf{x})$$

with: $S \subseteq C, \quad \mathbf{u}_f(\mathbf{x}) \leq f(\mathbf{x}), \ \forall \mathbf{x} \in X$

$C$ convex, $\quad \mathbf{u}_f$ convex on $X$



# Branch & Bound Procedure in a Nutshell

**Step 1 – Bounding Step:**

- Find upper and lower bounds:

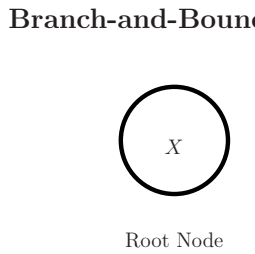
$$\underline{\text{LBD}} \leq \min_{\mathbf{x} \in S \subseteq X} f(\mathbf{x}) \leq \overline{\text{UBD}}$$

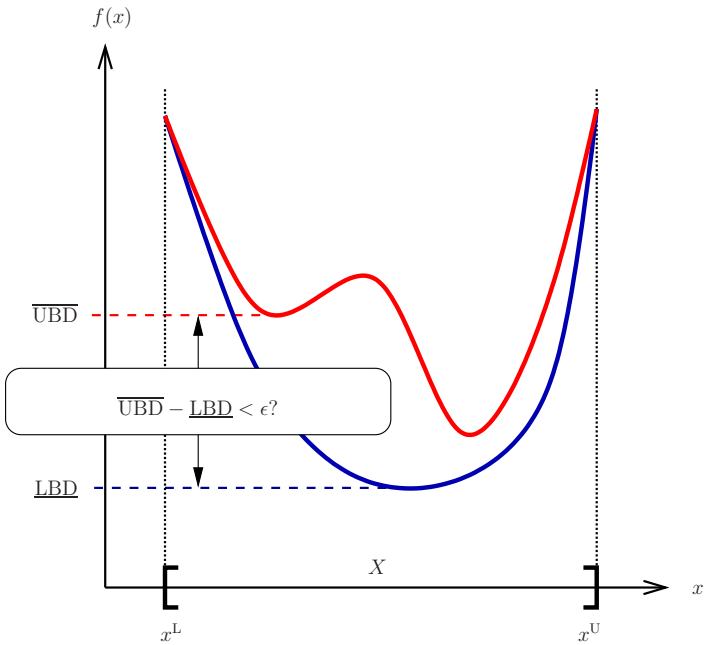
- **Upper Bound**
  - A feasible point
  - A local optimum
- **Lower Bound**
  - Solution of an LP relaxation
  - Solution of a convex NLP relax.

$$\underline{\text{LBD}} = \min_{\mathbf{x} \in C \cap X} \mathbf{u}_f(\mathbf{x})$$

with: $S \subseteq C, \quad \mathbf{u}_f(\mathbf{x}) \leq f(\mathbf{x}), \ \forall \mathbf{x} \in X$

$C$ convex, $\quad \mathbf{u}_f$ convex on $X$

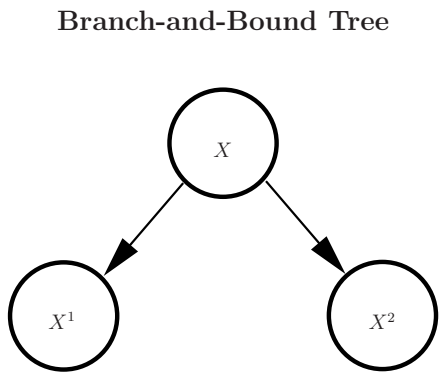


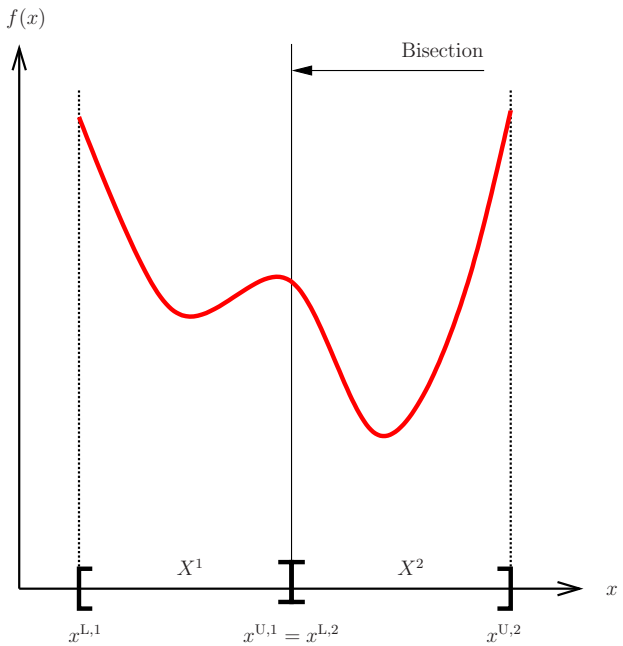



# Branch & Bound Procedure in a Nutshell [cont'd]

**Step 2A – Branching Step:**

- Exhaustive partitioning of search space $X$ as $\{X^k\}$



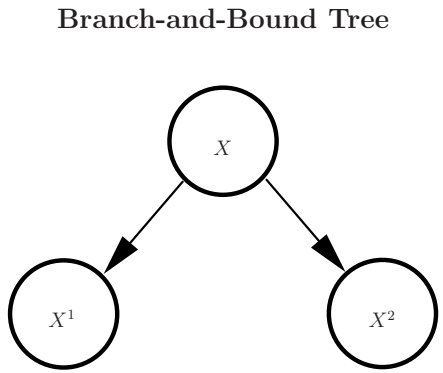



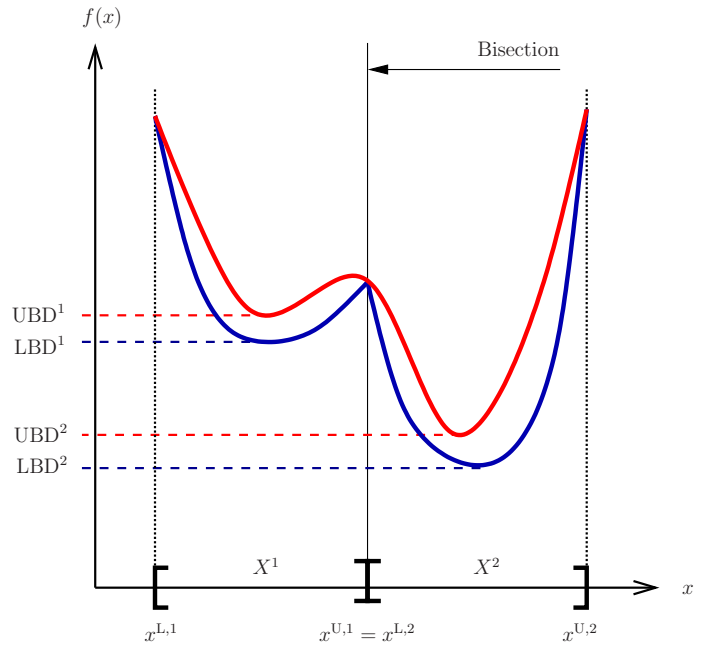
# Branch & Bound Procedure in a Nutshell [cont'd]

**Step 2A – Branching Step:**

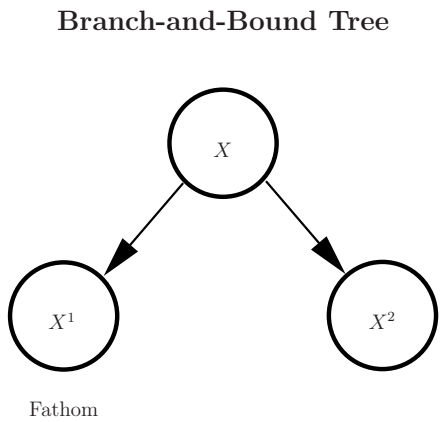
- Exhaustive partitioning of search space $X$ as $\{X^k\}$
- Construct lower and upper bounds $\mathrm{LBD}^k$ and $\mathrm{UBD}^k$ on each subset $X^k$:

$$\mathrm{LBD}^k \leq \min_{\mathbf{x} \in X^k} f(\mathbf{x}) \leq \mathrm{UBD}^k$$



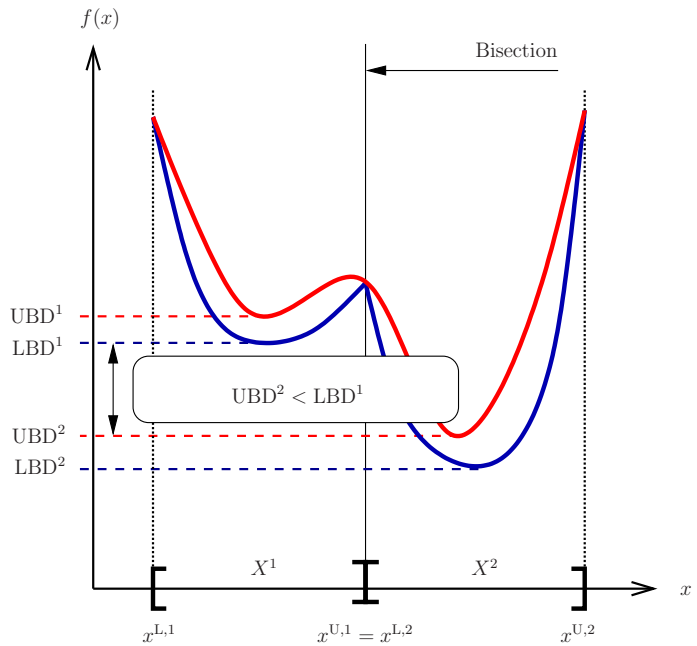
# Branch & Bound Procedure in a Nutshell [cont'd]




**Step 2A – Branching Step:**

- Exhaustive partitioning of search space $X$ as $\{X^k\}$
- Construct lower and upper bounds $\mathrm{LBD}^k$ and $\mathrm{UBD}^k$ on each subset $X^k$:

$$\mathrm{LBD}^k \leq \min_{\mathbf{x} \in X^k} f(\mathbf{x}) \leq \mathrm{UBD}^k$$




**Step 2B – Fathoming:**

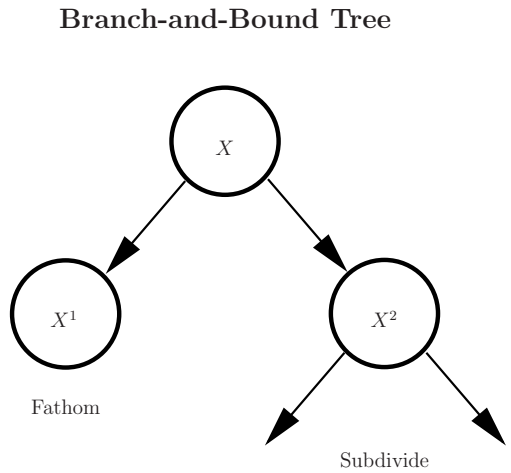
- Eliminate any partition $X^k$ in which $\mathrm{LBD}^k$ exceeds the best upper bound $\overline{\mathrm{UBD}} = \min_k \mathrm{UBD}^k$ (incumbent)

# Branch & Bound Procedure in a Nutshell [cont'd]

**Step 2A – Branching Step:**

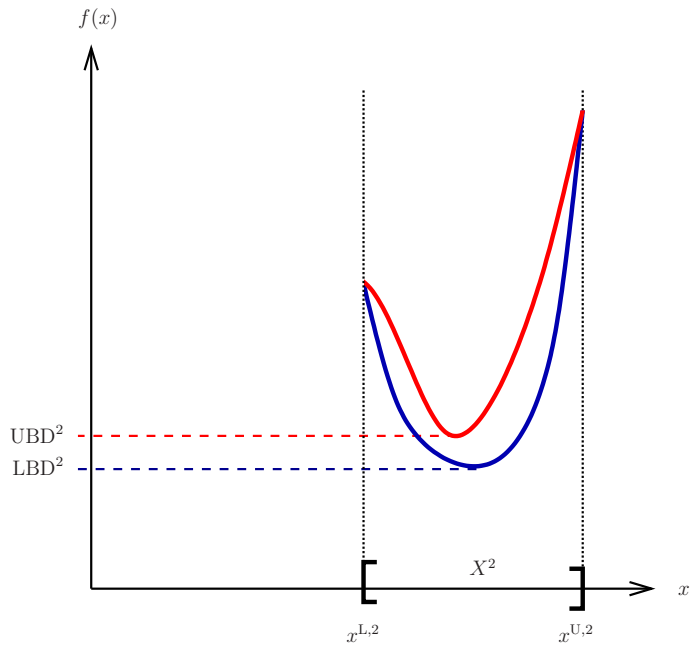
- Exhaustive partitioning of search space $X$ as $\{X^k\}$
- Construct lower and upper bounds $\mathrm{LBD}^k$ and $\mathrm{UBD}^k$ on each subset $X^k$:

$$\mathrm{LBD}^k \leq \min_{\mathbf{x} \in X^k} f(\mathbf{x}) \leq \mathrm{UBD}^k$$




**Step 2B – Fathoming:**

- Eliminate any partition $X^k$ in which $\mathrm{LBD}^k$ exceeds the best upper bound $\overline{\mathrm{UBD}} = \min_k \mathrm{UBD}^k$ (incumbent)



# Branch-and-Bound Algorithm Statement

Problem to solve: $\min\{f(\mathbf{x}) : \mathbf{x} \in S \subseteq X\}$

**Initialization:**

- Select $\epsilon > 0$
- Let $X^0 = X$, and set $K \leftarrow \{0\}$
- Determine a lower bound $\mathrm{LBD}^0$
- If lower bounding problem is infeasible, stop;
  otherwise, update $\underline{\mathrm{LBD}} \leftarrow \mathrm{LBD}^0$
- Determine an upper bound $\mathrm{UBD}^0$
- If upper bounding problem is successful, update $\overline{\mathrm{UBD}} \leftarrow \mathrm{UBD}^0$;
  otherwise, set $\overline{\mathrm{UBD}} \leftarrow +\infty$
- If $\overline{\mathrm{UBD}} - \underline{\mathrm{LBD}} < \epsilon$, stop

# Branch-and-Bound Algorithm Statement [cont'd]

**Main Loop:**

- Select a subset $X^\ell$, and update $K \leftarrow K \setminus \{\ell\}$
- Partition $X^\ell$ into finitely many subsets $X^j$, $j \in J^\ell$
- For each $j \in J^\ell$, determine lower and upper bounds satisfying

$$\mathrm{LBD}^j \leq \min\{f(\mathbf{x}) : \mathbf{x} \in S \cap X^j\} \leq \mathrm{UBD}^j$$

- Fathom all subsets $X^j, j \in J^\ell$, with infeasible lower bounding problem
- Set $K \leftarrow K \cup J^\ell$, and update $\underline{\mathrm{LBD}} \leftarrow \min\{\mathrm{LBD}^k : k \in K\}$, $\overline{\mathrm{UBD}} \leftarrow \min\{\mathrm{UBD}^k : k \in K\}$
- Fathom all subsets $X^k, k \in K$, with $\mathrm{LBD}^k > \overline{\mathrm{UBD}} - \epsilon$ cut-off test

**Stop when** $K = \emptyset$

# Branch-and-Bound in Practice

**Convergence:**

- Consistent Bounding Operation
  - Any open partition can be further refined
  - $\lim_{k \to \infty} UBD^k - LBD^k = 0$, for any infinitely decreasing sequence $\{X^k\}$

# Branch-and-Bound in Practice

**Convergence:**

- Consistent Bounding Operation
  - Any open partition can be further refined
  - $\lim_{k \to \infty} UBD^k - LBD^k = 0$, for any infinitely decreasing sequence $\{X^k\}$
- Bound Improving Selection Operation
  - The subset with worse lower bound is selected every finite number of steps

# Branch-and-Bound in Practice

**Convergence:**

- Consistent Bounding Operation
  - Any open partition can be further refined
  - $\lim_{k \to \infty} UBD^k - LBD^k = 0$, for any infinitely decreasing sequence $\{X^k\}$
- Bound Improving Selection Operation
  - The subset with worse lower bound is selected every finite number of steps

**A branch-and-bound procedure with (i) a consistent bounding operation and (ii) a bound improving selection operation, converges**

# Branch-and-Bound in Practice

**Convergence:**

- Consistent Bounding Operation
  - Any open partition can be further refined
  - $\lim_{k \to \infty} UBD^k - LBD^k = 0$, for any infinitely decreasing sequence $\{X^k\}$
- Bound Improving Selection Operation
  - The subset with worse lower bound is selected every finite number of steps

**A branch-and-bound procedure with (i) a consistent bounding operation and (ii) a bound improving selection operation, converges**

**Full Specification of B&B Algorithm Requires:**

# Branch-and-Bound in Practice

**Convergence:**

- Consistent Bounding Operation
  - Any open partition can be further refined
  - $\lim_{k \to \infty} UBD^k - LBD^k = 0$, for any infinitely decreasing sequence $\{X^k\}$
- Bound Improving Selection Operation
  - The subset with worse lower bound is selected every finite number of steps

**A branch-and-bound procedure with (i) a consistent bounding operation and (ii) a bound improving selection operation, converges**

**Full Specification of B&B Algorithm Requires:**

1. Node selection rule $\rightarrow$ Which subset to consider next?

# Branch-and-Bound in Practice

**Convergence:**

- Consistent Bounding Operation
  - Any open partition can be further refined
  - $\lim_{k \to \infty} UBD^k - LBD^k = 0$, for any infinitely decreasing sequence $\{X^k\}$
- Bound Improving Selection Operation
  - The subset with worse lower bound is selected every finite number of steps

**A branch-and-bound procedure with (i) a consistent bounding operation and (ii) a bound improving selection operation, converges**

**Full Specification of B&B Algorithm Requires:**

1. Node selection rule $\rightarrow$ Which subset to consider next?
2. Branching strategy $\rightarrow$ Which variable to partition next? How?

# Branch-and-Bound in Practice

**Convergence:**

- Consistent Bounding Operation
  - Any open partition can be further refined
  - $\lim_{k \to \infty} UBD^k - LBD^k = 0$, for any infinitely decreasing sequence $\{X^k\}$
- Bound Improving Selection Operation
  - The subset with worse lower bound is selected every finite number of steps

**A branch-and-bound procedure with (i) a consistent bounding operation and (ii) a bound improving selection operation, converges**

**Full Specification of B&B Algorithm Requires:**

1. Node selection rule → Which subset to consider next?
2. Branching strategy → Which variable to partition next? How?
3. Lower bounding techniques → How to obtain valid bounds?

# Branch-and-Bound in Practice

**Convergence:**

- Consistent Bounding Operation
  - Any open partition can be further refined
  - $\lim_{k \to \infty} UBD^k - LBD^k = 0$, for any infinitely decreasing sequence $\{X^k\}$
- Bound Improving Selection Operation
  - The subset with worse lower bound is selected every finite number of steps

**A branch-and-bound procedure with (i) a consistent bounding operation and (ii) a bound improving selection operation, converges**

**Full Specification of B&B Algorithm Requires:**

1. Node selection rule → Which subset to consider next?
2. Branching strategy → Which variable to partition next? How?
3. Lower bounding techniques → How to obtain valid bounds?
4. Bounds tightening techniques → How to obtain better bounds?

# Branch-and-Bound in Practice [cont'd]

## Node Selection and Branching Strategies:

1. Node selection
   - Typically, one of the subdomains with worst lower bound
   - ➥ Guarantees a bound improving selection operation

[1] Belotti et al., *Optim Meth Softw*, 2009

# Branch-and-Bound in Practice [cont'd]

## Node Selection and Branching Strategies:

1. Node selection
   - Typically, one of the subdomains with worst lower bound
   - ➥ Guarantees a bound improving selection operation
2. Branching variable selection
   - Often, select the least reduced variable (absolute or relative)
   - Other typical strategies associate a relaxation gap to each variable and selects the variable with the largest gap
   - ➥ Influences the structure of the B&B tree and the performance of B&B greatly — Why?

[1] Belotti et al., *Optim Meth Softw*, 2009

# Branch-and-Bound in Practice [cont'd]

## Node Selection and Branching Strategies:

1. Node selection
   - Typically, one of the subdomains with worst lower bound
   - ➥ Guarantees a bound improving selection operation
2. Branching variable selection
   - Often, select the least reduced variable (absolute or relative)
   - Other typical strategies associate a relaxation gap to each variable and selects the variable with the largest gap
   - ➥ Influences the structure of the B&B tree and the performance of B&B greatly — Why?
2. Branching point selection
   - Apply bisection most commonly — trisection or multisection possibly
   - Typically, bisect at a convex combination of the mid-point (guarantees exhaustiveness) & the incumbent point (*omega* strategy)[1]
   - ➥ Guarantees exhaustiveness — necessary for a consistent bounding operation

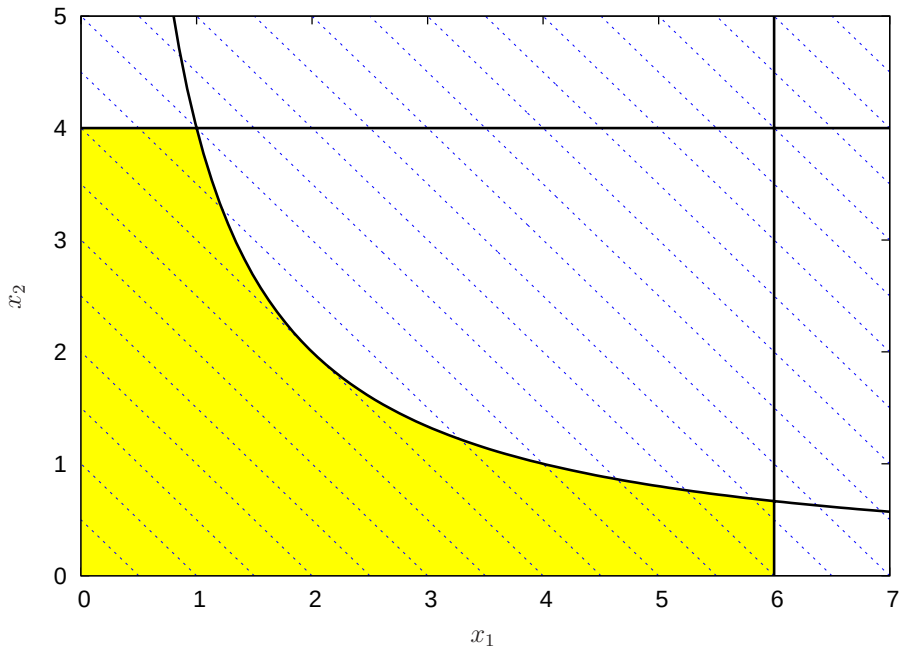
[1] Belotti et al., *Optim Meth Softw*, 2009

# Branch-and-Bound: Numerical Example

**Workshop.** Consider the following constrained optimization problem

$$\min_{x_1,x_2}\{-x_1 - x_2 : x_1x_2 \leq 4,\ (x_1,x_2) \in [x_1^{\mathrm{L}}, x_1^{\mathrm{U}}] \times [x_2^{\mathrm{L}}, x_2^{\mathrm{U}}] := [0,6] \times [0,4]\}$$

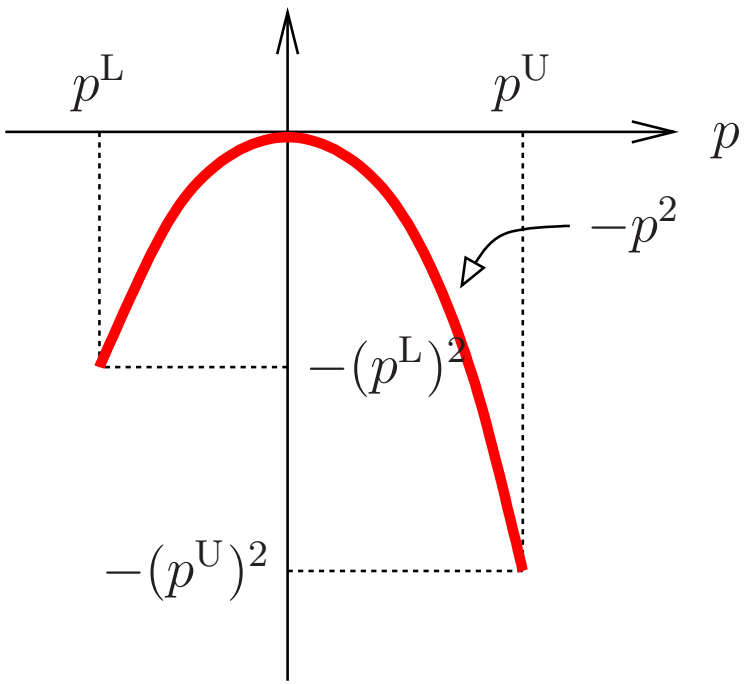
1. What are the sources of nonconvexity in this problem?
2. Identify the various local and global extrema



# Branch-and-Bound: Numerical Example [cont'd]

3. Formulate a convex underestimating program, by using the identity:

$$pq = \frac{1}{2}\left[(p+q)^2 - p^2 - q^2\right]$$

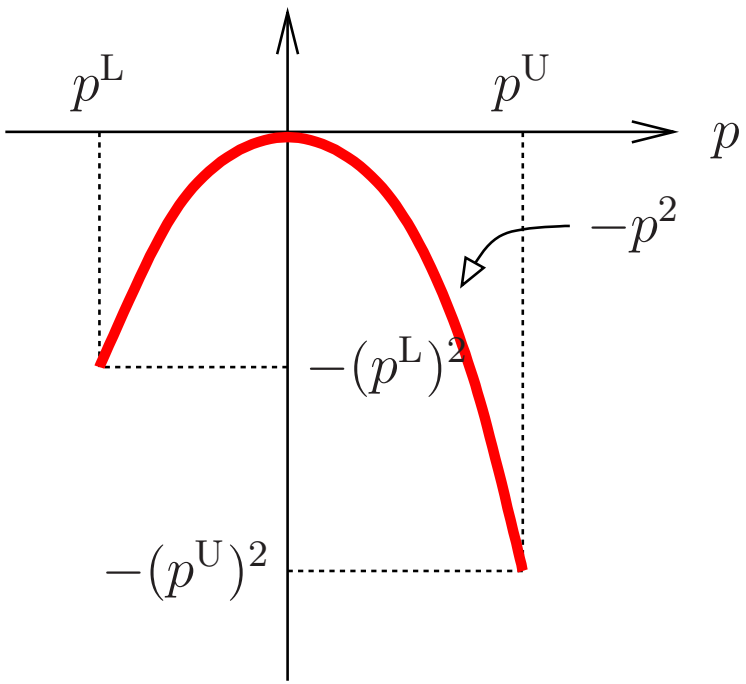



- Equivalent reformulation:
- Convex relaxation:

# Branch-and-Bound: Numerical Example [cont'd]

3. Formulate a convex underestimating program, by using the identity:

$$pq = \frac{1}{2}\left[(p+q)^2 - p^2 - q^2\right]$$




- Equivalent reformulation:

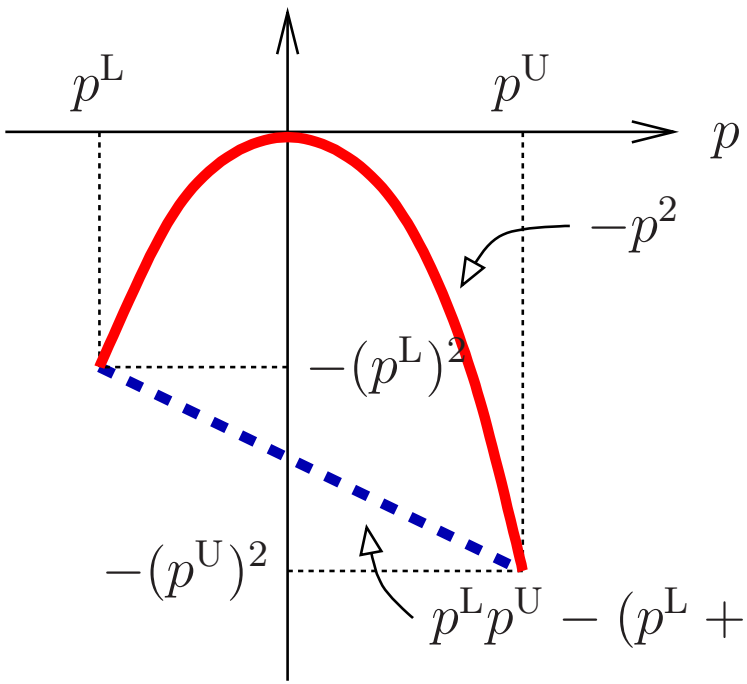
$$\begin{aligned} \min_{x_1,x_2} \quad & -x_1 - x_2 \\ \text{s.t.} \quad & (x_1+x_2)^2 - (x_1)^2 - (x_2)^2 \le 8 \\ & (x_1,x_2) \in [x_1^{\mathrm{L}}, x_1^{\mathrm{U}}] \times [x_2^{\mathrm{L}}, x_2^{\mathrm{U}}] \end{aligned}$$

- Convex relaxation:

# Branch-and-Bound: Numerical Example [cont'd]

3. Formulate a convex underestimating program, by using the identity:

$$pq = \frac{1}{2}\left[(p+q)^2 - p^2 - q^2\right]$$




- Equivalent reformulation:

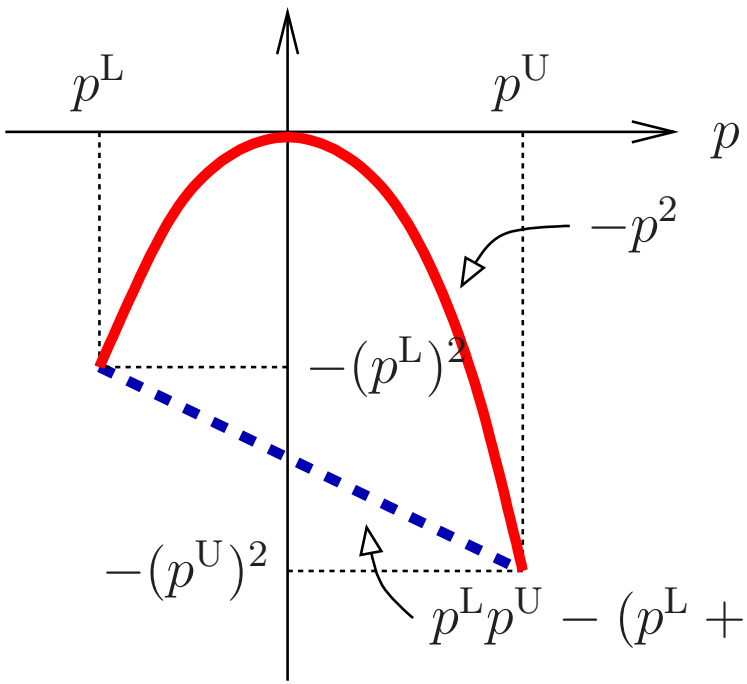
$$\begin{aligned} \min_{x_1,x_2} \quad & -x_1 - x_2 \\ \text{s.t.} \quad & (x_1+x_2)^2 - (x_1)^2 - (x_2)^2 \le 8 \\ & (x_1,x_2) \in [x_1^{\mathrm{L}}, x_1^{\mathrm{U}}] \times [x_2^{\mathrm{L}}, x_2^{\mathrm{U}}] \end{aligned}$$

- Convex relaxation:

# Branch-and-Bound: Numerical Example [cont'd]

3. Formulate a convex underestimating program, by using the identity:

$$pq = \frac{1}{2}\left[(p+q)^2 - p^2 - q^2\right]$$




- Equivalent reformulation:

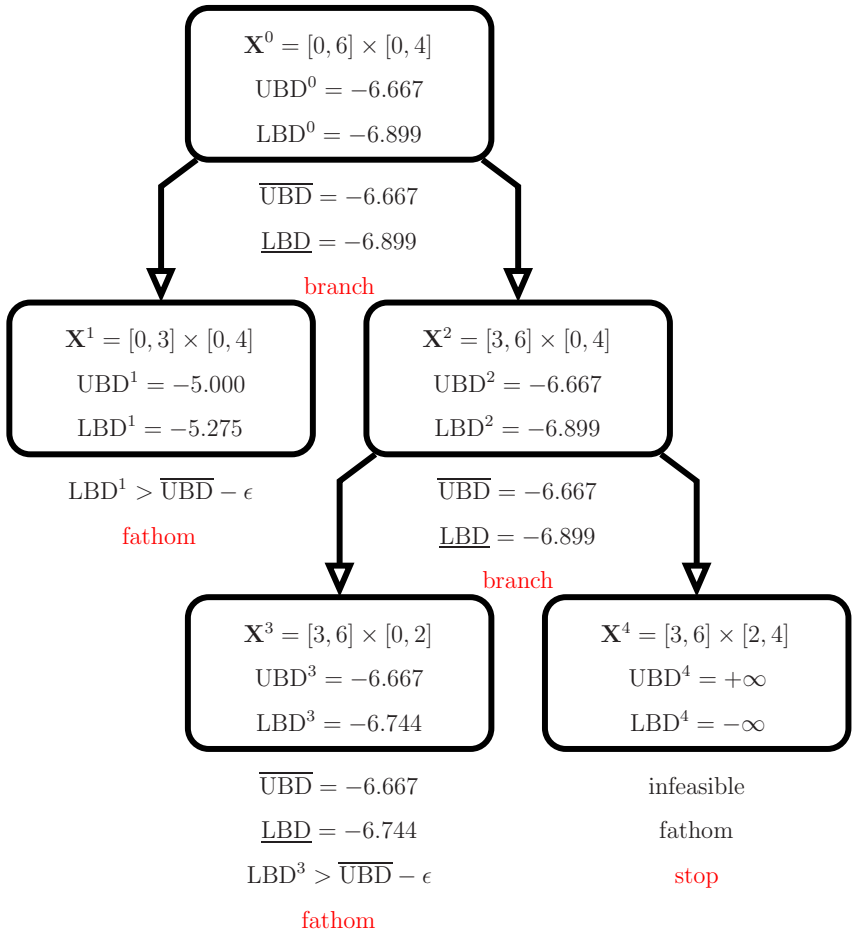
$$\begin{aligned} \min_{x_1,x_2} \quad & -x_1 - x_2 \\ \text{s.t.} \quad & (x_1+x_2)^2 - (x_1)^2 - (x_2)^2 \le 8 \\ & (x_1,x_2) \in [x_1^{\mathrm{L}}, x_1^{\mathrm{U}}] \times [x_2^{\mathrm{L}}, x_2^{\mathrm{U}}] \end{aligned}$$

- Convex relaxation:

$$\begin{aligned} \min_{x_1,x_2} \quad & -x_1 - x_2 \\ \text{s.t.} \quad & (x_1+x_2)^2 - x_1^{\mathrm{L}}x_1^{\mathrm{U}} + (x_1^{\mathrm{L}} + x_1^{\mathrm{U}})x_1 \\ & \qquad - x_2^{\mathrm{L}}x_2^{\mathrm{U}} + (x_2^{\mathrm{L}} + x_2^{\mathrm{U}})x_2 \le 8 \\ & (x_1,x_2) \in [x_1^{\mathrm{L}}, x_1^{\mathrm{U}}] \times [x_2^{\mathrm{L}}, x_2^{\mathrm{U}}] \end{aligned}$$

# Branch-and-Bound: Numerical Example [cont'd]

4. Apply a branch-and-bound procedure, with (absolute) tolerance $\epsilon = 0.1$, that considers nodes with least lower bound first, and selects the least-reduced axis for mid-point bisection at each node



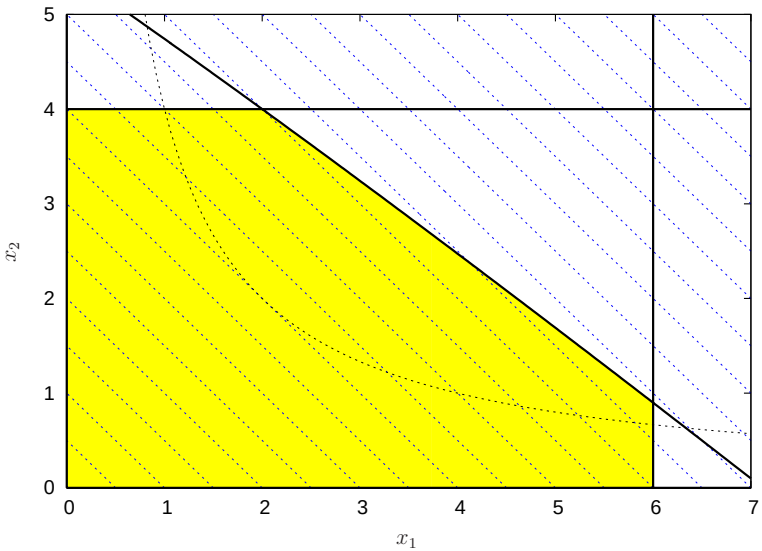
# Branch-and-Bound: Numerical Example [cont'd]

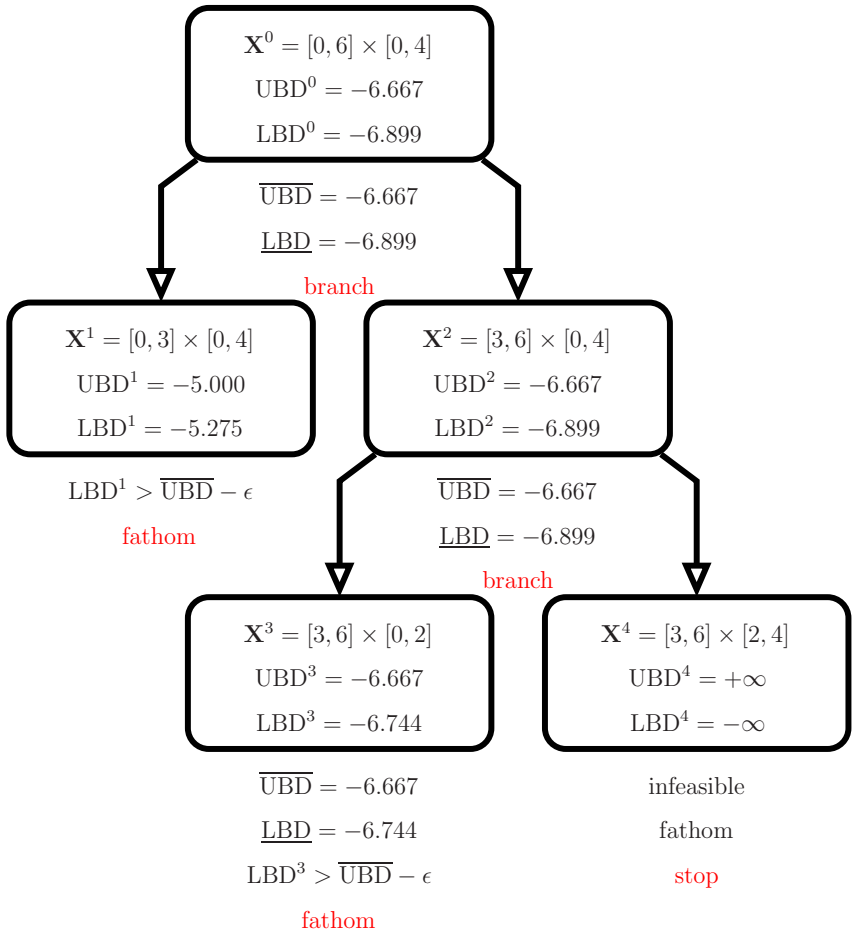
4. Apply a branch-and-bound procedure, with (absolute) tolerance $\epsilon = 0.1$, that considers nodes with least lower bound first, and selects the least-reduced axis for mid-point bisection at each node

$$\min_{x_1,x_2} \; -x_1 - x_2$$

$$\text{s.t.} \quad (x_1 + x_2)^2 - 6x_1 - 4x_2 \leq 8$$

$$(x_1, x_2) \in [0, 6] \times [0, 4]$$



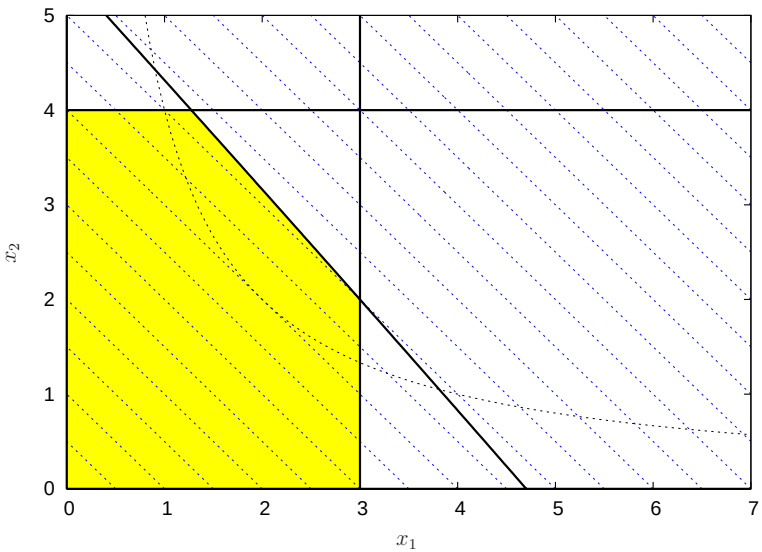
# Branch-and-Bound: Numerical Example [cont'd]

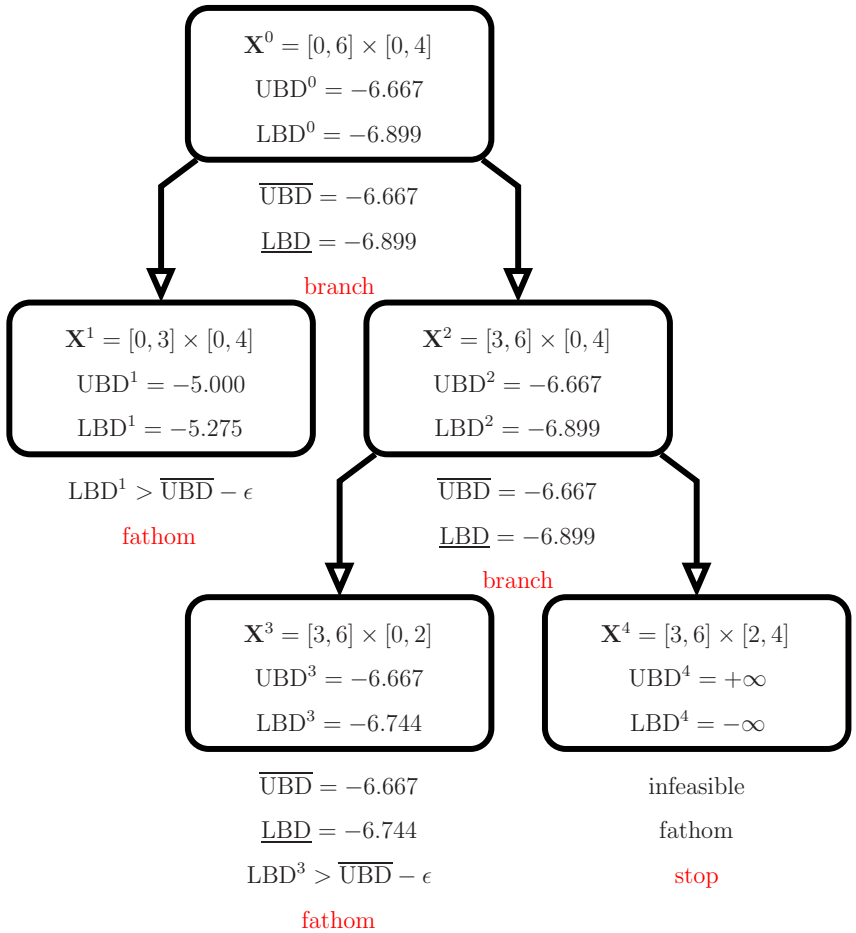
4. Apply a branch-and-bound procedure, with (absolute) tolerance $\epsilon = 0.1$, that considers nodes with least lower bound first, and selects the least-reduced axis for mid-point bisection at each node

$$\min_{x_1, x_2} \; -x_1 - x_2$$

$$\text{s.t.} \quad (x_1 + x_2)^2 - 3x_1 - 4x_2 \leq 8$$

$$(x_1, x_2) \in [0, 3] \times [0, 4]$$



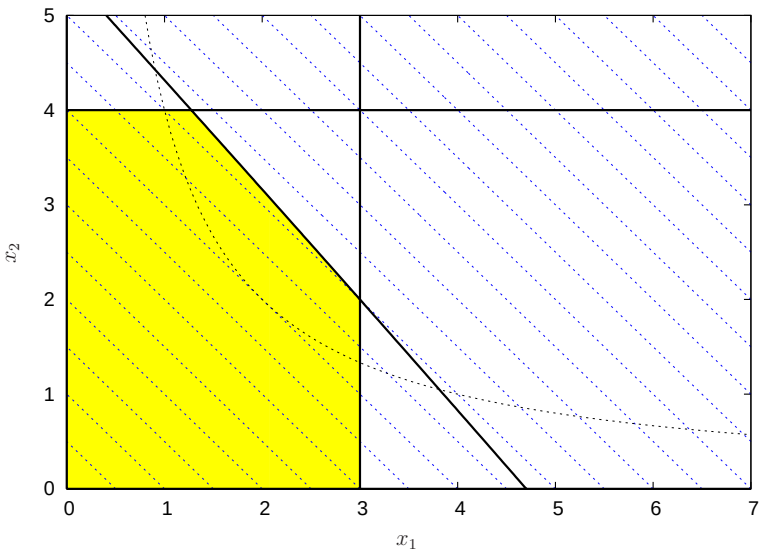
# Branch-and-Bound: Numerical Example [cont'd]

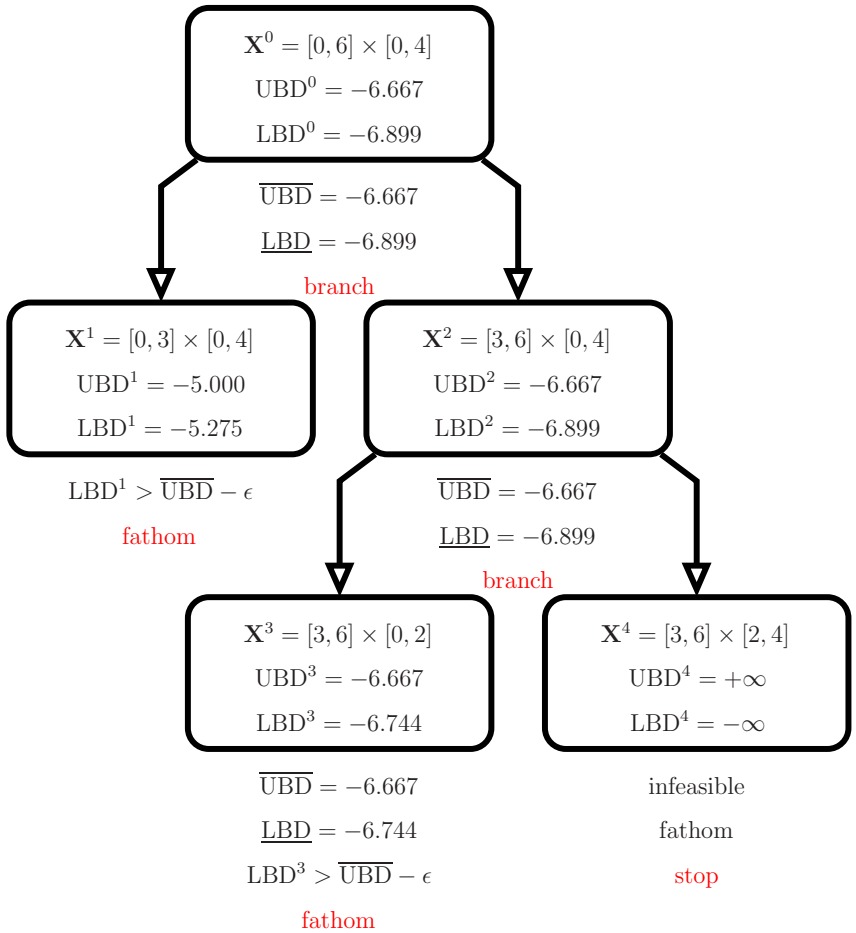
4. Apply a branch-and-bound procedure, with (absolute) tolerance $\epsilon = 0.1$, that considers nodes with least lower bound first, and selects the least-reduced axis for mid-point bisection at each node

$$\min_{x_1,x_2} \; -x_1 - x_2$$

$$\text{s.t.} \quad (x_1 + x_2)^2 - 9x_1 - 4x_2 \leq -10$$

$$(x_1, x_2) \in [3, 6] \times [0, 4]$$



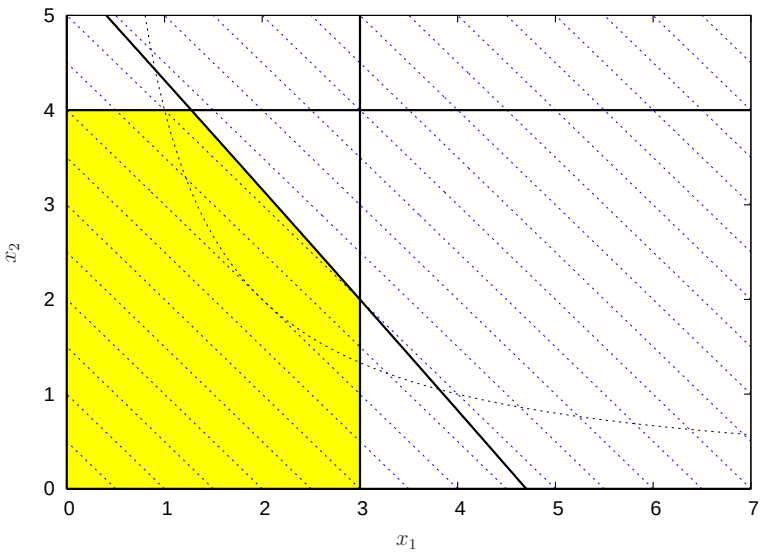
# Branch-and-Bound: Numerical Example [cont'd]

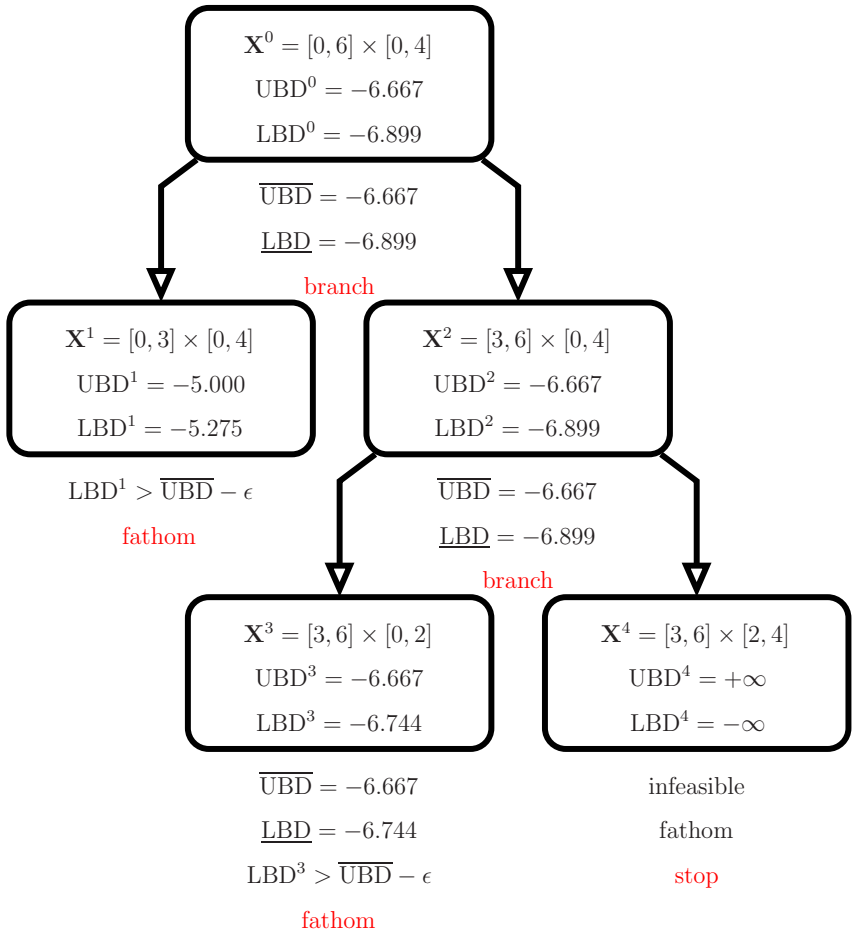
4. Apply a branch-and-bound procedure, with (absolute) tolerance $\epsilon = 0.1$, that considers nodes with least lower bound first, and selects the least-reduced axis for mid-point bisection at each node

$$\min_{x_1,x_2} \; -x_1 - x_2$$

$$\text{s.t.} \quad (x_1 + x_2)^2 - 9x_1 - 2x_2 \leq -10$$

$$(x_1, x_2) \in [3, 6] \times [0, 2]$$



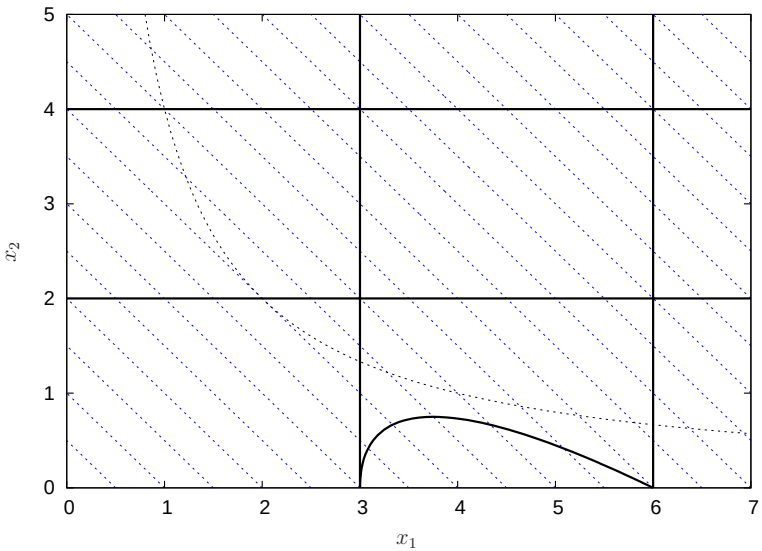
# Branch-and-Bound: Numerical Example [cont'd]

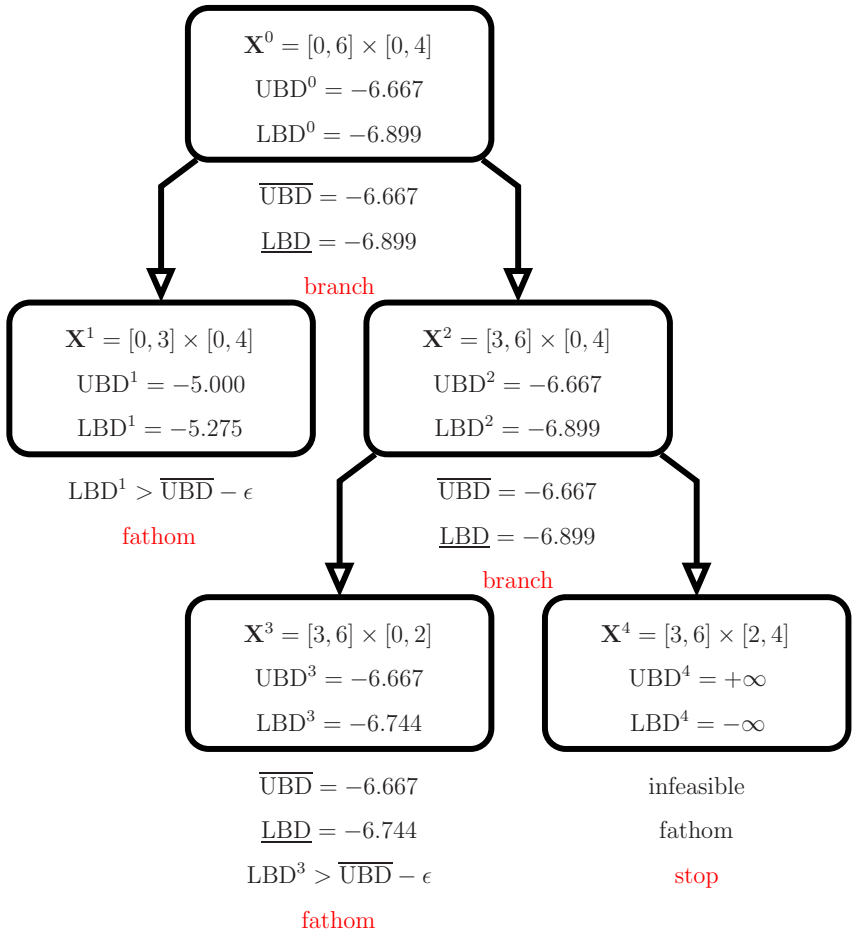
4. Apply a branch-and-bound procedure, with (absolute) tolerance $\epsilon = 0.1$, that considers nodes with least lower bound first, and selects the least-reduced axis for mid-point bisection at each node

$$\min_{x_1, x_2} \; -x_1 - x_2$$

$$\text{s.t.} \quad (x_1 + x_2)^2 - 9x_1 - 6x_2 \leq -18$$

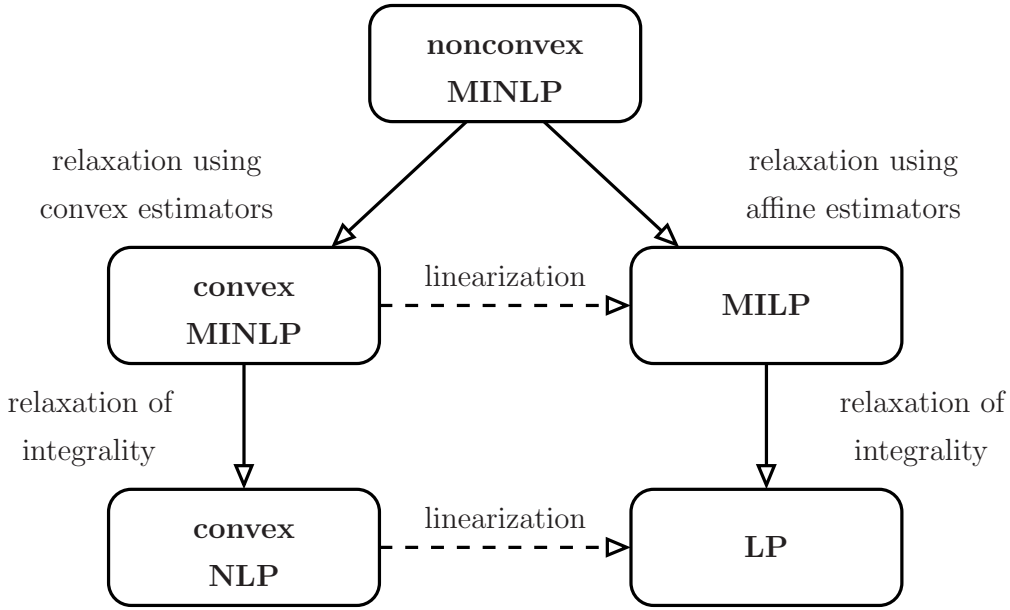
$$(x_1, x_2) \in [3, 6] \times [2, 4]$$



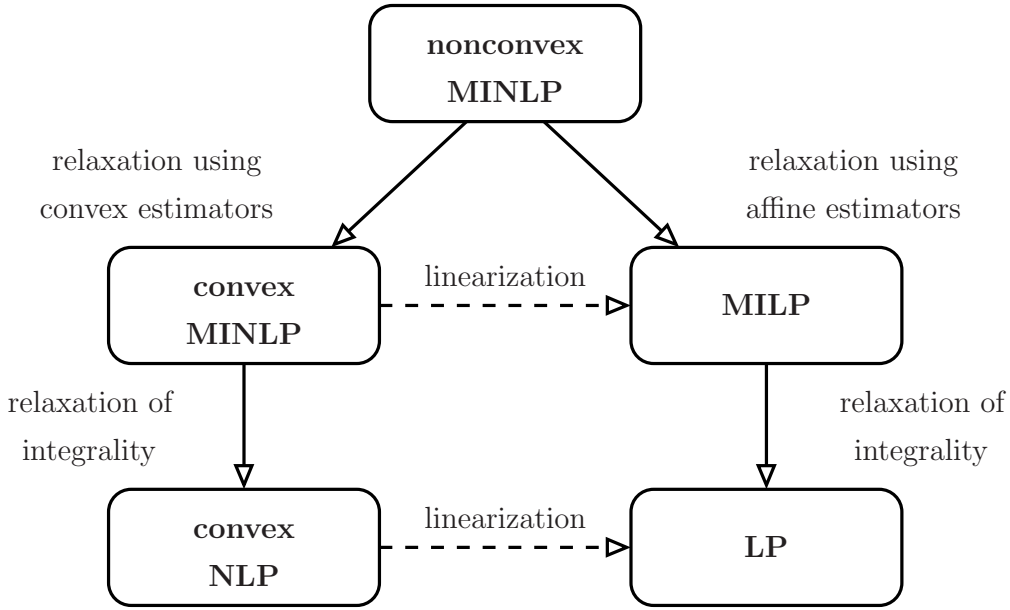
# Nonconvex MINLP Problems: Branch & Bound

## Computing Lower Bounds for Nonconvex MINLPs:



# Nonconvex MINLP Problems: Branch & Bound

## Computing Lower Bounds for Nonconvex MINLPs:




**Convex MINLP Relax.**

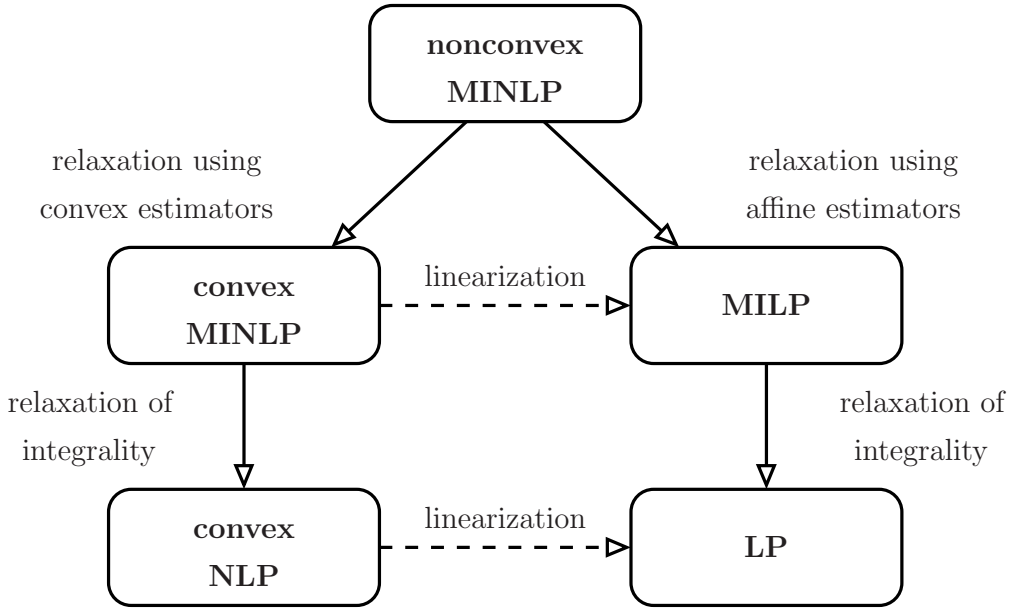
$$\min_{\mathbf{x},\mathbf{y}} \ \mathbf{u}_f(\mathbf{x},\mathbf{y})$$

$$\text{s.t.} \quad \mathbf{u}_\mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$

$$\mathbf{x} \in X, \ \mathbf{y} \in Y$$

# Nonconvex MINLP Problems: Branch & Bound

## Computing Lower Bounds for Nonconvex MINLPs:




### Convex MINLP Relax.

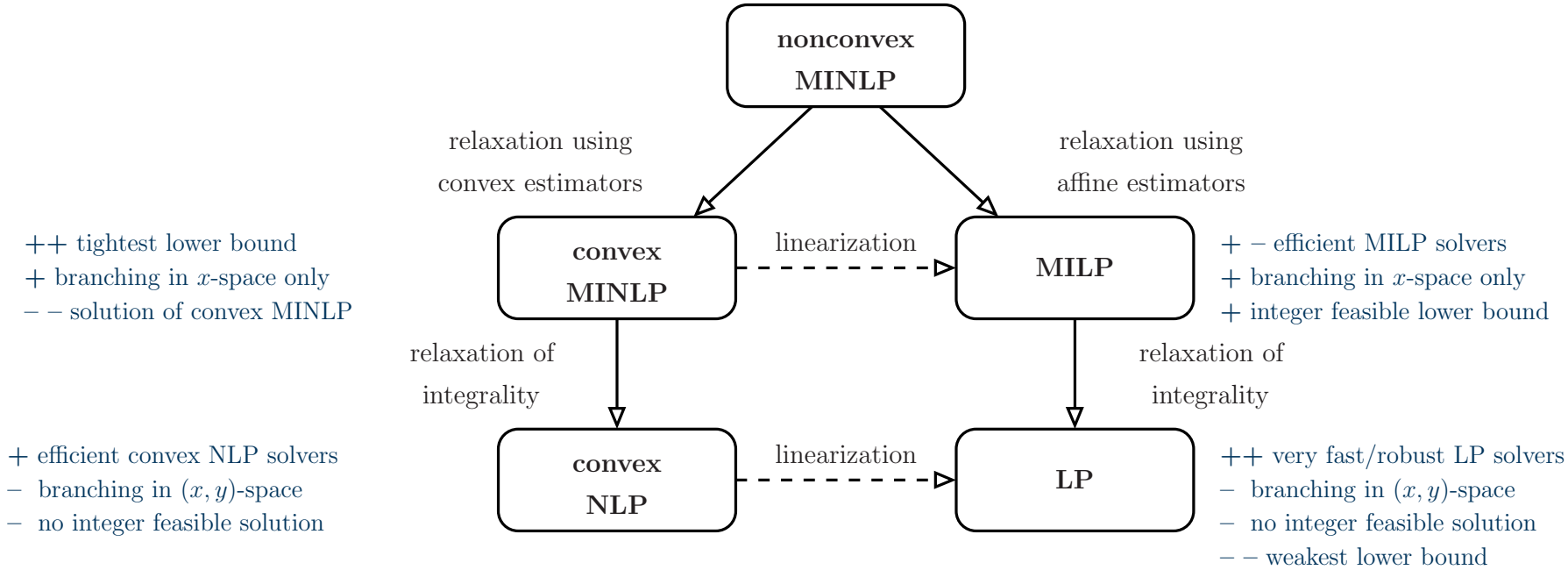
$$\min_{\mathbf{x},\mathbf{y}} \ \mathbf{u}_f(\mathbf{x},\mathbf{y})$$
$$\text{s.t.} \quad \mathbf{u}_g(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$
$$\mathbf{x} \in X, \ \mathbf{y} \in Y$$

### Convex NLP Relax.

$$\min_{\mathbf{x},\mathbf{y}} \ \mathbf{u}_f(\mathbf{x},\mathbf{y})$$
$$\text{s.t.} \quad \mathbf{u}_g(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$
$$\mathbf{x} \in X, \ \mathbf{y} \in \mathbf{u}_Y$$

# Nonconvex MINLP Problems: Branch & Bound

## Computing Lower Bounds for Nonconvex MINLPs:




### Convex MINLP Relax.

$$\min_{\mathbf{x},\mathbf{y}} \ \mathbf{u}_f(\mathbf{x},\mathbf{y})$$

$$\text{s.t.} \quad \mathbf{u}_\mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$

$$\mathbf{x} \in X, \ \mathbf{y} \in Y$$

### Convex NLP Relax.

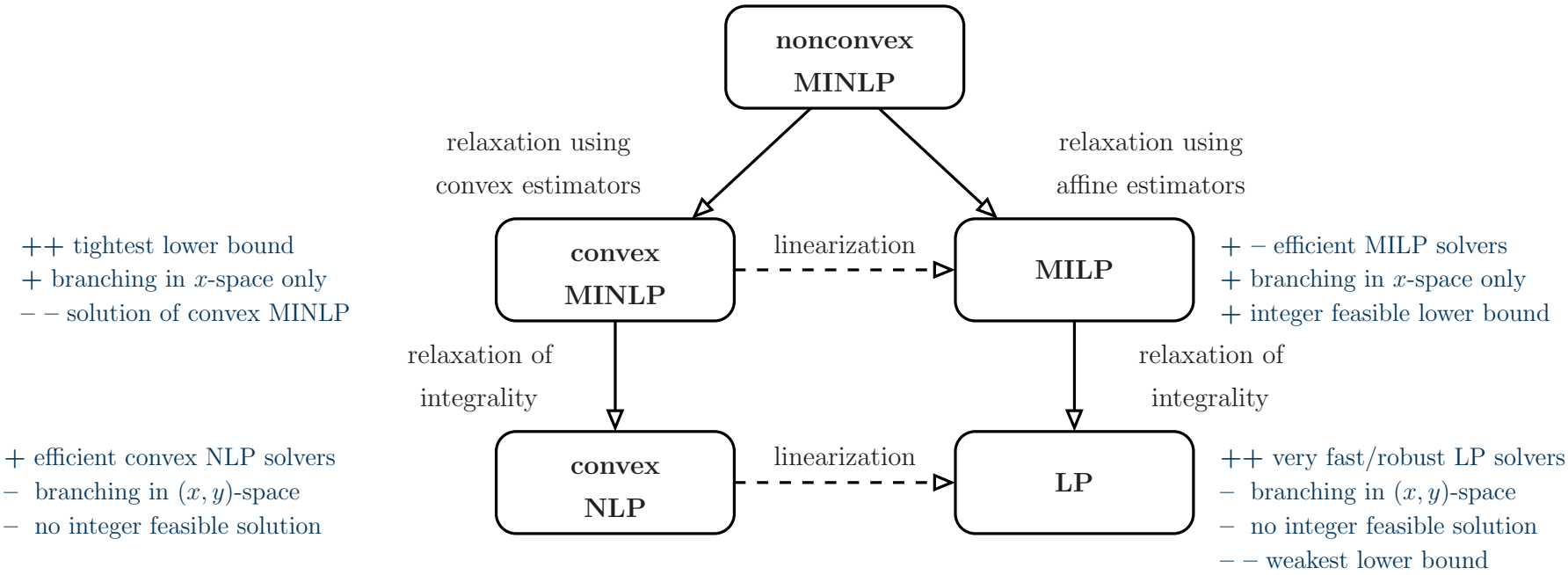
$$\min_{\mathbf{x},\mathbf{y}} \ \mathbf{u}_f(\mathbf{x},\mathbf{y})$$

$$\text{s.t.} \quad \mathbf{u}_\mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$

$$\mathbf{x} \in X, \ \mathbf{y} \in \mathbf{u}_Y$$

# Nonconvex MINLP Problems: Branch & Bound

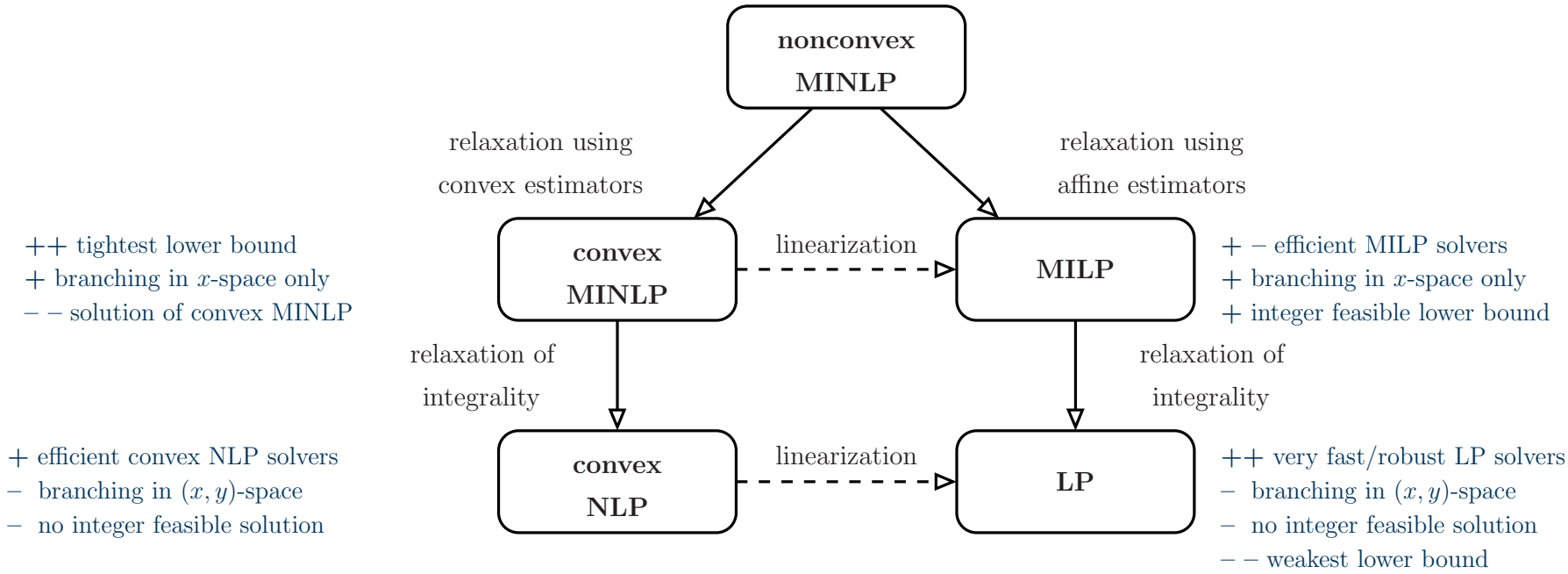
## Computing Lower Bounds for Nonconvex MINLPs:




## Computing Upper Bounds for Nonconvex MINLPs:

# Nonconvex MINLP Problems: Branch & Bound

## Computing Lower Bounds for Nonconvex MINLPs:

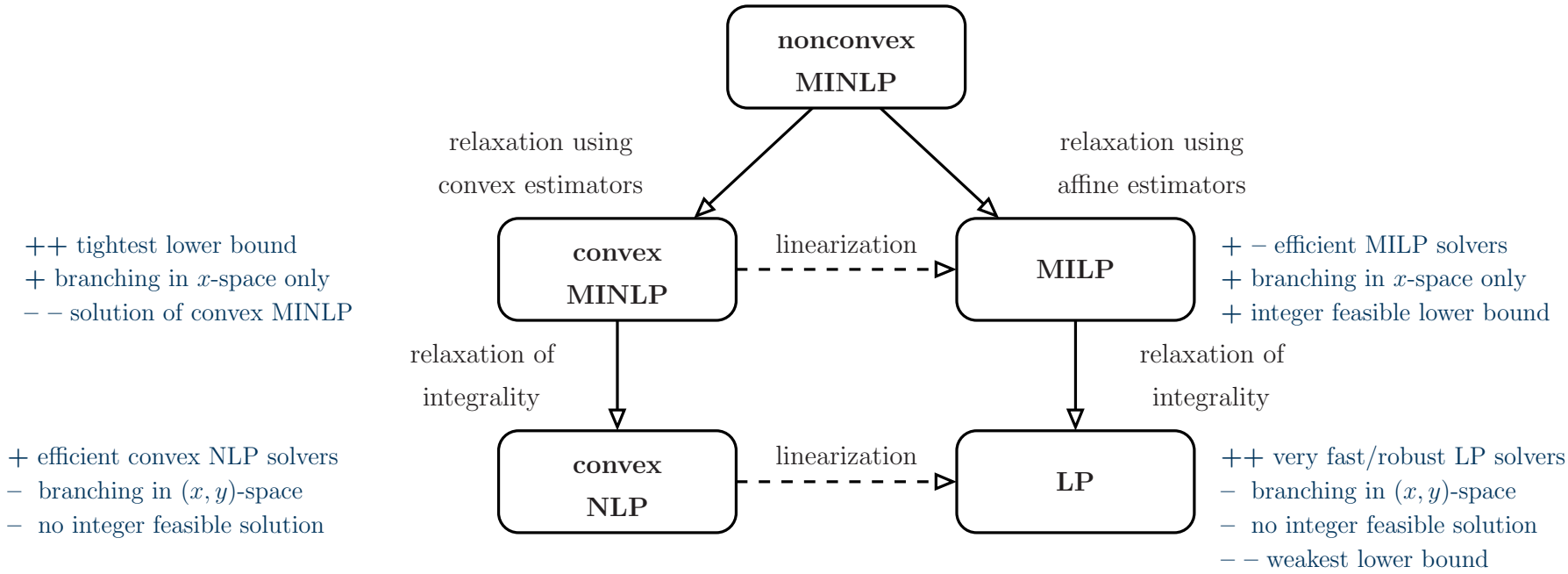



## Computing Upper Bounds for Nonconvex MINLPs:

1. Local Solution of Nonconvex MINLP
   - Straightforward, yet...
   - Lack of robustness
   - High computational burden

# Nonconvex MINLP Problems: Branch & Bound

## Computing Lower Bounds for Nonconvex MINLPs:

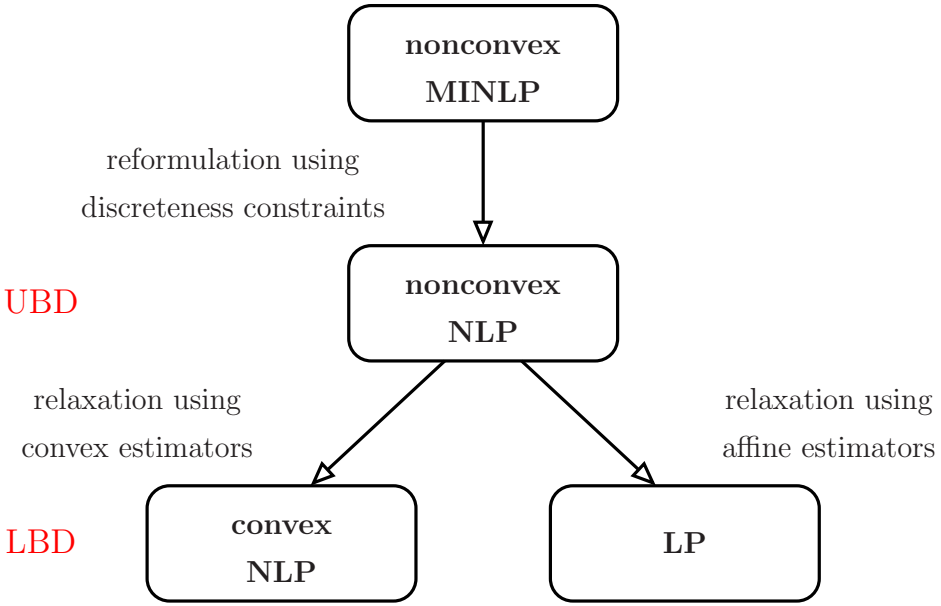


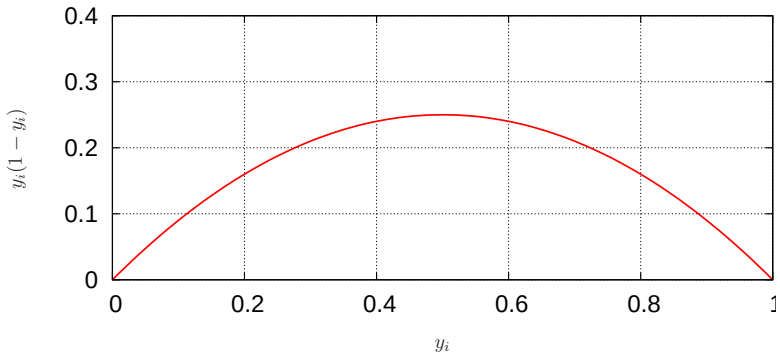

## Computing Upper Bounds for Nonconvex MINLPs:

1. Local Solution of Nonconvex MINLP
   - Straightforward, yet...
   - Lack of robustness
   - High computational burden
2. Local Solution of Nonconvex NLP
   - Fixed integer/binary variables
   - At solution of convex MINLP/MINLP lower bound
   - At rounded solution of convex NLP/LP lower bound

# Nonconvex MINLP Problems: Branch & Bound [cont'd]

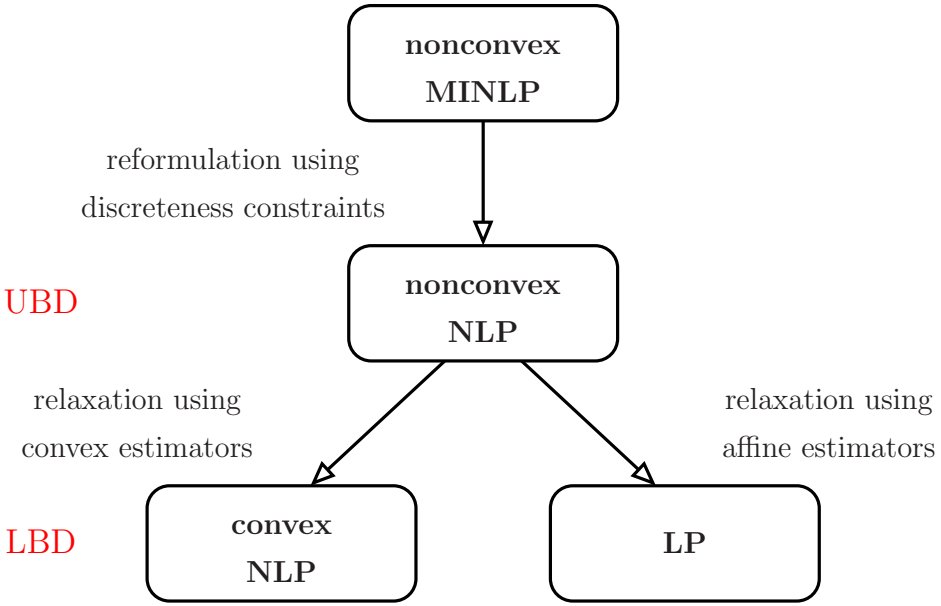
## Alternative Algorithms for Solution of Nonconvex MINLPs:

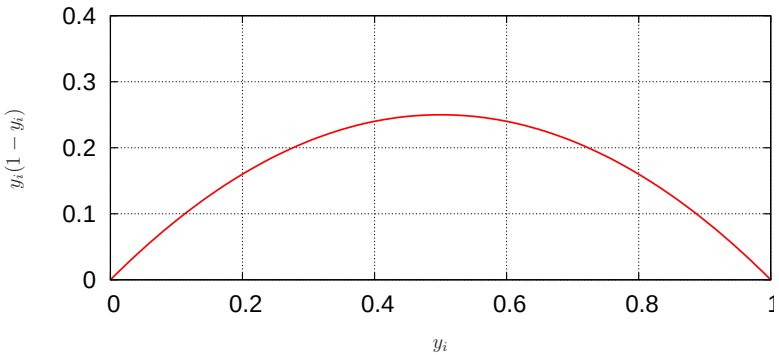


# Nonconvex MINLP Problems: Branch & Bound [cont'd]

## Alternative Algorithms for Solution of Nonconvex MINLPs:




### Nonconvex NLP Reformul.

$$\min_{\mathbf{x},\mathbf{y}} \; f(\mathbf{x},\mathbf{y})$$

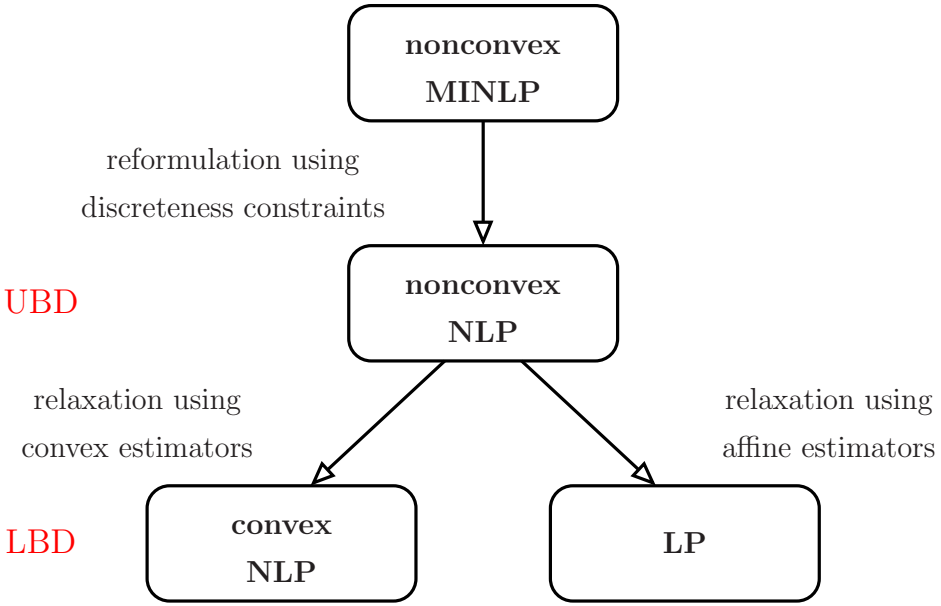
$$\text{s.t.} \quad \mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$

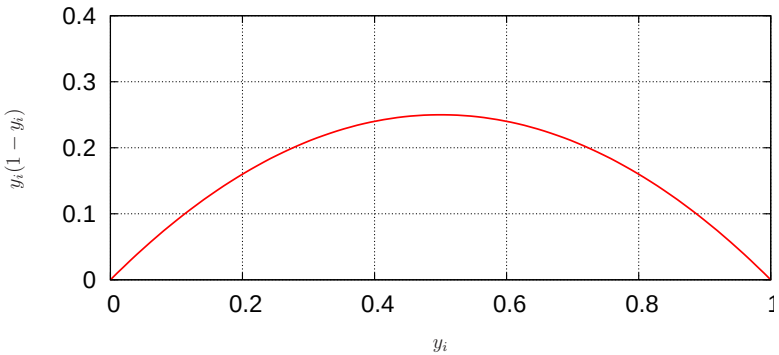
$$y_i(1-y_i) = 0, \;\; \forall i$$

$$\mathbf{x} \in X, \;\; \mathbf{y} \in \mathbf{u}_Y$$

# Nonconvex MINLP Problems: Branch & Bound [cont'd]

## Alternative Algorithms for Solution of Nonconvex MINLPs:




### Nonconvex NLP Reformul.

$$\min_{\mathbf{x},\mathbf{y}} \; f(\mathbf{x},\mathbf{y})$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$

$$y_i(1-y_i) = 0, \;\; \forall i$$

$$\mathbf{x} \in X, \;\; \mathbf{y} \in \mathbf{u}_Y$$

### Convex NLP Relax.

$$\min_{\mathbf{x},\mathbf{y},\boldsymbol{\omega}} \; \mathbf{u}_f(\mathbf{x},\mathbf{y})$$

$$\text{s.t.} \quad \mathbf{u}_\mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$

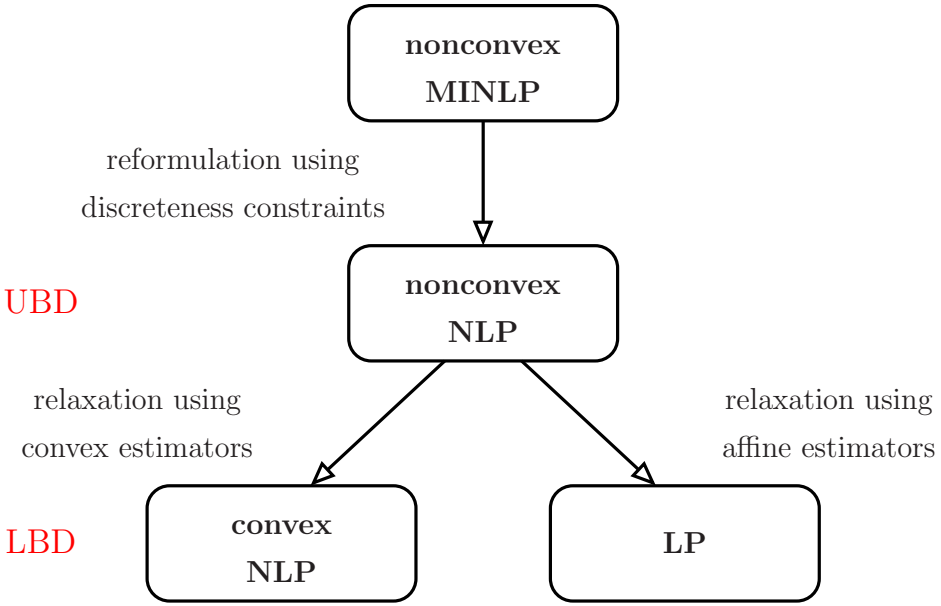
$$y_i - \omega_i = 0, \;\; \forall i$$

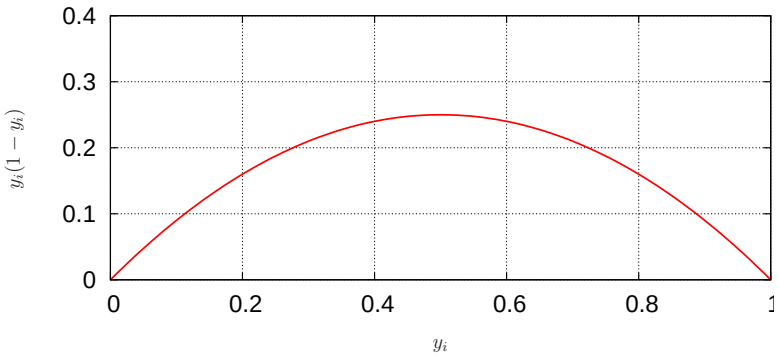
$$y_i^2 \leq \omega_i \leq y_i, \;\; \forall i$$

$$\mathbf{x} \in X, \;\; \mathbf{y} \in \mathbf{u}_Y$$

# Nonconvex MINLP Problems: Branch & Bound [cont'd]

## Alternative Algorithms for Solution of Nonconvex MINLPs:




- Branching in $(\mathbf{x}, \mathbf{y})$-space

### Nonconvex NLP Reformul.

$$\min_{\mathbf{x},\mathbf{y}} \; f(\mathbf{x},\mathbf{y})$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$

$$y_i(1-y_i) = 0, \;\; \forall i$$

$$\mathbf{x} \in X, \;\; \mathbf{y} \in \mathbf{u}_Y$$

### Convex NLP Relax.

$$\min_{\mathbf{x},\mathbf{y},\boldsymbol{\omega}} \; \mathbf{u}_f(\mathbf{x},\mathbf{y})$$

$$\text{s.t.} \quad \mathbf{u}_\mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0}$$

$$y_i - \omega_i = 0, \;\; \forall i$$

$$y_i^2 \leq \omega_i \leq y_i, \;\; \forall i$$

$$\mathbf{x} \in X, \;\; \mathbf{y} \in \mathbf{u}_Y$$