ARTICLE SUBMITTED TO OPTIMIZATION METHODS AND SOFTWARE

Combinatorial Integral Approximation Decompositions for Mixed-Integer Optimal Control

C. Zeile[a], T. Weber[a] and S. Sager[a]

[a]Faculty of Mathematics, Otto-von-Guericke-University Magdeburg, Germany

ARTICLE HISTORY
Compiled December 13, 2019

ABSTRACT
Solving mixed-integer nonlinear programs (MINLPs) is hard in theory and practice. Decomposing the nonlinear and the integer part seems promising from a computational point of view. In general, however, no bounds on the objective value gap can be guaranteed and iterative procedures with potentially many subproblems are necessary. The situation is different for mixed-integer optimal control problems with binary choices that switch over time. Here, a priori bounds were derived for a decomposition into one continuous nonlinear control problem and one mixed-integer linear program, the *combinatorial integral approximation* (CIA) problem.

We generalize and extend the decomposition idea. First, we derive different decompositions and analyze the implied a priori bounds. Second, we propose several strategies to generate promising candidate solutions for the binary control functions in the original problem. We present the extensions for problems constrained by ordinary differential equations. They are transferable in a straightforward way though to recently suggested variants for certain partial differential equations, for algebraic equations, for additional combinatorial constraints, and for discrete time problems.

All algorithms and subproblems were implemented in `AMPL` for proof of concept. Numerical results show the improvement compared to standard CIA decomposition with respect to objective function value and compared to general purpose MINLP solvers with respect to runtime.

KEYWORDS
Optimal control, Switched Dynamic Systems, Mixed-integer Nonlinear Programming, Mixed-integer Linear Programming, Ordinary Differential Equations, Approximation methods, heuristics

AMS CLASSIFICATION
49K15 and 34H05 and 93C65 and 90C59

1. Introduction

The goal of optimal control is to find control functions and state trajectories that are feasible and optimal. State trajectories are solutions of systems of differential equations for given control functions and boundary conditions. Feasibility refers to constraints on control functions and differential states, optimality refers to an objective functional of controls and states. Mixed-integer optimal control problems (MIOCP) do have additional integrality constraints on some of the control functions. Hence, the differential equations of the underlying system depend on the value of integer

CONTACT C. Zeile, Email: clemens.zeile@ovgu.de

control functions, or equivalently, one can switch instantly between several differential equations, [7, 9, 16, 43]. This problem class is ubiquitous in various application areas, e.g., water, gas, traffic, and supply chain networks [6, 8, 11, 18–21, 31], distributed autonomous systems [1], processes in chemical engineering that involve valves [28, 44], or the choice of gears in automotive control [13, 30]. We chose problems from a web-based MIOCP benchmark collection [37] to evaluate our algorithms.

Several families of methods have been proposed for solving MIOCPs. This comprehends *indirect* or *first-optimize then-discretize* approaches [15], Dynamic Programming or Hamilton-Jacobi-Bellman [24], moment relaxations [39], switching time optimization [13, 32, 34, 46, 47], and *direct* or *first-discretize then-optimize* approaches [7, 12, 35]. Surveys, references, and comparisons can be found, e.g., in [15, 39, 41, 49].

Our approach is based on previous ideas [5, 27, 35, 40] to decompose the MIOCP into a relaxed continuous optimal control problem (OCP) and a mixed-integer linear program (MILP). In contrast to general MINLP decompositions (see [45] for recent results on error bounds for rounding approaches and [2] for a MINLP survey), the particular setting with dependent (states) and independent (controls) variables allows the derivation of a priori bounds, [29, 36]. As specified in Section 4, the difference between state trajectories $\mathbf{x}(\cdot)$ and $\mathbf{y}(\cdot)$ that are the unique solutions of the initial value problems for given control functions ω and α, respectively, is bounded for a constant C and all $t \in \mathcal{T}$ by

$$||\mathbf{x}(t) - \mathbf{y}(t)|| \leq C \max_{t \in \mathcal{T}} \left|\left| \int_{t_0}^{t} \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right|\right|. \tag{1}$$

This motivates to solve the relaxed problem (OCP) to obtain $\alpha(\cdot)$, to calculate an integer control function ω in a second step such that the maximum on the right hand side of (1) is as small as possible, and to obtain $y(\cdot)$ by simulation. Continuity of constraint and objective functions implies "good" behavior with respect to feasibility and objective function value in the original MIOCP. The combinatorial integral approximation (CIA) problem in the second step can be formulated as a MILP [40] and can be solved either with generic MILP methods, with specifically tailored Branch and Bound methods [27], and in some cases even in linear time with the Sum Up Rounding (SUR) method [35].

The idea of CIA decomposition has also been used in the context of hyperbolic partial differential equations [22, 23], differential-algebraic equation systems [15], with additional combinatorial constraints [20, 40], and based on discrete time formulations [20, 25]. In this publication we generalize all of these approaches by presenting alternative ways to calculate ω based on α, using different MILPs. We formulate them for the case of initial value problems with ordinary differential equations, although they are applicable to all mentioned variants in a straightforward way.

Our results are also independent of the method that is applied to solve the relaxed problem (OCP). For the numerical results, we are going to use a *first-discretize then-optimize* approach with Radau collocation. *First-discretize* refers a) to approximating the control functions with parameterized basis functions, such as finitely many piecewise constant functions, and b) to relaxing path and control constraints from the domain of a time horizon to finitely many time points. The *then-optimize* refers to solving the resulting finite dimensional optimization problem numerically to optimality. An overview of direct methods for continuous optimal control problems can be found in, e.g., [3, 14]. For comparison, we also apply this approach directly to the MIOCP and solve the resulting mixed-integer nonlinear program (MINLP) with the

general purpose solver `Bonmin`.

For a general problem class of MIOCPs the integer gap depends linearly on the control discretization [38]. For many applications this control discretization is not fixed, but can be refined. Thus, the integer gap can be driven to zero (usually at the expense of frequent switching). For practical reasons good or even optimal solutions for a fixed control discretization are of interest, which is our main focus.

Contributions. We derive different versions of the CIA problem, leading to multiple MILPs. Noting that the computational effort to solve one MILP is usually small compared to the solution of the relaxed nonlinear problem and even smaller compared to the deterministic solution of the original MIOCP, we propose to solve several approximation problems. These solutions are candidate solutions themselves, and can be recombined into new switching sequences. We derive theoretical a priori bounds for them, discuss computational (dis)advantages, and show numerically the improvement to existing decomposition approaches with respect to objective function value and to general purpose MINLP solvers with respect to runtime.

The *outline* of the article is as follows. We define the considered MIOCP class and propose a general decomposition framework in Section 2. The algorithm consists of several MILP formulations, which are discussed in Section 3. We discuss theoretical properties in Section 4. In Section 5 we look at strategies that combine and improve existing binary control functions. In Section 6 we provide and discuss numerical results for the MIOCP benchmark library [37].

2. Problem class, definitions, and main algorithm

We denote the considered time horizon by $\mathcal{T} := [t_0, t_\mathrm{f}] \subset \mathbb{R}$ and write "for a.e. $t \in \mathcal{T}$" for all $t \in \mathcal{T}$ except on a set of measure zero. $C^k(\mathcal{X}_1, \mathcal{X}_2)$ is the space of k-times continuously differentiable functions $f : \mathcal{X}_1 \to \mathcal{X}_2$ where $\mathcal{X}_1, \mathcal{X}_2$ depict subsets of normed spaces. Similarly, we write $L^\infty(\mathcal{X}_1, \mathcal{X}_2)$ for the space of functions $f : \mathcal{X}_1 \to \mathcal{X}_2$ with bounded norm $\|f(x)\|_{\mathcal{X}_2}$ for almost all $x \in \mathcal{X}_1$, and $\mathcal{W}^\infty(\mathcal{X}_1, \mathcal{X}_2)$ for the space of Fréchet differentiable functions with respect to $L^\infty(\mathcal{X}_1, \mathcal{X}_2)$. The null vector is written as $\mathbf{o}$. We are interested in the following class of mixed-integer optimal control problems.

Definition 1. (MIOCP) We refer to the following control problem (2) as **(MIOCP)**.

$$\min_{\mathbf{x},\omega} \quad \Phi(\mathbf{x}(t_\mathrm{f})) \tag{2a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = \mathbf{f}_0(t, \mathbf{x}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) \mathbf{f}_i(t, \mathbf{x}(t)), \qquad \text{for a.e. } t \in \mathcal{T}, \tag{2b}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{2c}$$

$$1 = \sum_{i=1}^{n_\omega} \omega_i(t) \qquad \text{for a.e. } t \in \mathcal{T}, \tag{2d}$$

$$\omega(t) \in \{0,1\}^{n_\omega} \qquad \text{for a.e. } t \in \mathcal{T}, \tag{2e}$$

$$\mathbf{o} \leq \mathbf{c}(t, \mathbf{x}(t)) \qquad \text{for a.e. } t \in \mathcal{T}. \tag{2f}$$

We minimize a Mayer term $\Phi \in C^1(\mathbb{R}^{n_\mathrm{x}}, \mathbb{R})$ over differential states $\mathbf{x} \in \mathcal{W}^\infty(\mathcal{T}, \mathbb{R}^{n_\mathrm{x}})$ for binary control functions $\omega \in L^\infty(\mathcal{T}, \{0,1\}^{n_\omega})$. The system of ordinary differential

equations (ODE), (2b), is written using *partial outer convexification* to model the switched system, using a *Special Order Set of Type 1* (SOS1) constraint (2d), a drift term $\mathbf{f}_0$ and $n_\omega \geq 2$ functions $\mathbf{f}_i$, [35], both out of $C^0(\mathbb{R}^{n_\mathrm{x}+1}, \mathbb{R}^{n_\mathrm{x}})$. We assume fixed initial values $\mathbf{x}_0 \in \mathbb{R}^{n_\mathrm{x}}$ for the differential states. The functions $\mathbf{c} \in C^1(\mathbb{R}^{n_\mathrm{x}}, \mathbb{R}^{n_c})$ model nonlinear state inequalities.

Remark 1. See [15, 36, 41] for a discussion of the generality of (MIOCP) and extensions to cope with, e.g., Lagrange functionals, boundary and multi-point constraints, vanishing constraints, free final time, control values and the like. Particularly interesting are continuous control functions $\mathbf{u} \in L^\infty(\mathcal{T}, \mathcal{U})$ that often enter (2b,2f) in practical applications. From a theoretical point of view, in the interest of comparability, and for computational speed it is convenient to consider the continuous controls $u(\cdot)$ as fixed to the solution that was obtained by solving the continuous relaxation of (MIOCP) in our approach. It is also possible though, and often makes sense in practice, to improve the objective function value by reoptimizing $\mathbf{u}$ when (MIOCP) is evaluated for fixed ω. For notational convenience and without loss of generality, we omit the continuous controls $\mathbf{u}$ in the following. An *evaluation* of (MIOCP) for fixed ω is then a solution of an initial value problem.

Our algorithm solves continuous relaxations of (MIOCP), defined as follows.

Definition 2. (OCP) We define **(OCP)** as the canonical relaxation of (MIOCP) with respect to (2e). We substitute hence $\omega \in L^\infty(\mathcal{T}, \{0,1\}^{n_\omega})$ by $\alpha \in L^\infty(\mathcal{T}, [0,1]^{n_\omega})$.

The problem (OCP) in function space can be solved by different approaches, as mentioned above. To be able to work with MILPs to approximate control functions, we map between function space and $[0,1]^{n_\omega \times M}$ using a time grid as follows.

Definition 3. ($\mathcal{G}_\omega$, Δ, φ, φ^{-1}, Ω, W) Let the ordered set $\mathcal{G}_\omega := \{t_0 < \ldots < t_M = t_\mathrm{f}\}$ denote a time grid with $\Delta_j = t_{j+1} - t_j$ and $\Delta_{\max} = \max_j \Delta_j$ for $j = 0 \ldots M-1$.

We define the mapping

$$\varphi : [0,1]^{n_\omega \times M} \to L^\infty(\mathcal{T}, [0,1]^{n_\omega}), \quad \alpha = \varphi(\mathbf{a})$$

using n_ω piecewise constant functions

$$\alpha_i(t) := a_{i,j}, \qquad i \in 1, \ldots, n_\omega,\ t \in [t_j,\, t_{j+1}),\ j = 0 \ldots M-1,\ t_j \in \mathcal{G}_\omega.$$

A mapping in reverse direction

$$\varphi^{-1} : L^\infty(\mathcal{T}, [0,1]^{n_\omega}) \to [0,1]^{n_\omega \times M}, \quad \mathbf{a} = \varphi^{-1}(\alpha)$$

is defined by extracting integrals on the grid $\mathcal{G}_\omega$,

$$a_{i,j} := \frac{1}{\Delta_j} \int_{t_j}^{t_{j+1}} \alpha_i(\tau) \mathrm{d}\tau, \qquad i \in 1, \ldots, n_\omega,\ j = 0 \ldots M-1,\ t_j \in \mathcal{G}_\omega.$$

Both maps conserve integrality, i.e., $\alpha = \varphi(\mathbf{a}) \in L^\infty(\mathcal{T}, \{0,1\}^{n_\omega})$ for $\alpha \in \{0,1\}^{n_\omega \times M}$,

and the other way around. We denote integrality and SOS1 using the sets

$$\Omega := \left\{ \omega \in L^\infty(\mathcal{T}, \{0,1\}^{n_\omega}) : 1 = \sum_{i=1}^{n_\omega} \omega_i(t) \text{ for a.e. } t \in \mathcal{T} \right\},$$
$$W := \left\{ \mathbf{w} \in \{0,1\}^{n_\omega \times M} : 1 = \sum_{i=1}^{n_\omega} w_{i,j} \text{ for } j = 0 \dots M-1 \right\}.$$

To scale control variables we are going to need function evaluations and adjoint (dual) variables on the grid $\mathcal{G}_\omega$.

Definition 4. (λ, $\tilde{\mathbf{f}}$) Let for a solution of (OCP) $\lambda_{j,k} \in \mathbb{R}$, $t_j \in \mathcal{G}_\omega$, $k \in 1, \dots n_\mathrm{x}$ be the discretized and evaluated dual variables and $\{\tilde{f}_{i,j,k}\}_{k \in 1,\dots,n_\mathrm{x}}$ be the entries of

$$\mathbb{R}^{n_\mathrm{x}} \ni \tilde{\mathbf{f}}_{i,j} := \frac{1}{\Delta t_j} \int_{t_j}^{t_{j+1}} \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \, \mathrm{d}\tau, \quad i \in 1, \dots, n_\omega, \quad t_j \in \mathcal{G}_\omega.$$

Algorithm 1: Decomposition of (MIOCP)

Input : (MIOCP) instance, grid $\mathcal{G}_\omega$, algorithmic choices in sets S^{CIA} and S^{REC}.

1 Solve (OCP) $\rightarrow$ Φ_{rel}, $\mathbf{x}$, α, $\mathbf{a} = \varphi^{-1}(\alpha)$.
2 **for** *milp* $\in S^{CIA}$ **do**
3 | Solve milp for data $\mathbf{a}$ with MILP solver $\rightarrow$ $\mathbf{w}^{\mathrm{milp}}$.
4 | Evaluate (MIOCP) with fixed $\omega^{\mathrm{milp}} := \varphi(\mathbf{w}^{\mathrm{milp}})$ $\rightarrow$ Φ_{milp}, $\mathbf{x}$.
5 **end**
6 **for** *rec* $\in S^{REC}$ **do**
7 | Create $\mathbf{w}^{\mathrm{rec}}$ using $\mathbf{w}^{\mathrm{milp}}$, Φ_{milp} from all milp $\in S^{\mathrm{CIA}}$.
8 | Evaluate (MIOCP) with fixed $\omega^{\mathrm{rec}} := \varphi(\mathbf{w}^{\mathrm{rec}})$ $\rightarrow$ Φ_{rec}, $\mathbf{x}$.
9 **end**
10 Set $\Phi^* = \min \left\{ \min_{\mathrm{milp} \in S^{\mathrm{CIA}}} \Phi_{\mathrm{milp}}, \min_{\mathrm{rec} \in S^{\mathrm{REC}}} \Phi_{\mathrm{rec}} \right\}$, return $\Phi^*, \mathbf{x}^\star, \mathbf{w}^\star$, lower bound Φ_{rel}.

We propose to use Algorithm 1 to approximate the solution of (MIOCP) efficiently with a priori bounds. Relaxing (MIOCP) to (OCP) results in state and control trajectories (Line 1). We solve different MILPs to approximate the relaxed controls with binary ones in Lines 2–3. Their performance is evaluated in Line 4 by calculating their corresponding state trajectories and objective values. In Lines 6–7, we use the binary controls in several recombination heuristics in order to create new candidate binary controls, which we evaluate as well (Line 8). Finally, the best performing binary control is selected as solution in Line 10.

The main idea of Algorithm 1 is to decouple controls and states. We approximate the relaxed control function α that is optimal for (OCP) with binary controls ω such that a good objective function value is obtained when (MIOCP) is evaluated for ω. Which MILPs and recombination heuristics are used in the algorithm depends on the definition of the sets S^{CIA} and S^{REC}, which we discuss in Section 3 and Section 5. A theoretical motivation and error bounds are given in Section 4. Algorithm 1 is a

generalization of the decomposition approach in [27, 35, 40], for which S^{REC} is empty and S^{CIA} contains only one CIA problem formulation.

3. Combinatorial Integral Approximation MILPs

We are going to define the following MILP formulations of combinatorial integral approximation type for S^{CIA} in Algorithm 1

$$S^{\mathrm{CIA}} := \{(\mathrm{CIAmax}), (\mathrm{CIA1}), (\mathrm{CIAmaxB}), (\mathrm{CIA1B}), (\lambda\mathrm{CIA1}), (\lambda\mathrm{CIA1B}), \\ (\mathrm{SCIAmax}), (\mathrm{SCIA1}), (\mathrm{SCIAmaxB}), (\mathrm{SCIA1B})\}$$

CIA, λCIA, and SCIA refer to different scalings (Section 3.1), 1 and max to different norms $\|\cdot\|$ (Section 3.2), and B to a reversal of time (Section 3.3).

3.1. Combinatorial integral approximation and scaled variants

We consider the following approximation problems in the control function space.

Definition 5. (CIA, SCIA, λCIA) Let $\mathbf{x}$, α be optimal for (OCP). Let $\mathbf{f}_i(t, \mathbf{x}(t))$ be evaluations of the right hand side functions, and $\lambda \in L^\infty(\mathcal{T}, \mathbb{R}^{n_\mathrm{x}})$ be the dual variables of the ODE constraint (2b). We define the following (scaled) Combinatorial Integral Approximation problems and corresponding optimal objective values,

$$\delta^\star_{\mathrm{CIA}} := \min_{\omega(\cdot)\in\Omega} \max_{t\in\mathcal{T}} \left\| \int_{t_0}^{t} \alpha(\tau) - \omega(\tau) \,\mathrm{d}\tau \right\|, \tag{CIA}$$

$$\delta^\star_{\mathrm{SCIA}} := \min_{\omega(\cdot)\in\Omega} \max_{t\in\mathcal{T}} \left\| \int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i(\tau)) \mathbf{f}_i(t, \mathbf{x}(t)) \,\mathrm{d}\tau \right\|, \tag{SCIA}$$

$$\delta^\star_{\lambda\mathrm{CIA}} := \min_{\omega(\cdot)\in\Omega} \max_{t\in\mathcal{T}} \left| \sum_{k=1}^{n_\mathrm{x}} \lambda_k(t) \cdot \int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i(\tau)) f_{i,k}(t, \mathbf{x}(t)) \,\mathrm{d}\tau \right|. \tag{λCIA}$$

Discretizing with $\mathbf{a} = \varphi^{-1}(\alpha)$, $\tilde{\mathbf{f}}$, and λ from Definition 4 results in

$$\min_{\mathbf{w}\in W} \max_{t_l\in\mathcal{G}_\omega} \left\| \sum_{j=0}^{l} (\mathbf{a}_{\cdot,j} - \mathbf{w}_{\cdot,j}) \cdot \Delta t_j \right\|, \tag{3}$$

$$\min_{\mathbf{w}\in W} \max_{t_l\in\mathcal{G}_\omega} \left\| \sum_{j=0}^{l} \sum_{i=1}^{n_\omega} (a_{i,j} - w_{i,j}) \tilde{\mathbf{f}}_{i,j} \, \Delta t_j \right\|, \tag{4}$$

$$\min_{\mathbf{w}\in W} \max_{t_l\in\mathcal{G}_\omega} \left| \sum_{k=1}^{n_\mathrm{x}} \lambda_{l,k} \cdot \sum_{j=0}^{l} \sum_{i=1}^{n_\omega} (a_{i,j} - w_{i,j}) \tilde{f}_{i,j,k} \, \Delta t_j \right|, \tag{5}$$

respectively. To facilitate notation, we now assume an equidistant grid with $\Delta_j = 1$.

3.2. Norm dependent MILP formulation

We introduce slack variables to formulate (3–5) with different norms as MILPs. We start with the maximum norm in (3–4).

Definition 6. (CIAmax, SCIAmax) For given data **a** we define **(CIAmax)** as

$$\min_{\delta, \mathbf{w} \in W} \quad \delta \tag{6a}$$

$$\text{s.t.} \quad \delta \geq \pm \sum_{j=0}^{l} a_{i,j} - w_{i,j}, \qquad \text{for } i = 1, \ldots, n_\omega, \quad t_l \in \mathcal{G}_\omega, \tag{6b}$$

and **(SCIAmax)** as

$$\min_{\delta, \mathbf{w} \in W} \quad \delta \tag{7a}$$

$$\text{s.t.} \quad \delta \geq \pm \sum_{j=0}^{l} \sum_{i=1}^{n_\omega} (a_{i,j} - w_{i,j}) \cdot \tilde{f}_{i,j,k}, \qquad \text{for } k = 1, \ldots, n_\text{x}, \quad t_l \in \mathcal{G}_\omega. \tag{7b}$$

We now define MILPs based on the Manhattan norm.

Definition 7. (CIA1, SCIA1, λCIA1) For given data **a** and slack variables $s_{i,l}$ we define **(CIA1)** as

$$\min_{\delta, s_{i,l}, \mathbf{w} \in W} \quad \delta \tag{8a}$$

$$\text{s.t.} \quad \delta \geq \sum_{i=1}^{n_\omega} s_{i,l}, \qquad \text{for } t_l \in \mathcal{G}_\omega, \tag{8b}$$

$$s_{i,l} \geq \pm \sum_{j=0}^{l} a_{i,j} - w_{i,j}, \qquad \text{for } i = 1, \ldots, n_\omega, \quad t_l \in \mathcal{G}_\omega \tag{8c}$$

and with a different dimension for $s_{k,l}$ we define **(SCIA1)** as

$$\min_{\delta, s_{k,l}, \mathbf{w} \in W} \quad \delta \tag{9a}$$

$$\text{s.t.} \quad \delta \geq \sum_{k=1}^{n_\text{x}} s_{k,l}, \qquad \text{for } t_l \in \mathcal{G}_\omega, \tag{9b}$$

$$s_{k,l} \geq \pm \sum_{j=0}^{l} \sum_{i=1}^{n_\omega} (a_{i,j} - w_{i,j}) \cdot \tilde{f}_{i,j,k}, \qquad \text{for } k = 1, \ldots, n_\text{x}, \quad t_l \in \mathcal{G}_\omega. \tag{9c}$$

Motivated by the similarity of (5), we define **(λCIA1)** by replacing (9c) as

$$\min_{\delta, s_{k,l}, \mathbf{w} \in W} \quad \delta \tag{10a}$$

$$\text{s.t.} \quad \delta \geq \sum_{k=1}^{n_{\mathrm{x}}} s_{k,l}, \qquad \text{for } t_l \in \mathcal{G}_\omega, \tag{10b}$$

$$s_{k,l} \geq \pm \lambda_{l,k} \cdot \sum_{j=0}^{l} \sum_{i=1}^{n_\omega} (a_{i,j} - w_{i,j}) \tilde{f}_{i,j,k}, \qquad \text{for } k = 1 \dots n_{\mathrm{x}},\ t_l \in \mathcal{G}_\omega. \tag{10c}$$

Of course also other norms, such as the Euclidean norm, can be used. We do not consider them here, because of the nonlinearity of the constraints.

3.3. Chronologically ordered constraints

One further possibility to modify MILPs is the chronological order in the constraints for the accumulated difference $\|\mathbf{a} - \mathbf{w}\|$. Instead of starting from t_0, we may use an arbitrary ordering of time points. We consider backwards accumulation, starting from the interval $[t_{M-1}, t_M)$:

$$\delta \geq \pm \sum_{j=l}^{M-1} a_{i,j} - w_{i,j}, \qquad \text{for } i = 1, \dots, n_\omega, \quad t_l \in \mathcal{G}_\omega. \tag{11}$$

We denote the problem where (11) replaces (6b) in (CIAmax) by **(CIAmaxB)**. The other defined problems can be modified analogously with backward time accumulation and are named accordingly.

3.4. Combinatorial constraints

One advantage of using an MILP for obtaining binary controls after the relaxation step, rather than using SUR, is the possibility to impose combinatorial constraints that couple over time. An example of such constraints is to limit the number of allowed switches $\sigma_{\max} \in \mathbb{N}$ between activated binary controls, which can be formulated as

$$\sigma_{\max} \geq \frac{1}{2} \sum_{i=1}^{n_\omega} \sum_{j=1}^{M-1} |w_{i,j} - w_{i,j-1}|. \tag{12}$$

We will omit this constraint class in the next section, but refer to [42, 48] for a priori bounds and reformulations. Nevertheless, we are going to come back to these constraints in the numerical experiments section for testing the different MILP approaches also in this situation.

4. A priori bounds for CIA decompositions

We revise results for the a priori bounds resulting from a CIA decomposition [38] and extend them to alternative decompositions that may be used in S^{CIA} in Algorithm 1.

We stress here that Manns et al. [33] have recently presented a proof of improved regularity conditions. Nevertheless, we revise the proof from [38] here, because it results in natural algorithmic extensions.

4.1. Combinatorial integral approximation

This variant of Gronwall's Lemma is used in the following proofs.

Lemma 4.1. *Let us first define* $\|\mathbf{v}(t)\|_{\lambda(t)} := |\sum_{k=1}^{n_x} \lambda_k(t) \cdot v_k(t)|, \quad \lambda(t), \mathbf{v}(t) \in L^\infty(\mathcal{T}, \mathbb{R}^{n_x})$*. Let* $z \in L^\infty(\mathcal{T}, \mathbb{R})$*, and for a constant* $L_G \in \mathbb{R}_0^+$ *it holds*

$$\|\mathbf{v}(t)\|_{\lambda(t)} \leq z(t) + L_G \int_{t_0}^{t} \|\mathbf{v}(\tau)\|_{\lambda(t)} d\tau \tag{13}$$

for $t \in \mathcal{T}$ *a.e. Then, it also holds*

$$\|\mathbf{v}(t)\|_{\lambda(t)} \leq \|z(\cdot)\|_\infty \cdot e^{L_G(t-t_0)}. \tag{14}$$

Proof. See [13], pp. 169-182, for the case with $\lambda(t) = 1$, $\forall t \in \mathcal{T}$ and without applied absolute value. The proof is applicable to the general case where $v(t) \in L^\infty(\mathcal{T}, \mathbb{R})$ is replaced by $\|\mathbf{v}(t)\|_{\lambda(t)}$, which is by the choice of $\mathbf{v}(t), \lambda(t)$ also part of $L^\infty(\mathcal{T}, \mathbb{R})$. □

The idea behind the following results is to analyze the evolution of two trajectories $\mathbf{x}$ and $\mathbf{y}$ based on the same ODE system but driven by two different controls α and ω. We want to compare the distance between the two trajectories depending on the distance of the controls.

Theorem 4.2. *Let* $\alpha, \omega, \mathbf{f}_0, \mathbf{f}_i$ *be given as defined previously and* $t \in \mathcal{T}$*. Let* $\mathbf{x}(\cdot)$ *and* $\mathbf{y}(\cdot)$ *be the unique solutions of the initial value problems*

$$\dot{\mathbf{x}}(t) = \mathbf{f}_0(t, \mathbf{x}(t)) + \sum_{i=1}^{n_\omega} \alpha_i(t) \mathbf{f}_i(t, \mathbf{x}(t)), \qquad \mathbf{x}(t_0) = \mathbf{x_0}, \tag{15a}$$

$$\dot{\mathbf{y}}(t) = \mathbf{f}_0(t, \mathbf{y}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) \mathbf{f}_i(t, \mathbf{y}(t)), \qquad \mathbf{y}(t_0) = \mathbf{y_0}. \tag{15b}$$

Assume there are positive constants $\delta, C, L_i, B \in \mathbb{R}^+$*,* $i = 0, \ldots, n_\omega$*, together with a vector norm* $||\cdot||$ *and a consistent matrix norm* $|||\cdot|||$ *such that for all* $t \in \mathcal{T}$*:*

$$\left|\left|\left| \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{f}_i(t, \mathbf{x}(t)) \right|\right|\right| \leq C, \qquad \text{for } i = 1, \ldots, n_\omega, \tag{15c}$$

$$|||\mathbf{f}_i(t, \mathbf{y}(t)) - \mathbf{f}_i(t, \mathbf{x}(t))||| \leq L_i \, ||\mathbf{y}(t) - \mathbf{x}(t)|| \,, \qquad \text{for } i = 0, \ldots, n_\omega. \tag{15d}$$

Furthermore, let $\mathbf{f}_i(\cdot, \mathbf{x}(\cdot))$*,* $i = 1, \ldots, n_\omega$ *be essentially bounded by* B *on* $\mathcal{T}$*, and it holds for all* $t \in \mathcal{T}$

$$\left|\left| \int_{t_0}^{t} \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right|\right| \leq \delta, \tag{15e}$$

then it also holds for all $t \in \mathcal{T}$

$$||\mathbf{y}(t) - \mathbf{x}(t)|| \leq (||\mathbf{y_0} - \mathbf{x_0}|| + n_\omega \left(B + C(t - t_0)\right) \delta) \, e^{L(t-t_0)}. \tag{15f}$$

Proof. We start with some notations and observations before we approximate the normed difference of $\mathbf{y}$ and $\mathbf{x}$. At first, the Fundamental Theorem of Calculus yields for (15a, 15b) and $t \in \mathcal{T}$

$$\begin{aligned}
\mathbf{x}(t) &= \mathbf{x_0} + \int_{t_0}^{t} \dot{\mathbf{x}}(\tau) \, \mathrm{d}\tau = \mathbf{x_0} + \int_{t_0}^{t} \mathbf{f}_0(\tau, \mathbf{x}(\tau)) + \sum_{i=1}^{n_\omega} \alpha_i(\tau) \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \;\; \mathrm{d}\tau, \\
\mathbf{y}(t) &= \mathbf{y_0} + \int_{t_0}^{t} \dot{\mathbf{y}}(\tau) \, \mathrm{d}\tau = \mathbf{y_0} + \int_{t_0}^{t} \mathbf{f}_0(\tau, \mathbf{y}(\tau)) + \sum_{i=1}^{n_\omega} \omega_i(\tau) \mathbf{f}_i(\tau, \mathbf{y}(\tau)) \;\; \mathrm{d}\tau.
\end{aligned}$$

From the definition of α, ω it is clear that

$$||\alpha(t)|| \leq 1, \qquad ||\omega(t)|| \leq 1.$$

For brevity we define the auxiliary terms:

$$L := \sum_{i=0}^{n_\omega} L_i, \quad \Delta\omega_i(t) := \alpha_i(t) - \omega_i(t), \quad \Delta a_i(t) := \int_{t_0}^{t} \Delta\omega_i(\tau) \, \mathrm{d}\tau.$$

Note that $\Delta a_i(t_0) = 0$ and because of (15e) it holds $||\Delta a_i(t)|| \leq \delta$. Using these observations, it follows for all $t \in \mathcal{T}$

$$\begin{aligned}
||\mathbf{y}(t) - \mathbf{x}(t)|| &\leq ||\mathbf{y_0} - \mathbf{x_0}|| + \left\| \int_{t_0}^{t} \mathbf{f}_0(\tau, \mathbf{y}(\tau)) + \sum_{i=1}^{n_\omega} \omega_i(\tau) \mathbf{f}_i(\tau, \mathbf{y}(\tau)) \right. \\
&\qquad \left. - \mathbf{f}_0(\tau, \mathbf{x}(\tau)) - \sum_{i=1}^{n_\omega} \alpha_i(\tau) \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \, \mathrm{d}\tau \right\| \\
&\leq ||\mathbf{y_0} - \mathbf{x_0}|| + \left\| \int_{t_0}^{t} \mathbf{f}_0(\tau, \mathbf{y}(\tau)) + \sum_{i=1}^{n_\omega} \omega_i(\tau) \mathbf{f}_i(\tau, \mathbf{y}(\tau)) - \mathbf{f}_0(\tau, \mathbf{x}(\tau)) \right. \\
&\qquad \left. - \sum_{i=1}^{n_\omega} \omega_i(\tau) \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \right\| + \left\| \int_{t_0}^{t} \mathbf{f}_0(\tau, \mathbf{x}(\tau)) + \sum_{i=1}^{n_\omega} \omega_i(\tau) \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \right. \\
&\qquad \left. - \mathbf{f}_0(\tau, \mathbf{x}(\tau)) - \sum_{i=1}^{n_\omega} \alpha_i(\tau) \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \, \mathrm{d}\tau \right\| \\
&\leq ||\mathbf{y_0} - \mathbf{x_0}|| + \int_{t_0}^{t} \sum_{i=0}^{n_\omega} ||\mathbf{f}_i(\tau, \mathbf{y}(\tau)) - \mathbf{f}_i(\tau, \mathbf{x}(\tau))|| \, ||\omega_i(\tau)|| \;\; \mathrm{d}\tau \\
&\qquad + \left\| \sum_{i=1}^{n_\omega} \int_{t_0}^{t} \Delta\omega_i(\tau) \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \, \mathrm{d}\tau \right\| \\
&\leq ||\mathbf{y_0} - \mathbf{x_0}|| + \int_{t_0}^{t} \sum_{i=0}^{n_\omega} L_i \, ||\mathbf{y}(\tau) - \mathbf{x}(\tau)|| \;\; \mathrm{d}\tau
\end{aligned}$$

$$
\begin{aligned}
&+ \left|\left| \sum_{i=1}^{n_\omega} \Delta a_i(t) \mathbf{f}_i(t, \mathbf{x}(t)) - \int_{t_0}^{t} \Delta a_i(\tau) \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \, \mathrm{d}\tau \right|\right| \\
&\leq ||\mathbf{y_0} - \mathbf{x_0}|| + L \int_{t_0}^{t} ||\mathbf{y}(\tau) - \mathbf{x}(\tau)|| \, \mathrm{d}\tau \\
&\quad + \sum_{i=1}^{n_\omega} |\Delta a_i(t)| \, ||\mathbf{f}_i(t, \mathbf{x}(t))|| + \int_{t_0}^{t} |\Delta a_i(\tau)| \left|\left| \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \right|\right| \mathrm{d}\tau \\
&\leq ||\mathbf{y_0} - \mathbf{x_0}|| + L \int_{t_0}^{t} ||\mathbf{y}(\tau) - \mathbf{x}(\tau)|| \, \mathrm{d}\tau + n_\omega \left(B + C(t - t_0)\right) \delta.
\end{aligned}
$$

Partial integration was applied in step four. To apply Lemma 4.1 with $\lambda(t) = 1$, we define

$$
\begin{aligned}
v(t) &:= ||\mathbf{y}(t) - \mathbf{x}(t)|| \,, \\
z(t) &:= ||\mathbf{y_0} - \mathbf{x_0}|| + n_\omega \left(B + C(t - t_0)\right) \delta
\end{aligned}
$$

which satisfy the assumptions of the lemma, and using the result on the last inequality yields the claim

$$
||\mathbf{y}(t) - \mathbf{x}(t)|| \leq \left(||\mathbf{y_0} - \mathbf{x_0}|| + n_\omega \left(B + C(t - t_0)\right) \delta\right) e^{L(t - t_0)}.
$$

□

Corollary 1. (Approximation bounds via **(CIA)**)
Assume that $\mathbf{y}_{\mathrm{CIA}}(\cdot), \mathbf{x}(\cdot)$ are the solutions of the IVP of the above theorem, where the binary solutions of (CIA) have been applied for $\mathbf{y}$. Then, the state approximation error is bounded for all $t \in \mathcal{T}$ by

$$
||\mathbf{y}_{\mathrm{CIA}}(t) - \mathbf{x}(t)|| \leq \left(||\mathbf{y_0} - \mathbf{x_0}|| + \left(B + C(t - t_0)\right) \delta^\star_{\mathrm{CIA}}\right) e^{L(t - t_0)}.
$$

4.2. Scaled combinatorial integral approximation

The following result shows that (CIA) is dominated by (SCIA) in terms of approximation accuracy.

Corollary 2. (Approximation bounds via **(SCIA)**)
Apply the notations from Theorem 4.2 and let $\mathbf{y}_{\mathrm{SCIA}}$ be analogously defined as in Corollary 1. It follows for all $t \in \mathcal{T}$:

$$
\begin{aligned}
||\mathbf{y}_{\mathrm{SCIA}}(t) - \mathbf{x}(t)|| &\leq \left(||\mathbf{y_0} - \mathbf{x_0}|| + \delta^\star_{\mathrm{SCIA}}\right) e^{L(t - t_0)} \\
&\leq \left(||\mathbf{y_0} - \mathbf{x_0}|| + n_\omega \left(B + C(t - t_0)\right) \delta^\star_{\mathrm{CIA}}\right) e^{L(t - t_0)}.
\end{aligned}
$$

Proof. From the proof of Theorem 4.2 follows for $t \in \mathcal{T}$ and any ω with corresponding $\mathbf{y}$

$$
\left|\left| \int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i(\tau)) \, \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \, \mathrm{d}\tau \right|\right| \leq n_\omega \left(B + C(t - t_0)\right) \cdot \left|\left| \int_{t_0}^{t} \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right|\right|.
$$

Taking the minimum yields

$$\delta^\star_{\mathrm{SCIA}} \leq \left|\left| \int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega^{\mathrm{CIA}}(\tau))\ \mathbf{f}_i(\tau, \mathbf{x}(\tau))\ \mathrm{d}\tau \right|\right| \leq n_\omega\,(B + C(t-t_0))\,\delta^\star_{\mathrm{CIA}}.$$

□

The corollary states bounds for the normed difference of the trajectories based on relaxed and (SCIA)- binary controls. Moreover, the bounds are compared with those for (CIA)- binary controls from Theorem 4.2. The scaled bound turns out to be tighter than the existing (CIA)- bound. Thus, it is obvious to consult these alternative binary controls for an approximation study. Ideally, $\delta^\star_{\mathrm{SCIA}} < n_\omega\,(B + C(t-t_0))\,\delta^\star_{\mathrm{CIA}}$ holds and results in improved objective values. Yet, we confine this hope already with the following remark.

Remark 2. Using (SCIA) for binary control approximation results not necessarily in better state approximation or objective value than using (CIA). The computational results section specifies instances, but anticipate already what can happen:

(1) It may hold

$$||\mathbf{y}_{\mathrm{CIA}}(t) - \mathbf{x}(t)|| < ||\mathbf{y}_{\mathrm{SCIA}}(t) - \mathbf{x}(t)|| < \delta^\star_{\mathrm{SCIA}}$$

for $t \in \mathcal{T}$.

(2) Even if the above inequality holds reversely, the computed trajectories may fulfill

$$\Phi(\mathbf{y}_{\mathrm{CIA}}(t_{\mathrm{f}})) < \Phi(\mathbf{y}_{\mathrm{SCIA}}(t_{\mathrm{f}}))$$

because of, e.g., non-convexities of the objective.

4.3. λ-combinatorial integral approximation

The λ-approximation stems from the aforementioned theorem of approximating differential states, but with assessing the difference of the soon defined cost-to-go function. We define it in a more general MIOCP setting with a Bolza objective

$$\Phi(\mathbf{x}(t_{\mathrm{f}})) + \int_{t\in\mathcal{T}} L(\mathbf{x}(\tau), \omega(\tau))\ \mathrm{d}\tau, \tag{16}$$

where $L \in C^1(\mathbb{R}^{n_{\mathrm{x}}+1} \times \mathbb{R}^{n_\omega}, \mathbb{R})$.

Definition 8. (Cost-to-go function by Hamilton-Jacobi-Bellman)
Let the cost-to-go function $J \in L^\infty(\mathbb{R}^{n_{\mathrm{x}}} \times \mathcal{T}, \mathbb{R})$ with Bolza objective (16) be implicitly defined as

$$\begin{aligned} J(\mathbf{x}(t_{\mathrm{f}}), t_{\mathrm{f}}) &= \Phi(\mathbf{x}(t_{\mathrm{f}})), \\ -\frac{\partial J}{\partial t}(\mathbf{x}(t), t) &= \min_{\omega\in\Omega} L(\mathbf{x}(t), \omega(t)) + \frac{\partial J}{\partial x}(\mathbf{x}(t), t) \left(\mathbf{f}_0(t, \mathbf{x}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) \mathbf{f}_i(t, \mathbf{x}(t)) \right). \end{aligned}$$

We recognize that $\frac{\partial J}{\partial x}$ can be interpreted as Lagrange multiplier or dual variables of the ODE constraints in (MIOCP). Therefore we write in short $\lambda(t)$ instead of $\frac{\partial J}{\partial x}$. With the groundwork above we are ready to deduce bounds:

Corollary 3. (Approximation bounds via **(λCIA)**)
Apply the notations from Theorem 4.2 and Definition 8 with state trajectories $\mathbf{y}_{\lambda\text{CIA}}(\cdot)$ and $\mathbf{x}(\cdot)$ driven by controls $\omega^{\lambda\text{CIA}}$ and α. It follows for all $t \in \mathcal{T}$:

$$|J(\mathbf{y}_{\lambda\text{CIA}}(t), t) - J(\mathbf{x}(t), t)| \leq \left(\|\mathbf{y_0} - \mathbf{x_0}\|_{\lambda(t)} + \delta^{\star}_{\lambda\text{CIA}}\right) e^{L(t-t_0)} + \mathcal{O}\left(||\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x}||^2\right).$$

Proof. We consider the difference of the cost-to-go functions by approximating with a partial first order Taylor expansion around $J(\mathbf{x}(t), t)$. The approximation is done in respect of the trajectories $\mathbf{x}$, $\mathbf{y}_{\lambda\text{CIA}}$. Hence, for $t \in \mathcal{T}$

$$J(\mathbf{y}_{\lambda\text{CIA}}(t), t) - J(\mathbf{x}(t), t) = \frac{\partial J}{\partial \mathbf{x}}(\mathbf{x}(t), t)\left(\mathbf{y}_{\lambda\text{CIA}}(t) - \mathbf{x}(t)\right) + \mathcal{O}\left(||\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x}||^2\right).$$

Rewriting the terms with absolute values, using the notation for λ and $|\cdot|_{\lambda(t)}$ yields

$$\begin{aligned}
|J(\mathbf{y}_{\lambda\text{CIA}}(t), t) - J(\mathbf{x}(t), t)| &\leq \|\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x}\|_{\lambda(t)} + \mathcal{O}\left(||\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x}||^2\right) \\
&\leq \quad \ldots \text{ (as in proof of Theorem 4.2)} \\
&\leq ||\mathbf{y_0} - \mathbf{x_0}||_{\lambda(t)} + L \int_{t_0}^{t} ||\mathbf{y}_{\lambda\text{CIA}}(\tau) - \mathbf{x}(\tau)\, \mathrm{d}\tau||_{\lambda(t)} \\
&\quad + \left|\left| \sum_{i=1}^{n_\omega} \int_{t_0}^{t} (\omega_i^{\lambda\text{CIA}}(\tau) - \alpha_i(\tau)) \cdot \mathbf{f}_i(\tau, \mathbf{x}(\tau))\ \mathrm{d}\tau \right|\right|_{\lambda(t)} \\
&\quad + \mathcal{O}\left(||\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x}||^2\right).
\end{aligned}$$

The third addend of the last inequality represents the objective (λCIA1). Thus, we apply the adapted Gronwall Lemma 4.1 to $v = (\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x})$ and $z = ||\mathbf{y_0} - \mathbf{x_0}||_{\lambda(t)} + \left|\left| \sum_{i=1}^{n_\omega} \int_{t_0}^{t} (\omega_i^{\lambda\text{CIA}}(\tau) - \alpha_i(\tau)) \cdot \mathbf{f}_i(\tau, \mathbf{x}(\tau))\ \mathrm{d}\tau \right|\right|_{\lambda(t)}$ so that the claim is proven. □

Remark 3. One could modify and extend Corollary 3.

(1) For the sake of clearness in (λCIA), we added the Lagrange term to the objective. On the other hand, Mayer and Lagrange terms can be transformed into each other and therefore we have assumed a Mayer term as objective for the rest of the paper.
(2) The upper bounds from Corollary 3 resemble in a sense those obtained from (SCIA), i.e., Corollary 2. It is possible to further approximate the upper bound to obtain similar bounds as for the (CIA) case.
(3) (λCIA) benefits from a first degree Taylor approximation. One might think about higher degree expansions for tighter bounds.

4.4. Backwards accumulating constraints

If we adapt the MIOCP instance with fixed final states $\mathbf{x_f} \in \mathbb{R}^{n_x}$ and with a Lagrangian objective type, we can also apply the altered setting to Theorem 4.2. The following corollary deals with this issue.

Corollary 4. (Approximation bounds via backwards constraints)
Under the assumptions of Theorem 4.2, with $\mathbf{x}(\cdot)$ and $\mathbf{y}(\cdot)$ being the solutions not of the initial value problems, but of the problems (2b) togehter with fixed final states $\mathbf{x}(t_\mathrm{f}) = \mathbf{x_f}$ and $\mathbf{y}(t_\mathrm{f}) = \mathbf{y_f}$. Assume it holds for all $t \in \mathcal{T},\ \delta_b \in \mathbb{R}^+$

$$\left|\left| \int_t^{t_\mathrm{f}} \alpha(\tau) - \omega(\tau)\, \mathrm{d}\tau \right|\right| \leq \delta_b, \tag{18a}$$

then it also holds for all $t \in \mathcal{T}$

$$||\mathbf{y}(t) - \mathbf{x}(t)|| \leq (||\mathbf{y_f} - \mathbf{x_f}|| + n_\omega\, (B + C(t_\mathrm{f} - t))\, \delta_b)\, e^{L(t_\mathrm{f}-t)}. \tag{18b}$$

Proof. Note that by the Fundamental Theorem of Calculus

$$\mathbf{x}(t) = \mathbf{x}_f - \int_t^{t_\mathrm{f}} \mathbf{f}_0(\tau, \mathbf{x}(\tau)) + \sum_{i=1}^{n_\omega} \alpha_i(\tau) \mathbf{f}_i(\tau, \mathbf{x}(\tau))\, \mathrm{d}\tau,$$

holds and an analogous equality for $\mathbf{y}$. The proof of Theorem 4.2 can be applied to the altered setting. First, $||\mathbf{y_0} - \mathbf{x_0}||$ is replaced by $||\mathbf{y_f} - \mathbf{x_f}||$. Throughout the proof, we integrate over $[t, t_\mathrm{f}]$ instead of $[t_0, t]$, but the approximation ideas still apply. Finally, a variant of the Gronwall Lemma is used that bounds the normed $v(t)$ by $\|z(\cdot)\|_\infty \cdot e^{L_G(t_\mathrm{f}-t)}$. □

Remark 4. Relevance for MILP formulations:

(1) Corollary 4 can be used to deduce bounds for the discretized (CIA), (SCIA), and (λCIA) problems. For instance in case of (SCIA), the bounds are adapted according to Corollary 2.
(2) With the assumption of $\|\mathbf{x}(t_\mathrm{f}) - \mathbf{y}(t_\mathrm{f})\|$ being small, the backwards approach is appropriate also for (MIOCP) with given initial values $\mathbf{x}_0, \mathbf{y}_0$ and variable final states.

4.5. Connection to decomposition algorithm and optimization problem

As a last step in this chapter, the previous results are related to (MIOCP) and the decomposition Algorithm 1.

Remark 5. (Arbitrary close approximation of (OCP) solution)
We could extend Algorithm 1 with an outer loop which checks if Φ_bin is sufficiently close to Φ_rel and if not we would refine the grid $\mathcal{G}_\omega$. In [41], an arbitrary close approximation of the (OCP) solution has been deduced with this procedure and based on (CIA). With the assumption of $\Phi(\cdot)$, $c(\cdot)$ being continuous and that there exists a feasible trajectory $\mathbf{x}$ for (OCP), it follows that for any $\epsilon > 0$ there exists a grid $\mathcal{G}_\omega$, with grid size Δt

such that there is a feasible trajectory $\{\mathbf{y}(t_j)\}_{t_j \in \mathcal{G}_\omega}$ with

$$|\Phi(\mathbf{x}(t_\mathrm{f})) - \Phi(\mathbf{y}(t_\mathrm{f}))| \leq \epsilon,$$
$$\|\mathbf{c}(t_j, \mathbf{x}(t_j)) - \mathbf{c}(t_j, \mathbf{y}(t_j))\| \leq \epsilon.$$

The proof uses the Sum Up Rounding scheme that derives the binary control approximation with $\delta^\star_{\mathrm{CIA}} \leq \mathrm{Const}(n_\omega)\Delta t$ [29, 38], where $\mathrm{Const}(n_\omega)$ is a constant depending on n_ω. In case we extend Algorithm 1 with the refinement procedure and if (CIA) or (SCIA) are chosen to be elements of S^{CIA}, the same approximation result holds.

Corollary 5. (Solution accuracy of differential states of Algorithm 1)
Let $\delta^\star_{\mathrm{SCIA,max}}$, $\delta^\star_{\mathrm{SCIA},1}$ be the optimal objective values of (SCIAmax), respectively (SCIA1). Furthermore, let $\mathbf{x}$, $\mathbf{y}$ describe the trajectories obtained by applying the optimal $\mathbf{a}$, $\mathbf{w}$ from Algorithm 1 and $\mathcal{G}_\omega$ the applied grid. It follows for $t_i \in \mathcal{G}_\omega$

$$||\mathbf{y}(t_i) - \mathbf{x}(t_i)||_j \leq \left(||\mathbf{y_0} - \mathbf{x_0}||_j + \delta^\star_{\mathrm{SCIA},j}\right) e^{L(t_i - t_0)}, \qquad j \in \{\max, 1\}. \tag{19}$$

Proof. The claim is a direct result of Corollary 2. □

We have deliberately chosen the tightest bound from the previous corollaries, but could also use others. The received grid specific rounding error bounds aim primarily at approximating the differential states. But with (λCIA) or Lipschitz continuity of the objective, there are also tools to discuss the rounding error of the objectives. The so far omitted recombination heuristics work in the area of objective approximation without supporting error bounds.

5. Recombination heuristics

We present several recombination heuristics that recombine different binary controls $\mathbf{w}$ to new candidate solutions with potentially lower objective value. The general framework is open to apply different heuristics such as Genetic algorithms [17] that are not introduced in this article.

5.1. GreedyTime

Algorithm 2 establishes a routine to use the MILP solutions in a greedy approach, aiming at constructing solutions $\mathbf{w}$ with improved objective value $\Phi(\varphi(\mathbf{w}))$. Notice the abbreviations $[M] := \{1, \ldots, M\}$ and $\Phi_{m_1} = \Phi(\varphi(\mathbf{w}^{m_1}))$ in the algorithm.

GreedyTime iterates over all discretization intervals j in their natural order (Line 1). We check in Line 2 on every interval if there are MILP pairs (m_1, m_2), which differ in their binary solution vectors. For each of these pairs we recombine the m_1 solution with the binary vector solution from m_2 at interval j to create a temporary solution $\widetilde{\mathbf{w}}^{m_1}$ (Line 3). Then, we evaluate the objective of this new solution in Line 4 and overwrite the solution $\mathbf{w}^{m_1}$ with the recombined solution $\widetilde{\mathbf{w}}^{m_1}$ if it results in a better objective value (Lines 5–7). In the same way we proceed with the second solution m_2 when the (same) pair (m_2, m_1) comes up in the inner loop.

Note that with a high number of calculated $MILP$, there might be large number of pairs with unequal solutions. Instead of swapping and testing each variation for

Algorithm 2: GreedyTime heuristic for finding improved $\mathbf{w}$ variables

Input : Control grid $\mathcal{G}_\omega$, binary variable vectors $\mathbf{w}^{\text{milp}}$ as solutions of $milp \in S^{\text{CIA}}$, corresponding objectives $\Phi(\varphi(\mathbf{w}^{\text{milp}}))$.

1 **for** $j \in \mathcal{G}_\omega$ **do**
2 **for** $(m_1, m_2) \in S^{CIA} \times S^{CIA},\ m_1 \neq m_2,\ \mathbf{w}_j^{m_1} \neq \mathbf{w}_j^{m_2}$ **do**
3 Set $\widetilde{\mathbf{w}}_k^{m_1} = \mathbf{w}_k^{m_1}, k \neq j, k \in \mathcal{G}_\omega$ and $\widetilde{\mathbf{w}}_j^{m_1} = \mathbf{w}_j^{m_2}$.
4 Evaluate (MIOCP) with fixed $\widetilde{\omega}^{m_1} = \varphi(\widetilde{\mathbf{w}}^{m_1}) \rightarrow \widetilde{\Phi}_{m_1}$.
5 **if** $\widetilde{\Phi}_{m_1} \leq \Phi_{m_1}$ **then**
6 Set $\mathbf{w}_j^{m_1} = \widetilde{\mathbf{w}}_j^{m_1}$ and $\Phi_{m_1} = \widetilde{\Phi}_{m_1}$.
7 **end**
8 **end**
9 **end**
10 **return**: $\Phi(\varphi(\mathbf{w}^{\text{rec}})) := \min_{\text{milp} \in S^{\text{CIA}}} \Phi(\varphi(\mathbf{w}^{\text{milp}}))$.

every $\mathbf{w}^{m_1}$, it is advisable to use just the $\mathbf{w}^{m_2}$ solution for swapping with the so far lowest objective value. Fig. 1 illustrates an example recombination step for the pairs (CIA,SCIA) and (SCIA,CIA).

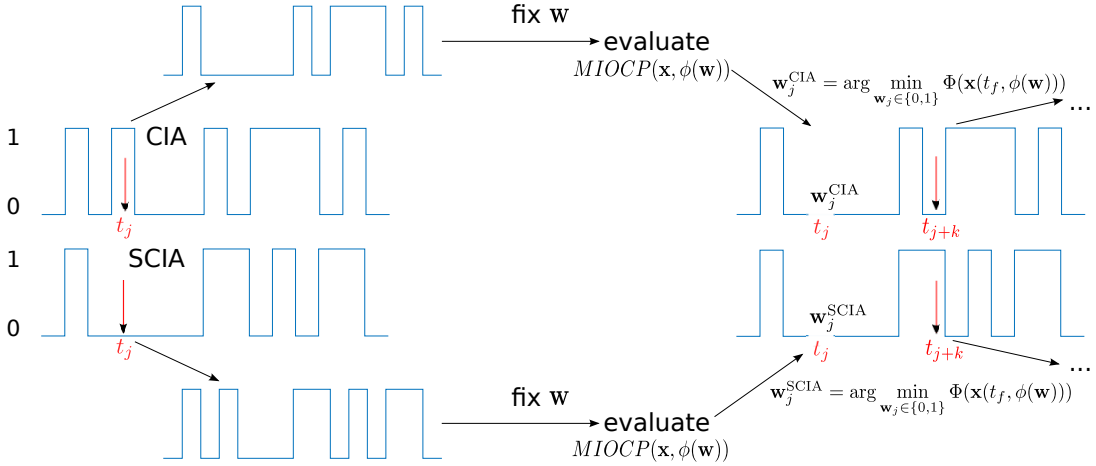

Figure 1. Visualization of GreedyTime algorithm. Two candidate solutions, here from CIA and SCIA, are used to construct new candidates. An enumeration between 0 and 1 is performed at all times t_j when the input vectors differ. Next, we fix the two candidate solutions $\mathbf{w}$ and evaluate (MIOCP) for both vectors. The resulting objective function values are compared with its previous values and the binary $\mathbf{w}_j$-values with the lower objective value are fixed in the candidate solutions. We repeat this procedure on the next grid point with unequal candidate solutions.

Remark 6. (GreedyTime speed up)

(1) The evaluation in Line 4 in iteration j can be accelerated by reusing the evaluation for the same solution from the previous iteration $j-1$ for the interval $[t_0, t_j]$.

(2) If a MILP solution m_1 differs from two MILPs m_2, m_3 with identical binary solutions vectors $\mathbf{w}_j^{m_2} = \mathbf{w}_j^{m_3}$, it is sufficient to simulate its recombination with only one of the two.

Remark 7. (GreedyTime modifications)

(1) The outer loop in Algorithm 2 can be also applied backward in time. We name the backward version *GreedyTimeBackward*.
(2) Instead of looping over all intervals, we may consider only singular arcs, i.e., the intervals where $\epsilon < a_{i,j} < 1 - \epsilon$ holds, with a certain threshold $\epsilon > 0$.
(3) *Greedy-cost-to-go*: Assume we have calculated the adjoints $\lambda_{j,k},\ k \in [n_{\mathrm{x}}],\ t_j \in \mathcal{G}_\omega$ of the state equations of (OCP). Then, re-sort $\mathcal{G}_\omega$ in descending order according to $\sum_{k\in[n_{\mathrm{x}}]} |\lambda_{j,k}|,\ t_j \in \mathcal{G}_\omega$. This results in a new ordered grid $\mathcal{G}_\omega^\lambda$ to be applied to Algorithm 2.

5.2. Singular arc recombination

Algorithm 3 aims at recombining singular arcs of the different MILP solutions. Usually, when $\mathbf{a}$ is binary or almost binary for a certain grid point, $\mathbf{w}$ should attain these binary values as well - regardless of the MILP choice. Therefore we partition $\mathcal{G}_\omega$ into singular and binary arcs.

Definition 9. ($\mathcal{G}_{arc}$, $\mathcal{G}_{bin}$) Let the union $\mathcal{G}_{arc} \cup \mathcal{G}_{bin}$ be a partition of $\mathcal{G}_\omega$ and both sets be of minimum cardinality. Elements $arc \in \mathcal{G}_{arc}$ consist of consecutive time points, i.e. $arc = \{t_j, t_{j+1}, \ldots, t_l\},\ j, l \in [M]$, for which the relaxed control satisfies $\mathbf{a}_k \in [\epsilon, 1-\epsilon]^{n_\omega}, \epsilon > 0,\ \forall\ t_k \in arc$. Sets of consecutive time points where the relaxed control takes values smaller ϵ or larger than $1 - \epsilon$ are summarized in $\mathcal{G}_{bin}$. We write $n_{arc} := |\mathcal{G}_{arc}|$ for the number of singular arcs and $\mathbf{w}_{arc} = \{\mathbf{w}_k\}_{k\in arc},\ arc \in \mathcal{G}_{arc}$ and accordingly $\mathbf{w}_{bin},\ bin \in \mathcal{G}_{bin}$.

In Algorithm 3 we set the temporary solution $\mathbf{w}^{tmp}$ on the binary arcs to the rounded relaxed solution for these binary arcs in Line 1. Then we test every possible variation (Line 2) of the different MILP solutions on the singular arcs to fill up the singular arcs of the temporary solution $\mathbf{w}^{tmp}$ (Line 3) and evaluate its objective value in Line 4. In case a recombination has a lower objective value than the so far best solution, it will be saved as so far best solution in Lines 5–8. Each variation $var \in (S^{\mathrm{CIA}})^{n_{arc}}$ consists of a specific MILP solution choice on each $arc \in \mathcal{G}_{arc}$, which is denoted by $var(arc) = milp \in S^{\mathrm{CIA}}$.

Remark 8. For computational effort, one has to take care of $|S^{\mathrm{CIA}}|^{n_{\mathrm{arc}}}$. Therefore, we choose for our numerical tests a small subset of MILPs out of S^{CIA} and instances with few singular arcs. In case of more than 4 singular arcs, Algorithm 3 may be modified to be greedy, applying the idea of Algorithm 2 on arcs instead of single grid points. Furthermore, if two MILP solutions are identical on a singular arc, only one has to be considered for the variations. Our algorithmic implementation checks if there are singular arcs with no unequal binary vectors and excludes these arcs from recombination.

Remark 9. The singular arc recombination supports an objective value that is at least as good as the previous found via the $MILP$s. However, there is no framework yet to quantify these possible improvements in terms of new rounding errors of the objective.

Algorithm 3: Singular arc block recombination heuristic for finding improved $\mathbf{w}$ variables

Input : Control grid $\mathcal{G}_\omega$, optimal relaxed variable vectors $\mathbf{a}$, binary variable vectors $\mathbf{w}^{\text{milp}}$, $milp \in S^{\text{CIA}}$, corresponding objectives $\Phi(\varphi(\mathbf{w}^{\text{milp}}))$, sets $\mathcal{G}_{arc}$, and $\mathcal{G}_{bin}$.

Set $\mathbf{w}_{bin}^{tmp} = \lfloor \mathbf{a}_{bin} + \epsilon \rfloor$, $\forall\, bin \in \mathcal{G}_{bin}$ and $\Phi_{rec} = \infty$.
for $var \in (S^{CIA})^{n_{arc}}$ **do**
  Set $\mathbf{w}_{arc}^{tmp} = \mathbf{w}_{arc}^{var(arc)}$, $\forall\, arc \in \mathcal{G}_{arc}$.
  Evaluate (MIOCP) with fixed $\omega^{tmp} := \varphi(\mathbf{w}^{tmp}) \to \Phi_{tmp}$.
  if $\Phi_{tmp} < \Phi_{rec}$ **then**
    Set $\Phi_{rec} = \Phi_{tmp}$.
    Set $\mathbf{w}^{rec} = \mathbf{w}^{tmp}$.
  end
end
return: $\mathbf{w}^{\text{rec}}$ together with Φ_{rec}.

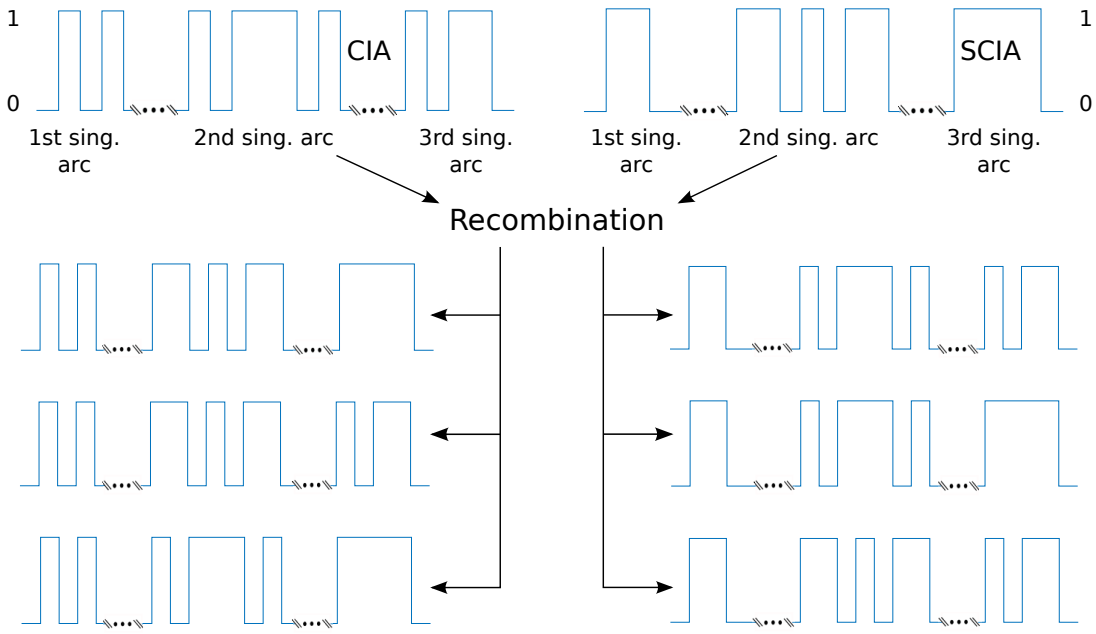

Figure 2. Visualization of singular arc block recombination heuristic for two MILP solution vectors with three singular arcs. We generate every variation from singular arcs and candidate solutions and evaluate (MIOCP) for each of the found variations. The minimal objective value of all variations represents the heuristic output.

6. Computational results

6.1. Software implementation and instances

We implemented Algorithm 1 in `AMPL` [10] using the code `ampl_mintoc`, which is a modeling framework for handling optimal control problems. It features different discretization schemes of ODEs, though we used only a Radau collocation from [3]. The tool is advantageous for our purpose, since it includes automatic differentiation, interfaces to MILP solvers, and its problem formulation stays close to mathematics.

Also AMPL provides the dual variables λ. Throughout the numerical study, we applied Gurobi 8.1 as MILP solver and IPOPT 3.12.4 as NLP solver to solve the discretized (OCP). We assumed that the choice of the MILP solver has little influence on solution quality and verified this by testing also with CPLEX 12.9. We tested our algorithms also with CasADi and received similar results as with ampl_mintoc. All results were obtained on a workstation with 4 Intel i5-4210U CPUs (1.7 GHz) and 7.7 GB RAM.

We included MIOCPs from the benchmark collection site mintoc.de [37] in our numerical study, which we specify further in the following subsections. For these problems, we chose a differential states discretization with N intervalls such that the objective value differs only to the 5th decimal place with respect to a finer discretization for constant M. Afterwards we varied M with fixed N in order to create several instances. For further details we refer to Appendix A. The problems involving path or terminal constraints might result in infeasible solutions after solving the binary approximation problem and solving the MIOCP with fixed binary controls. To this end, we relaxed these constraints and applied a merit function that penalizes constraint violation as part of the objective with sufficiently high penalty factor.

6.2. Scaled combinatorial integral approximation

We hypothesized that the MILPs based on scaled combinatorial integral approximation performed the best on instances where the binary control enters the control dependent right-hand side terms $\mathbf{f}_i$ of the ODE in an affine way, i.e.

$$\dot{\mathbf{x}} = \mathbf{f}_0(t, \mathbf{x}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) c_i, \qquad \text{for a.e. } t \in \mathcal{T}, \tag{20}$$

where $c_i \in \mathbb{R}$. If $\mathbf{f}_i$ depends on $\mathbf{x}(t)$, it may change rapidly over time resulting in possibly inaccurate ω solutions, since we only have the discretized state trajectory $\mathbf{x}(t)$ value at hand. We identified the MIOCPs "Double tank (Multimode)" and "Lotka Volterra (absolute fishing variant)" as candidate problems with the above right-hand side structure and calculated their solutions for different discretizations and both with and without the combinatorial constraint (12), see Appendix A for details. The results are presented in Fig. 3.

We chose to evaluate the MIOCP solutions according to the distance in the $\|\cdot\|_\infty$-norm of the differential state trajectories corresponding to either binary or relaxed controls. The theoretical justification of the CIA decomposition is built on this distance, as pointed out in Section 4 and, particularly, the proximity of objective values and constraint satisfaction follows as derived in Section 4.5. The performance plot shows that the differential state trajectories based on the (SCIA1) and (SCIAmax) solutions are significantly closer to the relaxed solution compared with their CIA counterparts. There are hardly differences between $\|\cdot\|_1$- and $\|\cdot\|_\infty$-norm results, although a tendency can be detected of better performing $\|\cdot\|_\infty$-variants.

We examined whether the (SCIA1) and (SCIAmax) are outperformed by (CIA1) or (CIAmax), if the state trajectory distance is measured in $\|\cdot\|_1$-norm or if the objective value deviation to the relaxed solution is taken into account, but the SCIA variants remained the clear winner. Analogously, the result remains similar, when comparing the algorithms solely on instances (not) including combinatorial constraints.

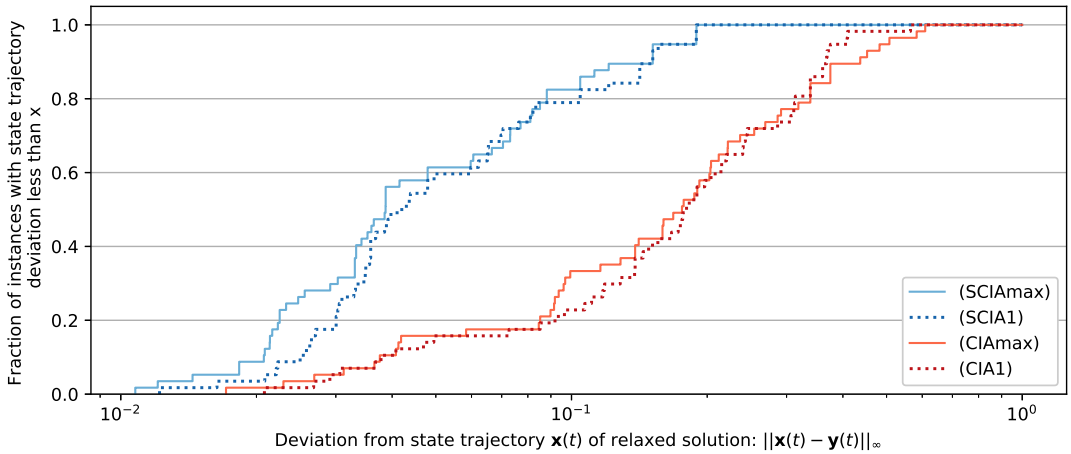

Figure 3. Performance profile comparing the deviation of differential states based on SCIA and CIA solutions. Relaxed solutions are shown in maximum norm and log-scale. The results are based on the instances "Double tank (Multimode)" and "Lotka Volterra (absolute fishing variant)" from the mintoc.de benchmark library. Using (SCIA1) or (SCIAmax) can improve the performance of the CIA decomposition significantly.

6.3. λ-combinatorial integral approximation

We derived (λCIA) as approximation of the cost-to-go function difference to the relaxed solution. Since this approximation is linear, the standard (CIA) approach is more suitable on most of the (nonlinear) MIOCPs. We postulated that the situation is different, when a regularization term enters the objective function, accounting for the cost of activating binary controls in the form of, e.g.,

$$\Phi(\mathbf{x}(t_\mathrm{f})) + \int_{t\in\mathcal{T}} \sum_{i=1}^{n_\omega} \omega_i(\tau) c_i \ \mathrm{d}\tau,$$

where $c_i \in \mathbb{R}$. The problem "Quadrotor (binary variant)" includes a cost function, where the controls enter in the above form, so that we used it with different discretizations and both with and without the combinatorial constraint (12) for comparing the (λCIA) solutions with the ones obtained via (CIA). We present in Fig. 4 the computation results. In contrast to the previous section, we compared here the objective deviations from the relaxed solution in percentage, since the λ-combinatorial integral approximation aims directly at improving the objective values. However, we remark that the latter algorithm performed worse than (CIA1) and (CIAmax), if the distance to the relaxed solution is measured in differential state space. The performance plot shows that (λCIA) provides solutions with improved objective value on some instances, but on many others it does not. Since (λCIA) turned out to provide even weaker approximations of the relaxed solutions for other MIOCPs, as will be shown in Section 6.5, we do not recommend to use it in general as a single MILP approximation step. It serves, however, as beneficial candidate solution for recombination and might be useful for not yet explored problem classes.

6.4. Backwards accumulating constraints

Our hypothesis for the MILP variants based on backwards accumulated constraints is that they are beneficial if the MIOCP involves terminal equality constraints on

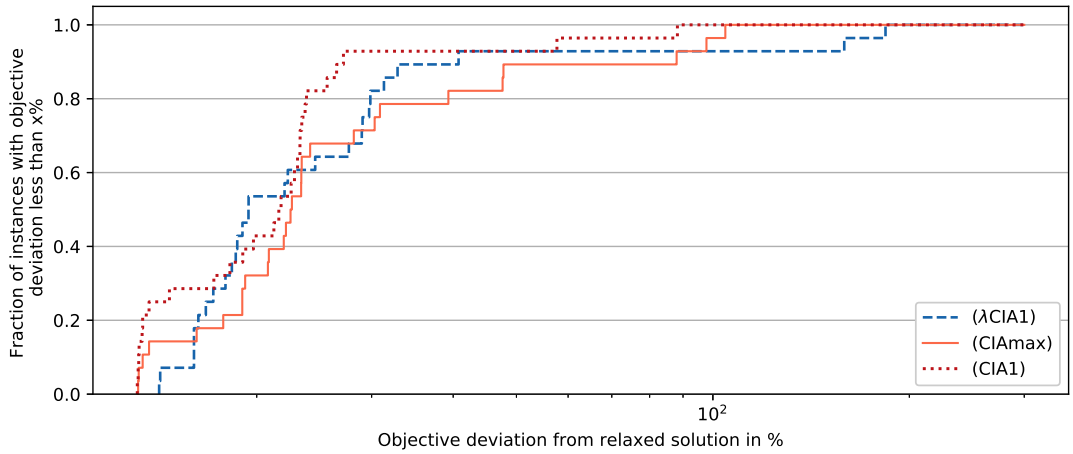

Figure 4. Performance profile comparing objective deviation from the relaxed solution in percentage and log-scale of λ-CIA and CIA solutions. The results are based on the instance "Quadrotor (binary variant)" from the mintoc.de benchmark library. (λCIA) appeared to provide no clear improvement compared with the (CIA) solutions.

the differential states. The standard (CIAmax) approach may construct a solution that does not satisfy this constraint, because the deviation to the relaxed solution can become large. However, the direct incorporation of terminal constraints into the MIOCP may lead to numerical difficulties already for the relaxed problem, so that we decided to deal with soft constraints meaning that we introduce slack variables in order to penalize a deviation of the the differential states from a desired terminal value. We identified the MIOCP "Lotka Volterra (terminal constraint violation)" as candidate problem and calculated its solutions for different discretizations and both with and without the combinatorial constraint (12). Fig. 5 illustrates the objective deviation from the relaxed solution in percentage of (CIA) and its backward variant solutions.

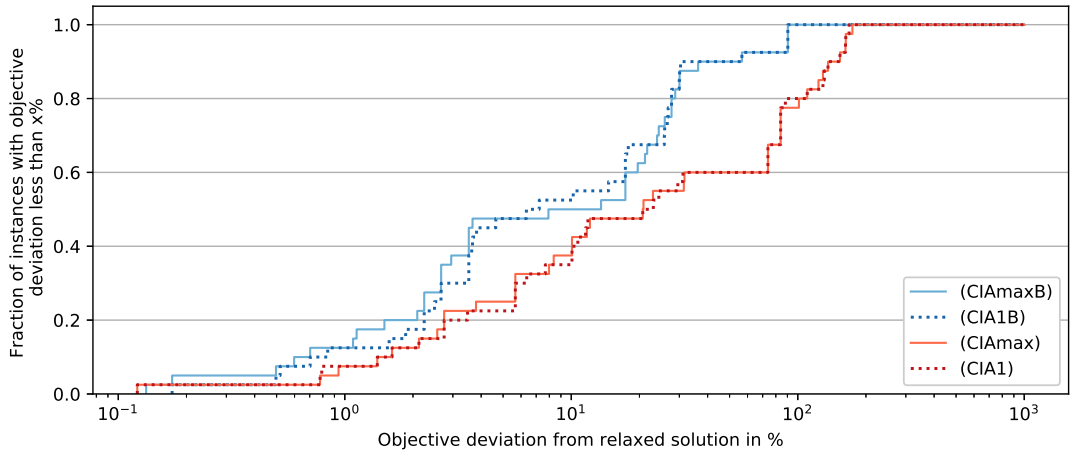

Figure 5. Performance profile comparing objective deviation from the relaxed solution in percentage and log-scale of (CIA) and its backward variant solutions. The results are based on the instance "Lotka Volterra (terminal constraint violation)" from the mintoc.de benchmark library. Using (CIA1B) or (CIAmaxB) can improve the performance of the CIA decomposition significantly.

We chose the objective deviation as performance measure for our comparison study, since the objective accounts for a violation of the terminal constraints via a slack

variable penalty term. The graphs of (CIAmaxB) and (CIA1B) indicate that their according MIOCP solutions yield clearly lower objective values than their forward CIA counterparts. We observed that this result seems to be independent from the chosen norm, since the performance differences of (CIAmaxB) to (CIA1B) are neglectable.

6.5. Recombination heuristics

We used the MILP solutions by (CIAmax), (CIA1), (SCIAmax), (λCIA1) and (CIAmaxB) as a base for running the recombination heuristics on a set of 13 MIOCPs from the benchmark collection site `mintoc.de` with different discretizations (see Appendix A for details). The box plot in Figure 6 summarizes the numerical results with respect to objective deviation of each algorithm to the relaxed solution in percentage.

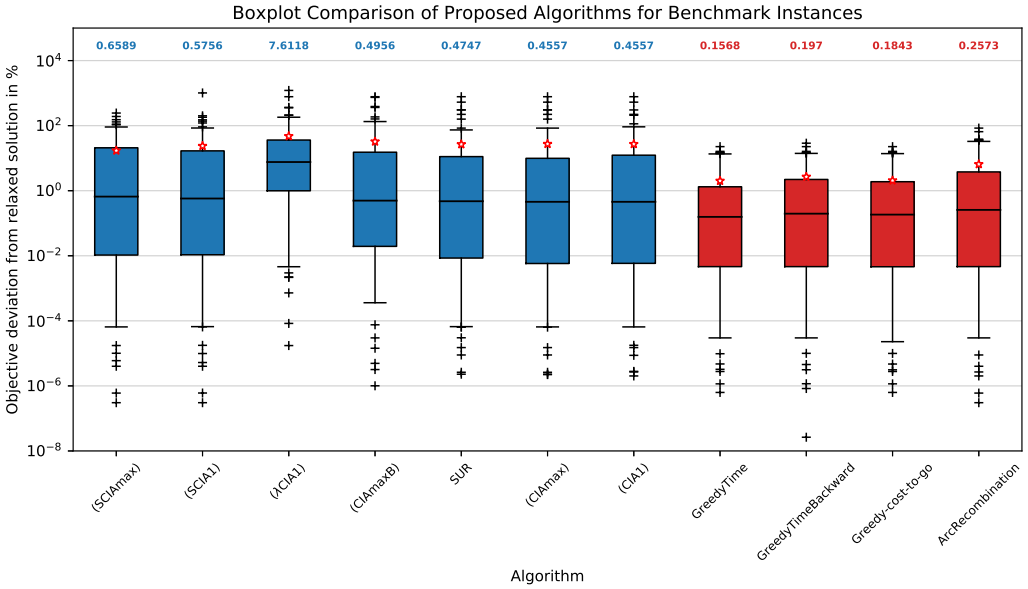

Figure 6. Box plot comparing objective deviation from the relaxed solution in percentage and log-scale of several MILP (marked in blue) and recombination heuristic (marked in red) solutions. The results are based on instances from the mintoc.de benchmark library. The box borders are 1/4 and 3/4-quantiles, whereas the whiskers represent 1/20 and 19/20-quantiles. We visualize the median values by black lines in the box and additionally display them numerically above the box. We represent the average values of the respective algorithms by red asterisks and the outliers by black crosses. The boxes of recombination strategies are shifted towards lower objective values compared with (CIA) algorithms and, thus, can improve the CIA decomposition performance significantly.

The boxes including median values of the SCIA and backwards approaches appear to be slightly higher, respectively their mean values and their outliers a bit lower, than the ones of the (CIA) MILPs. The numerical study revealed several instances, where (SCIAmax) or (SCIA1) run into a binary solution with active controls on some intervals with relaxed values close to zero. Under the assumption that the combinatorial approximation is done mainly on singular arcs, these cases might be called *degenerated*. We experienced underperforming objective values for SCIA in case of degenerated controls, which explains some of the underperforming instances. We conclude that (SCIA1), (SCIAmax) and (CIAmaxB) should be used with caution. For specific problem classes, as shown in the previous chapters, they can be very helpful. Here we have not specifically selected the problems, and on this general problem class there is no guarantee that these algorithms do provide any real improvement.

The solutions of (λCIA1) clearly underperform, but we stress as mentioned in Section 6.3 their importance for recombination. As a comparative calculation, we have also computed the solutions based on SUR and see that they provide similarly good, albeit somewhat worse, solutions compared with (CIA1) and (CIAmax). Note that, depending on the selected algorithm, some instances resulted in a deviation of more than 100%, which can be explained by highly penalized infeasible solutions of path or terminal-constrained problems.

The depicted recombination heuristics provide significantly better solutions than the MILP solutions. The median values are reduced by a factor of about 2 (ArcRecombination) to a factor of 3 (GreedyTime) in comparison with (CIA). The other characteristics such as mean values, box borders and outliers also reflect the improvements. Particularly noteworthy is the GreedyTime heuristic, which is robust against outliers as well as constructs on average solutions with small objective values. The ArcRecombination selects the solution of the MILP algorithms with the smallest objective value in the case of only one singular arc. Since many of the selected problems have only one singular arc, the box plot illustrates that this minimum over all MILPs can already provide a significant improvement.

6.6. Runtime evaluation

Figure 7 shows exemplarily the relationship between runtime and objective function values for the Lotka Volterra multimode problem with $N = 12000$ and varying M. We compare (CIAmax) values both with GreedyTime and the solutions obtained by the MINLP solver `Bonmin 1.8.6`. For a fair comparison, we run Bonmin with its four different main algorithms `B-BB`, `B-OA`, `B-QG` and `B-Hyb` and depicted the shortest runtime of these algorithms. Elapsed real time from `AMPL` represents runtime in our computations, since CPU time appeared to be very similar for our `Bonmin` calculations and `Gurobi` on the other hand is known to be a multi-threaded solver. First of all, the illustration shows that the objective spread to the relaxed solution vanished with increasing M - regardless of the selected approach. Second, CIA was for some instances already quite close to `Bonmin` ($M = 25, 50$) in terms of objective quality, so that GreedyTime cannot improve much. For other discretization with a considerable gap between CIA and Bonmin solution, GreedyTime could close most of this gap while being two orders of magnitude faster than `Bonmin`.

The average runtime over all instances for (CIAmax) was about a few seconds and increased slightly for (SCIAmax), see Appendix A.2. Gurobi needed on average more than one minute for the MILPs with 1-norm and thus considerably more time. For instances involving a fine discretization, the runtime increased enormously so that we set up a time limit of 30 minutes.

The greedy heuristics and the ArcRecombination are to be used cautiously, since an input of many MILPs leads to a high number of recombinations that have to be evaluated. ArcRecombination is relatively inexpensive and offers a solution that is at least as good as the best MILP. The algorithm is most beneficial in case of several singular arcs, in contrast to most applied problem instances where there is only one arc. The greedy algorithm variants are quite expensive (run times of up to 15 minutes), yet providing solutions with objective function values very close to those of the relaxed problem.

We could have accelerated the post-processing routines as described in Remark 6. A way to significantly reduce computation time of the MILPs is to apply Branch and

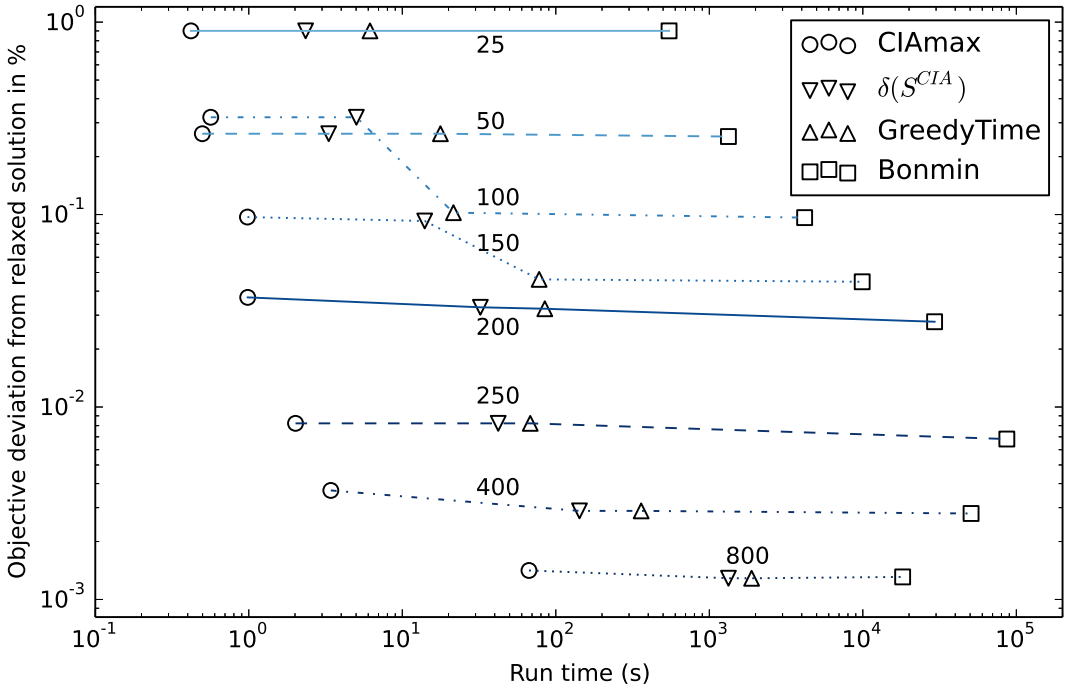

Figure 7. Log-plot of runtime and objective deviation from relaxed solution for the Lotka-Volterra multimode problem, $N = 12000$. Numbers indicate the corresponding number of control grid points M and by $\delta(S^{\mathrm{CIA}})$ we mean the minimal objective deviation over all MILP solutions. One observes the convergence of all approaches towards the lower bound provided by the relaxed solution, and a closure of the gap between CIA solution and Bonmin solution for a fixed discretization. GreedyTime is roughly two orders of magnitude slower than CIA, but faster than Bonmin.

Bound [26] or to use SUR for constructing approximate solutions. These algorithms are implemented in the open source software package *pycombina*[1] [4] and might be adapted to the scaled MILP case as part of a future study. If the binary controls enter linearly into the dynamics as in Equation 20, then the modification is straightforward, since only all differences $(a_i - w_i)$ have to be scaled with the factors c_i. Finally, run times of days or even weeks when it comes to MINLP solver cast a positive light on the proposed decomposition algorithm including recombination.

7. Summary and conclusion

We extended the decomposition approach based on combinatorial integral approximation [40], using multiple MILP formulations and recombination heuristics in an outer loop. At the price of additional MILP solutions and MIOCP evaluations, we obtain an improvement of the objective function value for every fixed control discretization grid.

A numerical study with benchmark problems shows that the novel MILP solutions indeed improve the existing CIA solutions in terms of objective value on specific problem classes. We conclude that the CIA decomposition can be reasonably modified for certain MIOC subproblem classes. Furthermore, the computational results for a set of non-specific MIOCPs resulted in a substantial improvement of the CIA solution

[1] see `https://github.com/adbuerger/pycombina`

through recombination strategies.

The main added value of this study is a decomposition algorithm that works much faster than `Bonmin`, but still offers qualitatively similar solutions and performs on average better than the existing (CIA) approach. The framework is open to extensions, both on the MILP and on the post-processing level.

Additional work is necessary to incorporate other constraints, such as vanishing constraints, to derive further tailored MILP formulations for specific problem classes and to develop numerical algorithms that generalize SUR and/or Branch and Bound algorithms to the various MILP formulations.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 647573) and from Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 314838170, GRK 2297 MathCoRe and SPP 1962.

References

[1] P. Abichandani, H.Y. Benson, and M. Kam. Multi-vehicle path coordination under communication constraints. In *American Control Conference*, pages 650–656, 2008.

[2] P. Belotti, C. Kirches, S. Leyffer, J.T. Linderoth, J. Luedtke, and A. Mahajan. Mixed-Integer Nonlinear Optimization. In Arieh Iserles, editor, *Acta Numerica*, volume 22, pages 1–131. Cambridge University Press, 2013.

[3] L.T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Series on Optimization. SIAM, 2010.

[4] A. Buerger, C. Zeile, Altmann-Dieses, S. A., Sager, and M. Diehl. Design, implementation and simulation of an mpc algorithm for switched nonlinear systems under combinatorial constraints. *Process Control*, 81:15–30, 2019.

[5] J. Burgschweiger, B. Gnädig, and M.C. Steinbach. Optimization Models for Operative Planning in Drinking Water Networks. Technical Report ZR-04-48, ZIB, 2004.

[6] J. Burgschweiger, B. Gnädig, and M.C. Steinbach. Nonlinear Programming Techniques for Operative Planning in Large Drinking Water Networks. *The Open Applied Mathematics Journal*, 3:1–16, 2009.

[7] M. Buss, M. Glocker, M. Hardt, O. v. Stryk, R. Bulirsch, and G. Schmidt. *Nonlinear Hybrid Dynamical Systems: Modelling, Optimal Control, and Applications*, volume 279. Springer-Verlag, Berlin, Heidelberg, 2002.

[8] Alina I Doban and Mircea Lazar. A switched systems approach to cancer therapy. In *2015 European Control Conference (ECC)*, pages 2718–2724. IEEE, 2015.

[9] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal Control of Switching Times in Switched Dynamical Systems. In *Proceedings of the 42nd IEEE Concference of Decision and Control*, 2003.

[10] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, 2002.

[11] A. Fügenschuh, M. Herty, A. Klar, and A. Martin. Combinatorial and Continuous Models for the Optimization of Traffic Flows on Networks. *SIAM Journal on Optimization*, 16(4):1155–1176, 2006.

[12] M. Gerdts. Solving mixed-integer optimal control problems by Branch&Bound: A case study from automobile test-driving with gear shift. *Optimal Control Applications and Methods*, 26:1–18, 2005.

[13] M. Gerdts. A variable time transformation method for mixed-integer optimal control problems. *Optimal Control Applications and Methods*, 27(3):169–182, 2006.
[14] M. Gerdts. *Optimal Control of ODEs and DAEs*. De Gruyter, 2012.
[15] M. Gerdts and S. Sager. Mixed-Integer DAE Optimal Control Problems: Necessary conditions and bounds. In L. Biegler, S.L. Campbell, and V. Mehrmann, editors, *Control and Optimization with Differential-Algebraic Constraints*, pages 189–212. SIAM, 2012.
[16] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, 2009.
[17] David E Goldberg. Genetic algorithms in search, optimization, and machine learning. 1989.
[18] S. Göttlich, M. Herty, C. Kirchner, and A. Klar. Optimal control for continuous supply network models. *Networks and Heterogenous Media*, 1(4):675–688, 2007.
[19] Simone Göttlich, Falk M Hante, Andreas Potschka, and Lars Schewe. Penalty alternating direction methods for mixed-integer optimal control with combinatorial constraints. *arXiv preprint arXiv:1905.13554*, 2019.
[20] Simone Göttlich, Andreas Potschka, and Ute Ziegler. Partial outer convexification for traffic light optimization in road networks. *SIAM Journal on Scientific Computing*, 39(1):B53–B75, 2017.
[21] M. Gugat, M. Herty, A. Klar, and G. Leugering. Optimal Control for Traffic Flow Networks. *Journal of Optimization Theory and Applications*, 126(3):589–616, 2005.
[22] Falk M. Hante. Relaxation methods for hyperbolic PDE mixed-integer optimal control problems. *Optimal Control Applications and Methods*, 38(6):1103–1110, 2017. oca.2315.
[23] F.M. Hante and S. Sager. Relaxation Methods for Mixed-Integer Optimal Control of Partial Differential Equations. *Computational Optimization and Applications*, 55(1):197–225, 2013.
[24] E. Hellström, M. Ivarsson, J. Aslund, and L. Nielsen. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice*, 17:245–254, 2009.
[25] M. Jung. *Relaxations and Approximations for Mixed-Integer Optimal Control*. PhD thesis, University Heidelberg, 2013.
[26] M. Jung, C. Kirches, and S. Sager. On Perspective Functions and Vanishing Constraints in Mixed-Integer Nonlinear Optimal Control. In M. Jünger and G. Reinelt, editors, *Facets of Combinatorial Optimization – Festschrift for Martin Grötschel*, pages 387–417. Springer Berlin Heidelberg, 2013.
[27] M. Jung, G. Reinelt, and S. Sager. The Lagrangian Relaxation for the Combinatorial Integral Approximation Problem. *Optimization Methods and Software*, 30(1):54–80, 2015.
[28] Y. Kawajiri and L.T. Biegler. A Nonlinear Programming Superstructure for Optimal Dynamic Operations of Simulated Moving Bed Processes. *I&EC Research*, 45(25):8503–8513, 2006.
[29] C. Kirches, F. Lenders, and P. Manns. Approximation properties and tight bounds for constrained mixed-integer optimal control. *Optimization Online*, April 2016. (submitted to SIAM Journal on Control and Optimization).
[30] C. Kirches, S. Sager, H.G. Bock, and J.P. Schlöder. Time-optimal control of automobile test drives with gear shifts. *Optimal Control Applications and Methods*, 31(2):137–153, March/April 2010.
[31] Thorsten Koch, Benjamin Hiller, Marc E. Pfetsch, and Lars Schewe, editors. *Evaluating Gas Network Capacities*. SIAM-MOS series on Optimization. SIAM, 2015.
[32] H.W.J. Lee, K.L. Teo, L.S. Jennings, and V. Rehbock. Control Parametrization Enhancing Technique for Optimal Discrete-Valued Control Problems. *Automatica*, 35(8):1401–1407, 1999.
[33] P. Manns and C. Kirches. Improved regularity assumptions for partial outer convexification of mixed-integer pde-constrained optimization problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 2019.
[34] Maik Ringkamp, Sina Ober-Blöbaum, and Sigrid Leyendecker. On the time transforma-

tion of mixed integer optimal control problems using a consistent fixed integer control function. *Mathematical Programming*, 161(1):551–581, 2017.
[35] S. Sager. *Numerical methods for mixed–integer optimal control problems*. Der andere Verlag, Tönning, Lübeck, Marburg, 2005. ISBN 3-89959-416-9.
[36] S. Sager. Reformulations and Algorithms for the Optimization of Switching Decisions in Nonlinear Optimal Control. *Journal of Process Control*, 19(8):1238–1247, 2009.
[37] S. Sager. A benchmark library of mixed-integer optimal control problems. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 631–670. Springer, 2012.
[38] S. Sager, H.G. Bock, and M. Diehl. The Integer Approximation Error in Mixed-Integer Optimal Control. *Mathematical Programming A*, 133(1–2):1–23, 2012.
[39] S. Sager, M. Claeys, and F. Messine. Efficient upper and lower bounds for global mixed-integer optimal control. *Journal of Global Optimization*, 61(4):721–743, 2015.
[40] S. Sager, M. Jung, and C. Kirches. Combinatorial Integral Approximation. *Mathematical Methods of Operations Research*, 73(3):363–380, 2011.
[41] S. Sager, G. Reinelt, and H.G. Bock. Direct Methods With Maximal Lower Bound for Mixed-Integer Optimal Control Problems. *Mathematical Programming*, 118(1):109–149, 2009.
[42] Sebastian Sager and Clemens Zeile. On mixed-integer optimal control with constrained total variation of the integer control. Technical report, 2019.
[43] Carla Seatzu, Daniele Corona, Alessandro Giua, and Alberto Bemporad. Optimal control of continuous-time switched affine systems. *IEEE Transactions on Automatic Control*, 51(5):726–741, 2006.
[44] C. Sonntag, O. Stursberg, and S. Engell. Dynamic Optimization of an Industrial Evaporator using Graph Search with Embedded Nonlinear Programming. In *Proc. 2nd IFAC Conf. on Analysis and Design of Hybrid Systems (ADHS)*, pages 211–216, 2006.
[45] Oliver Stein. Error bounds for mixed integer nonlinear optimization problems. *Optimization Letters*, 10(6):1153–1168, 2016.
[46] B. Stellato, S. Ober-Blbaum, and P. J. Goulart. Second-order switching time optimization for switched dynamical systems. *IEEE Transaction on Automatic Control*, 62(10):5407–5414, 2017.
[47] J. Till, S. Engell, S. Panek, and O. Stursberg. Applied Hybrid System Optimization: An Empirical Investigation of Complexity. *Control Engineering Practice*, 12:1291–1303, 2004.
[48] C. Zeile, N. Robuschi, and S. Sager. Mixed-integer optimal control under minimum dwell time constraints. Technical report, 2019.
[49] Feng Zhu and Panos J Antsaklis. Optimal control of hybrid switched systems: A brief survey. *Discrete Event Dynamic Systems*, 25(3):345–364, 2015.

Appendix A. Detailed Numerical Results

A.1. Problem Discretization Details

For generating the performance and box plots in Section 6, we applied Algorithm 1 on the following discretized problems:

"Lotka Volterra (absolute fishing variant)":
$N = 12000,\ M \in \{25, 50, 75, 80, 100, 120, 150, 160, 200\},\ \sigma_{\max} \in \{10, 20, \infty\}$,

"Quadrotor (binary variant)":
$N = 12000,\ M \in \{25, 50, 60, 80, 100, 150, 200, 300\},\ \sigma_{\max} \in \{4, 10, 20, \infty\}$,

"Lotka Volterra (terminal constraint violation)":
$N = 12000,\ M \in \{20, 30, 40, 50, 60, 100, 120, 200, 240, 300, 400, 600\}$,
$\sigma_{\max} \in \{4, 10, 20, \infty\}$,

"F-8 aircraft (AMPL variant)":
$N = 6000,\ M \in \{30, 40, 50, 60, 100, 120, 150, 200, 240, 300, 400, 500\}$,

"Egerstedt standard problem":
$N = 6000,\ M \in \{20, 30, 40, 60, 100, 120, 150, 200, 240, 300\}$,

"Double Tank":
$N = 18000,\ M \in \{25, 50, 100, 180, 250, 300, 360, 720\}$,

"Double Tank multimode":
$N = 12000,\ M \in \{20, 25, 50, 100, 200, 250, 300, 400, 600\},\ \sigma_{\max} \in \{10, 20, \infty\}$,

"Lotka Volterra fishing problem":
$N = 12000,\ M \in \{20, 30, 40, 60, 100, 120, 200, 300, 400, 600\}$,

"Lotka Volterra multi-arcs problem":
$N = 18000,\ M \in \{25, 50, 100, 150, 200, 250, 300, 400, 600\}$,

"Lotka Volterra multimode problem":
$N = 12000,\ M \in \{25, 50, 100, 150, 200, 250, 300, 400, 800\}$,

"Van der Pol Oscillator (binary variant)":
$N = 6000,\ M \in \{20, 30, 40, 50, 60, 100, 120, 150, 200, 300\}$,

"D'Onofrio chemotherapy model":
Scenario 1,2, and 3 with $N = 6000,\ M \in \{20, 30, 40, 50, 60, 100, 120, 150, 200, 300\}$,
Only $M \in \{20, 30, 60, 120\}$ for scenario 1, $M = 100$ for scenario 2, and $M \in \{40, 100\}$ for scenario 3 resulted in feasible relaxed solutions and were included.

"Catalyst Mixing problem":
$N = 3000,\ M \in \{10, 15, 20, 30, 50, 60, 75, 100, 120, 150\}$.

A.2. Average Performance Indicators and Individual Problem Results

Table A1. Comparison of mean values and standard deviation (σ) of objective deviation, switching values and runtime for different approaches. Objective deviation is given in percentage compared to relaxed objective and runtime describes elapsed real time.

Approach	obj. dev [%]	switches [#]	runtime [s]	σ(obj. dev)	σ(switches)	σ(runtime)
CIAmax	27.32	40.08	8.84	95.91	40.16	29.60
CIA1	27.08	39.54	106.11	93.60	38.99	385.77
SCIAmax	17.13	31.12	12.38	38.06	31.54	42.35
SCIA1	23.71	30.94	78.17	96.18	29.91	317.55
λCIA1	47.51	28.08	54.91	139.97	45.10	290.47
CIAmaxB	32.37	40.41	19.15	110.28	40.10	166.22
GreedyTime	2.06	33.36	106.26	4.27	34.47	133.61
GreedyTimeB	2.68	33.61	103.05	5.11	33.64	131.84
Greedy-Cost-to-go	2.01	34.05	117.24	4.24	34.09	172.88
ArcRecombination	6.53	35.34	11.26	13.55	37.01	37.12

Table A2. Results for the Lotka Volterra multimode problem with $N = 12000$ and varying M. The tables list objective values, differences to relaxed objective, number of switches, and runtime in seconds.

	(CIAmax)				(CIA1)			
M	Obj.	Diff. to rel.	S [#]	R [s]	Obj.	Diff. to rel.	S [#]	R [s]
25	1.84519	0.00920032	6	0.419997	1.84519	0.00920032	6	0.323492
50	1.83353	0.00189968	9	0.498163	1.83353	0.00189968	9	0.526022
100	1.83458	0.00470921	15	0.564993	1.83458	0.00470921	15	0.849123
150	1.83049	0.00129738	20	0.979946	1.83058	0.00138375	20	3.37327
200	1.8294	0.000412465	23	0.983907	1.8294	0.000412465	23	9.61383
250	1.82887	8.52473e-05	30	2.01582	1.82887	8.52473e-05	30	6.84566
300	1.82884	2.1597e-05	33	1.87382	1.82884	2.1597e-05	33	27.4496
400	1.82879	3.40892e-05	47	3.42292	1.82879	3.40892e-05	47	45.0224
800	1.82875	2.58672e-05	87	66.8739	1.82875	2.58672e-05	87	484.285
	(SCIAmax)				**(SCIA1)**			
25	1.84519	0.00920032	6	0.313487	1.84519	0.00920032	6	0.925208
50	1.83399	0.00235793	8	0.493533	1.83399	0.00235793	8	0.723298
100	1.91199	0.0821278	16	1.05474	1.91199	0.0821278	16	3.62938
150	1.8834	0.0542079	20	2.84568	1.8834	0.0542079	20	9.7413
200	1.86972	0.0407389	25	7.90383	1.86972	0.0407389	25	36.7948
250	1.82887	8.52473e-05	30	11.6632	1.82887	8.5189e-05	30	65.8286
300	1.82887	4.80446e-05	32	8.75161	1.82887	4.80449e-05	32	55.7173
400	1.82877	1.94567e-05	47	30.4913	1.82877	1.94577e-05	47	188.408
800	1.83859	0.00987316	88	233.701	1.8381	0.00937638	89	1479.19
	(λCIA1)				**(CIAmaxB)**			
25	1.84543	0.0094458	5	0.443169	1.87559	0.0395975	7	0.511785
50	1.84372	0.0120927	5	0.581385	1.84076	0.00912746	9	0.574335
100	1.8533	0.0234329	16	0.839147	1.8347	0.00483784	15	0.735999
150	1.85038	0.0211798	23	2.50497	1.83041	0.00121583	19	0.840867
200	1.83509	0.00610253	30	2.52277	1.82932	0.000336555	25	1.53587
250	1.8289	0.000119317	25	13.3181	1.82894	0.000159443	31	2.02886
300	1.8553	0.0264818	29	6.28425	1.82887	5.56473e-05	35	2.94341
400	2.07161	0.242853	128	12.5695	1.82878	2.53493e-05	47	5.77022
800	3.44174	1.61302	420	18.8157	1.82875	3.20174e-05	89	34.2567
	GreedyTime				**GreedyTimeBackward**			
25	1.84519	0.00920032	6	3.81442	1.84519	0.00920032	6	4.55248
50	1.83353	0.00189968	9	14.398	1.83353	0.00189968	9	14.5728
100	1.83059	0.000723242	15	16.5069	1.83117	0.00130419	13	16.7239
150	1.82956	0.000364781	19	63.9035	1.83	0.000802598	20	60.1375
200	1.82931	0.000326273	24	52.5325	1.82932	0.000336555	25	53.6014
250	1.82887	8.52473e-05	30	25.9954	1.82887	8.52473e-05	30	25.6038
300	1.82884	2.1597e-05	33	81.1383	1.82884	2.1597e-05	33	82.31
400	1.82877	1.94567e-05	47	217.31	1.82877	1.94567e-05	47	179.64
800	1.82874	2.35655e-05	87	553.42	1.82874	2.3582e-05	87	605.012
	ArcRecombination				**Greedy-cost-to-go**			
25	1.84519	0.00920032	6	0.6978	1.84519	0.00920032	6	3.89079
50	1.83353	0.00189968	9	0.5322	1.83353	0.00189968	9	14.4531
100	1.83458	0.00470907	15	0.3819	1.83318	0.00331505	17	27.2192
150	1.83041	0.00121583	19	0.8785	1.82965	0.00045483	17	67.8054
200	1.82932	0.000336555	25	0.6278	1.82931	0.000326273	24	60.569
250	1.82887	8.52473e-05	30	0.6946	1.82887	8.52473e-05	30	25.6708
300	1.82884	2.1597e-05	33	0.9826	1.82884	2.1597e-05	33	103.187
400	1.82877	1.94567e-05	47	0.5933	1.82877	1.94567e-05	47	302.851
800	1.82874	2.3582e-05	87	0.7660	1.82874	2.35652e-05	87	1166.96