

Article

Combinatorial Integral Approximation Decompositions for Mixed-Integer Optimal Control

Clemens Zeile , Tobias Weber and Sebastian Sager 

Department of Mathematics, Otto-von-Guericke-Universität Magdeburg, 39106 Magdeburg, Germany; tobias.weber@ovgu.de (T.W.); sager@ovgu.de (S.S.)

* Correspondence: clemens.zeile@ovgu.de

Abstract: Solving mixed-integer nonlinear programs (MINLPs) is hard from both a theoretical and practical perspective. Decomposing the nonlinear and the integer part is promising from a computational point of view. In general, however, no bounds on the objective value gap can be established and iterative procedures with potentially many subproblems are necessary. The situation is different for mixed-integer optimal control problems with binary variables 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. In this article, 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 recombine promising candidate solutions for the binary control functions in the original problem. We present the extensions for ordinary differential equations-constrained problems. These extensions 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. We implemented all algorithms and subproblems in AMPL for a proof-of-concept study. Numerical results show the improvement compared to the 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



Citation: Zeile, C.; Weber, T.; Sager, S. Combinatorial Integral Approximation Decompositions for Mixed-Integer Optimal Control. *Algorithms* **2022**, *15*, 121. <https://doi.org/10.3390/a15040121>

Academic Editors: Andrea Serani and Riccardo Pellegrini

Received: 17 February 2022

Accepted: 30 March 2022

Published: 31 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. General Context

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 (MIOCPs) 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 control functions, or equivalently, one can switch instantly between several differential equations [1–4]. This problem class is ubiquitous in various application areas, e.g., water, gas, traffic, and supply chain networks [5–11], distributed autonomous systems [12], processes in chemical engineering that involve valves [13,14], cardiac assist devices [15], or the choice of gears in automotive control [16,17]. We choose problems from a web-based MIOCP benchmark collection [18] to evaluate our algorithms.

Several families of methods have been proposed for solving MIOCPs. This comprehends *indirect* or *first-optimize-then-discretize* approaches [19], dynamic programming or Hamilton–Jacobi–Bellman equations [20], switching time optimization [16,21–25], and *direct*

or *first-discretize-then-optimize* approaches [3,26,27]. Surveys, references, and comparisons can be found in, e.g., [19,28–30].

1.2. Motivation

Our approach is based on previous ideas [27,31,32] 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 [33] for recent results on error bounds for rounding approaches and [34] for a MINLP survey), the particular setting with dependent (states) and independent (controls) variables allows the derivation of a priori bounds [35]. As specified in Section 4, the difference between state trajectories $\mathbf{x}(\cdot)$ and $\mathbf{y}(\cdot)$, which 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:

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

This motivates us 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 $\mathbf{y}(\cdot)$ by solving the initial value problem (IVP). The 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 an MILP [32] and can be solved either with generic MILP methods, with specifically tailored branch-and-bound methods [32], and in some cases even in linear time, with the sum-up rounding (SUR) method [27].

1.3. Review of the State of the Art

The idea of CIA decomposition has also been used in the context of hyperbolic partial differential equations [36,37], differential-algebraic equation systems [19], with additional combinatorial constraints [10,32], and based on discrete time formulations [10]. Recently, algorithms to solve CIA problems have been implemented into the software package *pycombina* [38].

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. *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., [39,40]. 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 [41]. 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.

Recently, the MIOCP class has been increasingly studied in the context of combinatorial constraints that couple over time. For example, as part of the CIA decomposition, the switching-cost aware rounding problem (SCARP) has been proposed to solve it [42,43]. In addition, the CIA decomposition has been studied in the context of multibang and

total variation regularization [44,45]. Switching constraints have been also investigated for mixed-integer partial differential equation optimal control problems [46,47]. Switching costs can also be included into the switching time optimization approach based on cardinality constraints [48].

1.4. 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 solving 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.

1.5. Outline of the Article

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 [18]. Finally, we conclude the article in Section 7, where we also summarize our findings.

2. Problem Class, Definitions, and Main Algorithm

We denote the considered time horizon by $\mathcal{T} := [t_0, t_f] \subset \mathbb{R}$, and write “for a.a. $t \in \mathcal{T}$ ” for all $t \in \mathcal{T}$, except on a set of measure zero. Let $[n] := \{1, \dots, n\}$, $[n]_0 := 0 \cup [n]$, for $n \in \mathbb{N}$. 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).

$$\begin{aligned} \inf_{\mathbf{x}, \omega} \quad & \Phi(\mathbf{x}(t_f)) & (2a) \\ \text{s.t.} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}_0(t, \mathbf{x}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) \mathbf{f}_i(t, \mathbf{x}(t)), \quad \text{for a.a. } t \in \mathcal{T}, & (2b) \\ & \mathbf{x}(t_0) = \mathbf{x}_0, & (2c) \\ & 1 = \sum_{i=1}^{n_\omega} \omega_i(t) \quad \text{for a.a. } t \in \mathcal{T}, & (2d) \\ & \omega(t) \in \{0, 1\}^{n_\omega} \quad \text{for a.a. } t \in \mathcal{T}, & (2e) \\ & \mathbf{o} \leq \mathbf{c}(t, \mathbf{x}(t)) \quad \text{for a.a. } t \in \mathcal{T}. & (2f) \end{aligned}$$

We minimize a Mayer term $\Phi \in C^1(\mathbb{R}^{n_x}, \mathbb{R})$ over differential states $\mathbf{x} \in \mathcal{W}^{1,\infty}(\mathcal{T}, \mathbb{R}^{n_x})$ 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 one-hot encoding (1hot) constraint (2d), a drift term \mathbf{f}_0 , and $n_\omega \geq 2$ functions \mathbf{f}_i , ref. [27], both out of $C^0(\mathbb{R}^{n_x+1}, \mathbb{R}^{n_x})$. We assume fixed initial values $\mathbf{x}_0 \in \mathbb{R}^{n_x}$ for the differential states. The functions $\mathbf{c} \in C^1(\mathbb{R}^{n_x}, \mathbb{R}^{n_c})$ model nonlinear state inequalities.

In the following, we assume that (MIOCP) has an optimal solution, so that we can write “min” instead of “inf”. We refer to [19,29,49] 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) and (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 the solution of an initial value problem. We stress that there are only n_ω possible solutions for the integer part of the problem due to the constraint (2d).

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), where we substitute $\omega \in L^\infty(\mathcal{T}, \{0, 1\}^{n_\omega})$ for $\alpha \in L^\infty(\mathcal{T}, [0, 1]^{n_\omega})$.

The problem (OCP) in function space can be solved by different approaches, as mentioned above. For using 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. ($\mathcal{G}_\omega, \Delta, \varphi, \varphi^{-1}, \Omega, \Omega_M$) Let the ordered set $\mathcal{G}_\omega := \{t_0 < \dots < t_M = t_f\}$ denote a time grid with $\Delta_j := t_{j+1} - t_j$ and $\Delta_{max} := \max_j \Delta_j$ for $j \in [M - 1]_0$. We define the mapping:

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

using n_ω piecewise constant functions:

$$\alpha_i(t) := a_{i,j}, \quad i \in [n_\omega], \quad t \in [t_j, t_{j+1}), \quad j \in [M - 1]_0, \quad t_j \in \mathcal{G}_\omega.$$

A mapping in reverse direction:

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

is defined by extracting integrals on the grid \mathcal{G}_ω :

$$a_{i,j} := \frac{1}{\Delta_j} \int_{t_j}^{t_{j+1}} \alpha_i(\tau) d\tau, \quad i \in [n_\omega], \quad j \in [M - 1]_0, \quad t_j \in \mathcal{G}_\omega.$$

We denote integrality and (2d) using the sets:

$$\Omega := \left\{ \omega \in L^\infty(\mathcal{T}, \{0, 1\}^{n_\omega}) : 1 = \sum_{i=1}^{n_\omega} \omega_i(t) \text{ for a.a. } t \in \mathcal{T} \right\},$$

$$\Omega_M := \left\{ \mathbf{w} \in \{0, 1\}^{n_\omega \times M} : 1 = \sum_{i=1}^{n_\omega} w_{i,j} \text{ for } j \in [M - 1]_0 \right\}.$$

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

Definition 4 ($(\tilde{\lambda}, \tilde{f})$). Let \mathbf{x}^* be the optimal solution of (OCP). The evaluated right-hand side function terms $\{\tilde{f}_{i,j,k}\}_{k \in [n_x]}$ from (2b) are defined as the entries of:

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

We denote, by $\tilde{\lambda}_{j,k} \in \mathbb{R}$, $t_j \in \mathcal{G}_\omega$, $k \in [n_x]$, the discretized and evaluated dual variables of the ODE constraints (2b) in (OCP).

Our decomposition algorithm is based on the algorithmic choices of the sets S^{CIA} and S^{REC} , which we define next.

Definition 5 ($S^{\text{CIA}}, S^{\text{REC}}$). We introduce the set of CIA problems S^{CIA} as:

$$S^{\text{CIA}} := \{(\text{CIAMax}), (\text{CIA1}), (\text{CIAMaxB}), (\text{CIA1B}), (\lambda\text{CIA1}), (\lambda\text{CIA1B}), (\text{SCIAMax}), (\text{SCIA1}), (\text{SCIAMaxB}), (\text{SCIA1B})\},$$

where we define the specific CIA problems in the next section. For a subset $\tilde{S}^{\text{CIA}} \subseteq S^{\text{CIA}}$, we denote, with $n_{\text{CIA}} := |\tilde{S}^{\text{CIA}}|$, the number of different CIA problem formulations. Let the elements of \tilde{S}^{CIA} be numbered by $1, \dots, n_{\text{CIA}}$. Let the set S^{REC} of recombination mappings $F^{\text{rec}} \in S^{\text{REC}}$ be defined via:

$$F^{\text{rec}} : \prod_{k \in [n_{\text{CIA}}]} \Omega_N \rightarrow \Omega_N, \quad F^{\text{rec}}(\mathbf{w}^1, \dots, \mathbf{w}^{n_{\text{CIA}}}) \mapsto \mathbf{w}^{\text{rec}}, \quad (3)$$

where \mathbf{w}^k denotes the optimal solution of the problem $(\text{milp})^k \in \tilde{S}^{\text{CIA}}$.

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 (feasible) state trajectories and objective values. In lines 6–7, we create new candidate binary controls in several recombination heuristics based on the existing binary controls, which we evaluate as well (line 8). As a final step, we select the best-performing binary control as the solution in line 10.

Algorithm 1: Decomposition of (MIOCP).

Input : (MIOCP) instance, grid \mathcal{G}_ω , algorithmic choices of sets $\tilde{S}^{\text{CIA}} \subseteq S^{\text{CIA}}$ and $\tilde{S}^{\text{REC}} \subseteq S^{\text{REC}}$.

- 1 Solve (OCP) $\rightarrow \Phi^{\text{rel}}, \mathbf{x}^*, \mathbf{a}^*, \mathbf{a}^* = \varphi^{-1}(\mathbf{a}^*)$.
- 2 **for** $\text{milp} \in \tilde{S}^{\text{CIA}}$ **do**
- 3 Solve milp with MILP solver $\mathbf{a}^* \rightarrow \mathbf{w}^{\text{milp}}$;
- 4 Simulate (MIOCP) with fixed $\omega^{\text{milp}} := \varphi(\mathbf{w}^{\text{milp}}) \rightarrow \Phi_{\text{milp}}, \mathbf{x}$;
- 5 **end**
- 6 **for** $F^{\text{rec}} \in \tilde{S}^{\text{REC}}$ **do**
- 7 Create \mathbf{w}^{rec} using $F^{\text{rec}}(\mathbf{w}^{\text{milp}}), \Phi_{\text{milp}}$ from all $\text{milp} \in \tilde{S}^{\text{CIA}}$;
- 8 Simulate (MIOCP) with fixed $\omega^{\text{rec}} := \varphi(\mathbf{w}^{\text{rec}}) \rightarrow \Phi^{\text{rec}}, \mathbf{x}$;
- 9 **end**
- 10 Find $\text{opt} \in \tilde{S}^{\text{CIA}} \cup \tilde{S}^{\text{REC}}$ with $\Phi^{\text{opt}} = \min \left\{ \min_{\text{milp} \in \tilde{S}^{\text{CIA}}} \Phi^{\text{milp}}, \min_{F^{\text{rec}} \in \tilde{S}^{\text{REC}}} \Phi^{\text{rec}} \right\}$;
- 11 **return:** $(\mathbf{x}, \mathbf{w}, \Phi) = (\mathbf{x}^{\text{opt}}, \mathbf{w}^{\text{opt}}, \Phi^{\text{opt}})$;

The main idea of Algorithm 1 is to decouple controls and states. We approximate the relaxed control function \mathbf{a}^* 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 \tilde{S}^{CIA} and \tilde{S}^{REC} , which we discuss in Sections 3 and 5. A theoretical motivation and error bounds on the approximation quality are given in Section 4. Algorithm 1 is a generalization of the decomposition approach in [27,32], for which \tilde{S}^{REC} is empty and \tilde{S}^{CIA} contains only one CIA problem formulation.

3. Combinatorial Integral Approximation MILPs

In the following, we define the MILP formulations of CIA type for S^{CIA} in Algorithm 1. CIA, λCIA , and SCIA refer to different scalings (Section 3.1), “1” and “max” to different norms $\|\cdot\|$ (Section 3.2), and the presence of “B” to a reversal of time (Section 3.3).

3.1. Combinatorial Integral Approximation and Scaled Variants

As part of the approximation step, we aim at finding binary control values that are close to the relaxed values with respect to the accumulated difference over all grid points. The following definition specifies the so-far-applied (CIA) problem together with two novel variants. We let the vector norm $\|\cdot\|$ be unspecified here, but discuss applicable norms later.

Definition 6 ($\theta_{CIA}^*, \theta_{SCIA}^*, \theta_{\lambda CIA}^*$, cf. Definition 4.17 in [50]). Let \mathbf{a}^* be the given optimal control solution of (OCP), and let the evaluated model function values \tilde{f} and dual variables $\tilde{\lambda}$ be given as introduced in Definition 4. Consider a vector norm $\|\cdot\|$. We introduce the following optimization problems:

$$\theta_{CIA}^* := \min_{\mathbf{w} \in \Omega_N} \max_{j \in [M-1]_0} \left\| \sum_{l \in [j]} (\mathbf{a}_{\cdot,l}^* - \mathbf{w}_{\cdot,l}) \Delta_l \right\|, \tag{4}$$

$$\theta_{SCIA}^* := \min_{\mathbf{w} \in \Omega_N} \max_{j \in [M-1]_0} \left\| \sum_{l \in [j]} \sum_{i \in [n_\omega]} (a_{i,l}^* - w_{i,l}) \Delta_l \tilde{f}_{i,l} \right\|, \tag{5}$$

$$\theta_{\lambda CIA}^* := \min_{\mathbf{w} \in \Omega_N} \max_{j \in [M-1]_0} \left| \sum_{k \in [n_x]} \tilde{\lambda}_{j,k} \sum_{l \in [j]} \sum_{i \in [n_\omega]} (a_{i,l}^* - w_{i,l}) \Delta_l \tilde{f}_{i,l,k} \right|. \tag{6}$$

3.2. Norm Dependent MILP Formulation

By introducing the auxiliary variable θ , we can reformulate (4) and (5) with the maximum norm.

Definition 7. (CIAmax, SCIAmax) Let \mathbf{a}^* be the given optimal control solution of (OCP) and let the evaluated model function values \tilde{f} be given as introduced in Definition 4. We define (CIAmax) as:

$$\min_{\theta \geq 0, \mathbf{w} \in \Omega_M} \theta \tag{7a}$$

$$s.t. \quad \theta \geq \pm \sum_{j=0}^l (a_{i,j}^* - w_{i,j}) \Delta_j, \quad \text{for } i \in [n_\omega], \quad l \in [M-1]_0, \tag{7b}$$

and (SCIAmax) as:

$$\min_{\theta \geq 0, \mathbf{w} \in \Omega_M} \theta \tag{8a}$$

$$s.t. \quad \theta \geq \pm \sum_{j=0}^l \sum_{i=1}^{n_\omega} (a_{i,j}^* - w_{i,j}) \Delta_j \tilde{f}_{i,j,k}, \quad \text{for } k \in [n_x], \quad l \in [M-1]_0. \tag{8b}$$

We introduce the MILP analogue formulations for the Manhattan norm with auxiliary variables $s_{i,j} \geq 0, i \in [n_\omega], j \in [M-1]_0$. In this way, we specify the norm choices for Definition 6.

Definition 8. (CIA1, SCIA1, λCIA1) Consider \mathbf{a}^* as the given optimal control solution of (OCP). Let the evaluated model function values \tilde{f} and dual variables $\tilde{\lambda}$ be given as introduced in Definition 4. Based on the auxiliary variables $s_{i,l}$ we define (CIA1) as:

$$\min_{\theta, s_{i,l} \geq 0, \mathbf{w} \in \Omega_M} \theta \tag{9}$$

$$\text{s.t. } \theta \geq \sum_{i=1}^{n_\omega} s_{i,l}, \quad \text{for } l \in [M-1]_0, \tag{10}$$

$$s_{i,l} \geq \pm \sum_{j=0}^l (a_{i,j}^* - w_{i,j}) \Delta_j, \quad \text{for } i \in [n_\omega], \quad l \in [M-1]_0. \tag{11}$$

With a different dimension for $s_{k,l}$, we define (SCIA1) as:

$$\min_{\theta, s_{k,l} \geq 0, \mathbf{w} \in \Omega_M} \theta \tag{12a}$$

$$\text{s.t. } \theta \geq \sum_{k=1}^{n_x} s_{k,l}, \quad \text{for } l \in [M-1]_0, \tag{12b}$$

$$s_{k,l} \geq \pm \sum_{j=0}^l \sum_{i=1}^{n_\omega} (a_{i,j}^* - w_{i,j}) \Delta_j \tilde{f}_{i,j,k}, \quad \text{for } k \in [n_x], \quad l \in [M-1]_0. \tag{12c}$$

Finally, we define (λCIA1) by modifying (12c) with the dual variables:

$$\min_{\theta, s_{k,l} \geq 0, \mathbf{w} \in \Omega_M} \theta \tag{13a}$$

$$\text{s.t. } \theta \geq \sum_{k=1}^{n_x} s_{k,l}, \quad \text{for } l \in [M-1]_0, \tag{13b}$$

$$s_{k,l} \geq \pm \lambda_{l,k} \cdot \sum_{j=0}^l \sum_{i=1}^{n_\omega} (a_{i,j}^* - w_{i,j}) \Delta_j \tilde{f}_{i,j,k}, \quad \text{for } k \in [n_x], \quad l \in [M-1]_0. \tag{13c}$$

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

3.3. Chronologically Ordered Constraints

We consider the possibility to modify the (CIA) problems by altering the chronological order in the constraints for the accumulated difference $\|\sum_{l \in [j]} (\mathbf{a}_{\cdot,l}^* - \mathbf{w}_{\cdot,l}) \Delta_l\|$ for $j \in [M-1]_0$. We may use an arbitrary ordering of time intervals, instead of starting from the first interval $j = 0$. Here, we consider backward accumulation, starting from the interval with index $j = M - 1$, i.e., $[t_{M-1}, t_M]$:

$$\theta \geq \pm \sum_{j=l}^{M-1} (a_{i,j}^* - w_{i,j}) \Delta_j, \quad \text{for } i \in [n_\omega], \quad l \in [M-1]_0. \tag{14}$$

Let the problem where (14) replaces (7b) in (CIAMax) be denoted by (CIAMaxB). The other introduced MILPs can be modified analogously with backward time accumulation and are named accordingly; e.g., (SCIA1B) refers to (SCIA1) with backward accumulation.

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:

$$\sigma_{\max} \geq \frac{1}{2} \sum_{i=1}^{n_\omega} \sum_{j=1}^{M-1} |w_{i,j} - w_{i,j-1}|. \tag{15}$$

We will omit this constraint class in the next section, but refer to [51,52] for a priori bounds and reformulations. Nevertheless, we are going to come back to these constraints in the numerical experiments section for also testing the different MILP approaches in this situation.

4. A Priori Bounds for CIA Decompositions

We revise results for the a priori bounds resulting from a CIA decomposition [41] and extend them to alternative decompositions that may be used in S^{CIA} in Algorithm 1. We stress here that Manns et al. [53] have recently presented a proof of improved regularity conditions. Nevertheless, we revise the theorem from [41] here, because it results in natural algorithmic extensions.

4.1. Combinatorial Integral Approximation

We recapitulate a variant of Grönwall’s Lemma from [41], which is needed for the main theorem.

Lemma 1 (A variant of Grönwall’s Lemma, see [41], Lemma 1). *Let $z_1, z_2 : \mathcal{T} \rightarrow \mathbb{R}$ be real-valued integrable functions and let z_2 also belong to $L^\infty(\mathcal{T}, \mathbb{R})$. If, for a constant $L \geq 0$, the following holds:*

$$z_1(t) \leq z_2(t) + L \int_{t_0}^t z_1(\tau) d\tau \quad \text{for a.a. } t \in \mathcal{T},$$

then we have:

$$z_1(t) \leq \|z_2\|_\infty e^{L(t-t_0)} \quad \text{for a.a. } t \in \mathcal{T}. \tag{16}$$

Proof. See [41], proof of Lemma 1. \square

In the following results, we analyze the evolution of two trajectories, \mathbf{x} and \mathbf{y} , based on the same ODE system (2b) but driven by two different controls, α and ω . The following theorem gives a statement on the distance between the two trajectories depending on the distance of the controls.

Theorem 1. *Consider α and $\omega \in \Omega$. We reuse the model functions $f_0, f_i : \mathcal{T} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$ from Definition 2 for $i \in [n_\omega]$. Let $\mathbf{x}(\cdot)$ and $\mathbf{y}(\cdot)$ be the unique solutions of the IVP:*

$$\dot{\mathbf{x}}(t) = f_0(t, \mathbf{x}(t)) + \sum_{i=1}^{n_\omega} \alpha_i(t) f_i(t, \mathbf{x}(t)), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{17a}$$

$$\dot{\mathbf{y}}(t) = f_0(t, \mathbf{y}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) f_i(t, \mathbf{y}(t)), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{17b}$$

where $\mathbf{x}_0, \mathbf{y}_0 \in \mathbb{R}^{n_x}$. Assume that there are positive constants $L, C \in \mathbb{R}^+$, together with a vector norm $\|\cdot\|$, such that, for a.a., $t \in \mathcal{T}$ holds:

$$\|f_i(t, \mathbf{x}(t)) - f_i(t, \mathbf{y}(t))\| \leq L \|\mathbf{x}(t) - \mathbf{y}(t)\|, \quad \text{for } i \in [n_\omega], \tag{17c}$$

$$\left\| \frac{d}{dt} f_i(t, \mathbf{x}(t)) \right\| \leq C, \quad \text{for } i \in [n_\omega]. \tag{17d}$$

Furthermore, let $f_i(\cdot, \mathbf{x}(\cdot))$, $i \in [n_\omega]$ be essentially bounded by $B \in \mathbb{R}^+$ on \mathcal{T} , and assume that for all $t \in \mathcal{T}$. It holds that:

$$\left\| \int_{t_0}^t \alpha(\tau) - \omega(\tau) \, d\tau \right\| \leq \theta, \tag{17e}$$

with the constant $\theta \geq 0$. Then, for a.a. $t \in \mathcal{T}$ we also have:

$$\|\mathbf{x}(t) - \mathbf{y}(t)\| \leq (\|\mathbf{x}_0 - \mathbf{y}_0\| + \theta n_\omega (B + C(t - t_0))) e^{L(t-t_0)}. \tag{17f}$$

Proof. This Theorem is an (equivalent) reformulation of Theorem 2 from [41]. The only differences are modified notations and the usage of f_i , $i \in [n_\omega]_0$ as differentiable mapping instead of A in [41]. \square

We first recognize that Theorem 1 is applicable for the CIA problem with any vector norm, which can be guessed from the equivalence of norms.

Corollary 1 (Approximation bounds via (CIA), cf. Corollary 5.2 in [50]). Consider the setting of Theorem 1; in particular, let the regularity assumptions on f_i , $i \in [n_\omega]_0$, hold. Assume that \mathbf{x} and \mathbf{y}_{CIA} are the solutions of the IVP (2b) and (2c), where \mathbf{x} is based on a given relaxed control \mathbf{a} and \mathbf{y}_{CIA} is based on \mathbf{w}^* , which is the optimal solution of (CIA_{no}), $n_o \in \{\max, 1\}$, with objective value θ_{CIA}^* from Definition 6. Then, the state approximation error is bounded for a.a. $t \in \mathcal{T}$ by:

$$\|\mathbf{x}(t) - \mathbf{y}_{CIA}(t)\| \leq \theta_{CIA}^* (B + C(t - t_0)) e^{L(t-t_0)}. \tag{18}$$

Proof. This corollary is a direct result from Theorem 1 with $\mathbf{x}_0 = \mathbf{y}_0$. We note that θ_{CIA}^* represents the norm of the accumulated control deviation, which appears in the proof of the Theorem and is bounded by θn_ω so that this replaced term is settled. \square

4.2. Scaled Combinatorial Integral Approximation

The proof of Theorem 1 contains a motivation for the (SCIA) problems as derived in the following result.

Corollary 2 (Approximation bounds via (SCIA), cf. Corollary 5.3 in [50]). Consider the setting of Theorem 1, and let $\|\cdot\|_{n_o}$ refer to the maximum or 1-norm, i.e., $n_o \in \{\max, 1\}$. Assume that \mathbf{x} and \mathbf{y}_{CIA} are the solutions of the IVP (2b) and (2c), where \mathbf{x} is based on a given \mathbf{a} , and \mathbf{y}_{CIA} is driven by \mathbf{w}^* , which is the optimal solution of (SCIA_{no}). Then, for a.a. $t \in \mathcal{T}$, the state approximation error is bounded by:

$$\|\mathbf{x}(t) - \mathbf{y}_{SCIA}(t)\| \leq \theta_{SCIA}^* e^{L(t-t_0)} \leq \theta_{CIA}^* (B + C(t - t_0)) e^{L(t-t_0)}, \tag{19}$$

where θ_{SCIA}^* is the optimal objective value of (SCIA_{no}), $n_o \in \{\max, 1\}$.

Proof. From the second and last inequalities in the proof of Theorem 2 from [41] and Corollary 1, it follows that, for a.a. $t \in \mathcal{T}$ and any $\omega \in \Omega$:

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

Let ω^{CIA} denote the control based on the optimal solution w^* of (CIA n_0), $n_0 \in \{\max, 1\}$. We take the minimum in the above inequality and obtain:

$$\theta_{\text{SCIA}}^* \leq \left\| \int_{t_0}^t \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i^{\text{CIA}}(\tau)) f_i(\mathbf{x}(\tau)) \, d\tau \right\| \leq \theta_{\text{CIA}}^* (B + C(t - t_0)).$$

□

We conclude from Corollary 1 that the approximation and convergence results of the decomposition still hold if (SCIA n_0), $n_0 \in \{\max, 1\}$, is used to construct the binary control. The approximation bound based on (SCIA n_0) is tighter than the existing (CIA n_0)-related bound. Hence, it is an obvious choice to consult these alternative binary controls for an approximation study. Ideally, the binary control constructed in this way will result in an improved state approximation and objective value for (MIOCP). However, we oppose this hope next.

Remark 1 (Construction by (SCIA) does not guarantee superior quality). *The binary control based on the optimal solution of (SCIA n_0), $n_0 \in \{\max, 1\}$ and used in the decomposition does not necessarily result in a state approximation or objective value that is superior to that obtained using (CIA n_0). It may hold that:*

$$\|\mathbf{x}(t) - \mathbf{y}_{\text{CIA}}(t)\| < \|\mathbf{x}(t) - \mathbf{y}_{\text{SCIA}}(t)\| < \theta_{\text{SCIA}}^*, \quad \text{for some } t \in \mathcal{T},$$

where we use the notation from Corollaries 1 and 2. Because of a possible non-convex objective, the computed trajectories may lead to a superior objective value for the solution based on (CIA n_0), compared with that based on (SCIA n_0), even if the above inequality does not hold.

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 cost-to-go function. We define it in a more general MIOCP setting with a Bolza objective:

$$\Phi(\mathbf{x}(t_f)) + \int_{t \in \mathcal{T}} L(\mathbf{x}(\tau), \omega(\tau)) \, d\tau, \tag{20}$$

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

Definition 9 (Cost-to-go function by Hamilton–Jacobi–Bellman). *Let the cost-to-go function $J \in L^\infty(\mathbb{R}^{n_x} \times \mathcal{T}, \mathbb{R})$ with Bolza objective (20) be implicitly defined as:*

$$J(\mathbf{x}(t_f), t_f) = \Phi(\mathbf{x}(t_f)),$$

$$-\frac{\partial J}{\partial t}(\mathbf{x}(t), t) = \min_{\omega \in \Omega} L(\mathbf{x}(t), \omega(t)) + \frac{\partial J}{\partial \mathbf{x}}(\mathbf{x}(t), t) \left(f_0(t, \mathbf{x}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) f_i(t, \mathbf{x}(t)) \right).$$

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

Corollary 3 (Approximation bounds via (λ CIA), cf. Corollary 5.4 in [50]). *Consider the setting of Theorem 1. In particular, let the regularity assumptions (17d), (17c), and the essential boundedness of f be true. Assume that \mathbf{x} and $\mathbf{y}_{\lambda\text{CIA}}$ are the solutions of the IVP (2b) and (2c), where \mathbf{x} is based on a given \mathbf{a} , and $\mathbf{y}_{\lambda\text{CIA}}$ is based on \mathbf{w}^* , which is the optimal solution of (λ CIA1).*

Let J be the cost-to-go function as defined in Definition 9 for (MIOCP) and $\lambda(t)$ be the adjoint vector at $t \in \mathcal{T}$ for the ODE system (2b). For a.a. $t \in \mathcal{T}$, it follows that:

$$|J(\mathbf{x}(t), t) - J(\mathbf{y}_{\lambda\text{CIA}}(t), t)| \leq \theta_{\lambda\text{CIA}}^* e^{L(t-t_0)} + o\left(\|\mathbf{x}(t) - \mathbf{y}_{\lambda\text{CIA}}(t)\|^2\right),$$

where o refers to Landau’s little- o notation.

Proof. Let us consider the difference of the cost-to-go functions by approximation with a partial first-order Taylor expansion around $J(\mathbf{x}(t), t)$. We perform the approximation with respect to the trajectories $\mathbf{x}, \mathbf{y}_{\lambda\text{CIA}}$. Thus, we apply Taylor’s theorem for $t \in \mathcal{T}$:

$$J(\mathbf{y}_{\lambda\text{CIA}}(t), t) - J(\mathbf{x}(t), t) = \frac{\partial J}{\partial \mathbf{x}}(\mathbf{x}(t), t)(\mathbf{y}_{\lambda\text{CIA}}(t) - \mathbf{x}(t)) + o\left(\|\mathbf{y}_{\lambda\text{CIA}}(t) - \mathbf{x}(t)\|^2\right). \quad (21)$$

As pointed out above, the dual variables of the ODE constraint (2b) are equal to $\frac{dJ}{dx}(\mathbf{x}(t), t)$. We use the notation $\|\mathbf{x}(t)\|_{\lambda(t)} := \left| \sum_{k \in [n_x]} \lambda_k(t) x_k(t) \right|$ for $t \in \mathcal{T}$, which defines a semi-norm. Next, for a.a. $t \in \mathcal{T}$, we transfer the proof of Theorem 1 to this notation and to (21):

$$\begin{aligned} |J(\mathbf{y}_{\lambda\text{CIA}}(t), t) - J(\mathbf{x}(t), t)| &\leq \|\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x}\|_{\lambda(t)} + o\left(\|\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x}\|^2\right) \\ &\leq \dots \text{ (as in proof of Theorem 1)} \\ &\leq \|\mathbf{y}_0 - \mathbf{x}_0\|_{\lambda(t)} + L \int_{t_0}^t \|\mathbf{y}_{\lambda\text{CIA}}(\tau) - \mathbf{x}(\tau)\|_{\lambda(t)} \, d\tau \\ &\quad + \left\| \sum_{i=1}^{n_\omega} \int_{t_0}^t (\omega_i^{\lambda\text{CIA}}(\tau) - \alpha_i(\tau)) \cdot \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \, d\tau \right\|_{\lambda(t)} \\ &\quad + o\left(\|\mathbf{y}_{\lambda\text{CIA}} - \mathbf{x}\|^2\right). \end{aligned}$$

The third summand of the last inequality is equal to the objective $\theta_{\lambda\text{CIA}}^*$ that is to be minimized in (λCIA1). Finally, we apply $\mathbf{x}_0 = \mathbf{y}_0$ and use the Grönwall Lemma 1 with the integrable functions:

$$z_1(t) = \|\mathbf{x} - \mathbf{y}_{\lambda\text{CIA}}\|_{\lambda(t)}, \quad z_2(t) = \left\| \sum_{i=1}^{n_\omega} \int_{t_0}^t (\omega_i^{\lambda\text{CIA}}(\tau) - \alpha_i(\tau)) \cdot \mathbf{f}_i(\tau, \mathbf{x}(\tau)) \, d\tau \right\|_{\lambda(t)},$$

so that the claim is proven. \square

Due to the first-order Taylor approximation, (λCIA1) requires that the relaxed trajectory \mathbf{x} can be well approximated by a trajectory \mathbf{y} that is based on binary controls. On the other hand, if there is no such trajectory in a close neighborhood of \mathbf{x} , then (λCIA1) may have an unintuitive binary control as its optimal solution.

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 this modified setting to Theorem 1. We express this issue in the following corollary.

Corollary 4 (Approximation bounds via backward constraints). Consider the setting of Theorem 1. Let \mathbf{x} and \mathbf{y} be the state trajectory solutions of the terminal value problems (2b), with $\mathbf{x}(t_f) = \mathbf{x}_f$ and $\mathbf{y}(t_f) = \mathbf{y}_f$ for $\mathbf{x}_f, \mathbf{y}_f \in \mathbb{R}^{n_x}$. Assume that for all $t \in \mathcal{T}$, $\theta_b \in \mathbb{R}^+$, it holds that:

$$\left\| \int_t^{t_f} \alpha(\tau) - \omega(\tau) \, d\tau \right\| \leq \theta_b. \quad (22)$$

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

$$\|\mathbf{x}(t) - \mathbf{y}(t)\| \leq \left(\|\mathbf{x}_f - \mathbf{y}_f\| + \theta_b n_\omega (B + C(t_f - t)) \right) e^{L(t_f - t)}. \tag{23}$$

Proof. We apply the proof of Theorem 1 to the altered setting, in which we integrate over $[t, t_f]$ instead of integrating over $[t_0, t]$.

□

With the assumption that $\|\mathbf{x}(t_f) - \mathbf{y}(t_f)\|$ is small, the backward (CIA) rounding problem approach from Section 3.3 is not only applicable for terminal constraint problems but is also appropriate for an (MIOCP) instance with a given initial value \mathbf{x}_0 and variable final-state values.

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 2 (Arbitrary close approximation of (OCP) solution). We could extend Algorithm 1 with an outer loop that checks if Φ^{opt} is sufficiently close to Φ^{rel} , and if not, we would refine the grid \mathcal{G}_ω . In [29], 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}_ω , with grid size Δ_{\max} , such that there is a feasible trajectory $\{\mathbf{y}(t_j)\}_{t_j \in \mathcal{G}_\omega}$ with:

$$\begin{aligned} |\Phi(\mathbf{x}(t_f)) - \Phi(\mathbf{y}(t_f))| &\leq \epsilon, \\ \|\mathbf{c}(t_j, \mathbf{x}(t_j)) - \mathbf{c}(t_j, \mathbf{y}(t_j))\| &\leq \epsilon. \end{aligned}$$

The proof uses the sum-up rounding scheme that derives the binary control approximation with $\theta_{\text{CIA}}^* \leq \text{Const}(n_\omega) \Delta_{\max}$ [35,41], where $\text{Const}(n_\omega)$ 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 $\theta_{\text{SCIA,max}}^*$, $\theta_{\text{SCIA,1}}^*$ be the optimal objective values of (SCIAMax) and (SCIA1), respectively. Consider the setting of Theorem 1. In particular, let the regularity assumptions (17d), (17c), and essential boundedness of f be true. Assume that \mathbf{x} and \mathbf{y} are the solutions of the IVP (2b) and (2c), where \mathbf{x} is based on a given \mathbf{a} , and \mathbf{y} is based on Algorithm 1. Let \mathcal{G}_ω be the applied grid. It follows for $t_i \in \mathcal{G}_\omega$ that:

$$\|\mathbf{y}(t_i) - \mathbf{x}(t_i)\|_j \leq \left(\|\mathbf{y}_0 - \mathbf{x}_0\|_j + \theta_{\text{SCIA,j}}^* \right) e^{L(t_i - t_0)}, \quad j \in \{\text{max}, 1\}. \tag{24}$$

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. With (λ CIA) or Lipschitz continuity of the objective, there are also tools to discuss the rounding error of the objectives. The recombination heuristics work in the area of objective approximation and, hence, are to be discussed next.

5. Recombination Heuristics

We present several recombination heuristics that recombine different binary controls \mathbf{w} to new candidate solutions with potentially smaller objective values. The general framework is open to apply different heuristics, such as genetic algorithms [54], that are not introduced in this article.

5.1. GreedyTime

Algorithm 2 is a routine for using the MILP solutions in a greedy pattern with the aim of constructing binary controls \mathbf{w} that exhibit an improved objective value $\Phi(\varphi(\mathbf{w}))$.

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

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

```

1 for  $j \in [M - 1]_0$  do
2   for  $(m_1, m_2) \in S^{\text{CIA}} \times S^{\text{CIA}}, m_1 \neq m_2, \mathbf{w}_j^{m_1} \neq \mathbf{w}_j^{m_2}$  do
3     Set  $\tilde{\mathbf{w}}_k^{m_1} = \mathbf{w}_k^{m_1}, k \neq j, k \in [M - 1]_0$  and  $\tilde{\mathbf{w}}_j^{m_1} = \mathbf{w}_j^{m_2}$ ;
4     Evaluate (MIOCP) with fixed  $\tilde{\omega}^{m_1} = \varphi(\tilde{\mathbf{w}}^{m_1}) \rightarrow \Phi(\varphi(\tilde{\mathbf{w}}^{m_1}))$ .
5     if  $\Phi(\varphi(\tilde{\mathbf{w}}^{m_1})) \leq \Phi(\varphi(\mathbf{w}^{m_1}))$  then
6       Set  $\mathbf{w}_j^{m_1} = \tilde{\mathbf{w}}_j^{m_1}$  and  $\Phi(\varphi(\mathbf{w}^{m_1})) = \Phi(\varphi(\tilde{\mathbf{w}}^{m_1}))$ .
7     end
8   end
9 end
10 return:  $\Phi(\varphi(\mathbf{w}^{\text{rec}})) := \min_{\text{milp} \in S^{\text{CIA}}} \Phi(\varphi(\mathbf{w}^{\text{milp}}))$ .
```

In *GreedyTime*, we iterate over all intervals $j \in [M - 1]_0$ in chronological order (line 1). On every interval, we check if there are MILP pairs (m_1, m_2) that differ in their binary control vectors in line 2. We recombine for each of these pairs the m_1 solution with the binary control vector from m_2 at interval j to construct a temporary control solution $\tilde{\mathbf{w}}^{m_1}$ (line 3). Based on this construction, we evaluate the objective of this new control solution in line 4 and overwrite the binary control \mathbf{w}^{m_1} with the recombined solution $\tilde{\mathbf{w}}^{m_1}$ if that latter results in an improved objective value (lines 5–6). We proceed in the same way with the second solution m_2 when the (same) pair (m_2, m_1) appears in the inner loop.

A large number of calculated MILPs may result in a large number of pairs with unequal control solutions. In this case, it is advisable to only swap the \mathbf{w}^{m_2} solution with the currently smallest objective value, instead of swapping and testing each variation for every \mathbf{w}^{m_1} .

If the control problem at hand involves no continuous controls \mathbf{u} , it is straightforward to evaluate the (MIOCP) in line 4 with the previously found and fixed \mathbf{x} until grid interval j . This speeds up the process in problems with fine grids and numerous MILP solutions, because (MIOCP) needs to be solved iteratively. Moreover, if an MILP solution m_1 differs from two MILPs m_2, m_3 with identical binary control vectors $\mathbf{w}_{:,j}^{m_2} = \mathbf{w}_{:,j}^{m_3}$, it is sufficient to test the recombination with only one of the two. We illustrate an example recombination step for the pairs (CIA, SCIA) and (SCIA, CIA) in Figure 1.

Remark 3 (Modifications of GreedyTime).

1. We can also apply the outer loop in Algorithm 2 backward in time and name the backward version *GreedyTimeBackward*;
2. We may consider only singular arcs, instead of looping over all intervals, since the constructed binary controls are likely to be equal on bang–bang arcs. With singular arcs, we mean the intervals where $\epsilon < a_{i,j}^* < 1 - \epsilon$ holds for the optimal control solution \mathbf{a}^* of (OCP), with a certain threshold $\epsilon > 0$;
3. *Greedy-cost-to-go modification*: Assume we have obtained the dual variables $\tilde{\lambda}_{j,k}, j \in [M - 1]_0, k \in [n_x]$, of the state equations of (OCP). Then, re-sort the intervals $[M - 1]_0$ in descending order according to $\sum_{k \in [n_x]} |\tilde{\lambda}_{j,k}|, j \in [M - 1]_0$. In this way, we construct a new ordered grid $\mathcal{G}_\omega^\lambda$ to be iterated over in Algorithm 2.

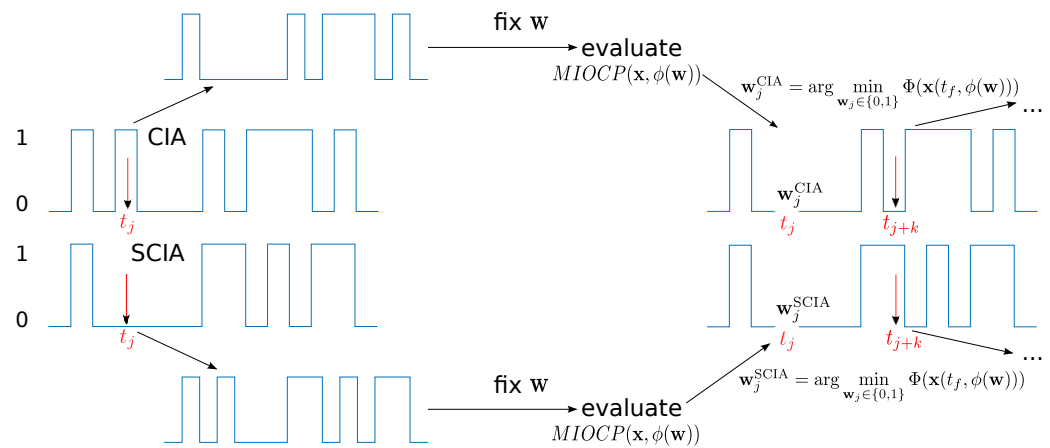


Figure 1. Example visualization of the GreedyTime algorithm. We use two candidate control solutions, here from CIA and SCIA, to construct new candidates. We perform an enumeration between 0 and 1 at all times t_j when the input vectors differ. Then, the two candidate solutions w are fixed and we evaluate (MIOCP) for both vectors. We compare the resulting objective function values with their previous values. Moreover, the binary w_j values with the lower objective values are fixed in the candidate solutions. This procedure is repeated on the next grid point with unequal candidate solutions.

5.2. Singular Arc Recombination

If a^* is (almost) binary on certain intervals, w should typically attain these binary values as well as an optimal solution of a CIA rounding problem—regardless of the MILP choice. To this end, we formalize singular arcs of a^* as sets of consecutive intervals on which the relaxed control takes values smaller than ϵ or larger than $1 - \epsilon$. Here, $\epsilon > 0$ is a chosen small tolerance.

Definition 10 (Number of singular arcs n_{sing} , singular arc interval sets $\mathcal{J}_l^{\text{sing}}$). Consider a^* as the optimal control solution of (OCP) and a small chosen tolerance $\epsilon > 0$. Let $k_0^{\text{end}} := -1$. We introduce the following singular arc interval index sets iteratively for $l \geq 1$:

$$\begin{aligned}
 k_l^{\text{start}} &:= \min \{ j \in [M - 1]_0 \mid j > k_{l-1}^{\text{end}} \wedge \exists i \in [n_\omega] : a_{i,j}^* \in [\epsilon, 1 - \epsilon] \}, \\
 k_l^{\text{end}} &:= \max \{ j \in [M - 1]_0 \mid \forall r = k_l^{\text{start}}, \dots, j \exists i \in [n_\omega] : a_{i,r}^* \in [\epsilon, 1 - \epsilon] \}, \\
 \mathcal{J}_l^{\text{sing}} &= \{ k_l^{\text{start}}, \dots, k_l^{\text{end}} \}.
 \end{aligned}$$

Let the number of singular arcs n_{sing} be defined as:

$$n_{\text{sing}} := \arg \max_{l \in \mathbb{N}} \{ k_l^{\text{end}} \}.$$

We aim to recombine singular arc realizations of the different MILP solutions from $SCIA$, which is performed in Algorithm 3.

We initialize the set of visited binary controls as empty in the algorithm and set the so-far best objective value Φ^{rec} to infinity (line 1). Next, the temporary binary control w^{tmp} on the bang–bang arcs is set as equal to the rounded relaxed control (line 2). In line 3, we test every possible variation of the different MILP solutions on the singular arcs. In this way, we fill up the singular arcs of the temporary binary control w^{tmp} (line 4). The constructed control w^{tmp} is checked if it has already been visited (line 5), and if so, the algorithm jumps to the next iteration (line 6). Otherwise, we include w^{tmp} in the set of visited controls (line 8), and we evaluate its objective value (line 9). When a recombination has an improved objective value than the so-far best control, it will be saved as the so-far

best control (lines 10–12). We illustrate the algorithm in Figure 2.

Algorithm 3: Singular arc block heuristic for recombining binary controls \mathbf{w}^m , $\mathbf{m} \in S^{\text{CIA}}$.

Input : Grid \mathcal{G}_ω , small singular arc tolerance $\epsilon > 0$, singular arcs interval sets $\mathcal{J}_l^{\text{sing}}$, relaxed control values form (OCP) \mathbf{a}^* , binary controls \mathbf{w}^m , $\mathbf{m} \in S^{\text{CIA}}$, corresponding objective values $\Phi(\varphi(\mathbf{w}^m))$.

- 1 Set $S^{\text{vis}} = \emptyset$ and $\Phi^{\text{rec}} = \infty$;
- 2 Set $w_{i,j}^{\text{tmp}} = \lfloor a_{i,j}^* + \epsilon \rfloor$, for $i \in [n_\omega]$, $j \in [M - 1]_0 \setminus \{ \mathcal{J}_l^{\text{sing}} \}_{l \in [n_{\text{sing}}]}$;
- 3 **for** $(m_1, \dots, m_{n_{\text{sing}}}) \in \times_{l \in [n_{\text{sing}}]} S^{\text{CIA}}$ **do**
- 4 Set $\mathbf{w}_{\cdot,j}^{\text{tmp}} = \mathbf{w}_{\cdot,j}^{m_l}$, for $j \in \mathcal{J}_l^{\text{sing}}$, $l \in [n_{\text{sing}}]$;
- 5 **if** $\mathbf{w}^{\text{tmp}} \in S^{\text{vis}}$ **then**
- 6 | continue;
- 7 **end**
- 8 **else**
- 9 Set $S^{\text{vis}} = S^{\text{vis}} \cup \{ \mathbf{w}^{\text{tmp}} \}$;
- 10 Evaluate (MIOCP) with $\mathbf{w} = \mathbf{w}^{\text{tmp}}$ fixed $\rightarrow \Phi(\varphi(\mathbf{w}^{\text{tmp}}))$;
- 11 **if** $\Phi(\varphi(\mathbf{w}^{\text{tmp}})) < \Phi^{\text{rec}}$ **then**
- 12 | Set $\Phi^{\text{rec}} = \Phi(\varphi(\mathbf{w}^{\text{tmp}}))$;
- 13 | Set $\mathbf{w}^{\text{rec}} = \mathbf{w}^{\text{tmp}}$;
- 14 **end**
- 15 **end**
- 16 **end**
- 17 **return:** \mathbf{w}^{rec} together with Φ^{rec} .

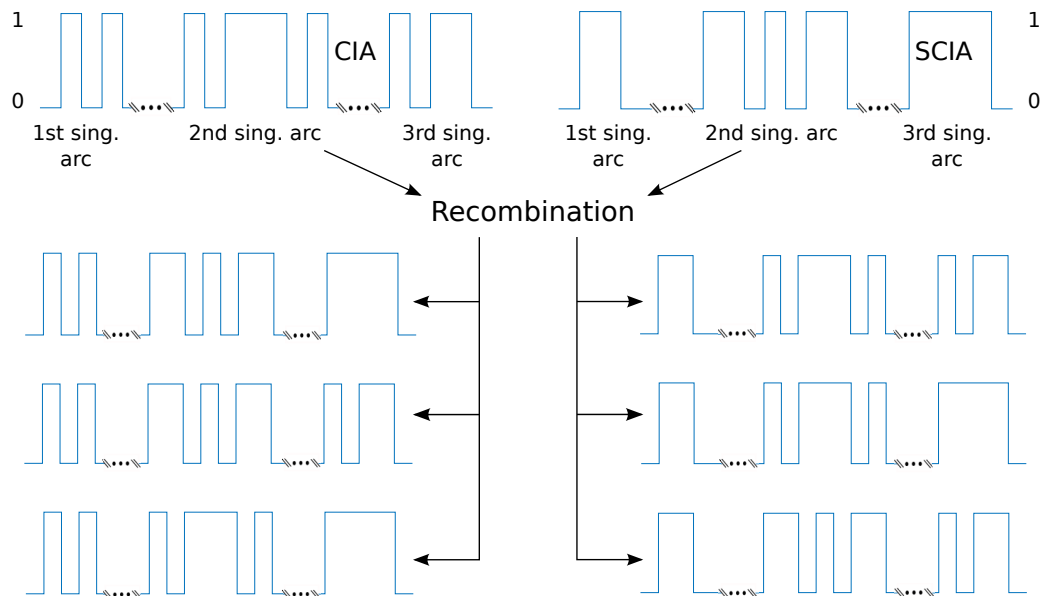


Figure 2. Visualization of the singular arc block recombination heuristic for two MILP control vectors (which we name CIA and SCIA) with three singular arcs. Every possible variation from the singular arcs and candidate controls is generated and we evaluate (MIOCP) for each of the constructed variation. The minimal objective value of all variations represents the heuristic’s result.

We have to take care of the number of possible variations of singular blocks and MILP solutions $|S^{\text{CIA}}|^{n_{\text{arc}}}$ to avoid a combinatorial explosion. Therefore, it is advisable to choose \tilde{S}^{CIA} in Algorithm 1 with a small number of MILPs. Only a few singular arcs result usually

after solving (OCP). We may modify Algorithm 3 to be greedy, i.e., to apply the idea of *GreedyTime* on arcs instead of on single intervals, if there are more than four singular arcs.

The singular arc recombination yields an objective value that is at least as good as those previously constructed via the MILPs. However, there currently exists no framework for quantifying these possible improvements in terms of new rounding errors of the objective.

6. Computational Results

6.1. Software Implementation and Instances

We implemented Algorithm 1 in AMPL [55] using the code `ampl_mintoc`, which is a modeling framework to solve optimal control problems. It features different discretization schemes of ODEs, though we used only a Radau collocation from [39]. The tool is advantageous for our purpose, since it includes automatic differentiation, interfaces to MILP solvers, and its problem formulation stays close to mathematics. In addition, AMPL provides the dual variables λ . Throughout the numerical study, we applied Gurobi 8.1 as the MILP solver and IPOPT 3.12.4 as the NLP solver, with default settings in each case, 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 four Intel i5-4210U CPUs (1.7 GHz) and 7.7 GB RAM.

We included MIOCPs from the benchmark collection site `mintoc.de` [18] in our numerical study, which we specify further in the following subsections. For these problems, we chose a differential states discretization with N intervals such that it was fine enough respect to the objective value. In this way, the objective value differs only to the fifth decimal place, with respect to a finer discretization for constant M . Afterwards, we varied M with fixed N in order to construct different instances. We refer to Appendix A for further details. Solving the binary approximation problem and then solving the MIOCP with fixed binary controls might result in infeasible solutions for problems involving path or terminal constraints. To this end, we relaxed these constraints and applied a merit function that penalizes constraint violation to be added to the objective with a sufficiently high penalty factor.

6.2. Scaled Combinatorial Integral Approximation

Our hypothesis is that the MILPs based on a scaled combinatorial integral approximation perform the best on instances where the binary control enters the control dependent right-hand side terms f_i of the ODE in an affine way, i.e.,:

$$\dot{\mathbf{x}}(t) = f_0(t, \mathbf{x}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) c_i, \quad \text{for a.a. } t \in \mathcal{T}, \quad (25)$$

where $c_i \in \mathbb{R}$. On the other hand, if f_i depends on $\mathbf{x}(t)$, it may change rapidly over time. To this end, this results in possibly inaccurate ω solutions, since we use only the discretized state trajectory $\mathbf{x}(t)$ value. The MIOCPs “Double tank (Multimode)” and “Lotka–Volterra (absolute fishing variant)” are identified as candidate problems with the above right-hand side structure and constructed their solutions for different discretizations and both with and without the combinatorial constraint (15); see Appendix A for details. The results are presented in Figure 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. We argued in Section 4 that the CIA decomposition is built on this distance, and, particularly, the proximity of objective values and constraint satisfaction follows as derived in Section 4.5. The differential state trajectories based on the (SCIA1) and (SCIAmax) solutions are significantly closer to the relaxed solution compared with their CIA counterparts, as shown in the performance plot. 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 winners. Analogously, the result remains similar when comparing the algorithms solely on instances (not) including combinatorial constraints.

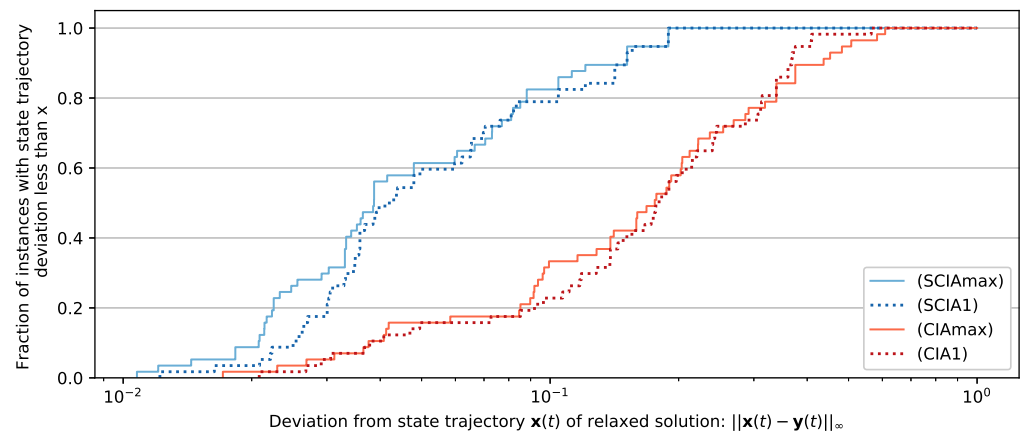


Figure 3. Performance profile comparing the deviation of differential states based on SCIA and CIA solutions. Relaxed solutions are shown in maximum norm and log-scale. The results are based on the instances “Double tank (Multimode)” and “Lotka–Volterra (absolute fishing variant)” from the mintoc.de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

6.3. λ -Combinatorial Integral Approximation

We derived (λ CIA) as an 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. Our hypothesis is 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.:

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

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 (15) for comparing the (λ CIA) solutions with the ones obtained via (CIA). We present the computation results in Figure 4.

In contrast to the previous section, we compared here the objective deviations from the relaxed solution in percentages, since the λ -combinatorial integral approximation aims directly at improving the objective values. However, we remark that the latter algorithm performs worse than (CIA1) and (CIAMax) if the distance to the relaxed solution is measured in differential state space. The performance plot shows that (λ CIA) provides solutions with improved objective values on some instances, but on many others it does not. Since (λ CIA) 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.

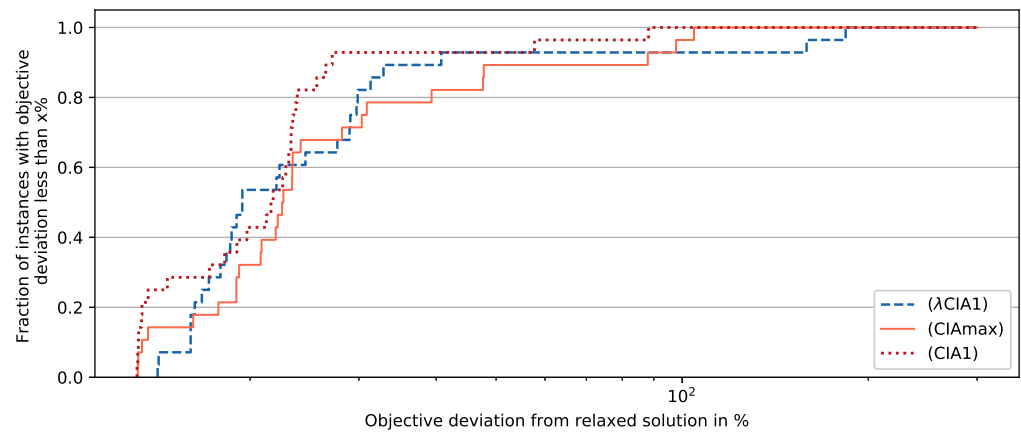


Figure 4. Performance profile comparing objective deviation from the relaxed solution in percentage and log-scale of λ -CIA and CIA solutions. The results are based on the instance “Quadrotor (binary variant)” from the mintoc.de benchmark library. (λ CIA) appeared to provide no clear improvement compared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

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 the differential states. Because the deviation to the relaxed solution can become large, the standard (CIAmax) approach may construct a solution that does not satisfy this constraint. Nevertheless, the direct incorporation of terminal constraints into the MIOCP may lead to numerical difficulties already for the relaxed problem. To this end, we considered soft constraints, meaning that we introduce slack variables in order to penalize a deviation of the differential states from a desired terminal value. We identified the MIOCP “Lotka–Volterra (terminal constraint violation)” as a candidate problem and calculated its solutions for different discretizations and both with and without the combinatorial constraint (15). We illustrate the objective deviation from the relaxed solution in percentage of (CIA) and its backward variant solutions in Figure 5.

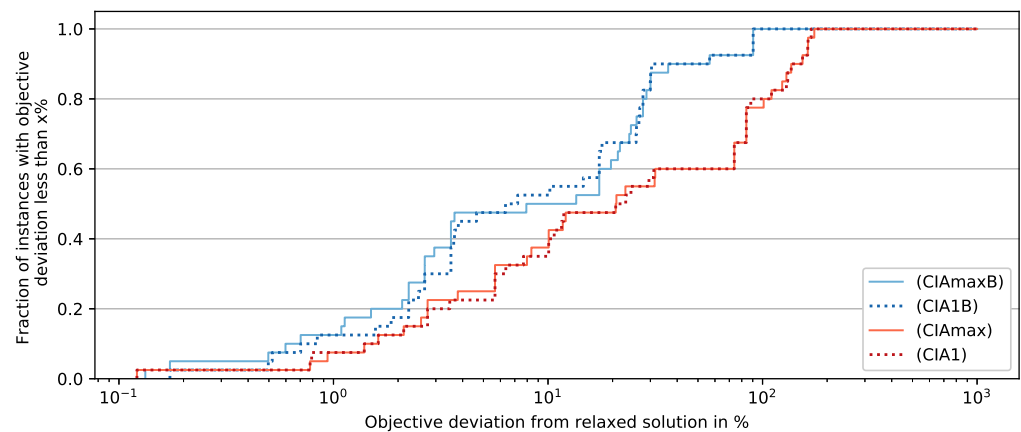


Figure 5. Performance profile comparing objective deviation from the relaxed solution in percentage and log-scale of (CIA) and its backward variant solutions. The results are based on the instance “Lotka–Volterra (terminal constraint violation)” from the mintoc.de benchmark library. Using (CIA1B) or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

We chose the objective deviation as the performance measure for our comparison study because 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 respective MIOCP solutions involve smaller 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), (λ CIA1), 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 illustrates the numerical results with respect to objective deviation of each algorithm to the relaxed solution in percentages.

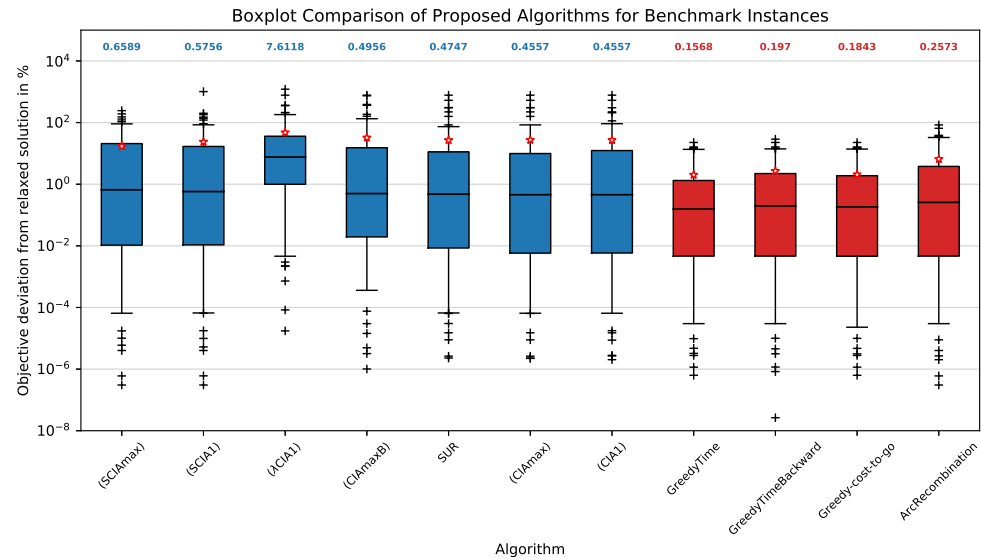


Figure 6. Box plot comparing objective deviation from the relaxed solution in percentage and log-scale of several MILP (marked in blue) and recombination heuristic (marked in red) solutions. The results are based on instances from the mintoc.de benchmark library. The box borders are 1/4 and 3/4-quantiles, 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 larger and their mean values and their outliers a bit smaller, respectively, than the ones of the (CIA) MILPs. The numerical study revealed several instances where (SCIAMax) or (SCIA1) ran into a binary solution with active controls on some intervals with relaxed values close to zero. Under the assumption that the combinatorial approximation is conducted 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 provide any real improvement.

The solutions of (λ CIA1) 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 in terms of the objective than the MILP-constructed binary 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 $N = 12,000$ 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-0A, 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 GreedyTime cannot improve much. For other discretizations 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.

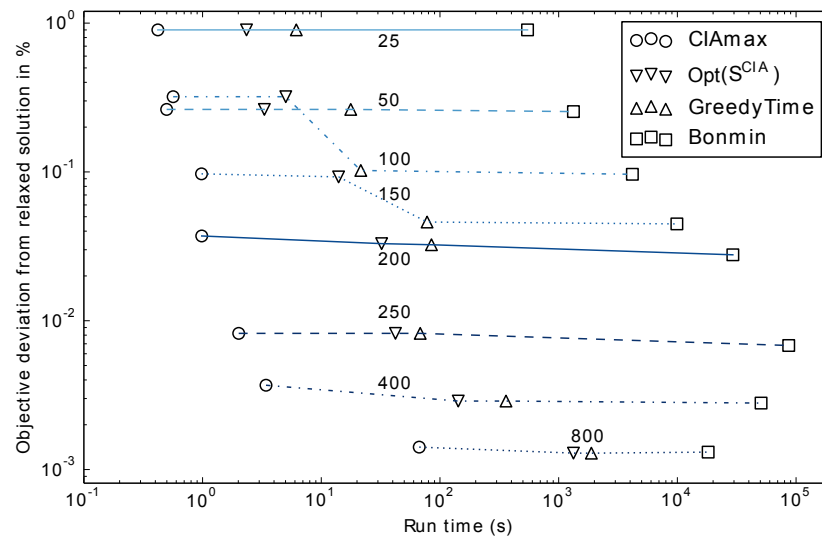


Figure 7. Log plot of run time and objective value deviation from the relaxed solution of the constructed solutions for different approaches and for the Lotka–Volterra multimode problem, with differential state discretization $N = 12,000$. The numbers in the plot indicate the applied corresponding number of control grid points M . We illustrate the outcomes of the solutions constructed by (CIAmax), $Opt(S^{CIA})$, the GreedyTime recombination heuristic, and the MINLP solver Bonmin. By $Opt(S^{CIA})$, we denote the best objective value outcome of over all MILP solutions. For each control discretization, we connect the outcomes of the four approaches with lines in order to compare the behavior for different discretizations. One observes the convergence of all approaches towards the lower bound provided by the relaxed solution and the closure of the gap between (CIAmax) and Bonmin solutions for a fixed discretization. GreedyTime is roughly two orders of magnitude slower than (CIAmax), but is 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 we set a time limit of 30 min.

We remark that the greedy heuristics and the ArcRecombination are to be used cautiously, because 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 cases 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 min), yet provide solutions with objective function values very close to those of the relaxed problem.

A way to significantly reduce the computation time of the MILPs is to apply branch-and-bound [32] or to use SUR for constructing approximate solutions. These algorithms are implemented in the open-source software package *pycombina* (see <https://github.com/adbuenger/pycombina>) (accessed on 14 February 2022) [56] 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 (25), 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 the MINLP solver, cast a positive light on the proposed decomposition algorithm including recombination.

7. Summary and Conclusions

We have extended the decomposition approach based on combinatorial integral approximation [32], 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 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.

Furthermore, it would be interesting to include and to compare the results of this study computationally with recent approaches for the CIA decomposition [43,44].

Author Contributions: C.Z. conducted the implementations and the numerical study, and was responsible for writing most of the article. T.W. contributed major algorithmic ideas and proofread the manuscript. S.S. contributed to the research design and discussions of the algorithmic ideas and wrote parts of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by European Research Council grant number 647573.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We acknowledge the financial support by the Federal Ministry of Education and Research of Germany with in the project P2Chem (support code 05M18OCB). This project has received funding from the European Research Council (ERC, grant agreement No 647573) under the European Union’s Horizon 2020 research and innovation program and and from Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—314838170, GRK 2297 MathCoRe and SPP 1962 and SPP 2331.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

The following mathematical symbols are used in this manuscript:

\mathcal{T}	Time horizon
t	Time variable
\mathbf{x}, \mathbf{y}	Differential state vectors
ω, α	Binary control and relaxed binary control
\mathbf{w}, \mathbf{a}	Discretized binary control and relaxed binary control variables
Φ	Objective function for (MIOCP)
f_0, f_i	Model functions of the ODE system
\mathbf{c}	Path constraint function for (MIOCP)
M, n_x, n_ω	Number of discretization intervals, differential states, and controls
\mathcal{G}_ω	Control discretization grid
Δ	Discretization grid length
φ	Canonical mapping from dicretized controls to control functions
Ω, Ω_M	Space of binary control functions and discretized binary control variables
λ	Evaluated dual variables of the ODE constraints (2b)
$\mathcal{S}^{\text{CIA}}, \mathcal{S}^{\text{REC}}$	Set of CIA problems and set of recombination mappings
F^{rec}	Recombination mapping
θ^*, θ	Optimal MILP objective value, auxiliary MILP objective variable
s	Auxiliary MILP constraint variable
σ_{\max}	Number of allowed switches
L	Lipschitz constant
B, C	Upper bounds on f_i and their derivatives
J	Cost-to-go function
$\mathcal{J}_l^{\text{sing}}$	The l th singular arc interval index set

Appendix A. Detailed Numerical Results

Appendix 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)”:

$N = 12,000$, $M \in \{25, 50, 75, 80, 100, 120, 150, 160, 200\}$, $\sigma_{\max} \in \{10, 20, \infty\}$,

“Quadrotor (binary variant)”:

$N = 12,000$, $M \in \{25, 50, 60, 80, 100, 150, 200, 300\}$, $\sigma_{\max} \in \{4, 10, 20, \infty\}$;

“Lotka–Volterra (terminal constraint violation)”:

$N = 12,000$, $M \in \{20, 30, 40, 50, 60, 100, 120, 200, 240, 300, 400, 600\}$,

$\sigma_{\max} \in \{4, 10, 20, \infty\}$;

“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”:

$N = 18,000$, $M \in \{25, 50, 100, 180, 250, 300, 360, 720\}$;

“Double Tank multimode”:

$N = 12,000$, $M \in \{20, 25, 50, 100, 200, 250, 300, 400, 600\}$; $\sigma_{\max} \in \{10, 20, \infty\}$,

“Lotka–Volterra fishing problem”:

$N = 12,000$, $M \in \{20, 30, 40, 60, 100, 120, 200, 300, 400, 600\}$;

“Lotka–Volterra multi-arcs problem”:

$N = 18,000, M \in \{25, 50, 100, 150, 200, 250, 300, 400, 600\}$;

“Lotka–Volterra multimode problem”:

$N = 12,000, M \in \{25, 50, 100, 150, 200, 250, 300, 400, 800\}$;

“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”:

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

Appendix 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 percentages 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
λ CIA1	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 $N = 12,000$ 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	8.52473×10^{-4}	30	2.01582	1.82887	8.52473×10^{-4}	30	6.84566
300	1.82884	2.1597×10^{-4}	33	1.87382	1.82884	2.1597×10^{-4}	33	27.4496
400	1.82879	3.40892×10^{-4}	47	3.42292	1.82879	3.40892×10^{-4}	47	45.0224
800	1.82875	2.58672×10^{-4}	87	66.8739	1.82875	2.58672×10^{-4}	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	8.52473×10^{-4}	30	11.6632	1.82887	8.5189×10^{-4}	30	65.8286
300	1.82887	4.80446×10^{-4}	32	8.75161	1.82887	4.80449×10^{-4}	32	55.7173
400	1.82877	1.94567×10^{-4}	47	30.4913	1.82877	1.94577×10^{-4}	47	188.408
800	1.83859	0.00987316	88	233.701	1.8381	0.00937638	89	1479.19
λ (CIA1)				(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	5.56473 e-05	35	2.94341
400	2.07161	0.242853	128	12.5695	1.82878	2.53493×10^{-4}	47	5.77022
800	3.44174	1.61302	420	18.8157	1.82875	3.20174×10^{-4}	89	34.2567

Table A2. Cont.

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	8.52473×10^{-4}	30	25.9954	1.82887	8.52473×10^{-4}	30	25.6038
300	1.82884	2.1597×10^{-4}	33	81.1383	1.82884	2.1597×10^{-4}	33	82.31
400	1.82877	1.94567×10^{-4}	47	217.31	1.82877	1.94567×10^{-4}	47	179.64
800	1.82874	2.35655×10^{-4}	87	553.42	1.82874	2.3582×10^{-4}	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	8.52473×10^{-4}	30	0.6946	1.82887	8.52473×10^{-4}	30	25.6708
300	1.82884	2.1597×10^{-4}	33	0.9826	1.82884	2.1597×10^{-4}	33	103.187
400	1.82877	1.94567×10^{-4}	47	0.5933	1.82877	1.94567×10^{-4}	47	302.851
800	1.82874	2.3582×10^{-4}	87	0.7660	1.82874	2.35652×10^{-4}	87	1166.96

References

- Egerstedt, M.; Wardi, Y.; Delmotte, F. Optimal Control of Switching Times in Switched Dynamical Systems. In Proceedings of the 42nd IEEE Conference of Decision and Control, Maui, HI, USA, 9–12 December 2003.
- Seatzu, C.; Corona, D.; Giua, A.; Bemporad, A. Optimal control of continuous-time switched affine systems. *IEEE Trans. Autom. Control.* **2006**, *51*, 726–741. [\[CrossRef\]](#)
- Buss, M.; Glocker, M.; Hardt, M.; Stryk, O.V.; Bulirsch, R.; Schmidt, G. *Nonlinear Hybrid Dynamical Systems: Modelling, Optimal Control, and Applications*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 279, pp. 311–335.
- Goebel, R.; Sanfelice, R.G.; Teel, A.R. Hybrid dynamical systems. *IEEE Control. Syst.* **2009**, *29*, 28–93. [\[CrossRef\]](#)
- Burgschweiger, J.; Gnädig, B.; Steinbach, M. Nonlinear Programming Techniques for Operative Planning in Large Drinking Water Networks. *Open Appl. Math. J.* **2009**, *3*, 1–16. [\[CrossRef\]](#)
- Doban, A.I.; Lazar, M. A switched systems approach to cancer therapy. In Proceedings of the 2015 European Control Conference (ECC), Linz, Austria, 15–17 July 2015; pp. 2718–2724.
- Koch, T.; Hiller, B.; Pfetsch, M.E.; Schewe, L. (Eds.) *Evaluating Gas Network Capacities*; SIAM-MOS Series on Optimization; SIAM: Philadelphia, PA, USA, 2015. Available online: https://archive.siam.org/books/mo21/mo21_toc.pdf (accessed on 15 February 2022).
- Gugat, M.; Herty, M.; Klar, A.; Leugering, G. Optimal Control for Traffic Flow Networks. *J. Optim. Theory Appl.* **2005**, *126*, 589–616. [\[CrossRef\]](#)
- Fügenshuh, A.; Herty, M.; Klar, A.; Martin, A. Combinatorial and Continuous Models for the Optimization of Traffic Flows on Networks. *SIAM J. Optim.* **2006**, *16*, 1155–1176. [\[CrossRef\]](#)
- Göttlich, S.; Potschka, A.; Ziegler, U. Partial Outer Convexification for Traffic Light Optimization in Road Networks. *SIAM J. Sci. Comput.* **2017**, *39*, B53–B75. [\[CrossRef\]](#)
- Göttlich, S.; Herty, M.; Kirchner, C.; Klar, A. Optimal control for continuous supply network models. *Netw. Heterog. Media* **2007**, *1*, 675–688.
- Abichandani, P.; Benson, H.; Kam, M. Multi-vehicle path coordination under communication constraints. In Proceedings of the American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 650–656. [\[CrossRef\]](#)
- Kawajiri, Y.; Biegler, L. A Nonlinear Programming Superstructure for Optimal Dynamic Operations of Simulated Moving Bed Processes. *Ind. Eng. Chem. Res.* **2006**, *45*, 8503–8513. [\[CrossRef\]](#)
- Sonntag, C.; Stursberg, O.; Engell, S. Dynamic Optimization of an Industrial Evaporator using Graph Search with Embedded Nonlinear Programming. *IFAC Proc. Vol.* **2006**, *39*, 211–216. [\[CrossRef\]](#)
- Zeile, C.; Rauwolf, T.; Schmeisser, A.; Mizerski, J.K.; Braun-Dullaeus, R.C.; Sager, S. An Intra-Cycle Optimal Control Framework for Ventricular Assist Devices Based on Atrioventricular Plane Displacement Modeling. *Ann. Biomed. Eng.* **2021**, *49*, 3508–3523. [\[CrossRef\]](#)
- Gerdts, M. A variable time transformation method for mixed-integer optimal control problems. *Optim. Control. Appl. Methods* **2006**, *27*, 169–182. [\[CrossRef\]](#)
- Robuschi, N.; Zeile, C.; Sager, S.; Braghin, F. Multiphase mixed-integer nonlinear optimal control of hybrid electric vehicles. *Automatica* **2021**, *123*, 109325. [\[CrossRef\]](#)
- Sager, S. A benchmark library of mixed-integer optimal control problems. In *Mixed Integer Nonlinear Programming*; Lee, J., Leyffer, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 631–670.

19. Gerdtts, M.; Sager, S. Mixed-Integer DAE Optimal Control Problems: Necessary conditions and bounds. In *Control and Optimization with Differential-Algebraic Constraints*; Biegler, L., Campbell, S., Mehrmann, V., Eds.; SIAM: Philadelphia, PA, USA, 2012; pp. 189–212.
20. Hellström, E.; Ivarsson, M.; Aslund, J.; Nielsen, L. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control. Eng. Pract.* **2009**, *17*, 245–254. [[CrossRef](#)]
21. Lee, H.; Teo, K.; Jennings, L.; Rehbock, V. Control Parametrization Enhancing Technique for Optimal Discrete-Valued Control Problems. *Automatica* **1999**, *35*, 1401–1407. [[CrossRef](#)]
22. Stellato, B.; Ober-Blöbaum, S.; Goulart, P.J. Second-Order Switching Time Optimization for Switched Dynamical Systems. *IEEE Trans. Autom. Control.* **2017**, *62*, 5407–5414. [[CrossRef](#)]
23. Till, J.; Engell, S.; Panek, S.; Stursberg, O. Applied Hybrid System Optimization: An Empirical Investigation of Complexity. *Control. Eng. Pract.* **2004**, *12*, 1291–1303. [[CrossRef](#)]
24. Ringkamp, M.; Ober-Blöbaum, S.; Leyendecker, S. On the time transformation of mixed integer optimal control problems using a consistent fixed integer control function. *Math. Program.* **2017**, *161*, 551–581. [[CrossRef](#)]
25. Sager, S.; Tetschke, M.; Zeile, C. *A Numerical Study of Transformed Mixed-Integer Optimal Control Problems*; Technical Report. 2022. Available online: http://www.optimization-online.org/DB_FILE/2020/03/7698.pdf (accessed on 15 February 2022).
26. Gerdtts, M. Solving mixed-integer optimal control problems by Branch&Bound: A case study from automobile test-driving with gear shift. *Optim. Control. Appl. Methods* **2005**, *26*, 1–18.
27. Sager, S. *Numerical Methods for Mixed-Integer Optimal Control Problems*; Der andere Verlag: Marburg, Germany, 2005. ISBN 3-89959-416-9.
28. Sager, S.; Claeys, M.; Messine, F. Efficient upper and lower bounds for global mixed-integer optimal control. *J. Glob. Optim.* **2015**, *61*, 721–743. [[CrossRef](#)]
29. Sager, S.; Reinelt, G.; Bock, H. Direct Methods With Maximal Lower Bound for Mixed-Integer Optimal Control Problems. *Math. Program.* **2009**, *118*, 109–149. [[CrossRef](#)]
30. Zhu, F.; Antsaklis, P.J. Optimal control of hybrid switched systems: A brief survey. *Discret. Event Dyn. Syst.* **2015**, *25*, 345–364. [[CrossRef](#)]
31. Burgschweiger, J.; Gnädig, B.; Steinbach, M. *Optimization Models for Operative Planning in Drinking Water Networks*; Technical Report ZR-04-48; ZIB: 2004. Available online: <file:///C:/Users/MDPI/Downloads/ZR-04-48.pdf> (accessed on 15 February 2022).
32. Sager, S.; Jung, M.; Kirches, C. Combinatorial Integral Approximation. *Math. Methods Oper. Res.* **2011**, *73*, 363–380. [[CrossRef](#)]
33. Stein, O. Error bounds for mixed integer nonlinear optimization problems. *Optim. Lett.* **2016**, *10*, 1153–1168. [[CrossRef](#)]
34. Belotti, P.; Kirches, C.; Leyffer, S.; Linderoth, J.; Luedtke, J.; Mahajan, A. Mixed-Integer Nonlinear Optimization. In *Acta Numerica*; Iserles, A., Ed.; Cambridge University Press: Cambridge, UK, 2013; Volume 22, pp. 1–131. [[CrossRef](#)]
35. Kirches, C.; Lenders, F.; Manns, P. Approximation Properties and Tight Bounds for Constrained Mixed-Integer Optimal Control. *SIAM J. Control. Optim.* **2016**, *58*, 1371–1402. [[CrossRef](#)]
36. Hante, F.; Sager, S. Relaxation Methods for Mixed-Integer Optimal Control of Partial Differential Equations. *Comput. Optim. Appl.* **2013**, *55*, 197–225. [[CrossRef](#)]
37. Hante, F.M. Relaxation methods for hyperbolic PDE mixed-integer optimal control problems. *Optim. Control. Appl. Methods* **2017**, *38*, 1103–1110. [[CrossRef](#)]
38. Bürger, A.; Zeile, C.; Hahn, M.; Altmann-Dieses, A.; Sager, S.; Diehl, M. pycombin: An open-source tool for solving combinatorial approximation problems arising in mixed-integer optimal control. *IFAC-PapersOnLine* **2020**, *53*, 6502–6508. [[CrossRef](#)]
39. Biegler, L. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*; Series on Optimization; SIAM: Philadelphia, PA, USA, 2010.
40. Gerdtts, M. *Optimal Control of ODEs and DAEs*; De Gruyter: Berlin, Germany, 2012.
41. Sager, S.; Bock, H.; Diehl, M. The Integer Approximation Error in Mixed-Integer Optimal Control. *Math. Program. A* **2012**, *133*, 1–23. [[CrossRef](#)]
42. Bestehorn, F.; Hansknecht, C.; Kirches, C.; Manns, P. Switching Cost Aware Rounding for Relaxations of Mixed-Integer Optimal Control Problems: The 2-D Case. *IEEE Control. Syst. Lett.* **2021**, *6*, 548–553. [[CrossRef](#)]
43. Bestehorn, F.; Hansknecht, C.; Kirches, C.; Manns, P. Mixed-integer optimal control problems with switching costs: A shortest path approach. *Math. Program.* **2021**, *188*, 621–652. [[CrossRef](#)]
44. Manns, P. Relaxed multibang regularization for the combinatorial integral approximation. *SIAM J. Control. Optim.* **2021**, *59*, 2645–2668. [[CrossRef](#)]
45. Leyffer, S.; Manns, P. *Sequential Linear Integer Programming for Integer Optimal Control with Total Variation Regularization*; Technical Report. 2021. Available online: <https://arxiv.org/pdf/2106.13453.pdf> (accessed on 15 February 2022).
46. Buchheim, C.; Kuhlmann, R.; Meyer, C. Combinatorial optimal control of semilinear elliptic PDEs. *Comput. Optim. Appl.* **2018**, *70*, 641–675. [[CrossRef](#)]
47. Göttlich, S.; Hante, F.M.; Potschka, A.; Schewe, L. Penalty alternating direction methods for mixed-integer optimal control with combinatorial constraints. *Math. Program.* **2021**, *188*, 599–619. [[CrossRef](#)]
48. De Marchi, A. On the Mixed-Integer Linear-Quadratic Optimal Control With Switching Cost. *IEEE Control. Syst. Lett.* **2019**, *3*, 990–995. [[CrossRef](#)]

49. Sager, S. Reformulations and Algorithms for the Optimization of Switching Decisions in Nonlinear Optimal Control. *J. Process. Control.* **2009**, *19*, 1238–1247. [[CrossRef](#)]
50. Zeile, C. Combinatorial Integral Decompositions for Mixed-Integer Optimal Control. Ph.D. Thesis, Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany, 2021.
51. Zeile, C.; Robuschi, N.; Sager, S. Mixed-integer optimal control under minimum dwell time constraints. *Math. Program.* **2021**, *188*, 653–694. [[CrossRef](#)]
52. Sager, S.; Zeile, C. On mixed-integer optimal control with constrained total variation of the integer control. *Comput. Optim. Appl.* **2021**, *78*, 575–623. [[CrossRef](#)]
53. Manns, P.; Kirches, C. Improved Regularity Assumptions for Partial Outer Convexification of Mixed-Integer PDE-Constrained Optimization Problems. *ESAIM Control. Optim. Calc. Var.* **2020**, *26*, 32. [[CrossRef](#)]
54. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1989.
55. Fourer, R.; Gay, D.; Kernighan, B. *AMPL: A Modeling Language for Mathematical Programming*; Duxbury Press: 2002. Available online: <https://vanderbei.princeton.edu/307/textbook/AMPLbook.pdf> (accessed on 15 February 2022).
56. Buerger, A.; Zeile, C.; Altmann-Dieses, A.; Sager, S.; Diehl, M. Design, Implementation and Simulation of an MPC algorithm for Switched Nonlinear Systems under Combinatorial Constraints. *Process. Control.* **2019**, *81*, 15–30. [[CrossRef](#)]