

Outer approximation, Benders' decomposition
Summer School on Nonlinear Optimization with Mixed Integer and
Complementarity Constraints

Moritz Diehl, Andrea Ghezzi, Sven Leyffer, and Armin Nurkanović

Argonne National Laboratory and University of Freiburg

September 29, 2026

Outer approximation, Benders' decomposition

- 1 Recap of Nonlinear Optimization
- 2 Introduction to Mixed-Integer Optimization
- 3 Multi-Tree Methods: Outer Approximation, Benders Decomposition, ECP
- 4 A Recent Algorithm and Implementation
- 5 Some Applications from Optimal Control
- 6 Summary and Take-Away Points

The Karush-Kuhn-Tucker Conditions

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0, \\ & h(x) = 0.\end{array}$$

Theorem (Karush-Kuhn-Tucker (KKT) Conditions)

x^* local minimizer of NLP and assume LICQ or MFCQ hold at x^* . Then there exist Lagrange multipliers, λ^* such that

$$\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = 0 \quad \text{first order condition} \quad (1)$$

$$h(x^*) = 0 \quad \text{feasibility} \quad (2)$$

$$g(x^*) \leq 0 \quad \text{feasibility} \quad (3)$$

$$\mu^* \geq 0 \quad \text{dual feasibility} \quad (4)$$

$$\mu_i^* g_i(x^*) = 0 \quad \text{complementary slackness.} \quad (5)$$

Interpretation of KKT Conditions

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0, \\ & h(x) = 0.\end{array}$$

Remark (Stationarity Conditions and Algorithms)

Take standard NLP & linearize about stationary point, x^ , then: KKT conditions are the FO conditions of linearized problem:*

$$\begin{array}{ll}\text{minimize}_{d} & f(x^*) + d^T \nabla f(x^*) \\ \text{subject to} & h(x^*) + d^T \nabla h(x^*) = 0, \\ & g(x^*) + d^T \nabla g(x^*) \geq 0.\end{array}$$

Outline

- 1 Recap of Nonlinear Optimization
- 2 Introduction to Mixed-Integer Optimization**
- 3 Multi-Tree Methods: Outer Approximation, Benders Decomposition, ECP
- 4 A Recent Algorithm and Implementation
- 5 Some Applications from Optimal Control
- 6 Summary and Take-Away Points

Formulation of a Mixed-Integer Nonlinear Problem

Mixed-Integer Nonlinear Programming (MINLP) problem

$$\underset{x,y}{\text{minimize}} \ f(x,y) \quad \text{subject to} \ 0 \geq g(x,y), \ x \in \mathcal{X} \cap \mathbb{R}^{n_x}, \ y \in \mathcal{Y} \cap \mathbb{Z}^{n_y}$$

Definition (Convex Differentiable Function)

A function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex, iff $\forall x^{(0)}, x^{(1)} \in \mathbb{R}^n$ we have:

$$h(x^{(1)}) \geq h(x^{(0)}) + (x^{(1)} - x^{(0)})^T \nabla h^{(0)}$$

In slight **abuse of terminology**, we say that ...

Definition (“Convex” MINLP)

MINLP is *convex* if the problem functions $f(x,y)$ and $g(x,y)$ are convex functions. If either $f(x,y)$ or any $g_i(x,y)$ is a nonconvex function, then MINLP is *nonconvex*.

Relaxations of Nonlinear Convex Constraints

Consider the **epigraph** reformulation of a MINLP

$$\underset{x, y, \eta}{\text{minimize}} \quad \eta \quad \text{subject to} \quad \eta \geq f(x, y), \quad 0 \geq g(x, y), \quad x \in \mathcal{X} \cap \mathbb{R}^{n_x}, \quad y \in \mathcal{Y} \cap \mathbb{Z}^{n_y}$$

Relaxing Convex Constraints

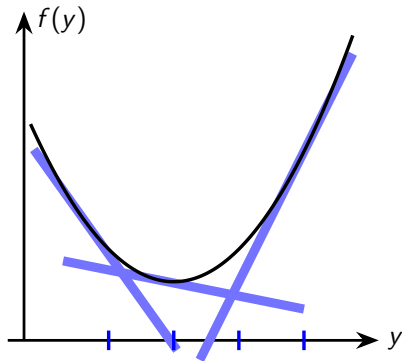
- Convex $0 \geq g(x, y)$ and $\eta \geq f(x, y)$ relaxed by supporting hyperplanes

$$\eta \geq \hat{f} + \nabla \hat{f}^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \right)$$

$$0 \geq \hat{g} + \nabla \hat{g}^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \right)$$

for any point $(\hat{x}, \hat{y})$

- Polyhedral relaxation of convex constraints.



Optimality Lemma

Fix integer variables $\hat{y}$ and let $\hat{x}$ solve NLP

$$\text{NLP}(\hat{y}) \left\{ \begin{array}{ll} \underset{x}{\text{minimize}} & f(x, \hat{y}) \end{array} \right. \text{ subject to } g(x, \hat{y}) \leq 0$$

with optimal value, $\hat{f} = f(\hat{x}, \hat{y})$

Lemma (Optimality Lemma)

Let $\hat{x}$ solve $\text{NLP}(\hat{y})$, and consider the linear constraints

$$\begin{aligned} \eta &\geq \hat{f} + \nabla \hat{f}^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \right) \\ 0 &\geq \hat{g} + \nabla \hat{g}^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \right) \end{aligned}$$

Then, it follows that for $y = \hat{y}$, and for any x , the value $\eta \geq \hat{f}$

Infeasibility Lemma

Fix integer variables $\hat{y}$ and let $\hat{x}$ solve FNLP

$$\text{FNLP}(\hat{y}) \left\{ \begin{array}{ll} \underset{x}{\text{minimize}} & \hat{h}(x, \hat{y}) := \sum_{i \in \mathcal{I}} g_i(x, \hat{y}) \\ \text{subject to} & g_i(x, \hat{y}) \leq 0, \quad i \in \{1, \dots, m\} \setminus \mathcal{I} \end{array} \right.$$

with optimal value, $\hat{h} = h(\hat{x}, \hat{y}) \geq \tau > 0$, bounded away from zero

Lemma (Infeasibility Lemma)

Let $\hat{x}$ solve $\text{FNLP}(\hat{y})$, and consider the linear constraints

$$0 \geq \hat{g} + \nabla \hat{g}^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \right)$$

Then, it follows that for $y = \hat{y}$, there exists no $\hat{x}$ that satisfies these constraints

Stronger statement than Optimality Lemma! Optimality Lemma gives a bound, Infeasibility Lemma excludes a value.

KKT & Outer Approximation: Summary and Take-Away Points

- Linearizations of convex functions are underestimators
- Showed Optimality and Infeasibility Lemma
 - ⇒ Linearizing around NLP-optimal points excludes previous integers
- Lemmas form basis for Outer Approximation algorithms

Outline

- 1 Recap of Nonlinear Optimization
- 2 Introduction to Mixed-Integer Optimization
- 3 Multi-Tree Methods: Outer Approximation, Benders Decomposition, ECP**
- 4 A Recent Algorithm and Implementation
- 5 Some Applications from Optimal Control
- 6 Summary and Take-Away Points

Multi-Tree Methods

MILP solvers much better developed than MINLP solvers

- LPs are easy to hot-start
- Decades of investment into software
- MILPs much easier; e.g. no need for nonlinear constraint qualifications

⇒ developed methods that exploit this technology

Multi-Tree Methods

- Outer approximation [[Duran and Grossmann, 1986](#)]
- Benders decomposition [[Geoffrion, 1972](#)]
- Extended cutting plane method [[Westerlund and Pettersson, 1995](#)]

... solve an alternating sequence of MILP and NLP problems

Solving a convex MINLP

Reformulate in epigraph form and separate linear constraints

$$\begin{array}{ll}\min_{x,y} f(x,y) & \min_{x,y} \eta \\ \text{subject to } g(x,y) \leq 0, & \text{subject to } f(x,y) - \eta \leq 0, \quad (\text{objective}) \\ y \in \mathbb{Z}^{n_y}, & g(x,y) \leq 0, \quad (\text{nonlinear constraints}) \\ & Ax + Dy \leq h, \quad (\text{linear constraints}) \\ & y \in \mathbb{Z}^{n_y},\end{array}$$

Suppose we ignore all nonlinear constraints in the first MILP:

$$\begin{array}{ll}\min_{x,y} \eta \\ \text{subject to } f(x,y) - \eta \leq 0, & (\text{Suppose } f \text{ is linear}) \\ Ax + Dy \leq h, & (\text{linear constraints}) \\ y \in \mathbb{Z}^{n_y}\end{array}$$

Solving MILPs

- Optimal objective value of the MILP is a lower bound to MINLP. Why?

Solving MILPs

- Optimal objective value of the MILP is a lower bound to MINLP. Why?
- If the above MILP is **infeasible**, STOP.

Solving MILPs

- Optimal objective value of the MILP is a lower bound to MINLP. Why?
- If the above MILP is **infeasible**, STOP.
- If the solution (x^*, y^*) satisfies all nonlinear constraints, STOP

Solving MILPs

- Optimal objective value of the MILP is a lower bound to MINLP. Why?
- If the above MILP is **infeasible**, STOP.
- If the solution (x^*, y^*) satisfies all nonlinear constraints, STOP
- If the above MILP is unbounded, we are in trouble.

Solving MILPs

- Optimal objective value of the MILP is a lower bound to MINLP. Why?
- If the above MILP is **infeasible**, STOP.
- If the solution (x^*, y^*) satisfies all nonlinear constraints, STOP
- If the above MILP is unbounded, we are in trouble.
- **Most common case: If an MILP solution violates a nonlinear constraint, we need to cut it off. How?**

Solving MILPs

- Optimal objective value of the MILP is a lower bound to MINLP. Why?
- If the above MILP is **infeasible**, STOP.
- If the solution (x^*, y^*) satisfies all nonlinear constraints, STOP
- If the above MILP is unbounded, we are in trouble.
- **Most common case: If an MILP solution violates a nonlinear constraint, we need to cut it off. How?**
- Convexity helps
- Suppose $g_1(x^*, y^*) > 0$
- Consider linearization inequality at (x^*, y^*)

$$g_1(x^*, y^*) + \nabla g_1^T \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^* \\ y^* \end{pmatrix} \right) \leq 0$$

- It is a valid inequality for the MINLP
- It is violated by (x^*, y^*)

An algorithm

- One can add the above linearization to the MILP and solve the new MILP
- (x^*, y^*) will then be avoided, and no solution of MINLP is cut
- Crude algo: MILP $\rightarrow$ linearization cut $\rightarrow$ MILP ...
- Extended cutting plane method (Westerlund et al, 1995, 2001)

An algorithm

- One can add the above linearization to the MILP and solve the new MILP
- (x^*, y^*) will then be avoided, and no solution of MINLP is cut
- Crude algo: MILP $\rightarrow$ linearization cut $\rightarrow$ MILP ...
- Extended cutting plane method (Westerlund et al, 1995, 2001)
- Limitation: may visit the same y^* again but a different x .

Role of NLP

Instead of a linearization cut at (x^*, y^*) can we try a different point?

Role of NLP

Instead of a linearization cut at (x^*, y^*) can we try a different point?

The OA idea: Solve a 'fixed NLP'!

Fix integer variables y^* and let $\hat{x}$ solve

$$\text{NLP}(y^*) \begin{cases} \underset{x}{\text{minimize}} & f(x, y^*) \\ \text{subject to} & g(x, y^*) \leq 0. \end{cases}$$

Let $\hat{f} := f(\hat{x}, y^*)$.

If $\text{NLP}(y^*)$ is feasible:

- $(\hat{x}, y^*)$ is feasible for the MINLP
- $\hat{f}$ gives an upper bound on the MINLP optimal value
- Add linearizations of f and g at $(\hat{x}, y^*)$

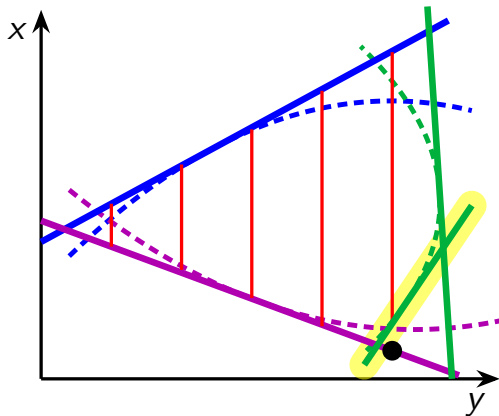
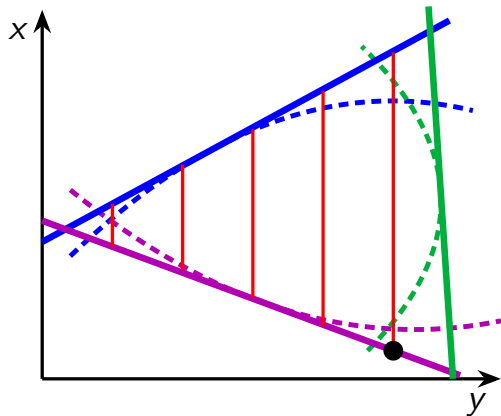
Infeasible Fixed NLP

If $\text{NLP}(y^*)$ is infeasible, choose a set $\mathcal{I}$ of constraints to relax and solve the nonlinear feasibility problem

$$\text{FNLP}(y^*) \quad \begin{cases} \underset{x}{\text{minimize}} & \hat{h}(x, y^*) := \sum_{i \in \mathcal{I}} g_i(x, y^*) \\ \text{subject to} & g_i(x, y^*) \leq 0, \quad i \in \mathcal{F} := \{1, \dots, m\} \setminus \mathcal{I}. \end{cases}$$

- The constraints in $\mathcal{F}$ remain hard constraints
- The constraints in $\mathcal{I}$ are moved into the feasibility objective
- If optimal value is positive, then no x satisfies $g(x, y^*) \leq 0$
- Linearizations generated at $(\hat{x}, y^*)$ are used to exclude the integer assignment y^*

Illustration of Infeasible Fixed NLP



Separate infeasible points by

- Solving NLP for fixed integers to generate cut
- Collect cuts in MILP master problem

Outer Approximation MILP Master

Define a set that collects all visited points, $\mathcal{F}$

$$\mathcal{F} := \left\{ x^{(j)} \in \mathcal{X} : x^{(j)} \text{ solves } (\text{NLP}(y^{(j)})) \text{ or } (\text{FNLP}(Y^{(j)})) \right\}.$$

... boundedness of $\mathcal{X}$ implies $|\mathcal{F}| < \infty$

Construct **equivalent OA-MILP** (outer approximation MILP)

$$\left\{ \begin{array}{ll} \underset{\eta, x, y}{\text{minimize}} & \eta \\ \text{subject to} & \eta \geq f^{(j)} + \nabla f^{(j)\top} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^{(j)} \\ y^{(j)} \end{pmatrix} \right), \quad \forall (x^{(j)}, y^{(j)}) \in \mathcal{F}, \\ & 0 \geq g^{(j)} + \nabla g^{(j)\top} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^{(j)} \\ y^{(j)} \end{pmatrix} \right), \quad \forall (x^{(j)}, y^{(j)}) \in \mathcal{F}, \\ & x \in \mathcal{X} \cap \mathbb{R}^{n_x}, y \in \mathcal{Y} \cap \mathbb{Z}^{n_y}. \end{array} \right.$$

Outer Approximation Algorithm

Solving OA-MILP clearly not sensible; define upper bound as

$$U^k := \min_{j \leq k} \left\{ f^{(j)} \mid (\text{NLP}(y^{(j)})) \text{ is feasible} \right\}.$$

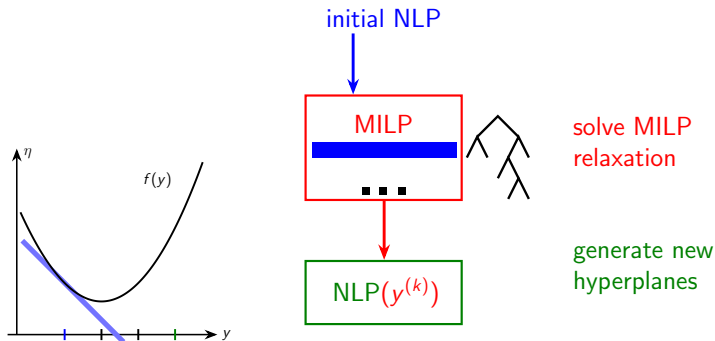
Define relaxation of OA-MILP, using $\mathcal{F}^k \subset \mathcal{F}$, with $\mathcal{F}^0 = \{\}$

$$M(\mathcal{F}^k) \left\{ \begin{array}{l} \underset{\eta, x, y}{\text{minimize}} \quad \eta \\ \text{subject to} \quad \eta \leq U^k - \epsilon \\ \eta \geq f^{(j)} + \nabla f^{(j)\top} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^{(j)} \\ y^{(j)} \end{pmatrix} \right), \quad \forall (x^{(j)}, y^{(j)}) \in \mathcal{F}^k, \\ 0 \geq g^{(j)} + \nabla g^{(j)\top} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^{(j)} \\ y^{(j)} \end{pmatrix} \right), \quad \forall (x^{(j)}, y^{(j)}) \in \mathcal{F}^k, \\ x \in \mathcal{X} \cap \mathbb{R}^{n_x}, \quad y \in \mathcal{Y} \cap \mathbb{Z}^{n_y}. \end{array} \right.$$

... build up better OA $\mathcal{F}^k$ iteratively for $k = 0, 1, \dots$

Outer Approximation Algorithm

Alternate between solve $\text{NLP}(y^{(k)})$ and MILP relaxation

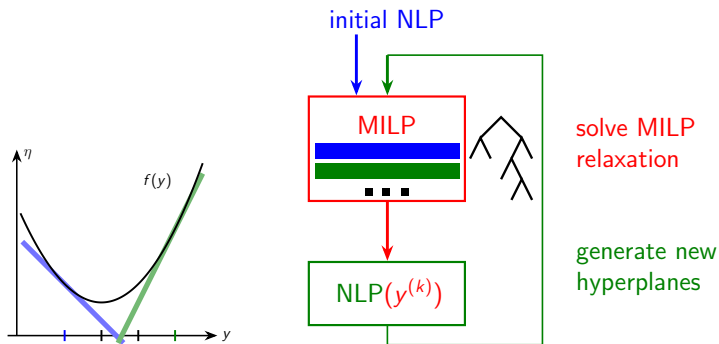


MILP $\Rightarrow$ lower bound; NLP $\Rightarrow$ upper bound

... convergence follows from convexity & finiteness

Outer Approximation Algorithm

Alternate between solve $\text{NLP}(y^{(k)})$ and MILP relaxation



MILP $\Rightarrow$ lower bound; NLP $\Rightarrow$ upper bound

... convergence follows from convexity & finiteness

Outer Approximation Algorithm

Outer approximation ;

Given $x^{(0)}$, choose $\text{tol } \epsilon > 0$, set $U^{-1} = \infty$, set $k = 0$, and $\mathcal{F}^{-1} = \emptyset$. ;

repeat

 Solve (NLP($y^{(k)}$) or FNLP($y^{(k)}$)); solution $x^{(k)}$;

 if (NLP($y^{(k)}$)) feasible & $f^{(k)} < U^{k-1}$ then

 Update best point: $(x^*, y^*) = (x^{(k)}, y^{(k)})$ and $U^k = f^{(k)}$;

 else

 Set $U^k = U^{k-1}$;

 end

 Linearize f and g about $(x^{(k)}, y^{(k)})$ and set $\mathcal{F}^k = \mathcal{F}^{k-1} \cup \{k\}$. ;

 Solve ($M(\mathcal{F}^k)$), let solution be $(x^{(k)}, y^{(k)})$ & set $k = k + 1$. ;

until MILP ($M(\mathcal{F}^k)$) is infeasible;

Convergence and Limitations of Outer Approximation

Theorem (Convergence of Outer Approximation)

Under convexity, boundedness, suitable constraint qualification, and exact solution of the subproblems, outer approximation terminates finitely at an optimal solution of the MINLP or indicates that the MINLP is infeasible.

Main ideas.

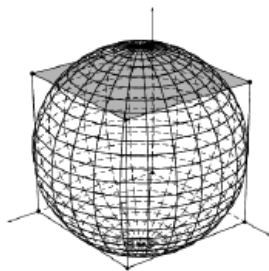
- Convexity implies that linearizations are globally valid supporting hyperplanes
- For feasible $\text{NLP}(y^{(j)})$, the OA cuts imply $\eta \geq f^{(j)}$ when $y = y^{(j)}$
- For infeasible $\text{NLP}(y^{(j)})$, feasibility cuts exclude $y = y^{(j)}$
- Boundedness implies finitely many integer assignments

Finite convergence does not imply fast convergence: OA may visit exponentially many integer assignments in the worst case.

Worst Case Example of Outer Approximation

[Hijazi et al., 2010] construct **infeasible** MINLP:

$$\begin{array}{ll}\underset{y}{\text{minimize}} & 0 \\ \text{subject to} & \sum_{i=1}^n \left(y_i - \frac{1}{2} \right)^2 \leq \frac{n-1}{4} \\ & y \in \{0, 1\}^n\end{array}$$



Intersection of ball of radius $\frac{\sqrt{n-1}}{2}$
with unit hypercube.

Lemma

*OA cannot cut more than one vertex of the hypercube
MILP master problem feasible for any $k < 2^n$ OA cuts*

Theorem

OA visits all 2^n vertices

Benders Decomposition: Motivation

Outer approximation keeps both x and y in the MILP master:

$$\underset{\eta, x, y}{\text{minimize}} \quad \eta.$$

Benders decomposition uses optimal multipliers of the fixed NLP to eliminate the continuous variables x from the cuts.

Let $x^{(j)}$ solve $\text{NLP}(y^{(j)})$ with optimal multipliers

$$\lambda^{(j)} \geq 0$$

for

$$g(x, y^{(j)}) \leq 0.$$

Define the Lagrangian

$$\mathcal{L}(x, y, \lambda) := f(x, y) + \lambda^T g(x, y).$$

The Benders master contains only η and the integer variables y .

Benders Decomposition: Optimality Cut

Deriving Benders cut from outer approximation cuts:

- Let $x^{(j)}$ solve $(\text{NLP}(y^{(j)}))$ with **optimal multipliers** $\mu^{(j)} \geq 0$
- Summing outer approximation cuts weighted by $(1, \mu^{(j)}) \geq 0$ gives valid inequality

$$\begin{array}{rcl} \eta & \geq & f^{(j)} + \nabla f^{(j)T} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^{(j)} \\ y^{(j)} \end{pmatrix} \right) \\ + \mu^{(j)T} & \left(0 \geq \right. & \left. g^{(j)} + \nabla g^{(j)T} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^{(j)} \\ y^{(j)} \end{pmatrix} \right) \right) \\ \hline & \eta \geq & f^{(j)} + \nabla_y \mathcal{L}^{(j)T} (y - y^{(j)}) \end{array}$$

- Using KKT conditions wrt x : $0 = \nabla_x \mathcal{L}^{(j)} = \nabla_x f + \nabla_x g \mu^{(j)}$ & $\mu^{(j)T} g^{(j)} = 0$
... eliminates continuous variables, x

Benders cut only involves integer variables y .

Benders Decomposition: Feasibility Cut

Deriving Benders feasibility cut from outer approximation cuts

Summing with **optimal multipliers** $\mu^{(j)}$ of (FNLP($y^{(j)}$)) (with Lagrangian $\mathcal{L}_F$) gives:

$$\begin{array}{rcl} e^T \left(0 \geq g_{\mathcal{I}}^{(j)} + \nabla g_{\mathcal{I}}^{(j)T} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^{(j)} \\ x^{(j)} \end{pmatrix} \right) \right) \\ \mu^{(j)T} \left(0 \geq g_{\mathcal{F}}^{(j)} + \nabla g_{\mathcal{F}}^{(j)T} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x^{(j)} \\ x^{(j)} \end{pmatrix} \right) \right) \\ \hline 0 \geq e^T g_{\mathcal{I}}^{(j)} + \nabla_y \mathcal{L}_F^{(j)T} (y - y^{(j)}) \end{array}$$

Using KKT conditions wrt x : $0 = \nabla_x \mathcal{L}_F^{(j)} = \nabla_x g_{\mathcal{I}} + \nabla_x g_{\mathcal{F}} \mu^{(j)}$ & $\mu^{(j)T} g_{\mathcal{F}}^{(j)} = 0$
... eliminates continuous variables, x

Benders Decomposition Master Problem

Generating Benders cuts for all $y^{(j)}$ not sensible; define upper bound as

$$U^k := \min_{j \leq k} \left\{ f^{(j)} \mid (\text{NLP}(y^{(j)})) \text{ is feasible} \right\}.$$

Define relaxation of Benders-MILP, using $\mathcal{F}^k \subset \mathcal{F}$, with $\mathcal{F}^0 = \{\}$

$$M(\mathcal{F}^k) \left\{ \begin{array}{ll} \underset{\eta, x}{\text{minimize}} & \eta, \\ \text{subject to} & \eta \leq U^k - \epsilon \\ & \eta \geq f^{(j)} + \nabla \mathcal{L}^{(j)T} (y - y^{(j)}), \quad \forall y^{(j)} \in \mathcal{F}^k \\ & 0 \geq e^T g_{\mathcal{I}_j}^{(j)} + \nabla \mathcal{L}_F^{(j)T} (y - y^{(j)}), \quad \forall y^{(j)} \in \mathcal{F}^k \\ & y \in \mathbb{Z}^{n_y} \end{array} \right.$$

... build up better approx. $\mathcal{F}^k$ iteratively for $k = 0, 1, \dots$

Benders Decomposition Algorithm

Benders Decomposition ;

Given $y^{(0)}$, choose $\text{tol } \epsilon > 0$, set $U^{-1} = \infty$, set $k = 0$, and $\mathcal{F}^{-1} = \emptyset$;

repeat

 Solve (NLP($y^{(k)}$)) or (FNLP($y^{(k)}$)); solution $x^{(k)}$;

if (NLP($y^{(k)}$)) feasible & $f^{(k)} < U^{k-1}$ **then**

 Update best point: $(x^*, y^*) = (x^{(k)}, y^{(k)})$ and $U^k = f^{(k)}$;

else

 Set $U^k = U^{k-1}$;

 Linearize f and g about $(x^{(k)}, y^{(k)})$ and set $\mathcal{F}^k = \mathcal{F}^{k-1} \cup \{k\}$;

 Solve MILP master problem ($M(\mathcal{F}^k)$), let solution be $x^{(k+1)}$ & set $k = k + 1$;

until MILP ($M(\mathcal{F}^k)$) is infeasible;

Benders Decomposition: Fixing Integer Variables

Exercise (Practical NLP Subproblem)

(NLP($y^{(j)}$)) equivalent to full-space NLP with additional equations:

$$\underset{x,y}{\text{minimize}} \ f(x,y) \quad \text{subject to} \ 0 \geq g(x,y), \quad y = y^{(j)}$$

Let $\mu^{(j)}$ be the multipliers of $y = y^{(j)}$ and show that the Benders cut is

$$\eta \geq f^{(j)} + \lambda^{(j)T} (y - y^{(j)})$$

Outer Approximation, Benders, and ECP

Method	Point used for cuts	Master variables
ECP	MILP solution $(x^{(j)}, y^{(j)})$	(x, y, η)
OA	fixed-NLP solution $(x^{(j)}, y^{(j)})$	(x, y, η)
Benders	fixed-NLP solution and multipliers	(y, η)

- ECP avoids fixed NLP solves, but may revisit the same integer assignment
- OA keeps separate objective and constraint linearizations
- Benders aggregates OA cuts using optimal multipliers
- Benders cuts are generally weaker than the corresponding collection of OA cuts
- Classical Benders generates one cut per subproblem: a **single cut**
- That cut can involve many or all components of y : a **dense cut**

Outline

- 1 Recap of Nonlinear Optimization
- 2 Introduction to Mixed-Integer Optimization
- 3 Multi-Tree Methods: Outer Approximation, Benders Decomposition, ECP
- 4 A Recent Algorithm and Implementation
- 5 Some Applications from Optimal Control
- 6 Summary and Take-Away Points

Refresher and a new idea¹

Considered MINLP

$$\begin{array}{ll}\min_{x,y} & f(x,y) \\ \text{s.t.} & g(x,y) \leq 0, \\ & h(x,y) = 0, \\ & x \in X \subseteq \mathbb{R}^{n_x}, \\ & y \in Y \subseteq \mathbb{Z}^{n_y}.\end{array}$$

Def. *convex* and *nonconvex* MINLPs

¹Ghezzi, Van Roy, Sager, and Diehl (2026). *A Sequential Benders-based Mixed-Integer Quadratic Programming Algorithm and Its Implementation in the CAMINO Toolbox*. Mathematical Programming Computation

Refresher and a new idea¹

Considered MINLP

$$\begin{array}{ll}\min_{x,y} & f(x,y) \\ \text{s.t.} & g(x,y) \leq 0, \\ & h(x,y) = 0, \\ & x \in X \subseteq \mathbb{R}^{n_x}, \\ & y \in Y \subseteq \mathbb{Z}^{n_y}.\end{array}$$

Def. *convex* and *nonconvex* MINLPs

Proposed approach: S-B-MIQP

Combine ideas from:

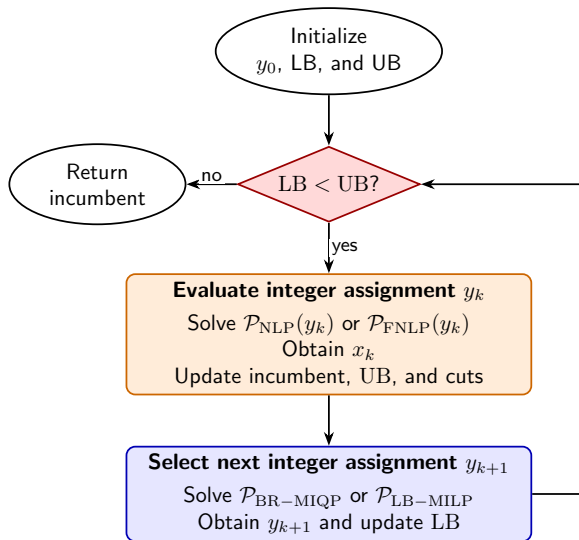
- ▶ **SQP**: quadratic local models
 - ▶ fast progress
- ▶ **OA/GBD**: cuts and global lower bounds
 - ▶ solution certification

Properties:

- ▶ Global minimizers for *convex* MINLPs
- ▶ Heuristic for *nonconvex* MINLPs

¹Ghezzi, Van Roy, Sager, and Diehl (2026). *A Sequential Benders-based Mixed-Integer Quadratic Programming Algorithm and Its Implementation in the CAMINO Toolbox*. Mathematical Programming Computation

S-B-MIQP: algorithm at a glance



$\mathcal{P}_{\text{NLP}}$ and $\mathcal{P}_{\text{FNLP}}$

- Evaluate a fixed integer assignment
- Improve the incumbent
- Generate cuts

Master problems:

- $\mathcal{P}_{\text{BR-MIQP}}$: quadratic local search
- $\mathcal{P}_{\text{LB-MILP}}$: compute lower bounds

Convex MINLPs: certification
Nonconvex MINLPs: heuristic

$\mathcal{P}_{\text{NLP}}$ – Obtaining value-function underestimators

Solve the original MINLP for a fixed integer assignment $\tilde{y} \in Y$

$$\begin{aligned} J(\tilde{y}) := \min_{x \in X} \quad & f(x, \tilde{y}) \\ \text{s.t.} \quad & g(x, \tilde{y}) \leq 0, \\ & h(x, \tilde{y}) = 0 \end{aligned}$$

$\mathcal{P}_{\text{NLP}}$ – Obtaining value-function underestimators

Solve the original MINLP for a fixed integer assignment $\tilde{y} \in Y$

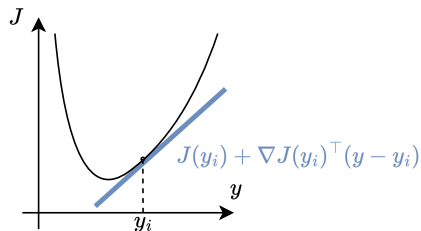
$$\begin{aligned} J(\tilde{y}) &:= \min_{x \in X, y \in \bar{Y}} f(x, y) \\ \text{s.t.} \quad &g(x, y) \leq 0, \\ &h(x, y) = 0, \\ &y - \tilde{y} = 0. \quad (\mu_{\tilde{y}}) \end{aligned}$$

- ▶ Lift y as a continuous variable and impose $y - \tilde{y} = 0$
- ▶ The multiplier $\mu_{\tilde{y}}$ is a subgradient of $J(\tilde{y}) \rightarrow \nabla J(\tilde{y}) := -\mu_{\tilde{y}}$
 - ▶ Under MFCQ, $-\mu_{\tilde{y}} \in \partial_C J(\tilde{y})$

$\mathcal{P}_{\text{NLP}}$ – Obtaining value-function underestimators

Solve the original MINLP for a fixed integer assignment $\tilde{y} \in Y$

$$\begin{aligned} J(\tilde{y}) &:= \min_{x \in X, y \in \bar{Y}} f(x, y) \\ \text{s.t.} \quad & g(x, y) \leq 0, \\ & h(x, y) = 0, \\ & y - \tilde{y} = 0. \end{aligned} \quad (\mu_{\tilde{y}})$$



- Lift y as a continuous variable and impose $y - \tilde{y} = 0$
- The multiplier $\mu_{\tilde{y}}$ is a subgradient of $J(\tilde{y}) \rightarrow \nabla J(\tilde{y}) := -\mu_{\tilde{y}}$
 - Under MFCQ, $-\mu_{\tilde{y}} \in \partial_C J(\tilde{y})$

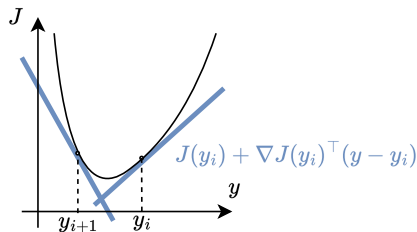
For convex MINLPs we can construct **global affine underestimators** of J :

$$J(y_i) + \nabla J(y_i)^\top (y - y_i) \leq J(y), \quad i \in \mathbb{T}_k.$$

$\mathcal{P}_{\text{NLP}}$ – Obtaining value-function underestimators

Solve the original MINLP for a fixed integer assignment $\tilde{y} \in Y$

$$\begin{aligned} J(\tilde{y}) &:= \min_{x \in X, \mathbf{y} \in \bar{Y}} f(x, y) \\ \text{s.t.} \quad &g(x, y) \leq 0, \\ &h(x, y) = 0, \\ &\mathbf{y} - \tilde{\mathbf{y}} = \mathbf{0}. \end{aligned} \quad (\mu_{\tilde{y}})$$



- Lift $\mathbf{y}$ as a continuous variable and impose $\mathbf{y} - \tilde{\mathbf{y}} = \mathbf{0}$
- The multiplier $\mu_{\tilde{\mathbf{y}}}$ is a subgradient of $J(\tilde{\mathbf{y}}) \rightarrow \nabla J(\tilde{\mathbf{y}}) := -\mu_{\tilde{\mathbf{y}}}$
 - Under MFCQ, $-\mu_{\tilde{\mathbf{y}}} \in \partial_C J(\tilde{\mathbf{y}})$

For convex MINLPs we can construct **global affine underestimators** of J :

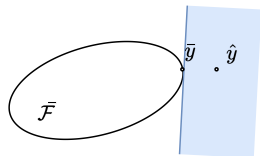
$$J(y_i) + \nabla J(y_i)^\top (y - y_i) \leq J(y), \quad i \in \mathbb{T}_k.$$

$\mathcal{P}_{\text{FNLP}}$ - Obtaining infeasibility cuts

For some $\hat{y} \in Y$, the problem $\mathcal{P}_{\text{NLP}}$ which we aim to solve might be infeasible.

$\mathcal{P}_{\text{FNLP}}$ **projects** $\hat{y}$ onto the relaxed feasible set and **constructs an infeasibility cut**.

$$\begin{aligned} \min_{x \in X, y \in \bar{Y}} \quad & \|y - \hat{y}\|_2^2 \\ \text{s.t.} \quad & g(x, y) \leq 0, \\ & h(x, y) = 0. \end{aligned}$$

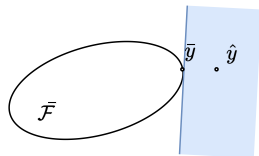


$\mathcal{P}_{\text{FNLP}}$ - Obtaining infeasibility cuts

For some $\hat{y} \in Y$, the problem $\mathcal{P}_{\text{NLP}}$ which we aim to solve might be infeasible.

$\mathcal{P}_{\text{FNLP}}$ **projects** $\hat{y}$ onto the relaxed feasible set and **constructs an infeasibility cut**.

$$\begin{aligned} \min_{x \in X, y \in \bar{Y}} \quad & \|y - \hat{y}\|_2^2 \\ \text{s.t.} \quad & g(x, y) \leq 0, \\ & h(x, y) = 0. \end{aligned}$$



- ▶ Let $(\bar{x}_i, \bar{y}_i), i \in \mathbb{S}_k$ denote the solution of $\mathcal{P}_{\text{FNLP}}$
- ▶ The **infeasibility cut** is defined as $(\hat{y}_i - \bar{y}_i)^\top (y - \bar{y}_i) \leq 0$
- ▶ For a convex MINLP the cuts exclude $\hat{y}_i$

Notation. k algorithm iteration; $\mathbb{T}_k$ index set with feasible y_i ; $\mathbb{S}_k$ index set with infeasible y_i

Master problems – $\mathcal{P}_{\text{BR-MIQP}}$

Notation. $b(k)$ incumbent solution; f_L , g_L and h_L first order Taylor series of f , g and h

$$\begin{aligned} \min_{x \in X, y \in Y} \quad & f_L(x, y; x_{b(k)}, y_{b(k)}) + \frac{1}{2} \begin{pmatrix} x - x_{b(k)} \\ y - y_{b(k)} \end{pmatrix}^\top B_{b(k)} \begin{pmatrix} x - x_{b(k)} \\ y - y_{b(k)} \end{pmatrix} \\ \text{s.t.} \quad & g_L(x, y; x_{b(k)}, y_{b(k)}) \leq 0, \\ & h_L(x, y; x_{b(k)}, y_{b(k)}) = 0, \\ & J(y_i) + \nabla J(y_i)^\top (y - y_i) \leq \bar{J}_k, \quad i \in \mathbb{T}_k \text{ (feasible iterations),} \\ & (y_i - \bar{y}_i)^\top (y - \bar{y}_i) \leq 0, \quad i \in \mathbb{S}_k \text{ (infeasible iterations),} \end{aligned}$$

where $\bar{J}_k := \alpha J(y_{b(k)}) + (1 - \alpha)\text{LB}$, $\alpha \in [0, 1)$.

- ▶ Quadratic model: fast local progress around the incumbent
- ▶ **Benders region**: excludes visited points and focuses the search
- ▶ Output: the next candidate integer assignment

Master problems – $\mathcal{P}_{\text{BR-MIQP}}$

Notation. $b(k)$ incumbent solution; f_L , g_L and h_L first order Taylor series of f , g and h

$$\min_{x \in X, y \in Y} \quad f_L(x, y; x_{b(k)}, y_{b(k)}) + \frac{1}{2} \begin{pmatrix} x - x_{b(k)} \\ y - y_{b(k)} \end{pmatrix}^\top B_{b(k)} \begin{pmatrix} x - x_{b(k)} \\ y - y_{b(k)} \end{pmatrix}$$

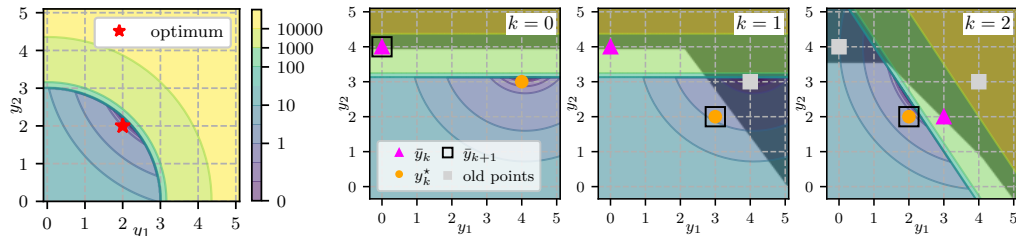
$$\text{s.t.} \quad g_L(x, y; x_{b(k)}, y_{b(k)}) \leq 0,$$

$$h_L(x, y; x_{b(k)}, y_{b(k)}) = 0,$$

$$J(y_i) + \nabla J(y_i)^\top (y - y_i) \leq \bar{J}_k, \quad i \in \mathbb{T}_k \text{ (feasible iterations),}$$

$$(y_i - \bar{y}_i)^\top (y - \bar{y}_i) \leq 0, \quad i \in \mathbb{S}_k \text{ (infeasible iterations),}$$

where $\bar{J}_k := \alpha J(y_{b(k)}) + (1 - \alpha)\text{LB}$, $\alpha \in [0, 1]$.



Master problems – $\mathcal{P}_{\text{LB-MILP}}$

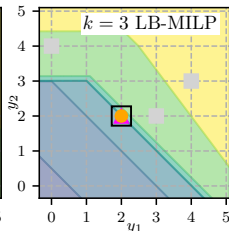
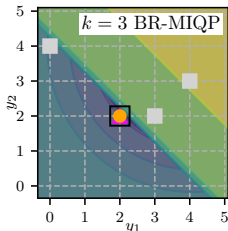
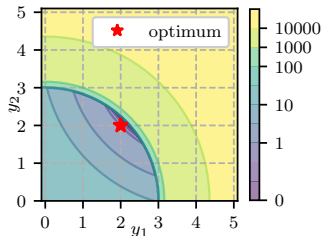
- Provide a **valid lower bound** for *convex* MINLPs and **certify global optimality** when $\text{LB} = \text{UB}$

$$\begin{aligned} \min_{\eta \in \mathbb{R}, x \in X, y \in Y} \quad & \eta \\ \text{s.t.} \quad & \eta \geq f_{\text{L}}(x, y; x_{b(k)}, y_{b(k)}), & \text{if } b(k) \in \mathbb{T}_k, \\ & 0 \geq g_{\text{L}}(x, y; x_{b(k)}, y_{b(k)}), & \text{if } b(k) \in \mathbb{T}_k, \\ & 0 = h_{\text{L}}(x, y; x_{b(k)}, y_{b(k)}), & \text{if } b(k) \in \mathbb{T}_k, \\ & \eta \geq J(y_i) + \nabla J(y_i)^\top (y - y_i), & i \in \mathbb{T}_k \setminus \{b(k)\}, \\ & 0 \geq (y_i - \bar{y}_i)^\top (y - \bar{y}_i), & i \in \mathbb{S}_k. \end{aligned}$$

Master problems – $\mathcal{P}_{\text{LB-MILP}}$

- Provide a **valid lower bound** for *convex* MINLPs and **certify global optimality** when $\text{LB} = \text{UB}$

$$\begin{aligned}
 & \min_{\eta \in \mathbb{R}, x \in X, y \in Y} \quad \eta \\
 & \text{s.t.} \quad \eta \geq f_L(x, y; x_{b(k)}, y_{b(k)}), & \text{if } b(k) \in \mathbb{T}_k, \\
 & \quad 0 \geq g_L(x, y; x_{b(k)}, y_{b(k)}), & \text{if } b(k) \in \mathbb{T}_k, \\
 & \quad 0 = h_L(x, y; x_{b(k)}, y_{b(k)}), & \text{if } b(k) \in \mathbb{T}_k, \\
 & \quad \eta \geq J(y_i) + \nabla J(y_i)^\top (y - y_i), & i \in \mathbb{T}_k \setminus \{b(k)\}, \\
 & \quad 0 \geq (y_i - \bar{y}_i)^\top (y - \bar{y}_i), & i \in \mathbb{S}_k.
 \end{aligned}$$



S-B-MIQP: implementation and evaluation

Implementation:

- ▶ Implemented in Python within the open-source CAMINO toolbox
- ▶ MINLPs modeled using CasADi or AMPL
- ▶ NLP and MIQP/MILP solvers interfaced via CasADi
- ▶ Numerical results shown here use:
 - ▶ IPOPT for NLPs
 - ▶ Gurobi for MIQP/MILP master problems

Evaluation:

- ▶ Broad benchmark on generic convex and nonconvex MINLPs from MINLPLib
- ▶ Application to mixed-integer optimal control problems

Reproducibility: Algorithm and benchmark code are open-source within CAMINO

The CAMINO toolbox

CAMINO is a open-source toolbox to solve MINLPs, two aims:

- ▶ **Flexibility**: rapid prototyping of novel ideas via high level object-oriented Python-code
- ▶ **Efficiency**: leverages CasADi² for symbolic modelling, automatic differentiation and interfacing NLP and MIP solvers

CAMINO can be easily installed as PyPi package

```
pip install caminopy
```

CAMINO implements several algorithms for MINLPs

- ▶ outer approximation, Benders decomposition, feasibility pump, S-B-MIQP

²Andersson, Gillis, Horn, Rawlings, and Diehl (2019). *CasADi – A software framework for nonlinear optimization and optimal control*. Mathematical Programming Computation

Formulating and solving problems in CAMINO

Option 1. Directly solve AMPL problem from command line

```
python -m camino run <solver_name> nl_file --args="path/to/nlfile"
```

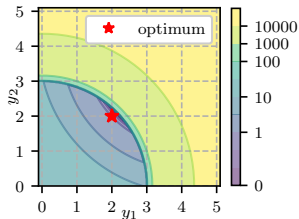
Option 2. Formulate the MINLP using CasADi from Python, then pass it to CAMINO

```
import camino
problem = MinlpProblem(f, g, x, p, idx_x_integer=is_integer)
data = MinlpData(p_val, x0, lbx, ubx, lbg, ubg)
settings = Settings()
settings.MIP_SOLVER = "highs" # gurobi
setting.MINLP_TOLERANCE = 1e-4
stats = Stats("oa", "example-problem")
solver = MinlpSolver("oa", problem, data, stats, settings)
result = solver.solve(data)
solver.stats.print()
```

Modelling example in CasADi and intro to CAMINO Python API

$$\begin{aligned} \min_{\substack{x \in \mathbb{R}, \\ (y_{[1]}, y_{[2]}) \in \mathbb{Z}^2}} \quad & (y_{[1]} - 4.1)^2 + (y_{[2]} - 4.0)^2 + \lambda x \\ \text{s.t.} \quad & 0 \leq y_{[1]}^2 + y_{[2]}^2 - r^2 - x, \\ & 0 \leq x, \\ & 0 \leq y_{[i]} \leq 5, \quad i = 1, 2, \end{aligned}$$

where $\lambda = 1000$ and $r = 3$.



```
import casadi as ca
import numpy as np
x = ca.SX.sym("x", 1)
y = ca.SX.sym("y", 2)
p = ca.SX.sym("p", 2)
f = (y[0] - 4.1) ** 2 + (y[1] - 4.0) ** 2 +
    x * p[0]
g = y[0] ** 2 + y[1] ** 2 - x - p[1] ** 2
x0 = np.array([0, 4, 100])
lbx = np.array([0, 0, 0])
ubx = np.array([5, 5, ca.inf])
ubg = np.array([ca.inf])
lbg = np.array([0])
p_val = np.array([1000, 3])
problem = MinlpProblem(f, g, ca.vcat(x, y),
    p, idx_x_integer=[0, 1, 1])
data = MinlpData(x0, lbx, ubx, lbg, ubg,
    p_val)
```

Available algorithms in CAMINO

Solver	Description	Global solution for convex MINLP
bonmin-bb	Nonlinear branch-and-bound in Bonmin	✓
bonmin-oa	Outer approximation in Bonmin	✓
fp	Feasibility pump	
gbd	Generalized Benders decomposition	✓
oa	Outer approximation	✓
oa-qp	Quadratic outer approximation	
s-b-miqp ³	Sequential Benders-based MIQP	✓
s-b-miqp-early-exit	Sequential Benders-based MIQP early termination heuristic	
...	for full list see README	

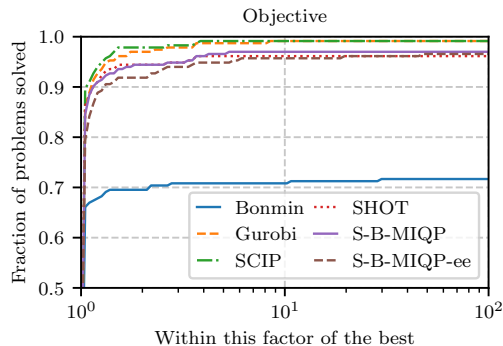
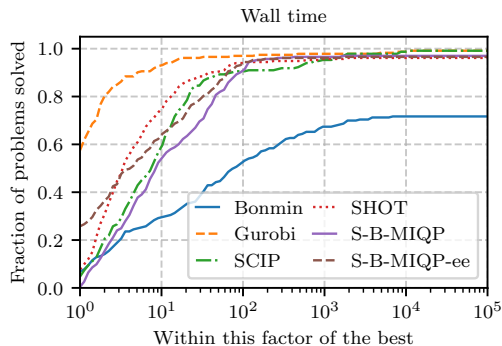
It is possible to “concatenate” solvers $\Rightarrow$ fp+s-b-miqp

The solution obtained by fp will be used as initial guess for s-b-miqp

³Ghezzi, Van Roy, Sager, and Diehl (2026). *A Sequential Benders-based Mixed-Integer Quadratic Programming Algorithm and Its Implementation in the CAMINO Toolbox*. Mathematical Programming Computation

Benchmark on MINLPLib instances – Results

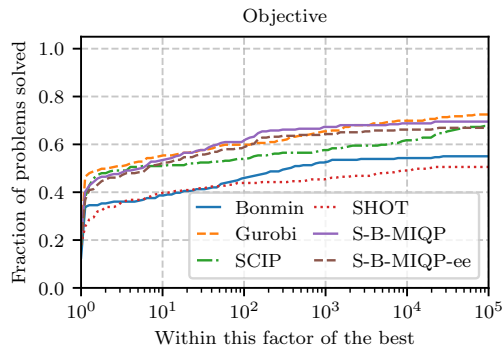
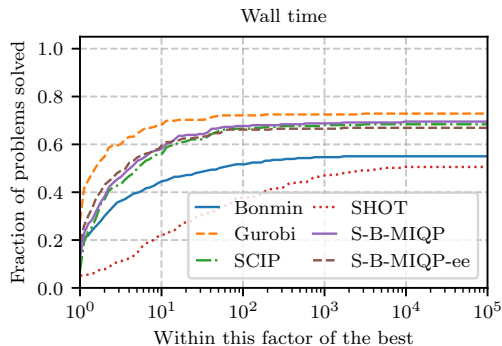
Results on 233 **convex MINLP** instances from the public collection MINLPLib



www.github.com/minlp-toolbox/CAMINO-benchmark

Benchmark on MINLPLib instances – Results

Results on 263 **nonconvex MINLP** instances from the public collection MINLPLib



www.github.com/minlp-toolbox/CAMINO-benchmark

Outline

- 1 Recap of Nonlinear Optimization
- 2 Introduction to Mixed-Integer Optimization
- 3 Multi-Tree Methods: Outer Approximation, Benders Decomposition, ECP
- 4 A Recent Algorithm and Implementation
- 5 Some Applications from Optimal Control**
- 6 Summary and Take-Away Points

Optimal control with discrete actuators

Consider an optimal control problem where some control variables are discrete

$$\begin{aligned} \min_{x_{0:N}, u_{0:N-1}} \quad & L_N(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k) \\ \text{s.t.} \quad & x_0 = \bar{x}_0, \\ & x_{k+1} = f^{\text{dyn}}(x_k, u_k), \\ & c(x_k, u_k) \leq 0, \\ & c_N(x_N) \leq 0, \\ & u_{i,k} \in \mathcal{U}_i \subseteq \mathbb{Z}, i \in \mathcal{I}, \\ & k \in \mathbb{Z}_0^{N-1}. \end{aligned}$$

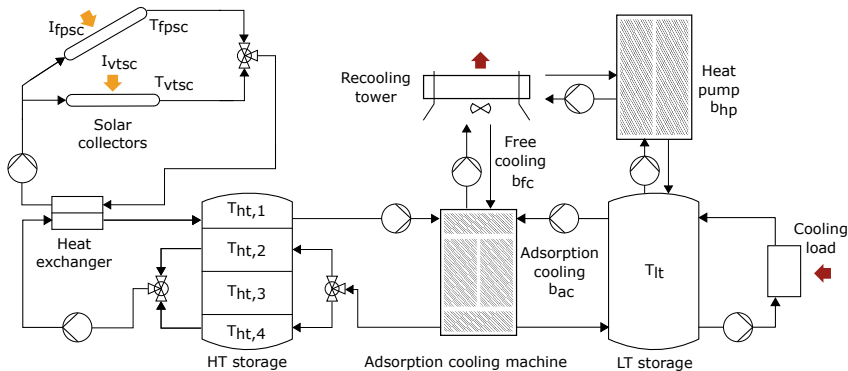
Optimal control with discrete actuators

Consider an optimal control problem where some control variables are discrete

$$\begin{aligned} \min_{x_{0:N}, u_{0:N-1}} \quad & L_N(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k) \\ \text{s.t.} \quad & x_0 = \bar{x}_0, \\ & x_{k+1} = f^{\text{dyn}}(x_k, u_k), \\ & c(x_k, u_k) \leq 0, \\ & c_N(x_N) \leq 0, \\ & u_{i,k} \in \mathcal{U}_i \subseteq \mathbb{Z}, i \in \mathcal{I}, \\ & k \in \mathbb{Z}_0^{N-1}. \end{aligned}$$

Do you have any example of such problems?

Control of a climatization system for a building

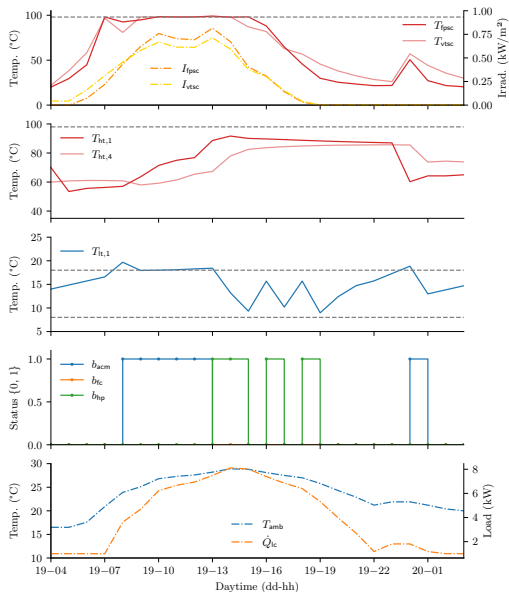


System dynamics is modelled by a nonlinear ODE: 19 differential states, 6 continuous controls, 3 binary controls

Discretize-then-Optimize: uniform grid with 24 intervals each one of 1 hour length, Gauss-Radau collocation with order 3, add 24 slack variables for soft constraints

Total variables: 2160 of which **72 binaries**

Control of a climatization system for a building



Algorithm	Objective	Runtime (mm:ss)	Timestamp Best Solution
<i>Relaxed</i> NLP	1547.33	00:05	-
Bonmin	2409.66	30:00	02:43
CIA (PyCombin)	6370.03	00:08	-
S-B-MIQP	2288.40	30:00	01:28
S-B-MIQP-ee	2288.40	30:00	01:28

Optimal control with logical decisions⁴

Focus on discrete-time optimal control problems with logical decisions

$$\min_{\substack{x_{0:N}, u_{0:N-1}, \\ \delta_{0:N-1}}} L_N(x_N) + \sum_{k=0}^{N-1} \left(L(x_k, u_k) - \sum_{i=1}^{n_\delta} w_i \delta_{k,i} \right)$$

$$\text{s.t. } x_0 = \bar{x}_0,$$

$$x_{k+1} = f^{\text{dyn}}(x_k, u_k),$$

$$c(x_k, u_k) \leq 0,$$

$$c_N(x_N) \leq 0,$$

$$\delta_{k,i} > 0 \Rightarrow G_i(x_k, u_k) \leq 0,$$

$$\delta_{k,i} \in [0, 1], k \in \mathbb{Z}_0^{N-1}, i \in \mathbb{Z}_0^{n_\delta}.$$

At each stage, taking decision i

- ▶ provides a reward, $w_i > 0$
- ▶ activates the associated constraint $G_i(x_k, u_k) \leq 0$

- ▶ How to represent the logical expression for a numerical solver?

⁴Ghezzi, Nurkanović, Weiss, Diehl, and Di Cairano (2026). *A Comparative Study of MINLP and MPVC Formulations for Solving Complex Nonlinear Decision-Making Problems in Aerospace Applications*. Optimal Control Applications and Methods

Computationally tractable logical expressions

$$\delta_i > 0 \quad \Rightarrow \quad G_i(z) \leq 0, \quad z = (x, u)$$

MINLP formulation

Consider decision as binary: $\delta_i \in \{0, 1\}$

Assuming a valid upper bound M_i ,

$$G_i(z) \leq M_i(1 - \delta_i).$$

- ▶ $\delta_i = 1$: $G_i(z) \leq 0$
- ▶ $\delta_i = 0$: $G_i(z) \leq M_i$

MPVC formulation

Consider decision as continuous: $\delta_i \in [0, 1]$

Introduce a vanishing constraint:

$$\delta_i G_i(z) \leq 0.$$

- ▶ $\delta_i > 0$: $G_i(z) \leq 0$
- ▶ $\delta_i = 0$: constraint vanishes

Computationally tractable logical expressions

$$\delta_i > 0 \Rightarrow G_i(z) \leq 0, \quad z = (x, u)$$

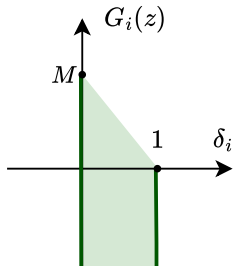
MINLP formulation

Consider decision as binary: $\delta_i \in \{0, 1\}$

Assuming a valid upper bound M_i ,

$$G_i(z) \leq M_i(1 - \delta_i).$$

- ▶ $\delta_i = 1$: $G_i(z) \leq 0$
- ▶ $\delta_i = 0$: $G_i(z) \leq M_i$



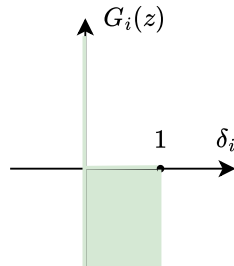
MPVC formulation

Consider decision as continuous: $\delta_i \in [0, 1]$

Introduce a vanishing constraint:

$$\delta_i G_i(z) \leq 0.$$

- ▶ $\delta_i > 0$: $G_i(z) \leq 0$
- ▶ $\delta_i = 0$: constraint vanishes



Objective $\min_{z, \delta} f(z) - \sum_i w_i \delta_i$
Locally optimal solution $\delta^* \in \{0, 1\}^{n_\delta}$

Solving the MPVC formulation⁵

- ▶ Vanishing constraints violate standard constraint qualifications
- ▶ Direct application of a standard NLP solver may fail

Homotopy relaxation

Replace

$$\delta_i G_i(z) \leq 0$$

by

$$\delta_i G_i(z) \leq \tau \mathbf{1}, \quad \tau > 0,$$

and solve a sequence of NLPs with $\tau \rightarrow 0$.
Each solution warm-starts the next NLP.

⁵Scholtes (2001). *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*.
SIAM Journal on Optimization

Solving the MPVC formulation⁵

- ▶ Vanishing constraints violate standard constraint qualifications
- ▶ Direct application of a standard NLP solver may fail

Homotopy relaxation

Replace

$$\delta_i G_i(z) \leq 0$$

by

$$\delta_i G_i(z) \leq \tau \mathbf{1}, \quad \tau > 0,$$

and solve a sequence of NLPs with $\tau \rightarrow 0$.
Each solution warm-starts the next NLP.

⇒ **Compare** the two formulations **on powered descent guidance with divert-feasible regions**

⁵Scholtes (2001). *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*.
SIAM Journal on Optimization

Powered descent guidance with divert-feasible regions

Objective:

- ▶ land with **minimum fuel** usage
- ▶ stay in **divert-feasible regions**

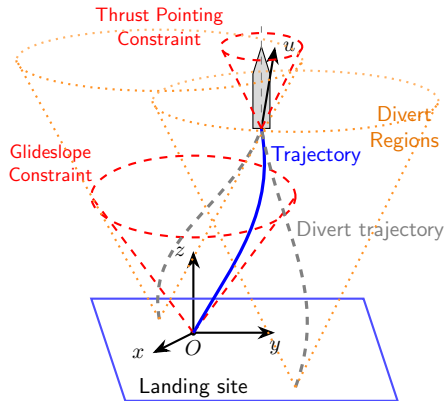
Nonlinear dynamics:

- ▶ Double point-mass with time-varying mass

$$\begin{cases} \dot{r}(t) = v(t), \\ \dot{v}(t) = \frac{u(t)}{m(t)} - \omega^\times \omega^\times r(t) - 2\omega^\times v(t) + g_{\text{mars}} \hat{e}_z, \\ \dot{m}(t) = -\frac{\|u(t)\|_2}{g_{\text{earth}} I_{\text{sp}}}, \end{cases}$$

Constraints:

- ▶ **conic stage constraints**
- ▶ **ring constraint on thrust norm**
 - ▶ $\rho_{\text{lb}} \leq \|u\|_2 \leq \rho_{\text{ub}}$
- ▶ **linear terminal constraints**
 - ▶ $r_N = v_N = 0, m_N \geq m_{\text{dry}}$



Divert-feasible regions

Divert-feasible region: a state from which a safe alternative landing maneuver is possible.

At each stage $k = 1, \dots, N$

- ▶ region i is represented by the polytope $\mathcal{D}_i = \{x_k \mid A_i x_k + b_i \leq 0\}$
- ▶ introduce the decision $\delta_{k,i} > 0 \Rightarrow A_i x_k + b_i \leq 0$
- ▶ reward time spent in divert-feasible regions: $\min -m_N - w \sum_{k=0}^{N-1} \sum_{i=1}^{n_\delta} \delta_{k,i}$

Divert-feasible regions

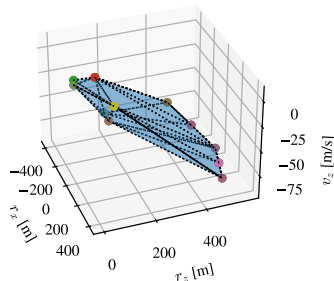
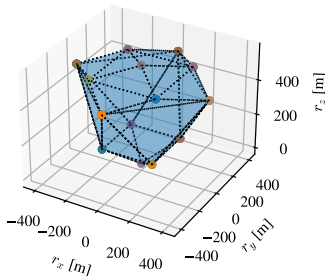
Divert-feasible region: a state from which a safe alternative landing maneuver is possible.

At each stage $k = 1, \dots, N$

- ▶ region i is represented by the polytope $\mathcal{D}_i = \{x_k \mid A_i x_k + b_i \leq 0\}$
- ▶ introduce the decision $\delta_{k,i} > 0 \Rightarrow A_i x_k + b_i \leq 0$
- ▶ reward time spent in divert-feasible regions: $\min -m_N - w \sum_{k=0}^{N-1} \sum_{i=1}^{n_\delta} \delta_{k,i}$

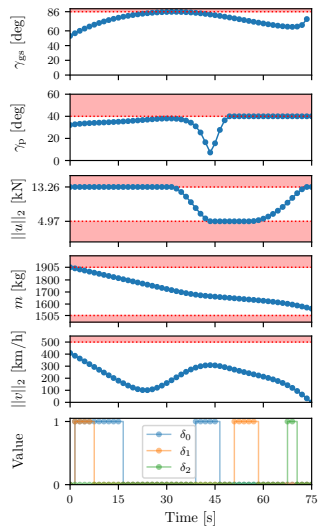
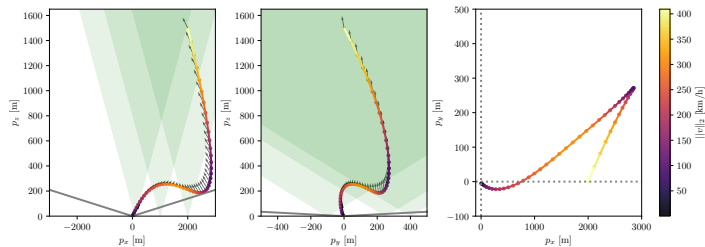
Realistic regions

- ▶ six-dimensional polytopes in position and velocity
- ▶ described by thousands halfspaces or hundreds vertices



Example of an optimal trajectory

Considering **simplified regions** – Three 3D polytopes on position (6 halfspaces)



Results

Simplified regions – Three 3D polytopes on position (6 halfspaces)

	Bonmin	S-B-MIQP	MPVC
Suboptimality	-5.0%	0%	46.4%
Final mass (kg)	1561.52	1561.79	1560.79
$\sum_i \sum_k \delta_{i,k}$	27	28	25
Runtime	10m31s	4m23s	7s

Results

Simplified regions – Three 3D polytopes on position (6 halfspaces)

	Bonmin	S-B-MIQP	MPVC
Suboptimality	-5.0%	0%	46.4%
Final mass (kg)	1561.52	1561.79	1560.79
$\sum_i \sum_k \delta_{i,k}$	27	28	25
Runtime	10m31s	4m23s	7s

Realistic regions – Three 6D polytopes on position and velocity (1672 halfspaces)

	Bonmin	S-B-MIQP	MPVC
Suboptimality	failed	0%	35.6%
Final mass (kg)	-	1531.40	1533.28
$\sum_i \sum_k \delta_{i,k}$	-	8	11.12
Runtime	-	1h0m	5h30m

Results

Simplified regions – Three 3D polytopes on position (6 halfspaces)

	Bonmin	S-B-MIQP	MPVC
Suboptimality	-5.0%	0%	46.4%
Final mass (kg)	1561.52	1561.79	1560.79
$\sum_i \sum_k \delta_{i,k}$	27	28	25
Runtime	10m31s	4m23s	7s

Realistic regions – Three 6D polytopes on position and velocity (1672 halfspaces)

	Bonmin	S-B-MIQP	MPVC
Suboptimality	failed	0%	35.6%
Final mass (kg)	-	1531.40	1533.28
$\sum_i \sum_k \delta_{i,k}$	-	8	11.12
Runtime	-	1h0m	5h30m

Take-away:

- ▶ MPVC is attractive when the logical structure is limited
- ▶ MINLP solved with S-B-MIQP is slower but finds better solution

Summary of Multi-Tree Methods

Three classes of multi-tree methods:

- ① Extended cutting plane method: linearize at MILP solutions
- ② Outer approximation: linearize at solutions of fixed NLP subproblems
- ③ Benders decomposition: aggregate OA cuts and eliminate continuous variables using KKT multipliers

Common properties:

- Convexity makes first-order cuts globally valid
- MILP master problems provide lower bounds and candidate integer assignments
- Feasible fixed NLPs provide upper bounds
- Feasibility subproblems exclude infeasible integer assignments
- OA is generally stronger than Benders, but has a larger master
- Worst-case behavior can still be exponential



Duran, M. A. and Grossmann, I. (1986).

An outer-approximation algorithm for a class of mixed-integer nonlinear programs.

Mathematical Programming, 36:307–339.



Geoffrion, A. M. (1972).

Generalized Benders decomposition.

Journal of Optimization Theory and Applications, 10(4):237–260.



Hijazi, H., Bonami, P., and Ouorou, A. (2010).

An outer-inner approximation for separable MINLPs.

Technical report, LIF, Faculté des Sciences de Luminy, Université de Marseille.



Westerlund, T. and Pettersson, F. (1995).

A cutting plane method for solving convex MINLP problems.

Computers & Chemical Engineering, 19:s131–s136.

- 
- Andersson, J. A. E., J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. “CasADi – A software framework for nonlinear optimization and optimal control”. In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: 10.1007/s12532-018-0139-4.
- 
- Ghezzi, A., A. Nurkanović, A. Weiss, M. Diehl, and S. Di Cairano. “A Comparative Study of MINLP and MPVC Formulations for Solving Complex Nonlinear Decision-Making Problems in Aerospace Applications”. In: *Optimal Control Applications and Methods* 47.1 (2026), pp. 4–23.
- 
- Ghezzi, A., W. Van Roy, S. Sager, and M. Diehl. “A Sequential Benders-based Mixed-Integer Quadratic Programming Algorithm and Its Implementation in the CAMINO Toolbox”. In: *Mathematical Programming Computation* (2026), pp. 1–70.
- 
- Scholtes, S. “Convergence properties of a regularization scheme for mathematical programs with complementarity constraints”. In: *SIAM Journal on Optimization* 11.4 (2001), pp. 918–936.