Vietnam Journal of Mathematics
https://doi.org/10.1007/s10013-024-00704-z

REVIEW ARTICLE

Solving Mathematical Programs with Complementarity Constraints Arising in Nonsmooth Optimal Control

Armin Nurkanović[1] · Anton Pozharskiy[1] · Moritz Diehl[1,2]

Received: 3 November 2023 / Accepted: 30 April 2024

Abstract

This paper examines solution methods for mathematical programs with complementarity constraints (MPCC) obtained from the time-discretization of optimal control problems (OCPs) subject to nonsmooth dynamical systems. The MPCC theory and stationarity concepts are reviewed and summarized. The focus is on relaxation-based methods for MPCCs, which solve a (finite) sequence of more regular nonlinear programs (NLP), where a regularization/homotopy parameter is driven to zero. Such methods perform reasonably well on currently available benchmarks. However, these results do not always generalize to MPCCs obtained from nonsmooth OCPs. To provide a more complete picture, this paper introduces a novel benchmark collection of such problems, which we call NOSBENCH. The problem set includes 603 different MPCCs and we split it into a few representative subsets to accelerate the testing. We compare different relaxation-based methods, NLP solvers, homotopy parameter update and relaxation parameter steering strategies. Moreover, we check whether the obtained stationary points allow first-order descent directions, which may be the case for some of the weaker MPCC stationarity concepts. In the best case, the Scholtes' relaxation (SIAM J. Optim. **11**, 918–936, 2001) with IPOPT (Math. Program. **106**, 25–57, 2006) as NLP solver manages to solve 73.8% of the problems. This highlights the need for further improvements in algorithms and software for MPCCs.

Keywords MPECs · Nonlinear programming · Optimal control · Benchmark

Mathematics Subject Classification (2010) 65K05 · 90-02 · 90C30 · 90C33 · 90C90 · 49M37 · 90-04

This article is dedicated to the 70th birthday of Professor Tamás Terlaky, whose seminal contributions to interior point methods and computational optimization have inspired many developments in the field. Most of the developments reviewed in this article rely on efficient interior point methods for solving nonlinear subproblems.

✉ Armin Nurkanović
armin.nurkanovic@imtek.uni-freiburg.de

Anton Pozharskiy
anton.pozharskiy@imtek.uni-freiburg.de

Moritz Diehl
moritz.diehl@imtek.uni-freiburg.de

[1] Systems Control and Optimization Laboratory, Department of Microsystems Engineering (IMTEK), University of Freiburg, Freiburg im Breisgau, Germany

[2] Department of Mathematics, University of Freiburg, Freiburg im Breisgau, Germany

Published online: 27 August 2024

1 Introduction

This paper investigates numerical methods for solving Mathematical Programs with Complementarity Constraints (MPCCs) obtained from the time-discretization of Optimal Control Problems (OCPs) with nonsmooth dynamical systems. We consider different types of nonsmooth dynamical systems: (a) where the vector field is nonsmooth but continuous, or (b) nonsmooth and discontinuous (e.g., switched systems), and (c) systems with state jumps. In many cases, the nonsmoothness and combinatorial structure in such systems can be modeled by a coupling of differential algebraic equations with complementarity constraints. This gives rise to so-called Dynamic Complementarity Systems (DCSs) [10]. A complementarity constraint reads as $0 \leq x \perp y \geq 0$, which means that all entries of the two vectors $x, y \in \mathbb{R}^n$ must be nonnegative, i.e., $x_i \geq 0$, $y_i \geq 0$, and that at least one of the components in every pair is zero, i.e., $x_i y_i = 0$.

A continuous-time optimal control problem (OCP) subject to a DCS has the following form:

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot),y(\cdot)} \quad & \int_0^T L(x(t),u(t))\mathrm{d}t + E(x(T)) && \text{(1a)}\\
\text{s.t.} \quad & x(0) = \bar{x}_0, && \text{(1b)}\\
& \dot{x}(t) = f_{\mathrm{c}}(x(t),y(t),u(t)), && \text{(1c)}\\
& 0 = g_{\mathrm{c}}(x(t),y(t)), && \text{(1d)}\\
& 0 \leq G_{\mathrm{c}}(y(t)) \perp H_{\mathrm{c}}(y(t)) \geq 0, && \text{(1e)}\\
& 0 \leq h_{\mathrm{c}}(x(t),u(t)), \ \text{for almost all } t \in [0,T], && \text{(1f)}\\
& 0 \leq r(x(T)), && \text{(1g)}
\end{aligned}
$$

where $x \in \mathbb{R}^{n_x}$ are the differential states, $y \in \mathbb{R}^{n_y}$ the algebraic states and $u \in \mathbb{R}^{n_u}$ the control inputs. The function $L : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ models the stage cost and $E : \mathbb{R}^{n_x} \to \mathbb{R}$ is the terminal cost, $x_0 \in \mathbb{R}^{n_x}$ is a given parameter. The path and terminal constraints are grouped into the functions $h_{\mathrm{c}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_{h_{\mathrm{c}}}}$ and $r : \mathbb{R}^{n_x} \to \mathbb{R}^{n_r}$, respectively. The system (1c)–(1e) is a DCS, where the function $f_{\mathrm{c}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_x}$ models the right-hand side of the differential equation, the function $g_{\mathrm{c}} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_{g_{\mathrm{c}}}}$ defines the smooth algebraic equation in the DCS, and the functions $G_{\mathrm{c}} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_{\mathrm{cc}}}$, $H_{\mathrm{c}} : \mathbb{R}^{n_y} \to \mathbb{R}^{n_{\mathrm{cc}}}$ define the complementarity part of the DCS. It is assumed that all functions are at least twice continuously differentiable. Note that even if L and E are smooth and convex, f_{c}, g_{c} affine, and $-h_{\mathrm{c}}$ and $-r$ smooth convex functions, the complementarity constraints render the OCP (1) nonsmooth and nonconvex. In principle, it is also possible to have path and terminal constraints on the algebraic variables, which we have omitted to keep the notation light here and below in the discretization.

The DCS (1c)–(1e) abstraction allows one to model a variety of different nonsmooth systems. Examples are: Filippov differential inclusions reformulated into a DCS via Stewart's [83, 98] or the Heaviside step reformulation [79, 80], complementarity Lagrangian systems (modeling rigid bodies with friction and impacts) [10], relay systems [54], projected dynamical systems [46], Moreau's sweeping processes [72], and many more [10]. Moreover, with the use of time-freezing, several classes of systems with state jumps can be reformulated into Filippov systems [42, 76, 77, 81], which in turn lead to DCS. A detailed overview is given in [82].

Here, we consider a direct approach, where one first discretizes the continuous-time OCP (1) and obtains a finite-dimensional nonlinear program. For example, in a direct transcription method, one can use time-stepping methods (e.g., implicit Runge–Kutta (IRK) methods) to discretize the dynamics in time. In the case of smooth dynamical systems, direct methods are at a very mature stage [91]. However, in the case of nonsmooth systems such as DCS, this approach has some severe limitations. In particular, the time-stepping methods, with fixed integration step sizes, have at best first-order accuracy and the derivatives of the solutions/state transition maps with respect to parameters do not converge to the correct values [75, 100]. As a consequence of the wrong derivatives, an optimizer may converge to a spurious solution close to the initial guess [75, 100].

These limitations were recently overcome by the Finite Elements with Switch Detection (FESD) method [83, 84], which, inspired by [8], lets the integration step sizes to be degrees of freedom and introduces additional constraints that enable exact switch detection. This allows FESD to recover the higher-order accuracy properties of IRK methods and to compute correct sensitivities. This method is available in the open-source package `NOSNOC` [78]. In this work, we discretize the OCP (1) with the FESD method and obtain a discrete-time OCP. We introduce a uniform control discretization grid with N intervals and grid points $t_k = t_{k-1} + \frac{T}{N}$. Note that the control interval is not the same as the integration interval, and on every control interval $[t_k, t_{k+1}]$ one can apply a FESD method with multiple variable integration steps. The state approximations are denoted by $x_k \approx x(t_k)$, the control discretization is taken to be constant on the whole interval, i.e., $u(t) = u_k$, $t \in [t_k, t_{k+1}]$.

The vector z_k collects all internal variables of the FESD method, e.g., the Runge–Kutta stage variables of the algebraic and differential states, and the approximations of the algebraic variables $y_k \approx y(t_k)$. With a slight abuse of notation, they are summarized in the vectors $x = \left[x_0^\top, \dots, x_N^\top\right]^\top$, $z = \left[z_0^\top, \dots, z_{N-1}^\top\right]^\top$, and $u = \left[u_0^\top, \dots, u_{N-1}^\top\right]^\top$. A discretized version of the OCP (1) reads as [83]:

$$\min_{x,u,z} \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \tag{2a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_0, \tag{2b}$$

$$x_{k+1} = \phi_f(x_k, z_k, u_k), \qquad \text{for all } k \in \{0, \dots, N-1\}, \tag{2c}$$

$$0 = \phi_{\text{int}}(x_k, z_k, u_k), \qquad \text{for all } k \in \{0, \dots, N-1\}, \tag{2d}$$

$$0 \le \phi_G(z_k) \perp \phi_H(z_k) \ge 0, \qquad \text{for all } k \in \{0, \dots, N-1\}, \tag{2e}$$

$$0 \le h_{\text{c}}(x_k, u_k), \qquad \text{for all } k \in \{0, \dots, N-1\}, \tag{2f}$$

$$0 \le r(x_N). \tag{2g}$$

The functions ϕ_f, ϕ_{int}, ϕ_G and ϕ_H defining a discrete time DCS are obtained by applying the FESD method to the DCS (1c)–(1e), for a detailed definition of these functions, cf. [80, 83]. The terms $\ell(x_k, u_k)$ approximate the stage cost integral in (1a) over the intervals $[t_k, t_{k+1}]$, $k = 0, \dots, N-1$. The path constraints (1f) are for simplicity only evaluated at the points t_k, which yields (2f).

By setting $w = \left[x^\top, z^\top, u^\top\right]^\top \in \mathbb{R}^n$ and defining appropriate functions: $f : \mathbb{R}^n \to \mathbb{R}$ for the objective expression, $g : \mathbb{R}^n \to \mathbb{R}^{n_g}$ for collecting the equality constraints, $h : \mathbb{R}^n \to \mathbb{R}^{n_h}$ for the inequality constraints, and $G : \mathbb{R}^n \to \mathbb{R}^m$ and $H : \mathbb{R}^n \to \mathbb{R}^m$ for the complementarity functions, the discrete-time OCP (2) can be compactly written as a generic MPCC:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & f(w) \\ \text{s.t.} \quad & g(w) = 0, \end{aligned}$$

$$h(w) \geq 0,$$
$$0 \leq G(w) \perp H(w) \geq 0.$$

Mathematical Programs with Equilibrium Constraints (MPEC) are NLPs that have a parametric variational inequality or optimization problem as an constraint [71]. Such constraints can be under suitable conditions replaced by equivalent complementarity conditions. However, in the literature, because of the easier pronunciation, the acronym MPEC is frequently used for the problem above. There are few equivalent ways to state the complementarity constraints $0 \leq G(w) \perp H(w) \geq 0$ as formally smooth constraints:

1. $G(w) \geq 0$, $H(w) \geq 0$, $G_i(w)H_i(w) \leq 0$, *for all* $i \in \{1, \dots m\}$,
2. $G(w) \geq 0$, $H(w) \geq 0$, $G(w)^\top H(w) \leq 0$,
3. $G(w) \geq 0$, $H(w) \geq 0$, $G(w)^\top H(w) = 0$,
4. $G(w) \geq 0$, $H(w) \geq 0$, $G_i(w)H_i(w) = 0$, *for all* $i \in \{1, \dots m\}$,
5. $\Phi_{\mathrm{C}}(G(w), H(w)) = 0$.

In (4), Φ_{C} is a so-called C-function [22], which has the property $\Phi_{\mathrm{C}}(G(w), H(w)) = 0$ if and only if $0 \leq G(w) \perp H(w) \geq 0$. C-functions can be smooth or nonsmooth. To be consistent with most of the MPCC literature, we will work with complementarity constraints written via the inequality constraints in (1):

$$\min_{w \in \mathbb{R}^n} \quad f(w) \tag{3a}$$
$$\text{s.t.} \quad g(w) = 0, \tag{3b}$$
$$h(w) \geq 0, \tag{3c}$$
$$G_i(w) \geq 0,\ H_i(w) \geq 0,\ G_i(w)H_i(w) \leq 0, \qquad \textit{for all } i \in \{1, \dots m\}. \tag{3d}$$

There are no significant theoretical differences or computational advantages in using one of the other equivalent forms.

It is very common to introduce slack variables for the functions $G(w)$ and $H(w)$ to have only linear functions in the complementarity conditions, i.e., instead of (3d), one has:

$$s_G = G(w),\ \ s_H = H(w),\ \ s_G \geq 0,\ \ s_H \geq 0,\ \ s_{G,i}s_{H,i} \leq 0, \quad \text{for all } i \in \{1, \dots, m\}.$$

This is called the *vertical form* of the MPCC. This does not change any of the theoretical considerations and we stick to the notation of most of the MPCC literature where the slacks are not introduced. However, for the efficacy of numerical solvers, it is often beneficial to introduce the slacks [31], and we do so in the numerical experiments.

At this point, we mention also the class of Mathematical Programs with Vanishing Constraints (MPVC), where the complementarity constraints (3d) are replaced by $G_i(w)H_i(w) \geq 0$, $H_i(w) \geq 0$, for all $i = \{1, \dots, m\}$ [1]. They can be reformulated into equivalent MPCC, but it is often numerically more beneficial to treat them directly [1]. MPVCs arise often in the relaxation of OCPs with integer control and combinatorial constraints [55, 60].

The MPCC (3) is a nonlinear program (NLP) for which we need efficient and robust numerical solution methods. If a point w^* satisfies a Constraint Qualification (CQ), e.g., the Linear Independence Constraint Qualification (LICQ), then the Karush–Kuhn–Tucker (KKT) conditions are necessary for w^* to be local minimizer of (3) [73]. Standard nonlinear programming algorithms solve the KKT conditions to find a solution candidate. Unfortunately, due to the complementarity constraints (3d), standard CQs, such as LICQ and Mangasarian-Fromovitz Constraint Qualification (MFCQ) are violated at all feasible points [92, 104]. This implies both numerical and theoretical difficulties. On the one hand, the violation of MFCQ means that the set of Lagrange multipliers is necessarily unbounded, which leads to computational difficulties [53]. On the other hand, because of the absence of CQs, the KKT conditions may no longer be necessary for optimality anymore. This has led to the development of a tailored theory and solution methods for MPCCs, which we recall in detail in the following two sections.

A good way to assess the performance and robustness of a tailored MPCC method is to apply it to a benchmark set. Two widely used benchmarks for MPCCs are the MPECLib [18] and MacMPEC [64]. Baumrucker et al. [7] test several MPCC methods on the MPECLib test set, and examples from optimal process control. Hoheisel et al. [47] compare several relaxation-based methods on the MacMPEC problem set. All these experiments report success rates above 90%. In MPCCs arising from nonsmooth OCPs, we did not observe such robustness and high success rates, which motivated us to introduce a new collection of test problems to assess the performance of MPCC methods.

Contributions To learn more about the performance of MPCC methods in solving discrete-time nonsmooth OCPs, we introduce the NOSBENCH problem set. It contains 603 MPCCs generated via NOSNOC [78] from 33 continuous-time OCP and simulation problems. Together with a review of the theory and MPCC solution methods, the introduction of NOSBENCH is the main contribution of this paper. Furthermore, we compare nine different relaxation-based methods from the literature together with three NLP solvers: IPOPT [103], SNOPT [37] and WORHP [12]. We also compare homotopy parameter update and steering strategies. Some of the weaker multiplier-based MPCC stationarity concepts may allow first-order descent directions. We check is this the case for the solutions computed in our experiments. In our experiments, the Scholtes' relaxation [94] with IPOPT [103] as NLP subproblem solver is the most successful method-solver combination and solves 73.8% of the problems in the full NOSBENCH problem set. Furthermore, we validate the correctness of our implementations by running them on the MacMPEC test set, where we obtain results that aligns with those reported in literature [27, 51, 101].

Outline Section 2 reviews briefly the standard MPCC theory. In Section 3, we review easy-to-implement MPCC solution methods and comment on some other promising methods, which yet lack good implementations. Section 4 introduces the NOSBENCH problem test set and Section 5 discusses the results we obtain. We summarize our findings in Section 6.

2 Optimality Conditions for MPCCs

Due to the violation of the CQs, the standard NLP theory is often not applicable to MPCCs in the form of (3). This has several negative consequences: (a) the set of Lagrange multipliers is necessarily unbounded, (b) the gradients of the active constraints are linearly dependent at all feasible points, and (c) the linearization of (3) can be inconsistent arbitrarily close to a stationarity point [30].

We review first-order necessary optimality conditions and several stationarity concepts for MPCCs. Some optimality conditions are purely geometric, others rely on Lagrange multipliers. If there exists an i such that $G_i(w) = 0$, $H_i(w) = 0$, the multiplier-based stationary concepts may not be strong enough to characterize local minimizers. The MPCC-tailored theory presented here has its origins in [24, 31, 71, 92, 104].

The feasible set of the MPCC (3) is denoted by $\Omega_{\mathrm{MPCC}} = \{w \in \mathbb{R}^n \mid g(w) = 0, h(w) \geq 0, G(w) \geq 0, H(w) \geq 0, G_i(w)H_i(w) \leq 0, \text{ for all } i \in \{1, \dots, m\}$. We define the following index sets which depend on a feasible point $w \in \Omega_{\mathrm{MPCC}}$:

$$\begin{aligned}
\mathcal{I}_{+0}(w) &= \{i \in \{1, \dots, m\} \mid G_i(w) > 0,\, H_i(w) = 0\},\\
\mathcal{I}_{0+}(w) &= \{i \in \{1, \dots, m\} \mid G_i(w) = 0,\, H_i(w) > 0\},\\
\mathcal{I}_{00}(w) &= \{i \in \{1, \dots, m\} \mid G_i(w) = 0,\, H_i(w) = 0\}.
\end{aligned}$$

For ease of notation, if clear from the context, we omit the argument in the index sets. The set $\mathcal{I}_{00}$ is called the set of degenerate indices and is the source of most theoretical and numerical difficulties. If $\mathcal{I}_{00}$ is empty, we say that a solution w^* satisfies *strict complementarity*. This notion should not be confused with the notion of strict complementarity of an inequality constraint (3c) and the corresponding Lagrange multiplier in the NLP.

For a closed set Ω and a point $w \in \Omega$, a vector d is said to be tangent to Ω at w if there exists a sequence $w_k \in \Omega$ with $w_k \to w$, along with a sequence of positive scalar $t_k \to 0$ such that $d_k = \frac{w_k - w}{t_k} \to d$. The set of all vectors d that are tangent to Ω at a point $w \in \Omega$ is called the tangent cone to Ω at w and denoted by $\mathcal{T}_{\Omega}(w)$. Hence, we denote by $\mathcal{T}_{\Omega_{\mathrm{MPCC}}}(w)$ the tangent cone of a feasible point of (3). The active set of the inequality constraints $h(w) \geq 0$ is defined as the set $\mathcal{A}(w) = \{i \in \{1, \dots, n_h\} \mid h_i(w) = 0\}$. The linearized feasible cone $\mathcal{F}_{\Omega_{\mathrm{MPCC}}}(w)$ of the MPCC (3) at a feasible point $w \in \Omega_{\mathrm{MPCC}}$ is defined as the set:

$$\begin{aligned}
\mathcal{F}_{\Omega_{\mathrm{MPCC}}}(w) = \{d \in \mathbb{R}^n \mid\ & \nabla g(w)^\top d = 0,\\
& \nabla h_i(w)^\top d \geq 0, \text{ for all } i \in \mathcal{A}(w),\\
& \nabla G_i(w)^\top d = 0, \text{ for all } i \in \mathcal{I}_{0+}(w),\\
& \nabla H_i(w)^\top d = 0, \text{ for all } i \in \mathcal{I}_{+0}(w),\\
& \nabla G_i(w)^\top d \geq 0, \text{ for all } i \in \mathcal{I}_{00}(w),\\
& \nabla H_i(w)^\top d \geq 0, \text{ for all } i \in \mathcal{I}_{00}(w)\}.
\end{aligned}$$

This set is a polyhedral convex cone. On the other hand, if $\mathcal{I}_{00} \neq \emptyset$, then the tangent cone $\mathcal{T}_{\Omega_{\mathrm{MPCC}}}(w)$ is, due to the complementarity constraints, a union of polyhedral cones. Consequently, it is possibly a nonconvex cone. To see this, regard the set $\Omega = \{(w_1, w_2) \in \mathbb{R}^2 \mid w_1 \geq 0, w_2 \geq 0, w_1 w_2 \leq 0\}$. It can be verified that $\mathcal{T}_{\Omega}(0) = \Omega$, which is nonconvex, and $\mathcal{F}_{\Omega}(0) = \{(d_1, d_2) \in \mathbb{R}^2 \mid d_1 \geq 0, d_2 \geq 0\}$, which is convex. Therefore, the linearized feasible cone cannot locally capture the structural nonconvexity of the complementarity constraints. The KKT conditions require a CQ to hold in order to be necessary for optimality. We can see that even the rather nonrestrictive Abadie CQ (ACQ), which simply requires $\mathcal{F}_{\Omega_{\mathrm{MPCC}}}(w) = \mathcal{T}_{\Omega_{\mathrm{MPCC}}}(w)$, cannot be expected to hold for MPCCs [25]. Only the weaker Guignard CQ (GCQ) [39], which requires that the polar cones of $\mathcal{F}_{\Omega_{\mathrm{MPCC}}}(w)$ and $\mathcal{T}_{\Omega_{\mathrm{MPCC}}}(w)$ are equal, has a chance to hold [26]. However, it is difficult to verify this condition in practice.

To have a more powerful theoretical tool, the MPCC linearized feasible cone $\mathcal{F}^{\mathrm{MPCC}}_{\Omega_{\mathrm{MPCC}}}(w)$ can be used [24, 86, 92]. This cone is defined at a feasible point w as

$$\begin{aligned}
\mathcal{F}^{\mathrm{MPCC}}_{\Omega_{\mathrm{MPCC}}}(w) = \{d \in \mathbb{R}^n \mid\ & \nabla g(w)^\top d = 0,\\
& \nabla h_i(w)^\top d \geq 0, \text{ for all } i \in \mathcal{A}(w),\\
& \nabla G_i(w)^\top d = 0, \text{ for all } i \in \mathcal{I}_{0+}(w),\\
& \nabla H_i(w)^\top d = 0, \text{ for all } i \in \mathcal{I}_{+0}(w),\\
& 0 \leq \nabla G_i(w)^\top d \perp \nabla H_i(w)^\top d \geq 0, \text{ for all } i \in \mathcal{I}_{00}(w)\}.
\end{aligned}$$

The combinatorial structure is kept for the degenerate index set $\mathcal{I}_{00}$ and the cone $\mathcal{F}^{\text{MPCC}}_{\Omega_{\text{MPCC}}}(w)$ is nonconvex, if $\mathcal{I}_{00}$ is nonempty. In example from above, it holds that $\mathcal{F}^{\text{MPCC}}_{\Omega}(0) = \mathcal{T}_{\Omega}(0)$.

We proceed with stating optimality conditions and defining stationarity concepts for MPCCs. First-order necessary optimality conditions can be stated in terms of the tangent cone.

Theorem 1 *Let $w^* \in \Omega_{\text{MPCC}}$ be a local minimizer of (3), then it holds that*

$$\nabla f(w^*)^\top d \geq 0 \quad \textit{for all } d \in \mathcal{T}_{\Omega_{\text{MPCC}}}(w^*).$$

If a point w^* satisfies the condition above, in the MPCC literature it is said that geometric Bouligand stationarity (geometric B-stationarity) holds [24, 71]. For computational purposes, algebraic stationarity concepts are more useful. The algebraic Bouligand stationarity (or just B-stationarity) [71, 92] reads as follows.

Definition 1 (B-stationarity) A feasible point $w^* \in \Omega_{\text{MPCC}}$ of the MPCC (3) satisfies B-stationarity if it holds that

$$\nabla f(w^*)^\top d \geq 0 \quad \text{for all } d \in \mathcal{F}^{\text{MPCC}}_{\Omega_{\text{MPCC}}}(w^*).$$

Or equivalently [71, 92], a point $w^* \in \Omega_{\text{MPCC}}$ is called B-stationary if $d = 0$ is a local minimizer of the following linear program with complementarity constraints:

$$\min_{d \in \mathbb{R}^n} \nabla f(w^*)^\top d \quad \text{s.t.} \;\; d \in \mathcal{F}^{\text{MPCC}}_{\Omega_{\text{MPCC}}}(w^*). \tag{4}$$

It was shown in [24] that $\mathcal{T}_{\Omega_{\text{MPCC}}}(w) \subseteq \mathcal{F}^{\text{MPCC}}_{\Omega_{\text{MPCC}}}(w^*)$. This means that B-stationarity is less restrictive and it implies geometric B-stationarity. However, the converse is not always true, and we discuss below conditions when this is the case and when B-stationarity is necessary for optimality. B-stationarity is expensive to verify, as it requires the solution of a nonconvex optimization problem. In the worst case, this may require the solution of an exponential number of linear programs, unless some stronger regularity conditions hold.

As in the standard NLP theory, we may want to use Lagrange multiplier-based stationarity concepts, where we hopefully do not need to solve a combinatorial problem to find a solution candidate. As the standard CQs do not hold, we cannot use the KKT conditions of the MPCC (3), but we apply them to some auxiliary Nonlinear Programs (NLPs). The stationarity conditions for MPCCs are derived from more regular NLPs (defined next) which are locally associated with the initial MPCC (3) [92].

Definition 2 (Auxiliary NLP) Let $w^* \in \Omega_{\text{MPCC}}$. We define the following auxiliary NLPs:

- The Relaxed NLP (RNLP) for $w^* \in \Omega_{\text{MPCC}}$ is defined as

$$\begin{aligned}
\min_{w \in \mathbb{R}^n} \quad & f(w) && \text{(5a)}\\
\text{s.t.} \quad & g(w) = 0, && \text{(5b)}\\
& h(w) \geq 0, && \text{(5c)}\\
& G_i(w) = 0,\; H_i(w) \geq 0, \quad i \in \mathcal{I}_{0+}(w^*), && \text{(5d)}\\
& G_i(w) \geq 0,\; H_i(w) = 0, \quad i \in \mathcal{I}_{+0}(w^*), && \text{(5e)}\\
& G_i(w) \geq 0,\; H_i(w) \geq 0, \quad i \in \mathcal{I}_{00}(w^*). && \text{(5f)}
\end{aligned}$$

- The Tight NLP (TNLP) for $w^* \in \Omega_{\mathrm{MPCC}}$ is defined as the RNLP (5), except that the constraints (5f) are replaced by:

$$G_i(w) = 0, \quad H_i(w) = 0, \qquad i \in \mathcal{I}_{00}(w^*).$$

- Let $(\mathcal{I}_1, \mathcal{I}_2)$ be a partition of $\mathcal{I}_{00}$ such that $\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{I}_{00}$ and $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$. The Branch NLP ($\mathrm{BNLP}_{(\mathcal{I}_1,\mathcal{I}_2)}$) for $w^* \in \Omega_{\mathrm{MPCC}}$ is defined as the RNLP (5), except that the constraints (5d)–(5f) are now replaced by:

$$\begin{aligned} G_i(w) \geq 0, \quad H_i(w) = 0, \qquad i \in \mathcal{I}_{+0}(w^*) \cup \mathcal{I}_1(w^*), \\ G_i(w) = 0, \quad H_i(w) \geq 0, \qquad i \in \mathcal{I}_{0+}(w^*) \cup \mathcal{I}_2(w^*). \end{aligned}$$

Figure 1 illustrates the feasible sets of the auxiliary NLPs for $i \in \mathcal{I}_{00}$. We denote the feasible sets of the RNLP, TNLP, and $\mathrm{BNLP}_{(\mathcal{I}_1,\mathcal{I}_2)}$ by Ω_{RNLP}, Ω_{TNLP} and $\Omega_{\mathrm{BNLP}_{(\mathcal{I}_1,\mathcal{I}_2)}}$, respectively.

The usual NLP concepts such as first-order optimality conditions, stationary points, second-order conditions, and constraint qualification for MPCC are defined in terms of these auxiliary NLP. To see that this approach makes sense we look at how these problems and their solutions are related. It is not difficult to see that the following holds for $w^* \in \Omega_{\mathrm{MPCC}}$ [92]:

$$\Omega_{\mathrm{TNLP}} = \bigcap_{(\mathcal{I}_1,\mathcal{I}_2)} \Omega_{\mathrm{BNLP}_{(\mathcal{I}_1,\mathcal{I}_2)}} \subset \Omega_{\mathrm{MPCC}} = \bigcup_{(\mathcal{I}_1,\mathcal{I}_2)} \Omega_{\mathrm{BNLP}_{(\mathcal{I}_1,\mathcal{I}_2)}} \subseteq \Omega_{\mathrm{RNLP}}. \tag{6}$$

The same relations hold for the corresponding tangent cones at w^* as well. Furthermore, for a feasible point of the MPCC (3) $w^* \in \Omega_{\mathrm{MPCC}}$ the following can be said [92]. If w^* is a local minimizer of the RNLP, then it is a local minimizer of the MPCC. The converse is not true. If w^* is a local minimizer of the MPCC then it is a local minimizer of the TNLP. The point w^* is a local minimizer of the MPCC if and only if it is a local minimizer of every $\mathrm{BNLP}_{(\mathcal{I}_1,\mathcal{I}_2)}$. The last assertion once again highlights the combinatorial nature of MPCCs, since $2^{|\mathcal{I}_{00}|}$ branch NLPs must be checked to make conclusions about optimality. Fortunately, as we will see below, under reasonable assumptions we do not have to check every branch NLP but only the RNLP or TNLP to characterize a stationary point of the MPCC.

All these difficulties arise because of the degenerate indices $i \in \mathcal{I}_{00}$. If this set is empty, all auxiliary NLPs collapse to the same problem, and there is no combinatorial structure due to the BNLP anymore. It can be seen that the tangent cone of Ω_{MPCC} will be convex since there will be no rays that start from the degenerate point. Assuming that other constraints in the MPCC do not cause the violation of the ACQ, then we have that the standard ACQ holds for the MPCC and thus we can apply the KKT-conditions to verify the stationarity of $w^* \in \Omega_{\mathrm{MPCC}}$ in this fortunate case. In other words, $w^* \in \Omega_{\mathrm{MPCC}}$ is a local minimizer of

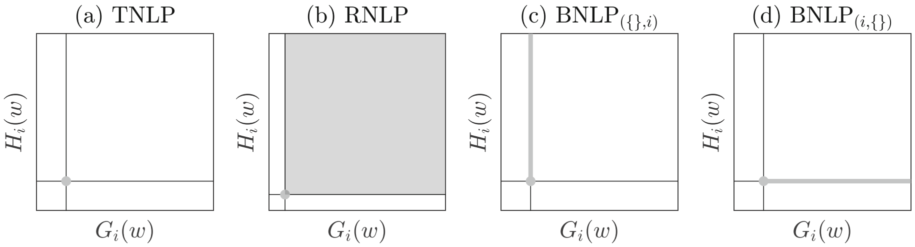

(a) TNLP
(b) RNLP
(c) $\mathrm{BNLP}_{(\{\},i)}$
(d) $\mathrm{BNLP}_{(i,\{\})}$
$H_i(w)$
$G_i(w)$

Fig. 1 Feasible sets of the auxiliary NLPs as defined in Definition 2

the MPCC if and only if it is a local minimizer of the RNLP/TNLP, which are equal in this case [92].

Next, we define the MPCC-specific Lagrangian, CQs, and stationarity concepts. The MPCC Lagrangian is the *standard* Lagrangian for the RNLP/TNLP, and reads as:

$$\mathcal{L}^{\mathrm{MPCC}}(w, \lambda, \mu, \nu, \xi) = f(w) - \lambda^\top g(w) - \mu^\top h(w) - \nu^\top G(w) - \xi^\top H(w),$$

with the MPCC Lagrange multipliers $\lambda \in \mathbb{R}^{n_g}$, $\mu \in \mathbb{R}^{n_h}$, $\nu \in \mathbb{R}^m$ and $\xi \in \mathbb{R}^m$. It differs from the standard Lagrangian for the MPCC (3) in omitting the bilinear terms $G_i(w)H_i(w) \leq 0$ and their multipliers.

Next we define some tailored MPCC CQs.

Definition 3 The MPCC (3) is said to satisfy the MPCC-LICQ (MPCC-MFCQ) at a feasible point w^* if the corresponding TNLP for w^* satisfies the LICQ (MFCQ) at the same point w^*.

The linearized feasible cone of a TNLP is always convex, and as seen in the discussion at the beginning of this section we can expect the standard ACQ to be violated if $\mathcal{I}_{00} \neq \emptyset$. This motivated the definition of the MPCC-ACQ and MPCC-Guignard CQ (MPCC-GCQ) in terms of the nonconvex cones [24, 25]. First, recall that given a cone $\mathcal{K} \subseteq \mathbb{R}^n$, its polar cone is defines as $\mathcal{K}^\circ = \{d \in \mathbb{R}^n \mid d^\top v \leq 0, \text{ for all } v \in \mathcal{K}\}$.

Definition 4 The MPCC-ACQ (MPCC-GCQ) holds at $w^* \in \Omega_{\mathrm{MPCC}}$ if and only if $\mathcal{F}^{\mathrm{MPCC}}_{\Omega_{\mathrm{MPCC}}}(w^*) = \mathcal{T}_{\Omega_{\mathrm{MPCC}}}(w^*)$ $(\mathcal{F}^{\mathrm{MPCC}}_{\Omega_{\mathrm{MPCC}}}(w^*)^\circ = \mathcal{T}_{\Omega_{\mathrm{MPCC}}}(w^*)^\circ)$.

Similar as for the standard CQ, for the MPCC CQs the following implications hold [95, 104]: MPCC-LICQ $\implies$ MPCC-MFCQ $\implies$ MPCC-ACQ $\implies$ MPCC-GCQ.

Inspired by the KKT conditions for standard non-degenerate NLPs, several stationarity concepts that rely on the auxiliary NLPs and their Lagrange multipliers can be defined [92]. If an appropriate MPCC-CQ holds, they are necessary for optimality, as we will discuss below.

Definition 5 (Stationarity concepts for MPCCs) Let w^* be feasible for the MPCC (3).

- Weak Stationarity (W-stationarity) [92]: A point $w^* \in \Omega_{\mathrm{MPCC}}$ is called W-stationary if the corresponding TNLP admits the satisfaction of the KKT conditions, i.e., there exist Lagrange multipliers λ^*, μ^*, ν^* and ξ^* such that:

$$\begin{aligned}
&\nabla_w \mathcal{L}^{\mathrm{MPCC}}(w^*, \lambda^*, \mu^*, \nu^*, \xi^*) = 0,\\
&g(w^*) = 0,\\
&0 \leq \mu^* \perp h(w^*) \geq 0,\\
&G_i(w^*) \geq 0,\ \nu_i^* = 0, \quad \text{for all } i \in \mathcal{I}_{+0}(w^*),\\
&H_i(w^*) \geq 0,\ \xi_i^* = 0, \quad \text{for all } i \in \mathcal{I}_{0+}(w^*),\\
&G_i(w^*) = 0,\ \nu_i^* \in \mathbb{R}, \quad \text{for all } i \in \mathcal{I}_{0+}(w^*) \cup \mathcal{I}_{00}(w^*),\\
&H_i(w^*) = 0,\ \xi_i^* \in \mathbb{R}, \quad \text{for all } i \in \mathcal{I}_{+0}(w^*) \cup \mathcal{I}_{00}(w^*).
\end{aligned}$$

- Strong Stationarity (S-stationarity) [92]: A point $w^* \in \Omega_{\mathrm{MPCC}}$ is called S-stationary if it is weakly stationary and $\nu_i^* \geq 0, \xi_i^* \geq 0$ for all $i \in \mathcal{I}_{00}(w^*)$. In other words, it is a KKT point of the corresponding RNLP.
- Clarke Stationarity (C-stationarity) [92]: A point $w^* \in \Omega_{\mathrm{MPCC}}$ is called C-stationary if it is weakly stationary and $\nu_i^* \xi_i^* \geq 0$ for all $i \in \mathcal{I}_{00}(w^*)$.

- Mordukhovich Stationarity (M-stationarity) [92]: A point $w^* \in \Omega_{\mathrm{MPCC}}$ is called M-stationary if it is weakly stationary and if either $\nu_i^* > 0$ and $\xi_i^* > 0$ or $\nu_i^* \xi_i^* = 0$ for all $i \in \mathcal{I}_{00}(w^*)$.
- Abadie Stationarity (A-stationarity) [24]: A point $w^* \in \Omega_{\mathrm{MPCC}}$ is called A-stationary if it is weakly stationary and $\nu_i^* \geq 0$ or $\xi_i^* \geq 0$ for all $i \in \mathcal{I}_{00}(w^*)$.

The feasible sets for the MPCC multipliers ν^* and ξ^* are depicted in Fig. 2. Observe that if $\mathcal{I}_{00} = \emptyset$, then all multiplier-based stationarity concepts collapse to the same.

The many different stationarity concepts might be confusing and one might be wondering if some of them are needed at all. However, these stationarity concepts are crucial for studying numerical methods for MPCC. As we will see in the next section, MPCCs are usually solved by solving a (finite) sequence of related and more regular NLPs. Depending on the underlying assumptions, the accumulation points of these methods are some of the stationary points defined above. Therefore, it is important to understand under which conditions these stationarity concepts are indeed necessary for optimality. It turns out that all of them can be necessary for local optimality if some additional specialized CQs hold [31, 92]. The results from [31, 71, 85, 92, 104] are summarized in the diagram in Fig. 3. For the missing CQ definitions see [95, 104].

We make a few comments on the relations in Fig. 3. As already discussed, if the ACQ does not hold, the KKT conditions are not applicable for characterizing a geometric B-stationary point. B-stationarity always implies geometric B-stationarity, the converse requires additionally the MPCC-GCQ to hold [24]. Note that by Definition 1, B-stationarity does not allow a first-order descent direction. Weaker concepts, mainly the multiplier-based ones, may allow first-order descent directions, even if the MPCC-LICQ holds. We illustrate this in the next example from [92].

Example 1 (Descent directions for C-stationarity) Regard the MPCC:

$$\min_{w_1, w_2 \in \mathbb{R}} (w_1 - 1)^2 + (w_2 - 1)^2 \quad \text{s.t.} \;\; w_1 \geq 0, \;\; w_2 \geq 0, \;\; w_1 w_2 \leq 0.$$

This example satisfies the MPCC-LICQ at all feasible points. The point $w^* = (0, 0)$ is C-stationary with the multipliers $\xi = -2$, $\nu = -2$. However, it has two descent directions $d = (0, 1)$ and $d = (1, 0)$, and is thus not B-stationary. The points $w = (1, 0)$ and $w = (0, 1)$ are local minimizers and S-stationary points.

Arguably, stationarity conditions that allow first-order descent directions might be considered as too weak. S-stationarity is the only multiplier-based and non-combinatorial condition that has a chance to be equivalent to B-stationarity. S-stationarity corresponds to the KKT conditions of the RNLP and it implies B-stationarity because of the inclusion relation in (6). Conversely, if the MPCC-LICQ holds, then B-stationarity implies also S-stationarity [92].

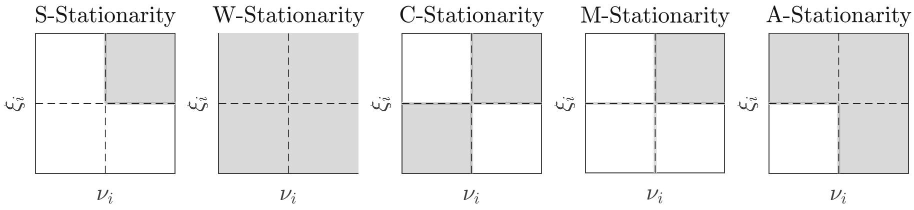


Fig. 2 Sign restrictions for MPCC multiplier in different stationarity concepts

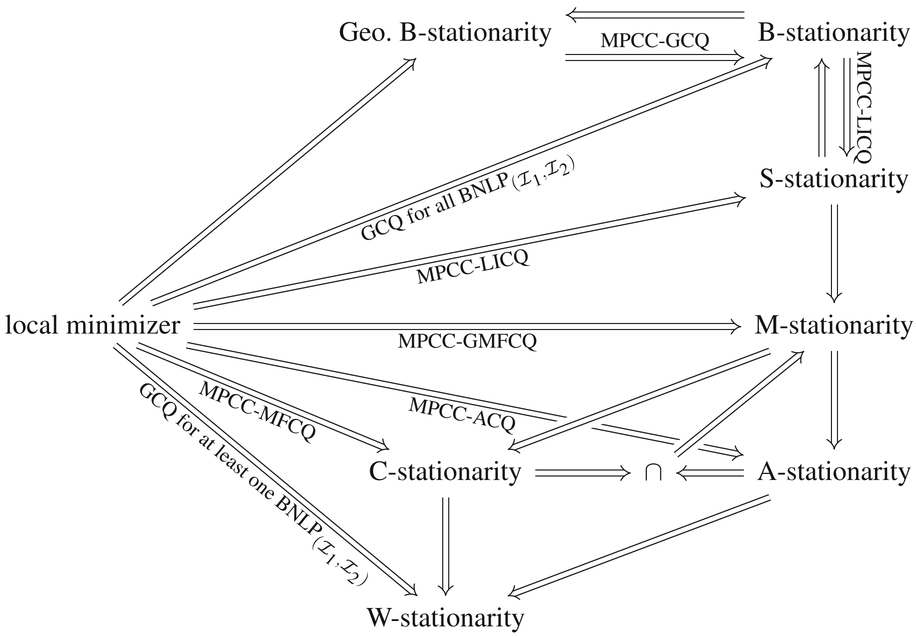


Fig. 3 Diagram summarizing MPCC-CQs and necessity of different stationarity concepts for optimality. This diagram was inspired by [63, Theorem 5.13] and [59, Fig. 2]

Unfortunately, weaker conditions, e.g. the MPCC-MFCQ are already not sufficient for S-stationarity, see [92, Example 3] for a counterexample. In particular, M-stationarity is the strongest necessary condition under MPCC-MFCQ [26, 92]. This means that B-stationary points that are not S-stationary can not be identified via any stationarity concept based on the auxiliary NLPs [102]. Instead, an exponential number of linear programs (4) must be solved for verification.

To summarize, under suitable CQs, local optimality is sufficient for any of the stationarity concepts defined here. However, everything weaker than S-stationarity is often considered to be too weak, since such points may allow first-order descent directions. The problem in practice is that iterative MPCC methods are often attracted by M- or C-stationary points, even though the MPCC may have B- or S-stationary points [67]. There exist also second-order optimality conditions tailored to MPCCs. They are defined in terms of S-stationary points and the corresponding MPCC multipliers. We omit their statement for brevity and refer the reader to [90, 92] for more details.

3 MPCC Solution Methods

In the last three decades, many tailored MPCC solution methods have been proposed. A recent survey of MPCC methods is given in [59]. The references [34, 47, 58] provide comparisons of several solution strategies. We classify the MPCC solution methods into two distinct classes:

(a) active-set-based/combinatorial methods,
(b) relaxation and smoothing-based methods.

We review both classes, but give more details for the second class, because we will later benchmark them on the OCP-based problem set.

3.1 Active-set Based MPCC Methods

These methods explicitly treat the combinatorial nature of the complementarity constraints (3d). They have the strongest convergence properties as they are usually guaranteed to converge to B-stationary points. They can be subdivided into branching methods [6, 59], and active-set methods [36, 40, 59, 61, 67]. These methods rely on guessing the correct active set by solving a linear program with complementarity constraints (LPCC), such as (4) [40, 61, 67]. If $d = 0$ solves the LPCC subproblem, a B-stationary point is found. To promote faster convergence rates, after fixing the active set, an equality-constrained QP can be solved [61, 67]. As the solution of an LPCC can be computationally expensive, Kirches et al. [61] regard LPCC only with complementarity and bound constraints. They suggest treating the remaining equality and inequality constraints in an augmented Lagrangian fashion. This is later done by Guo and Deng [40], where convergence to M-stationary points is proven. The main practical drawback of this method class is the lack of robust open-source implementations. In contrast to the next group we treat, they are not easily implementable by using an available NLP code. For more references we refer the reader to [63, Section 5.6.2] and [59, Section 3.2].

3.2 Relaxation and Smoothing-based Methods

The main idea behind relaxation-based methods is to replace (3d) with more regular constraints:

$$G_i(w) \geq 0, \quad H_i(w) \geq 0, \qquad \text{for all } i \in \{1, \ldots, m\}, \tag{7a}$$

$$\Phi(G_i(w), H_i(w), \sigma) \leq 0, \qquad \text{for all } i \in \{1, \ldots, m\}, \tag{7b}$$

which do not lead to the violation of LICQ and MFCQ. The function $\Phi : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a regularization function. A smaller value of σ yields a better approximation to the original problem and for $\sigma = 0$ we recover the original constraint (3d). Next one solves a (finite) sequence of these NLPs, by driving the parameter $\sigma > 0$ to zero. The availability of robust NLP codes makes their implementation easy and practical. We denote the solution of the initial MPCC (3) by w^* and the solution of the regularized NLP by $w^*(\sigma)$. The obvious goal is that $w^*(\sigma) \rightarrow w^*$ as $\sigma \rightarrow 0$. Hoheisel et al. [47] provide a detailed numerical and theoretical comparison of several methods from this family. If the problems are solved exactly, under mild assumptions, accumulation points of the sequence of solutions $w^*(\sigma_k)$ are usually C-stationary points [47, 90, 93]. Solving the NLPs inexactly usually weakens the convergence results [58]. Of course, stronger assumptions also result in convergence to M- or S-stationary points. In the sequel, we discuss several examples of functions Φ and strategies for driving σ to zero.

3.2.1 Direct Solution

One can use standard NLP techniques such as Sequential Quadratic Programming (SQP) and Interior Point (IP) methods for directly solving (3) [27, 29, 65, 70]. Despite the degeneracy discussed so far, this approach can sometimes perform surprisingly well in practice. However, it tends to have convergence difficulties or to converge to spurious stationary points if the

MPCC-LICQ does not hold. Fletcher and Leyffer study the practical performance of SQP methods on numerous MPCCs in [28] and investigate their local convergence properties in [29]. Under the assumptions that the MPCC-LICQ and that all QPs remain feasible, and other technical assumptions, they show quadratic convergence to S-stationary points. Interior-point methods perform reasonably well if applied directly to the NLP formulation (3) [101]. However, their performance improves when paired with relaxation and exact penalty formulations, as we will highlight several times below.

NLP formulations with NCP functions In this approach the complementarity constraints (3d) are replaced by a C-function $\Phi_{\mathrm{C}}(G(w), H(w)) = 0$. The resulting NLP is solved by a standard globalized NLP solver. The use of SQP methods in such formulations was studied by Leyffer [65]. If the used C-function is not differentiable at $(0, 0)$, its subgradient can be used [65]. Evidently, using squared versions (or higher powers) of C-functions will not improve the situation and lead to a violation of LICQ at $(0, 0)$ since we obtain a zero-gradient at this point. Similar to [29], assuming MPCC-LICQ, Lipschitz continuity of the NLP functions and their derivatives, and other technical assumptions, local superlinear convergence to S-stationary points was shown. Leyffer [65] tests this approach on a wide number of test problems and shows that different NCP functions can lead to large differences in performance.

3.2.2 Global Relaxation/Smoothing Method by Scholtes

Scholtes introduces arguably the easiest-to-implement approach and relaxes the complementarity constraint by using [93]:

$$\Phi_{\mathrm{S}}(G_i(w), H_i(w), \sigma) = G(w)_i H_i(w) - \sigma. \tag{8}$$

An illustration of the relaxed feasible set is given in Fig. 4 (a). Alternatively, the bilinear term might be smoothed by using $\Phi_{\mathrm{S}}(G_i(w), H_i(w), \sigma) = 0$ in (7b). Lumped versions $\Phi_{\mathrm{S}}(G(w), H(w), \sigma) = G(w)^\top H(w) - \sigma$ are also used frequently [93]. Observe that in contrast to the smoothed variant, the relaxed version contains the feasible set Ω_{MPCC}, and one might find with it a stationary point without driving $\sigma \to 0$. Assuming MPCC-LICQ, Scholtes [93] shows convergence to C-stationary points. Hoheisel et al. [47] obtain the same result under the weaker MPCC-MFCQ. Ralph et al. [90] study the convergence speed of this approach and show that, under rather strict assumptions, the local solution map $w^*(\sigma)$ of the relaxed formulation is a piecewise continuous function and that the solution converges with a rate $O(\sigma)$. Milder assumptions result in the rate $O(\sigma^{\frac{1}{2}})$ for the relaxed variant and $O(\sigma^{\frac{1}{4}})$ for the smoothed variant.

For more efficacy, Raghunathan and Biegler [89], Liu and Sun [70] propose interior-point methods where the relaxation parameter σ is proportional to the barrier parameter τ. Under some stronger assumptions, this strategy is shown to converge to S-stationary points.

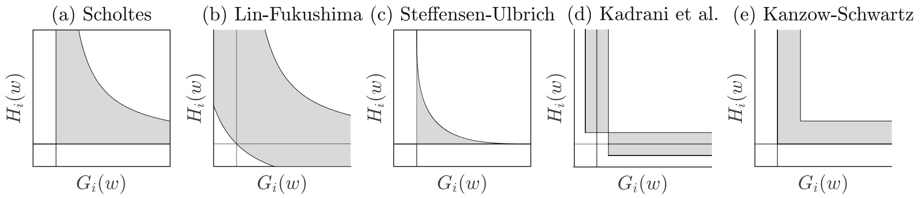


Fig. 4 Illustration of the regularized complementarity sets

Smoothed NCP functions Early MPCC algorithms considered smoothed and everywhere differentiable variants of C-functions. These methods are closely related to Scholtes' approach since one can often by simple algebraic manipulation obtain the same feasible set as with using (8). In our experiments, we will test three different smoothed NCP functions: the smoothed Fischer–Burmeister (FB) function $\Phi_{\mathrm{FB}}(a, b, \sigma) = a + b - \sqrt{a^2 + b^2 + \sigma^2}$, the Natural Residual (NR) function $\Phi_{\mathrm{NR}}(a, b, \sigma) = (a+b-\sqrt{(a-b)^2 + \sigma^2})$, the Chen–Chen–Kanzow (CCK) [16] function $\Phi_{\mathrm{CCK}}(a, b, \sigma) = \lambda(a+b-\sqrt{a^2 + b^2 + \sigma^2})+(1-\lambda)(ab-\sigma)$, where $\lambda \in (0, 1)$ is a parameter, which is set to 0.5 in our implementations. Facchinei et al. [21] show the convergence of this approach to C-stationary points. In our implementation, we use the smoothed versions of the NCP functions in the form of (7). Hence, we obtain relaxations of the original problem (3).

3.2.3 The Smooth Relaxation Method by Lin and Fukushima

This method is similar to Scholtes' regularization and replaces the complementarity conditions (3) by:

$$\begin{aligned} &G_i(w)H_i(w) \leq \sigma^2, && \text{for all } i \in \{1, \dots, m\},\\ &(G_i(w)+\sigma)(H_i(w)+\sigma) \geq \sigma^2, && \text{for all } i \in \{1, \dots, m\}. \end{aligned}$$

Figure 4(b) shows an illustration of the feasible set. Lin and Fukushima [68] obtain similar convergence results as Scholtes [93]. Hoheisel et al. [47] extend this result by proving convergence to C-stationary points under the MPCC-MFCQ. Moreover, they show that the feasible points of the relaxed NLP satisfy the MFCQ in a neighborhood of a point $x \in \Omega_{\mathrm{MPCC}}$.

We continue with reviewing more sophisticated regularization schemes that converge to M-stationary points under fairly mild conditions [47]. As we will see later, this may not always imply better performance in practice.

3.2.4 The Local Relaxation Method by Steffensen and Ulbrich

Almost all regularization methods make global changes to the feasible set. Motivated by the fact that most difficulties arise for degenerate complementarity pairs $i \in \mathcal{I}_{00}$, Steffensen and Ulbrich follow a different approach [96, 102]. Their main idea is to relax the complementarity constraint only locally at the corner of the L-shaped set arising from the complementarity constraints.

The relaxation is achieved by the following steps: the L-shaped set is rotated with a linear transformation by $\frac{\pi}{4}$ counterclockwise for every complementarity pair, and one obtains the graph of the abs-function. On the interval $[-\sigma, \sigma]$, the kink is replaced by a sufficiently smooth function such that the continuity of the functions and their derivatives is preserved at the interval boundaries. Finally, the inverse transformation is carried out, and a locally relaxed set is obtained, cf. Fig. 4 (c). This reasoning expressed in equations reads as

$$\Phi_{\mathrm{SU}}(G_i(w), H_i(w), \sigma) \leq 0, \qquad \text{for all } i \in \{1, \dots, m\},$$

where $\Phi_{\mathrm{SU}} : \mathbb{R}\times\mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is defined in terms of the auxiliary functions $\phi^a_{\mathrm{SU}} : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ and $\phi^b_{\mathrm{SU}} : [-1, 1] \to \mathbb{R}$ as follows

$$\begin{aligned} \Phi_{\mathrm{SU}}(y_1, y_2; \sigma) &= y_1 + y_2 - \phi^a_{\mathrm{SU}}(y_1 - y_2, \sigma),\\ \phi^a_{\mathrm{SU}}(z, \sigma) &= \begin{cases} |z| & \text{if } |z| \geq \sigma,\\ \sigma\phi^b_{\mathrm{SU}}(\frac{z}{\sigma}) & \text{if } |z| < \sigma. \end{cases} \end{aligned}$$

The function ϕ_{SU}^{b} has to satisfy some smoothness and mononoticitiy properties [96]. For our experiments we implement two variants of such functions proposed in [96]: $\phi_{\mathrm{SU}}^{b}(z) = \frac{1}{8}(-z^4 + 6z^2 + 3)$ and $\phi_{\mathrm{SU}}^{b}(z) = \frac{2}{\pi}\sin(z\frac{\pi}{2} + \frac{3\pi}{2}) + 1$. Under the MPCC-CRCQ (cf. [47, Definition 2.4]) convergence to C-stationary and under the MPCC-LICQ to M-stationary points is shown [96].

3.2.5 The Nonsmooth Relaxation Method by Kadrani et al.

Another interesting relaxation by Kadrani et al. [56] reads as:

$$\begin{aligned} &G_i(w) \geq -\sigma, \quad H_i(w) \geq -\sigma, && \text{for all } i \in \{1, \ldots, m\},\\ &(G_i(w) - \sigma)(H_i(w) - \sigma) \leq 0, && \text{for all } i \in \{1, \ldots, m\}. \end{aligned}$$

Figure 4(d) illustrates the nonsmooth feasible set obtained from the constraint above. The convergence study of [56] is carried out assuming the MPCC-LICQ. Once again, Hoheisel et al. [47] improve the result and show convergence to M-stationary points under the MPCC-CPLD (cf. [47, Definition 2.4], a CQ weaker than the MPCC-MFCQ and stronger than the MPCC-ACQ). It is evident from the structure of the feasible set of this relaxation that verifying standard CQs is more difficult.

3.2.6 The Relaxation Method by Kanzow and Schwartz

This relaxation has stronger theoretical properties than the previous one and a more satisfactory shape of the feasible set, cf. Fig. 4 (e) [57]. In contrast to the approach of Kadrani et al., it contains the feasible set of the MPCC. This relaxation is modeled with the following equations:

$$\Phi_{\mathrm{KS}}(G_i(w), H_i(w), \sigma) \leq 0, \quad \text{for all } i \in \{1, \ldots, m\},$$

with $\Phi_{\mathrm{KS}} : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $\phi_{\mathrm{KS}} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, where $\Phi_{\mathrm{KS}}(y_1, y_2, \sigma) = \phi_{\mathrm{KS}}(y_1 - \sigma, y_2 - \sigma)$ and

$$\phi_{\mathrm{KS}}(y_1, y_2) = \begin{cases} y_1 y_2 & \text{if } y_1 + y_2 \geq 0,\\ -\frac{1}{2}(y_1^2 + y_2^2) & \text{if } y_1 + y_2 < 0. \end{cases}$$

The function ϕ_{KS} is a continuously differentiable C-function [57]. Under the MPCC-CPLD (cf. [47, Definition 2.4]) convergence to M-stationary points is shown [47, 57].

3.2.7 Exact Penalty Methods

Exact penalty reformulations are one of the most often used approaches to treat degenerate NLPs [9, 13, 37]. In exact penalty algorithms for MPCCs, the bilinear term (3d) is added to the objective in some suitable form and multiplied by a penalty factor ρ. To be consistent with our notation above, and the implementations in NOSNOC [78], we use $\rho^k = \frac{1}{\sigma_k}$. Assuming sufficient regularity of other constraints and having the bilinear term in the objective, we obtain a regular NLP. Under suitable assumptions and for a sufficiently large and finite ρ, the solution $w^*(\sigma)$ matches the solution w^* of the initial MPCC after a single NLP solve [90]. In practice, a sequence of NLPs is solved to improve the convergence and to estimate the correct penalty parameter value. One of the simplest formulations is to penalize the term

$G(w)^\top H(w)$ in the objective. This corresponds to the ℓ_1 norm of the complementarity residual. Therefore, we solve a sequence of the following NLPs:

$$\begin{aligned} \min_{x} \quad & f(w) + \frac{1}{\sigma_k} G(w)^\top H(w) && \text{(9a)} \\ \text{s.t.} \quad & g(w) = 0, && \text{(9b)} \\ & h(w) \geq 0, && \text{(9c)} \\ & G(w) \geq 0, \quad H(w) \geq 0. && \text{(9d)} \end{aligned}$$

This approach was first proposed in [23] for solving practical problems. Anitescu [3] provided the first convergence analysis for the ℓ_1 penalty approach paired with active-set SQP methods. Ralph et al. [90] show that an MPCC solution w^* is also a solution to (9) for a sufficiently large ρ and that regularity conditions of the MPCC (e.g., MPCC-LICQ) imply regularity of the NLP (9). However, if the local minimizers are only B-stationary but not S-stationary points, the penalty parameter must grow to infinity [59].

Leyffer et al. [66] propose an interior-point algorithm to solve the NLP (9) while dynamically updating the penalty parameter ρ. For each fixed ρ^k, the barrier subproblem is solved inexactly to a tolerance proportional to the barrier parameter τ^k. Strategies to steer the penalty parameter that avoid too large increases and unbounded subproblems are proposed as well. Fukushima et al. [33] suggest an SQP method paired with a penalized NCP function. Hu and Ralph [50] relate relaxation methods of [93] and give conditions for convergence to B-stationary points. They study more general formulations than (9) and suggest for example to use $\sum_{i=1}^{m} \Phi_{\mathrm{FB}}(G_i(w), H_i(w))^3$ as a penalty function. Furthermore, by comparing the KKT conditions, Leyffer et al. [66] show that there exists a one-to-one correspondence between the iterates of the penalty and a smoothing Scholtes approach (i.e., the bilinear constraint in (3d) is replaced by $G_i(w)H_i(w) = \sigma_k$).

Hall et al. [43, 44] propose a Sequential Convex Programming (SCP) method for solving the penalty problem arising from quadratic programs with complementarity constraints. In particular, the method makes use of the fact that QP matrices do not need to be re-factorized in an SCP approach, which enables fast and cheap iterations. The algorithm is paired with an exact analytic line search.

The great practical difficulty in exact penalty methods is steering the penalty parameter. Byrd et al. [13] introduce a line-search SQP ℓ_1-exact penalty method for general degenerate NLPs. They propose penalty update rules based on solving LPs and QPs with a trust region to predict the decrease of the merit function. Favorable theoretical properties and good numerical performance on a series of test problems including MPCCs are reported. Thierry and Biegler [101] adapt the ℓ_1 strategy of Byrd et al. [13] including the penalty steering rules, to solve degenerate problems with IPOPT [103]. Good practical performance is reported on the MacMPEC test set [64] with an improvement in terms of speed and robustness compared to a direct application of IPOPT.

Moreover, using an ℓ_∞ penalty function for MPCC is also very common [9]. The complementarity constraints (3d) are replaced by:

$$G(w) \geq 0, \quad H(w) \geq 0, \quad G_i(w)H_i(w) \leq s, \qquad \text{for all } i \in \{1, \dots, m\},$$

where $s \in \mathbb{R}$ is a slack variable, which may have an upper bound $\bar{s} > 0$. The term $\frac{1}{\sigma}s$ is added to the objective function. This enables us to express the ℓ_∞ norm of the complementarity constraints residual smoothly. Similarly, if the bilinear terms are lumped together, and we use the constraint $G(w)^\top H(w) \leq s$, we end up with an ℓ_1 formulation that is equivalent to (9).

A mixture of the approaches above is the elastic mode, which takes an ℓ_1 norm of the bilinear complementarity terms and an ℓ_∞ penalization of the relaxation of the standard equality and inequality constraints. Anitescu et al. [5] show under the MPCC-LICQ and other assumptions the global convergence of an elastic mode SQP approach to C-, M- and S-stationary points. The elastic mode with a fixed penalty parameter is implemented in SNOPT as a fallback strategy if an infeasible or unbounded QP is detected [38].

Finally, we mention the family of augmented Lagrangian methods for MPCC, which also belong to the class of penalty methods [34, 52]. We do not treat them in detail here. Assuming MPCC-LICQ, and that the sequence of Lagrange multipliers is bounded, convergence to S-stationary points is shown in [34, 52].

3.2.8 Lifting Methods

Lifting methods are somewhat in between relaxation and penalty methods. The main idea is to introduce *lifting variables* and regard a more regular feasible set in a higher-dimensional space whose orthogonal projection is the L-shaped set, coming from the complementarity constraints. Some of them require penalization of the lifting variables to recover the solution of the initial problem [45], and others might require additional regularization [97]. Unfortunately, they have weaker theoretical properties than regularization methods, cf. Section 3.4. Thus, we do not implement these methods and do not treat them in further detail. We mention the methods of Stein [97], Hatz et al. [45], and Izmailov et al. [52].

3.3 Steering the Homotopy Parameter to Zero

The initial value for the homotopy parameter σ_0 and deciding how to steer it to zero play an important role in the practical performance of the relaxation-based methods. In our implementations, we take three different approaches in steering the relaxation in (7):

1. directly change the relaxation parameters σ outside of the homotopy loop, which is the standard approach in most of the literature,
2. steer a single relaxation parameter with an ℓ_∞ penalty approach to zero,
3. steer several relaxation parameters with an ℓ_1 penalty approach to zero.

The ways in which the different approaches manifest in a relaxed NLP are summarized in Table 1. In the standard approach, we simply use a fixed parameter σ_k for (7b) in every NLP solve, and update the parameter after every iteration via:

$$\sigma_{k+1} = \kappa \sigma_k, \tag{10}$$

with $\kappa \in (0, 1)$. Alternatively, we may use the update formula (as often used in IP methods [73]):

$$\sigma_{k+1} = \min(\kappa \sigma_k, \sigma_k^\eta),$$

with $\eta > 1$. Next, we may let the optimizer steer the relaxation, by using e.g., the same scalar variable s in all (7b). The slack variable is pushed to zero by being more and more penalized in the objective function, with a weight of $\rho_k = \frac{1}{\sigma_k}$. In the case of Scholtes' relaxation, we recover the ℓ_∞ penalty approach discussed in Section 3.2.7. Similarly, we can add a new scalar variable s_i for every constraint in (7b) and penalize their deviation from zero in the objective and weighted by $\rho_k = \frac{1}{\sigma_k}$. In the case of Scholtes' relaxation, we recover the exact ℓ_1 penalty approach and obtain a NLP equivalent to (9).

Table 1 Homotopy parameter steering strategies

Standard relaxation	ℓ_∞ penalty relaxation	ℓ_1 penalty relaxation
$\min\limits_{w \in \mathbb{R}^{n_w}} f(w)$ s.t. $0 = g(w),$ $0 \leq h(w),$ $0 \leq G(w),$ $0 \leq H(w),$ $0 \geq \Phi(G_i(w), H_i(w), \sigma_k),$ for all $i \in \{1, \dots, m\}$.	$\min\limits_{\substack{w \in \mathbb{R}^{n_w} \\ s \in \mathbb{R}}} f(w) + \frac{1}{\sigma_k} s$ s.t. $0 = g(w)$ $0 \leq h(w)$ $0 \leq G(w)$ $0 \leq H(w)$ $0 \geq \Phi(G_i(w), H_i(w), s),$ for all $i \in \{1, \dots, m\}$.	$\min\limits_{\substack{w \in \mathbb{R}^{n_w} \\ s \in \mathbb{R}^m}} f(w) + \frac{1}{\sigma_k} \sum_{i=1}^{m} s_i$ s.t. $0 = g(w),$ $0 \leq h(w),$ $0 \leq G(w),$ $0 \leq H(w),$ $0 \geq \Phi(G_i(w), H_i(w), s_i),$ for all $i \in \{1, \dots, m\}$.

In our experiment, we solve all NLPs in the homotopy loop to the same prescribed tolerance. Alternatively, one may solve the NLPs in the homotopy loop inexactly. Some authors suggest updating the relaxation parameter simultaneously or tying it to other relaxation parameters of the algorithm. Raghunathan and Biegler [89] update the parameter σ simultaneously with the barrier parameter τ in an IP method. Similarly, in Leyffer et al. [66] in an IP-ℓ_1 exact penalty approach the parameter update is related to the update of τ. Lin and Othsuka [69] use a non-interior point method with Scholtes' relaxation. The parameter of the Scholtes' relaxation and the complementarity constraint relaxation parameter in the KKT conditions are updated simultaneously.

To make the various relaxations better comparable and less dependent on the scaling of σ, we bring them to a similar scale for a given σ. In particular, for a given σ, we want that the distance of $\Phi(a, b, \sigma) = 0$ to the origin is always the same, independent of the particular choice of Φ. Moreover, for a given κ we require that this distance shrinks with the same rate for all relaxations. By simple algebraic manipulations one can find how σ needs to enter a given relaxation function. Our initial experiments confirm that this makes the performance more consistent and better comparable.

3.4 Summary of MPCC Methods

With a standard NLP solver at hand, all methods from Section 3.2 paired with the steering strategies from Section 3.3 are easy to implement. Table 2 provides an overview of known convergence results for direct, regularization, lifting, penalty methods, and active-set methods. Together with the diagram in Fig. 3, it can help one to decide which algorithm to choose to compute a stationary point of the MPCC (3). The strongest multiplier-based stationarity concept is S-stationarity, followed by M-, C-, A- and W-stationarity, sorted from stronger to weaker. One should not be discouraged by the weaker limiting points of the relaxation methods. In contrast to the other methods, these results are obtained under much weaker assumptions. In general, more restrictive assumptions give a better result. For example, the global relaxation method by Scholtes [93] converges to B-stationary points under the MPCC-LICQ and the upper-level strict complementarity, cf. [90, Definition 2.6]. We note that all methods converging to an S-stationary point under the MPCC-LICQ also assume other restrictive assumptions, such as the upper-level strict complementarity [31, 45, 65, 89].

Table 2 Overview of convergence properties of MPCC methods

Type	Method	CQ for MPCC	Limiting Point	Subproblem NLP satisfies	Citation
Direct	Fletcher et al.	MPCC-LICQ	S	GCQ	[29, 31]
	Leyffer	MPCC-LICQ	S	GCQ	[66]
Regularization	Scholtes	MPCC-MFCQ	C	MFCQ	[47, 93]
	Lin-Fukushima	MPCC-MFCQ	C	MCFQ	[47, 68]
	Kadrani et al.	MPCC-MFCQ	M	GCQ	[47, 56]
	Steffensen-Ulbrich	MPCC-CPLD	C	ACQ	[47, 96]
	Kanzow-Schwartz	MPCC-CPLD	M	GCQ	[47, 57]
	Raghunathan-Biegler	MPCC-LICQ	C	MFCQ	[89]
Lifting	Stein	MPCC-LICQ	C	LICQ	[97]
	Izmailov-Solodov	MPCC-LICQ	C	LICQ	[51]
	Hatz et al.	MPCC-LICQ	S	GCQ	[45]
Penalty	ℓ_1-Penalty	MPCC-LICQ	S	LICQ	[65, 90]
	Leyffer et al.	MPCC-LICQ	C	LICQ	[65, 90]
	ℓ_∞-Penalty	MPCC-LICQ	S	LICQ	[9, 90]
	Elastic mode	MPCC-LICQ	C	MFCQ	[4, 5]
Active-set	Leyffer-Munson	MPCC-MFCQ	B	GCQ	[67]
	Kirches et al.	–	B	GCQ	[61]
	Guo-Deng	MPCC-RCPLD	M	GCQ	[40]

In practice, the NLP subproblems cannot be solved exactly, due to the solver tolerances and finite arithmetic precision. However, solving the NLPs in the sequence inexactly can weaken the convergence results [5, 58]. For example, under inexact solves the methods of Kadrani et al., Kanzow–Schwartz [57] and Steffensen–Ulbrich [96] converge only to W-stationary points. Surprisingly, the methods of Scholtes and Lin–Fukushima are immune to this, and they still converge to C-stationary points [58]. However, if some stronger assumptions are not satisfied, usually the MPCC-LICQ, they can experience slow convergence rates and converge to weaker points. This motivated the development of combinatorial methods, which have excellent theoretical properties but currently no mature open-source implementations.

4 A Benchmark Set of MPCC from Nonsmooth OCPs

The main contribution of this paper is the introduction of a benchmark suite of MPCCs that come from nonsmooth optimal control and simulation problems. We discuss the problem format, provide references for the original continuous time OCP and simulation problems, and discuss how we generate MPCCs from them. Moreover, we split the whole problem set into several subsets to facilitate the testing of the variety of available algorithms.

4.1 Problem Format

In the benchmark we provide the MPCCs in a slightly different format than in (3):

$$\min_{w \in \mathbb{R}^n} f(w, p) \tag{11a}$$

$$\text{s.t.} \quad \ell_w \leq w \leq u_w, \tag{11b}$$

$$\ell_g \leq g(w, p) \leq u_g, \tag{11c}$$

$$0 \leq G(w, p) \perp H(w, p) \geq 0, \tag{11d}$$

where $p \in \mathbb{R}^{n_p}$ a given parameter. If we disregard (11d), this format compiles with the CasADi NLP solver interface [2]. All problem functions in (11) are CasADi functions generated via NOSNOC. By treating the complementarity constraint with some of the methods described in Section 3.2 we obtain an NLP that can be directly passed to the CasADi NLP solver interface. On the benchmarks homepage[1] we provide all MPCCs in the form of (11) format using a JSON object with the following components:

- w as a string encoded CasADi variable with the key "w", along with the ℓ_w and u_w with the labels "lbw" and "ubw", respectively,
- $f(w)$ as a string encoded CasADi function with the key "f_fun",
- $g(w)$ as a string encoded CasADi function with the key "g_fun", along with the ℓ_g and u_g with the labels "lbg" and "ubg", respectively,
- $G(w)$ and $H(w)$ as string encoded CasADi functions with keys "G_fun" and "H_fun", respectively,
- parameters p and their values as string encoded CasADi variables and an array of doubles with keys "p" and "p_val", respectively.

From this, a user can simply reconstruct the problem by loading in all of the components and using the provided CasADi deserialization functionality to interface their solver with NOSBENCH problems.

4.2 Description of the Benchmark Collection

The NOSBENCH problems are generated by using the algorithmic toolchain available in NOSNOC. In particular, we regard both simulation and optimal control problems, reformulate them, and discretize them with the FESD method [83, 84]. We list in Table 3 the origins of each of these problems in continuous time along with references to them in the literature. We briefly describe each of the systems along with a classification. Moreover, we also list the number of MPCCs we generated from the particular continuous-time problem. For more details on each discrete-time problem, we refer the reader to the benchmark's homepage.

4.2.1 Reformulation and Discretization Options

Depending on the reformulation and discretization method, the continuous-time problems can be reformulated into significantly different MPCCs. In order to generate problems of varying complexity and internal structure we vary several discretization and MPCC parameters. In all cases we obtain an MPCC of the form of (2). We list some variations, which are available in NOSNOC:

[1] NOSBENCH homepage: https://github.com/apozharski/nosbench.

Table 3 Continuous time problems used to generate the NOSBENCH test set

Nr.	Problem name	Short description	Type	Class	Number of MPCCs	Citation
1	3CPCLS	Two dimensional representation of three carts with only one actuated.	OCP	CLS direct	18	[69]
2	3CPTF	One dimensional representation of three carts with only one actuated.	OCP	CLS time-freezing	18	[69]
3	CARHYS	Turbo car example with hysteresis.	OCP	HYS time-freezing	12	[77]
4	CARTIM	Turbo car example with velocity-dependent switch.	OCP	Fillipov System	24	[83]
5	CPWF	Inverted pendulum on a cart with Couloumb friction.	OCP	Fillipov System	48	[48]
6	DAOBCLS	Disc control example.	OCP	CLS direct	9	[84]
7	DISCM	MDisc control example.	OCP	CLS time-freezing	6	[84]
8	DRNLND	Two dimensional control of a landing drone.	OCP	CLS time-freezing	18	[62]
9	DSCOB	Disc control example with a central obstacle, modeled as a nonlinear constraint.	OCP	CLS time-freezing	27	[84]
10	DSCSP	Disc control example with discs switching positions.	OCP	CLS time-freezing	9	[84]
11	HOPOCP	Hopping robot actuated with a linear leg and reaction wheel in two dimensions.	OCP	CLS time-freezing	12	[48]
12	MFTOPT	Time optimal control of a linear voice-coil motor.	OCP	Fillipov System	24	[17]
13	MNPED	Control of a planar monoped robot.	OCP	CLS time-freezing	36	[15]
14	MWFOCP	Optimal control of a linear voice-coil motor.	OCP	Fillipov System	24	[17]
15	SCHUMI	Simple two-dimensional car model with track constraints. Both time optimal and non-time optimal forms.	OCP	Fillipov System	72	[100]
16	SMOCP	Optimal control problem with multiple sliding modes.	OCP	Fillipov System	48	[83]
17	TFBIB	Control of a planar particle in a box with a rotating reference	OCP	CLS time-freezing	12	[81]
18	TNKCSC	Cascade of tanks with state-dependent switches in flow rate.	OCP	Fillipov System	6	[8]

Table 3 continued

Nr.	Problem name	Short description	Type	Class	Number of MPCCs	Citation
19	TIMF1D	One dimensional particle with contacts.	Simulation	CLS time-freezing	12	[84]
20	2BCLS	Two balls connected with a stiff spring, that undergo contact dynamics.	Simulation	CLS direct	9	[11]
21	986EQ	Simplified model of structural pounding used in the study of the effects of earthquakes.	Simulation	Fillipov System	12	[14]
22	986FO	Two masses linked by a spring and moving on a surface with Coulomb friction.	Simulation	Fillipov System	18	[14]
23	986FV	Oscillator with varying friction force.	Simulation	Fillipov System	18	[14]
24	986OM	Oscillating mass with contact surfaces.	Simulation	Fillipov System	12	[14]
25	CLS1D	One dimensional particle with contacts.	Simulation	CLS direct	12	[84]
26	FBS1S	Spring-connected blocks with Coulomb friction.	Simulation	Fillipov System	18	[99]
27	OSCIL	An unstable oscillator with a state dependent switch.	Simulation	Fillipov System	12	[83]
28	RFB1S	Relay feedback system.	Simulation	Fillipov System	18	[87]
29	SMCRS	Simple Filippov system with a boundary crossing.	Simulation	Fillipov System	6	[83]
30	SMLSM	Simple Filippov system with state entering a sliding mode.	Simulation	Fillipov System	6	[83]
31	SMSLM	Simple Filippov system with state leaving a sliding mode.	Simulation	Fillipov System	6	[83]
32	SMSPS	Simple Filippov system with state spontaneously leaving an unstable sliding mode.	Simulation	Fillipov System	6	[83]
33	TFPOB	A collapsing pile of balls with contacts	Simulation	CLS time-freezing	12	[78, 81]

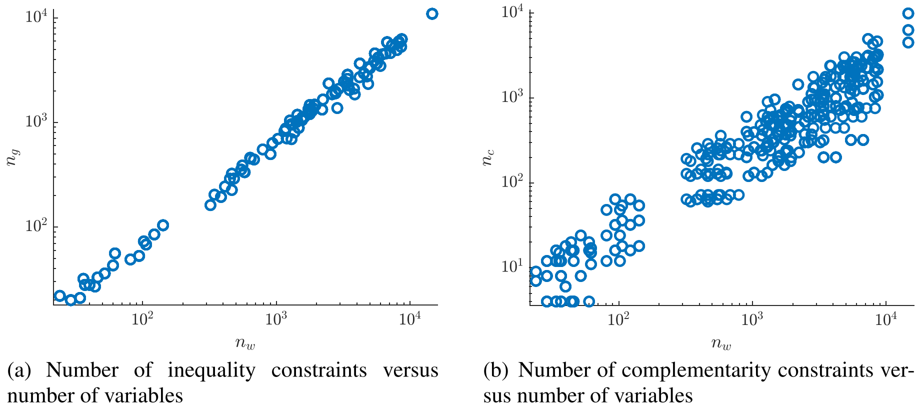
- We distinguish between simulation and optimal control problems (OCPs).
- If the system has a nonsmooth right hand side (r.h.s.), but no state jumps, we treat it as Filippov system, which one can reformulate either via Stewart's [83] or via the Heaviside step reformulation [80] into a DCS. Afterwards, they are discretized with the corresponding FESD method.
- Complementarity Lagrangian Systems (CLS) with state jumps can be either treated directly with FESD-J [84] or reformulated via time-freezing into a Filippov system [76, 81]. Hybrid systems with hysteresis are always reformulated via time-freezing into Filippov systems.

Moreover, we can vary the discretization parameters:

- N - the number of control intervals in the OCP. The controls are piecewise constant after the discretization. This is always set to $N = 1$ in the case of simulation problems.
- N_{FE}: number of integration steps (finite elements) within a control interval.
- The number of stage points used by the selected Runge–Kutta (RK) method within each finite element in the FESD discretization is given by n_{s}.
- Choice of underlying RK method (defined by its Butcher tableau) in FESD discretization. In our experiments, we regard the Radau-IIA or Gauss–Legendre schemes.
- There are several choices for grouping the cross-complementarity conditions in the FESD discretization, cf. [83]. They provide different sparsity to the complementarity constraints that enforce switch detection.

Furthermore, in some cases, we vary the problem parameters. This is done to provide some variety in the benchmark problems and to mitigate the effects of pre-tuning which has been done on some of these examples in the corresponding references. It also allows for simulation problems to be tested both in cases where there are no switches and in cases where switches must be detected.

By varying all the aforementioned parameters we generate a total of 603 distinct MPCCs within the full problem set. Due to space limitations, we cannot provide full details in this paper, but they are available on the benchmark's homepage. Figure 5 shows the number of primal variables versus the dimensions n_g of $g(w)$ in (11) and versus the number of complementarity pairs in each problem.



(a) Number of inequality constraints versus number of variables

(b) Number of complementarity constraints versus number of variables

Fig. 5 Size characteristics of the NOSBENCH test set

4.2.2 Problem Naming Convention

The names of the problem files are encoded with some information about the problem. This is done so that the formation of subsets of the whole benchmark can be done systematically. The structure of these names is an underscore delimited string containing the following data in order: Problem name, parameter index, N, N_{FE}, n_{s}, RK method, DCS type ("Step", "Stewart", or "CLS"), cross complementarity mode, source, and a boolean flag if the problem is lifted into the vertical form. The type of the problem is split into "FIL", Fillipov systems, "IEC", problems with only inelastic collisions, "ELC", problems with elastic (and possibly inelastic) collisions, and "HYS" for problems containing hysteresis. For example the problem titled: `986EQ_001_001_003_2_GL_STEP_7_FIL_1` would be the earthquake example from [14], with the first parameter set and $N = 1$, $N_{\mathrm{fe}} = 3$, $n_s = 2$, using a Gauss-Legendre integrator. The problem is generated using the Step reformulation and cross-complementarity mode 7 (cf. [74]), and is lifted into the vertical form.

4.2.3 Problem Subsets

It takes a lot of CPU time to run the full suite and doing so is not necessary to gain some first insight into the comparative performance of different solution methods. Therefore, we provide several smaller subsets of problems that can be used to benchmark any future solvers and are used in Section 5 to evaluate existing methods.

Simple problem benchmark - `NOSBENCH-S` The first subset of `NOSNOC` is a benchmark that only uses 100 MPCCs which come exclusively from Fillipov systems and only contain the simplest time-freezing and CLS examples. These tend to produce relatively easier-to-solve MPCCs and as such this set is an effective way to identify particularly poor-performing solvers. It contains approximately equivalent numbers of simulation and optimal control problems and typically all of the problems can be solved with the existing state-of-the-art in less than an hour per problem.

Representative small benchmark - `NOSBENCH-RS` This subset of `NOSBENCH` is an even smaller but more representative benchmark. It contains 32 MPCCs that include ones from FESD-J and time-freezing reformulations. This subset is primarily used for a second preliminary screening of solvers as it provides more insight into the performance of solvers on problems ranging from the easiest to the most difficult within `NOSBENCH`.

Representative large benchmark - `NOSBENCH-RL` This subset is a set of 167 problems that is made up of a representative sample of all problem difficulties. This benchmark is meant to be the main problem set that is used to benchmark solvers, and will continue to be expanded as new and interesting problems are added to `NOSBENCH`.

The full benchmark collection - `NOSBENCH-F` This set contains all 603 MPCCs generated in this version of `NOSBENCH`. We aim to test the most competitive solver-method combinations on this set to get a clear performance picture.

4.3 Stopping Criteria and Solution Quality

Beyond the NLP stopping criteria used in the underlying NLP solver, we use a further stopping criterion on the homotopy iteration that is based on the "complementarity residual" which is defined as:

$$r_{\perp}(w) = \max(\{G_i(w)H_i(w) \mid \text{for all } i \in \{1, \dots, m\}\}).$$

The stopping condition used is the achievement of a successful NLP solve in the homotopy loop, whose result has a complementarity residual smaller than a given tolerance. For all the following experiments (except if otherwise noted) we use a complementarity residual tolerance of `comp_tol` $= 10^{-7}$. In the case of `IPOPT` we treat any solution that is reported as optimal or "solved to an acceptable level" as a successful solve. We further accept solutions where the search direction becomes too small if they meet the complementarity tolerance as well as are primal feasible. We set the runtime limit for a single MPCC solve for all solvers to one hour (3600 seconds) cumulative wall time.

When analyzing the results in the next section, we include in the analysis the quality of a given solution if the problem comes from a discretized OCP. We do not apply this check for simulation problems as the solutions of such problems are usually unique or at least locally isolated [83]. This verification is done by checking the relative objective value of a given problem-solver pair against the best-known found solution. We then treat solutions that exceed the best-known objective by at least a factor of two as failures. This approach is used in order to better evaluate the solution methods specifically in an optimal control context as in this context a significantly worse solution is often a sign of a failure of the solver to achieve the goals of the controller.

5 Computational Results

In this section, several experiments are carried out to evaluate different solution methods for MPCCs. We first compare the different regularization methods discussed in Section 3.2 paired with the different homotopy parameter steering strategies from Section 3.3 (and Table 1). We then explore the space of the homotopy parameters which are used to drive the complementarity regularizations toward the exact complementarity set. This is followed by a comparison of three NLP solvers (`IPOPT` [103], `SNOPT` [37], and `WORHP` [12]) used to solve the regularized NLPs. Moreover, we identify the type of stationary points to which the solver converges. In case they are not S-stationary, we solve an LPCC to check if we have a B-stationary point. Further numerical results on the `NOSBENCH` test set can be found in the master thesis [88].

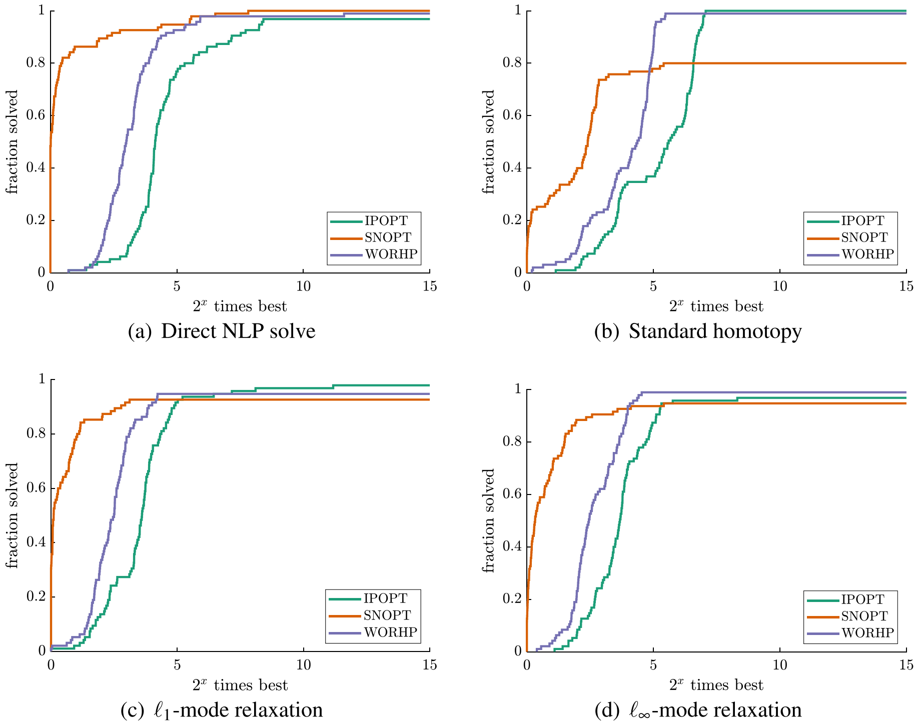
5.1 General Experiment Setups

All benchmarks are run using an Intel Xeon W-2225 4-core processor with a base clock of 4.1 GHz and a boost clock of 4.6 GHz. In all cases where we are not explicitly varying the NLP solver, we used `IPOPT` [103] as the default option. It is used with its default options except for the following changes: `bound_relax_factor = 0`, `mu_strategy = adaptive`, `mu_oracle = quality-function`, `acceptable_tol` $= 10^{-6}$, `tol` $=10^{-12}$, `dual_inf_tol` $= 10^{-12}$, `comp_inf_tol` $= 10^{-12}$. We also default to using `MA27` [20] as the linear solver in `IPOPT` as it has, in our experience, been the most stable of the `HSL` solvers [49]. This is due to our observation that both `MA57`, and `MA97`, the solvers recommended as state of the art, occasionally cause IPOPT crashes due to segmentation faults. The single-threaded nature of MA27 also allows us to run multiple IPOPT instances in parallel in order to improve the throughput of the benchmark.

We measure the performance of each solver primarily using a wall time timer which sums the real time taken to solve all NLPs, and ignore any processing time in between, both as it is not relevant to solver performance, and as it is generally equivalent between different solution methods as it mostly depends on the size of the problem. The benchmark results are given in the Dolan–Moré performance profiles [19].

5.2 Validating our Implementations on MacMPEC

Before we test the MPCC methods on the NOSBENCH collection, we verify the correctness of our the implementation of the regarded MPCC method-NLP solver combinations on the MacMPEC problem set. The MacMPEC problem set is available in the form of a tarball of AMPL [32] format .mod and .dat files from https://wiki.mcs.anl.gov/leyffer/index.php/MacMPEC. We use a modified version of CasADi [2] and manage to successfully extract 95 out 106 MPCC problems from this benchmark set. This is done by first generating .nl files for each problem, then reading these in and generating CasADi MPCCs of the form in (11). We drop any problems that contain complementarities with a "body" parameter of 3, as described in [35], which are slightly more generic than our implementation permits. This test suite is then run on four different approaches: the direct method (Section 3.2.1), standard Scholtes



(a) Direct NLP solve

(b) Standard homotopy

(c) ℓ_1-mode relaxation

(d) ℓ_∞-mode relaxation

Fig. 6 Evaluating MPCC methods with different NLP solvers on the MacMPEC test set. Each solver-method variant is compared to the other, but the results are split into four plots for better readability

relaxation (Section 3.2.2), ℓ_1-penalty (Section 3.2.7), and the ℓ_∞-mode Scholtes relaxation (Section 3.3), using the three NLP solvers: IPOPT [103], SNOPT [37], and WORHP [12]. These were the overall most successful variants also in the later experiments, and for the sake of brevity we do not run the benchmark on all possible solver-method combinations.

We note that several of these approaches solve each of the 95 problems in under ten minutes, and most solve more than 90% of the problem set in the same amount of time. These results are summarized in Fig. 6. They align with the results reported in other papers [27, 52, 101], which validates the correctness of our implementation.

In contrast to what we will see in the subsequent sections, we observe better performance of the direct method on the smaller problems in this test set, along with much better performance from SNOPT, again tied to the smaller size of the problems. In general this supports our assertions on the relative difficulty of the NOSBENCH test suite when compared to existing state-of-the-art benchmarks.

5.3 Evaluating Different MPCC Methods

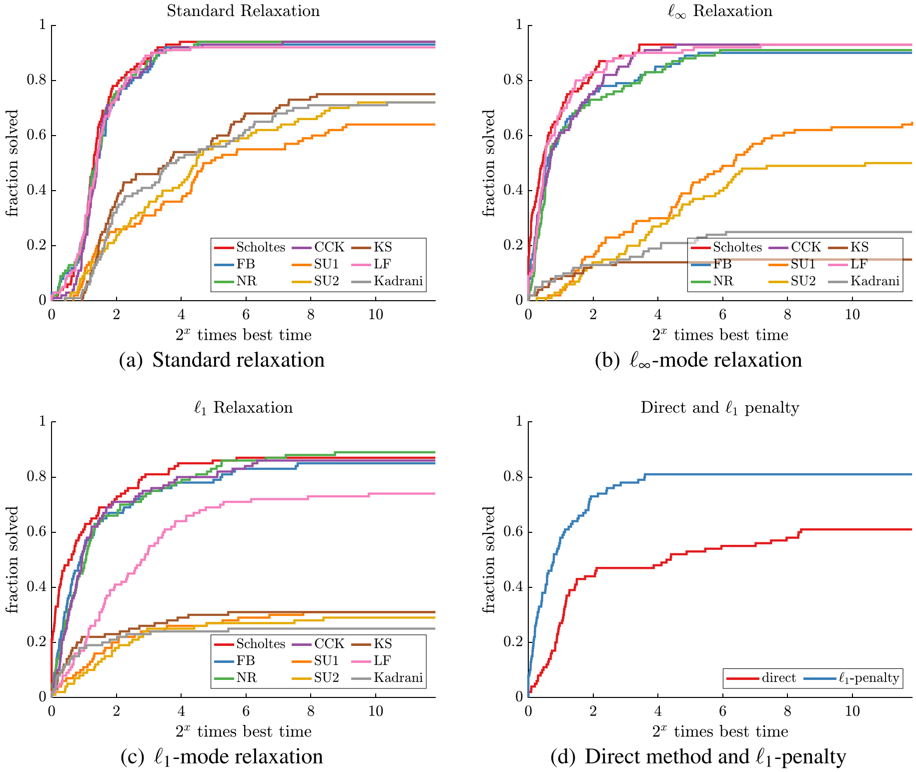
In this section, we compare nine variants of the relaxation-based methods discussed in Section 3.2: Scholtes' relaxation, three smoothed NCP functions (Fischer–Burmeister (FB), Natural-Residual (NR) and Chen–Chen–Kanzow (CCK), cf. Section 3.2.2), the Lin–Fukushima (LF) method. We also test the Steffensen–Ulbrich method with the two test functions mentioned in Section 3.2.4 (denoted by SU1 and SU2), and the kinked relaxations by Kadrani et al. and Kanzow–Schwartz (KS). Each of the nine methods is tested along with one of the three methods for steering the relaxation parameter summarized in Table 1, which results in a total of 27 different methods. Furthermore, we solve the MPCCs directly as NLPs (cf. Section 3.2.1) and with an exact-ℓ_1-penalty method without slacks (cf. (9)). An implementation of all aforementioned methods is available in NOSNOC.

As we discussed in Section 3.3, the algorithms implemented in NOSNOC for solving MPCCs have several free parameters, namely σ_0, κ, and η. In these experiments we use (10) for the σ_k updated with $\sigma_0 = 1$, $\kappa = 0.1$. In the subsequent sections, we vary these parameters in order to evaluate a good set of default parameters. The comparison is run on the NOSNOC-S subset.

An overview of the relative performance of each MPCC solution method with respect to all others can be seen in Fig. 7. The comparison is split into four subplots for better readability. Figure 8 summarizes the reasons for the failure of each of the 29 methods we test.

The general outcome of this benchmark is a victory for the Scholtes relaxation and the smoothed NCP functions (which lead to the same feasible set as Scholtes' method) in all of its three steering strategies. From the performance plots, we can clearly see that for all three methods of steering the relaxation parameter, the Scholtes relaxation successfully solves almost as many or more problems than the other relaxation methods. On the other hand, the more sophisticated local and nonsmooth relaxation methods perform worse in all cases. This complies with the fact that these methods have weaker theoretical properties [58] if the subproblems are solved inexactly (which is inevitable in finite precision arithmetic).

We can in detail compare the relaxations using the standard approach to drive the regularization parameter to zero. The results are depicted in Fig. 7 (a). Here we see a performance lead for the Scholtes relaxation, albeit a very slim one. Methods arising from the standard parameter steering can be split into three different groups based on performance (in order): the group containing the Scholtes relaxation as well as those that use NCP functions, Lin–Fukushima and Kanzow–Schwartz which are nearly as fast but plateau earlier (i.e., solve



(a) Standard relaxation

(b) ℓ_∞-mode relaxation

(c) ℓ_1-mode relaxation

(d) Direct method and ℓ_1-penalty

Fig. 7 Evaluating regularization methods for MPCCs on the `NOSBENCH-S` subset

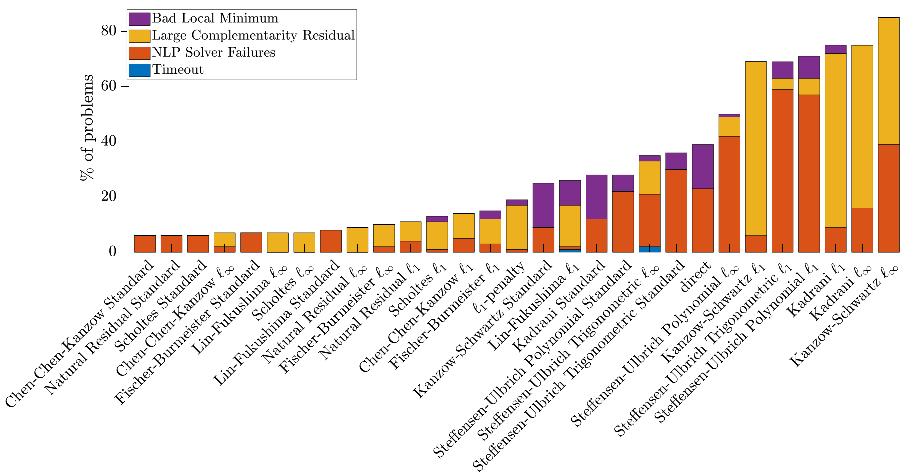


Fig. 8 Failure reasons for different regularization methods on the `NOSBENCH-S` subset

fewer problems), and the Steffensen–Ulbrich and Kadrani relaxations which gain only about 10% robustness over the direct method but are somewhat slower. We note that Fig. 7 suggests that we see very few outright victories for the standard Scholtes method, however, it is the most robust, achieving the highest fraction of problems solved, 95%.

We then move on to our analysis of the ℓ_∞-mode relaxation with the results given in Fig. 7 (b). One can see that the Scholtes method paired with this parameter steering strategy wins, with the largest fraction of successful solves. It maintains its lead but only reaches a solution on about 92% of the problems. Once again we see that the NCP function relaxations approach the performance of the Scholtes relaxation and match it in robustness. We also see the Lin–Fukushima relaxation perform well again but not quite at the level of the Scholtes-type group. On the other hand, we see extremely poor performance from the Steffensen–Ulbrich relaxations and the kinky relaxations of Kadrani et al. and Kanzow–Schwartz. From Fig. 8 we see that both the Kadrani and Kanzow–Schwartz relaxations fail frequently with a point at which the NLP solver claims it is optimal, but still has a large complementarity residual.

Finally, we analyze the performance of the ℓ_1-mode relaxations and the ℓ_1-penalty formulation, given in Fig. 7(c) and (d), respectively. The ℓ_1-penalty method nearly ties the ℓ_∞ Scholtes relaxation in absolute wins, however, it fails to break the 90% of problems solved. This performance is nearly mirrored by the ℓ_1-mode Scholtes relaxation albeit without the outright victories of the penalty method. Here we also see the second major gap between the Scholtes relaxation and the NCP function-based relaxations, though they remain the best-performing group. Surprisingly, the remaining relaxations perform worse than even the direct NLP solve approach.

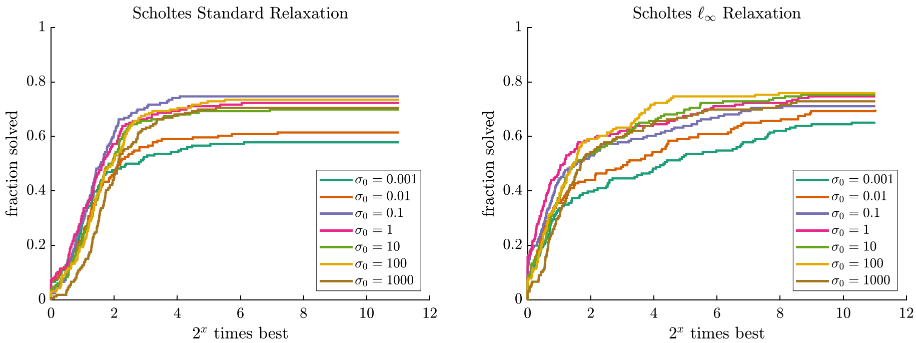
Another notable result is that while for about half the problems in this set the direct NLP approach solves the problem in an acceptable time frame, it quickly stalls at that point, cf. Fig. 7(d). Its failures, as seen in Fig. 8, are fairly evenly split between converging to unacceptable local minima, and failing to converge at all, primarily due to either step calculation failures at unacceptable points, or due to claiming infeasibility. This already points to the usefulness of the homotopy methods as it is clear to see that those methods perform better on a large proportion of problems with little to no impact on absolute performance.

We conclude that the best methods in terms of speed and robustness are the ℓ_∞-mode Scholtes relaxation and the Scholtes relaxation with standard homotopy parameter steering, respectively. The latter is certainly the choice for robustness as we will show in future experiments. The robustness can even be improved by adjusting the σ_0 and κ parameters which govern the trajectory of the regularization parameter.

5.4 The Role of the Homotopy Parameters

As described in the previous section, we note that the Scholtes relaxation is the optimal choice in its standard and ℓ_∞ forms. We do not examine further the ℓ_1 strategy, as it is not competitive with the other two. In order to elucidate the role of homotopy parameter update strategies, we run an experiment on `NOSBENCH-RL` (representative large subset), first varying the initial regularization parameter σ_0 and then varying κ, the rate at which we drive the regularization parameter to zero. It turns out that these two parameters have a moderate influence on the stability and speed of the homotopy solver converging to an acceptable solution.

We first discuss the effect of the initial value of the homotopy parameter σ_0 for which the performance plots can be seen in Fig. 9. Once again, we compare each method with the other, but split it into two subplots for readability. The first major takeaway from the analysis of the performance plots is that this has a much less significant effect on the ℓ_∞-



(a) Standard relaxation (b) ℓ_∞-mode relaxation

Fig. 9 Evaluating the influence of σ_0, the initial homotopy parameter in the NOSBENCH-RL test set

mode relaxation. Intuitively this does make sense as in this mode we simply use a penalty factor $\frac{1}{\sigma_k}$ in order to drive the complementarity residual to zero. Thus, σ_k is not a limiting factor on the complementarity residual in each step, depending of course on the scaling of the problem at hand. However, minimal effect is not no effect and we still see a worsened convergence if we choose a σ_0 that is too small. In contrast, it is observed that for the standard relaxation, for very small σ_0 the solvers converge to points with a complementarity residual exactly equal to σ. It is notable that this reduction in performance primarily comes from convergence to significantly worse local minimizers as seen in Fig. 10.

On the other hand, we see much earlier and much more pronounced decay in performance for the standard Scholtes regularization. We see almost a 30% reduction in the number of problems solved if σ_0 is chosen to small. Moreover, from Fig. 10 we see that the primary reason for failure is the NLP solver converging to bad local minimizers, compared to the more successful cases, e.g. with σ_0 equal to 10 or 0.1. One possible reason is that, for larger

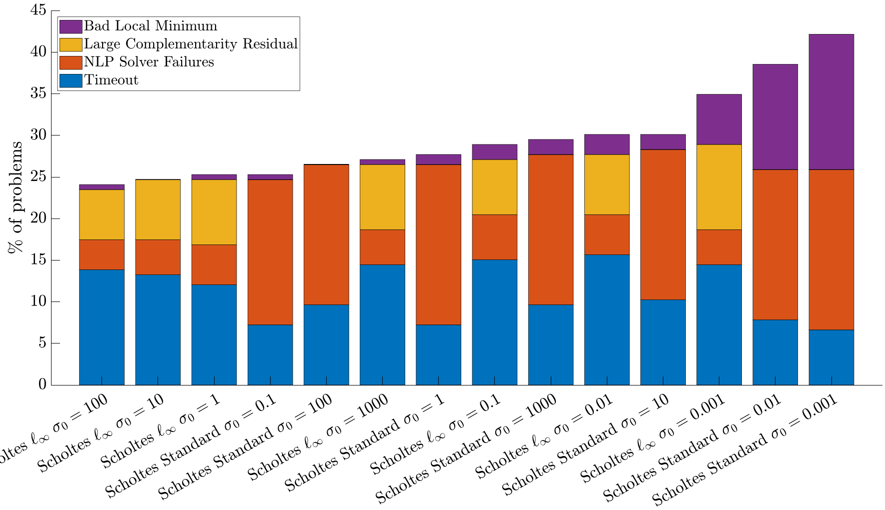
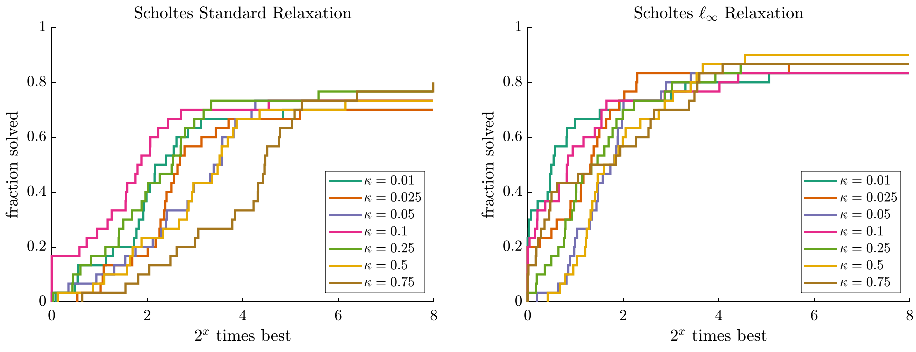


Fig. 10 Failure reasons for different values of σ_0



(a) Standard relaxation (b) ℓ_∞-mode relaxation

Fig. 11 Evaluating the influence of the homotopy slope parameter κ in the NOSBENCH-RL test set

values of σ_0 the problems are more relaxed and the solver may get easier attracted by better minimizers. In conclusion, for σ_0 in the rage of 0.1 to 10 the best performance is achieved.

Next, we examine the effect of the homotopy parameter update factor κ. In the literature, several different choices are used. Examples are $\kappa = 0.2$ in [56], $\kappa = 0.01$ in [93] and [47], and $\kappa = 0.1$ in [96]. In this experiment, we fix $\sigma_0 = 1$. In Fig. 11 one can see that the effect of κ on the overall success is surprisingly small compared to the initial regularization parameter σ_0. In particular, we see essentially no difference in performance for the ℓ_∞-mode regularization. This is very likely due to the fact that for a majority of problems, we see only several (and occasionally only one) homotopy iterations before the solver converges to an acceptable complementarity residual. The standard relaxation clearly shows both the weakness and the strength of a relatively slow homotopy. We note that for smaller κ the problem converges quicker due to having to take less homotopy iterations to converge to a sufficiently accurate solution. We also see that this benefit disappears as we get to more difficult problems. On the other hand, with larger κ, we have to solve more problems, but they can be solved quicker since the initial guess of the previous solution is much better. We also see a mild improvement (around 10%) in the number of problems that are solved with the larger κ. This makes it likely that a larger homotopy update rate is particularly useful for ensuring convergence for more difficult problems while a smaller κ (or use of the ℓ_∞-mode) is the superior option for simpler problems. To conclude, a good default choice is $\kappa = 0.1$ and $\sigma_0 = 1$ with the ℓ_∞ mode.

5.5 Evaluating Different NLP Solvers

So far we used in our experiments only IPOPT as NLP solver in the homotopy loop. Next, we compare this solver to SNOPT [37], and WORHP [12] on the NOSBENCH-F set.

All three solvers are tested on the Scholtes relaxation with the standard and ℓ_∞ relaxation parameter steering strategies. In this case, we use the existing default homotopy parameters $\sigma_0 = 1$ and $\kappa = 0.1$ and use the standard default settings for both SNOPT and WORHP, except for those related to maximum iterations and timeouts which are set as for IPOPT. The solvers are tested on the full NOSBENCH-F test suite, which contains 603 MPCCs. Figure 12 shows the performance profiles for the three solvers for the two different parameter steering strategies.

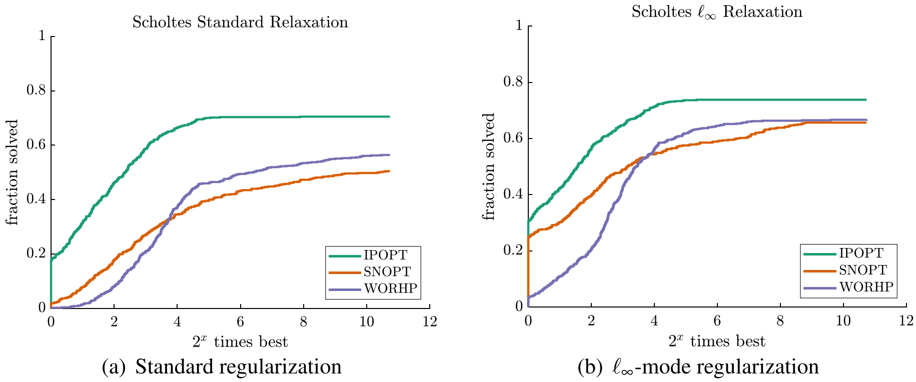


Fig. 12 Evaluating different NLP solvers on the NOSBENCH-F test set

On some easier and smaller problems SNOPT is the fastest solver, which is expected due to the specialization of active set methods on small to medium size problems. WORHP on the other hand is is not particularly fast but is more robust than SNOPT, and this gap is more prevalent in the standard relaxation. It is however clear to see that IPOPT is by far the winner here with both the most overall wins in the ℓ_∞-mode where it successfully solves 73.8% of the problem set. This may be caused by slightly optimized solver settings we have used for IPOPT. Moreover, the homotopy parameters $\kappa = 0.1$, $\sigma_0 = 1$ are also somewhat tuned for IPOPT. Other values might be beneficial to the performance of the other solvers.

In Fig. 13 we also report the reasons for the failure of the NLP solver on the test set. Some optimal control problems have very nonlinear dynamics and combined with the relaxed complementarity constraints one obtains difficult NLP subproblems. Moreover, in the ℓ_∞ approach, the slack variable and consequently the complementarity residual cannot be brought to a sufficiently small value, despite a very large penalty parameter in the objective. We note

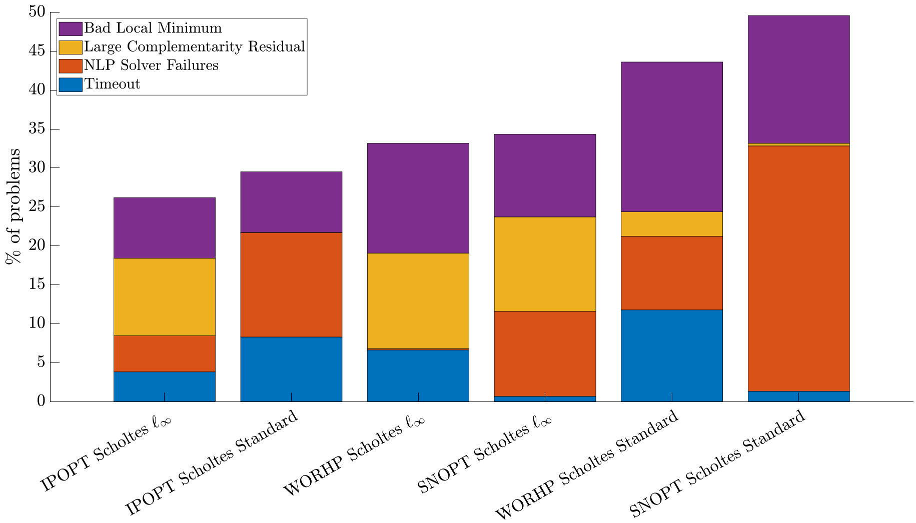
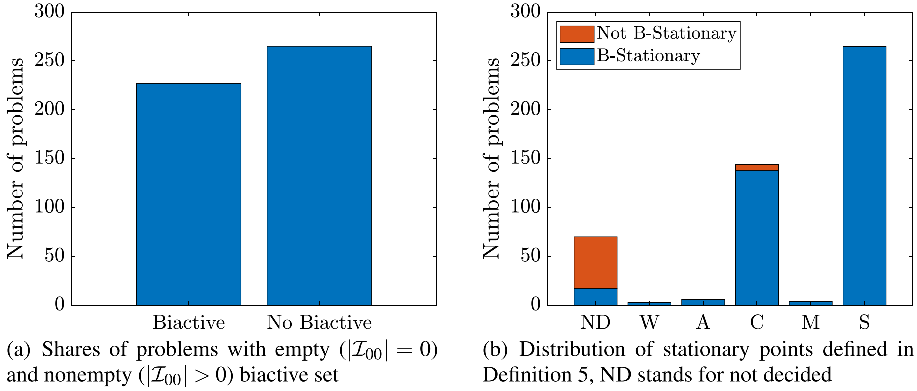


Fig. 13 Failure reasons for different NLP solvers



(a) Shares of problems with empty ($|\mathcal{I}_{00}| = 0$) and nonempty ($|\mathcal{I}_{00}| > 0$) biactive set

(b) Distribution of stationary points defined in Definition 5, ND stands for not decided

Fig. 14 Evaluating type of stationarity points in the NOSBENCH-F test set

also that some of the problems in NOSBENCH-F are very large, and they might get solved if the solvers were allowed more time. Interestingly, in more than 5% of cases IPOPT fails because the other approaches have found significantly better local minima. This is usually the case for MPCCs coming from OCPs with nonsmooth systems with state jumps.

Type of stationary points We report also the statistics of the type of stationarity points for the most successful solver-method combination, namely IPOPT with the Scholtes relaxation and ℓ_∞-parameter steering. The set $\mathcal{I}_{00}$ is often called the biactive complementarity constraints Fig. 14 (a) shows the number of problems with an empty ($|\mathcal{I}_{00}| = 0$) and nonempty set ($|\mathcal{I}_{00}| > 0$) of biactive complementarity constraints. Recall that problems with a biactive set are more regular. The biactive set is calculated via checking the values of $G(w)$ and $H(w)$. We first calculate the biactive set $\mathcal{I}_{00}$ as all i such that $G_i(w) < \sqrt{\texttt{comp\_tol}}$ and $H_i(w) < \sqrt{\texttt{comp\_tol}}$. We then calculate $\mathcal{I}_{0+}$ as all i not in $\mathcal{I}_{00}$ that satisfies $G_i(w) < H_i(w)$ and $\mathcal{I}_{+0}$ as all i not in $\mathcal{I}_{00}$ that satisfies $G_i(w) > H_i(w)$. This however sometimes fails to correctly identify the active set which leads the TNLP to fail to converge. In these cases we iteratively remove elements from $\mathcal{I}_{00}$, with the pairs $(G_i(w), H_i(w))$ that are furthest from the origin being removed first and added to the corresponding set based on the magnitude of the components. We set the maximum number of iterations in this procedure to $|\mathcal{I}_{00}|$. In the cases where the TNLP is infeasible or leads to a significantly different objective than obtained by the homotopy approach indicates that we did not identify the active-set correctly. We denoted these cases as Not Decided (ND).

Recall that if the biactive set is empty, then the solution is automatically S-stationary. For those with a nonempty biactive set, we solve the corresponding TNLP (cf. Definition 2) and asses the type of the stationary point according to Definition 5. The results are summarized in Fig. 14 (b). It turns out none of points with a nonempty $\mathcal{I}_{00}$ is S-stationary. Next, we check additionally for all these points if they are B-stationary, i.e., we check if they permit first-order descent directions. This can be done by solving the nonconvex LPCC in (4). We reformulate the LPCC into an equivalent mixed-integer linear program (MILP) [22] and solve it with Gurobi [41]. Note that for the verification we need $d = 0$ to solve (4). However, this point can be either a local or global solution, but the MILP approach always finds a global minimizer. To address this, we add a trust region constraint $\|d\|_\infty \leq 10^{-2}$ to make sure that we isolate the (possibly) local optimum $d = 0$. If this constraint becomes active, we shrink the trust region radius down to 10^{-4}, and if $d = 0$ is still not optimal, we conclude that it is not a local optimum and that the point w^* is not B-stationary. Moreover, as sanity check we confirm with the LPCC approach that all S-stationary points are indeed B-stationary. It turns out that in almost all cases, where we could identify the stationary point the points are B-stationary, except in a few cases of C-stationary points. This provides evidence that MPCCs obtained from nonsmooth optimal control problems, despite sometimes being difficult to solve, are not very degenerate. On the other hand, for the ND problems where we could not identify the active set in the TNLP, first-order descent directions often exist. One reasons could be that the homotopy loop terminated too early. In some cases, we noticed that lowering the complementarity tolerance would either make the solver fail, or help to identify the more accurate solution as B-stationary.

6 Conclusion and Outlook

The goal of this paper was to create a benchmark collection of Mathematical Programs with Complementarity Constraints (MPCCs) obtained from the time-discretization of nonsmooth simulation and Optimal Control Problems (OCPs) and to use it for the evaluation of tailored MPCC solution methods. This provided a large source of practical MPCCs. However, MPCCs violate standard constraint qualifications and thus require specialized first-order optimality conditions and solution methods, which were reviewed in detail. The literature reports very good numerical performance of standard MPCC methods on existing benchmarks. Unfortunately, we have not observed such robust performance on MPCCs obtained from nonsmooth OCPs.

To better assess the limitations of the current state-of-the-art and to motivate further development of MPCC methods, we introduce a new benchmark set, which we call NOSBENCH. The novel benchmark consists of a total of 603 problems. Moreover, we derive several subsets from it to facilitate the analysis of a variety of solution methods. All the methods we test solve a sequence of regularized nonlinear programs (NLPs) in a homotopy approach. We compare different regularization strategies, different NLP solvers, different approaches to controlling the degree of relaxation in the subproblems, and the influence of the homotopy meta-parameters. We find that the oldest and simplest methods, namely Scholtes' global relaxation [93] and smoothed nonlinear complementarity functions [21] (which are often equivalent to Scholtes' approach), perform best. This is consistent with previous extensive experiments such as those performed by [47]. Surprisingly, our implementations of the more sophisticated regularization strategies show quite disappointing results even on easier subsets of our test set. Two positive results are that solutions of nonsmooth OCPs are quite often S-stationary and that the weaker stationary points rarely allow first-order descent directions. However, in the best case, we only manage to solve 73.8% of the problems on the full problem set, which is not yet satisfactory.

It would be interesting to test some of the active-set MPCC methods in the future, should robust open-source implementations of them become available. We aim to extend the NOSBENCH test suite by further challenging nonsmooth simulation and optimal control problems.

In conclusion, relaxation-based MPCC methods, coupled with a robust NLP solver, can perform reasonably well even on large and highly nonlinear problems. The experiments performed in this paper helped us to extract some rules of thumb for the default solver settings in NOSNOC. However, there is still room for improvement, and the benchmark collection introduced in this paper can help to test novel methods.

Acknowledgements This research was supported by DFG via Research Unit FOR 2401, projects 424107692 and 525018088, by BMWK via 03EI4057A and 03EN3054B, and by the EU via ELO-X 953348.

Funding Open Access funding enabled and organized by Projekt DEAL. The first and second authors have contributed equally.

References

1. Achtziger, W., Kanzow, C.: Mathematical programs with vanishing constraints: optimality conditions and constraint qualifications. Math. Program. Ser. A **114**, 69–99 (2008)
2. Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi: a software framework for nonlinear optimization and optimal control. Math. Program. Comput. **11**, 1–36 (2019)
3. Anitescu, M.: On solving mathematical programs with complementarity constraints as nonlinear programs. Preprint ANL/MCS-P864-1200, Argonne National Laboratory, Argonne, IL **3** (2000)
4. Anitescu, M.: Global convergence of an elastic mode approach for a class of mathematical programs with complementarity constraints. SIAM J. Optim. **16**, 120–145 (2005)
5. Anitescu, M., Tseng, P., Wright, S.J.: Elastic-mode algorithms for mathematical programs with equilibrium constraints: global convergence and stationarity properties. Math. Program. **110**, 337–371 (2007)
6. Bard, J.F., Moore, J.T.: A branch and bound algorithm for the bilevel programming problem. SIAM J. Sci. Stat. Comput. **11**, 281–292 (1990)
7. Baumrucker, B.T., Renfro, J.G., Biegler, L.T.: MPEC problem formulations and solution strategies with chemical engineering applications. Comput. Chem. Eng. **32**, 2903–2913 (2008)
8. Baumrucker, B.T., Biegler, L.T.: MPEC strategies for optimization of a class of hybrid dynamic systems. J. Process Control **19**, 1248–1256 (2009)
9. Benson, H.Y., Sen, A., Shanno, D.F., Vanderbei, R.J.: Interior-point algorithms, penalty methods and equilibrium problems. Comput. Optim. Appl. **34**, 155–182 (2006)
10. Brogliato, B., Tanwani, A.: Dynamical systems coupled with monotone set-valued operators: Formalisms, applications, well-posedness, and stability. SIAM Rev. **62**, 3–129 (2020)
11. Brüls, O., Acary, V., Cardona, A.: Simultaneous enforcement of constraints at position and velocity levels in the nonsmooth generalized-α scheme. Comput. Methods Appl. Mech. Eng. **281**, 131–161 (2014)
12. Büskens, C., Wassel, D.: The ESA NLP solver WORHP. In: Fasano, G., Pintér, J.D. (eds.) Modeling and Optimization in Space Engineering. Springer Optimization and Its Applications, vol. 73, pp. 85–110. Springer, New York (2013)
13. Byrd, R.H., Lopez-Calva, G., Nocedal, J.: A line search exact penalty method using steering rules. Math. Program. **133**, 39–73 (2012)
14. Calvo, M., Montijano, J.I., Rández, L.: Algorithm 968: DISODE45: a Matlab Runge-Kutta solver for piecewise smooth IVPs of Filippov type. ACM Trans. Math. Softw. (TOMS) **43**, 25 (2016)
15. Carius, J., Farshidian, F., Hutter, M.: MPC-Net: a first principles guided policy search. IEEE Robot. Autom. Lett. **5**, 2897–2904 (2020)
16. Chen, B., Chen, X., Kanzow, C.: A penalized Fischer-Burmeister NCP-function. Math. Program. **88**, 211–216 (2000)
17. Christiansen, B., Maurer, H., Zirn, O.: Optimal control of a voice-coil-motor with Coulombic friction. In: 2008 47th IEEE conference on decision and control, pp. 1557–1562. IEEE (2008)
18. Dirkse, S.: MPECLib: a collection of mathematical programs with equilibrium constraints (2004)
19. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
20. Duff, I.S., Reid, J.K.: MA27 – a set of fortran subroutines for solving sparse symmetric sets of linear equations (1982)

21. Facchinei, F., Jiang, H., Qi, L.: A smoothing method for mathematical programs with equilibrium constraints. Math. Program. **85**, 107–134 (1999)
22. Facchinei, F., Pang, J.-S.: Finite-dimensional Variational Inequalities and Complementarity Problems, vols. 1–2. Springer, New York (2003)
23. Ferris, M.C., Tin-Loi, F.: On the solution of a minimum weight elastoplastic problem involving displacement and complementarity constraints. Comput. Methods Appl. Mech. Eng. **174**, 108–120 (1999)
24. Flegel, M.L., Kanzow, C.: Abadie-type constraint qualification for mathematical programs with equilibrium constraints. J. Optim. Theory Appl. **124**, 595–614 (2005)
25. Flegel, M.L., Kanzow, C.: On the Guignard constraint qualification for mathematical programs with equilibrium constraints. Optimization **54**, 517–537 (2005)
26. Flegel, M.L., Kanzow, C.: A direct proof for M-stationarity under MPEC-GCQ for mathematical programs with equilibrium constraints. In: Dempe, S., Kalashnikov, V. (eds.) Optimization with Multivalued Mappings. Springer Optimization and Its Application, vol. 2, pp. 111–122. Springer, New York (2006)
27. Fletcher, R., Leyffer, S.: Numerical experience with solving MPECs as NLPs. In: Numerical Analysis Report NA/210, Department of Mathematics, University of Dundee, Dundee, UK (2002)
28. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Math. Program. **91**, 239–269 (2002)
29. Fletcher, R., Leyffer, S.: Solving mathematical programs with complementarity constraints as nonlinear programs. Optim. Methods Softw. **19**, 15–40 (2004)
30. Fletcher, R.: A new low rank quasi-Newton update scheme for nonlinear programming. Technical Report NA/223, University of Dundee (2005)
31. Fletcher, R., Leyffer, S., Ralph, D., Scholtes, S.: Local convergence of SQP methods for mathematical programs with equilibrium constraints. SIAM J. Optim. **17**, 259–286 (2006)
32. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: a Modeling Language for Mathematical Programming, 2nd edn. Thomson (2003)
33. Fukushima, M., Luo, Z.-Q., Pang, J.-S.: A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints. Comput. Optim. Appl. **10**, 5–34 (1998)
34. Fukushima, M., Lin, G.-H.: Smoothing methods for mathematical programs with equilibrium constraints. In: International Conference on Informatics Research for Development of Knowledge Society Infrastructure, 2004. ICKS 2004, IEEE, pp. 206–213 (2004)
35. Gay, D.M.: Writing .nl files. Technical report, Optimization and Uncertainty Estimation, Sandia National Laboratories (2005)
36. Giallombardo, G., Ralph, D.: Multiplier convergence in trust-region methods with application to convergence of decomposition methods for MPECs. Math. Program. **112**, 335–369 (2008)
37. Gill, P., Murray, W., Saunders, M.: SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev. **47**, 99–131 (2005)
38. Gill, P.E., Wong, E.: Methods for convex and general quadratic programming. Math. Program. Comput. **7**, 71–112 (2015)
39. Guignard, M.: Generalized Kuhn-Tucker conditions for mathematical programming problems in a Banach space. SIAM J. Control **7**, 232–241 (1969)
40. Guo, L., Deng, Z.: A new augmented Lagrangian method for MPCCs–theoretical and numerical comparison with existing augmented Lagrangian methods. Math. Oper. Res. **47**, 1229–1246 (2022)
41. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2022). https://www.gurobi.com
42. Halm, M., Posa, M.: Set-valued rigid body dynamics for simultaneous frictional impact. arXiv:2103.15714 (2021)
43. Hall, J., Nurkanović, A., Messerer, F., Diehl, M.: A sequential convex programming approach to solving quadratic programs and optimal control problems with linear complementarity constraints. IEEE Control Syst. Lett. **6**, 536–541 (2022)
44. Hall, J., Nurkanović, A., Messerer, F., Diehl, M.: LCQPow–A solver for linear complementarity quadratic programs. Math. Program. Comput. (2023). Accepted for publication
45. Hatz, K., Leyffer, S., Schlöder, J.P., Bock, H.G.: Regularizing bilevel nonlinear programs by lifting. Preprint ANL/MCS-P4076-0613. Argonne National Laboratory, USA (2013)
46. Heemels, W.P.M.H., Schumacher, J.M., Weiland, S.: Projected dynamical systems in a complementarity formalism. Oper. Res. Lett. **27**, 83–91 (2000)
47. Hoheisel, T., Kanzow, C., Schwartz, A.: Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints. Math. Program. **137**, 257–288 (2013)
48. Howell, T.A., Le Cleac'h, S., Singh, S., Florence, P., Manchester, Z., Sindhwani, V.: Trajectory optimization with optimization-based dynamics. IEEE Robot. Autom. Lett. **7**, 6750–6757 (2022)

49. HSL: A collection of Fortran codes for large scale scientific computation (2011). http://www.hsl.rl.ac.uk
50. Hu, X.M., Ralph, D.: Convergence of a penalty method for mathematical programming with complementarity constraints. J. Optim. Theory Appl. **123**, 365–390 (2004)
51. Izmailov, A.F., Pogosyan, A., Solodov, M.V.: Semismooth Newton method for the lifted reformulation of mathematical programs with complementarity constraints. Comput. Optim. Appl. **51**, 199–221 (2012)
52. Izmailov, A.F., Solodov, M.V., Uskov, E.I.: Global convergence of augmented Lagrangian methods applied to optimization problems with degenerate constraints, including problems with complementarity constraints. SIAM J. Optim. **22**, 1579–1606 (2012)
53. Izmailov, A.F., Solodov, M.V.: Newton-Type Methods for Optimization and Variational Problems, 1st edn. Springer Series in Operations Research and Financial Engineering. Springer, Cham (2014)
54. Johansson, K.H., Barabanov, A.E., Aström, K.J.: Limit cycles with chattering in relay feedback systems. IEEE Trans. Autom. Control **47**, 1414–1423 (2002)
55. Jung, M.N., Kirches, C., Sager, S.: On perspective functions and vanishing constraints in mixed-integer nonlinear optimal control. In: Jünger, M., Reinelt, G. (eds.) Facets of Combinatorial Optimization - Festschrift for Martin Grötschel, pp. 387–417. Springer, Berlin Heidelberg (2013)
56. Kadrani, A., Dussault, J.-P., Benchakroun, A.: A new regularization scheme for mathematical programs with complementarity constraints. SIAM J. Optim. **20**, 78–103 (2009)
57. Kanzow, C., Schwartz, A.: A new regularization method for mathematical programs with complementarity constraints with strong convergence properties. SIAM J. Optim. **23**, 770–798 (2013)
58. Kanzow, C., Schwartz, A.: The price of inexactness: convergence properties of relaxation methods for mathematical programs with complementarity constraints revisited. Math. Oper. Res. **40**, 253–275 (2015)
59. Kim, Y., Leyffer, S., Munson, T.: MPEC methods for bilevel optimization problems. In: Kim, Y., Leyffer, S., Munson, T. (eds.) Bilevel Optimization. Springer Optimization and Its Applications, vol. 161, pp. 335–360. Springer, Cham (2020)
60. Kirches, C.: Fast numerical methods for mixed-integer nonlinear model-predictive control. PhD thesis, University of Heidelberg (2010)
61. Kirches, C., Larson, J., Leyffer, S., Manns, P.: Sequential linearization method for bound-constrained mathematical programs with complementarity constraints. SIAM J. Optim. **32**, 75–99 (2022)
62. Kong, N.J., Council, G., Johnson, A.M.: iLQR for piecewise-smooth hybrid dynamical systems. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 5374–5381. IEEE (2021)
63. Lenders, F.J.M.: Numerical methods for mixed-integer optimal control with combinatorial constraints. PhD thesis, Heidelberg University (2018)
64. Leyffer, S.: MacMPEC: Ampl collection of mpecs". Argonne National Laboratory (2000)
65. Leyffer, S.: Complementarity constraints as nonlinear equations: Theory and numerical experience. In: Dempe, S., Kalashnikov, V. Optimization with Multivalued Mappings. Springer Optimization and Its Applications, vol. 2, pp. 169–208. Springer, New York (2006)
66. Leyffer, S., López-Calva, G., Nocedal, J.: Interior methods for mathematical programs with complementarity constraints. SIAM J. Optim. **17**, 52–77 (2006)
67. Leyffer, S., Munson, T.S.: A globally convergent filter method for MPECs. Preprint ANL/MCS-P1457-0907, Argonne National Laboratory, Mathematics and Computer Science Division (2007)
68. Lin, G.-H., Fukushima, M.: A modified relaxation scheme for mathematical programs with complementarity constraints. Ann. Oper. Res. **133**, 63–84 (2005)
69. Lin, K., Ohtsuka, T.: A non-interior-point continuation method for the optimal control problem with equilibrium constraints. arXiv:2210.10336 (2022)
70. Liu, X., Sun, J.: Generalized stationary points and an interior-point method for mathematical programs with equilibrium constraints. Math. Program. **101**, 231–261 (2004)
71. Luo, Z.-Q., Pang, J.-S., Ralph, D.: Mathematical Programs with Equilibrium Constraints. Cambridge University Press, Cambridge (1996)
72. Moreau, J.J.: Evolution problem associated with a moving convex set in a Hilbert space. J. Differ. Equ. **26**, 347–374 (1977)
73. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)
74. NOSNOC: https://github.com/nurkanovic/nosnoc (2022)
75. Nurkanović, A., Albrecht, S., Diehl, M.: Limits of MPCC formulations in direct optimal control with nonsmooth differential equations. In: 2020 European Control Conference (ECC), pp. 2015–2020. IEEE (2020)
76. Nurkanović, A., Sartor, T., Albrecht, S., Diehl, M.: A time-freezing approach for numerical optimal control of nonsmooth differential equations with state jumps. IEEE Control Syst. Lett. **5**, 439–444 (2021)

77. Nurkanović, A., Diehl, M.: Continuous optimization for control of hybrid systems with hysteresis via time-freezing. IEEE Control Syst. Lett. **6**, 3182–3187 (2022)
78. Nurkanović, A., Diehl, M.: NOSNOC: A software package for numerical optimal control of nonsmooth systems. IEEE Control Syst. Lett. **6**, 3110–3115 (2022)
79. Nurkanović, A., Frey, J., Pozharskiy, A., Diehl, M.: Finite elements with switch detection for direct optimal control of nonsmooth systems with set-valued step functions. In: 2023 62nd IEEE Conference on Decision and Control. IEEE (2023)
80. Nurkanović, A., Pozharskiy, A., Frey, J., Diehl, M.: Finite elements with switch detection for numerical optimal control of nonsmooth dynamical systems with set-valued step functions. arXiv:2307.03482 (2023)
81. Nurkanović, A., Albrecht, S., Brogliato, B., Diehl, M.: The time-freezing reformulation for numerical optimal control of complementarity Lagrangian systems with state jumps. Automatica **158**, 111295 (2023)
82. Nurkanović, A.: Numerical methods for optimal control of nonsmooth dynamical systems. PhD thesis, University of Freiburg (2023)
83. Nurkanović, A., Sperl, M., Albrecht, S., Diehl, M.: Finite elements with switch detection for direct optimal control of nonsmooth systems. Numer. Math. **156**, 1115–1162 (2024)
84. Nurkanović, A., Frey, J., Pozharskiy, A., Diehl, M.: FESD-J: Finite elements with switch detection for numerical optimal control of rigid bodies with impacts and Coulomb friction. Nonlinear Anal. Hybrid Syst. **52**, 101460 (2024)
85. Outrata, J.V.: Optimality conditions for a class of mathematical programs with equilibrium constraints. Math. Oper. Res. **24**, 627–644 (1999)
86. Pang, J.-S., Fukushima, M.: Complementarity constraint qualifications and simplified B-stationarity conditions for mathematical programs with equilibrium constraints. Comput. Optim. Appl. **13**, 111–136 (1999)
87. Piiroinen, P.T., Kuznetsov, Y.A.: An event-driven method to simulate Filippov systems with accurate computing of sliding motions. ACM Trans. Math. Softw. (TOMS) **34**, 13 (2008)
88. Pozharskiy, A.: Evaluating methods for solving mathematical programs with complementarity constraints arising from nonsmooth optimal control. Master's thesis, Albert-Ludwigs-University Freiburg (2023)
89. Raghunathan, A.U., Biegler, L.T.: An interior point method for mathematical programs with complementarity constraints (MPCCs). SIAM J. Optim. **15**, 720–750 (2005)
90. Ralph, D., Wright, S.J.: Some properties of regularization and penalization schemes for MPECs. Optim. Methods Softw. **19**, 527–556 (2004)
91. Rawlings, J.B., Mayne, D.Q., Diehl, M.M.: Model Predictive Control: Theory, Computation, and Design, 2nd edn. Nob Hill, Santa Barbara (2017)
92. Scheel, H., Scholtes, S.: Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. Math. Oper. Res. **25**, 1–22 (2000)
93. Scholtes, S.: Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. SIAM J. Optim. **11**, 918–936 (2001)
94. Scholtes, S.: Combinatorial structures in nonlinear programming. Technical report, University of Cambridge (2002)
95. Schwartz, A.: Mathematical programs with complementarity constraints: Theory, methods and applications. PhD thesis, Universität Würzburg (2011)
96. Steffensen, S., Ulbrich, M.: A new relaxation scheme for mathematical programs with equilibrium constraints. SIAM J. Optim. **20**, 2504–2539 (2010)
97. Stein, O.: Lifting mathematical programs with complementarity constraints. Math. Program. **131**, 71–94 (2012)
98. Stewart, D.: A high accuracy method for solving ODEs with discontinuous right-hand side. Numer. Math. **58**, 299–328 (1990)
99. Stewart, D.E.: A numerical method for friction problems with multiple contacts. The ANZIAM Journal **37**, 288–308 (1996)
100. Stewart, D.E., Anitescu, M.: Optimal control of systems with discontinuous differential equations. Numer. Math. **114**, 653–695 (2010)
101. Thierry, D., Biegler, L.: The ℓ_1–exact penalty-barrier phase for degenerate nonlinear programming problems in ipopt. IFAC-PapersOnLine **53**, 6496–6501 (2020)
102. Veelken, S.: A new relaxation scheme for mathematical programs with equilibrium constraints: Theory and numerical experience. PhD thesis, Technische Universität München (2009)
103. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**, 25–57 (2006)

104. Ye, J.J.: Necessary and sufficient optimality conditions for mathematical programs with equilibrium constraints. J. Math. Anal. Appl. **307**, 350–369 (2005)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.