Computational Optimization and Applications (2023) 84:191–223
https://doi.org/10.1007/s10589-022-00386-8

An improved penalty algorithm using model order reduction for MIPDECO problems with partial observations

Dominik Garmatter[1] · Margherita Porcelli[2,3] · Francesco Rinaldi[4] · Martin Stoll[1]

Received: 7 October 2021 / Accepted: 7 June 2022 / Published online: 27 June 2022

Abstract

This work addresses optimal control problems governed by a linear time-dependent partial differential equation (PDE) as well as integer constraints on the control. Moreover, partial observations are assumed in the objective function. The resulting problem poses several numerical challenges due to the mixture of combinatorial aspects, induced by integer variables, and large scale linear algebra issues, arising from the PDE discretization. Since classical solution approaches such as the branch-and-bound framework are typically overwhelmed by such large-scale problems, this work extends an improved penalty algorithm proposed by the authors, to the time-dependent setting. The main contribution is a novel combination of an interior point method, preconditioning, and model order reduction yielding a tailored local optimization solver at the heart of the overall solution procedure. A thorough numerical investigation is carried out both for the heat equation as well as a convection-diffusion problem demonstrating the versatility of the approach.

Keywords Mixed integer optimization · PDE-constrained optimization · Exact penalty methods · Interior point methods · Model order reduction

✉ Margherita Porcelli
margherita.porcelli@unibo.it

Dominik Garmatter
dominik.garmatter@math.tu-chemnitz.de

Francesco Rinaldi
rinaldi@math.unipd.it

Martin Stoll
martin.stoll@math.tu-chemnitz.de

[1] Department of Mathematics, Chemnitz University of Technology, Chemnitz, Germany

[2] Department of Mathematics & AM2, University of Bologna, Bologna, Italy

[3] ISTI–CNR, Pisa, Italy

[4] Department of Mathematics "Tullio Levi-Civita", University of Padova, Padova, Italy

Mathematics Subject Classification 34C20 · 90C06 · 90C11 · 93C20 · 90C51

1 Introduction

Optimal control problems with PDE constraints and additional integer (and possible other constraints) are usually referred to as mixed integer PDE-constrained optimization (MIPDECO) problems. Such problems arise in a variety of real world applications such as gas networks [1, 2], the placement of tidal and wind turbines [3–5] or power networks [6]. Approximating their solution poses significant difficulties as being in the intersection of two fields: integer programming and PDEs. While integer optimization problems have an inherent combinatorial complexity that needs to be accounted for, PDE-constrained optimization problems have to deal with large-scale linear systems resulting from the discretization of the PDE, see, e.g., [7, 8].

A classical solution approach for a MIPDECO problem is to *first-discretize-then-optimize*: the PDE and the control are discretized, thus resulting in the continuous MIPDECO problem being approximated by a large-scale finite-dimensional mixed-integer nonlinear programming problem (MINLP). This approach was outlined in the previous work by the authors [9], where it was also shown numerically that standard techniques such as branch-and-bound, see, e.g., [10] for an overview, indeed struggle to solve the resulting MINLP in reasonable time (both the large amount of integer variables as well as the large PDE discretization are challenging here).

As a remedy, [9] introduced a novel *improved penalty algorithm* (IPA) that repeatedly solves an equivalent continuous penalty reformulation of the original problem for an increasing penalty parameter, with the penalty reformulation being obtained by relaxing the integer constraints and adding a suitable penalty term to the objective function to avoid non-integer solutions. The IPA was based on an exact penalty (EXP) algorithm [11] that provides a theoretical framework for when to increase the penalization and when to search for a better minimizer. The IPA deviated from the EXP algorithm by employing a probabilistic search approach to determine a new iterate. Such a search was closely connected to basin hopping or iterated local search methods, see, e.g., [12, 13]. The upside of this change was that the IPA only relied on a local optimization solver, where a suitable interior point method (IPM) utilizing a tailored preconditioner for the Newton system was used in [9]. As a result, the IPA was able to provide either the global or a high quality local minimum for a Poisson problem as well as a convection-diffusion model problem.

This article focuses on extending the IPA developed in [9] to MIPDECO problems with a linear, time-dependent PDE constraint as well as partial observations in the objective function. In this case, the resulting discretized MINLP will definitely be of large scale. To overcome the inherent complexity of this problem, we approximate the PDE constraint using *balanced truncation* model order reduction (MOR), see, e.g., [14], and then develop a suitable IPM for this *reduced penalty formulation*. The IPM is again well-equipped for the problem as it:

- explicitly handles the non-convexity introduced by the penalty term;

- incorporates a specific preconditioner to handle the linear algebra as well as the singularity due to the partial observation.

Embedding this IPM into the IPA framework then allows for the solution of large-scale MIPDECO problems and the resulting algorithm is numerically investigated, both for the heat equation as well as for a convection-diffusion problem.

While the use of MOR is standard in general optimization contexts, see, e.g., [15–18], MOR for MIPDECO problems is far less investigated, see [19] for a first result. Furthermore, applying preconditioning to a reduced system of equations has only been considered once [20], while [21] considers preconditioning during the generation of reduced models. To the knowledge of the authors, the combination of an IPM, MOR, and preconditioning has not been considered so far in the literature to handle MIPDECO problems.

Finally, other methods for MIPDECO problems such as Sum-up-Rounding strategies [22, 23], derivative-free approaches [24], and sophisticated rounding techniques [25], might become too costly when adapted to tackle the large-scale problems considered in this article.

The paper is organized as follows: the time-dependent model problem, its discretization, as well as the equivalent penalty formulation are presented in Sect. 2. Section 3 contains the MOR approach, including some theoretical aspects, and the interior point method. Section 4 reviews the IPA framework, adapts it to the time-dependent setting, and discusses different perturbation strategies for the probabilistic search approach. Section 5 contains the numerical investigation of the new algorithm and final conclusions are drawn in Sect. 6.

2 Problem formulation

We begin with the description of the optimal control model problem in function spaces. Following the first-discretize-then-optimize approach, we then present the discretized model problem, its continuous relaxation, and then move towards the penalty reformulation of the problem.

2.1 Time-dependent binary optimal control problem

We begin with the description of the PDE in order to formulate the optimal control problem. Consider a bounded domain $\Omega \subset \mathbb{R}^2$ with Lipschitz boundary, the time interval $[0, T]$ with final time $T > 0$, source functions $\phi_1, \dots, \phi_l \in L^2(\Omega)$, and based on these the parabolic PDE: for a given control function $u : (0, T) \to \mathbb{R}^l : t \mapsto (u^{(1)}(t), \dots, u^{(l)}(t))^\intercal$ find the state $y \in L^2(0, T, H_0^1(\Omega))$ solving

$$\begin{aligned} \frac{\partial}{\partial t} y(t,x) - \Delta y(t,x) = \sum_{i=1}^{l} u^{(i)}(t)\phi_i(x), \quad (t,x) \in (0,T) \times \Omega, \\ y(0,x) = 0, \quad x \in \overline{\Omega}, \end{aligned} \tag{1}$$

where the PDE is to be understood in the weak sense. Existence and uniqueness of a solution $y \in L^2(0, T, H_0^1(\Omega))$ of (1) follow from the Lions-Lax-Milgram theorem.

For now, we choose to model the sources $\phi_1, \dots, \phi_l$ as Gaussian functions with centers $\tilde{x}_1, \dots, \tilde{x}_l$ in the interior of Ω. Thus, for $x \in \mathbb{R}^2$,

$$\phi_i(x) := \kappa e^{-\frac{\|x-\tilde{x}_i\|_2^2}{\omega}} \quad \text{(Gaussian)}\,,\; i = 1, \dots, l, \tag{2}$$

with height $\kappa > 0$ and width $\omega > 0$, and we will provide further details in Sect. 5. Introducing the space of binary control functions in time

$$\mathcal{U} := \{u \in L^\infty((0,T) \times \mathbb{R}^l) | u : (0,T) \to \{0,1\}^l\},$$

the optimal control problem in function spaces then reads: given a desired state $y_d \in L^2((0,T) \times \Omega)$, find a solution pair $(y,u) \in L^2(0,T,H_0^1(\Omega)) \times \mathcal{U}$ of

$$\begin{aligned} \min_{\substack{y \in L^2(0,T,H_0^1(\Omega)) \\ u \in \mathcal{U}}} \quad & \tfrac{1}{2}\int_0^T \int_{\Omega_{\text{obs}}} (y - y_d)^2 \,\mathrm{d}x\,\mathrm{d}t, \\ \text{s.t.} \quad & (y,u) \text{ fulfill (1), and } \textstyle\sum_{i=1}^l u^i(t) \le S \in \mathbb{N},\ \forall t \in (0,T), \end{aligned} \tag{3}$$

where $\Omega_{\text{obs}} \subset \Omega$ is our domain of observation and the inequality constraint in (3) is commonly referred to as a *knapsack constraint*. This problem can be interpreted as fitting a desired heating pattern y_d over a domain of observation Ω_{obs} by activating up to $S \in \mathbb{N}$ many sources at each point in time where the sources are distributed over Ω. Clearly, as soon as the control is suitably discretized such that the discretized feasible set only contains finitely many controls (and since for each control there is a uniquely determined state y), this discretized problem will in its essence be a combinatorial problem such that existence of at least one global minimizer will be guaranteed.

2.2 Discretized model problem and continuous relaxation

We begin with a semidiscretization of (1) in space via the well-known *method of lines*. Introducing a conforming mesh over Ω using N vertices and letting $M \in \mathbb{R}^{N\times N}$ and $K \in \mathbb{R}^{N\times N}$ denote the mass and stiffness matrices (do note that K and M are positive definite and M is symmetric), we end up with the system of ordinary differential equations (ODEs)

$$M\frac{\partial}{\partial t}y(t) + Ky(t) = M\Phi u(t), \quad t \in (0,T), \quad y(0) = 0. \tag{4}$$

Here, $\Phi \in \mathbb{R}^{N\times l}$ contains the finite element coefficients of the source functions in its columns, i.e., each column contains the evaluation of the respective source function at the N vertices of the grid. Thus, $M\Phi u(t)$ with the vector-valued control function $u(t) : [0,T] \to \mathbb{R}^l$ realizes the semidiscrete right-hand side. Finally, $y : [0,T] \to \mathbb{R}^N$ now contains the FEM-coefficients of the solution.

The ODE system (4) can now be solved with a time integration method of choice and we choose the Crank-Nicholson scheme. Introducing an equidistant time-grid with $n_t \in \mathbb{N}$ points and step size $\delta_t := \frac{T}{n_t - 1}$ and letting $y_i \approx y(i\delta_t) \in \mathbb{R}^N$ as well as $u_i \approx u(i\delta_t) \in \mathbb{R}^l$ denote the corresponding approximations, the scheme reads

$$(M + \frac{\delta_t}{2}K)y_{i+1} = (M - \frac{\delta_t}{2}K)y_i + \frac{\delta_t}{2}M\Phi u_i + \frac{\delta_t}{2}M\Phi u_{i+1}, \tag{5}$$

for $i = 0, \dots, n_t - 1$. Introducing the matrices $K_1 := M + \frac{\delta_t}{2}K$ and $K_2 := M - \frac{\delta_t}{2}K$ as well as the matrices

$$I_1 := \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{n_t \times n_t} \quad \text{and} \quad I_2 := \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{n_t \times n_t},$$

we define

$$\tilde{K} := I_1 \otimes K_1 + I_2 \otimes -K_2 \in \mathbb{R}^{n_t \cdot N \times n_t \cdot N} \text{ and} \tag{6}$$

$$\tilde{\boldsymbol{\Phi}} := \frac{\delta_t}{2}\left(I_1 \otimes M\boldsymbol{\Phi} + I_2 \otimes M\boldsymbol{\Phi}\right)\mathbb{R}^{n_t \cdot N \times n_t \cdot N}, \tag{7}$$

where $\otimes$ denotes the Kronecker product of matrices. Using $\tilde{K}$ and $\tilde{\boldsymbol{\Phi}}$, equation (5) for $i = 0, \dots, n_t - 1$ can be written as

$$\tilde{K}y = \tilde{\boldsymbol{\Phi}}u, \tag{8}$$

where from now on $y := \left(y_1^\top, \dots, y_{n_t}^\top\right)^\top \in \mathbb{R}^{n_t \cdot N}$ and $u := \left(u_1^\top, \dots, u_{n_t}^\top\right)^\top \in \mathbb{R}^{n_t \cdot l}$ denote the fully space-time discretized state and control vectors.

Assuming that the observation domain Ω_{obs} is aligned with the FEM grid and that it contains p vertices of the grid, $M_{\text{obs}} \in \mathbb{R}^{p \times p}$ denotes the mass matrix of Ω_{obs} and the matrix $C \in \mathbb{R}^{p \times N}$ then realizes the evaluation of the state on Ω_{obs}. Letting $\mathbf{1}_l := (1, \dots, 1)^\top \in \mathbb{R}^l$ denote the l-dimensional unit column vector, we define

$$\begin{aligned} \tilde{M} &:= I_1 \otimes \left(C^\top M_{\text{obs}} C\right) \in \mathbb{R}^{n_t \cdot N \times n_t \cdot N} \quad \text{as well as} \\ C_{\text{ineq}} &:= I_1 \otimes \mathbf{1}_l^\top \in \mathbb{R}^{n_t \times n_t \cdot l}, \quad \text{and} \quad S_{\text{vec}} := \mathbf{1}_{n_t} S, \end{aligned} \tag{9}$$

where the matrix $\tilde{M}$ is singular as, roughly speaking the observation operator C zeros out the contributions to the mass matrix that are not observed. This property of $\tilde{M}$ will be taken into account in the construction of the iterative schemes for solving the linear systems involving $\tilde{M}$ in Sect. 3.2.1. With this notation at hand, we can formulate the *discretized optimal control problem*

$$\begin{aligned} \min_{y\in\mathbb{R}^{n_t\cdot N},u\in\mathbb{R}^{n_t\cdot l}} \quad & \tfrac{1}{2}(y-y_d)^\top \tilde{M}(y-y_d), \\ \text{s.t.} \quad & \tilde{K}y = \tilde{\boldsymbol{\Phi}}u,\ u\in\{0,1\}^{n_t\cdot l}, C_{\text{ineq}}u \leq S_{\text{vec}}. \end{aligned} \tag{10}$$

In (10) and for the remainder of this article, y_d represents a finite element coefficient vector instead of an actual $L^2((0,T)\times\Omega)$-function. Relaxing the integer constraints in (10) yields the *continuous relaxation*

$$\begin{aligned} \min_{y\in\mathbb{R}^{n_t\cdot N},u\in\mathbb{R}^{n_t\cdot l}} \quad & \tfrac{1}{2}(y-y_d)^\top \tilde{M}(y-y_d), \\ \text{s.t.} \quad & \tilde{K}y = \tilde{\boldsymbol{\Phi}}u,\ u\in[0,1]^{n_t\cdot l},\ C_{\text{ineq}}u \leq S_{\text{vec}}. \end{aligned} \tag{11}$$

We reformulate both problems (10) and (11) in a more compact way.

Lemma 1 *Introducing for $x\in\mathbb{R}^{n_t(N+l)}$ the functions*

$$\tilde{J}(x) := \frac{1}{2}x^\top \begin{bmatrix} \tilde{M} & 0 \\ 0 & 0 \end{bmatrix} x - x^\top \begin{bmatrix} \tilde{M}y_d \\ 0 \end{bmatrix} + \frac{1}{2}y_d^\top \tilde{M}y_d$$

and $f:\mathbb{R}^{n_t\cdot l}\to\mathbb{R}^{n_t\cdot N}: u\mapsto \tilde{K}^{-1}\tilde{\boldsymbol{\Phi}}u$, problems (10) and (11) are equivalent to

$$\begin{aligned} &\min_{x\in W}\tilde{J}(x) \quad \text{with the feasible set} \\ &W := \left\{ x = (f(u)^\top, u^\top)^\top \in \mathbb{R}^{n_t(N+l)} \;\middle|\; u\in\{0,1\}^{n_t\cdot l}, C_{\text{ineq}}u\leq S_{\text{vec}} \right\} \end{aligned} \tag{P}$$

and

$$\begin{aligned} &\min_{x\in X}\tilde{J}(x) \quad \text{with the feasible set} \\ &X := \left\{ x = (f(u)^\top, u^\top)^\top \in \mathbb{R}^{n_t(N+l)} \;\middle|\; u\in[0,1]^{n_t\cdot l}, C_{\text{ineq}}u\leq S_{\text{vec}} \right\}, \end{aligned} \tag{Pcont}$$

respectively. Furthermore, $W\subset\mathbb{R}^{n_t(N+l)}$ is compact and $X\subset\mathbb{R}^{n_t(N+l)}$ is compact and convex such that (Pcont) is a convex problem.

Proof The proof follows the same arguments as [9, Lemma 2.2] and is thus omitted here. □

2.3 Penalty reformulation

Starting from the continuous relaxation (11), we add the well-known penalty term

$$\frac{1}{\varepsilon}\sum_{j=1}^{n_t\cdot l} u^{(j)}(1-u^{(j)}) \tag{12}$$

to the objective function. Obviously, this concave penalty term penalizes a non-binary control, where $\varepsilon > 0$ determines the amount of penalization. This yields the following *penalty formulation*

$$\begin{aligned} \min_{y\in\mathbb{R}^{n_t\cdot N}, u\in\mathbb{R}^{n_t\cdot l}} \quad & \tfrac{1}{2}(y-y_d)^{\mathsf{T}}\tilde{M}(y-y_d) + \tfrac{1}{\varepsilon}\textstyle\sum_{j=1}^{n_t\cdot l} u^{(j)}(1-u^{(j)}), \\ \text{s.t.} \quad & \tilde{K}y = \tilde{\Phi}u, \; u \in [0,1]^{n_t\cdot l}, C_{\text{ineq}}u \le S_{\text{vec}}. \end{aligned} \tag{13}$$

Following Lemma 1, (13) can be rewritten as

$$\begin{aligned} &\min_{x\in X} J(x;\varepsilon), \qquad \text{with} \\ &J(x;\varepsilon) := \frac{1}{2}x^{\mathsf{T}}\begin{bmatrix}\tilde{M} & 0\\ 0 & -\frac{2}{\varepsilon}I_{n_t\cdot l}\end{bmatrix}x - x^{\mathsf{T}}\begin{bmatrix}\tilde{M}y_d\\ -\frac{1}{\varepsilon}\mathbf{1}_{n_t\cdot l}\end{bmatrix} + \frac{1}{2}y_d^{\mathsf{T}}\tilde{M}y_d, \end{aligned} \tag{Ppen}$$

where $I_{n_t\cdot l} \in \mathbb{R}^{n_t\cdot l\times n_t\cdot l}$ is the identity-matrix.

Proposition 1 *There exists an $\tilde{\varepsilon} > 0$ such that for all $\varepsilon \in (0,\tilde{\varepsilon}]$ problems (P) and (Ppen) have the same global minima (if there exist multiple). In this sense both problems (P) and (Ppen) are equivalent.*

Proof From Lemma 1 we know that W and X are compact and since $\tilde{J}$ is a quadratic function, it clearly holds that $\tilde{J} \in C^1(\mathbb{R}^{N+l})$. Together with the results derived in [26, Section 3] all assumptions of [26, Theorem 2.1] are fulfilled such that the desired statement follows. □

Proposition 1 holds for a variety of concave penalty terms, see, e.g., [26, Eqs 19–23] or [27, Eq. 21]. Nevertheless, we chose the penalty term (12) here since it is quadratic and thus the combined objective function J remains quadratic.

The repeated solution of the penalty formulation (Ppen) for an increasing value of the penalty parameter ε will be the core of our solution procedure for the overall MIPDECO problem (P). This procedure will be based on the IPA algorithm proposed in [9] and its extension to the time-dependent setting is postponed to Sect. 4. In the following section we present the main algorithmic novelty instead, that is the suitable combination of the model order reduction and the interior point method for the efficient solution of the penalty formulation (Ppen).

3 Model Order Reduction and Interior Point Methods

Solving the overall MIPDECO problem (P) via a penalty approach requires numerous solves of (Ppen). Thus, an efficient method to handle the problem (Ppen) for a given ε is crucial to an overall effective solution procedure. With this purpose, we present two procedures based on interior point methods: the first one, a generalization of the interior point method (IPM) proposed in [9] to the time-dependent case, will be denoted full-IPM. The second one, a novel combination of model order reduction (MOR) and an IPM, will be denoted MOR-IPM and is the main contribution of this work. Both methods are equipped with a preconditioning technique that exploits the specific problem structure in (Ppen).

3.1 Model order reduction approach

The central idea is to derive a low-dimensional approximation of the PDE constraint

$$\tilde{K}y = \tilde{\boldsymbol{\Phi}}u \quad \text{via} \quad \tilde{K}_{\mathrm{red}}y_{\mathrm{red}} = \tilde{\boldsymbol{\Phi}}_{\mathrm{red}}u,$$

with suitable $\tilde{K}_{\mathrm{red}} \in \mathbb{R}^{n_t \cdot r \times n_t \cdot r}$ and $\tilde{\boldsymbol{\Phi}}_{\mathrm{red}} \in \mathbb{R}^{n_t \cdot r \times n_t \cdot l}$ such that the *reconstruction* $\hat{C}y_{\mathrm{red}} \in \mathbb{R}^{n_t \cdot N}$, with $\hat{C} \in \mathbb{R}^{n_t \cdot N \times n_t \cdot r}$, is a good approximation to y. It is clear that only the dimension of the state is reduced to $r \ll N$, where the dimension of the control (in fact the control as a whole) remains untouched. Based on this approximation, we can then (similarly to Lemma 1) introduce the linear mapping $f_{\mathrm{red}} : \mathbb{R}^{n_t \cdot l} \to \mathbb{R}^{n_t \cdot N} : u \mapsto \hat{C}\tilde{K}_{\mathrm{red}}^{-1}\tilde{\boldsymbol{\Phi}}_{\mathrm{red}}u$ and formulate the reduced version of the penalty formulation

$$\begin{aligned}&\min_{x \in X_{\mathrm{red}}} J(x;\varepsilon) \quad \text{with the feasible set}\\ &X_{\mathrm{red}} := \left\{ x = (f_{\mathrm{red}}(u)^{\top}, u^{\top})^{\top} \in \mathbb{R}^{n_t(N+l)} \,\middle|\, u \in [0,1]^{n_t \cdot l}, C_{\mathrm{ineq}}u \le S_{\mathrm{vec}} \right\}.\end{aligned} \tag{Ppen$_{\mathrm{red}}$}$$

Thus, only the linear map inside the feasible set changes and the better f_{red} approximates f, the closer X and X_{red} are. In the same fashion, the reduced mixed-integer control problem can be formulated as

$$\begin{aligned}&\min_{x \in W_{\mathrm{red}}} \tilde{J}(x) \quad \text{with the feasible set}\\ &W_{\mathrm{red}} := \left\{ x = (f_{\mathrm{red}}(u)^{\top}, u^{\top})^{\top} \in \mathbb{R}^{n_t(N+l)} \,\middle|\, u \in \{0,1\}^{n_t \cdot l}, C_{\mathrm{ineq}}u \le S_{\mathrm{vec}} \right\}.\end{aligned} \tag{P$_{\mathrm{red}}$}$$

Again, the more accurate our approximation of the PDE is, the better f_{red} approximates f and the closer W_{red} is to W. Furthermore, the reduced penalty formulation (Ppen$_{\mathrm{red}}$) links to the reduced optimal control problem (P$_{\mathrm{red}}$) in the same way as (Ppen) links to (P) in Proposition 1, i.e., there exists an $\tilde{\varepsilon} > 0$ such that for all $\varepsilon \in (0, \tilde{\varepsilon}]$ problems (P$_{\mathrm{red}}$) and (Ppen$_{\mathrm{red}}$) have the same minimum points.

Before we elaborate on the theoretical justification of this approach, we want to actually apply our model order reduction technique of choice, the *balanced truncation*, see, e.g., [14], and explicitly derive the unknown quantities inside f_{red}, i.e., $\hat{C}$, $\tilde{K}_{\mathrm{red}}$ and $\tilde{\boldsymbol{\Phi}}_{\mathrm{red}}$.

3.1.1 Balanced truncation and reduced state system

Since the balanced truncation (BT) is a model order reduction technique for linear time-invariant (LTI) systems, we have to refer to the semidiscretized ODE-system described in (4). We reformulate the system and add an output equation (that corresponds to the evaluation of the state on the domain of observation Ω_{obs}) to fit the formulation within the standard BT literature, that is

$$
\begin{aligned}
&M\frac{\partial}{\partial t}y(t) = -Ky(t) + M\Phi u(t), \quad t \in (0,T), \quad y(0) = 0,\\
&y_{\text{out}}(t) = Cy(t).
\end{aligned} \tag{14}
$$

Note that the addition of the output equation is natural here since only the state values inside Ω_{obs} are of interest for the objective function. After the application of the BT to this system, one can then apply a time-integration method to the resulting reduced system of ODEs to obtain $\tilde{K}_{\text{red}}$, $\tilde{\Phi}_{\text{red}}$, and $\hat{C}$ and thus f_{red}.

Equation (14) is an LTI system in generalized state-space form, such that we apply the generalized BT, see, e.g., [28, 29]. We briefly recapitulate the key steps. Our aim is to construct projection matrices $T_1 \in \mathbb{R}^{r\times N}$ and $T_2 \in \mathbb{R}^{N\times r}$ such that

$$
\begin{aligned}
M_{\text{red}} &:= T_1MT_2 \in \mathbb{R}^{r\times r}, \qquad & K_{\text{red}} &:= T_1KT_2 \in \mathbb{R}^{r\times r},\\
\Phi_{\text{red}} &:= T_1M\Phi \in \mathbb{R}^{r\times l}, \qquad & C_{\text{red}} &:= CT_2 \in \mathbb{R}^{p\times r},
\end{aligned} \tag{15}
$$

yielding the reduced LTI system. First, we require factorizations $P = RR^{\mathsf{T}} \in \mathbb{R}^{N\times N}$ and $Q = LL^{\mathsf{T}} \in \mathbb{R}^{N\times N}$ of the solutions of the following generalized Lyapunov equations

$$
\begin{aligned}
&-KPM^{\mathsf{T}} - MPK^{\mathsf{T}} + M\Phi\Phi^{\mathsf{T}}M^{\mathsf{T}} = 0,\\
&-K^{\mathsf{T}}QM - M^{\mathsf{T}}QK + C^{\mathsf{T}}C = 0.
\end{aligned} \tag{16}
$$

It is well-known that P and Q are positive semi-definite, such that these factorizations exist (R and L are often called "Cholesky" factors of P and Q even if they are not Cholesky factors in the strict sense). With these factors at hand, we calculate the singular value decomposition (SVD) of $L^{\mathsf{T}}MR = U\Sigma V^{\mathsf{T}}$ and mention that up to now, all these steps can be performed in a one-time offline fashion.

Now, we choose a reduced dimension $r \ll N$ and based on this, we split the SVD with respect to this dimension r as

$$
L^{\mathsf{T}}MR = U\Sigma V^{\mathsf{T}} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^{\mathsf{T}} \\ V_2^{\mathsf{T}} \end{bmatrix} \tag{17}
$$

with $\Sigma_1 := \operatorname{diag}(\sigma_1, \dots, \sigma_r)$ and $\Sigma_2 := \operatorname{diag}(\sigma_{r+1}, \dots, \sigma_N)$, where $\sigma_r > \sigma_{r+1}$ and σ_j, $j = 1, \dots, N$ are the so-called *Hankel singular values* of the system (4). Based on this truncated SVD, we define the projection matrices

$$
T_1 := \Sigma_1^{-1/2}V_1^{\mathsf{T}}R^{\mathsf{T}} \quad \text{and} \quad T_2 := LU_1\Sigma_1^{-1/2}
$$

such that we obtain the *reduced model* as in (15). As a result, we obtain the reduced LTI system for the *reduced state* $\hat{y}_{\text{red}}(t) \in \mathbb{R}^r$

$$
\begin{aligned}
&M_{\text{red}}\frac{\partial}{\partial t}\hat{y}_{\text{red}}(t) = -K_{\text{red}}\hat{y}_{\text{red}}(t) + \Phi_{\text{red}}u(t), \quad t \in (0,T), \quad \hat{y}_{\text{red}}(0) = 0,\\
&y_{\text{red,out}}(t) = C_{\text{red}}\hat{y}_{\text{red}}(t),
\end{aligned} \tag{18}
$$

which only depends on the reduced dimension $r \ll N$, and note that the control dimension remains untouched. Similar to Sect. 2.2, we apply the Crank-Nicholson scheme to the state equation of (18), which can then again be written in an all at once formulation using Kronecker-product matrices. Letting $y_{i,\text{red}} \approx \hat{y}_{\text{red}}(i\delta_t) \in \mathbb{R}^r$, we collect these approximations in $y_{\text{red}} := \left(y_{1,\text{red}}^\mathsf{T}, \dots, y_{n_t,\text{red}}^\mathsf{T}\right)^\mathsf{T} \in \mathbb{R}^{n_t \cdot r}$ and obtain

$$\tilde{K}_{\text{red}} y_{\text{red}} = \tilde{\boldsymbol{\Phi}}_{\text{red}} u, \tag{19}$$

where

$$\begin{aligned} \tilde{K}_{\text{red}} &:= I_1 \otimes K_{1,\text{red}} + I_2 \otimes -K_{2,\text{red}} \in \mathbb{R}^{n_t \cdot r \times n_t \cdot r}, \\ \tilde{\boldsymbol{\Phi}}_{\text{red}} &:= \frac{\delta_t}{2}\left(I_1 \otimes \boldsymbol{\Phi}_{\text{red}} + I_2 \otimes \boldsymbol{\Phi}_{\text{red}}\right)\mathbb{R}^{n_t \cdot r \times n_t \cdot l}, \end{aligned} \tag{20}$$

with $K_{1,\text{red}} := M_{\text{red}} + \frac{\delta_t}{2} K_{\text{red}}$ and $K_{2,\text{red}} := M_{\text{red}} - \frac{\delta_t}{2} K_{\text{red}}$. Finally, we define

$$\hat{C} = I_1 \otimes T_2 \tag{21}$$

such that we obtain the *reconstruction* $f_{\text{red}}(u) = \hat{C} y_{\text{red}} \in \mathbb{R}^{n_t \cdot N}$, which then approximates $f(u) = \tilde{K}^{-1}\tilde{\boldsymbol{\Phi}} u$. Thus, all quantities in the reduced optimal control problem (Ppen$_{\text{red}}$) are now known and an IPM for the problem can be derived. We note that the matrix C_{red} does not appear here, since $C_{\text{red}} = CT_2$ where the T_2 part is integrated in $\hat{C}$ and the C part is already included in $\tilde{M}$ inside the objective function, see (9). We also highlight that the approximation quality of the BT relies on the size of the reduced dimension r and the investigation of this parameter will be the subject of the next section together with the analysis of further theoretical properties.

3.1.2 Theoretical insights

We present two known theoretical results for BT and relate them to our problem. First, a standard result targets the error between the output $y_{\text{out}}(t)$ of the LTI system (14), and $y_{\text{red,out}}$, the output of the reduced LTI system (18). It requires that the system is asymptotically stable, which is the case here since both M and K are positive definite. For control functions $u \in L^2(0,T)$ and if the reduced LTI system (18) was obtained via balanced truncation with reduced dimension $r \leq N$ it holds, see, e.g., [14, 30, 31], that

$$\|y_{\text{out}} - y_{\text{red,out}}\|_{L^2(0,T)} \leq 2\|u\|_{L^2(0,T)}\left(\sigma_{r+1} + \cdots + \sigma_N\right), \tag{22}$$

where $\sigma_{r+1} + \cdots + \sigma_N$ is the sum of the truncated Hankel singular values. As a result, the approximation quality of the balanced truncation depends on the size of this sum and the LTI system (14) can be well-approximated if the Hankel singular values are quickly decaying. If the decay in the singular values is very slow, the reduced dimension has to be chosen comparably large to still ensure a good approximation. But a large r negatively impacts the computational time required to solve the reduced system (19) (since it is dense) and one might not even gain a speed-up if r is too large. We state the second result from the literature, see [31, Corollary 1].

The result gives an error bound for the error between u_*, the solution of a generic quadratic optimal control problem

$$\begin{aligned} \min_{u(t)} \quad & \tfrac{1}{2}\int_0^T \left\|\mathcal{C}y(t) + \mathcal{D}u(t) - y_d(t)\right\|^2 \,\mathrm{d}t, \\ \text{s.t.} \quad & \mathcal{M}\frac{\partial}{\partial t}y(t) = \mathcal{A}y(t) + \mathcal{B}u(t), \quad t \in (0,T), \quad y(0) = y_0, \end{aligned} \tag{23}$$

and $\hat{u}_*$, the solution of a corresponding reduced optimal control problem

$$\begin{aligned} \min_{u(t)} \quad & \tfrac{1}{2}\int_0^T \left\|\hat{\mathcal{C}}\hat{y}(t) + \mathcal{D}u(t) - y_d(t)\right\|^2 \,\mathrm{d}t, \\ \text{s.t.} \quad & \frac{\partial}{\partial t}\hat{y}(t) = \hat{\mathcal{A}}\hat{y}(t) + \hat{\mathcal{B}}u(t), \quad t \in (0,T), \quad \hat{y}(0) = \hat{y}_0, \end{aligned} \tag{24}$$

where the reduced ODE system was obtained via balanced truncation. The result furthermore requires that $\mathcal{M}$ is symmetric, positive definite, that there exists an $\alpha > 0$ such that $v^\top \mathcal{A}v \le -\alpha v^\top \mathcal{M}v$ for all $v \in \mathbb{R}^N$, and that the objective function of (23) is strictly convex. Then, [31, Corollary 1] yields the bound

$$\left\|u_* - \hat{u}_*\right\|_{L^2(0,T)} \le \frac{2}{\kappa}\Big(c\left\|\hat{u}_*\right\|_{L^2(0,T)} + \left\|\hat{z}_*\right\|_{L^2(0,T)}\Big)\big(\sigma_{r+1} + \cdots + \sigma_N\big), \tag{25}$$

where κ is a constant associated with the convexity of the objective of (23), $\hat{z}_* = \hat{\mathcal{C}}\hat{y}_* + \mathcal{D}\hat{u}_* - y_d$ with $\hat{y}_*$ being the state corresponding to $\hat{u}_*$, and c is a constant associated to the ODE system.

This result could be applied to the continuous relaxation (Pcont) with $\mathcal{C} = C, \mathcal{D} = 0$, $\mathcal{M} = M$, $\mathcal{A} = -K$, and $\mathcal{B} = M\Phi$. Conversely, a bound similar to (25) cannot be expected for the penalty formulation (Ppen) (on which the mixed-integer approach in this work is based) as convexity of the objective function is out of reach there.

Nonetheless, we want to stress that the driving term in the bound (25) again is the sum of the remaining Hankel singular values. Thus, if this sum is small and the reduced system provides a good approximation, one can infer that a solution of (Ppen$_{\text{red}}$) is sufficiently close to the corresponding solution of (Ppen). With (22) at hand, the feasible sets X_{red} and X should then be close enough. The solutions of the overall MIPDECO problems (P$_{\text{red}}$) and (P) should be close, or even the same, as well.

3.2 The interior point framework

We now briefly describe the main steps of the two interior point methods that will be employed to solve the reduced (Ppen$_{\text{red}}$) and full (Ppen) formulations, respectively. The derivation of the IPMs follows [32] and, more specifically, [9]. We first observe that problems (Ppen$_{\text{red}}$) and (Ppen) can be rewritten as

$$\begin{aligned} \min_{\substack{y_{\text{red}} \in \mathbb{R}^{n_t \cdot r}, u \in \mathbb{R}^{n_t \cdot l}, \\ z \in \mathbb{R}^{n_t}}} \quad & \tfrac{1}{2}(\hat{C}y_{\text{red}} - y_d)^{\intercal} \tilde{M}(\hat{C}y_{\text{red}} - y_d) + \tfrac{1}{\varepsilon}(\mathbf{1}_{n_t \cdot l}^{\intercal} u - u^{\intercal} u), \\ \text{s.t.} \quad & \tilde{K}_{\text{red}} y_{\text{red}} = \tilde{\boldsymbol{\Phi}}_{\text{red}} u \qquad \text{and} \qquad C_{\text{ineq}} u + z - S_{\text{vec}} = 0, \\ & 0 \le u \le 1 \qquad \text{and} \qquad z \ge 0, \end{aligned} \tag{26}$$

and

$$\begin{aligned} \min_{y \in \mathbb{R}^{n_t \cdot N}, u \in \mathbb{R}^{n_t \cdot l}, z \in \mathbb{R}^{n_t}} \quad & \tfrac{1}{2}(y - y_d)^{\intercal} \tilde{M}(y - y_d) + \tfrac{1}{\varepsilon}(\mathbf{1}_{n_t \cdot l}^{\intercal} u - u^{\intercal} u), \\ \text{s.t.} \quad & \tilde{K} y = \tilde{\boldsymbol{\Phi}} u \qquad \text{and} \qquad C_{\text{ineq}} u + z - S_{\text{vec}} = 0, \\ & 0 \le u \le 1 \qquad \text{and} \qquad z \ge 0, \end{aligned} \tag{27}$$

respectively, where $0 \le z \in \mathbb{R}^{n_t}$ is a vector of slack variables. We recall that $\tilde{M}$ is defined in (9) and handles the observation on the subdomain Ω_{obs}.

The main idea of an IPM is the elimination of the inequality constraints on u and z via the introduction of corresponding logarithmic barrier functions weighted by the barrier parameter $\mu > 0$ that controls the relation between the barrier term and the original objectives. Then, first-order optimality conditions are derived by applying duality theory resulting in a nonlinear system parametrized by μ. For problem (26) the nonlinear system takes the form

$$\hat{C}^T \tilde{M} \hat{C} y_{\text{red}} - \hat{C}^T \tilde{M} y_d + \tilde{K}_{\text{red}}^{\intercal} p = 0, \tag{28a}$$

$$\frac{1}{\varepsilon}(\mathbf{1}_{n_t \cdot l} - 2u) - \tilde{\boldsymbol{\Phi}}_{\text{red}}^{\intercal} p + C_{\text{ineq}}^{\intercal} q - \lambda_{u,0} + \lambda_{u,1} = 0, \tag{28b}$$

$$q - \lambda_{z,0} = 0, \quad \tilde{K}_{\text{red}} y_{\text{red}} - \tilde{\boldsymbol{\Phi}}_{\text{red}} u = 0, \quad C_{\text{ineq}} u + z - S_{\text{vec}} = 0, \tag{28c}$$

where the Lagrange multipliers $\lambda_{u,0}$, $\lambda_{u,1} \in \mathbb{R}^{n_t \cdot l}$, and $\lambda_{z,0} \in \mathbb{R}^{n_t}$ are defined as

$$(\lambda_{u,0})_i := \frac{\mu}{u_i}, \ (\lambda_{u,1})_i := \frac{\mu}{1 - u_i}, \ i = 1, \dots, n_t \cdot l, \ \text{ and } \ (\lambda_{z,0})_i := \frac{\mu}{z_i}, \ i = 1, \dots, n_t.$$

Furthermore, the bound constraints $\lambda_{u,0} \ge 0$, $\lambda_{u,1} \ge 0$, and $\lambda_{z,0} \ge 0$ then enforce the constraints on u and z. Here $p \in \mathbb{R}^{n_t \cdot r}$ is the reduced Lagrange multiplier (or adjoint variable) associated with the reduced state equation and $q \in \mathbb{R}^{n_t}$ is the Lagrange multiplier associated with the equations $C_{\text{ineq}} u + z - S_{\text{vec}} = 0$.

The crucial step of deriving the IPM is the application of Newton's method to the above nonlinear system. Letting y_{red}, u, z, p, q, $\lambda_{u,0}$, $\lambda_{u,1}$, and $\lambda_{z,0}$ denote the most recent Newton iterates, these are then updated in each iteration by computing the corresponding Newton steps Δy_{red}, Δu, Δz, Δp, Δq, $\Delta \lambda_{u,0}$, $\Delta \lambda_{u,1}$, and $\Delta \lambda_{z,0}$ through the solution of the Newton system with the following coefficient matrix

$$\mathcal{N}_{\text{red}} = \begin{bmatrix} \hat{C}^T\tilde{M}\hat{C} & 0 & 0 & \tilde{K}_{\text{red}}^{\mathsf{T}} & 0 \\ 0 & -\frac{2}{\varepsilon}I_{n_t\cdot l} + \Theta_u & 0 & -\tilde{\Phi}_{\text{red}}^{\mathsf{T}} & C_{\text{ineq}}^{\mathsf{T}} \\ 0 & 0 & \Theta_z & 0 & I_{n_t} \\ \tilde{K}_{\text{red}} & -\tilde{\Phi}_{\text{red}} & 0 & 0 & 0 \\ 0 & C_{\text{ineq}} & I_{n_t} & 0 & 0 \end{bmatrix}. \tag{29}$$

Here, $\Theta_u := U^{-1}\Lambda_{u,0} + (I_l - U)^{-1}\Lambda_{u,1}$, $\Theta_z := Z^{-1}\Lambda_{z,0}$, and $U,\ Z,\ \Lambda_{u,0},\ \Lambda_{u,1}$, as well as $\Lambda_{z,0}$ are diagonal matrices with the most recent iterates of $u,\ z,\ \lambda_{u,0},\ \lambda_{u,1}$, and $\lambda_{z,0}$ appearing on their diagonal entries. Once the the Newton system is solved, one can compute the steps for the Lagrange multipliers via

$$\begin{aligned} \Delta\lambda_{u,0} &= -\lambda_{u,0} - U^{-1}(\Lambda_{u,0}\Delta u - \mu\mathbf{1}_{n_t\cdot l}), \\ \Delta\lambda_{u,1} &= -\lambda_{u,1} + (I_{n_t\cdot l} - U)^{-1}(\Lambda_{u,1}\Delta u + \mu\mathbf{1}_{n_t\cdot l}), \\ \Delta\lambda_{z,0} &= -\lambda_{z,0} - Z^{-1}(\Lambda_{z,0}\Delta z - \mu\mathbf{1}_{n_t}). \end{aligned}$$

A general IPM implementation only involves one Newton step per iteration. Thus, after choosing suitable step-lengths so that the updated iterates remain feasible, the new iterates can be calculated and the barrier parameter μ is reduced, thus concluding one iteration of the IPM. Finally, we report the primal and dual feasibilities

$$\xi_p := \begin{bmatrix} \tilde{K}_{\text{red}}y_{\text{red}} - \tilde{\Phi}_{\text{red}}u \\ C_{\text{ineq}}u + z - S_{\text{vec}} \end{bmatrix}, \xi_d := \begin{bmatrix} \hat{C}^T\tilde{M}\hat{C}y_{\text{red}} - \hat{C}^T\tilde{M}y_d + \tilde{K}_{\text{red}}^{\mathsf{T}}p \\ \frac{1}{\varepsilon}(\mathbf{1}_{n_t\cdot l} - 2u) - \tilde{\Phi}_{\text{red}}^{\mathsf{T}}p + C_{\text{ineq}}^{\mathsf{T}}q - \lambda_{u,0} + \lambda_{u,1} \\ q - \lambda_{z,0} \end{bmatrix}$$

as well as the complementarity gap

$$\xi_c := \left[U\lambda_{u,0} - \mu\mathbf{1}_{n_t\cdot l},\ (I_{n_t\cdot l} - U)\lambda_{u,1} - \mu\mathbf{1}_{n_t\cdot l},\ Z\Lambda_{z,0} - \mu\mathbf{1}_{n_t}\right]^{\mathsf{T}},$$

where measuring the change in the norms of ξ_p, ξ_d, and ξ_c allows us to monitor the convergence of the entire process. This completes the general description of the MOR-IPM.

The derivation of the full-IPM for problem (27) is analogous taking into account that the adjoint variable $p \in \mathbb{R}^{n_t\cdot N}$ now depends on the full dimension N instead of the reduced dimension r (thus, we chose not to introduce extra notation). The coefficient matrix of the resulting Newton system takes the form

$$\mathcal{N} = \begin{bmatrix} \tilde{M} & 0 & 0 & \tilde{K}^{\mathsf{T}} & 0 \\ 0 & -\frac{2}{\varepsilon}I_{n_t\cdot l} + \Theta_u & 0 & -\tilde{\Phi}^{\mathsf{T}} & C_{\text{ineq}}^{\mathsf{T}} \\ 0 & 0 & \Theta_z & 0 & I_{n_t} \\ \tilde{K} & -\tilde{\Phi} & 0 & 0 & 0 \\ 0 & C_{\text{ineq}} & I_{n_t} & 0 & 0 \end{bmatrix}, \tag{30}$$

where the diagonal matrices Θ_u and Θ_z are defined as for the matrix $\mathcal{N}_{\text{red}}$ in (29).

We note that the matrices $\mathcal{N}_{\text{red}}$ and $\mathcal{N}$ in (29) and (30) have the same block structure. Moreover we observe that $\tilde{M}$ is symmetric as by (9) it inherits the symmetry from M_{obs}, and singular; $\tilde{K}_{\mathrm{red}}$ and $\tilde{K}$ are not symmetric no matter the symmetry of the original stiffness matrix, see definitions (20) and (6) respectively; Θ_u and $\Theta_z > 0$, while being positive definite, are typically very ill-conditioned. Moreover, due to the term $-\frac{2}{\varepsilon} I_{n_t \cdot l}$, the block $-\frac{2}{\varepsilon} I_{n_t \cdot l} + \Theta_u$ may be indefinite, especially for small values of ε. Following suggestions in [33, Chapter 19.3] to handle nonconvexities in the objective function by promoting the computation of descent directions, we heuristically keep the diagonal matrix $-\frac{2}{\varepsilon} I_{n_t \cdot l} + \Theta_u$ positive definite by setting any negative values to a small positive value $\gamma > 0$. This strategy was already implemented in [9] demonstrating very promising numerical performance.

From a computational point of view, the burden of any IPM lies in the solution of the Newton system at each iteration. Clearly, the developed MOR-IPM is expected to be more efficient than the full-IPM since the dominant state dimension is reduced as depicted in Fig. 1.

We employ the following strategy to handle the linear algebra phase inside the IPMs: on the one hand we employ an inexact Krylov strategy for the solution of the Newton system and on the other hand we design a suitable preconditioner to speed up the convergence of our Krylov method of choice. This strategy allows to implement the IPMs in a matrix-free manner so that the matrices defined by a Kronecker product need not be explicitly formed as the corresponding products can be performed by suitable multiplication functions using the Kronecker factors. Regarding the inexactness strategy, the idea is to increase the accuracy in the solution of the Newton equation as μ decreases in order to get savings in the computational time. Global convergence results to a solution of the first-order optimality conditions for inexact IPMs can be found in [34]. Finally, we remark that for the MOR-IPM, compared to the full-IPM, the inexactness strategy did not play an essential role and in fact in our numerical experiments the linear systems will be solved to high accuracy without affecting the overall cpu time.

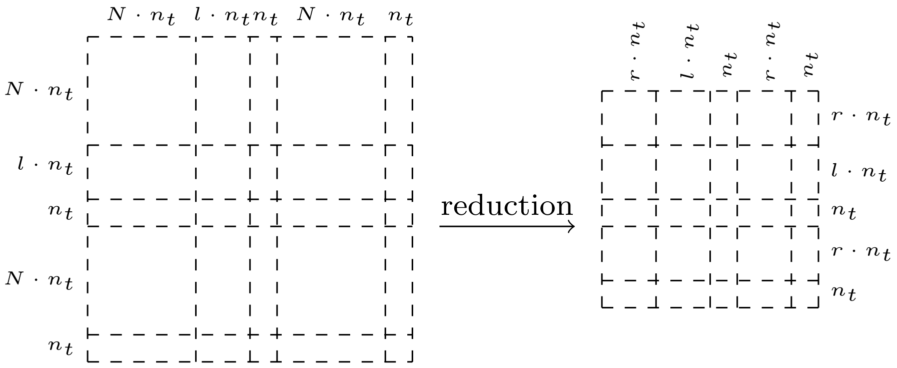

Fig. 1 Schematic dimension reduction from the Newton equation with $\mathcal{N}$ in (30) to the reduced one with $\mathcal{N}_{red}$ in (29) obtained via balanced truncation

3.2.1 Preconditioning

Preconditioning is a crucial tool for accelerating the speed of convergence of any Krylov method. We here focus on the Newton scheme relying on the solution of the Newton system with coefficient matrix $\mathcal{N}$ given in (30) but the same considerations can be applied to linear systems with $\mathcal{N}_{\text{red}}$ in (29) and are not reported for the sake of conciseness.

Given the partial observation problem, the matrix $\tilde{M}$ is indeed singular while the overall saddle point system is invertible but requires a carefully designed preconditioner. For deriving the preconditioner we follow the strategy presented in [35] where we consider a permutation of the Newton system. We only do this for the sake of deriving a preconditioner as the permuted systems is amenable to standard saddle point theory. This permutation results in the saddle point system $\left[A\ B_2^T; B_1\ D\right]$ with blocks

$$A = \begin{bmatrix} \tilde{K} & -\tilde{\Phi} & 0 \\ 0 & -\frac{2}{\varepsilon} I_{n_t \cdot l} + \Theta_u & 0 \\ 0 & 0 & \Theta_z \end{bmatrix}, B_1 = \begin{bmatrix} \tilde{M} & 0 & 0 \\ 0 & C_{\text{ineq}} & I_{n_t} \end{bmatrix}, B_2 = \begin{bmatrix} 0 & -\tilde{\Phi} & 0 \\ 0 & C_{\text{ineq}} & I_{n_t} \end{bmatrix}, D = \begin{bmatrix} \tilde{K}^{\mathsf{T}} & 0 \\ 0 & 0 \end{bmatrix}.$$

Our preconditioning strategy is now based on creating a preconditioner of block-triangular form $\left[\tilde{A}\ 0; B_1\ \ -\tilde{S}\right]$ where $\tilde{A} \approx A$ and $\tilde{S}$ approximates the Schur-complement S. This is achieved via

$$\begin{aligned} S &\approx \begin{bmatrix} \tilde{K}^{\mathsf{T}} & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \tilde{M} & 0 & 0 \\ 0 & C_{\text{ineq}} & I_{n_t} \end{bmatrix} \begin{bmatrix} \tilde{K}^{-1} & 0 & 0 \\ 0 & (-\frac{2}{\varepsilon} I_{n_t \cdot l} + \Theta_u)^{-1} & 0 \\ 0 & 0 & \Theta_z^{-1} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -\tilde{\Phi}^{\mathsf{T}} & C_{\text{ineq}}^{\mathsf{T}} \\ 0 & I_{n_t} \end{bmatrix} \\ &\approx \begin{bmatrix} \tilde{K}^{\mathsf{T}} & 0 \\ 0 & -\Theta_z^{-1} - C_{\text{ineq}}(-\frac{2}{\varepsilon} I_{n_t \cdot l} + \Theta_u)^{-1} C_{\text{ineq}}^T, \end{bmatrix}, \end{aligned}$$

where in the first approximation we replace the (1, 1)-block by its diagonal and then in the second one we ignore the (2, 1)-block of the approximation to obtain a block-diagonal approximation of the Schur-complement. We embed this into the overall preconditioner obtained as

$$\mathcal{P} = \begin{bmatrix} \tilde{K} & 0 & 0 & 0 & 0 \\ 0 & (-\frac{2}{\varepsilon} I_{n_t \cdot l} + \Theta_u) & 0 & 0 & 0 \\ 0 & 0 & \Theta_z & 0 & 0 \\ \tilde{M} & 0 & 0 & -\tilde{K}^{\mathsf{T}} & 0 \\ 0 & C_{\text{ineq}} & 0 & 0 & \Theta_z^{-1} + C_{\text{ineq}}(-\frac{2}{\varepsilon} I_{n_t \cdot l} + \Theta_u)^{-1} C_{\text{ineq}}^T \end{bmatrix}. \tag{31}$$

We have now derived a preconditioner for the permuted problem that we do not want to form and now translate this theoretical detour back to the original saddle point system following [35]. We then obtain the preconditioner for the original problem as $\tilde{\mathcal{P}}$, given in an efficiently implemented version via

$$\tilde{\mathcal{P}}^{-1}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} = \begin{bmatrix} \tilde{K}^{-1}w_4 \\ C^{-1}w_2 \\ \Theta_z^{-1}w_3 \\ \tilde{K}^{-T}\left(-w_1 + \tilde{M}\tilde{K}^{-1}w_4\right) \\ (\Theta_z^{-1} + C_{\text{ineq}}(-\frac{2}{\varepsilon}I_{n_t\cdot l} + \Theta_u)^{-1}C_{\text{ineq}}^T)^{-1}(C_{\text{ineq}}C^{-1}w_2 + w_5) \end{bmatrix},$$

which clearly shows that we need to solve once with $\tilde{K}$ and once with $\tilde{K}^T$. We combine this preconditioner with the GMRES method of [36]. We here focus on the highly relevant case of a partial observation domain, which as we already pointed out renders the matrix $\tilde{M}$ singular. In case of a full observation, we proposed a preconditioner in [9] that we believe can be easily extended to the time-dependent case, see [37] for a preconditioning method for full observation optimization.

4 Time-dependent Improved Penalty Algorithm (tIPA)

With the IPMs from the previous section at hand, we want to solve the overall MIPDECO problem. In order to do so, we adapt the improved penalty algorithm (IPA), developed in [9], to this time-dependent setting. Before that, we make the following clarifying remark.

Remark 1

(i) Applying the (soon described) IPA strategy to the MIPDECO problem (P) involves repeated solutions of the penalty formulation (Ppen) where the full-IPM from Sect. 3.2 can be used.

(ii) In the same way, the IPA strategy can be applied to the reduced MIPDECO problem (P_{red}) which then involves solutions of the reduced penalty formulation (Ppen_{red}) where the MOR-IPM from Sect. 3.2 can be used.

(iii) To avoid confusion, we will describe the IPA based on (P) and (Ppen), but want to stress that the MOR version of the algorithm can be easily obtained by simply replacing the feasible sets X and W by X_{red} and W_{red} as well as replacing the linear map f by f_{red} throughout this section.

(iv) As a result, we will obtain two algorithms: one solving (P) and one solving (P_{red}), where (P_{red}) approximates (P) and the approximation quality is based on the quality of our model order reduction.

We first extend the rounding strategy developed in [9, Definition 3.4] to the time-dependent setting to again suitably handle the knapsack constraint in X and W. The idea is to apply the previously developed strategy in each time step to the time-dependent control u.

Definition 1 Letting $x = [y^{\mathsf{T}}, u^{\mathsf{T}}]^{\mathsf{T}} \in X$ and $S \in \mathbb{N}$, with $S \leq l$, we split up $u \in \mathbb{R}^{n_t \cdot l}$ into $u = (u_1^{\mathsf{T}}, \dots, u_{n_t}^{\mathsf{T}})^{\mathsf{T}}$ with $u_i \in \mathbb{R}^l$ being the control coefficients representing the i-

the time-step. We then apply the *smart rounding* introduced in [9, Definition 3.4] to every u_i, that is

- for $i = 1, \dots, n_t$:
  - Let $u_{S,i} \in \mathbb{R}^S$ denote the S largest components of u_i.
  - Define $\left[u_i\right]_{SR}$ by rounding $u_{S,i}$ component-wise to the closest integer and set the remaining components to 0.
- Define $[u]_{SR} := \left(\left[u_1\right]_{SR}^{\top}, \dots, \left[u_{n_t}\right]_{SR}^{\top} \right)^{\top}$.
- Define $[x]_{SR} := \left(f([u]_{SR})^{\top}, [u]_{SR}^{\top}\right)^{\top} \in W$.

We illustrate this rounding concept by the following simple example involving only control values. We will see that the smart rounding does, by definition, satisfy the knapsack constraint, while the usual rounding to the closest integer may fail to do so.

Example 1 Let $S = 2$, $l = 3$, $n_t = 2$, and let $[\cdot]$ denote the usual rounding to the closest integer. Then, for

$$v = (0.8, 0.7, 0.1, 0.3, 0.6, 0.9)^{\top} \quad \text{and} \quad w = (0.63, 0.62, 0.61, 0.3, 0.6, 0.9)^{\top}$$

it is $v_1 = (0.8, 0.7, 0.1)^{\top}$ and $v_2 = (0.3, 0.6, 0.9)^{\top}$ such that

$$[v]_{SR} = ([v_1]_{SR}^{\top}, [v_2]_{SR}^{\top})^{\top} = (1, 1, 0, 0, 1, 1)^{\top} = [v],$$

but with $w_1 = (0.63, 0.62, 0.61)^{\top}$ and $w_2 = (0.3, 0.6, 0.9)^{\top}$ it is

$$[w]_{SR} = ([w_1]_{SR}^{\top}, [w_2]_{SR}^{\top})^{\top} = (1, 1, 0, 0, 1, 1)^{\top} \neq [w] = (1, 1, 1, 0, 1, 1)^{\top}.$$

In [9], the starting point for the development of the IPA was an exact penalty (EXP) algorithm initially reported in [11]. Such an EXP algorithm can, analogously to [9], be formulated for the time-dependent setting presented in this article. Furthermore, the convergence property for this EXP algorithm is analogous to the one derived in [9, Prop. 3.6], where the only necessary theoretical update is an equivalent of [9, Prop. 3.5] for the time-dependent setting. This can easily be obtained: the first half of the the proof of [9, Prop. 3.5] directly carries over and in the second half, when arguing how any $\tilde{z} \in W$ can be obtained from $\bar{z} \in W$, one has to consider additional cases due to the fact that, in the time-dependent setting, the knapsack constraint might be satisfied as an equality in some timesteps and as an inequality in some other timesteps.

Since the IPA slightly deviates from the EXP algorithm (such that the convergence properties do not directly apply to, but rather support the IPA) and to keep the manuscript length healthy, we decided to spare the details regarding the EXP algorithm and its convergence property. Instead, we adapt the IPA to the time-dependent setting in the following section, review its properties and discuss potential perturbation strategies.

4.1 The algorithm and its details

The key idea of the IPA was to suitably adapt the framework of the EXP algorithm reported in [9, Algorithm 3.1] and originally developed in [11]. The EXP algorithm repeatedly solves the penalty formulation (Ppen) and provides a theoretical framework that tell us when to increases the amount of penalization in the objective function and when to search for a better minimizer. In order to obtain its theoretical convergence properties, the EXP algorithm, at each iteration, requires the use of a global optimization solver. This makes the algorithm impractical in a large-scale PDE constrained optimization setting. As a consequence, the IPA employed one main change: the next iterate in the IPA only has to reduce the objective function (in the EXP algorithm, it had to be a global minimum up to a tolerance δ). This next iterate is searched for via an a probabilistic approach combining a tailored local search strategy with a perturbation of the current iterate (see Sub-Algorithm 1.a below). For the sake of completeness, we report the *time-dependent improved penalty algorithm* (tIPA), i.e., the combination of Algorithms 1 and 1.a, where the time-dependent nature lies in the smart rounding introduced in Definition 1 as well as the underlying model problems (P) and (Ppen) with their feasible sets *W* and *X*, respectively.

Algorithm 1 tIPA($x^0 \in X$, $\varepsilon^0 > 0$, $\sigma \in (0, 1)$, $p_{\max} \in \mathbb{N}$)

```
$n = 0$, $x^n = x^0$, $\varepsilon^n = \varepsilon^0$
Step 1. Call Algorithm 1.a($x^n$, $p_{\max}$, $\varepsilon^n$) to generate a new iterate $x^{n+1}$.
Step 2.
if $x^{n+1} \notin W$ and $J(x^{n+1}; \varepsilon^n) - J([x^{n+1}]_{SR}; \varepsilon^n) \leq \varepsilon^n \left\|x^{n+1} - [x^{n+1}]_{SR}\right\|_2$ then
    $\varepsilon^{n+1} = \sigma\varepsilon^n$
else
    $\varepsilon^{n+1} = \varepsilon^n$
end if
Step 3.
if $x^n = x^{n+1}$ then
    return $[x^{n+1}]_{SR}$
else
    Set $n = n + 1$ and go to Step 1.
end if
```

Algorithm 1.a Reduction via perturbation($x \in X$, $p_{\max} \in \mathbb{N}$, $\varepsilon > 0$)

```
1: $x^{init} = x$
2: for $j = 1, \dots, p_{\max}$ do
3:     Use a local optimization solver to determine a solution $x^{loc}$ of (Ppen) for $\varepsilon$ using
       $x^{init}$ as initial guess.
4:     if $J(x^{loc}; \varepsilon) < J(x; \varepsilon)$ then
5:         return $x^{loc}$
6:     else
7:         Generate a point $x^{pert} = \text{Perturbation}(x^{loc})$ and set $x^{init} = x^{pert}$.
8:     end if
9: end for
10: return $x$
```

In the following, we list some of the key features that the tIPA inherits from the IPA, as they are structurally identical:

- The tIPA terminates via line 11 as soon as the iteration limit $p_{\max}$ is reached inside Algorithm 1.a at Step 1. Thus, the choice of $p_{\max}$ and the perturbation strategy determine the quality of the solution found by the tIPA. Taking into account the scheme of Algorithm 1.a, it is indeed easy to understand that these two features influence the ability to explore the feasible set. A large enough number of iterations should then be used in order to guarantee a proper exploration without significantly increasing the overall CPU time. We will discuss our perturbation strategies in the second part of this section.
- The tIPA is expected to have a two-phase behavior: in the first phase, the penalization is increased due to line 5 of Algorithm 1 until a feasible integer iterate $x^{n+1} \in W$ is found and in this phase the for-loop of Algorithm 1.a should terminate in the first iteration. In the second phase, Algorithm 1.a is then the driving force in finding better points that provide further reductions in the objective function.
- A new iterate is always feasible with $x^{n+1} \in X$. Thus, $x^{n+1} \notin W$ in line 4 of Algorithm 1 can, in a practical implementation, be replaced by

$$\left\| u^{n+1} - [u^{n+1}]_{SR} \right\|_{\infty} > \varepsilon_{feas}$$

  with a feasibility tolerance ε_{feas}. Thus, it is reasonable to return $[x^{n+1}]_{SR}$ such that the control of our output iterate is always integer and satisfies the knapsack constraint.

For a more detailed discussion and interpretation of the algorithm as well as an explanation for the above key features we refer to [9, Section 3.2].

Inside the tIPA, we use the full-IPM developed in Sect. 3.2 to obtain a (local) solution of (P) in Algorithm 1.a. As mentioned in Remark 1, the tIPA can also be formulated for the reduced problem (P_{red}), where the MOR-IPM from Sect. 3.2 is then used inside Algorithm 1.a to obtain a (local) solution of (P_{red}) and we call the resulting algorithm the MOR-tIPA.

The perturbation performed in line 7 plays an important role in the overall strategy. Some tailored perturbation, depending on the problem one intends to solve, might be more beneficial in the end. We hence want to conclude this section with a discussion on the perturbation strategies that we will employ inside Algorithm 1.a during our numerical investigation in the next section. We present two perturbation strategies: the first one being the extension of [9, Algorithm 2.b] to the time-dependent setting, that is we *flip* $\theta \in \mathbb{N}$ many sources in each time step of the control to generate the perturbed control. The corresponding state is then calculated afterwards and the details are described in Algorithm 1.b.

Algorithm 1.b Perturbation($x \in X$, $\theta \in \mathbb{N}, r > 0$)

```
Split $x = (y^\intercal, u^\intercal)^\intercal$ into the state $y \in \mathbb{R}^{n_t \cdot N}$ and control $u = (u_1^\intercal, \dots, u_{n_t}^\intercal)^\intercal \in \mathbb{R}^{n_t \cdot l}$.
  Define $u^{pert} := u$.
for $i = 1, \dots, n_t$ do
    Find $I_S$, the set of indices of the entries of $u_i$ that are larger than $\frac{1}{2}$.
    for $j = 1, \dots, \min\{|I_S|, \theta\}$ do
        Randomly select $\hat{\imath} \in I_S$.
        Define $I^r_{adj}$ the set of indices corresponding to sources adjacent to $\tilde{x}_{\hat{\imath}}$
        Randomly select $\hat{\imath}_{adj} \in I^r_{adj}$.
        Set $\left(u_i^{pert}\right)_{\hat{\imath}}$ to a randomly chosen value smaller than $\frac{1}{2}$.
        Set $\left(u_i^{pert}\right)_{\hat{\imath}_{adj}}$ to a randomly chosen value larger than $\frac{1}{2}$.
        Remove $\hat{\imath}$ from $I_S$.
    end for
end for
Compute $y^{pert} = f(u^{pert})$ if called inside the tIPA or $y^{pert} = f_{\mathrm{red}}(u^{pert})$ if called inside
  the MOR-tIPA.
return $x^{pert} := \left[(y^{pert})^\intercal, (u^{pert})^\intercal\right]^\intercal$
```

When Algorithm 1.b is called inside the tIPA, x is equal to the current iterate x^n. The algorithm then performs $n_t \cdot \theta \in \mathbb{N}$ *flips* to the current control u^n (θ flips per time step of the control), where a flip is one iteration of the inner for-loop of Algorithm 1.b. Before we discuss the second perturbation strategy, we report here our definition of adjacency from [9, Definition 3].

Definition 2 Given a collection of points $x_1, \dots, x_n \in \Omega$ and a radius $r > 0$, we define for a point x_i the set of *adjacent indices*

$$I^r_{adj} := \{j \in \{1, \dots, n\} \mid j \neq i, \left\|x_i - x_j\right\|_\infty \leq r\}.$$

As a result, the set of adjacent indices I^r_{adj} in Algorithm 1.b is obtained via Definition 2 with the centers $\tilde{x}_1, \dots, \tilde{x}_l \in \Omega$ of our source functions as points. Assuming that they are arranged in a uniform $m \times m$ grid, a possible radius might be $r = \frac{1}{m}$.

Algorithm 1.b performs θ flips per time step, which may be disadvantageous: for large n_t the total amount of flips may become very large and since flips are being made in every time step, the resulting perturbation may be too far away from the current iterate to yield a productive initial guess for the local solver in Algorithm 1.a. As a result, the overall perturbation strategy may be unable to find new iterates that improve the objective function. We thus propose a second strategy that simply performs a fixed amount of $\theta \in \mathbb{N}$ flips randomly spread out over the time steps. The details are found in Algorithm 1.c.

Algorithm 1.c Perturbation($x \in X$, $\theta \in \mathbb{N}, r > 0$)

```
Split $x = (y^\top, u^\top)^\top$ into the state $y \in \mathbb{R}^{n_t \cdot N}$ and control $u \in \mathbb{R}^{n_t \cdot l}$. Define $u^{pert} := u$.
Find $I_S$, the set containing the indices of the entries of $u$ that are larger than $\frac{1}{2}$.
for $j = 1, \ldots, \min\{|I_S|, \theta\}$ do
    Randomly select $\hat{\imath} \in I_S$.
    Define $I^r_{adj}$ the set of indices corresponding to sources adjacent to $\tilde{x}_{\hat{\imath}}$.
    Randomly select $\hat{\imath}_{adj} \in I^r_{adj}$.
    Set $\left(u_i^{pert}\right)_{\hat{\imath}}$ to a randomly chosen value smaller than $\frac{1}{2}$.
    Set $\left(u_i^{pert}\right)_{\hat{\imath}_{adj}}$ to a randomly chosen value larger than $\frac{1}{2}$.
    Remove $\hat{\imath}$ from $I_S$.
end for
Compute $y^{pert} = f(u^{pert})$ if called inside the tIPA or $y^{pert} = f_{\mathrm{red}}(u^{pert})$ if called inside
    the MOR-tIPA.
return $x^{pert} := \left[(y^{pert})^\top, (u^{pert})^\top\right]^\top$
```

With Algorithm 1.c one has much better control over the amount of flips resulting in the perturbation x^{pert}. Thus, the hope is to find a balanced θ such that the resulting perturbations lie outside the current basin of attraction of the objective functional and therefore are a qualitative initial guess for the local solver in Algorithm 1.a, resulting in a point with a potentially better objective function value. We note that with the notion of adjacency from Definition 2 the output of Algorithm 1.c does again satisfy the knapsack constraint.

Although the perturbation strategies presented depend on the uniform grid of source centers used to determine the index set I^r_{adj}, we want to stress that the underlying concept of this *flipping* does not depend on the chosen modelling as it was outlined in [9, Section 4.1] alongside other details of our implementation that we do not repeat here.

5 Numerical experiments

In this numerical section, we first investigate the effectiveness of the model order reduction as an approximation technique and then we test the tIPA and the MOR-tIPA also in comparison with `cplexmiqp`, the branch-and-bound routine of CPLEX [38] for quadratic mixed integer problems. Before that, we introduce a second model problem based on a convection-diffusion PDE for which most of the numerical tests will also be carried out.

Consider the original optimal control problem (3), but governed by the parabolic convection-diffusion PDE

$$\begin{aligned} \frac{\partial}{\partial t} y(t,x) - \Delta y(t,x) + w(x)\cdot \nabla y(t,x) &= \sum_{i=1}^{l} u^i(t)\chi_i(x), \quad (t,x) \in (0,T)\times \Omega, \\ y(0,x) &= 0, \quad x \in \overline{\Omega}, \end{aligned} \tag{32}$$

with the constant-in-time wind vector $w(x) = (2x_2(1-x_1^2), -2x_1(1-x_2^2))^\intercal$ and piecewise constant source functions $\chi_1, \dots, \chi_l \in L^2(\Omega)$ that have the same height κ as the Gaussian source functions defined in (2). Using Q1 finite elements, while also employing the Streamline Upwind Petrov-Galerkin (SUPG) [39] upwinding scheme as implemented in the `IFISS` software package [40] to discretize (32) and building the relevant finite element matrices, the semidiscretization in space is achieved. Following the approach made in Sect. 2.2, we then obtain the resulting discretized optimal control problem, its continuous relaxation and its penalty formulation such that experiments can be carried out for this model problem as well.

In the following, we refer to this problem as the *convection-diffusion* problem, while we refer to problem (1) as *heat equation* problem. The convection-diffusion problem typically provides more numerical challenges due to the nonsymmetric nature of its discretized stiffness matrix and to the possibly strong convective wind $w(x)$.

5.1 Numerical setting and parameter choices

We present the numerical setting for the experiments including default parameter choices for the algorithms. If different choices are used, it will be mentioned.

We choose $\Omega := [0,1]^2$ as our computational domain, $\Omega_{\text{obs}} := [0.25, 0.5]^2$ as the domain of observation, and [0, 1] as the time horizon. Regarding the source functions, we choose $l = 25$ sources with centers $\tilde{x}_1, \dots, \tilde{x}_l$ being arranged in a uniform 5×5 grid with step size $\frac{1}{6}$ (resulting in a radius $r = \frac{1}{5}$ for Definition 2). For the piecewise constant sources of the convection-diffusion problem the points $\tilde{x}_1, \dots, \tilde{x}_l$ are the centers of the squares $\Omega_1, \dots, \Omega_l \subset \Omega$ that form a uniform decomposition of Ω. The height of the sources is $\kappa = 100$ and the width ω of the Gaussian sources is chosen such that every source takes 5% of its center-value at a neighboring center. We mention that this choice of height and width is motivated by [5, Section 4.2]. The PDE (1) is discretized using uniform piece-wise linear finite elements in space with a step size of 2^{-6} (unless specified otherwise) resulting in $N = 4225$ vertices (the same step size is used for the aforementioned discretization of (32)). For the temporal dimension, we stick to an equidistant grid with $n_t = 40$ timesteps such that the overall problem consists of $N \cdot n_t = 169000$ continuous and $l \cdot n_t = 1000$ integer variables.

Regarding the full-IPM and the MOR-IPM, the outer interior point iteration is stopped as soon as either $\max\{\left\|\xi_p\right\|_2, \|\xi_d\|_2, \|\xi_c\|_2\} \le 10^{-6}$ or the safeguard $\mu \le 10^{-15}$ is triggered. Furthermore, starting from an initial $\mu = 1$ we decrease μ by the factor 0.1 in each outer interior point iteration. The inexactness for the full-IPM is implemented by stopping GMRES when the norm of the unpreconditioned relative residual is below $\eta = \max\{\min\{10^{-4}, \mu\}, 10^{-10}\}$, while for the MOR-IPM we always use $\eta = 10^{-10}$. Finally, the diagonal block $-\frac{2}{\varepsilon} I_l + \Theta_u$ in either Newton system (30) or (29) is kept positive definite by setting any negative values to $\gamma = 10^{-6}$.

Regarding the balanced truncation introduced in Sect. 3.1.1, we solve the Lyapunov equations (16) via the `mess_lyap` routine of the M-M.E.S.S. toolbox [41] for Matlab, using the default setting. This routine computes low-rank approximations $\hat{R}\hat{R}^T \approx P$ and $\hat{L}\hat{L}^T \approx Q$ and we use $\hat{R}$ and $\hat{L}$ which approximate the Cholesky factors R and L, respectively.

Default parameters for the tIPA and MOR-tIPA are $\varepsilon^0 = 10^6$, $\sigma = 0.5$, and the feasibility tolerance $\varepsilon_{feas} = 0.1$. Both algorithms use the respective solution of (Pcont) or (Ppen$_{\mathrm{red}}$) for ε^0 as initial guess. Do note that this is not necessary since both problems, for large enough ε^0, are usually still convex such that any initial guess would be sufficient.

Regarding `cplexmiqp`, we use default options except that we set a time limit of 50 hours and a memory limit of 32000 megabytes for the search tree.

All experiments were conducted on a PC with 32 GB RAM and a QUAD-Core-Processor INTEL-Core-I7-4770 (4x 3400MHz, 8 MB Cache) utilizing Matlab 2021a via which CPLEX 12.9.0 was accessed.

5.2 The experiments

First experiment

In this first experiment we determine a good choice of the reduced dimension r for both model problems in question. Using the M-M.E.S.S. toolbox, we can calculate an approximation to the first (and thus largest) $\tilde{N} < N$ Hankel singular values $\sigma_1, \ldots, \sigma_{\tilde{N}}$. Due to the theoretical investigations in Section 3.1.2, we are interested in the quantity $\Sigma(r) := \sigma_{r+1} + \cdots + \sigma_{\tilde{N}}$ that is the dominant term in the balanced truncation error bounds. Figure 2 depicts $\Sigma(r)$ over a possible reduced dimension $r = 1, \ldots, \tilde{N} - 1$ for both the heat equation and the convection-diffusion problem. We note that this neglects the singular values $\sigma_{\tilde{N}+1}, \ldots, \sigma_N$ but as

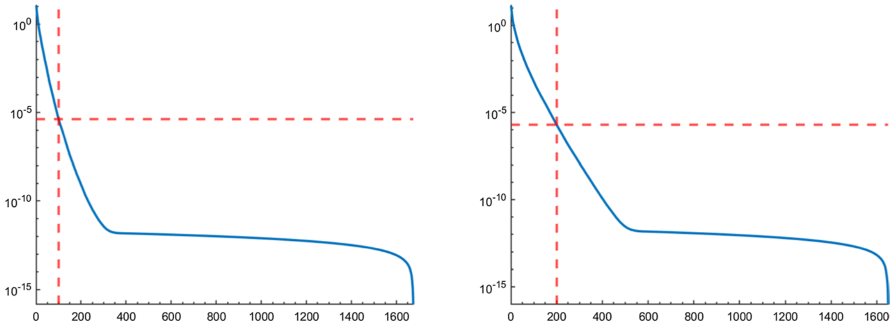

Fig. 2 $\Sigma(r)$ over $r = 1, \ldots, \tilde{N} - 1$ for the heat equation (left) and the convection-diffusion (right) problem. Red dotted lines indicate the chosen reduced dimensions for the later numerical experiments

the Hankel singular values are sorted in descending order it becomes clear from Fig. 2 that it is indeed justified to neglect them.

Based on these calculations, we choose a reduced dimension $r = 100$ for the heat equation problem and, to keep a similar approximation quality, $r = 200$ for the convection-diffusion problem. We stress that even though the convection-diffusion problem requires a larger reduced dimension, the factor of reduction from the full state dimension $N = 4225$ to the reduced dimension 200 is still noticeable. As a result, (Ppen_{red}) consists of $r \cdot n_t = 4000$ continuous variables for the heat equation and 8000 continuous variables for the convection-diffusion problem (compared to the 169000 continuous variables of (Ppen)). As already mention in Sect. 3.1.1 the control and thus the amount of integer variables remains untouched by the chosen MOR approach.

Second experiment Following up on the first experiment, we are now interested in the selection of the reduced dimension r that is required to obtain the approximation quality of $\Sigma(r) \leq 10^{-5}$ if the FEM step size is changing. This is then an indicator on how robust our MOR approach is. We thus calculate the value of r for which $\Sigma(r) \leq 10^{-5}$ for a decreasing FEM step size of $h = 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}$ for both the heat equation and the convection-diffusion problem and the result is depicted in Table 1.

Clearly, the reduced dimension r required for the desired accuracy of the reduced model is robust w.r.t. the FEM step size for the heat equation problem. For the convection-diffusion problem the required reduced dimension does increase. Internal tests showed that this is not due to the convection term (the convection would become more and more challenging for the MOR the smaller the diffusion coefficient would be) but rather due to the piece-wise constant source functions. Since we are still satisfied with the factor of reduction that is achieved, we did not further investigate this matter.

Third experiment We carry out a first comparison of the tIPA and the MOR-tIPA, where we have two aims:

- Observing that both algorithms yield pretty much the same solution. Clearly, there might be slight differences due to the probabilistic nature of the IPA framework, but we want to notice that the MOR approach does not negatively influence the quality of the solution found.
- Investigating the behaviour of the preconditioner inside the full-IPM as well as the MOR-IPM during a tIPA iteration.

Table 1 Results of the second experiment

h	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Heat Eq.	87	93	93	97
Conv-Diff	102	146	172	190

Reduced dimension r such that $\Sigma(r) \leq 10^{-5}$ over a changing FEM step size h for both model problems

To this end, we construct a single problem instance in the following way: we generate a desired state y_d as a solution of (the discretized version of) (1) with $S = 3$ active sources in the right-hand side and the centers of these sources are randomly distributed over $[0.1, 0.9]^2$, where the height and width of these sources coincides with the values presented in Sect. 5.1. In the same fashion, a problem instance is drawn for the convection-diffusion problem.

We now solve each problem instance with the tIPA as well as the MOR-tIPA, where we always use Algorithm 1.b for the perturbation strategy, perturbing $\theta = 1$ source per timestep and limiting the overall perturbation cycle inside Algorithm 1.a to $p_{\max} = 1000$ iterations. We are interested in the number of nonlinear (outer) iterations (NLI) required by IPM and the average number of preconditioned GMRES iterations (aGMRES) for each value of ε visited during the two versions of the IPA algorithm. The result is depicted in Fig. 3.

We have, for the solutions x_{tIPA} and $x_{\text{MOR-tIPA}}$ of the heat equation problem, the objective function values

$$\tilde{J}(x_{\text{tIPA}}) \approx 0.00496, \quad \tilde{J}(x_{\text{MOR-tIPA}}) \approx 0.00495, \quad \text{and}$$
$$\|\tilde{J}(x_{\text{tIPA}}) - \tilde{J}(x_{\text{MOR-tIPA}})\| / \|\tilde{J}(x_{\text{tIPA}})\| \approx 0.00131.$$

For the convection-diffusion problem, we have

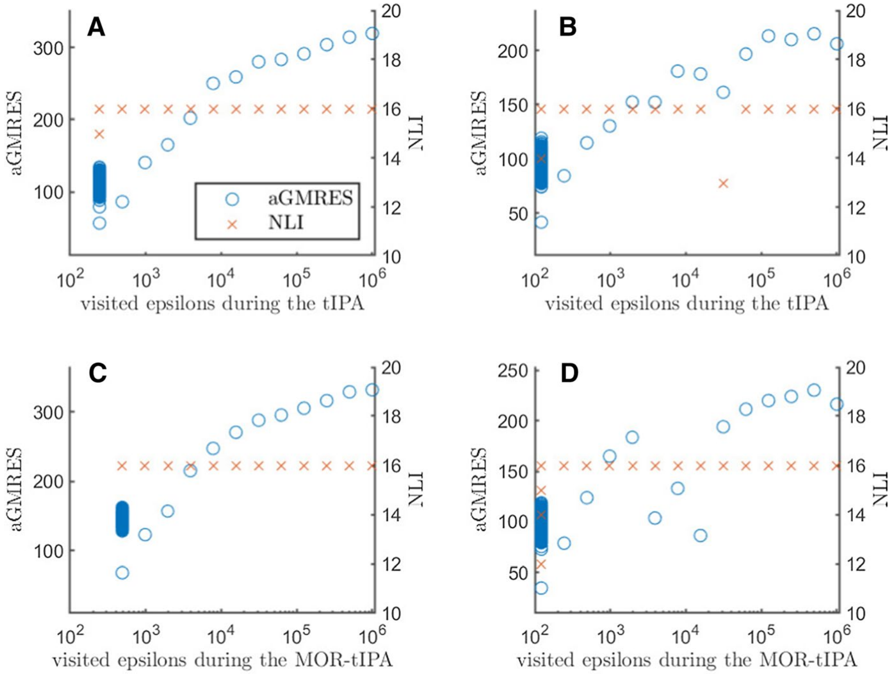

Fig. 3 Number of outer IPM iterations (right y-axis) and average GMRES iterations (left y-axis) during the tIPA for the heat equation (**A**) and convection-diffusion (**B**) problem as well as the MOR-tIPA for the heat equation (**C**) and convection-diffusion (**D**) problem

$$\tilde{J}(x_{\text{tIPA}}) \approx 0.0571, \quad \tilde{J}(x_{\text{MOR-tIPA}}) \approx 0.0556, \quad \text{and}$$
$$\|\tilde{J}(x_{\text{tIPA}}) - \tilde{J}(x_{\text{MOR-tIPA}})\| / \|\tilde{J}(x_{\text{tIPA}})\| \approx 0.0263.$$

Clearly, the obtained solutions for these problem instances are of high quality and the solutions obtained using the MOR-tIPA are even slightly better than the ones obtained with the tIPA.

Regarding the solution time, the tIPA required 55.57 hours for the heat equation and 46.84 hours for the convection-diffusion problem, where the MOR-tIPA only required 3.33 and 11.73 hours, respectively. The increased time in the MOR-tIPA for the convection-diffusion problem is due to the larger dimension of the reduced problem. The results illustrate the efficiency of the MOR approach. Possible improvements made to the preconditioner would lead to a further reduction of the computing time, especially noticeable for the full tIPA. This is backed up by the average GMRES iterations depicted in Fig. 3: while they may be in a reasonable range for smaller values of ϵ (multiple blue circles for a singular value of ε are due to the perturbation step), a still large amount of GMRES iterations is required in the first iterations of both tIPA and MOR-tIPA. The deterioration of the preconditioner is likely due to the Schur-complement approximation ignoring several terms. We believe that an improved approximation of the Schur-complement part of the preconditioner will cause a reduced number of GMRES iterations.

Fourth experiment The perturbation strategy significantly impacts the quality of the overall algorithm. Therefore, we want to determine a qualitative strategy in this experiment. To keep the manuscript length as well as the computational times healthy, this comparison is only carried out using the MOR-tIPA applied to the heat equation problem. We distinguish the following four variants:

- Variant 1 (V1): the perturbation strategy from Algorithm 1.b is used with $\theta = 1$ perturbation per timestep.
- Variants 2-4 (V2-V4): the perturbation strategy from Algorithm 1.c is used with a total of $\theta \in \{\lceil \frac{n_t \cdot S}{20} \rceil, \lceil \frac{n_t \cdot S}{10} \rceil, \lceil \frac{n_t \cdot S}{5} \rceil\}$ many perturbations. Thus, a total amount of 5%, 10%, 20% of the active sources is perturbed.

In each variant, we select $p_{\max} = 1000$ to keep a reasonable balance between computational cost of the overall algorithm and the solution quality (of course, a larger $p_{\max}$ will on average always improve the solution quality due to the probabilistic search approach). For the comparison, we construct a test set of 10 problem instances per value of $S \in \{1, 2, 3, 4, 5\}$ (we described in the previous experiment how such a problem instance is created) and it is clear that with an increased S the combinatorial complexity and thus the difficulty of the MIPDECO problem increases.

We then solve this test set with the algorithms under analysis (the variants of the MOR-tIPA) and compare the results with respect to solution time and quality. For the solution time, we report 't_av' the average solution time and for the solution quality, we choose the following two criteria.

- ’min_count’: for each desired state, we check which algorithm achieved the smallest objective function value. This algorithm is then awarded a score. Surely, multiple algorithms can be awarded a score in the same run (when multiple algorithms find the same ’best’ solution).
- ’rel_err_av’: for each desired state, we store for each algorithm the relative error between the objective function value achieved by that algorithm and the smallest objective function value in that run (the one that was awarded a ’min_count’-score). Only runs resulting in a non-zero relative error are taken into account when computing this average relative error.

Since the global minimum of the tackled optimization problem is not known analytically, the ’min_count’-value tells us how often an algorithm performed best compared to the other algorithms and the average relative error is an additional measure of quality. Furthermore, we collect ’av_subsolvercalls’ the average amount of calls of the local solver to understand how good the perturbation strategy is (the closer this value is to $p_{\max} = 1000$, the less effective the perturbation strategy is). The results of this experiment can be found in Table 2.

The major takeaway from Table 2 is that the second variant (using Algorithm 1.c perturbing a total amount of 5% active sources) is vastly superior to the other variants. Not only is it the fastest variant, but it also has the best solution quality: it has the largest or a very large min_count score and very small average relative error in the instances were it does not produce the best minimizer. Going more into the details, it is very interesting to inspect the last part of Table 2, i.e., av_subsolvercalls. We observe that for both variants 1 and 4 the respective strategy is not actively finding better iterates since the number of calls to the local solver are close to the $p_{\max} = 1000$ iterations of Algorithm 1.a that are required to terminate the overall MOR-tIPA. This strengthens the intuition we already mentioned in Sect. 4.1 that these strategies are flipping too many sources such that the resulting perturbations are useless initial guesses for the local solver (in the sense that they do not lead to better iterates of the overall MIPDECO problem). With strategies V3 and V2 it can then be seen that more subsolvercalls are made on average indicating that the perturbation strategy is actively finding better iterates inside the MOR-tIPA leading to better overall solutions of the MIPDECO problem.

Finally, to put the results of this experiment into a better perspective, Fig. 4 contains, for each part of the test set, a Box-Plot related to the objective function of the final solutions attained by each algorithm (i.e., for each value of S the test set contains 10 instances, such that for each algorithm a Box-Plot is created for the 10 objective function values related to the solutions we found). A Box-Plot consists of several parts: the lower and upper end of box represent the 25th and the 75th percentile of the data vector represented in the respective Box-Plot, the red line inside the box depicts the median of the data and the black dashed lines extending the box are the so called whiskers which represent the remaining data points that are not considered outliers. The outliers are then depicted as red crosses.

Besides showing the absolute values and thus the quality of the points obtained with the MOR-tIPA, the results of Fig. 4 further strengthen the

Table 2 Results of the fourth experiment

	t_av (h)					min_count					rel_err_av (%)					av_subsolvercalls				
S	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
MOR-tIPA V1	1.41	2.09	2.37	2.43	2.24	5	4	4	4	4	17.54	14.64	12.52	8.61	5.45	1014	1015	1014	1014	1014
MOR-tIPA V2	0.72	1.12	1.26	1.22	1.35	7	10	10	8	9	5.75	0.00	0.00	4.58	1.40	1132	1427	1399	1164	1200
MOR-tIPA V3	0.89	1.49	1.74	1.82	1.67	8	4	4	6	4	21.20	11.75	10.05	6.86	5.31	1221	1427	1180	1124	1016
MOR-tIPA V4	0.94	1.36	2.00	2.12	2.32	5	4	4	4	5	15.86	14.64	12.52	8.61	5.29	1017	1015	1014	1014	1075

Comparison of different MOR-tIPA variants for different values of *S*

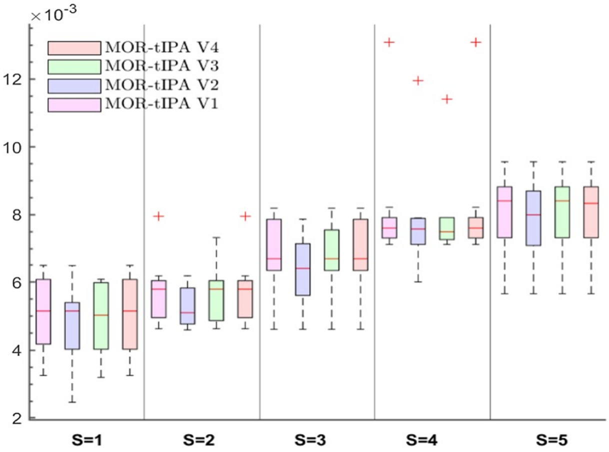

Fig. 4 Results of the fourth experiment: for each part of the test set, a Box-Plot related to the objective function of the final solutions obtained by each algorithm is depicted

observations made in Table 2: variant 2 either achieves the smallest median (for $S = 2$, $S = 3$, and $S = 5$) or found significantly better solutions lying in the lower whiskers than the other variants (for $S = 1$ and $S = 4$).

Fifth experiment In our final experiment, we want to compare the MOR-tIPA, the tIPA, and `cplexmiqp`, the branch-and-bound routine of CPLEX to verify that the MOR-tIPA is indeed the best algorithm for the MIPDECO problems tackled in this article. The experiment is carried out for the heat equation as well as the convection-diffusion problem. We create one problem instance for each $S \in \{1, 2, 3, 4, 5\}$ (where we described in the third experiment how such a problem instance is created) and solve it with `cplexmiqp` given a time limit of 50 hours, as well as the tIPA and the MOR-tIPA, where both algorithms use the perturbation strategy used in variant 2 from the previous experiment. Concerning the computational times of the tIPA, we employ a timelimit of 50 hours to keep a fair comparison with `cplexmiqp`.

Regarding the solution quality, the algorithm with the lowest objective function value is indicated with '**min**' in Table 3 and for each other algorithm the relative error with respect to this objective function value is then displayed. Furthermore, Table 3 contains the running times in hours for each algorithm in each instance, where '**TL**' indicates that the time limit was reached by the given algorithm.

Focusing on this test, we may conclude that the MOR-tIPA and the tIPA do find equally good solutions and the MOR approach does not severely deteriorate the solution quality. Moreover, the MOR-tIPA definitely outperforms the tIPA in terms of computational time. Finally, results clearly show that `cplexmiqp` is not able to find a good solution to the large-scale problems tackled in this article in the prescribed (although large) amount of time.

Table 3 Results of the fifth experiment

S	1		2		3		4		5	
	Time (h)	rel_err (%)	time (h)	rel_err (%)	Time (h)	rel_err (%)	Time (h)	rel_err (%)	Time (h)	rel_err (%)
Heat Eq.										
`cplexmiqp`	**TL**	2266.13	**TL**	4971.64	**TL**	12495.90	**TL**	23032.98	**TL**	30936.74
tIPA	43.58	**min**	31.18	**min**	40.51	**min**	25.85	21.20	26.01	**min**
MOR-tIPA	1.86	13.45	1.65	11.05	1.93	13.25	1.55	**min**	1.95	3.70
Convection-diffusion problem										
`cplexmiqp`	**TL**	1934.08	**TL**	7671.52	**TL**	5366.21	**TL**	3626.17	**TL**	3743.28
tIPA	33.14	**min**	37.71	**min**	39.71	4.87	33.22	4.84	**TL**	**min**
MOR-tIPA	8.47	75.17	10.42	0.04	7.15	**min**	6.28	**min**	10.41	**min**

For each problem instance the algorithm with the lowest objective function value is indicated with '**min**'. The respective relative error of other algorithms as well as the solution times are furthermore reported ('**TL**' indicates that the time limit was reached by the given algorithm)

6 Conclusion and outlook

A standard MIPDECO problem with a linear time-dependent PDE constraint and a modelled control was presented and discretized. An improved penalty algorithm (IPA), developed by the authors in a previous work, was suitably adapted to the time-dependent setting, where the core of the IPA is an efficient local optimization solver paired with a probabilistic basin hopping strategy as well as an updating tool for the penalty parameter. In order to handle the large-scale context of the time-dependent PDE constraint, we introduced a combination of an interior point method (IPM), model order reduction (MOR), and preconditioning resulting in the MOR-IPM. Integrating the MOR-IPM in the time-dependent IPA framework yielded the MOR-tIPA for the solution of the overall MIPDECO problem, which represents the main novelty of this work.

A thorough numerical investigation, dealing with a heat equation as well as a convection-diffusion problem, showed the efficiency of the model order reduction, revealed a promising perturbation strategy inside the IPA framework, and highlighted how efficiently the MOR-tIPA provides significant solutions for the difficult MIPDECO problems considered in this article (and how much `cplexmiqp`, the branch-and-bound routine of CPLEX, struggles).

On the contrary, the numerical investigation also revealed that the developed preconditioner leaves room for improvement and this will have to be considered in future work. Besides this, the next step is the development of an IPA framework (and especially an efficient local solver) for time-dependent nonlinear problems, where devising an effective model order reduction will certainly be a challenging task.

Acknowledgements D. Garmatter and M. Stoll acknowledge the financial support by the Federal Ministry of Education and Research of Germany (support code 05M18OCB). D. Garmatter thanks Dr. Jens Saak for vital discussions on the generalized balanced truncation and the M.E.S.S. toolbox. M. Porcelli is member of the INdAM Research Group GNCS and this work was partially supported by INdAM-GNCS under Progetti di Ricerca 2020-2021.

Funding Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement.

Data Availability The data that support the findings of this study are available from the corresponding author upon request.

References

1. Hahn, M., Leyffer, S., Zavala, V. M.: Mixed-integer PDE-constrained optimal control of gas networks, Argonne National Laboratory, MCS Division Preprint ANL/MCS-P9040-0218, (2017)
2. Pfetsch, M.E., Fügenschuh, A., Geißler, B., Geißler, N., Gollmer, R., Hiller, B., Humpola, J., Koch, T., Lehmann, T., Martin, A., et al.: Validation of nominations in gas network optimization: Models, methods, and solutions. Optimization Method Soft **30**(1), 15–53 (2015)
3. Funke, S., Farrell, P., Piggott, M.: Tidal turbine array optimisation using the adjoint approach. Renewable Energy **63**, 658–673 (2014)
4. Zhang, P.Y., Romero, D.A., Beck, J.C., Amon, C.H.: Solving wind farm lyaout optimization with mixed integer programs and constraint programs. EURO J. Comput. Optim. **2**(3), 195–219 (2014)
5. Wesselhoeft, C.: "Mixed-integer PDE-constrained optimization," Master's thesis, Imperial College London, (2017)
6. Göttlich, S., Potschka, A., Teuber, C.: A partial outer convexification approach to control transmission lines. Comput. Optim. Appl. **72**(2), 431–456 (2019)
7. Tröltzsch, F.: Optimal control of partial differential equations: theory, methods, and applications. American Mathematical Soc. **112**, 23 (2010)
8. Leugering, G., Engell, S., Griewank, A., Hinze, M., Rannacher, R., Schulz, V., Ulbrich, M., Ulbrich, S.: Constrained optimization and optimal control for partial differential equations. Springer Sci. Business Media **160**, 11 (2012)
9. Garmatter, D., Porcelli, M., Rinaldi, F., Stoll, M.: Improved penalty algorithm for mixed integer PDE constrained optimization (MIPDECO) problems. Comput. Math. Appl. **116**, 2–14 (2022)
10. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. Acta Numerica **22**, 1–131 (2013)
11. Lucidi, S., Rinaldi, F.: An exact penalty global optimization approach for mixed-integer programming problems. Optim. Lett. **7**(2), 297–307 (2013)
12. Grosso, A., Locatelli, M., Schoen, F.: A population-based approach for hard global optimization problems based on dissimilarity measures. Mathemat. Program. **110**(2), 373–404 (2007)
13. Leary, R.H.: Global optimization on funneling landscapes. J. Global Optim. **18**(4), 367–383 (2000)
14. Antoulas, A.C.: Approximation of large-scale dynamical systems. SIAM, New Delhi (2005)
15. Gubisch, M., Volkwein, S.: Proper orthogonal decomposition for linear-quadratic optimal control. Model reduction and approximation: theory and algorithms **5**, 66 (2017)
16. De Los Reyes, J.C., Stykel, T.: A balanced truncation-based strategy for optimal control of evolution problems. Optim. Method. Soft. **26**(4–5), 671–692 (2011)
17. Dihlmann, M.A., Haasdonk, B.: Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. Comput. Optim. Appl. **60**(3), 753–787 (2015)
18. Antil, H., Heinkenschloss, M., Hoppe, R.H.: Domain decomposition and balanced truncation model reduction for shape optimization of the stokes system. Optim. Meth. Soft. **26**(4–5), 643–669 (2011)
19. Freya, B., Dennis, B., Jianjie, L., Stefan, V.: POD-based mixedinteger optimal control of the heat equation. J. Sci. Comput. **81**(1), 48–75 (2019)
20. Elman, H.C., Forstall, V.: Preconditioning techniques for reduced basis methods for parameterized elliptic partial differential equations. SIAM J. Sci. Comput. **37**(5), S177–S194 (2015)
21. Singh, N.P., Ahuja, K.: Preconditioned linear solves for parametric model order reduction. Int. J. Comput. Mathemat. **97**(7), 1484–1502 (2020)
22. Manns, P., Kirches, C.: Multi-dimensional sum-up rounding for elliptic control systems. SIAM J. Num. Analysis **58**(6), 3427–3447 (2020)
23. Leyffer, S., Manns, P., Winckler, M.: Convergence of sum-up rounding schemes for cloaking problems governed by the helmholtz equation. Comput. Optim. Appl. **79**(1), 193–221 (2021)
24. Larson, J., Leyffer, S., Palkar, P., Wild, S.M.: A method for convex black-box integer global optimization. J. Global Optimi. **1**, 1–39 (2021)
25. Sharma, M., Hahn, M., Leyffer, S., Ruthotto, L., van Bloemen Waanders, B.: Inversion of convection-diffusion equation with discrete sources. Optim. Eng. **1**, 1–39 (2020)
26. Lucidi, S., Rinaldi, F.: Exact penalty functions for nonlinear integer programming problems. J. Optim. Theory Appl. **145**(3), 479–488 (2010)
27. Rinaldi, F.: New results on the equivalence between zero-one programming and continuous concave programming. Optim. Lett. **3**(3), 377–386 (2009)

28. Saak, J.: “Efficient numerical solution of large scale algebraic matrix equations in PDE control and model order reduction,” PhD thesis, (2009)
29. Badia, J. M., Benner, P., Mayo, R., Quintana-Orti, E. S., QuintanaOrti, G., Remón, A.: “Balanced truncation model reduction of large and sparse generalized linear systems,” Chemnitz Scientific Computing Preprints, pp. 06-04, (2006)
30. Benner, P., Sachs, E., Volkwein, S.: “Model order reduction for PDE constrained optimization,” Trends in PDE constrained optimization, pp. 303- 326, (2014)
31. Antil, H., Heinkenschloss, M., Hoppe, R.H., Sorensen, D.C.: Domain decomposition and model reduction for the numerical solution of PDE constrained optimization problems with localized optimization variables. Comput. Visual. Sci. **13**(6), 249–264 (2010)
32. Gondzio, J.: Interior point methods 25 years later. Eu. J. Operat. Res. **218**(3), 587–601 (2012)
33. Nocedal, J., Wright, S.J. (eds.): Numerical Optimization. SpringerVerlag, Berlin (1999)
34. Bellavia, S.: Inexact interior-point method. J. Optim. Theory Appl. **96**(1), 109–121 (1998)
35. Pearson, J.W., Porcelli, M., Stoll, M.: Interior-point methods and preconditioning for PDE-constrained optimization problems involving sparsity terms. Numerical Linear Algebra Appl. **27**, 2 (2020)
36. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statistical Comput. **7**(3), 856–869 (1986)
37. Pearson, J.W., Stoll, M., Wathen, A.J.: Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. SIAM J. Matrix Analysis Appl. **33**(4), 1126–1152 (2012)
38. IBM ILOG CPLEX, https://www.ibm.com/analytics/cplex-optimizer
39. Brooks, A.N., Hughes, T.J.: Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. Comput. Meth. Appl. Mech. Eng. **32**(1–3), 199–259 (1982)
40. Elman, H.C., Ramage, A., Silvester, D.J.: Algorithm 866: IFISS, a matlab toolbox for modelling incompressible flow. ACM Transactions on Mathematical Software (TOMS) **33**(2), 14 (2007)
41. Saak, J., Köhler, M., Benner, P.: M-M.E.S.S.-2.1 - the matrix equations sparse solvers library, see also: https://www.mpi-magdeburg.mpg.de/projects/mess,Apr. (2021). https://doi.org/10.5281/zenodo.4719688