

A numerical study of transformed mixed-integer optimal control problems

Sebastian Sager · Manuel Tetschke ·
Clemens Zeile

Received: date / Accepted: date

Abstract Time transformation is a ubiquitous tool in theoretical sciences, especially in physics. It can also be used to transform switched optimal control problems into control problems with a fixed switching order and purely continuous decisions. This approach is known either as enhanced time transformation, time-scaling, or switching time optimization (STO) for mixed-integer optimal control. The approach is well understood and used widely due to its many favorable properties. Recently, several extensions and algorithmic improvements have been proposed. We use an alternative formulation, the partial outer convexification (POC), to study convergence properties of (STO).

We introduce the open-source software package `ampl_mintoc`. It is based on AMPL, designed for the formulation of mixed-integer optimal control problems, and allows to use almost identical implementations for (STO) and (POC). We discuss and explain our main numerical result: (STO) is likely to result in more local minima for each discretization grid than (POC), but the number of local minima is asymptotically identical for both approaches.

Keywords Mixed-integer optimal control · switched systems · enhanced time transformation · switching time optimization · optimization software

Mathematics Subject Classification (2010) 49M37 · 90C20 · 90C30 · 90C53 · 90C55 · 97N80

Sebastian Sager

Otto-von-Guericke-Universität, Faculty of Mathematics, Magdeburg, Germany
Tel.: +49-391-6758745 E-mail: sager@ovgu.de

Manuel Tetschke

Otto-von-Guericke-Universität, Faculty of Mathematics, Magdeburg, Germany
Tel.: +49-391-6751449 E-mail: manuel.tetschke@ovgu.de

Clemens Zeile

Otto-von-Guericke-Universität, Faculty of Mathematics, Magdeburg, Germany
Tel.: +49-391-6751449 E-mail: clemens.zeile@ovgu.de

1 Introduction

In this study, we consider mixed-integer optimal control problems. The integer aspect results from time-dependent control functions that are restricted to take values in a finite set. Prominent examples include gear switches, compressors and valves, the applicability of laws and regulations, combinations of administered drugs, performing measurements yes or no, or general on/off controls. See [49] for an online benchmark library. There are various mathematically equivalent ways to formulate such control problems. The choice of formulation usually has a large impact on solution algorithms, though. A convenient formulation is based on switched optimal control problems, where at any given point t of time exactly one of n_ω different right-hand sides $f_{j(t)}(x(t))$ is active. In practice, often a common drift term $f_0(x(t))$ is present, such that the ordinary differential equation reads as

$$\dot{x}(t) = f_0(x(t)) + f_{j(t)}(x(t))$$

where j is the active *mode* at time $t \in \mathcal{T} := [0, t_N]$. Note, that the interval starts with 0 without loss of generality. For example, f_0 could comprise the dynamics that are independent of a gear choice, such as the effect of steering or air friction, whereas the gear-specific dynamics $f_{j(t)}$ could be calculated using different transmission ratios and degrees of efficiency, e.g., [27].

In this study, we assume Lipschitz-continuity of the right-hand side functions f_0 and $f_{j(t)}$ for guaranteeing the existence of a (unique) ODE solution by the Picard-Lindelöf theorem [44]. Throughout the paper, we use the notation $[N] = \{1, 2, \dots, N\}$ and $[N]_0 = \{0, 1, 2, \dots, N\}$. Also, we use the exclusive-or operator \oplus for choosing exactly one active mode for all time points and on the same time all other modes are inactive. We define the problem class of interest as follows.

Definition 1 (Mixed-integer optimal control problem) *We refer to*

$$\begin{aligned} & \inf_{x(\cdot), j(\cdot)} \quad \phi(x(t_N)) \\ & \text{subject to} \\ & \bigoplus_{j(t) \in [n_\omega]} \quad \dot{x}(t) = f_0(x(t)) + f_{j(t)}(x(t)) \quad \text{for } t \in \mathcal{T} \text{ a.e.,} \\ & \quad \quad \quad x(0) = x_0, \end{aligned} \tag{MIOCP}$$

as a *mixed-integer optimal control problem (MIOCP)* on a fixed time horizon $\mathcal{T} = [0, t_N]$ with differential states $x : \mathcal{T} \mapsto \mathbb{R}^{n_x}$, fixed initial values $x_0 \in \mathbb{R}^{n_x}$, and a continuous objective function $\phi : \mathbb{R}^{n_x} \mapsto \mathbb{R}$ of Mayer type. The degrees of freedom are switching decisions, i.e., the selection of the active mode j almost everywhere on \mathcal{T} .

In the following, we omit the time dependency of $f_{j(t)}$ and write f_j instead for a more compact notation and we assume the existence of a unique optimal solution for (MIOCP). There are a variety of different approaches to solve

(MIOCP) to local or global optimality. They comprise indirect methods based on the global maximum principle, dynamic programming, moment relaxations, or direct first-discretize then-optimize methods which result in mixed integer nonlinear programs. A survey is beyond the scope of this paper and we refer to [6, 16, 17, 23, 27, 52] for further references.

Note that the modeling of practical problems may result in more general formulations than Problem (MIOCP), e.g., using additional continuous control functions $u : [0, t_N] \mapsto \mathbb{R}^{n_u}$, additional path constraints $0 \leq c(x(t), u(t))$ (mode-specific or for all modes), a Lagrange term $\int_0^{t_N} L(x(\tau))d\tau$ in the objective, non-fixed terminal time t_N , more general boundary conditions, path-control, combinatorial, dwell time, or vanishing constraints, multi-stage problems with changes in dynamics or even in the dimension, differential algebraic equations, or semi-discretized partial differential equations. In the interest of a clear presentation of the convergence properties, which are likely to carry over to the more general cases, and because above formulation covers our benchmark problems, we work in the following with Problem (MIOCP).

In this paper, we are going to numerically study convergence properties of two established approaches to solve (MIOCP): *switching time optimization (STO)* and *partial outer convexification (POC)*. To our knowledge this is the first comparison of this type. In the interest of understanding the main differences between the two approaches, we use comparable implementations within a new open-source software package, similar notation and discretization grids, and two benchmark problems from the literature. The paper is organized as follows. In Section 2, we survey two transformations of (MIOCP). They are formulated in a way such that the small, but fundamental difference between them is clearly visible. We also discuss their relation to problem (MIOCP). We derive approximation properties of the two approaches in Section 3. In Section 4, two test problems are presented. In Section 5, we describe the novel open-source software package `ampl_mintoc`. It is placed on top of AMPL [13] and allows to formulate finite-dimensional approximations of (mixed-integer) optimal control problems in a compact way. In Section 6, we present simulation- and optimization-based numerical results for comparing the two approaches. In Section 7 we discuss these results, focusing on nonconvexity, algorithmic properties and number of local minima. A summary concludes the paper.

2 Switching Time Optimization and Partial Outer Convexification

In this paper, we focus on two transformations of (MIOCP) that have received increasing attention in the last years. In the following, let

$$\mathcal{G}_N := \{t_0, t_1, t_2, \dots, t_N\}$$

be an ordered set of equidistant time points $t_i = i\Delta t$ for all $i \in [N]_0$ and grid size $\Delta t > 0$.

2.1 Switching Time Optimization

The first one is referred to as *enhanced time transformation* or as *switching time optimization* in the literature. The underlying idea is fundamental in many sciences, particularly in physics, e.g., [7]. A dynamic process with differential states $x : [0, t_N] \mapsto \mathbb{R}^{n_x}$ specified via $f : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$ and

$$\dot{x}(t) = f(x(t)) \quad t \in [0, t_N] \quad (1)$$

can be written equivalently as

$$\dot{x}(\tau) = t_N f(x(\tau)) \quad \tau \in [0, 1], \quad (2)$$

exploiting that $t = t_N \tau$ implies $dt = t_N d\tau$ (in a slightly abused Leibniz's notation). The advantage is obvious if the terminal time t_N is unknown. Formulation (2) allows the treatment of t_N as an optimization variable, while the integration horizon is fixed. By gluing several horizons with fixed horizon lengths together, this idea can also be applied to several interior time points, e.g.,

$$\dot{x}(t) = \begin{cases} f_1(x(t)) & \text{if } t \in [0, t_1), \\ f_2(x(t)) & \text{if } t \in [t_1, t_2], \end{cases} \quad (3)$$

can be written equivalently as

$$\dot{x}(\tau) = \begin{cases} t_1 f_1(x(\tau)) & \text{if } \tau \in [0, 1), \\ (t_2 - t_1) f_2(x(\tau)) & \text{if } \tau \in [1, 2]. \end{cases} \quad (4)$$

Applying this idea to transform problem (MIOCP) needs only three more ingredients. First, we define two time grids in Definition 2. The second step is to assume an order in which the switches occur. The corresponding switching sequence is introduced in Definition 3. Finally, in Definition 4 we make sure that the variables indicating the duration of the active modes, denoted by w_{ij} , are normalized to get an equivalence to the outer grid time points of the original time horizon.

Definition 2 (Equidistant grids) Let numbers $N \in \mathbb{N}$ and $n_\sigma \in \mathbb{N}$ be given and let Δt be the equidistant outer grid size. We write

$$\mathcal{T}_i := [t_{i-1}, t_i), \quad i \in [N], t_i \in \mathcal{G}_N,$$

for a coarse grid partition $\mathcal{T} = \cup_{i \in [N]} \mathcal{T}_i$. Furthermore, let

$$\mathcal{T}_{ij} := [t_{i-1, j-1}, t_{i-1, j}), \quad i \in [N], j \in [n_\sigma]$$

for a fine grid partition $\mathcal{T}_i = \cup_{j \in [n_\sigma]} \mathcal{T}_{ij}$ with equidistant inner grid points $t_{ij} := t_i + j \frac{\Delta t}{n_\sigma}$ and inner grid size $\Delta_{\text{int}} t := \frac{\Delta t}{n_\sigma}$. We include t_N in the last outer and inner intervals \mathcal{T}_N and $\mathcal{T}_{N, n_\sigma}$, respectively.

We stress that despite fixed inner and outer grids, we still optimize the duration of mode activations based on the control variables defined in Definition 4. For this, we specify the sequence of active control modes.

Definition 3 (Switching sequence) Let $n_\sigma \in \mathbb{N}$ be the maximum number of control mode activations per outer grid interval. We define a switching sequence as a surjective mapping $\sigma : [n_\sigma] \mapsto [n_\omega]$ on \mathcal{T}_i and, repeated N times, also on \mathcal{T} .

The above definition implies that at most $n_\sigma - 1$ switches are possible per outer interval. The choice $n_\sigma = n_\omega$ allows to activate each control mode once on every interval, in a specific order. With larger choices of n_σ arbitrary switching sequences can be generated by shrinking the duration of specific mode activations to zero.

Definition 4 (Feasible control set) For a given number of intervals N and maximum number of control mode activations n_σ , we define the feasible set of controls \mathcal{W} as:

$$\mathcal{W} := \left\{ w \in \mathbb{R}^{N \times n_\sigma} : w_{ij} \in [0, 1] \ \forall i \in [N], j \in [n_\sigma], \sum_{j=1}^{n_\sigma} w_{ij} = 1 \ \forall i \in [N] \right\}.$$

The variables w_{ij} indicate how the time points of the inner grid move and can thus be interpreted as the fraction of the duration $w_{ij} \Delta t$ of the activated mode $\sigma(j)$ on the interval \mathcal{T}_i . The condition $\sum_{j=1}^{n_\sigma} w_{ij} = 1$, which is often referred to as *one-hot condition*, ensures that the outer grid points t_i are not moved by the time-transformation.

We consider the differential equation with a transformed time τ on time intervals of unit length as in (4), with time scaling depending on w_{ij} for all $i \in [N]$ and $j \in [n_\sigma]$:

$$\dot{x}(\tau) = w_{ij} \Delta t (f_0(x(\tau)) + f_{\sigma(j)}(x(\tau))), \quad \tau \in [(i-1)n_\sigma + j - 1, (i-1)n_\sigma + j]. \quad (5)$$

For comparability of the problem formulations, we scale the total length of the intervals for τ to the length of \mathcal{T} . This is equivalent to scaling the individual intervals for τ of length 1 to the length of \mathcal{T}_{ij} via multiplication with $\frac{1}{\Delta_{\text{int}}}$. Exploiting $n_\sigma = \frac{\Delta t}{\Delta_{\text{int}}}$ according to Definition 2 and using t again instead of τ to denote the time variable, we introduce the problem (**P-STO**):

Definition 5 (Switching time optimization) With the definitions from above we refer to

$$\begin{aligned} & \min_{x, w \in \mathcal{W}} \phi(x(t_N)) \\ & \text{subject to} \\ & \quad \dot{x}(t) = w_{ij} n_\sigma (f_0(x(t)) + f_{\sigma(j)}(x(t))), \quad t \in \mathcal{T}_{ij}, \\ & \quad x(0) = x_0, \end{aligned} \quad (\text{P-STO})$$

for $i \in [N]$ and $j \in [n_\sigma]$ as the switching time optimization (P-STO) (or the enhanced time or time-scaling) transformation of (**MIOCP**).

Please note that the way in which we define the switching time optimization transformation differs from the literature, e.g., [57], although we use the same algorithmic approach. The definition above enables us to highlight the major difference to the partial outer convexification formulation introduced below. The switching time optimization is well-posed: assume a solution \tilde{x}, \tilde{w} of (P-STO). Noting that $n_\sigma = \frac{\Delta t}{\Delta_{\text{int}}}$ indicates the time transformation factor from the equidistant inner to the outer grid, we obtain a feasible trajectory x of (MIOCP) via

$$\begin{aligned} x(t(\tau)) &= \tilde{x}(\tau) \quad \text{for } \tau \in \mathcal{T}_{ij} \text{ and} \\ t(\tau) &= t_{i-1} + \sum_{k=1}^{j-1} w_{ik} \Delta t + (\tau - t_{i-1,j-1}) w_{ij} n_\sigma \in \mathcal{T}_{ij}^* \end{aligned}$$

for time transformed intervals

$$\mathcal{T}_{ij}^* := \left[t_{i-1} + \sum_{k=1}^{j-1} w_{ik} \Delta t, t_{i-1} + \sum_{k=1}^j w_{ik} \Delta t \right]$$

that also partition \mathcal{T} . Note that \tilde{x} may be constant on certain time periods (i.e., whenever $w_{ij} = 0$), whereas the corresponding intervals \mathcal{T}_{ij}^* vanish. This is illustrated in the following short example.

Example 6 (Illustration of the time transformation) *Consider an outer grid with $N = 3$ intervals. Let $n_\sigma = 3$ and the switching sequence σ be the identity. Then the inner grid consists in total of 9 intervals. Consider the values*

$$w_1 := (0, 1, 0), \quad w_2 := \left(\frac{3}{5}, 0, \frac{2}{5} \right), \quad w_3 := \left(\frac{1}{5}, \frac{4}{5}, 0 \right) \in [0, 1]^{n_\sigma}.$$

Figure 1 illustrates the corresponding switches and the transformation in time.

For optimal control, time transformations have been used intensively. In indirect approaches the unknown time points of structural changes have to be determined, e.g., when a bang-bang arc becomes a singular arc or a path constraint becomes active. Also, the potential to avoid the discrete decision of which mode to activate at time t in problem (MIOCP) by continuous decisions when to switch from one mode f_{j_1} to another mode f_{j_2} was already discovered in the 1960s. Dubovitskii and Milyutin used a time transformation to transform (MIOCP) into an equivalent optimal control problem without discrete control variables, [9, 10], [18, p. 95], [24, p. 148]. Necessary optimality conditions were obtained by applying suitable local maximum principles to the transformed problem. Gerdt used the time transformation to prove a global maximum principle that can be applied to mixed-integer optimal control problems [16, 17]. An STO formulation was also used for discretized dynamical systems [12], the numerical solution of free end-time [28], bi-objective [38, 39], multiple model stages [62], constrained MIOCPs [47, 59],

or real-time problems [36, 37], and was studied concerning specific numerical aspects [33, 34, 35, 55]. The method has been investigated with respect to second-order sufficient conditions [42] and recently been extended to include vanishing constraints [43] and time delays [63]. An efficient structure exploiting way to calculate functions and derivatives was proposed in [57]. The mapping σ , which in our setting is assumed to be identical on all intervals \mathcal{T}_i for notational convenience, also has an impact. As stated in [17], the order is irrelevant for (P-STO) to be an equivalent reformulation of (MIOCP) if N is large enough and hence no more than one switching point of the optimal bang-bang solution occurs per interval \mathcal{T}_i . However, σ may have an impact on computational performance as discussed in [46]. An efficient way to determine optimal sequences has been proposed in [1, 11].

2.2 Partial Outer Convexification

An alternative approximation of (MIOCP) is the following, using the same outer time grid \mathcal{T}_i as in (P-STO) for control discretization and relaxation of the integrality constraints.

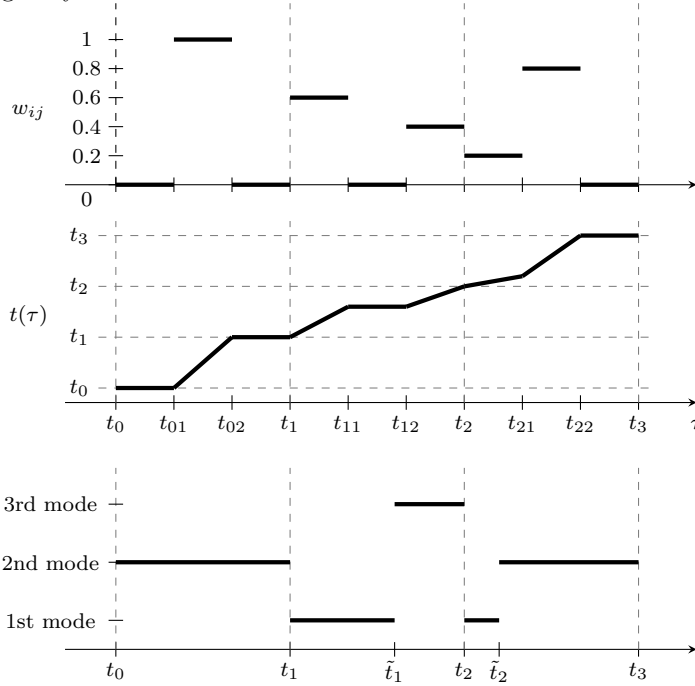


Fig. 1 Illustration of the time transformation with the control values w_{ij} for Example 6, adapted from [17]. Top: The control values indicate the percentage of control mode activation for the corresponding outer interval and for the three control modes. Center plot: Corresponding progress of the transformed time variable $t \in \mathcal{T}_{ij}^*$ as a function of $\tau \in \mathcal{T}_{ij}$. Bottom: Resulting activated control modes in (MIOCP) over time that correspond to the values w_{ij} . The mode switches occur at $\tilde{t}_1 = t_1 + w_{21}(t_2 - t_1)$ and $\tilde{t}_2 = t_2 + w_{31}(t_3 - t_2)$.

Definition 7 (Partial outer convexification) *With the definitions from above (using $n_\sigma = n_\omega$ in Definition 4 of \mathcal{W}) we refer to*

$$\begin{aligned} & \min_{x, w \in \mathcal{W}} \phi(x(t_N)) \\ & \text{subject to} \\ & \quad \dot{x}(t) = \sum_{j=1}^{n_\omega} w_{ij} (f_0(x(t)) + f_j(x(t))), \quad t \in \mathcal{T}_i \text{ for } i \in [N] \\ & \quad x(0) = x_0, \end{aligned} \quad (\text{P-POC})$$

as the partial outer convexification (P-POC) transformation of (MIOCP).

Note that *partial* refers to the fact that the dynamics are only convexified with respect to the switching decision, but the f_j may still be nonconvex. *Outer* is in contrast to an inner convexification as suggested, e.g., in [15], and compared in [25]. The formulation itself is well-known in many areas. In integer programming it is usually referred to as a *one-row* or *one-hot relaxation*. The usage in mixed-integer optimal control and the name itself were initiated by [47, 50]. In the control engineering community, the term *embedding transformation* was introduced [2] for the same reformulation idea and led to various subsequent publications [45, 60, 61]. As discussed in the next subsection, (P-POC) only provides relaxed solutions for a discretized version of (MIOCP). Methods for the generation of integer solutions have been proposed in [47, 48, 50, 54]. Extensions comprise the explicit treatment of combinatorial [26, 40, 53, 64] and vanishing constraints [27, 29], of differential-algebraic equations [17], and of partial differential equations [19, 20, 21, 22, 31, 41].

To highlight the similarities and differences to formulation (P-STO), we have defined (P-POC) in a first-discretize, then-optimize way with finitely many degrees of freedom. Here, the controls w can be interpreted as piecewise constant functions $w_j(t) = w_{ij}$ for $t \in \mathcal{T}_i$ almost everywhere.

We recapitulate two methods that round the optimal solution of (P-POC). We use these methods in the numerical experiments for a fair comparison with the STO approach, which directly constructs binary-feasible solutions for (MIOCP).

Definition 8 (Sum-up rounding) *Let the optimal w^* for (P-POC) be given. Sum-up rounding computes for $i \in [N]$ the binary control values $b_{i,j}$:*

$$b_{i,j} := \begin{cases} 1, & \text{if } j = \arg \max_{l=1, \dots, n_\omega} \left\{ \sum_{k=1}^i w_{k,l}^* - \sum_{k=1}^{i-1} b_{k,l} \right\}, \\ 0, & \text{else,} \end{cases} \quad \text{for } j \in [n_\omega].$$

We break ties arbitrarily in the above $\arg \max$.

One option to limit the number of switches in the binary solution is to impose total variation (TV) constraints as part of a mixed-integer linear program.

Definition 9 ((CIA) with TV constraint) *Let the optimal w^* for (P-POC) and the number of maximal allowed number of switches n_s be given. The combinatorial integral approximation (CIA) problem with TV constraints is defined by*

$$\begin{aligned} \min_{b \in \mathcal{W}, b_{i,j} \in \{0,1\}, \theta \geq 0} \quad & \theta \\ \text{s. t.} \quad & \theta \geq \pm \sum_{l \in [i]} (w_{l,j}^* - b_{l,j}), \quad \text{for } i \in [N], j \in [n_\omega]. \\ & n_s \geq \frac{1}{2} \sum_{i \in [N-1]} \sum_{j \in [n_\omega]} |b_{i+1,j} - b_{i,j}|. \end{aligned}$$

2.3 Relations to the Original Problem Formulation

Both problem formulations (P-STO) and (P-POC) are approximations of (MIOCP). Both are discretizing (MIOCP), which is usually modeled with $\mathcal{L}^\infty(\mathcal{T}, \{0, 1\}^{n_\omega})$ control functions, with finitely many decision variables w . In addition to this finite-dimensional approximation which depends on N , the formulation (P-POC) additionally relaxes the decision variables, as \mathcal{W} is defined with $w_{ij} \in [0, 1]$ instead of $w_{ij} \in \{0, 1\}$.

The optimal solution of (P-STO) is a feasible solution $\omega(\cdot)$ of the original problem (MIOCP) (compare ω in the proof of Corollary 12). Depending on the number and location of switches in the optimal solution of (MIOCP), the numbers N and n_σ need to be chosen sufficiently large to ensure that the optimal switching points of (MIOCP) can be realized within (P-STO). In the case that the optimal solution shows chattering behavior (i.e., an infinite number of switches as in the famous example of Fuller [14]), the optimal objective function value can be approximated arbitrarily well with increasing N or n_σ .

There is a bijection between feasible solutions of formulations (P-POC) and (MIOCP) if one replaces $w_{ij} \in [0, 1]$ by $\omega_j(t) \in \{0, 1\}$ and one uses the same discretization on (MIOCP). Formulated for continuous variables w_{ij} , (P-POC) only provides a relaxed solution for the control discretization grid \mathcal{G}_N , however with strong theoretical properties and tailored algorithms like *Combinatorial Integral Approximation*, *Sum Up Rounding (SUR)*, or *Next Forced Rounding* to derive binary solutions with error bounds, see [47, 48, 50, 54]. The need to derive $\omega_j(t) \in \{0, 1\}$ from $w_{ij} \in [0, 1]$ seems like a big disadvantage at first sight. However, the aforementioned rounding strategies have excellent asymptotic behavior with respect to the integrality gap and a runtime which is often negligible compared to the solution of (P-POC). The need to transform the solution back even offers advantages, e.g., to take a penalization of switches or dwell time constraints into account [3, 4, 64].

2.4 A First Comparison

Comparing **(P-STO)** and **(P-POC)**, one observes a small, but important difference in the dynamics. For $i \in [N]$ and $j \in [n_\omega]$ we have within **(P-STO)**

$$\dot{x}(t) = w_{ij} n_\sigma (f_0(x(t)) + f_{\sigma(j)}(x(t))), \quad t \in \mathcal{T}_{ij}, \quad x(0) = x_0 \quad (\text{STO-f})$$

while due to the one-hot condition in \mathcal{W} we have within **(P-POC)**

$$\dot{x}(t) = f_0(x(t)) + \sum_{j=1}^{n_\omega} w_{ij} f_j(x(t)), \quad t \in \mathcal{T}_i, \quad x(0) = x_0. \quad (\text{POC-f})$$

The way the control variables w enter the right-hand sides look very similar in **(STO-f)** and **(POC-f)**. Figure 2 gives an example to visualize how they differ in their impact on the original dynamic system.

Exemplary STO and POC control functions

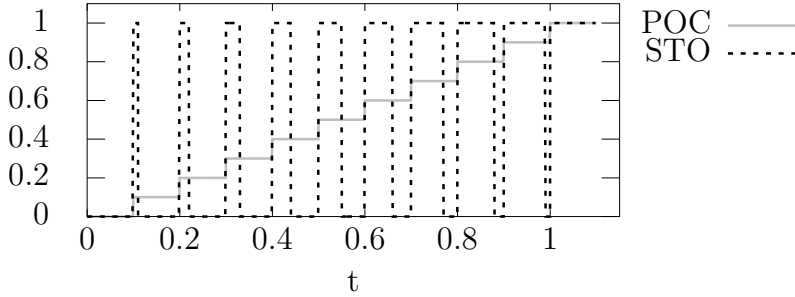


Fig. 2 Shown is the impact of the controls on the mode selection in **(MIOCP)** for STO and POC controls $w_{i1}^{\text{STO}} = w_{i1}^{\text{POC}} = 0.1(i - 1)$ for $i \in [11]$. The values of w_{i1}^{STO} are reflected by the duration of the mode selections (width), whereas the values of w_{i1}^{POC} are reflected by the relative share of mode 1 on the current time interval (height). The other $n_\sigma - 1$ modes would fill the diagram vertically **P-POC** or horizontally **P-STO**, if they were also plotted.

Problems **(P-STO)** and **(P-POC)** optimize the same objective function over the finite-dimensional set \mathcal{W} . For both, the accuracy of the approximation of **(MIOCP)** depends on the grid \mathcal{G}_N .

3 Approximation properties

Behind this asymptotic analysis an approximation theorem from [50] will be useful. It also can be used for the study of how far optimal objective function values of **(P-STO)** and **(P-POC)** may differ from one another depending on N . We repeat it here for convenience and use it in the discussion in Section 7.

Assumption 10 (Properties of the switched system) *Let measurable functions $\alpha, \omega : [0, t_N] \rightarrow [0, 1]^{n_\omega}$ be given and let $x(\cdot)$ and $y(\cdot)$ be solutions of the initial value problems on $\mathcal{T} = [0, t_N]$*

$$\dot{x}(t) = f_0(x(t)) + \sum_{j=1}^{n_\omega} \alpha_j(t) f_j(x(t)), \quad x(0) = x_0, \quad (6)$$

$$\dot{y}(t) = f_0(y(t)) + \sum_{j=1}^{n_\omega} \omega_j(t) f_j(y(t)), \quad y(0) = x_0. \quad (7)$$

Let all $f_j : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$ be differentiable for $j \in [n_\omega]_0$ and their sum be essentially bounded by $M \in \mathbb{R}^+$ on $[0, t_N]$, and positive numbers $C, L \in \mathbb{R}^+$ exist such that for $t \in [0, t_N]$ almost everywhere and a vector norm $\|\cdot\|$ we have

$$\left\| \frac{d}{dt}(f_0, f_1, f_2, \dots, f_{n_\omega})(x(t)) \right\| \leq C, \\ \|(f_1(y(t)) - f_1(x(t)), \dots, f_{n_\omega}(y(t)) - f_{n_\omega}(x(t)))\| \leq L \|y(t) - x(t)\|.$$

Theorem 11 *If Assumption 10 holds and if there exists $\varepsilon \in \mathbb{R}^+$ such that for all $t \in [0, t_N]$*

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) d\tau \right\| \leq \varepsilon, \quad (8)$$

then, also for all $t \in [0, t_N]$, we have

$$\|y(t) - x(t)\| \leq (M + Ct)e^{Lt} \varepsilon.$$

To understand the asymptotic behavior of the optimal objective function values of (P-STO) and (P-POC), the following corollaries will be helpful. Theorem 11 allows to a priori bound the difference of the optimal objective function values of (P-STO) and (P-POC) by a constant multiple (depending on properties of f_j) of the grid size Δt .

Corollary 12 (Difference in objective and constraints 1) *Let $n_\sigma \in \mathbb{N}$, a map $\sigma : [n_\sigma] \mapsto [n_\omega]$, and $w^{STO} \in \mathcal{W}$ be given and differential states $x^{STO} : [0, t_N] \mapsto \mathbb{R}^{n_x}$ be the solution of (STO-f) with $w = w^{STO}$.*

Under Assumption 10, there exist a constant C_1 and a vector $w^{POC} \in \mathcal{W}$ such that for the solution $x^{POC} : [0, t_N] \mapsto \mathbb{R}^{n_x}$ of (POC-f) with $w = w^{POC}$ we have

$$|\phi(x^{STO}(t_N)) - \phi(x^{POC}(t_N))| \leq C_1 \Delta t.$$

Proof As already observed above, we have an $x^{STO} = x$ equivalence

$$\begin{aligned} \dot{x}^{STO}(\tau) &= w_{ij}^{STO} n_\sigma (f_0(x^{STO}(\tau)) + f_{\sigma(j)}(x^{STO}(\tau))), \quad \tau \in \mathcal{T}_{ij} \\ \Leftrightarrow \dot{x}(t) &= f_0(x(t)) + f_{\sigma(j)}(x(t)), \quad t \in \mathcal{T}_{ij}^* \end{aligned}$$

between **(P-STO)** and **(MIOCP)** solutions for transformed time intervals

$$\mathcal{T}_{ij}^* := \left[t_{i-1} + \sum_{k=1}^{j-1} w_{ik}^{\text{STO}} \Delta t, t_{i-1} + \sum_{k=1}^j w_{ik}^{\text{STO}} \Delta t \right].$$

The functions

$$\omega_k(t) := \begin{cases} 1 & \text{if } t \in \mathcal{T}_{ij}^* \text{ for } j \in [n_\sigma] \text{ and } k = \sigma(j) \in [n_\omega] \\ 0 & \text{else} \end{cases}$$

are obviously well-defined and have the one-hot property, such that

$$\begin{aligned} \dot{x}(t) &= f_0(x(t)) + \omega_{\sigma(j)}(t) f_{\sigma(j)}(x(t)), \quad t \in \mathcal{T}_{ij}^* \\ &= f_0(x(t)) + \sum_{k=1}^{n_\omega} \omega_k(t) f_k(x(t)), \quad t \in \mathcal{T}_{ij}^*. \end{aligned}$$

Defining

$$\alpha_j(t) := w_{ij}^{\text{POC}} \text{ for } t \in \mathcal{T}_i$$

we have two measurable functions $\alpha, \omega : [0, t_N] \mapsto [0, 1]^{n_\omega}$ and differential equations of the form (6–7) such that Assumption 10 is satisfied. To apply Theorem 11, we choose w^{POC} in a particular way. With

$$w_{ij}^{\text{POC}} := \int_{t_{i-1}}^{t_i} \omega_j(\tau) \, d\tau$$

it follows $\left\| \int_0^{t_i} \alpha(\tau) - \omega(\tau) \, d\tau \right\| = 0$ for all $i \in [N]$. The maximum over all $t \in [0, t_N]$ can hence not exceed the integral of length Δt over extreme cases such as $\alpha \equiv 1$ and $\omega \equiv 0$, thus

$$\max_{t \in \mathcal{T}} \left\| \int_0^t \alpha(\tau) - \omega(\tau) \, d\tau \right\| \leq \varepsilon := \Delta t.$$

Theorem 11 now gives a bound $\|x^{\text{POC}}(t) - x^{\text{STO}}(t)\| \leq \hat{C} \Delta t$ for all $t \in \mathcal{T}$ which carries over to the objective function via assumed continuity. This concludes the proof. \square

Given the symmetry of Theorem 11, the reverse is also true.

Corollary 13 (Difference in objective and constraints 2) *Let $w^{\text{POC}} \in \mathcal{W}$, $x^{\text{POC}} : [0, t_N] \mapsto \mathbb{R}^{n_x}$ be the solution of **(POC-f)** with $w = w^{\text{POC}}$.*

*Under Assumption 10, there exist a constant C_1 , a number n_σ , a switching order $\sigma : [n_\sigma] \mapsto [n_\omega]$, and a vector $w^{\text{STO}} \in \mathcal{W}$ such that for the solution $x^{\text{STO}} : [0, t_N] \mapsto \mathbb{R}^{n_x}$ of **(STO-f)** with $w = w^{\text{STO}}$ we have*

$$|\phi(x^{\text{STO}}(t_N)) - \phi(x^{\text{POC}}(t_N))| \leq C_1 \Delta t.$$

Proof The claim can be proved analogously to the proof of Corollary 12. Here n_σ , σ , and w^{STO} can be constructed with the well-known bound that is linear in Δt using SOS1 Sum Up Rounding [47, 50]. \square

Both corollaries together show that the optimal objective function values of (P-STO) and (P-POC) are close with respect to Δt . Therefore, one can expect that for larger N the objective function landscapes look very similar for the transformations (P-STO) and (P-POC).

Moreover, from Theorem 11 in conjunction with Corollaries 12 and 13, it can be deduced that the (local) optima of (P-STO) and (P-POC) are ϵ -optima with respect to the modified problem (P-POC) in which w_{ij} is replaced by the non-discretized $w_j(t)$. Since this modified (P-POC) is a true relaxation of (MIOCP), this implies the minima of (P-STO) and the rounded minima of (P-POC) are also ϵ -optima of (MIOCP).

4 Two Test Problems

We are going to compare numerical properties of (P-STO) and (P-POC) in Section 6, using the new software package `ampl_mintoc` introduced in Section 5. The numerical results are obtained for two test mixed-integer optimal control problems that we shortly present in this section. In this study, we limit ourselves to two benchmark problems, since a more detailed analysis of other problems is beyond the scope of this paper. We have specifically chosen two structurally different problems. The optimal solution of the first problem consists of singular and bang-bang arcs, which renders its solution to be challenging, while the other involves only bang-bang arcs. Moreover, the problems have different convexity properties. One problem is convex over a large part of the domain and, thus, in this sense relatively easy to solve, while the other problem is very nonconvex and raises numerically more difficulties. The Lotka-Volterra fishing benchmark problem was first formulated in [51]. The numerically constructed optimal relaxed solution has bang-bang and singular arcs, which can be derived using Pontryagin's maximum principle [47]. The (P-POC) formulation is convex on large parts of the domain and for many choices of the discretization grid, initial values, and bounds, as we will show in the numerical results in Section 6 in Figure 7. The second problem, a calcium signaling pathway inhibition, was formulated in [32]. The numerically constructed optimal relaxed solution consists of three bang-bang arcs. Both the (P-POC) and the (P-STO) formulation are nonconvex. Both problems have oscillatory dynamics and only use one binary control function. Further references and details can be found in [49]. Here, we only state and use the mathematical formulations of these problems.

Definition 14 (Lotka problem) We refer to

$$\begin{aligned}
& \min_{x, \omega} x_3(t_N) \\
& \text{s.t. } \dot{x}_1(t) = x_1(t) - x_1(t)x_2(t) - c_1x_1(t)\omega(t) \\
& \quad \dot{x}_2(t) = -x_2(t) + x_1(t)x_2(t) - c_2x_2(t)\omega(t) \\
& \quad \dot{x}_3(t) = \sum_{i=1}^2 (x_i(t) - \tilde{x}_i)^2 \\
& \quad x(t_0) = (0.5, 0.7, 0.0), \quad \tilde{x} = (1, 1) \\
& \quad \omega(t) \in \{0, 1\}
\end{aligned} \tag{Lotka}$$

as the Lotka-Volterra fishing control problem on $\mathcal{T} = [0, t_N = 12]$ with $c_1 = 0.4$ and $c_2 = 0.2$.

Definition 15 (Calcium problem) We refer to

$$\begin{aligned}
& \min_{x, \omega} x_5(t_N) \tag{Calcium} \\
& \text{s.t. } \dot{x}_1(t) = k_1 + k_2x_1(t) - \frac{k_3x_1(t)x_2(t)}{x_1(t)+K_4} - \frac{k_5x_1(t)x_3(t)}{x_1(t)+K_6} \\
& \quad \dot{x}_2(t) = k_7x_1(t) - \frac{k_8x_2(t)}{x_2(t)+K_9} \\
& \quad \dot{x}_3(t) = \frac{k_{10}x_2(t)x_3(t)x_4(t)}{x_4(t)+K_{11}} + k_{12}x_2(t) + k_{13}x_1(t) - \frac{k_{16}x_3(t)}{x_3(t)+K_{17}} + \frac{x_4(t)}{10} \\
& \quad \quad - \omega \frac{k_{14}x_3(t)}{\bar{u}^{\max} x_3(t)+K_{15}} - (1 - \omega(t)) \frac{k_{14}x_3(t)}{\bar{u}^{\min} x_3(t)+K_{15}} \\
& \quad \dot{x}_4(t) = -\frac{k_{10}x_2(t)x_3(t)x_4(t)}{x_4(t)+K_{11}} + \frac{k_{16}x_3(t)}{x_3(t)+K_{17}} - \frac{x_4(t)}{10} \\
& \quad \dot{x}_5(t) = \sum_{i=1}^4 \left(\frac{x_i(t) - \tilde{x}_i}{\tilde{x}_i} \right)^2 + 100 \omega(t) \\
& \quad x(t_0) = (0.03966, 1.09799, 0.00142, 1.65431, 0) \\
& \quad \omega(t) \in \{0, 1\}
\end{aligned}$$

as the Calcium control problem on $\mathcal{T} = [0, t_N = 22]$ with data $\bar{u}^{\min} = 1$, $\bar{u}^{\max} = 1.3$, $k_1 = 0.09, k_2 = 2.30066, k_3 = 0.64, K_4 = 0.19, k_5 = 4.88, K_6 = 1.18, k_7 = 2.08, k_8 = 32.24, K_9 = 29.09, k_{10} = 5.0, K_{11} = 2.67, k_{12} = 0.7, k_{13} = 13.58, k_{14} = 153.0, K_{15} = 0.16, k_{16} = 4.85, K_{17} = 0.05$, and reference unstable steady state $\tilde{x}_1 = 6.78677, \tilde{x}_2 = 22.65836, \tilde{x}_3 = 0.38431, \tilde{x}_4 = 0.28977$.

Both control problems are of the form (MIOCP), where the $n_\omega = 2$ different modes correspond to values $\omega(t) = 0$ and $\omega(t) = 1$, respectively. They use a single binary control, as the second control was eliminated via the one-hot constraint.

5 The Software Package `ampl_mintoc`

We have designed the AMPL [13] code `ampl_mintoc` for the formulation of Mixed INTeger Optimal Control problems, in particular for problems of type (MIOCP). The package `ampl_mintoc` is available as open-source on GitHub¹. We explain

¹ `ampl_mintoc` is available under https://github.com/czeile/ampl_mintoc.

the concept and idea of the software in Section 5.1. In Section 5.2, we first describe the most relevant problem-independent files and explain the different algorithmic options and discretizations. We then dedicate the remaining section to the implementation of the transformations (P-STO) and (P-POC). We also provide an explanation of how optimal control problems can be formulated and solved with `ampl_mintoc` in Appendix A.1.

5.1 Conceptual Description

`ampl_mintoc` consists of a set of code files that allow the user to efficiently formulate control problems. These are solved by off-the-shelf NLP or MINLP solvers interfaced by AMPL. In particular, problem-independent code exists to apply (P-STO) as well as (P-POC) formulations to problem-dependent functions.

The application of `ampl_mintoc` is illustrated in Figure 3. The goal of this software is to be able to formulate and solve optimal control problems easily. For this purpose, there is a set of problem-independent code files that define generic parameters, variables, problems, and transformations that can be used by the user depending on the application. For formulation of an MIOCP, the user needs to specify files `problem.mod` and `problem.dat` to define the objective function and constraints as well as parameter values, respectively. In the file `problem.run` algorithmic choices are set, such as the choice of the solver, the integrator, and the problem transformation. To this end, problem-independent files, e.g. `mintocPre.mod`, `mintocPost.mod`, and `solve.run`, are included in this file to reuse generic structures.

After applying the numerical solver interfaced via AMPL, the constructed optimal solution can be visualized with plot routines from the software package and `gnuplot`. The obtained solution can also be used for further optimization steps as specified in `problem.run`. For instance, after solving (P-POC), one could be interested in computing a rounded solution via SUR or Combinatorial Integral Approximation [53].

The advantage of using AMPL is that the syntax remains close to the mathematical formulation. Due to the chosen direct approach for the solution of the OCPs, the implementation is done directly in a discretized setting. As a disadvantage, the usage of AMPL implies that function pointers are not straightforward to use. However, the discretized problem formulation can also be beneficial, since the program's preprocessor can eliminate variables and constraints from discretized functions before the solution process.

We note that there are other AMPL packages for optimal control, such as TACO [30] that interfaces a multiple shooting code for optimal control. The novel feature of `ampl_mintoc` is the possibility for rapid prototyping of transformation and decomposition algorithms and to facilitate the formulation and solution of MIOCPs, by using the flexibility of solver choice provided with AMPL.

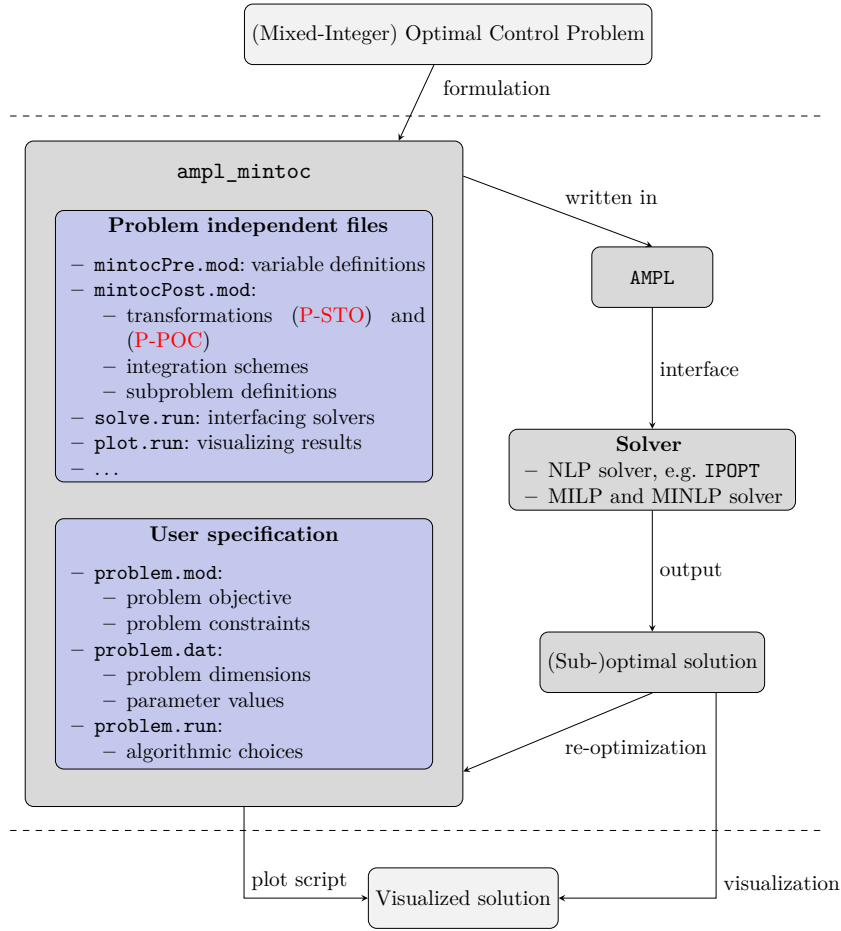


Fig. 3 Flowchart overview of the application of `ampl_mintoc`. The user can formulate an MIOCP in this code by creating problem-specific files, as listed in the lower box. `ampl_mintoc` provides a set of problem-independent files that implement generic routines and which simplify the problem formulation, as listed in the upper box. The modeling language is **AMPL**, which facilitates to interface a wide range of numerical solvers. The constructed optimal solution can be visualized or used for further optimization steps by using `ampl_mintoc` code routines. The dashed lines mark the software related area without input and output.

5.2 Details on Problem-independent Files

In order to save implementation effort, problem-independent variables and generic problem structures are outsourced to dedicated **AMPL** files. The file `mintocPre.mod` contains the declaration of general parameters such as the time discretization and the number of differential states and controls. It also defines the variables for the ODE model function, the control values, differential states, and discretized constraint functions. In `mintocPost.mod`, we define different

objectives, constraints, numerical integration schemes for the ODEs, and the formulation of NLPs and Combinatorial Integral Approximation [65] problems.

The files `mintocPre.mod` and `mintocPost.mod` can be considered as the core of `ampl_mintoc`. Both files need to be included in the problem-specific run file and reduce the effort to formulate the optimal control problems significantly.

Currently available numerical integrator schemes are the Radau collocation and explicit and implicit Euler methods, which can be set via the variable `integrator`. Various problem variants can be solved via setting the variable `mode`. For instance, in the `Simulate` mode, all control variables are fixed and in the `Relaxed` mode the optimization problem is solved with relaxed integer variables. The file `solve.run` contains the commands to interface and run the NLP solver based on the chosen algorithmic options. Thus, one needs to use `include ../solve.run` in the problem-specific run file to let the problem at hand be solved. The files `solveMILP.run` and `solveRound.run` work similarly as `solve.run`, but interface MILP solvers for specific combinatorial integral approximation problems [53] and contain specific rounding algorithms, respectively, as part of the Combinatorial Integral Approximation decomposition [65]. The declaration of default settings for solvers, problem classes, and integrators are outsourced in the file `set.run`, while `checkConsistency.run` verifies if valid options/modes have been chosen for solver, integrator, and discretization.

Standardized routines are implemented in the `ampl_mintoc` code to post-process, store, and restore problem solutions. These can be retrieved via the file `problem.run` and the AMPL command `include`. For instance, by calling the script `printOutput.run`, detailed solution results can be output, such as the objective function value, constraint violation metrics, and the run time. If the user includes the script `plot.run`, the determined solution is stored in a file `problem.plot`, which in turn can be used by `gnuplot` to plot result data such as the state or control trajectories. Furthermore, one can use the scripts `storeTrajectory.run` and `readTrajectory.run` to save or read problem solutions in order to solve a problem again later in a modified form. The problem-independent files do not need to be modified by users, but can be extended by algorithm developers.

We illustrate the implementation of the problem-independent formulations of the ODE constraints for STO and POC in the Listing 1. The presented implementation is done in `mintocPost.mod` and is based on a discretization with an explicit Euler scheme. We note that the (discretized) ODE equations in `ampl_mintoc` closely resemble the corresponding mathematical ODE equations from (P-STO) and (P-POC).

Listing 1 AMPL model file with problem-independent ODE constraint formulations for (P-STO) and (P-POC) and explicit Euler integration scheme

```
### Switching Time Optimization
ode_STO {k in X, o in 1..no, i in IU, ii in 0..nsto-1}:
  x[k,i*ntperu+(o-1)*nsto+ii+1] = (
    if (integrator=="explicitEuler") then
      x[k,i*ntperu+(o-1)*nsto+ii] + wi[o,i] * dt*no *
```

```

        (f[k,0,i*ntperu+(o-1)*nsto+ii] + f[k,o,i*ntperu+(o-1)*nsto+ii])
    );

### Partial Outer Convexification
ode_POC {k in X, i in 0..nt-1}:
    x[k,i+1] = (
        if (integrator=="explicitEuler") then
            x[k,i] + dt * (f[k,0,i] + sum {o in Omega} w[o,i] * f[k,o,i])
        );

```

6 Numerical Experiments

We are going to investigate the two test problems from Section 4 numerically. Special focus is given on nonconvexity of the objective functions over \mathcal{W} . We present first simulation-based analyses. To be able to plot results, we consider low dimensions, projections, and the concept of mid-point convexity, respectively. We complement the findings with optimization-based analyses, comparing the number of optimal solutions that are found for random initial values. An interesting approach is to initialize (P-STO) with the binary solution determined from (P-POC), which we test as part of this investigation. We also use different algorithms to compare integer solutions, that can be derived using the solution of (P-POC). In the following, we are only inspecting local optimality up to the precision of the numerical method.

6.1 Discretization and Numerical Integration

All numerical results were produced using a machine with 160 Intel CPU cores (2 GHz each) and 1 TB of RAM running Ubuntu 18.04.5 LTS. We implemented the Problems (P-STO) and (P-POC) in `ampl_mintoc` with a direct collocation approach and used IPOPT 3.12.13 to solve the resulting NLPs. In the following, we refer to the determined optima of IPOPT as local minima, neglecting the fact that in theory the determined stationary points can also be saddle points. In practice we did not observe such behavior. The underlying motivation was to get as close to objective comparability as possible, ignoring issues of efficacy. Thus, the absolute computational times are irrelevant and only a relative comparison between them will play a role. Our main focus will be the impact of the different dynamics on the objective function values. To this end, we used direct collocation on the same discretization grids. For both test problems, an explicit Euler scheme with equidistant step sizes was used. We tested different numbers of grid points n_t for the evaluation of differential states, with control values arbitrarily fixed to 1, compare Table 1. Then, we chose $n_t = 48000$ since the objective function values hardly change (less than 0.05%) with finer discretizations.

We also qualitatively compared the simulation results to those of an implicit Euler scheme and of a Gauß Radau collocation of order 3. These approaches are more efficient in the sense of computational speed, but we experienced also additional numerical issues such as convergence issues in IPOPT.

n_t	(Lotka)		(Calcium)	
	(STO-f)	(POC-f)	(STO-f)	(POC-f)
24 000	9.42072	9.41165	1992.72	1991.31
48 000	9.41165	9.40712	1991.31	1990.62
96 000	9.40712	9.40485	1990.62	1990.27
192 000	9.40485	9.40372	1990.27	1990.10

Table 1 Simulated objective function values for the two test problems and fixed controls $w_{i1}^{\text{STO}} = w_{i1}^{\text{POC}} = 1$ for all i and an equidistant explicit Euler scheme with n_t discretization points. As expected for $w \in \{0, 1\}$, one obtains the same results for $n_t^{\text{STO}} = 2n_t^{\text{POC}}$. In the following, we used $n_t^{\text{STO}} = n_t^{\text{POC}} = 48000$ for (Lotka) and (Calcium) as a good compromise between computational costs and acceptable relative numerical errors below 0.05%.

The results for the explicit Euler scheme are hence easier to compare and might always serve as an initialization of the more involved schemes.

Unfortunately, even for this simplistic numerical scheme, a fair comparison with the same number of integration intervals for STO and POC is not straightforward because the discretization grid depends on the values in w^{STO} . If $w_{ij}^{\text{STO}} \in \{0, 1\}$ for all i, j , then for the overall number of state discretization time points n_t^{STO} it holds that $n_t^{\text{STO}} = n_\sigma n_t^{\text{POC}}$ since $n_\sigma - 1$ sub-intervals on each discretization interval will not be used for integration in the STO approach. However, if $w_{ij}^{\text{STO}} \in (0, 1)$ for all i, j , then $n_t^{\text{STO}} = n_t^{\text{POC}}$. As these values w_{ij}^{STO} are going to change during the optimization, there is no perfect choice. In the interest of getting an idea about the numerical effort, we chose n_t identical for (STO-f) and (POC-f). Also in the interest of simplicity, we focused on $n_\sigma = 2$ and a switching order $\sigma(1) = 1, \sigma(2) = 0$ on each sub-interval. As we consider cases with only two possible modes $\omega(t) \in \{0, 1\}$, we can omit the second index, such that $w_i = w_{i1}$ and $1 - w_i = w_{i2}$.

6.2 Measuring Convexity

We would like to evaluate the constructed solutions by STO and POC in terms of (non-)convexity. One quantitative measure of nonconvexity is the *nonconvex ratio* [58], a measure of the nonconvexity of black-box functions. It can be easily applied to our `ampl_mintoc` implementation via outer simulation loops. The approach uses the midpoint convexity as a special case of convexity. We repeat the definitions here for convenience.

Definition 16 (Midpoint convexity) *The continuous function $f : \mathbb{R}^n \supset \mathbb{M} \rightarrow \mathbb{R}$ is called midpoint convex on the convex domain \mathbb{M} , if and only if*

$$f(x_1) + f(x_2) \geq 2f\left(\frac{x_1 + x_2}{2}\right) \quad \forall x_1, x_2 \in \mathbb{M}. \quad (9)$$

The following definition already takes the practical calculation of the nonconvex ratio into account. Details of the development of the method can be found in [58].

Definition 17 (Nonconvex ratio R^*) Let $\zeta : \mathbb{M}^2 \rightarrow \{0, 1\}$ denote a function with $\zeta(x_1, x_2) = 1$ if and only if x_1 and x_2 fulfill (9) and let S denote the number of samples. For uniformly chosen $x_{1i}, x_{2i}, i \in [S]$, the nonconvex ratio R^* can be computed as

$$R^* := 1 - \frac{1}{S} \sum_{i=1}^S \zeta(x_{1i}, x_{2i}) \quad (\{x_{1i}, x_{2i}\} \sim \mathcal{U}(\mathbb{M}^2)). \quad (10)$$

Note, that a value of $R^* = 1$ corresponds to a concave function. Please also note that R^* is only an approximation of the ratio of the number of points fulfilling the midpoint convexity property to the number of pairs of points in \mathbb{M} . Thus, the parameter S has to be chosen sufficiently large. Due to the fact that each function evaluation in our setting comes at the cost of solving a discretized ODE, we restricted the sample size S to 1000 in this study, still resulting in an overall computation time of approximately one week.

6.3 Simulation-based Analysis

We start with visualizations of the objective function landscapes for two-dimensional problems. For both problems (**Lotka**) and (**Calcium**) we performed simulations with $N = 2$ and varied the controls w_1, w_2 between 0 and 1 with an increment of 0.01. Figure 4 shows a comparison for (**Lotka**) and the case

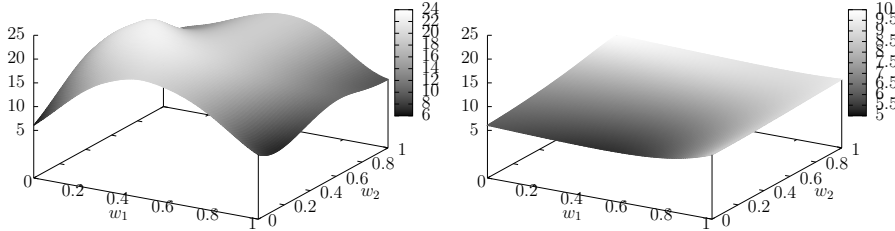


Fig. 4 Comparison of simulated objective function landscapes for (**Lotka**) example using two degrees of freedom w_1 and w_2 . Left: (**P-STO**), i.e., $\dot{x} = w_1(f_0 + f_1)$ for $t \in [0, 6w_1]$, $\dot{x} = (1 - w_1)f_0$ for $t \in [6w_1, 6]$ and $\dot{x} = w_2(f_0 + f_1)$ for $t \in [6, 6 + 6w_2]$, $\dot{x} = (1 - w_2)f_0$ for $t \in [6 + 6w_2, 12]$. Right: (**P-POC**), i.e., $\dot{x} = f_0 + w_i f_1$ for $t \in \mathcal{T}_i$ with $t_0 = 0, t_1 = 6, t_2 = 12$. The (**P-STO**) formulation has at least 3 local minima, see Table 2.

$N = 2$. The switching time optimization transformation on the left appears to be nonconvex, while the partial outer convexification transformation on the

right is much more convex. This is also reflected in the nonconvex ratios (see Definition 17). While $R_{STO}^* = 0.849$ is indicating a very nonconvex objective landscape, $R_{POC}^* = 0.188$. Note, that the nonconvexity of the results is mainly attributed to the very small number of controls ($N = 2$).

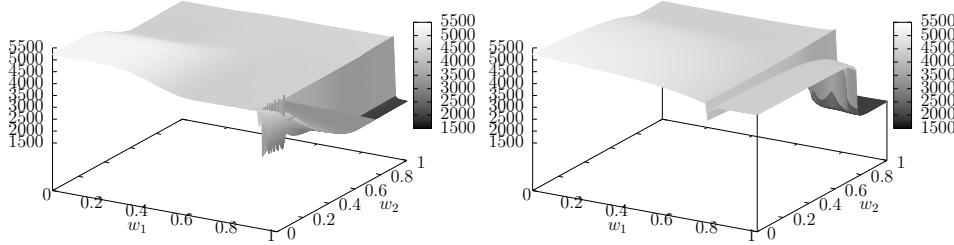


Fig. 5 Same setting as Figure 4, but for problem (Calcium). Both formulations are non-convex. The objective function landscapes are dominated by a bifurcation.

Figure 5 shows a comparison for (Calcium) and the case $N = 2$. Both transformations result in nonconvex objective function landscapes, which is also reflected in the high nonconvex ratios $R_{STO}^* = 0.472$ and $R_{POC}^* = 0.707$.

To get an impression of the nonconvexity in higher dimensions we performed another simulation. Here we used a one-dimensional parameterization

$$w_i = \beta, \quad i \in [N],$$

with β running from 0 to 1 in steps of 0.01. The resulting control for (P-POC) is simply a constant function with the value β and hence independent of N . However, in the case of (P-STO), each value of N results in different dynamics (cf. Figure 2). Figure 6 shows a comparison of the resulting one-dimensional slices through the N -dimensional objective function landscapes for (Lotka) and for (Calcium).

Comparing the resulting objective function values for (Lotka) on the left of Figure 6, one gets an idea of the nonconvexity of the different transformations of the optimization problem depending on N . For each N , the formulation (P-STO) is less convex than the (P-POC) formulation. For $N = 2$ one sees two of the local minima that we observed already in Figure 4 (left). When looking at the objective functions for increasing N , one sees a convergence of the (P-STO) objective function values to the (P-POC) objective function values.

A similar picture emerges from the right plot of Figure 6. Here, however, the objective function landscape is also visibly nonconvex for formulation (P-POC). Again, for small values of N more local minima can be seen in (P-STO), and for large N the objective functions look similar.

We considered the R^* values also in higher dimensions. Figure 7 illustrates the values of R^* as a function of N . For (Lotka), the (P-STO) formulation

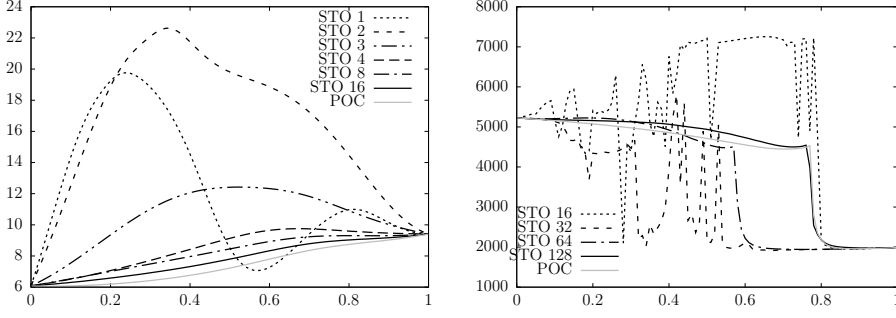


Fig. 6 Simulated objective function values for fixed $w_i = \beta$. The x -axis reflects the fixed values $w_i = \beta$. We show the values based on the transformed problem (P-STO) for (Lotka) and $N \in \{1, 2, 3, 4, 8, 16\}$ (left) and for (Calcium) and $N \in \{16, 32, 64, 128\}$ (right). Note that (P-POC) has identical results for all values of N due to the constant choice of w_i . STO- N converges to POC as N increases.

produces higher nonconvexities, which approaches that of (P-POC) for large N . For (Calcium), we have a similar result in the sense that (P-STO) appears to be more convex than (P-POC) with the exception of $N \in \{1, 2, 64\}$. Note, that because of our relatively small sample size, the results suffer from high uncertainty and need to be interpreted with care. Also, deviating from the rest of this section, we used $n_t = 51200$ in this case. Otherwise, the larger values of N would have resulted in a corresponding non-integer value for n_σ .

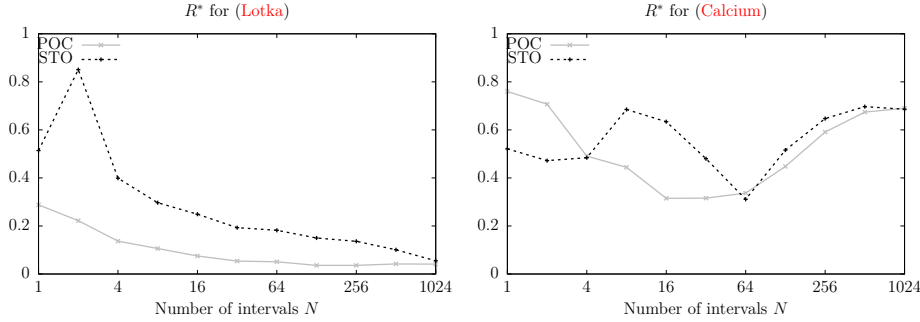


Fig. 7 Nonconvex ratios R^* for different discretizations N for (Lotka) on the left and (Calcium) on the right-hand side for sample size $S = 1000$. Here, only the x axes are logarithmic. One observes higher R^* values for (P-STO) compared to (P-POC) for small N and similar values for larger N .

6.4 Optimization-based Analysis

We performed an additional study to understand the convergence behavior for (P-STO) and (P-POC). We calculated local minima for $n_t = 48000$ and

different values of N using IPOPT. For both formulations, we used random initializations for the controls w by means of AMPL's `Uniform01` function. For (P-STO) and (P-POC) we used the same random seed via `option randseed`, such that the starting values $w_0^{\text{STO}} = w_0^{\text{POC}}$ were uniformly randomly distributed in $[0, 1]^N$ and identical. Also for the corresponding state variables $x(\cdot)$, initial values were provided. We used the initializations STOb and POOb where we initialized x discontinuously, but possibly close to an optimal solution via

$$x_{ki} = \begin{cases} \tilde{x}_k & \text{if } k < n_x \\ 0 & \text{if } k = n_x \end{cases}$$

for $i \in [n_t]$. Here, we used the special structure of the control problems (Lotka) and (Calcium) and their (steady-state \tilde{x} seeking) Mayer terms in x_3 and x_5 , respectively. As a second scenario, we considered initializations STOb and POOb, respectively, for values $x(\cdot)$ that were obtained from a (single shooting) forward solution based on the fixed initial values $x(0)$ and $w_0^{\text{STO}} = w_0^{\text{POC}}$.

To obtain an idea about the distribution of local minima for the two reformulations, we performed 100 optimization runs for each of the four transformations plus initializations STOb, STOb, POOb, and POOb, and for different values of N . An upper limit of 60 minutes CPU time was imposed via an AMPL and IPOPT option and additionally via a system `timeout` command. The latter was necessary because often IPOPT got stuck in the linear system solves. Hence, the category “no convergence” in the tables is composed of several different numerical phenomena, such as too large number of iterations or problems with dual variables in the range of 10^{12} in the linear system solves. As stated before, the implementations are by no means efficient and the CPU times shall only give a hint on how the different transformations relate to one another. Iteration counts and CPU times (and their standard deviations) are only calculated from the instances that converged within the time limit.

The numbers of times (out of 100 each) that IPOPT converged to a local minimum are shown in Tables 2 and 3. Stationary points are clustered with a tolerance of 10^{-3} for (Lotka) and 1 for (Calcium), comparing objective function values (not the values of w).

We observe that for small numbers of N the number of local minima are larger for (P-STO) than for (P-POC), e.g., 3 : 1 for (Lotka) and $N = 2$ (compare also Figure 4 left for this special case) or 11 : 6 for (Calcium) and $N = 3$. For $N = 100$ the behavior is almost identical for (P-STO) and (P-POC): converging to only one local minimum for (Lotka) and to a comparable amount of five to six local minima for (Calcium).

One can also see differences for state initialization a (i.e., setting $x^0 = \tilde{x}$) and b (i.e., calculating x_i^0 from x_0 and w^0). While STOb and POOb have a tendency to converge to better local minima, they also have a tendency to not converge at all.

Finally, to provide an idea of how local minima differ in the space of the corresponding controls and differential states, Figure 8 shows exemplary solutions for (Calcium). All four plots correspond to results calculated for $N = 100$

N	Objective function		Convergence to local minima			
	(P-STO)	(P-POC)	STOa	STOb	POCa	POCb
2	6.08124	5.40938	74×		100×	100×
	7.06487			87×		
	9.96625		2×	13×		
	No convergence		24×			
	Iterations		487 ± 586	32 ± 7	94 ± 115	22 ± 2
	CPU time [s]		453 ± 591	21 ± 5	72 ± 88	17 ± 2
3	4.07543	4.98114	76×	69×	99×	100×
	7.06418	7.44638	5×	31×	1×	
	No convergence		19×			
	Iterations		494 ± 693	40 ± 8	30 ± 9	26 ± 11
	CPU time [s]		368 ± 496	25 ± 5	26 ± 7	21 ± 9
4	1.72221	2.75505	81×	100×	100×	100×
	No convergence		19×			
	Iterations		681 ± 679	43 ± 9	51 ± 36	31 ± 2
	CPU time [s]		861 ± 881	32 ± 8	41 ± 29	24 ± 2
5	1.35203	1.38501	79×	100×	100×	100×
	No convergence		21×			
	Iterations		500 ± 581	41 ± 3	50 ± 6	35 ± 2
	CPU time [s]		795 ± 985	29 ± 3	40 ± 4	27 ± 1
8	1.67690	1.46873	77×	100×	100×	100×
	No convergence		23×			
	Iterations		265 ± 381	47 ± 6	61 ± 4	46 ± 2
	CPU time [s]		385 ± 508	33 ± 5	47 ± 3	33 ± 2
100	1.34568	1.34488	100×	100×	100×	100×
	Iterations		208 ± 26	98 ± 4	184 ± 10	133 ± 7
	CPU time [s]		171 ± 29	74 ± 3	139 ± 9	98 ± 6

Table 2 Convergence results for different numbers N of controls, the transformations and initializations STOa, STOb, POCa, and POCb introduced in Section 6.4, and for problem (Lotka). We show the number of times that IPOPT converged to a local minimum with the objective function value given in the left columns. A trend towards more local minima for transformation (P-STO) can be observed for small values of N , as well as a higher dependence on the initialization and in general more issues with convergence. The number of iterations and the CPU times are using the format *mean value* ± *standard deviation*.

as shown in Table 3. Comparing the first two and the last two rows, respec-

N	Objective function		Convergence to local minima			
	(P-STO)	(P-POC)	STOa	STOb	POCa	POCb
2	1958.14	1959.28		14×	40×	15×
	4412.69	4418.30		35×		84×
	No convergence		100×	51×	60×	1×
	Iterations		—	1063 ± 450	459 ± 392	246 ± 237
	CPU time [s]		—	2191 ± 921	781 ± 596	391 ± 356
3	1721.85	1894.62		9×		8×
	1838.70	1908.83		9×	19×	2×
	1846.60	2372.22		1×		5×
	4429.67	4322.82		16×		15×
	4480.46	4399.00		2×		15×
	4482.31	4402.87		8×		54×
	4514.67			10×		
	4528.46			6×		
	4530.22			2×		
	4608.23			2×		
	4755.96			1×		
	No convergence		100×	34×	81×	1×
	Iterations		—	851 ± 423	668 ± 560	284 ± 382
	CPU time [s]		—	1832 ± 936	1538 ± 1084	586 ± 665
4	1690.03		4×	5×		
	1790.12	1950.00		1×	16×	4×
	1794.49	1954.58		36×	3×	44×
	4337.71	4289.55		23×		14×
	4402.78	4291.79		1×		37×
	4458.08			2×		
	No convergence		96×	32×	81×	1×
	Iterations		280 ± 105	1046 ± 341	274 ± 152	626 ± 359
	CPU time [s]		885 ± 360	2312 ± 763	535 ± 268	1195 ± 631
100	1597.36		72×			
	1597.61	1596.66		10×	100×	13×
		4193.96				1×
	4211.54			1×		
	4222.59	4221.75		1×		1×
	4223.87	4222.82		9×		2×
	4225.18	4223.89		68×		76×
	No convergence		28×	11×		7×
	Iterations		696 ± 136	392 ± 156	227 ± 64	624 ± 209
	CPU time [s]		2110 ± 473	1555 ± 614	446 ± 128	2156 ± 723

Table 3 As Table 2, but for problem (Calcium).

tively, one sees how the activations of mode 1 in the (P-STO) and (P-POC) formulations are very similar in the state space. The corresponding trajectories are almost identical, which is also reflected in the objective function values (1597.61 versus 1596.66 and 4225.18 versus 4223.89). In control space, the solutions are only similar in a weak topology and look different to the eye. A comparison of rows 1 and 2 on the one hand, and rows 3 and 4 on the other hand, gives a comparison between two different local minima also in state space. One state remains in the steady state after time $t \approx 7$, while the other state continues to oscillate. A second control activation at $t \approx 15$ is necessary to bring the system into the unstable steady state.

6.5 Initialization of STO with solution of POC

As a remedy to avoid bad local minima and the also observed convergence issues, it seems a logical choice to use (P-POC) in a first phase to derive an initialization for (P-STO), as first suggested and implemented in [47].

We thus investigated the behavior of (P-STO), when being initialized with the optimal solution of (P-POC). For this purpose, we simply performed a "hot-start" of the STOb and STOb methods after running POCa and POCb, respectively. Other than that, the experimental setup remained unchanged. We also did some computations with an additional run of SUR for the solutions of POCa and POCb to use integer solution for the initialization. The results with respect to convergence properties and objective values were much worse than for the original initializations of STOb and STOb. We omit detailed results here and instead focus on the experiments without this additional algorithm.

Looking again at the results for (Lotka) in Table 2, in all cases but one (POCa and $N = 3$) the methods POCa and POCb converged to the same objective function value and correspondingly to the same optimal controls. Thus, it is sufficient to inspect one run of STOb and one run of STOb for each N for this problem. In Table 4 the objective function values and numbers of STO-iterations for the four combinations of initializations are presented. As in the case of STOb without POC-initialization, all instances could be solved in this scenario. Also, the number of iterations until convergence is close to the results of STOb. In the cases of $N = 2$ and $N = 3$, only the smaller objective function values were achieved. In comparison with STOb without POC-initialization, the hot-start gave an advantage in numbers of iterations and stability, but was not able to capture the best objective in the case of $N = 2$.

Very similar results can be observed in the calcium example in Figure 9. Here, we chose a graphical representation due to the much higher number of local minima. Again, the initialization of the method with the results of POCa and POCb can help by means of more successful runs for STOb. As a drawback, again, the best results for $N = 4$ could not be achieved. In comparison to STOb without POC-initialization, the number of solved instances is lower for each combination, but one.

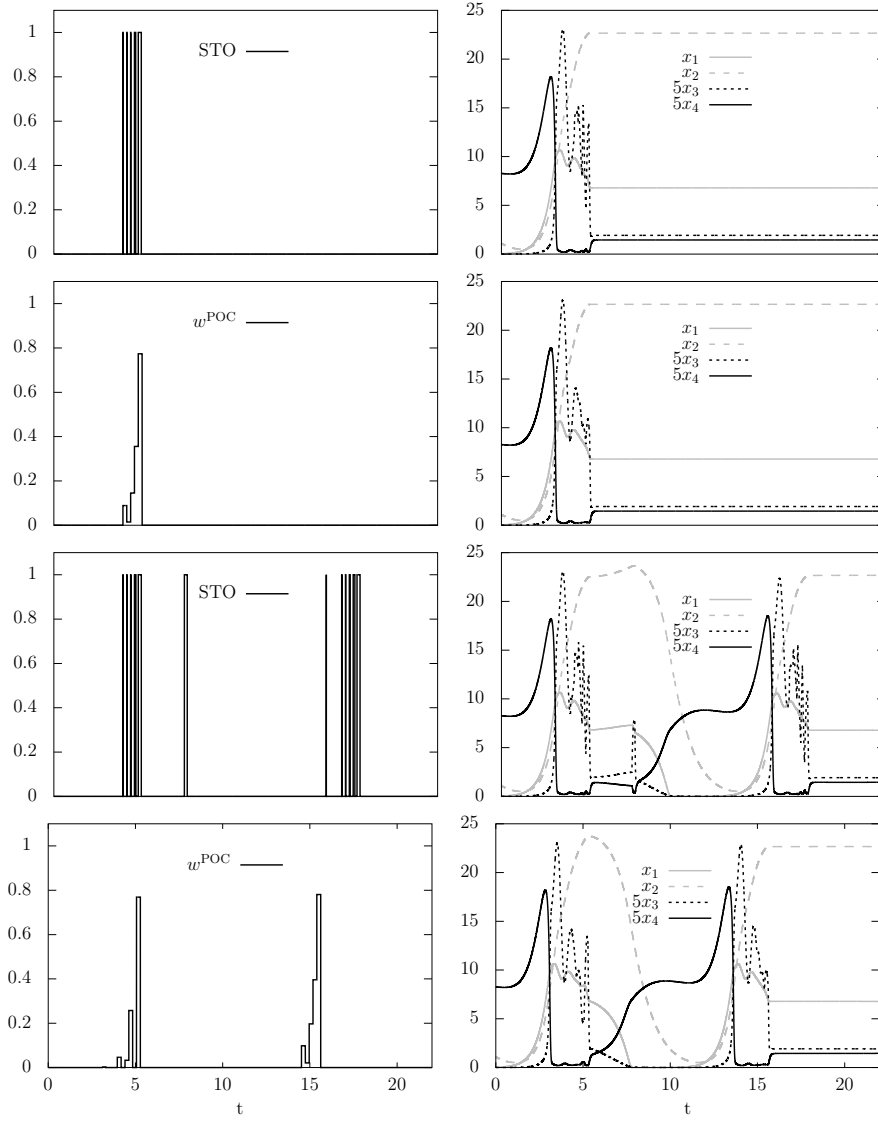


Fig. 8 Exemplary stationary points for (Calcium) and $N = 100$, corresponding to the objective function values 1597.61, 1596.66, 4225.18, and 4223.89 (top to bottom).

6.6 Comparison of Constructed (MIOCP) Solutions

To fairly compare the two approaches STO and POC, it is useful to also construct and analyze the resulting integer (MIOCP) solutions. For this, we used the Lotka-Volterra problem with $N = 100$ intervals and investigated both, the number of switches and the corresponding (MIOCP) objective function values

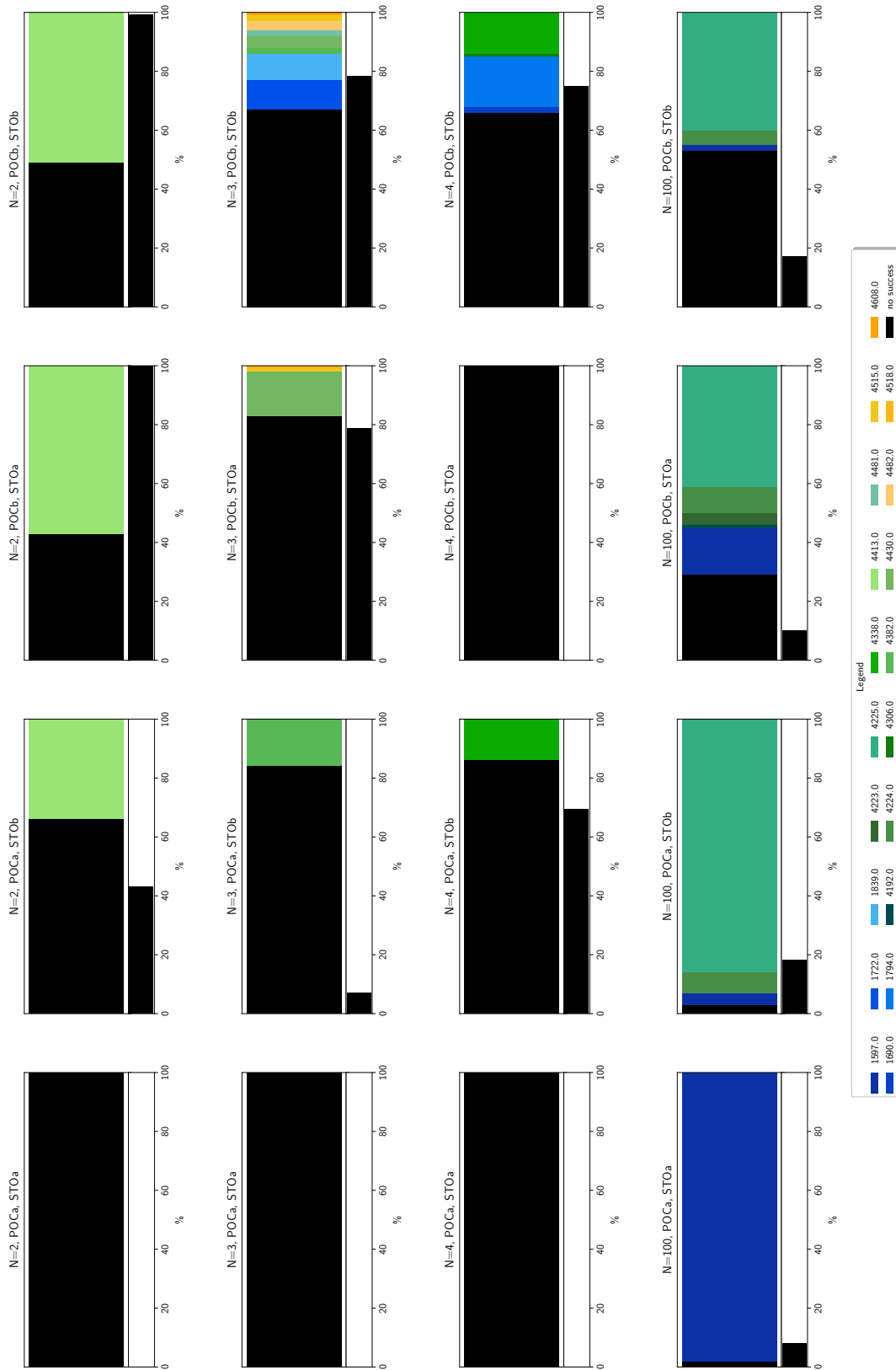


Fig. 9 Results of initialization of STOb and STOb with the optimal solutions from POCa and POCb as described in Section 6.4 for $N \in \{2, 3, 4, 100\}$. The upper bar shows the number of local minima, where the number of instances with no convergence is presented in black. The black lower bar represents the ratio of the median number of iterations for each category to the maximum median number of iterations over all categories. In the case of no success for all instances, the bars are empty.

n_t	Objective function	POCa		POCb	
		STOa	STOb	STOa	STOb
2	7.06487	36	30	36	30
3	4.07543	53	42	53	42
4	1.72221	47	39	48	39
5	1.35203	40	37	40	37
8	1.67690	56	63	56	63
100	1.34568	117	117	117	115

Table 4 Objective function values and numbers of iterations of STOa and STOb for (Lotka) after hot-starting the problems with the results of POCa and POCb, respectively.

of the constructed solutions. We note that, in contrast to the optimal solution of (P-POC), the (P-STO) solution is already feasible for (MIOCP). Based on the (P-POC) solution w^* , we used the CIA problem from Def. (9) to compute solutions with a limited number of switches, where we varied the bound on the number of switches n_s between 2 and 20. Focusing on the (MIOCP) objective function value, we also applied SUR on w^* . The approximation Theorem 11 implies that for a given $\epsilon > 0$, we can construct a binary control via SUR with a corresponding (MIOCP) objective value that is ϵ -close to the optimal solution by refining the discretization grid. For this purpose, we created refined (P-POC) solutions \tilde{w} with $2^k N$, $k = 0, 1, 2, 3$, discretization intervals, where the w^* value on the unrefined interval is used for definition of \tilde{w} values on the refined intervals:

$$\tilde{w}_{i,j} := w_{l,j}^*, \quad \text{for } i \in \{2^k(l-1) + 1, \dots, 2^k l\}.$$

Then, we applied SUR on these relaxed control solutions \tilde{w} with $2^k N$ discretization intervals.

We illustrate the numerical result in Figure 10, where the x -axis depicts the number of switches and the y -axis shows the deviation of the objective value of the constructed solution from the (P-POC) in percentage. The (P-STO) solution is in terms of the objective function value very close to the (P-POC) solution, at the expense of a large number of switches. Since the (P-POC) solution is fractional and therefore has no meaningful switching value, we represent it as a line in the plot.

If we permit only a small number of switches in the CIA problem, the corresponding (MIOCP) solution deviates more than 7% from the (P-POC) objective value. The more switches we allowed, the better the (MIOCP) objective function value became. This holds up to $n_s = 14$. For larger n_s values, the determined binary control still involves only 14 switches, since more switches result in no improvement in the objective (on the unrefined grid).

The determined solution from SUR on the unrefined grid, i.e. $k = 0$, has a small number of switches compared to the (P-STO) solution but yields a larger objective value. By refining the discretization grid, the constructed SUR solutions converged to the optimal objective value of (P-POC) or even reached

Numerical Result 1 (Number of Iterations) *There is an increased number of iterations and of convergence issues for (P-STO) compared to (P-POC).*

A closer look at (STO-f) and (POC-f) reveals one detail: in (STO-f) the drift term f_0 is multiplied with the optimization variables w . In (POC-f), this is not the case owing to the one-hot property of $w \in \mathcal{W}$. It is well known that higher nonlinearities, e.g., measured via Lipschitz constants, typically lead to increased regions of fast local convergence and to worse convergence rate constants, e.g., [5, 8, 56]. Thus we think that Numerical Result 1 is due to the augmented nonlinearity of the differential equations that appear in function and derivative evaluations in (P-STO).

One may possibly construct cases in which this multiplication does not increase nonlinearity, because $f_0(x) \approx 1/w$, or the effect might be negligible compared to the nonlinearities in the other f_j . However, for most practically relevant cases we can assume that the multiplication of f_0 with w increases nonlinearity of the optimization problem. There might be ways to reduce this nonlinearity again via a lifting procedure and usage of big M or McCormick-type constraints. Since this would involve case-specific upper bounds on the right-hand sides and a further level of approximation, we did not pursue this direction here, but worked with the formulation that is usually used in the STO literature.

A second observation in Tables 2 and 3 was

Numerical Result 2 (Computational Costs) *The computational costs per iteration are roughly similar for (P-STO) and (P-POC).*

For our numerical approach and test problems the computational costs per iteration were mainly governed by the evaluation of the differential equations and the sensitivity differential equations. Thus, the computational effort to evaluate the right-hand sides in (STO-f) and in (POC-f) for the whole time horizon \mathcal{T} are important, as they carry over in a more or less straightforward way to the effort of derivative calculation. For many practical optimal control problems the evaluation of function and derivatives dominates the computational costs. Evaluating (POC-f) for a fixed $t \in \mathcal{T}$ needs more function evaluations, as all $n_\omega + 1$ functions f_j need to be evaluated, whereas for (STO-f) only two evaluations are necessary. However, if we assume that the computational effort to evaluate (STO-f) on \mathcal{T}_{ij} is approximately constant (this certainly depends on the numerical integration scheme), then the evaluation on all n_σ sub intervals \mathcal{T}_{ij} increases the overall effort by a factor of n_σ . This would then give advantage to (POC-f), because the drift term f_0 needs to be evaluated only once for each $t \in \mathcal{T}$ and multiple times for STO (because it is evaluated at different times). Another interesting question is the one about situations in which $w_{ij}^{\text{STO}} = 0$. While it is clear that in this case no integration on \mathcal{T}_{ij} would be necessary, an implementation of this is rather tricky to realize, especially as also derivatives are needed to determine whether the search direction of the optimization algorithm is positive for $w_{ij}^{\text{STO}} = 0$ or not.

For (P-STO) a speedup by two orders of magnitude was achieved for test problems based on a linearization of the dynamics [57]. The approach benefits

from the usage of an exponential integrator and closed-form analytical derivatives. The main idea should also carry over to POC transformations, with the main difficulty of deriving analytical formulas for the derivatives of the matrix exponential for non-commutative matrices, though.

As a summary, we see reasons that in many cases the computational costs for (P-STO) might be higher than for (P-POC) because of more iterations of a generic iterative optimization algorithm, with similar costs per iteration.

In Table 4 and Figure 9, the experiments of initializing (P-STO) with the optimal solution of (P-POC) led to ambiguous results. For some problem sizes, the number of converging instances could be increased; however, for few others, the opposite was the case. The same holds true with respect to the constructed best objective values. This non-intuitive behavior can have several reasons. An interior-point solver like IPOPT is not well suited for restarts, and active-set based approaches could rather be used. Also, efficient schemes like the one proposed in [57] could be taken into account, possibly exploiting additional structures in a hot-start. The transfer of differential states from one time grid to another has to be carefully implemented unless one uses a single shooting initialization with the controls resulting from the first phase, as we did in this study. This initialization led overall to better convergence properties than using only the optimal control values of (P-POC).

7.2 Nonconvexity of the Objective Function

There were three main observations in the previous section for the two benchmark problems, which might carry over to other mixed-integer optimal control problems. We are somewhat sloppy with our formulations concerning “small” or “large” values of N , as this is certainly problem-specific. In Figure 7 we got indications that

Numerical Result 3 (Nonconvexity Evaluation) *Quantified measures of nonconvexity such as R^* are higher for (P-STO) compared to (P-POC) for small N , and identical for large N .*

Almost as a corollary, but also from Figures 4, 5, and 6 as well as from Tables 2 and 3 we obtained

Numerical Result 4 (Number of Local Minima) *For fixed N , there are typically at least as many local minima for (P-STO) as for (P-POC).*

The case $N = 100$ in Table 3 indicated that

Numerical Result 5 (Asymptotic Objective Function Values) *The number of ε -optimal local minima is asymptotically, i.e. for $N \rightarrow \infty$, identical for (P-STO) and (P-POC).*

The additional nonconvexity in (P-STO) is again due to the additional multiplication of the drift term f_0 by w , as discussed above. Therefore, it is not surprising that there are instances in which the number of local minima

of (P-POC) is dominated by the number corresponding to (P-STO), e.g., the case $N = 3$ in Table 3.

Numerical Result 5 is less intuitive. The asymptotic behavior of the number of local minima can be explained by the fact that the optimal objective function values of (P-STO) and (P-POC) are converging to the same value for increasing N as derived in Corollaries 12 and 13. Thus, the effect of the increased nonconvexity on the number of local minima can only be observed for coarse control discretizations (large Δt). We summarize that (P-STO) is likely to have more local minima for each discretization grid than (P-POC); however, the number of local minima is asymptotically identical.

7.3 Obtaining (MIOCP)-feasibility

While the STO approach constructs integer-feasible solutions for (MIOCP) and can be viewed as a primal heuristic, the main motivation of the POC approach is to provide a relaxed solution of (MIOCP). The necessary post-processing of the (P-POC) solution for determining an integer solution can be considered as a drawback, in particular since the rounded solution can result in relatively large objective values compared with the relaxed one. However, we showed in Section 6.6 that the rounded (P-POC) solution converges to the optimal solution of (MIOCP) by applying refinement of the discretization grid. Moreover, rounding methods such as SUR are computationally inexpensive. Regardless of whether we use STO or POC as solution approach for (MIOCP), a trade-off between objective quality and the number of used switches can be observed. The more switches we allow by means of control discretization or as part of CIA, the better the optimal objective value.

8 Conclusion

In this paper, we compared the (P-STO) and the (P-POC) approximations of Problem (MIOCP). We have demonstrated with numerical simulation and optimization results that (P-STO) tends to result in more local minima than (P-POC) for few degrees of freedom (small N), whereas both problem formulations show similar behavior for large N . We explained theoretically the underlying reasons. The additional nonconvexities and nonlinearities are due to a multiplication of the drift term $f_0(\cdot)$ with the control w in (P-STO). However, asymptotically in $N \rightarrow \infty$, i.e., for $\Delta t \rightarrow 0$, every solution of (P-STO) can be approximated arbitrarily close by a solution of (P-POC), and vice versa. The initialization of (P-STO) with optimal solutions of (P-POC) led to ambiguous results with respect to convergence properties but could be investigated further in the future. Another future research direction could be an investigation if singular, path-constrained, and bang-bang arcs in optimal solutions of (P-POC) have an impact in this context. Also, a broader numerical investigation with more control problems, more instances, various switching

orders σ , more efficient implementations, additional underlying constraints, and aspects of global and/or robust optimization could shed additional light on the question of how the two approximations can be combined to solve Problem (MIOCP).

Data availability The data that support the findings of this study are reproducible from the `ampl_mintoc` github library.

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 the Deutsche Forschungsgemeinschaft (DFG) – 314838170, GRK 2297 MathCoRe and SPPs 1962 and 2331.

References

1. Axelsson, H., Wardi, Y., Egerstedt, M., Verriest, E.: Gradient descent approach to optimal mode scheduling in hybrid dynamical systems. *Journal of Optimization Theory and Applications* **136**(2), 167–186 (2008)
2. Bengua, S., DeCarlo, R.: Optimal control of switching systems. *Automatica* **41**(1), 11–27 (2005). DOI 10.1016/j.automatica.2004.08.003
3. Bestehorn, F., Hansknecht, C., Kirches, C., Manns, P.: Mixed-integer optimal control problems with switching costs: A shortest path approach. submitted to Mathematical Programming B (2020). URL http://www.optimization-online.org/DB_HTML/2020/02/7630.html. Submitted
4. Bestehorn, F., Kirches, C.: Matching algorithms and complexity results for constrained mixed-integer optimal control with switching costs. *SIAM Journal on Optimization* (2020). URL http://www.optimization-online.org/DB_HTML/2020/10/8059.html. (submitted)
5. Bock, H.: Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen, *Bonner Mathematische Schriften*, vol. 183. Universität Bonn, Bonn (1987). URL <http://www.iwr.uni-heidelberg.de/groups/agbock/FILES/Bock1987.pdf>
6. Burger, M., Gerdts, M., Göttlich, S., Herty, M.: Dynamic programming approach for discrete-valued time discrete optimal control problems with dwell time constraints. In: IFIP Conference on System Modeling and Optimization, pp. 159–168. Springer (2015)
7. Caputi, W.J.: Stretch: A time-transformation technique. *IEEE Transactions on Aerospace and Electronic Systems* (2), 269–278 (1971)
8. Deuffhard, P., Heindl, G.: Affine invariant convergence theorems for newton's method and extensions to related methods. *SIAM Journal on Numerical Analysis* **16**(1), 1–10 (1979)
9. Dubovitskii, A., Milyutin, A.: Extremum problems in the presence of restrictions. *U.S.S.R. Comput. Math. Math. Phys.* **5**(3), 1–80 (1965). DOI 10.1016/0041-5553(65)90148-5
10. Dubovitskij, A., Milyutin, A.: Extremum problems with constraints. *Sov. Math., Dokl.* **4**, 452–455 (1963)
11. Egerstedt, M., Wardi, Y., Axelsson, H.: Transition-time optimization for switched-mode dynamical systems. *IEEE Transactions on Automatic Control* **51**, 110–115 (2006)
12. Flaßkamp, K.: On the optimal control of mechanical systems - hybrid control strategies and hybrid dynamics. Ph.D. thesis, University of Paderborn (2014). URL <https://nbn-resolving.de/urn:nbn:de:hbz:466:2-12756>
13. Fourer, R., Gay, D., Kernighan, B.: *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press (2002)
14. Fuller, A.: Study of an optimum nonlinear control system. *Journal of Electronics and Control* **15**, 63–71 (1963)

15. Gerdt, M.: 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)
16. Gerdt, M.: A variable time transformation method for mixed-integer optimal control problems. *Optimal Control Applications and Methods* **27**(3), 169–182 (2006)
17. Gerdt, M., Sager, S.: Mixed-integer dae optimal control problems: Necessary conditions and bounds. In: L. Biegler, S. Campbell, V. Mehrmann (eds.) *Control and Optimization with Differential-Algebraic Constraints*, pp. 189–212. SIAM (2012). URL <https://mathopt.de/PUBLICATIONS/Gerdt2012.pdf>
18. Girsanov, I.V.: *Lectures on Mathematical Theory of Extremum Problems*. Springer, Berlin-Heidelberg-New York (1972)
19. Göttlich, S., Potschka, A., Teuber, C.: A partial outer convexification approach to control transmission lines. *Computational Optimization and Applications* **72**(2), 431–456 (2019)
20. Göttlich, S., Potschka, A., Ziegler, U.: Partial outer convexification for traffic light optimization in road networks. *SIAM Journal on Scientific Computing* **39**(1), B53–B75 (2017)
21. Hante, F., Sager, S.: Relaxation methods for mixed-integer optimal control of partial differential equations. *Computational Optimization and Applications* **55**(1), 197–225 (2013)
22. Hante, F.M.: Relaxation methods for hyperbolic PDE mixed-integer optimal control problems. *Optimal Control Applications and Methods* **38**(6), 1103–1110 (2017). DOI 10.1002/oca.2315. Oca.2315
23. Hellström, E., Ivarsson, M., Aslund, J., Nielsen, L.: Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice* **17**, 245–254 (2009)
24. Ioffe, A.D., Tihomirov, V.M.: *Theory of extremal problems*. North-Holland Publishing Company, Amsterdam, New York, Oxford (1979)
25. Jung, M., Kirches, C., Sager, S.: On perspective functions and vanishing constraints in mixed-integer nonlinear optimal control. In: M. Jünger, G. Reinelt (eds.) *Facets of Combinatorial Optimization – Festschrift for Martin Grötschel*, pp. 387–417. Springer Berlin Heidelberg (2013). URL <https://mathopt.de/PUBLICATIONS/Jung2013.pdf>
26. Jung, M., Reinelt, G., Sager, S.: The lagrangian relaxation for the combinatorial integral approximation problem. *Optimization Methods and Software* **30**(1), 54–80 (2015)
27. Jung, M.N., Kirches, C., Sager, S., Sass, S.: Computational approaches for mixed integer optimal control problems with indicator constraints. *Vietnam Journal of Mathematics* **46**, 1023–1051 (2018). DOI <https://doi.org/10.1007/s10013-018-0313-z>
28. Kaya, C., Noakes, J.: Computations and time-optimal controls. *Optimal Control Applications and Methods* **17**, 171–185 (1996)
29. Kirches, C.: Fast numerical methods for mixed-integer nonlinear model-predictive control. Ph.D. thesis, Ruprecht-Karls-Universität Heidelberg (2010). URL <http://www.ub.uni-heidelberg.de/archiv/11636/>. Available at <http://www.ub.uni-heidelberg.de/archiv/11636/>
30. Kirches, C., Leyffer, S.: TACO – A toolkit for AMPL control optimization. *Mathematical Programming Computation* **5**(2), 227–265 (2013). DOI 10.1007/s12532-013-0054-7. URL <http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s12532-013-0054-7>
31. Kirches, C., Manns, P., Ulbrich, S.: Compactness and convergence rates in the combinatorial integral approximation decomposition. *Mathematical Programming* pp. 1–30 (2020)
32. Lebedz, D., Sager, S., Bock, H., Lebedz, P.: Annihilation of limit cycle oscillations by identification of critical phase resetting stimuli via mixed-integer optimal control methods. *Physical Review Letters* **95**, 108303 (2005)
33. Lee, H.W.J., Teo, K.L., Cai, X.Q.: An optimal control approach to nonlinear mixed integer programming problems. *Computers & Mathematics with Applications* **36**(3), 87–105 (1998)
34. Lee, H.W.J., Teo, K.L., Rehbock, V., Jennings, L.S.: Control parameterization enhancing technique for time optimal control problems. *Dynamic Systems and Applications* **6**(2), 243–262 (1997)

35. Lee, H.W.J., Teo, K.L., Rehbock, V., Jennings, L.S.: Control parametrization enhancing technique for optimal discrete-valued control problems. *Automatica* **35**(8), 1401–1407 (1999)
36. Leineweber, D., Bauer, I., Bock, H., Schlöder, J.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: Theoretical aspects. *Computers & Chemical Engineering* **27**, 157–166 (2003)
37. Leineweber, D., Bock, H., Schlöder, J.: Fast direct methods for real-time optimization of chemical processes. In: *Proc. 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics Berlin*. Wissenschaft- und Technik-Verlag, Berlin (1997)
38. Liu, C., Gong, Z., Lee, H.W.J., Teo, K.L.: Robust bi-objective optimal control of 1, 3-propanediol microbial batch production process. *Journal of Process Control* **78**, 170–182 (2019)
39. Liu, C., Gong, Z., Teo, K.L., Loxton, R., Feng, E.: Bi-objective dynamic optimization of a nonlinear time-delay system in microbial batch process. *Optimization Letters* **12**(6), 1249–1264 (2018)
40. Manns, P.: Relaxed multibang regularization for the combinatorial integral approximation. *arXiv preprint arXiv:2011.00205* (2020)
41. Manns, P., Kirches, C.: Improved regularity assumptions for partial outer convexification of mixed-integer pde-constrained optimization problems. *ESAIM: Control, Optimisation and Calculus of Variations* (2019). URL http://www.optimization-online.org/DB_HTML/2018/04/6585.html
42. Maurer, H., Büskens, C., Kim, J., Kaya, Y.: Optimization methods for the verification of second-order sufficient conditions for bang-bang controls. *Optimal Control Methods and Applications* **26**, 129–156 (2005). URL http://arachne.uni-muenster.de:8000/num/Arbeitsgruppen/ag_maurer/Publikationen/pubmaurer.html
43. Palagachev, K., Gerdt, M.: Mathematical programs with blocks of vanishing constraints arising in discretized mixed-integer optimal control problems. *Set-Valued and Variational Analysis* **23**(1), 149–167 (2015)
44. Picard, C.: Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives. *Journal de Mathématiques Pures et Appliquées* **6**, 145–210 (1890). URL <http://gallica.bnf.fr/ark:/12148/bpt6k107454k>
45. Ramanarayan Vasudevan Humberto Gonzalez, R.B., Sastry, S.S.: Consistent approximations for the optimal control of constrained switched systems—part 2: An implementable algorithm. *SIAM Journal on Control and Optimization* **51**, 4484–4503 (2013)
46. Ringkamp, M., Ober-Blöbaum, S., Leyendecker, S.: On the time transformation of mixed integer optimal control problems using a consistent fixed integer control function. *Mathematical Programming* **161**(1), 551–581 (2017). DOI 10.1007/s10107-016-1023-5. URL <http://dx.doi.org/10.1007/s10107-016-1023-5>
47. Sager, S.: Numerical methods for mixed-integer optimal control problems. *Der andere Verlag, Tönning, Lübeck, Marburg* (2005). URL <https://mathopt.de/PUBLICATIONS/Sager2005.pdf>. ISBN 3-89959-416-9
48. Sager, S.: On the Integration of Optimization Approaches for Mixed-Integer Nonlinear Optimal Control. University of Heidelberg (2011). URL <https://mathopt.de/PUBLICATIONS/Sager2011d.pdf>. Habilitation
49. Sager, S.: A benchmark library of mixed-integer optimal control problems. In: J. Lee, S. Leyffer (eds.) *Mixed Integer Nonlinear Programming*, pp. 631–670. Springer (2012). URL <https://mathopt.de/PUBLICATIONS/Sager2012b.pdf>
50. Sager, S., Bock, H., Diehl, M.: The integer approximation error in mixed-integer optimal control. *Mathematical Programming A* **133**(1–2), 1–23 (2012). URL <https://mathopt.de/PUBLICATIONS/Sager2012a.pdf>
51. Sager, S., Bock, H., Diehl, M., Reinelt, G., Schlöder, J.: Numerical methods for optimal control with binary control functions applied to a Lotka-Volterra type fishing problem. In: A. Seeger (ed.) *Recent Advances in Optimization, Lectures Notes in Economics and Mathematical Systems*, vol. 563, pp. 269–289. Springer, Heidelberg (2009). ISBN 978-3-5402-8257-0
52. Sager, S., Claeys, M., Messine, F.: Efficient upper and lower bounds for global mixed-integer optimal control. *Journal of Global Optimization* **61**(4), 721–743 (2015). DOI 10.1007/s10898-014-0156-4

53. Sager, S., Jung, M., Kirches, C.: Combinatorial integral approximation. *Mathematical Methods of Operations Research* **73**(3), 363–380 (2011). DOI [10.1007/s00186-011-0355-4](https://doi.org/10.1007/s00186-011-0355-4). URL <https://mathopt.de/PUBLICATIONS/Sager2011a.pdf>
54. Sager, S., Reinelt, G., Bock, H.: Direct methods with maximal lower bound for mixed-integer optimal control problems. *Mathematical Programming* **118**(1), 109–149 (2009). URL <https://mathopt.de/PUBLICATIONS/Sager2009.pdf>
55. Siburian, A.: Numerical Methods for Robust, Singular and Discrete Valued Optimal Control Problems. Ph.D. thesis, Curtin University of Technology, Perth, Australia (2004)
56. Smale, S.: Newton’s method estimates from data at one point. Springer-Verlag (1986)
57. Stellato, B., Ober-Blöbaum, S., Goulart, P.J.: Second-order switching time optimization for switched dynamical systems. *IEEE Transaction on Automatic Control* **62**(10), 5407–5414 (2017). DOI [10.1109/TAC.2017.2697681](https://doi.org/10.1109/TAC.2017.2697681). URL <http://doi.org/10.1109/TAC.2017.2697681>
58. Tamura, K., Gallagher, M.: Quantitative measure of nonconvexity for black-box continuous functions. *Information Sciences* **476**, 64 – 82 (2019). DOI <https://doi.org/10.1016/j.ins.2018.10.009>. URL <http://www.sciencedirect.com/science/article/pii/S0020025518308053>
59. Teo, K.L., Jennings, L.S., Lee H. W. J. Rehbock, V.: The control parameterization enhancing transform for constrained optimal control problems. *Journal of the Australian Mathematics Society* **40**(3), 314–335 (1999)
60. Uthaichana, K., Bengea, S., DeCarlo, R., Pekarek, S., Zefran, M.: Hybrid model predictive control tracking of a sawtooth driving profile for an HEV. In: *American Control Conference*, 2008, pp. 967–974 (2008)
61. Vasudevan, R., Gonzalez, H., Bajcsy, R., Sastry, S.S.: Consistent approximations for the optimal control of constrained switched systems—part 1: A conceptual algorithm. *SIAM Journal on Control and Optimization* **51**(6), 4463–4483 (2013)
62. Wang, L., Yuan, J., Wu, C., Wang, X.: Practical algorithm for stochastic optimal control problem about microbial fermentation in batch culture. *Optimization Letters* **13**(3), 527–541 (2019)
63. Wu, D., Bai, Y., Yu, C.: A new computational approach for optimal control problems with multiple time-delay. *Automatica* **101**, 388 – 395 (2019). DOI <https://doi.org/10.1016/j.automatica.2018.12.036>. URL <http://www.sciencedirect.com/science/article/pii/S0005109818306435>
64. Zeile, C., Robuschi, N., Sager, S.: Mixed-integer optimal control under minimum dwell time constraints. *Mathematical Programming* **188**, 653–694 (2021). DOI <https://doi.org/10.1007/s10107-020-01533-x>. URL <https://link.springer.com/article/10.1007/s10107-020-01533-x>
65. Zeile, C., Weber, T., Sager, S.: Combinatorial integral approximation decompositions for mixed-integer optimal control. submitted to *Optimization Methods and Software* (2018). URL http://www.optimization-online.org/DB_HTML/2018/02/6472.html. Submitted

A Appendix

A.1 How to set up and solve an MIOC with `ampl_mintoc`

Following AMPL standards, a problem-specification usually involves one data file, one model file and one run file. An optimal control problem `prob` can be formulated using `ampl_mintoc` via the problem-specific files `prob.dat`, `prob.mod`, and `prob.run`. First, problem dimensions (e.g. of the differential states and controls), parameters (e.g. the number of discretization intervals), and initial values are specified in `prob.dat`. In the `prob.mod` file, model-specific functions can be defined (in the discretized sense). This includes the model function f , mixed state-control constraints, path constraints c , and vanishing constraints. Listings 2 and 3 show the AMPL implementations for the ([Lotka](#)) and ([Calcium](#)) problems from Section 4 as examples of how to formulate f and, hence, the problems' dynamics.

Listing 2 Problem-dependent functions f_j as AMPL code for ([Lotka](#)). The index k denotes the entry of x_k , the index $o \in \{1, \dots, n_\omega\}$ the switching mode in f_o , and i is the time index.

```
var f {k in X,o in 0..no,i in I} = (
  if (k==1 && o==0) then x[1,i] - x[1,i]*x[2,i]
  else if (k==2 && o==0) then - x[2,i] + x[1,i]*x[2,i]
  else if (k==1 && o==1) then - x[1,i]*p[1]
  else if (k==2 && o==1) then - x[2,i]*p[2]
  else if (k==3 && o==0) then sum{l in 1..2} (x[1,i] - xtilde[l])^2
);
```

Listing 3 Problem-dependent functions f_j as AMPL code for ([Calcium](#)).

```
var f {k in X,o in 0..no,i in I} = (
  if (k==1 && o==0) then k1 + k2*x[1,i]
  - k3*x[1,i]*x[2,i]/(x[1,i]+K4) - k5*x[1,i]*x[3,i]/(x[1,i]+K6)
  else if (k==2 && o==0) then k7*x[1,i] - k8*x[2,i]/(x[2,i]+K9)
  else if (k==3 && o==0) then k10*x[2,i]*x[3,i]*x[4,i]/(x[4,i]+K11)
  + p18*k12*x[2,i] + k13*x[1,i] - k14*x[3,i]/(1.0*x[3,i]+K15)
  - k16*x[3,i]/(x[3,i]+K17) + x[4,i]/10
  else if (k==4 && o==0) then - k10*x[2,i]*x[3,i]*x[4,i]/(x[4,i]+K11)
  + k16*x[3,i]/(x[3,i]+K17) - x[4,i]/10
  else if (k==3 && o==1) then k14*x[3,i]/(1.0*x[3,i]+K15)
  - k14*x[3,i]/(1.3*x[3,i]+K15)
  else if (k==5 && o==0) then sum{k in 1..4} (x[k,i]-xtilde[k])^2
);
```

Note that we use the time index i to define the functions. While this might not be a very elegant way to formulate problems, this discretized definitions allows the user to easily specify piecewise functions on different sub-domains.

Finally, algorithmic choices can be provided in `prob.run`, such as the integration scheme, the problem reformulation, the simulation or optimization algorithm, relaxation of integer variables, or postprocessing of results.

Listing 4 Exemplary run file `lotka.run` as AMPL code.

```
model ../mintocPre.mod;
model lotka.mod;
model ../mintocPost.mod;
data lotka.dat;

let nlpsolver := "ipopt";
let integrator := "explicitEuler";
let mode := "SimulateST0";

include ../solve.run;
display objective;
```