A Sequential Mixed-Integer Quadratic Programming Algorithm for Solving MINLP Arising in Optimal Control

Moritz Diehl[1]
joint work with **Andrea Ghezzi**[1], Sebastian Sager[2], Wim Van Roy[3]

[1]Systems Control and Optimization Laboratory,
Department of Microsystems Engineering and Department of Mathematics,
University of Freiburg, Germany

[2] Otto-von-Guericke Universität Magdeburg, Germany

[3] Atlas Copco, Antwerpen, and KU Leuven University, Belgium

Oberwolfach
August 13-18, 2023

universität freiburg

Mixed-integer nonlinear programming

This talk's aim: solve mixed-integer nonlinear program (MINLP) over polyhedral sets

Problem (Generic MINLP)

$$\mathcal{P}_{\mathrm{MINLP}}: \quad \begin{aligned} \min_{x \in X,\, y \in \mathbb{Z}^{n_y} \cap \bar{Y}} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & h(x,y) = 0. \end{aligned} \tag{1}$$

where

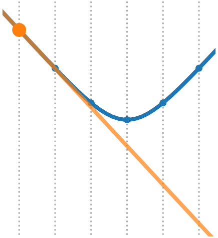
1. both $X \subset \mathbb{R}^{n_x}$ and $\bar{Y} \subset \mathbb{R}^{n_y}$ are convex polyhedral sets
2. functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_g}$, $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_h}$ are once continuously differentiable

Remark: **nonlinear equality constraints** $h(x,y) = 0$ arise naturally in **discretized optimal control problems** and cannot easily be addressed by existing MINLP solution methods. They always render the problem nonconvex.

Linearization

Throughout the talk, we often use the first order Taylor series of a nonlinear differentiable function $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_h}$ at a linearization point $(\bar{x}, \bar{y})$ that we denote by

$$h_{\mathrm{L}}(x, y; \bar{x}, \bar{y}) := h(\bar{x}, \bar{y}) + \frac{\partial h}{\partial x}(\bar{x}, \bar{y})(x - \bar{x}) + \frac{\partial h}{\partial y}(\bar{x}, \bar{y})(y - \bar{y})$$



Mixed Integer Optimal Control Problem with Binary Inputs $y(t)$

Formulated in outer convexified form, cf. [Sager 2005, 2009] and talk by Christian Kirches on Tuesday

$$\underset{z(\cdot),u(\cdot),y(\cdot),s(\cdot)}{\text{minimize}} \quad \int_0^T L\big(z,u,y,s\big)\,\mathrm{d}t + M\big(z(T)\big) \tag{2a}$$

$$\text{subject to} \quad z(0) = \bar{z}_0 \tag{2b}$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \sum_{i=1}^{n_b} y_i \cdot f_i\big(z,u,c\big), \quad \sum_{i=1}^{n_b} y_i(t) = 1, \tag{2c}$$

$$y_i(t) \in \{0,1\} \quad \text{for } \; i = 1,\dots n_b, \tag{2d}$$

$$-s + r_\mathrm{l} \le r\big(z,u,y,c\big) \le r_\mathrm{u} + s, \qquad \text{for } \; t \in \big[0,T\big] \tag{2e}$$

(+ additional combinatorial constraints such as min-up-times) (2f)

$z(t)$: states, $u(t)$: continuous controls, $s(t)$: slack variables, $y(t)$: binary controls

$c(t)$: time-varying parameters, f_i: system dynamics, $r_\mathrm{l} \le r \le r_\mathrm{u}$: path constraints

Discretize e.g. via Direct Multiple Shooting [Bock and Plitt 1984] to obtain MINLP in thousands of variables $x = (z,u,s) \in \mathbb{R}^{n_x}$ and $y \in \mathbb{Z}^{n_y} \cap [0,1]^{n_y}$. Discretization of nonlinear dynamics (2c) results in nonlinear equality constraints with long expressions → global solution out of reach.

Three step decomposition for fast approximate MIOCP solution

Combinatorial Integral Approximation (CIA) [Sager et al. 2011] [1]

1. Solve relaxed NLP with $y(t) \in [0,1]^{n_b}$ to obtain relaxed solution $y^*(t)$ for $t \in [0,T]$.
2. Solve minimum distance problem to find binary trajectory $y^{**}(\cdot)$ closest to $y^*(\cdot)$.
3. Solve an NLP where the binary controls are fixed to $y^{**}(t)$, to adjust $z(\cdot)$, $u(\cdot)$, and $s(\cdot)$.

Distance function in Step 2 is the "CIA distance" which measures the maximum of the integral of the difference of the trajectories. Fast tailored solvers for this special problem – an MILP – exist, e.g. in the python package `pycombina` [Bürger et al. 2019].

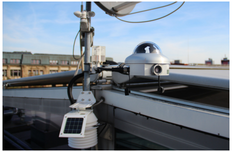
[1] S. Sager, M. Jung, and C. Kirches: Combinatorial Integral Approximation, Mathematical Methods of Operations Research, vol. 73, no. 3, pp. 363-380, 2011.

Nonlinear Model Predictive Control (MPC) for a solar thermal test plant

at Karlsruhe University of Applied Sciences, with two discrete actuators [PhD Adrian Bürger 2020]

Control cabinet, cold storage, ACM, hot storage, pumps (cellar)

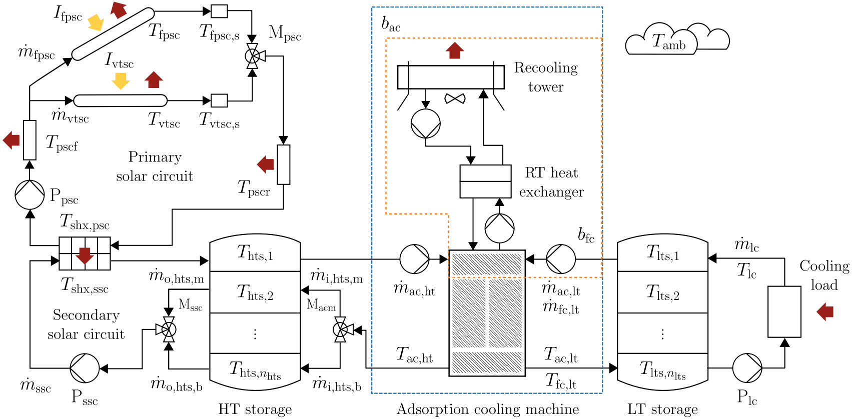
Recooling unit (roof)



Ambient sensors (roof)

Plate collectors (roof)

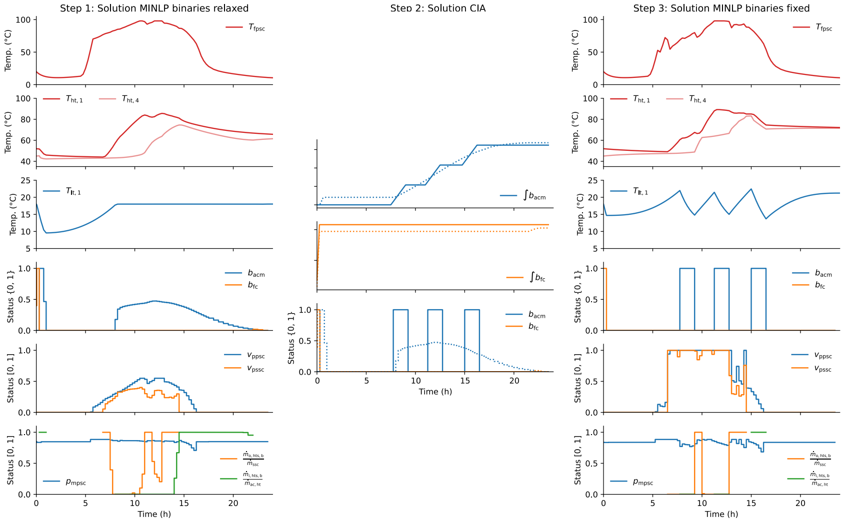
Vacuum tube collectors (roof)



Nonlinear switched system ODE model with $n_x = 20$, $n_b = 2$, $n_u = 5$, and $n_c = 4$,

differentiable in all arguments within the domain of interest

Numerical results: Three Step CIA Decomposition

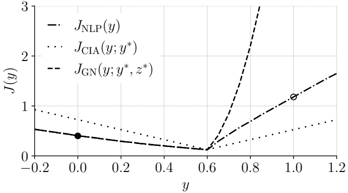


(25 CPU sec) (0.02 CPU sec) (18 CPU sec)

Alternative to CIA Decomposition: Gauss-Newton based MIQP

[Bürger et al., Automatica, 2023]

- Derive convex Gauss-Newton-type approximation of original MINLP from linearization at relaxed MINLP solution.
- Solution of resulting MIQP can yield improved integer solution in terms of objective and feasibility of the original MINLP.
- MIQP is equivalent to minimization of a distance function that is a first order accurate approximation of the true objective.



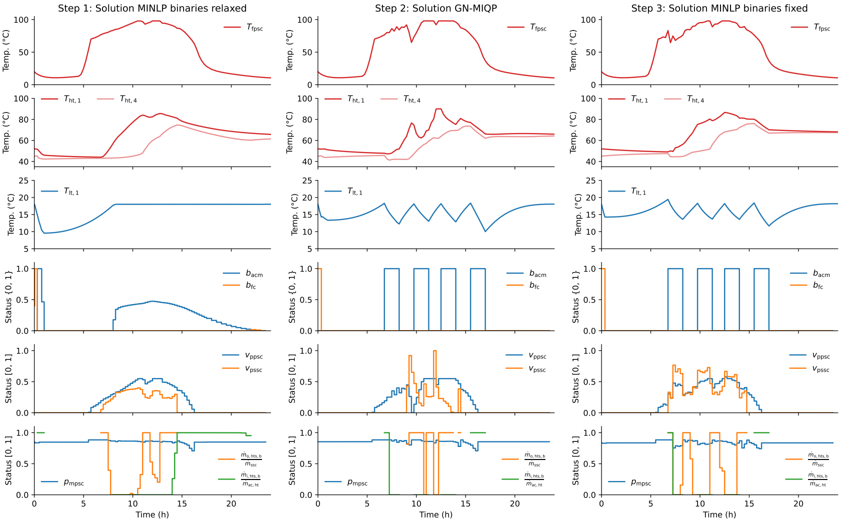
Original MINLP

$$\begin{aligned} \min_{x,y} \quad & \frac{1}{2} \|F_1(x,y)\|_2^2 + f_2(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & h(x,y) = 0 \\ & y \in \mathbb{Z}^{n_y} \end{aligned}$$

GN-MIQP from linearization at (x^*, y^*)

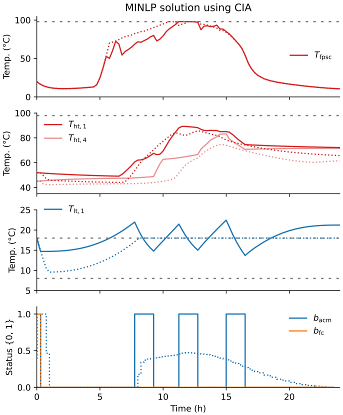
$$\begin{aligned} \min_{x,y} \quad & \frac{1}{2} \|F_{1,\mathrm{L}}(x,y;x^*,y^*)\|_2^2 + f_{2,\mathrm{L}}(x,y;x^*,y^*) \\ \text{s.t.} \quad & g_{\mathrm{L}}(x,y;x^*,y^*) \le 0 \\ & h_{\mathrm{L}}(x,y;x^*,y^*) = 0 \\ & y \in \mathbb{Z}^{n_y} \end{aligned}$$

Numerical results: Three Step GN-MIQP Decomposition

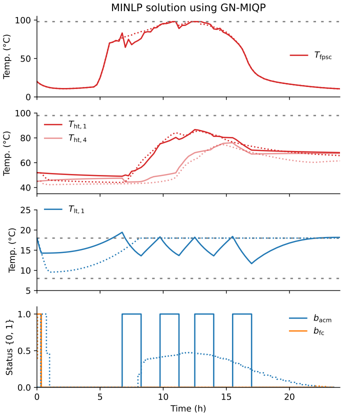


(25 CPU sec) (130 CPU sec) (15 CPU sec)

Comparison of CIA and GN-MIQP Solution



(43 CPU sec)



(170 CPU sec)

GN-MIQP delivers significant feasibility improvements, at the expense of increased computational cost.

Could one apply GN-MIQP sequentially for further improvements?

Guidelines of Sequential MIQP algorithm from this talk

- do not aim at global optimality in general...
- ...but ensure algorithm finds global solution if accidentally applied to a convex MINLP
- use convex quadratic model based on Taylor expansion at incumbent solution
- sequentially solve mixed integer quadratic programs (MIQP), similar to Sequential Quadratic Programming (SQP)
- use Gauss-Newton (GN) Hessian or generalizations (GGN, SCQP, cf. [Messerer et al. 2021])
- avoid revisiting previously visited suboptimal points
- use first order information from previous points but modify it to account for nonconvexities
- (terminate only if lower and upper bound coincide)

Note: do not aim for global optimality as in ANTIGONE [Misener, Floudas], BARON [Sahinidis, Khajavirad, Tawarmalani, ...], or COUENNE [Belotti, Lee, Liberti, Margot, Wächter].

But algorithm is related to Quadratic Outer Approximation (QOA) variants from [Fletcher and Leyffer , 1994], and from [Kronquist, Bernal and Grossmann, 2020].

Utopian Convexity Assumption

Recall:

Problem (Generic MINLP)

$$\mathcal{P}_{\mathrm{MINLP}}: \quad \begin{aligned} \min_{x \in X,\, y \in \mathbb{Z}^{n_y} \cap \bar{Y}} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0, \\ & h(x,y) = 0. \end{aligned}$$

We might for some test cases assume a "convex MINLP" as follows:

Assumption (convexity, for theoretical analysis only)

1. *function h is affine and functions f and g are convex on $X \times \bar{Y}$*
2. *integer set $Y = \mathbb{Z}^{n_y} \cap \bar{Y}$ is finite*

Ideally, algorithm works independently of this assumption, but is guaranteed to find global solution if applied to a "convex MINLP" (without knowing it)

Auxiliary Problem with fixed integers

The presented algorithm will iterate between

- high level ("master") problems which optimize over both continuous and integer variables;
- **an auxiliary lower level NLP** with fixed integers $y \in \bar{Y}$ with value $J(y)$ as follows.

Problem (Auxiliary nonlinear program (NLP))

$$\mathcal{P}_{\mathrm{NLP}}: \quad \begin{aligned} J(y) := \min_{x \in X} \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0, \\ & h(x,y) = 0. \end{aligned} \tag{5}$$

To focus on the main algorithmic ideas, we make an optimistic assumption.

Assumption (feasibility of all NLPs, for clarity of exposition)

For any $y \in \bar{Y}$, problem (5) is feasible, and admits a minimizer with finite objective value.

(can be achieved by the use of penalized slack variables for potentially infeasible constraints)

Two High Level Mixed Integer Problems

Using function J, the MINLP (1) can compactly be written as

Problem (Generic MINLP in compact notation)

$$\min_{y \in Y} J(y).$$

In the algorithm we solve two high level mixed integer problems based on different approximations of the function J:

1. J_{QP}, a quadratic programming based approximation of J (an MIQP)
2. J_{LB}, a lower bound function that collects first order information from all other visited points so far (an MILP)

Quadratic programming approximation

Definition

The quadratic programming approximation $J_{\mathrm{QP}} \approx J$ at linearization point $(\bar{x}, \bar{y})$ is defined as

$$
\begin{aligned}
J_{\mathrm{QP}}(y; \bar{x}, \bar{y}, B) := \min_{x \in X} \quad & f_{\mathrm{L}}(x, y; \bar{x}, \bar{y}) + \frac{1}{2} \begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix}^{\top} B \begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix} \\
\text{s.t.} \quad & g_{\mathrm{L}}(x, y; \bar{x}, \bar{y}) \leq 0, \\
& h_{\mathrm{L}}(x, y; \bar{x}, \bar{y}) = 0,
\end{aligned} \tag{6}
$$

where B is a positive semidefinite Hessian approximation.

Remark: If $B = 0$ and under the convexity assumption, the piecewise linear convex function J_{QP} is an underestimator of J, i.e., $J_{\mathrm{QP}}(y; \bar{x}, \bar{y}, 0) \leq J(y)$ for all $y \in \bar{Y}$ (as in outer approximation). For $B \succ 0$ no such guarantee exists, though we expect the quadratic approximation to often be better than the linear one.

A favourable choice for least squares objectives $f(x, y) = \|F(x, y)\|_2^2$ is the Gauss-Newton Hessian $B_{\mathrm{GN}} = 2\nabla F(\bar{x}, \bar{y}) \nabla F(\bar{x}, \bar{y})^{\top}$ which makes the QP objective equal to $\|F_{\mathrm{L}}(x, y; \bar{x}, \bar{y})\|_2^2$.

MIQP Formulation with level constraints to exclude suboptimal points

After having evaluated the auxiliary NLP at k integer points, giving $J(y_1), \ldots, J(y_k)$ along with their (sub)gradients (as in GBD), we linearize at the best point so far, $y_{b(k)}$ ("incumbent solution"). We also impose level constraints that exclude all non-optimal points y_i with $i \in \mathbb{I}_k := \{1, 2, \ldots, k\} \setminus \{b(k)\}$.

Problem (Level constrained "trust-region" MIQP)

$$\mathcal{P}_{\mathrm{TR-MIQP}}: \quad \begin{aligned} \min_{x \in X, y \in Y} \quad & f_{\mathrm{L}}(x, y; x_{b(k)}, y_{b(k)}) + \frac{1}{2} \begin{pmatrix} x - x_{b(k)} \\ y - y_{b(k)} \end{pmatrix}^\top B \begin{pmatrix} x - x_{b(k)} \\ y - y_{b(k)} \end{pmatrix} \\ \text{s.t.} \quad & g_{\mathrm{L}}(x, y; x_{b(k)}, y_{b(k)}) \leq 0, \\ & h_{\mathrm{L}}(x, y; x_{b(k)}, y_{b(k)}) = 0, \\ & J(y_i) + \nabla J(y_i)^\top (y - y_i) \leq J(y_{b(k)}), \; i \in \mathbb{I}_k. \end{aligned} \tag{7}$$

Note: level constraints only in integer space, as in GBD. They form a polyhedral "Benders Trust Region" $\mathbb{B}_k$ that excludes all points y for which we have a certificate (in the convex case) that they are worse than the incumbent solution. OA cutting planes are not easily applicable here due to the presence of nonlinear equality constraints.

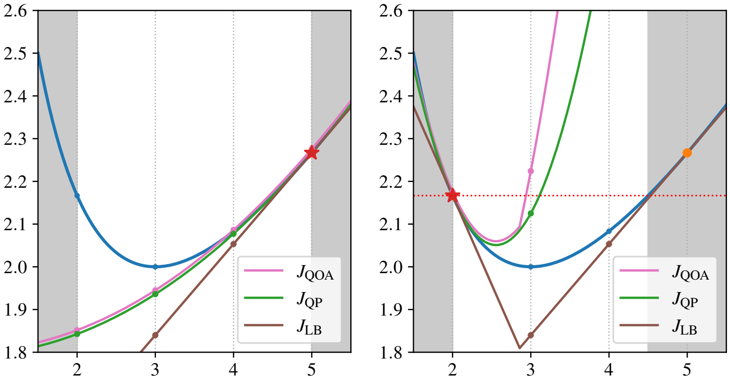
Comparison with Quadratic Outer Approximation [Fletcher and Leyffer 1994]

R. Fletcher, S. Leyffer / Mathematical Programming 66 (1994) 327–349

$$
\begin{aligned}
\min_{x,y,\eta} \quad & \eta + \tfrac{1}{2} \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix}^{\mathrm{T}} [\nabla^2 \mathscr{L}^i] \begin{pmatrix} x - x^i \\ y - y^i \end{pmatrix}, \\
\text{subject to} \quad & \eta < \mathrm{UBD}, \\
& \left.\begin{aligned} & \eta \geqslant f^j + (\nabla f^j)^{\mathrm{T}} \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix}, \\ & 0 \geqslant g^j + [\nabla g^j]^{\mathrm{T}} \begin{pmatrix} x - x^j \\ y - y^j \end{pmatrix}, \end{aligned}\right\} \quad \forall j \in T^i, \\
& 0 \geqslant g^k + [\nabla g^k]^{\mathrm{T}} \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix}, \quad \forall k \in S^i, \\
& x \in X, \quad y \in Y \text{ integer},
\end{aligned}
$$

Comparison with Quadratic Outer Approximation [Fletcher and Leyffer 1994]

Test problem $\min_{y \in \mathbb{Z} \cap [2,5]} \underbrace{3/y + y/3}_{=J(y)}$ solved with $\underbrace{\text{QOA}}_{\to J_{\text{QOA}}}$ and $\underbrace{\text{S-MIQP}}_{\to J_{\text{QP}}}$, started at $y_0 = 5$ (left)



At iteration two (right), QOA gets stuck while S-MIQP does not. In QOA, linear and quadratic models are added, making the MIQP too pessimistic.

Comparison with Quadratic Model by [Kronquist, Bernal, Grossmann 2020]

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{y},\mu}\ & \nabla_{\mathbf{x},\mathbf{y}}\mathcal{L}(\bar{\mathbf{x}},\bar{\mathbf{y}},\bar{\lambda})^T \begin{bmatrix}\Delta\mathbf{x}\\ \Delta\mathbf{y}\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\Delta\mathbf{x}\\ \Delta\mathbf{y}\end{bmatrix}^T \nabla^2_{\mathbf{x},\mathbf{y}}\mathcal{L}(\bar{\mathbf{x}},\bar{\mathbf{y}},\bar{\lambda}) \begin{bmatrix}\Delta\mathbf{x}\\ \Delta\mathbf{y}\end{bmatrix}\\
\text{s.t.}\ & \mu \le \hat{f}_k^*\\
& f(\mathbf{x}^\mathbf{i},\mathbf{y}^\mathbf{i}) + \nabla f(\mathbf{x}^\mathbf{i},\mathbf{y}^\mathbf{i})^T \begin{bmatrix}\mathbf{x}-\mathbf{x}^\mathbf{i}\\ \mathbf{y}-\mathbf{y}^\mathbf{i}\end{bmatrix} \le \mu \quad \forall i = 1,\ldots,k\\
& g_j(\mathbf{x}^\mathbf{i},\mathbf{y}^\mathbf{i}) + \nabla g_j(\mathbf{x}^\mathbf{i},\mathbf{y}^\mathbf{i})^T \begin{bmatrix}\mathbf{x}-\mathbf{x}^\mathbf{i}\\ \mathbf{y}-\mathbf{y}^\mathbf{i}\end{bmatrix} \le 0 \quad \forall i = 1,\ldots k, \forall j \in \mathcal{I}_i,\\
& \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} \le \mathbf{b},\\
& \mathbf{x} \in \mathbb{R}^n,\ \mathbf{y} \in \mathbb{Z}^m, \mu \in \mathbb{R},
\end{aligned}
\tag{QOA-master}
$$

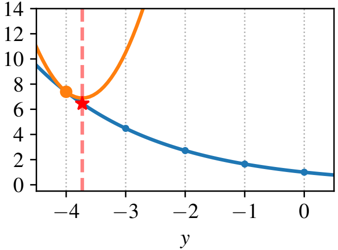
In Kronquist's QOA-Master, no "1994 QOA double counting" thanks to region constraint.
Differences to today's S-MIQP:

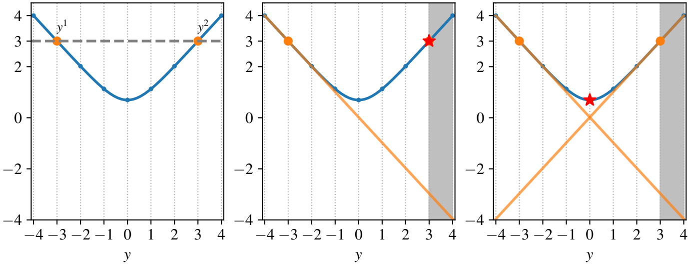
(a) gradient of the Lagrangian instead of objective gradient (I believe the objective is better)
(b) exact Hessian instead of Gauss-Newton Hessian (good choice if PSD)
(c) use of OA instead of GBD (better, but only if possible)
(d) reduced level region with $\hat{f}_k^* < J(\bar{y})$ to enforce progress (very good idea)

Lower Bound Function J_{LB}

If the proposed S-MIQP algorithm would only entail the MIQP problem

- no guarantees to find a global minimizer (even under convexity assumption, left plot)
- cycling if two integer solutions have the same objective (middle plot)





Idea: use convex piecewise linear underestimator $J_{\mathrm{LB}} \leq J$ (visualized right) and solve MILP.

Definition (Lower Bound Function J_{LB})

$$J_{\mathrm{LB}}(y; \mathbb{I}_k, \mathcal{D}_k) := \max_{i \in \mathbb{I}_k} J(y_i) + \nabla J(y_i)^\top (y - y_i). \tag{8}$$

Lower bound MILP

If MIQP solution delivers no new point or a point with equal objective, solve an MILP to

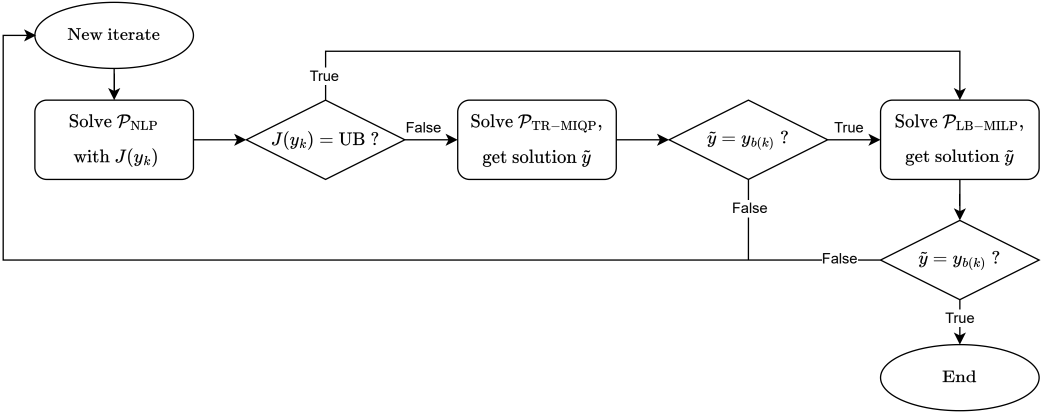
- generate a lower bound LB
- check if termination is possible, i.e, if $J(y_{b(k)}) = \mathrm{LB}$
- deliver a new point y_k

Problem (Lower bound mixed integer linear program $\mathcal{P}_{\mathrm{LB-MILP}}$)

$$
\mathcal{P}_{\mathrm{LB-MILP}}: \quad
\begin{aligned}
\mathrm{LB}_k := \min_{\eta\in\mathbb{R},\, x\in X,\, y\in Y} \quad & \eta \\
\text{s.t.} \quad & \eta \geq f_{\mathrm{L}}(x, y; x_{b(k)}, y_{b(k)}), \\
& 0 \geq g_{\mathrm{L}}(x, y; x_{b(k)}, y_{b(k)}), \\
& 0 = h_{\mathrm{L}}(x, y; x_{b(k)}, y_{b(k)}), \\
& \eta \geq J(y_i) + \nabla J(y_i)^\top (y - y_i), \quad i \in \mathbb{I}_k.
\end{aligned}
\tag{9}
$$

Note: mix of Outer Approximation (OA) at best point and Generalized Benders Decomposition (GBD) for all other points. This way, linearized equality constraints appear only once.

Full S-MIQP algorithm



store all NLP solution tuples $(k, x_k, y_k, J(y_k), \nabla J(y_k))$ in a growing data structure $\mathcal{D}_k$

Init.: $y_0 \in Y$, $\mathbb{B}_0 \leftarrow Y$, $\mathrm{UB} = +\infty$, $\mathrm{LB} = -\infty$
for $k = 0, 1, 2, \ldots$ **do**:
  LBFlag $\leftarrow$ `False`
  Solve $\mathcal{P}_{\mathrm{NLP}}$ with $J(y_k)$, store solution $(k, x_k, y_k, J(y_k), \nabla J(y_k))$ in $\mathcal{D}_k$
  if $J(y_k) < \mathrm{UB}$ **then**: ▷ `Update the best solution`
    $\mathrm{UB} \leftarrow J(y_k)$, $b(k) \leftarrow k$, $\mathbb{I}^{\mathrm{b}} \leftarrow \{b(k)\}$ ▷ `Create set of all optimal points so far`
  else if $J(y_k) = \mathrm{UB}$ **then**:
    LBFlag $\leftarrow$ `True`, $\mathbb{I}^{\mathrm{b}} \leftarrow \mathbb{I}^{\mathrm{b}} \cup \{k\}$
  end if
  Modify gradients in $\mathcal{D}_k$, obtain $\tilde{\mathcal{D}}_k$
  if not LBFlag **then**:
    Compute Benders trust region $\mathbb{B}_k$ based on $\tilde{\mathcal{D}}_k$
    Solve $\mathcal{P}_{\mathrm{TR-MIQP}}$ with $J_{\mathrm{QP}}(y; x_{b(k)}, y_{b(k)}, B)$ and $\mathbb{B}_k$, store solution $\tilde{y}$
    if $\tilde{y} \in \{y_i | i \in \mathbb{I}^{\mathrm{b}}\}$ **then**:
      LBFlag $\leftarrow$ `True`
    end if
  end if
  if LBFlag **then**: ▷ `Tie-breaking mechanism`
    Solve $\mathcal{P}_{\mathrm{LB-MILP}}$ with $J_{\mathrm{LB}}(y; \mathbb{I}_k, \tilde{\mathcal{D}}_k)$, obtain solution $\tilde{y}$, and $\mathrm{LB} \leftarrow V_{\mathrm{MILP}}$
    if $\tilde{y} \in \{y_i \mid i \in \mathbb{I}^{\mathrm{b}}\}$ **then**:
      return $\tilde{x}$, $\tilde{y}$, UB, LB ▷ `Algorithm termination`
    end if
  end if
  $y^{k+1} \leftarrow \tilde{y}$, $b(k+1) \leftarrow b(k)$
end for

Algorithm properties

Lemma (On the termination condition)

Under the convexity assumption, $\mathcal{P}_{\mathrm{LB-MILP}}$ is an underestimator of problem (1). *Thus, if its solution equals one of the best points found so far, a global minimizer is found.*

Lemma (On the computation of the Benders trust region)

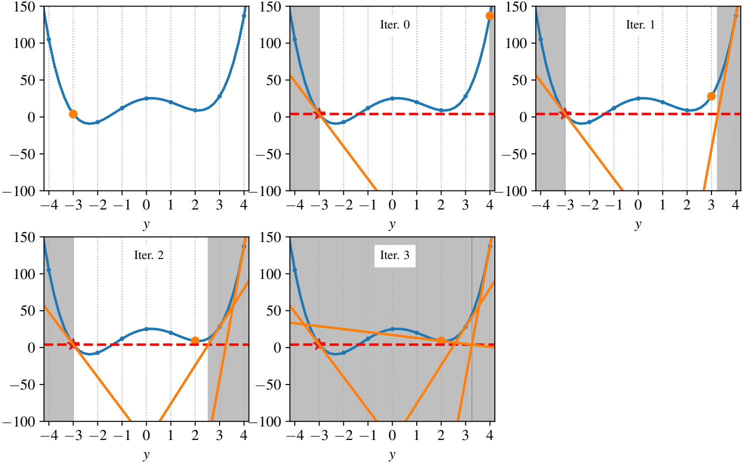
Under the convexity assumption, the Benders trust region
$\mathbb{B}_k = \{y \in \mathbb{R}^{n_y} \mid J_{\mathrm{LB}}(y; \mathbb{I}_k, \mathcal{D}_k) \leq J(y_{b(k)})\}$ *is non-empty and contains the best point* $y_{b(k)}$.

Theorem (On algorithm convergence)

Under the convexity assumption, the proposed algorithm stops at the global optimal solution of problem (1) *within a finite number of iterations.*

If we drop the convexity assumption, none of the above is guaranteed. In particular, we have to devise strategies to modify the lower bound function J_{LB} and the trust region $\mathbb{B}_k$ such that at least $y_{b(k)} \in \mathbb{B}_k$ holds.

Aim: J_{LB} shall still serve as an approximate underestimator of J. Not possible without modifications, as the example below (with $B = 0$) shows at iteration 3, where $\mathbb{B}_k$ becomes empty and excludes even the incumbent solution.



Nonconvex Modification 1: Gradient Correction

At the k-th iteration of the algorithm, each level constraint is given by

$$J(y_i) + \nabla J(y_i)^\top (y - y_i) \leq J(y_{b(k)}),$$

But what shall we do if the best point $y_{b(k)}$ does not satisfy it for some $i \neq b(k)$?

Idea 1: replace $\nabla J(y_i)$ by "corrected gradient" (with weighting matrix $W \succ 0$):

$$g^{\text{corr}}_{(i,k)} := \underset{g \in \mathbb{G}_{(i,k)}}{\arg\min} \quad \frac{1}{2} \|g - \nabla J(y_i)\|_W^2 \tag{10}$$

which is in the set of "admissible" gradients $\mathbb{G}_{(i,k)} := \{\tilde{g} \mid J(y_i) + \tilde{g}^\top (y_{b(k)} - y_i) \leq J(y_{b(k)})\}$

Lemma

Under the convexity assumption, $g^{\text{corr}}_{(i,k)} = \nabla J(y_i)$ *for all* $i \in \mathbb{I}_k$ *in each iteration* k.

Thus, gradient correction does not impair the nice algorithm properties in the convex case.

Nonconvex Modification 2: Gradient Amplification

Gradient correction alone might lead to very small regions due to the minimal correction. Can we enlarge the "Benders trust region" $\mathbb{B}_k$ without impairing the convergence guarantees in the convex case?

Idea 2: introduce a constant value $\rho \geq 1$ to amplify all gradients as

$$g_{(i,k,\rho)}^{\mathrm{ampl}} := \rho g_{(i,k)}^{\mathrm{corr}}$$

- ρ is a parameter of the algorithm
- ρ is chosen offline and kept fixed at runtime
- it is possible to devise more elaborate strategies to choose ρ

Denote by $\mathcal{D}_k^{\mathrm{ampl}}$ the dataset with modified gradient entries $(i, x_i, y_i, J(y_i), g_{(i,k,\rho)}^{\mathrm{ampl}})$.
The final nonconvex version of the algorithm sets $\tilde{\mathcal{D}}_k := \mathcal{D}_k^{\mathrm{ampl}}$.

Properties of the final (nonconvex) S-MIQP Algorithm

Theorem (Well defined iterates)

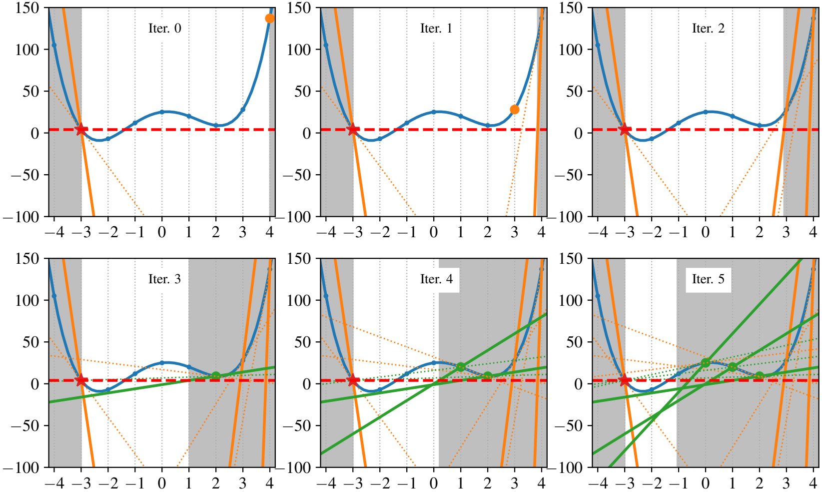
The final nonconvex S-MIQP algorithm – enhanced with the gradient correction and amplification procedure – will always have nonempty regions $\mathbb{B}_k$ that contain the incumbent solution ($y_{b(k)} \in \mathbb{B}_k$). The algorithm stops if neither MIQP nor MILP propose a novel point.

Theorem (Nonconvex S-MIQP Behaviour for Convex MINLP)

If the final nonconvex S-MIQP algorithm is applied to a MINLP (1) that accidentally satisfies the convexity assumption, it stops at the global optimal solution $(x_{b(k)}, y_{b(k)})$ within a finite number of iterations.

Sketch of the proof: gradient correction is inactive in the nonconvex case, and gradient amplification only decreases the lower bound function J_{LB} inside the unmodified $\mathbb{B}_k$ so it increases the region $\mathbb{B}_k$ and still serves as a lower bound to J in this region. The lower bound becomes tight once the optimal point is found. The algorithm stops if an old point is proposed by MIQP and MILP, and the number of integer points is finite, so it can only have a finite number of iterations.

Gradient Modification Visualization Example 1



- $\rho = 5,\ B = 0$

  ... original gradients

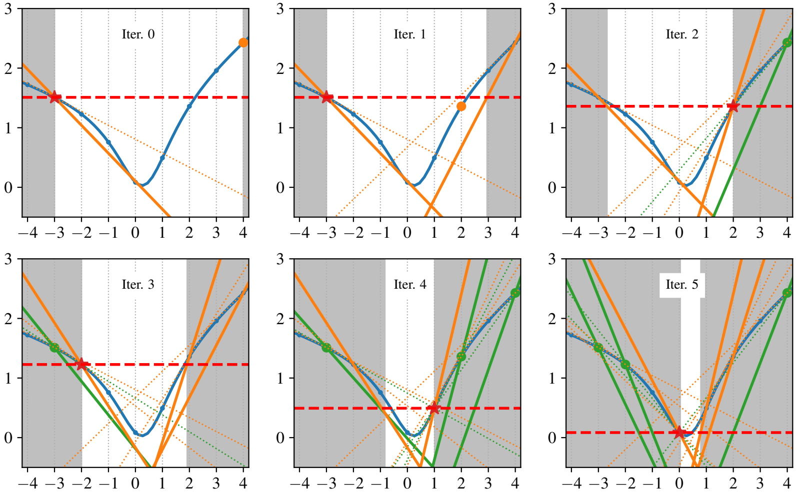
  — with amplification

  ... with correction only

  — with correction and amplification

- (Algorithm stopped early for visualization)

Gradient Modification Visualization Example 2



- $\rho = 2, \ B = 0$

original gradients
with amplification
with correction only
with correction and amplification

Consider a mixed-integer optimal control problem for a nonlinear unstable system

- with one state $x \in \mathbb{R}$ and one binary control $u \in \{0,1\}$
- the dynamics given by $\dot{x} = x^3 - u$
- transformed to discrete time using a one RK4 step integrator with $h = 0.05\,\mathrm{s}$
- introduce a minimum dwell time constraint of $0.1\,\mathrm{s}$

Aim is to track a reference $x_{\mathrm{ref}} = 0.7$, starting from $x_0 = 0.8$ on a horizon of length $N = 30$.

We obtain the following MINLP:

$$\begin{aligned}\min_{\mathbf{x},\mathbf{u}} \quad & \frac{1}{2}\sum_{k=0}^{N}(x_k - x_{\mathrm{ref}})^2 \\ \text{s.t.} \quad & x(0) = x_0, \\ & x_{k+1} = f(x_k, u_k), \quad k = 0,\dots,N-1, \\ & u_k \in \mathbf{U} \qquad\qquad\quad k = 0,\dots,N-1,\end{aligned} \tag{11}$$

where $\mathbf{U} := \{u \in \{0,1\}^N \mid u_k \geq u_{k-1} - u_{k-2}$ **and** $u_k \geq u_{k-1} - u_{k-3},\ k = 0,\dots,N-1\}$.

The nonconvex S-MIQP algorithm (with $\rho = 1.5$ and $B = B_{\mathrm{GN}}$) started non-smartly at $\mathbf{u_0} = \mathbf{0}$ found the minimum objective value $2.07 \cdot 10^{-2}$ after 6 iterations but took 50 more to improve the lower bound (which accidentally is the global minimum, that can be found by BnB here).

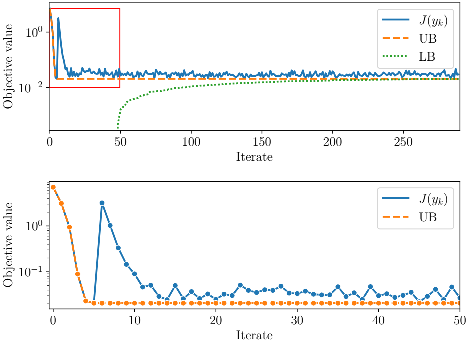


Figure: Algorithm's iterations

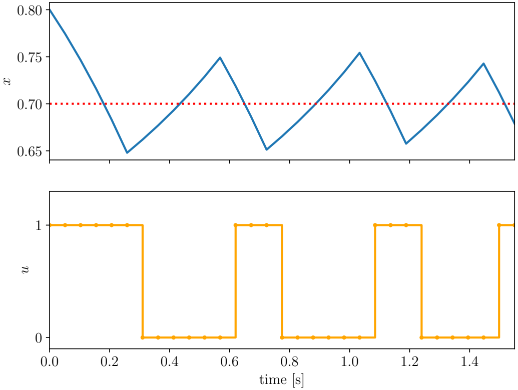


Figure: Optimal system trajectory

Conclusions

This talk presented an early-stage, untested, unbenchmarked algorithm for nonconvex MINLP that is based on sequential MIQP solutions that we call "S-MIQP".

- S-MIQP is a heuristic for nonconvex problems but finds global solution for convex MINLP.
- S-MIQP is applicable to MINLP with nonlinear equality constraints.
- MIQP and MILP contain only one linearization of the nonlinear equality constraints.
- Level constraints in MIQP ensure that past suboptimal points are avoided.
- Gauss-Newton Hessian leads to a very good QP approximation for least squares problems.
- Gradient correction and gradient amplification can deal well with (mild) nonconvexities.
- Many future enhancements are possible:
  - rigorously formalize and prove theoretical properties of algorithm
  - introduce strict level decrease condition as in [Kronquist et al. 2020]
  - incorporate more information from past visited points (e.g. some "OA cutting planes")
  - extend algorithm to deal with infeasible auxiliary problems
  - estimate nonconvexity, amplifiy gradients with adaptive factors $\rho_{(i,k)}$
  - exploit similarities between subsequent MIQP and MILP problems to speed up their solution
  - perform extensive testing and benchmarking...

Thank you