COMBINATORIAL OPTIMAL CONTROL OF SEMILINEAR ELLIPTIC PDES

CHRISTOPH BUCHHEIM*, CHRISTIAN MEYER†, AND RENKE SCHÄFER‡

Abstract. Optimal control problems (OCP) containing both integrality and partial differential equation (PDE) constraints are very challenging in practice. The most wide-spread solution approach is to first discretize the problem, resulting in huge nonlinear mixed-integer optimization problems that can be solved to proven optimality only in very small dimensions. In this paper, we propose a novel outer approximation approach to efficiently solve such OCPs in the case of certain semilinear elliptic PDEs with static integer controls over arbitrary combinatorial structures. The basic idea is to decompose the OCP into an integer linear programming (ILP) master problem and a subproblem for calculating linear cutting planes. These cutting planes rely on the pointwise concavity of the PDE solution operator in terms of the control variables, which we prove in the case of PDEs with a non-decreasing convex nonlinear part. The decomposition allows to use standard solution techniques for ILPs as well as for PDEs. We further benefit from reoptimization strategies due to the iterative structure of the algorithm. Experimental results show that the new approach is capable of solving the combinatorial OCP of a semilinear Poisson equation with up to 230 binary controls to global optimality within a 5h time limit. Applied to the screened Poisson equation, problems with even 2200 binary controls are globally solvable.

Key words. Optimal Control, Partial Differential Equations, Outer Approximation, Integer Nonlinear Programming

AMS subject classifications. 49J20, 90C10

1. Introduction. Optimal control is the optimization of a system described by partial or ordinary differential equations (PDEs/ODEs) over a control input. In a broad range of applications, all or some of the control variables have to be considered discrete, e.g. motor- or gear-switches in automotive engineering [6, 24], state transitions or feed locations in chemical engineering [2, 4] or – in case of PDEs – placements of wind turbines in a wind park [35] or switches for valves or compressors in gas or water networks [14, 16]. Consequently, the demand for efficient algorithms to address optimal control problems with (partly) discrete controls, often referred to as mixed-integer optimal control problems (MIOCP), mixed-integer dynamic optimization (MIDO), or hybrid optimal control problems (HOCP), is very high. Most approaches discussed in the literature consider applications where the discrete variables are dynamic, i.e. depend on time or space, while their number remains low and no complicating combinatorial constraints are taken into account. As a typical example, in the case of gear-switches, a single dynamic integer variable is considered. In this paper, we address a different class of applications: we assume that the discrete controls are static but many, and subject to combinatorial constraints that may render the problem hard even in the absence of differential equations.

The most straightforward and widely used approach to address MIOCPs is to *first-discretize-then-optimize*. The basic idea is to discretize the control and, if desired, the state of the dynamic process in time or space, in order to approximate the MIOCP by a finite-dimensional mixed-integer nonlinear programming problem (MINLP) and then use standard techniques for solving the latter; see [5] for a recent survey on algorithms for MINLP. Although specific MIOCPs have been successfully solved to

*Fakultät für Mathematik, TU Dortmund, Germany. christoph.buchheim@tu-dortmund.de
†Fakultät für Mathematik, TU Dortmund, Germany. christian.meyer@math.tu-dortmund.de
‡Fakultät für Mathematik, TU Dortmund, Germany. renke.schaefer@math.tu-dortmund.de

(global) optimality by direct methods [17, 34, 2, 3, 13, 27], the discretization approach often fails if applied to more general problem classes [32]. Other common methods from optimal control theory, e.g. dynamic programming, have similar shortcomings [8].

As a consequence, various numerical methods have been developed to quickly compute feasible, but suboptimal solutions. The most prominent heuristic is the Sum-Up Rounding strategy [30]. It is capable of finding a feasible mixed-integer solution constructed out of an integral-relaxed NLP solution, the latter being obtained by a direct method, so that the relaxed and rounded state are arbitrarily close (depending on the OCP discretization). Sum-Up Rounding can also be applied to time-dependent controls in MIOCPs with PDE constraints [20]. As general combinatorial constraints cannot be considered within the rounding scheme, the Combinatorial Integral Approximation (with a focus on restrictions on the number of switches) has been proposed [31], where the relaxed control is tracked by an integer control. It leads to a quickly solvable mixed-integer linear programming problem (MILP) and may serve as a first upper bound for other MINLP solution methods. However, both Sum-Up Rounding and Combinatorial Integral Approximation are designed to address time-dependent discrete controls and cannot handle static controls. Thus, MIOCPs with static controls and PDE constraints are usually handled differently in the literature: either by concentrating on linear PDEs only [11] or by linearization [14, 16].

As pointed out above, mixed-integer optimal control with static discrete controls and combinatorial as well as PDE constraints is an open field of research, especially if it comes to global solvers. In this paper, we consider a problem with integer decisions u that define a linear cost function $c^\top u$ to be minimized, subject to combinatorial constraints $u \in \mathcal{U}$. We further require the state y of a semilinear elliptic PDE (depending on u) to reach a given reference state $y_{\min}$. The problem can be written as

$$\text{(COCP)} \qquad \begin{cases} \min & c^\top u & & \\ \text{s.t.} & y(x) \geq y_{\min}(x) & & \text{a.e. in } \Omega \\ & Ay + g(y) = \sum_{i=1}^{\ell} u_i \, \psi_i & & \text{in } \Omega \\ & \frac{\partial y}{\partial n_A} + b(y) = \sum_{j=\ell+1}^{n} u_j \, \phi_j & & \text{on } \Gamma_N \\ & y = 0 & & \text{on } \Gamma_D \\ \text{and} & u \in \mathcal{U}. & & \end{cases}$$

Herein Ω denotes a bounded domain $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, and Γ_D and Γ_N are disjoint parts of its boundary such that $\Gamma_D \cup \Gamma_N = \partial\Omega$. Moreover, A is a linear, elliptic operator, and $\partial/\partial n_A$ denotes the co-normal derivative associated with A. In addition, g and b denote Nemyzki operators associated with nonlinear functions. The functions ψ_i, $i = 1, \ldots, \ell$, and ϕ_j, $j = \ell+1, \ldots, n$, are given and will be called form functions in all what follows. Finally, $\mathcal{U} \subseteq \mathbb{Z}^n$ is a bounded set of (discrete) admissible controls. The precise assumptions on the data and quantities in (COCP) are formulated in Section 2, where we also give an application example. Generally speaking, Problem (COCP) can model applications in areas where the optimization of a static diffusion process is desired, subject to a given minimum state. Our algorithmic approach employs the special structure of (COCP), in particular the state constraints $y \geq y_{\min}$. In the context of classical optimal control problems without integrality constraints, pointwise state constraints of this form are known to cause severe difficulties from a theoretical as well as a numerical point of view; see [9, 1, 21, 12, 25] and the references therein. These difficulties are mainly caused by the poor regularity of the Lagrange multipliers

associated with the state constraints, which are only Borel measures in general, see [10] and [26, Section 6.2]. By contrast, in our setting we benefit from the pointwise state constraints, as our algorithmic approach exploits the particular problem structure induced by these constraints.

From the discrete point of view, (COCP) is a nonlinear combinatorial optimization problem: the objective is to minimize a linear function over the combinatorial variables $u \in \mathcal{U}$, where the state variables y implicitly define an infinite number of additional nonlinear constraints on the feasible set. Under the assumptions listed in Section 2, in particular the convexity of the functions $g(x,.)$ and $b(x,.)$ for almost all x, we are able to show that the latter constraints actually define a convex set, and derive valid linear cutting planes in the discrete controls u; see Section 3. These cutting planes form the basis of an outer approximation algorithm for Problem (COCP) devised in Section 4. Due to the iterative structure of the algorithm, we apply reoptimization strategies to efficiently resolve the PDE for updated candidate solutions u; see Section 5.

Finally, in Section 6 we perform extensive numerical experiments to demonstrate the benefits of our algorithm and its dependence on the problem parameters. It turns out that our new approach is capable of solving the combinatorial OCP of a semilinear Poisson equation with up to 200 binary controls to global optimality within a 5h time limit.

2. Standing Assumptions and Known Results. We start with the precise assumptions on the data and quantities in (COCP). Throughout the paper $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, denotes a bounded domain, i.e. bounded, open, and connected set, with regular boundary $\Gamma = \partial\Omega$. For the precise definition of a regular boundary we refer to [15, Def. 1.17, Lem. 1.27]. Furthermore, Γ_N and Γ_D are disjoint parts of Γ such that $\Gamma = \Gamma_D \cup \Gamma_N$. We define the space V as the linear subset of $H^1(\Omega)$ given by

$$V = \{v \in H^1(\Omega) : v = 0 \text{ a.e. on } \Gamma_D\}$$

equipped with the standard H^1-norm. Furthermore, V^* denotes its dual space. The operator $A : V \to V^*$ is given by the following linear elliptic differential operator of second order

$$Ay = -\sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}\Big(a_{ij}(x)\frac{\partial}{\partial x_j}y(x)\Big) + \sum_{i=1}^{d} \beta_i(x)\frac{\partial}{\partial x_i}y(x) + a_0(x)y(x),$$

where $a_{ij}, \beta_k, a_0 \in L^\infty(\Omega)$, $i,j,k = 1,\dots,d$, are such that A is coercive on V, i.e., we have

$$\langle Av,\, v\rangle_{V^*,V} \geq \alpha \,\|v\|_V^2 \quad \forall\, v \in V \tag{2.1}$$

for some constant $\alpha > 0$. To keep the discussion concise, we restrict ourselves to coercive bilinear forms, satisfying (2.1). Depending on the particular structure of the nonlinearities, this assumptions can be weakened, see Example 1 below. By $\partial/\partial n_A$ we denote the co-normal derivative associated with A, i.e.,

$$\frac{\partial y}{\partial n_A} = \sum_{i,j=1}^{n} n_i\, a_{ij}\, \frac{\partial y}{\partial x_j},$$

where $n : \Gamma \to \mathbb{R}^d$ is the outward unit normal on Γ. Moreover, we require that the nonlinear functions $g : \Omega \times \mathbb{R} \to \mathbb{R}$ and $b : \Gamma_N \times \mathbb{R} \to \mathbb{R}$ satisfy the following conditions:

1. Both g and b satisfy the Carathéodory condition, i.e., $g(.,y)$ is measurable for every fixed $y \in \mathbb{R}$ and $g(x,.)$ is continuous for almost every $x \in \Omega$, and analogously for b.
2. Both $g(x,.)$ and $b(x,.)$ are non-decreasing for almost every $x \in \Omega$ and almost every $x \in \Gamma_N$, respectively.
3. The mappings $g(x,.)$ and $b(x,.)$ are differentiable for almost every $x \in \Omega$ and almost every $x \in \Gamma_N$, respectively. Their derivatives are denoted by $g'(x,.)$ and $b'(x,.)$ and are assumed to satisfy the Carathéodory assumption as well.

The form functions ψ and ϕ satisfy $\psi_i \in L^r(\Omega)$ with $r > d/2$ for all $i = 1, \dots, \ell$ and $\phi_j \in L^s(\Gamma_N)$ with $s > d-1$ for all $j = \ell+1, \dots, n$.

We are now in the position to introduce the notion of weak solutions to the PDE appearing in (COCP). For this purpose let us define the space

$$Y := V \cap L^\infty(\Omega).$$

A function $y \in Y$ is said to be a weak solution if it satisfies

$$\begin{aligned} &\langle Ay,\, v\rangle_{V^*,V} + \int_\Omega g(x,y)\, v\, dx + \int_{\Gamma_N} b(x,y)\, v\, ds \\ &\qquad = \sum_{i=1}^{\ell} \int_\Omega \psi_i\, v\, dx\, u_i + \sum_{j=\ell+1}^{n} \int_{\Gamma_N} \phi_j\, v\, dx\, u_j \quad \forall\, v \in V. \end{aligned} \tag{2.2}$$

Proposition 2.1.

(i) For every $u \in \mathbb{R}^n$ there exists a unique weak solution $y \in Y$ of (2.2). We denote the associated solution operator by $S : \mathbb{R}^n \to Y$.

(ii) The operator S is continuously Fréchet differentiable from $\mathbb{R}^n$ to Y and its derivative $\eta = S'(u)h$ in direction $h \in \mathbb{R}^n$ is given by the solution of the linearized PDE

$$\begin{aligned} &\langle A\eta,\, v\rangle_{V^*,V} + \int_\Omega g'(x,y)\eta\, v\, dx + \int_{\Gamma_N} b'(x,y)\eta\, v\, ds \\ &\qquad = \sum_{i=1}^{\ell} \int_\Omega \psi_i\, v\, dx\, h_i + \sum_{j=\ell+1}^{n} \int_{\Gamma_N} \phi_j\, v\, dx\, h_j \quad \forall\, v \in V\ . \end{aligned} \tag{2.3}$$

Proof. The proof is standard. Nevertheless we shortly recall the main arguments for convenience of the reader.

(i) First one shows by means of the Browder-Minty-theorem that there is a unique solution to the weak formulation provided that the nonlinearities g and b are truncated. Then the well-known Stampacchia technique yields an L^∞-estimate for the solutions of the truncated problems, which is independent of the truncation thresholds [23, Chapter II, Appendix B]. The positivity of the trace operator yields the same L^∞-bound for the trace; see [23, Prop. 5.2]. By setting the truncation thresholds larger than the L^∞-bound we find a solution of the original problem. Uniqueness follows from coercivity of A. For details we refer to [33, Section 4.2].

(ii) Due to the monotonicity of g and b and the coercivity of A, the left hand side of (2.3) defines a coercive and bounded bilinear form on V for every $y \in L^\infty(\Omega)$ so that the Lax-Milgram lemma gives the unique existence of solutions to (2.3). The boundedness of η and its trace again follows from the Stampacchia argument and the positivity of the trace. Since the Nemyzkii operators associated with g and b

are continuously Fréchet-differentiable in $L^\infty(\Omega)$ and $L^\infty(\Gamma_N)$, respectively [18], the implicit function theorem gives the result, see [29, Theorem 2.9]. For details we refer to [33, Section 4.5]. □

REMARK 1. *Under mild additional assumptions on the problem data, in particular the coefficient function a_{ij} and the domain, it can be shown that the state is even continuous and the same holds true for the linearized state, i.e., the solution of* (2.3)*, see [19]. However, as the continuity of the state is not mandatory for our algorithmic approach, we do not impose these additional assumptions. We point out that no additional regularity assumptions on the form functions are needed for this continuity result.*

From the remaining quantities in (COCP) we require the following: the set $\mathcal{U}$ is assumed to be bounded and given by an integer linear description

$$\mathcal{U} = \{u \in \mathbb{Z}^n : Gu \leq h\}$$

with $G \in \mathbb{R}^{m\times n}$ and $h \in \mathbb{R}^m$, for some $m \in \mathbb{N}$, while the vector c in the objective function is an arbitrary vector in $\mathbb{R}^n$. The reference state is supposed to satisfy $y_{\min} \in L^1(\Omega)$, and we assume that the feasible set is non-empty, i.e., there is at least one control vector $u \in \mathcal{U}$ such that $S(u) \geq y_{\min}$ a.e. in Ω.

For the computation of global lower bounds we additionally require

ASSUMPTION 2.2. *There exist numbers $y_a, y_b \in [-\infty, \infty]$, $y_a \leq y_b$, with*

$$S(u)(x) \in [y_a, y_b] \quad \text{a.e. in } \Omega \quad \forall\, u \in \operatorname{conv}(\mathcal{U}),$$

such that the functions $g(x,.)\colon [y_a, y_b] \to \mathbb{R}$ and $b(x,.)\colon [y_a, y_b] \to \mathbb{R}$ are convex for almost all $x \in \Omega$ and almost all $x \in \Gamma_N$, respectively.

While the above assumptions on g and b concerning their monotonicity and their differentiability are quite standard for the discussion of semilinear elliptic PDEs in the context of optimal control, Assumption 2.2 is fairly restrictive. Nevertheless, the following example shows that there are application driven problems where this assumption is satisfied.

EXAMPLE 1. *We consider the stationary heating of a metallic workpiece. If the workpiece is assumed to be homogeneous and isotropic, the operator A is given by*

$$A = -\kappa\Delta = -\kappa \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2} \ ,$$

where $\kappa > 0$ denotes the (constant) heat conductivity of the material. If the material is heated up to higher temperatures, radiation has to be taken into account. This leads to Boltzmann type radiation boundary conditions of the form

$$\kappa \nabla y \cdot n + \sigma\, |y|^d y = \sigma\, y_0^{d+1}, \tag{2.4}$$

where $\sigma > 0$ denotes the Boltzmann radiation constant for the particular dimension d and $y_0 > 0$ is a fixed external temperature. The boundary condition (2.4) *models the radiation of an ideal black radiator, see [22] for details. Note that the coercivity assumption in* (2.1) *is not satisfied in this example. However, this assumption is only needed to ensure existence and uniqueness of solutions, which, in case of this example, easily follows from the particular structure of the nonlinearity in* (2.4) *as the derivative of $\|.\|^{d+2}_{L^{d+2}(\Gamma)}$.*

If we assume that the workpiece is heated up by $n \in \mathbb{N}$ fixed volume sources $\psi_1, \dots, \psi_n$, generated for instance by induction heating, then the PDE in strong form reads

$$\begin{aligned} -\kappa \Delta y &= \sum_{i=1}^{n} u_i \psi_i \quad \text{in } \Omega \\ \kappa \nabla y \cdot n + \sigma\, |y|^d y &= \sigma\, y_0^{d+1} \qquad \text{on } \Gamma. \end{aligned}$$

Herein the discrete control variables $u \in \mathcal{U} := \{0,1\}^n$ model the switching of the heat sources. By setting $g \equiv 0$, $\Gamma_D = \emptyset$, $\Gamma_N = \Gamma$, and $b = \sigma(|y|^d y - y_0^{d+1})$, this problem fits into our general setting. If we focus on pure heating processes, then we may assume that $\psi_i(x) \geq 0$ a.e. in Ω for all $i = 1, \dots, n$. Consequently, the (weak) maximum principle gives $S(u)(x) \geq 0$ a.e. in Ω for all $u \in \mathcal{U}$. By the positivity of the trace operator, we obtain $S(u)(x) \geq 0$ a.e. on Γ, and thus

$$b(S(u)(x)) = \sigma\big([S(u)(x)]^{d+1} - y_0^{d+1}\big) \quad \text{a.e. on } \Gamma.$$

Thus Assumption 2.2 is satisfied in this example. If the vector $c \in \mathbb{R}^n$ in the objective function measures the cost of each source function ψ_i, e.g. in terms of energy consumption, then every solution of (COCP) *yields a most efficient way of switching on the sources in order to pointwisely keep the temperature at a desired minimal temperature $y_{\min}$. This is of interest for the optimization of hardening processes of steel workpieces, where it is essential to pointwisely reach the austenitic temperature.*

3. Computation of Cutting Planes. In all what follows we denote by $\max(.,0)$ the function $\mathbb{R} \ni r \mapsto \max\{r, 0\} \in \mathbb{R}$, and the associated Nemyzkii operators in $H^1(\Omega)$ and $L^2(\Gamma)$, respectively, are denoted in the same way for the sake of convenience.

LEMMA 3.1. *For every $v \in H^1(\Omega)$ we have $\tau \max(v,0) = \max(\tau v, 0)$ a.e. on Γ, where $\tau : H^1(\Omega) \to L^2(\Gamma)$ denotes the trace operator.*

Proof. As $\Gamma = \partial\Omega$ is regular, the set $\mathcal{C}(\bar{\Omega}) := \{\varphi|_\Omega : \varphi \in C_0^\infty(\mathbb{R}^d)\}$ is dense in $H^1(\Omega)$ by [15, Lemma 1.30]. Thus $v \in H^1(\Omega)$ can be approximated by a sequence $\{v_n\} \subset \mathcal{C}(\bar{\Omega})$. Then the continuity of $\max(.,0)$ in $H^1(\Omega)$ and $L^2(\Gamma)$, respectively, and the one of the trace $\tau : H^1(\Omega) \to L^2(\Gamma)$ imply

$$\begin{aligned} \tau \max(v,0) = \tau \max\big(\lim_{n\to\infty} v_n, 0\big) &= \lim_{n\to\infty} \tau \max(v_n, 0) \\ &= \lim_{n\to\infty} \max(\tau v_n, 0) = \max(\tau v, 0), \end{aligned}$$

where we used the continuity of v_n up to the boundary Γ. ∎

LEMMA 3.2. *Under our standing assumptions, in particular Assumption 2.2, the mappings*

$$\operatorname{conv}(\mathcal{U}) \ni u \mapsto S(u)(x) \in \mathbb{R} \quad \text{and} \quad \operatorname{conv}(\mathcal{U}) \ni u \mapsto \big(\tau S(u)\big)(x) \in \mathbb{R}$$

are concave for almost every $x \in \Omega$ and almost every $x \in \Gamma_N$.

Proof. The proof is similar to the one of the weak maximum principle. Consider $u_1, u_2 \in \operatorname{conv}(\mathcal{U})$ and $\lambda \in [0,1]$. Define $y_i \in Y$, $i = 1,2,3$, by

$$y_1 := S(u_1),\ y_2 := S(u_2),\ y_3 := S(\lambda u_1 + (1-\lambda) u_2).$$

If one subtracts the weak formulation for y_3 from the sum of the ones for y_1 and y_2 scaled by λ and $(1-\lambda)$, respectively, it follows that

$$\begin{aligned}\langle Ay_4,\, v\rangle_{V^*,V} + \int_\Omega \big(\lambda g(x,y_1) + (1-\lambda)g(x,y_2) - g(x,y_3)\big)v\,dx \\ + \int_{\Gamma_N} \big(\lambda b(x,y_1) + (1-\lambda)b(x,y_2) - b(x,y_3)\big)v\,ds = 0 \quad \forall\, v \in V,\end{aligned} \tag{3.1}$$

where $y_4 := \lambda y_1 + (1-\lambda)y_2 - y_3$. Next we choose $v = y_4^+ := \max(y_4, 0)$ as test function, which is in V due to [23, Thm. A.1]. Let us define (up to sets of zero measure)

$$\Omega_+ := \{x \in \Omega : y_4(x) > 0\} \quad \text{and} \quad \Gamma_+ := \{x \in \Gamma_N : (\tau y_4)(x) > 0\}.$$

Then the convexity of $g(x,.)$ by Assumption 2.2 implies for the second addend on the left hand side of (3.1) that

$$\begin{aligned}&\int_\Omega \big(\lambda g(x,y_1) + (1-\lambda)g(x,y_2) - g(x,y_3)\big)y_4^+\,dx \\ &\quad= \int_{\Omega_+} \big(\lambda g(x,y_1) + (1-\lambda)g(x,y_2) - g(x,y_3)\big)y_4 dx \\ &\quad\geq \int_{\Omega_+} \big(g(x,\lambda y_1 + (1-\lambda)y_2) - g(x,y_3)\big)y_4 dx \\ &\quad\geq \int_{\Omega_+} \big(g(x,y_3) - g(x,y_3)\big)y_4 dx = 0,\end{aligned} \tag{3.2}$$

where the second inequality follows from monotonicity of $g(x,.)$ and since

$$\lambda y_1 + (1-\lambda)y_2 > y_3 \ \text{ a.e. in } \Omega_+$$

by definition of Ω_+. In view of Lemma 3.1, we can argue completely analogously in case of the third addend in (3.1) to obtain

$$\int_{\Gamma_N} \big(\lambda b(x,y_1) + (1-\lambda)b(x,y_2) - b(x,y_3)\big)y_4^+\,ds \geq 0. \tag{3.3}$$

All in all, thanks to $\nabla y_4^+ = \chi_{\Omega_+}\nabla y_4$, see [23, Thm. A.1], (3.1)–(3.3) yield

$$\begin{aligned}\alpha\,\|y_4^+\|^2_{H^1(\Omega)} &\leq \langle Ay_4^+,\, y_4^+\rangle_{V^*,V} \\ &= \int_\Omega \Big[\sum_{i=1}^d \Big(\sum_{j=1}^d a_{ij}\frac{\partial y_4^+}{\partial x_j}\frac{\partial y_4^+}{\partial x_i}dx + \beta_i \frac{\partial y_4^+}{\partial x_i}\,y_4^+\Big) + a_0\,(y_4^+)^2\Big]\mathrm{d}x \\ &= \int_\Omega \Big[\sum_{i=1}^d \Big(\sum_{j=1}^d a_{ij}\frac{\partial y_4}{\partial x_j}\frac{\partial y_4^+}{\partial x_i}dx + \beta_i \frac{\partial y_4}{\partial x_i}\,y_4^+\Big) + a_0\,y_4\,y_4^+\Big]\mathrm{d}x \\ &= \langle Ay_4,\, y_4^+\rangle_{V^*,V} \leq 0,\end{aligned}$$

and hence $y_4^+ = \max(y_4, 0) = 0$. The definition of y_4 thus implies

$$\lambda y_1(x) + (1-\lambda)y_2(x) \leq y_3(x) \quad \text{a.e. in } \Omega,$$

which is the desired concavity of $u \mapsto S(u)(x)$. The result for the trace again follows from the positivity of the trace operator. □

Completely analogously one shows that $u \mapsto S(u)(x)$ and $u \mapsto (\tau S(u))(x)$ are *convex* provided that $g(x,.)$ and $b(x,.)$ are *concave*.

LEMMA 3.3. *For every $u \in \operatorname{conv}(\mathcal{U})$ and every $h \in \mathbb{R}^n$ with $u+h \in \operatorname{conv}(\mathcal{U})$ we have*

$$S(u+h)(x) \leq S(u)(x) + (S'(u)h)(x) \quad \text{a.e. in } \Omega.$$

Proof. The operator S is Fréchet-differentiable from $\mathbb{R}^n$ to $L^\infty(\Omega) \hookrightarrow Y$ by Proposition 2.1(ii). For arbitrary $u, h \in \mathbb{R}^n$ we thus have

$$\lim_{t \to 0} \frac{S(u+t\,h)(x) - S(u)(x)}{t} = (S'(u)h)(x) \quad \text{f.a.a. } x \in \Omega.$$

Together with the pointwise concavity from Lemma 3.2, we obtain

$$S(u+h)(x) - S(u)(x) \leq \lim_{t \searrow 0} \frac{S(u+t\,h)(x) - S(u)(x)}{t} = (S'(u)h)(x)$$

for almost all $x \in \Omega$. □

COROLLARY 3.4. *For all $\overline{u} \in \operatorname{conv}(\mathcal{U})$ and almost all $x \in \Omega$, the inequality*

$$S(\overline{u})(x) + S'(\overline{u})(u-\overline{u})(x) \geq y_{\min}(x)$$

is valid for all feasible solutions of (COCP).

Note that the inequalities introduced in Corollary 3.4 are linear in the control variables u. In the following section, we will use constraints of this type in order to replace the (infinite) set of constraints $y(x) \geq y_{\min}(x)$ within an outer approximation scheme.

4. Outer Approximation Algorithm. Let

$$\tilde{\mathcal{U}} := \{u \in \mathcal{U} : S(u)(x) \geq y_{\min}(x) \text{ a.e. in } \Omega\}$$

denote the feasible set of Problem (COCP), in terms of the control variables. Our objective is to devise an algorithm for solving (COCP) to global optimality. Equivalently, we aim at solving the problem

$$\text{(COCP')} \qquad \begin{cases} \min & c^\top u \\ \text{s.t.} & u \in \tilde{\mathcal{U}}\,. \end{cases}$$

The complexity of (COCP') is now hidden in the definition of the set $\tilde{\mathcal{U}}$. The results of the previous section allow us to define an outer approximation algorithm for (COCP'). It is based on Corollary 3.4, which yields an efficient method to cut off any vector $u \in \mathcal{U}$ violating some of the constraints

$$y^*(x) \geq y_{\min}(x)$$

by a cutting plane, i.e., by a linear constraint on $\mathcal{U}$ that is valid for $\tilde{\mathcal{U}}$.

Outer Approximation Algorithm for (COCP)

1. Set $\mathcal{U}_0 := \mathcal{U}$.
2. Minimize $c^\top u$ over $u \in \mathcal{U}_0$, let u^* be the resulting optimizer.
3. Compute y^* by solving

$$\begin{aligned} Ay + g(y) &= \sum_{i=1}^{\ell} u_i^* \, \psi_i && \text{in } \Omega \\ \frac{\partial y}{\partial n_A} + b(y) &= \sum_{j=\ell+1}^{n} u_j^* \, \phi_j && \text{on } \Gamma_N \\ y &= 0 && \text{on } \Gamma_D \ . \end{aligned}$$

4. If $y^* \geq y_{\min}$ a.e., return u^* as optimal solution.
5. Choose some $x^* \in \Omega$ with $y^*(x^*) < y_{\min}(x^*)$ at random, add

$$y^*(x^*) + S'(u^*)(u - u^*)(x^*) \geq y_{\min}(x^*)$$

as linear inequality in u to $\mathcal{U}_0$, and go to Step 2.

THEOREM 4.1. *The above algorithm terminates in finite time. With probability one, it returns a globally optimal solution to Problem* (COCP).

Proof. First note that in every iteration, the algorithm either cuts off a point from $\mathcal{U}$ or terminates. Indeed, the left hand side of the constraint added in Step 5 agrees with $S(u)(x^*)$ when $u = u^*$. The number of iterations is thus limited by $|\mathcal{U}|$, which is finite by the boundedness of $\mathcal{U} \subseteq \mathbb{Z}^n$. The correctness follows from the fact that all added linear inequalities are valid for $\mathcal{U}_0$ with probability one, which was shown in the previous section. □

REMARK 2. *As indicated in Remark 1, the range of S and $S'(u)$ is contained in $C(\overline{\Omega})$ under mild assumptions on the data. The inequality in Corollary 3.4 then holds for every $x \in \overline{\Omega}$ rather than almost everywhere. In particular, our outer approximation algorithm certainly returns a globally optimal solution in this case, not only with probability one.*

For Step 2 of the above algorithm, one can use any standard solver for integer linear programs. Step 3 requires to solve a nonlinear PDE. We emphasize that the PDE associated with the inequality constraint in Step 5 is *linear*. Thus, to evaluate the right hand side, i.e., $S'(u^*)u$ for a given control vector u, one solves n linear PDEs of the form (2.3) corresponding to the n form functions and employs the superposition principle to compute

$$S'(u^*)u = \sum_{i=1}^{\ell} u_i \, S'(u^*)\phi_i + \sum_{j=\ell+1}^{n} u_j \, S'(u^*)\psi_j \ .$$

Therefore, we need to solve n linear PDEs to produce the cutting plane in Step 5.

In practice, the main challenge is to keep the number of outer iterations small. For this, it is preferable to compute more than one cutting plane per iteration, e.g., by considering several points $x \in \Omega$. The details of our implementation are discussed in Section 6. Moreover, the iterative structure of the algorithm suggests to use reoptimization techniques, in particular for initializing the solution algorithm for the PDE in Step 3. This is exploited in the following section.

5. Reoptimization. Due to the iterative structure of our outer approximation algorithm presented in the previous section, the semilinear elliptic PDE in (COCP) has to be solved many times for different values of u. Due to this, it is crucial to develop fast reoptimization techniques that can exploit the information collected in prior iterations. More precisely, when solving the PDE, we propose to speed up the Newton method by deriving an initial solution from either Taylor approximations or interpolations of $S(u)$ in the new control vector u. Both approaches are evaluated experimentally in Section 6.

5.1. Taylor Approximation. The first approach is to approximate $S(u)$ for a new control vector u by using a first order Taylor approximation in one of the vectors $\overline{u}$ considered in an earlier iteration, assuming that

$$S(u)(x) \approx S(\overline{u})(x) + (S'(\overline{u})(u-\overline{u}))(x), \tag{5.1}$$

see Proposition 2.1(ii). Note that in our algorithm the derivatives $S'(\overline{u})h$ are calculated anyway for the construction of cutting planes as devised in Corollary 3.4, using the linearized PDE in Proposition 2.1; we thus obtain the Taylor approximation for free. It can easily be shown that for linear functions g and b equality holds in (5.1). More generally, this approach can be expected to work well whenever the PDE in (COCP) is nearly linear.

5.2. Inter-/Extrapolation. The second approach uses inter- or extrapolation. It aims at predicting the solution $S(u)(x)$ for a new u, if enough sample points $(u^{(j)}, y^{(j)})$, $j = 1, \dots, t$, are available. The approach depends on the specific semilinear elliptic PDE. We assume in the following that the inverse function $g^{-1}(x,.)\colon \mathbb{R} \to \mathbb{R}$ of $g(x,.)$ exists and restrict ourselves to the special case of $\Gamma_N = \emptyset$ for sake of simplicity. The PDE can then be written as

$$g(x,y) = \sum_{i=1}^{\ell} u_i\, \psi_i(x) - (Ay)(x)\ .$$

By neglecting the term Ay, we assume that $S(u)(x)$ depends on u in the form

$$g(x, S(u)(x)) \approx a(x)^\top u + b(x)$$

for each fixed $x \in \Omega$ and some $a(x) \in \mathbb{R}^\ell$, $b(x) \in \mathbb{R}$. Within an interpolation scheme, one first calculates the coefficients $a(x)$ and $b(x)$ by solving a least squares problem based on the equations

$$g(x, y^{(j)}(x)) = a(x)^\top u^{(j)} + b(x),\ j = 1, \dots, t\ .$$

The initial guess for the state y in u can then be chosen as

$$y_{\text{init}}(x) = g^{-1}\left(a(x)^\top u + b(x)\right).$$

Note that this interpolation has to be performed for all $x \in \Omega$.

This approach can be accelerated further for problems where the impact of the control variables u on Ay is low. In this case, we simply set $a(x)_i = \psi_i(x)$ and construct $b(x)$ out of the previously calculated Ay, e.g.,

$$b(x) = \text{mean}_{j \in J}\left(g(x, y^{(j)}(x)) - \sum_{i=1}^{\ell} u_i^{(j)}\, \psi_i(x)\right).$$

The index set J can be chosen differently from $\{1, \dots, t\}$. For example, one may consider only the solutions $u^{(j)}$ nearest to the new iterate u.

6. Experimental Results. To evaluate the potential of our algorithm experimentally, we implemented it in MATLAB R2014A, using CPLEX 12.5 as ILP solver (toolbox function `cplexbilp` for binary controls and `cplexmilp` for integer controls). CPLEX is run in default settings except that the parallel mode is switched off. All computations have been performed on a 64bit Linux system with an Intel Xeon E5-2640 CPU @ 2.5 GHz. In all experiments, we set the time limit to 5 CPU hours.

Throughout our experiments we consider a square domain $\Omega = [0,1]^2$ and partition this domain into as many parts as we have binary optimization variables, i.e. $\Omega = \cup_i^n P_i$ with pairwise disjoint P_i, $i = 1, \dots, n$. The test problem is defined as follows:

$$\left\{\begin{array}{lll} \min \quad c^\top u & & \\ \text{s.t.} & y \geq 0.5\chi_{[0.1,0.9]^2} & \text{a.e. in } \Omega \\ & -\Delta y + \dfrac{1}{2p} y^p = \displaystyle\sum_{i=1}^{\ell} 100 u_i \, \chi_{P_i} & \text{in } \Omega \\ & y = 0 & \text{on } \partial\Omega \\ \text{and} \quad u \in \mathcal{U}. & & \end{array}\right. \tag{6.1}$$

In particular, we do not consider any Neumann boundary conditions in our experiments, so that $\ell = n$.

Unless stated differently, we choose $n = 25$, $p = 2$, $c_i = 1$ for $i = 1, \dots, n$, and $\mathcal{U} = \{0,1\}^n$. Note that the above problem satisfies the conditions of Section 2. In particular, $g(x,y) = \frac{1}{2p} y^p$ is non-decreasing and convex since (6.1) is defined so that $y \geq 0$ holds for all $x \in \Omega$. The factor $\frac{1}{2p}$ and the other constants are chosen in order to avoid trivial solutions where all switches are on (or all switches are off).

For the solution of the semilinear elliptic PDE we use a finite element method. To be more precise, we employ a standard Galerkin scheme with continuous and piecewise linear ansatz and test functions. For the computational mesh, we use a uniform Friedrich-Keller-triangulation with 10201 vertices. The discrete system arising in this way is is solved by Newton's method, which terminates successfully if the residuum is less than 10^{-6}. The linear systems of equations are solved by direct solvers based on sparse LU decompositions. The computational mesh is aligned with the above mentioned partitioning defining the sets P_i, so that these sets are resolved exactly. In our experiments we use the `kmeans` algorithm with as many clusters as discrete controls n.

6.1. Choice of Reoptimization Strategy. As described in Section 5, the solution of the semilinear elliptic PDE can be speeded up by reoptimization. We compare the Taylor approximation and the interpolation approach with two straightforward heuristics. The resulting four strategies differ in the choice of the initial solution for the Newton method:

`PDE_ZERO`: Zero vector, i.e., $y(x) = 0 \; \forall x \in \Omega$.

`PDE_LINEAR`: Solution of a linear PDE obtained by neglecting the term $g(x,y)$.

`PDE_TAYLOR`: The first order Taylor approximation of $S(u)$, calculated in the closest point $\overline{u}$ to u that has been considered before; see Section 5.1.

`PDE_INTERP`: Interpolation of $S(u)$, taking into account only the five nearest solutions $u^{(j)}$ of u; see Section 5.2.

Whenever an initialization as described above is not applicable, e.g. in the first iteration, we choose the zero vector.

We compare the four choices above for a fixed number of $n = 25$ binary controls. As the nonlinearity of g, i.e. the exponent p in (6.1), has got the highest influence on the performance of the reoptimization methods, we evaluate the cases $p \in \{1, 2, 3, 4\}$. For each p, Problem (6.1) is solved, so that each iteration of our algorithm serves as a test instance for the reoptimization heuristics. The results (number of iterations of the Newton method and time) are shown in Table 1. The dependence of the Newton method on the exponent p is highly visible. In particular, the number of iterations grows with p in a similar order for all reoptimization strategies except `PDE_INTERP`. The latter method clearly dominates all others for $p = 2, 3, 4$, needing less iterations and CPU time. For $p = 1$ it is known from theory that the Taylor approximation is exact, which leads to zero further Newton iterations.

Based on these results, we choose the interpolation strategy of Section 5.2 throughout the following experiments.

	PDE_ZERO		PDE_TAYLOR	
p	iter	cpu time [s]	iter	cpu time [s]
1	1 / 1 / 1.0	0.25 / 0.26 / 0.26	0 / 0 / 0.0	0.06 / 0.08 / 0.07
2	17 / 20 / 19.1	3.19 / 3.72 / 3.43	9 / 19 / 13.8	1.64 / 3.35 / 2.49
3	26 / 32 / 29.5	4.97 / 6.28 / 5.69	16 / 27 / 22.3	3.10 / 5.33 / 4.34
4	34 / 44 / 39.7	7.09 / 9.71 / 8.33	22 / 38 / 30.8	4.56 / 7.82 / 6.43
	PDE_LINEAR		PDE_INTERP	
p	iter	cpu time [s]	iter	cpu time [s]
1	1 / 1 / 1.0	0.26 / 0.26 / 0.26	1 / 1 / 1.0	0.26 / 0.28 / 0.27
2	16 / 19 / 18.1	2.92 / 3.42 / 3.23	3 / 7 / 5.4	0.61 / 1.35 / 1.02
3	25 / 31 / 28.5	4.77 / 5.99 / 5.52	3 / 8 / 6.5	0.67 / 1.70 / 1.34
4	33 / 43 / 38.7	6.85 / 9.06 / 8.14	3 / 10 / 7.5	0.72 / 2.22 / 1.63

TABLE 1

Comparison of different reoptimization heuristics for the solution of the semilinear elliptic PDE for four different exponents p. Entries are split up into minimum, maximum, and mean. For the minimum and mean the first trivial solution with zero iterations is neglected.

6.2. Choice of Cutting Planes. The inequalities of Corollary 3.4, which form the basis of our outer approximation algorithm, are valid for almost all $x \in \Omega$. This allows to add, for any infeasible $u \in \mathcal{U}$, as many cutting planes as there are vertices x_i^* of the finite element discretization that violate $y(x_i^*) \geq y_{\min}(x_i^*)$. We noticed, however, that nearby points often lead to inequalities cutting off the same vectors from $\mathcal{U}$, and thus have a negative influence on the efficiency of the overall algorithm since the solution of the ILPs is slowed down. Therefore we choose some minimal distance r and enumerate all points x_i^* in descending order according to the violation of the constraint $y(x_i^*) \geq y_{\min}(x_i^*)$, adding the corresponding cutting plane if and only if no point closer than r to x_i^* has been used before to produce a cutting plane. In particular, we obtain a set $J \subseteq \{i : y^*(x_i^*) < y_{\min}(x_i^*)\}$ such that

$$\left\|x_i^* - x_j^*\right\|_2 \geq r \quad i, j \in J,\ i \neq j\ .$$

The influence of the choice of r on the performance of our algorithm is shown in Figure 1 and Figure 2, for $p = 2$ and $p = 3$, respectively. As expected, the difficulty of solving the ILPs decreases with growing r, as the number of constraints becomes smaller, whereas the number of iterations (and hence the total time needed to solve the PDEs) increases, as less vectors from $\mathcal{U}$ are cut off in one iteration. When combining

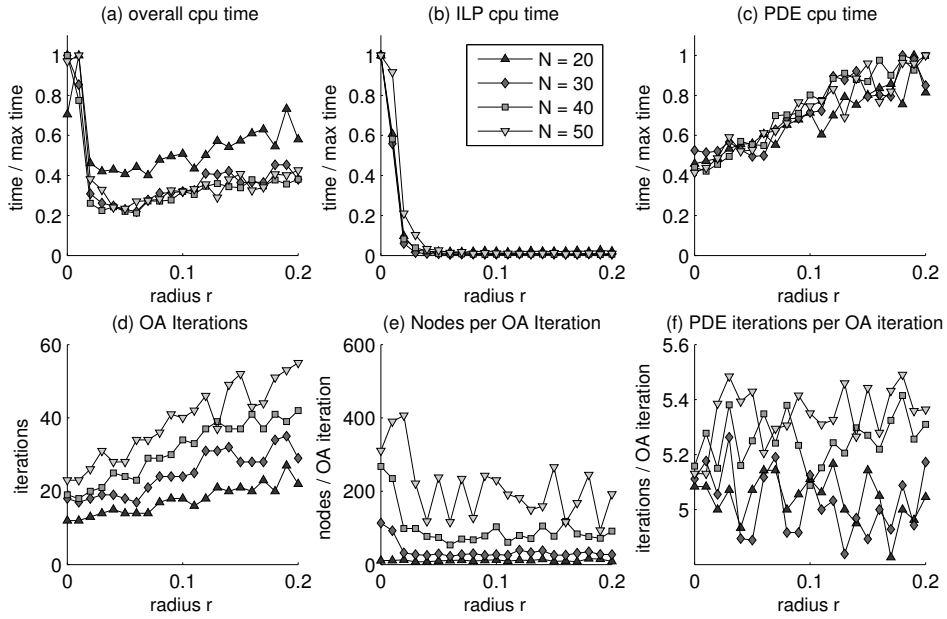

FIG. 1. *Comparison of different radii r for the choice of cutting planes with p = 2.*

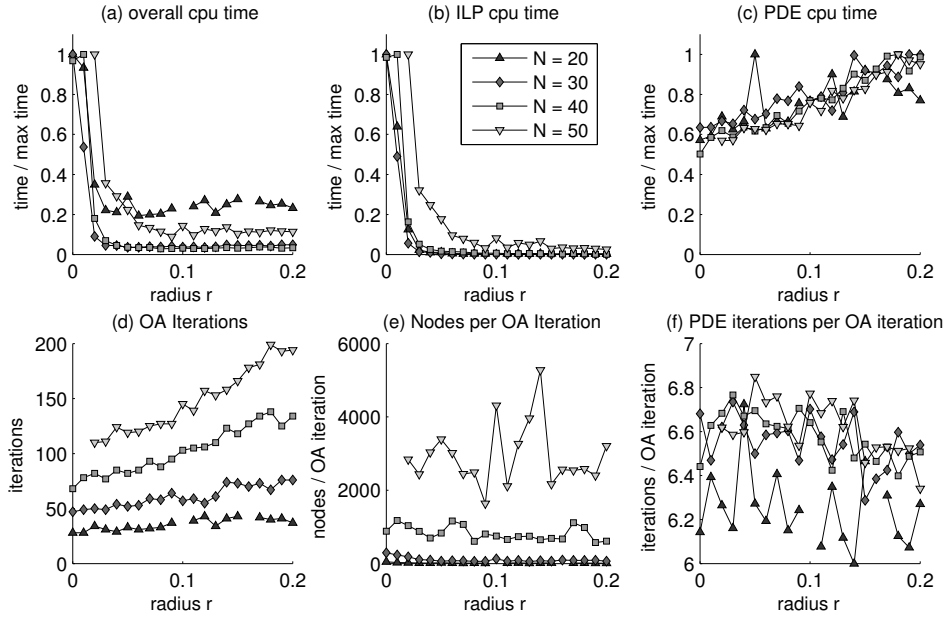

FIG. 2. *Comparison of different radii r for the choice of cutting planes with p = 3.*

these two effects in terms of the overall CPU time, it turns out in our experiments that the minimum is attained at $r \approx 0.04$. Although it may not be the optimal choice for arbitrary parameters, we choose $r = 0.04$ in the following for sake of comparability.

6.3. Example 1: Uniform Costs and Binary Controls. We first investigate the case of uniform costs and binary variables, i.e. we keep $c_i = 1$ for all $i = 1, \dots, n$

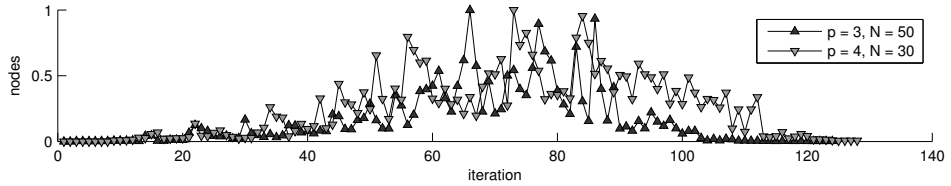

FIG. 3. *ILP nodes needed in each major iteration for the solution of* (6.1) *with* $p = 3, n = 50$ *and* $p = 4, n = 30$. *The number of nodes is scaled by the maximum number of nodes.*

and $\mathcal{U} = \{0,1\}^n$. In other words, sources can be switched on or off and we minimize the number of sources we need to switch on in order to reach the pointwise minimum temperature. In our experiments, we vary the number of discrete controls n as well as the exponent p, in order to illustrate the influence of both the problem size and the nonlinearity. The results are listed in Table 2, where we report the number of major iterations of the outer approximation algorithm, the number of added cutting planes, the number of nodes and the CPU time required by the ILP solver, as well as the number of iterations and the CPU time required by the PDE solver; all figures are summed up over the major iterations. The last column states the total running time in order to solve the instance to global optimality.

We are able to solve the problem with up to 230 ($p = 1$), 110 ($p = 2$), 50 ($p = 3$) and 30 ($p = 4$) binary controls. For growing n (and fixed p), the computation times of the ILP solves, the PDE solves and thus of the overall process increase. While the computation time for solving the PDE is dominant for small problems, the time for solving the ILPs becomes dominant for larger problems. In fact, while the size of the discretized PDE does not depend on n, the ILP solution time can be expected to grow exponentially in n for reasons of complexity. Furthermore, the nonlinearity of the PDE – varied through the exponent p – has a significant influence on the number of Newton iterations, as already known from Section 6.1, but also on the number of nodes within the ILP solves and major iterations of the outer approximation algorithm. Both increase with the exponent p. This leads to the conclusion that the cutting planes' quality is reduced for stronger nonlinearities.

When investigating the solution process over the major iterations of the outer approximation, it can be noticed that the number of nodes and computation time required of the ILP solver is not equally distributed. In fact, for most of the problems, the most difficult ILPs are those in the middle of the process, while the ILPs in the beginning and in the end are relatively easy to solve. This behavior is plotted in Figure 3 for $p = 3$, $n = 50$ and $p = 4$, $n = 30$. A possible explanation is the following: while in the beginning the growing number of cutting planes makes the problem harder to solve, the smaller number of remaining feasible solutions leads to a faster solution of the ILPs towards the end.

6.4. Example 2: Randomly Distributed Costs and Binary Controls. In addition to Example 1, we consider a problem with randomly distributed costs c. Therefore, in every problem instance, each c_i, $i = 1, \ldots, n$, is chosen independently in the interval $(0, 1)$ using a uniformly distributed random number generator. The results are shown in Table 3 for different p and increasing n, starting at the largest possible n of Table 2.

It turns out that with randomly distributed costs problem (6.1) can be solved much more efficiently than with uniform costs, resulting in less computation time for

					ILP solve		PDE solve		
p	n	obj	iter	#cuts	nodes	time [s]	iter	time [s]	time [s]
1	10	10.00	2	323	0	0.07	0	0.09	1.20
	30	23.00	4	332	0	0.14	3	0.86	6.56
	50	26.00	3	325	0	0.07	2	0.56	6.57
	70	31.00	6	357	636	0.28	5	1.27	20.12
	90	31.00	8	360	6809	0.98	7	1.79	37.09
	110	30.00	7	356	5929	1.16	6	2.11	41.84
	130	32.00	11	381	11079	2.65	10	2.92	79.85
	150	34.00	12	374	59658	6.90	11	2.91	97.26
	170	34.00	8	362	38561	5.29	7	1.79	71.05
	190	36.00	14	400	134121	17.43	13	3.36	151.25
	210	38.00	21	414	1205120	125.40	20	5.17	356.43
	230	39.00	17	434	4895172	643.05	16	4.10	844.04
	250	-	-	-	-	-	-	-	$>$ 5h
2	10	10.00	9	1214	19	0.34	42	8.06	13.30
	20	20.00	15	2113	119	0.68	74	15.03	32.11
	30	26.00	19	2446	481	1.57	93	18.53	49.71
	40	24.00	25	3250	1919	7.40	129	30.54	96.41
	50	27.00	28	3624	3296	13.03	151	30.49	116.41
	60	33.00	36	5048	15599	49.18	195	38.78	197.11
	70	43.00	39	4955	31156	74.94	216	42.47	251.47
	80	47.00	55	6870	136625	313.88	295	57.59	592.31
	90	43.00	52	6637	422640	756.62	281	54.00	1045.21
	100	41.00	62	7867	2916648	6708.09	338	64.84	7084.63
	110	40.00	64	8206	4424287	12230.87	358	69.06	12658.52
	120	-	-	-	-	-	-	-	$>$ 5h
3	10	10.00	12	1702	35	0.45	59	13.06	20.36
	20	20.00	29	5142	447	3.45	195	42.53	79.82
	30	23.00	54	9588	5163	49.69	358	78.74	221.28
	40	23.00	85	14696	59028	359.29	567	121.18	669.36
	50	26.00	124	20198	376184	2016.61	818	172.36	2521.54
	60	-	-	-	-	-	-	-	$>$ 5h
4	10	10.00	24	3979	153	1.28	132	29.79	46.00
	20	20.00	49	8439	862	8.35	356	78.30	143.63
	30	18.00	128	20777	89147	546.52	998	218.24	975.05
	40	-	-	-	-	-	-	-	$>$ 5h

TABLE 2

Results for different numbers n of binary controls and different exponents p. The number of added cuts (#cuts), the number of B&B nodes and the time for ILP solution, as well as the number of iterations and the time for the PDE solution are summed up over the major iterations. All times are CPU times in seconds.

the same number of binary controls n, e.g., 65.03s instead of 844.04s ($p = 1, n = 230$) or 334.04s instead of 975.05s ($p = 4, n = 30$). On the one hand – which is the main factor here – the ILPs can be solved more efficiently. In fact, in discrete optimization, uniform objective functions often lead to harder problems in practice as they admit more feasible solutions with similar or equal objective function values. On the other hand, the outer approximation algorithm needs less major iterations compared to the

					ILP solve		PDE solve		
p	n	obj	iter	#cuts	nodes	time [s]	iter	time [s]	time [s]
1	230	12.97	6	355	3504	1.10	5	1.33	65.03
	300	10.36	5	361	2187	0.93	4	1.09	65.78
	400	12.17	8	381	19111	6.07	7	1.83	153.48
	500	11.69	9	396	26921	8.59	8	2.09	216.31
	600	11.48	9	373	8771	6.16	8	2.08	257.13
	700	10.99	7	388	227584	57.32	6	1.53	267.01
	800	9.06	5	375	27855	8.77	4	1.05	174.01
	900	10.22	10	403	1209536	362.98	9	2.30	787.69
	1000	10.47	10	423	3806043	1261.92	9	2.33	1727.43
	1100	8.98	8	421	816158	278.18	7	1.80	664.26
	1200	8.39	7	399	297526	86.89	6	1.55	467.00
	1300	10.37	9	425	1480619	617.35	8	2.06	1159.90
	1400	9.41	11	437	2763061	1136.97	10	2.62	1870.41
	1500	8.32	11	426	34749222	14125.29	10	2.58	14910.34
	1600	8.93	10	425	22317320	9968.18	9	2.36	10718.31
	1700	7.86	10	417	2637934	983.20	9	2.30	1755.28
	1800	9.32	8	436	23222795	10947.27	7	1.85	11598.10
	1900	-	-	-	-	-	-	-	$>$ 5h
	2000	6.95	7	420	4901397	1836.15	6	1.49	2424.99
	2100	-	-	-	-	-	-	-	$>$ 5h
	2200	7.79	7	433	11392770	6348.67	6	1.55	7042.11
	2300	-	-	-	-	-	-	-	$>$ 5h
2	110	16.30	35	4585	69439	199.54	184	36.16	427.60
	160	15.38	51	4996	257748	1010.15	259	50.24	1470.00
	170	14.30	52	6099	295986	1503.51	264	51.12	2002.70
	180	14.73	72	5930	1084171	4769.91	347	66.88	5489.70
	190	15.84	58	6371	907184	4077.83	299	57.60	4696.08
	200	16.33	63	6699	3056593	13606.88	314	59.66	14295.82
	210	14.43	55	6740	1059472	5063.15	286	53.13	5676.58
	220	15.18	73	6449	4085796	16659.66	363	69.07	17521.25
	230	15.85	60	5926	1101797	6104.15	307	59.80	6849.80
	240	-	-	-	-	-	-	-	$>$ 5h
3	50	10.96	73	9183	10766	103.52	452	95.06	388.18
	60	12.61	89	10978	132101	560.70	569	118.75	958.42
	70	10.87	97	14534	541775	3540.78	618	127.82	4022.44
	80	9.04	74	9035	82003	681.40	447	92.42	1075.49
	90	-	-	-	-	-	-	-	$>$ 5h
4	30	7.08	84	9884	7267	63.06	583	130.16	334.04
	40	6.45	133	15993	134260	915.58	937	208.77	1418.82
	50	4.19	126	17825	45593	799.74	826	183.96	1312.61
	60	-	-	-	-	-	-	-	$>$ 5h

TABLE 3

Results for different numbers n of binary controls and different exponents p with randomly distributed costs c. The number of added cuts (#cuts), the number of B&B nodes and the time for ILP solution, as well as the number of iterations and the time for the PDE solution are summed up over the major iterations. All times are CPU times in seconds.

					ILP solve		PDE solve		
n	$u_{\max}$	obj	iter	#cuts	nodes	time [s]	iter	time [s]	time [s]
20	1	20.00	6	1049	0	0.14	30	5.86	11.81
	10	12.00	7	1470	20	0.20	22	4.53	11.97
	100	11.40	7	1659	579	0.37	14	3.04	10.69
40	1	40.00	6	964	22	0.19	27	5.74	17.02
	10	24.00	8	1466	144	0.29	27	5.55	21.55
	100	22.80	8	1679	26259	9.38	18	3.93	29.74
60	1	54.00	7	1206	24	0.24	37	7.07	25.24
	10	32.50	7	1491	314	0.38	28	5.54	24.80
	100	30.86	8	1662	223196	78.02	26	5.32	105.41
80	1	52.00	6	1086	84	0.23	30	5.71	25.21
	10	32.00	7	1429	72057	9.58	27	5.24	38.43
	100	30.25	8	1731	7309074	3089.59	23	4.58	3121.86
100	1	68.00	7	1165	58	0.31	34	6.49	36.56
	10	40.90	7	1430	300436	50.29	26	5.03	85.25
	100	-	-	-	-	-	-	-	$>$ 5h

TABLE 4

Results for different numbers n of integer controls for p = 2. The number of added cuts (#cuts), the number of B&B nodes and the time for ILP solution, as well as the number of iterations and the time for the PDE solution are summed up over the major iterations. All times are CPU times in seconds.

uniform case, e.g., 35 instead of 64 (for $p = 2$ and $n = 110$) or 84 instead of 128 (for $p = 4$ and $n = 30$). In summary, we are able to solve problems with up to 2200 ($p = 1$), 230 ($p = 2$), 80 ($p = 3$) and 50 ($p = 4$) binary controls in this example.

6.5. Example 3: Uniform Costs and Integer Controls. In the next example, we apply our algorithm to an optimal control problem with integer controls. Therefore, we set $\mathcal{U} = \{0, \dots, u_{\max}\}$ with $u_{\max} \in \mathbb{N}$. In order to avoid trivial solutions, the PDE in (6.1) is altered to

$$\left.\begin{aligned} y &\geq 15.0\chi_{[0.1,0.9]^2} && \text{a.e. in } \Omega \\ -\Delta y + \frac{1}{2p} y^p &= \frac{100}{u_{\max}} \sum_{i=1}^{\ell} u_i \, \chi_{P_i} && \text{in } \Omega \\ y &= 0 && \text{on } \partial\Omega. \end{aligned}\right\} \tag{6.2}$$

Moreover, the objective function is scaled by $u_{\max}^{-1}$ for sake of comparibility.

Table 4 lists the results for $n \in \{20, 40, 60, 80, 100\}$ and $u_{\max} \in \{1, 10, 100\}$, where $u_{\max} = 1$ is the binary case for comparison. Apart from the case $n = 100$ for $u_{\max} = 100$ all instances could be solved within the 5h CPU time limit. When comparing the results for different $u_{\max}$, it becomes clear that the increase of the integer domain has low influence on the outer approximation algorithm (increase of approximately 2 iterations), but high influence on the ILP solves, which lead to unsuccessful terminations in the above mentioned case. However, the expansion of the discrete set $\mathcal{U}$ through enlarging the integer domain instead of an increase of variables is less computationally demanding, e.g. compare the case $p = 2, n = 60$ in Table 2 (36 iterations, 197.11s) with the case $n = 20, u_{\max} = 10$ in Table 4 (7 iterations, 11.97s).

6.6. Example 4: Stationary Heating of a Metallic Workpiece. We finally show some results for the application mentioned in Section 2, i.e., the stationary heating of a metallic workpiece. We solve the problem in two dimensions, which requires to adapt the Boltzmann type radiation boundary condition to

$$\kappa \nabla y \cdot n + \sigma\, y^3 = \sigma\, y_0^3$$

with $\sigma = 1.92 \cdot 10^{-10}$. The workpiece is given as $[0,1]^2$ and the heat sources correspond to 0.02 by 0.02 squares arranged on a $k \times k$-grid regularly covering the workpiece, each one equipped with a power of 2500 W. We consider uniform costs again, thus aiming at a minimal number of sources switched on. The surrounding temperature is chosen as 293 K. In Figure 4, optimal temperature distributions are depicted for $y_{\min} \equiv 1000\,\mathrm{K}$, for the cases $k = 5$ (12 sources needed, 195 CPU seconds) and $k = 15$ (11/3501). Figure 5 shows optimal solutions for $k = 15$ with $y_{\min} \equiv 750\,\mathrm{K}$ (5/1152) and $y_{\min} \equiv 1250\,\mathrm{K}$ (20/6767).

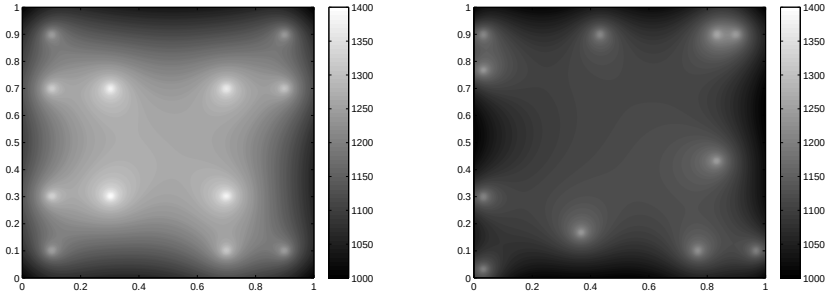

FIG. 4. *Optimal states for $k = 5$ (left) and $k = 15$ (right) with $y_{\min} \equiv 1000\,K$.*

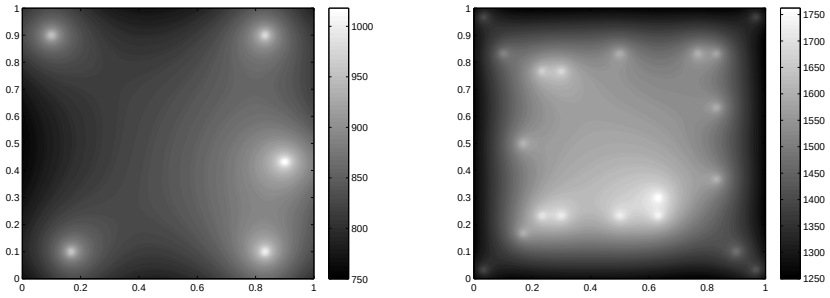

FIG. 5. *Optimal states for $k = 15$ with $y_{\min} \equiv 750\,K$ (left) and $y_{\min} \equiv 1250\,K$ (right).*

7. Conclusions and Future Directions. We have presented an outer approximation approach for solving a large class of semilinear elliptic optimal control problems with static combinatorial controls, yielding globally optimal solutions in finitely many iterations. The algorithm exploits the pointwise concavity of the solution operator in terms of the control variables in order to generate valid linear cutting planes.

The basic idea of the algorithm can be easily extended to the case of mixed-integer controls, since the resulting cutting planes remain valid in this case. However,

in the presence of continuous control variables, we cannot expect finite convergence anymore, and more care is needed in the selection of cutting planes. Another possible extension is to replace the linear objective function by a quadratic or more general nonlinear function in the control variables. This requires that a sufficiently fast solver for the resulting class of (mixed-)integer nonlinear programs is at hand. Note that convexity is not required here as long as the mixed-integer problem can be solved to global optimality in practice.

As observed in our experiments, most of our algorithm's running time is spent for solving ILPs, particularly for larger instances. A significant speed-up can thus be expected from a more sophisticated solution strategy for these ILPs, exploiting the iterative structure of the algorithm again. For this, one could use general ideas discussed for outer approximation algorithms such as branch-and-cut-based outer approximation [28, 7].

An open question is how to deal with pointwise upper bounds on the state instead of pointwise lower bounds. In this case, the concavity of the solution operator cannot easily be exploited for producing cutting planes, since the tangent inequalities do not give rise to valid inequalities on the control variables anymore. A related question is how to deal with tracking-type objective functions. This is left as future work.

REFERENCES

[1] J.-J. Alibert and J.-P. Raymond, *Boundary control of semilinear elliptic equations with discontinuous leading coefficients and unbounded controls*, Numerical Functional Analysis and Optimization, 18 (1997), pp. 235–250.

[2] M. Avraam, N. Shah, and C. Pantelides, *Modelling and optimisation of general hybrid systems in the continuous time domain*, Computers & Chemical Engineering, 22, Supplement 1 (1998), pp. S221–S228.

[3] S. Balakrishna and L. T. Biegler, *A unified approach for the simultaneous synthesis of reaction, energy, and separation systems*, Industrial & Engineering Chemistry Research, 32 (1993), pp. 1372–1382.

[4] V. Bansal, V. Sakizlis, R. Ross, J. D. Perkins, and E. N. Pistikopoulos, *New algorithms for mixed-integer dynamic optimization*, Computers & Chemical Engineering, 27 (2003), pp. 647–668.

[5] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, *Mixed-integer nonlinear optimization*, Acta Numerica, 22 (2013), pp. 1–131.

[6] T. J. Boehme, M. Schori, B. Frank, M. Schultalbers, and B. Lampe, *Solution of a hybrid optimal control problem for parallel hybrid vehicles subject to thermal constraints*, in 2013 IEEE 52nd Annual Conference on Decision and Control (CDC), IEEE, 2013, pp. 2220–2226.

[7] P. Bonami, L. Biegler, A. Conn, G. Cornuéjols, I. Grossmann, C. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, *An algorithmic framework for convex mixed integer nonlinear programs*, Discrete Optimization, 5 (2008), pp. 186–204.

[8] M. Buss, O. von Stryk, R. Bulirsch, and G. Schmidt, *Towards hybrid optimal control*, at-Automatisierungstechnik: Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik, 48 (2000), pp. 448–459.

[9] E. Casas, *Control of an elliptic problem with pointwise state constraints*, SIAM Journal on Control and Optimization, 24 (1986), pp. 1309–1318.

[10] ———, *Boundary control of semilinear elliptic equations with pointwise state constraints*, SIAM Journal on Control and Optimization, 31 (1993), pp. 993–1006.

[11] R. Chandra, *Partial differential equations constrained combinatorial optimization on an adiabatic quantum computer*, master's thesis, Purdue University, 2013.

[12] K. Deckelnick and M. Hinze, *Convergence of a finite element approximation to a state constrained elliptic control problem*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1937–1953.

[13] V. D. Dimitriadis and E. N. Pistikopoulos, *Flexibility analysis of dynamic systems*, Industrial & Engineering Chemistry Research, 34 (1995), pp. 4451–4462.

[14] A. Fügenschuh, B. Geiler, A. Martin, and A. Morsi, *The transport PDE and mixed-integer*

linear programming, in Models and Algorithms for Optimization in Logistics, C. Barnhart, U. Clausen, U. Lauther, and R. H. Möhring, eds., no. 09261 in Dagstuhl Seminar Proceedings, Schloss Dagstuhl - Leibniz-Zentrum fr Informatik, Germany, 2009.

[15] H. Gajewski, K. Gröger, and K. Zacharias, *Nichtlineare Operatorgleichungen und Operatordifferentialgleichungen*, Akademie-Verlag, Berlin, 1974.

[16] B. Geiler, O. Kolb, J. Lang, G. Leugering, A. Martin, and A. Morsi, *Mixed integer linear models for the optimization of dynamical transport networks*, Mathematical Methods of Operations Research, 73 (2011), pp. 339–362.

[17] M. Gerdts, *A variable time transformation method for mixed-integer optimal control problems*, Optimal Control Applications and Methods, 27 (2006), pp. 169–182.

[18] H. Goldberg, W. Kampowsky, and F. Tröltzsch, *On Nemytskij operators in L_p-spaces of abstract functions*, Mathematische Nachrichten, 155 (1992), pp. 127–140.

[19] R. Haller-Dintelmann, C. Meyer, J. Rehberg, and A. Schiela, *Hlder continuity and optimal control for nonsmooth elliptic problems*, Applied Mathematics and Optimization, 60 (2009), pp. 397–428.

[20] F. M. Hante and S. Sager, *Relaxation methods for mixed-integer optimal control of partial differential equations*, Computational Optimization and Applications, 55 (2013), pp. 197–225.

[21] M. Hintermüller and K. Kunisch, *PDE-constrained optimization subject to pointwise constraints on the control, the state, and its derivative*, SIAM Journal on Optimization, 20 (2009), pp. 1133–1156.

[22] F. Incropera and D. D. Witt, *Fundamentals of Heat and Mass Transfer*, Wiley, Chichester, 1985.

[23] D. Kinderlehrer and G. Stampacchia, *An introduction to variational inequalities and their applications*, vol. 31, SIAM, 2000.

[24] C. Kirches, S. Sager, H. G. Bock, and J. P. Schlder, *Time-optimal control of automobile test drives with gear shifts*, Optimal Control Applications and Methods, 31 (2010), pp. 137–153.

[25] C. Meyer, *Error estimates for the finite-element approximation of an elliptic control problem with pointwise state and control constraints*, Control and Cybernetics, 37 (2008), pp. 51–85.

[26] C. Meyer, U. Prüfert, and F. Tröltzsch, *On two numerical methods for state-constrained elliptic control problems*, Optimization Methods and Software, 22 (2007), pp. 871–899.

[27] M. J. Mohideen, J. D. Perkins, and E. N. Pistikopoulos, *Optimal design of dynamic systems under uncertainty*, AIChE Journal, 42 (1996), pp. 2251–2272.

[28] I. Quesada and I. Grossmann, *An LP/NLP based branched and bound algorithm for convex MINLP optimization problems*, Computers and Chemical Engineering, 16 (1992), pp. 937–947.

[29] M. Ruzicka, *Nichtlineare Funktionalanalysis*, Springer, 2004.

[30] S. Sager, H. Bock, and M. Diehl, *The integer approximation error in mixed-integer optimal control*, Mathematical Programming, 133 (2012), pp. 1–23.

[31] S. Sager, M. Jung, and C. Kirches, *Combinatorial integral approximation*, Mathematical Methods of Operations Research, 73 (2011), pp. 363–380.

[32] J. Till, S. Engell, S. Panek, and O. Stursberg, *Applied hybrid system optimization: An empirical investigation of complexity*, Control Engineering Practice, 12 (2004), pp. 1291–1303.

[33] F. Tröltzsch, *Optimale Steuerung partieller Differentialgleichungen*, Vieweg, Wiesbaden, 2nd ed., 2009.

[34] O. von Stryk and M. Glocker, *Decomposition of mixed-integer optimal control problems using branch and bound and sparse direct collocation*, in Proc. ADPM 2000 The 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems, S. Engell, S. Kowalewski, and J. Zaytoon, eds., Dortmund, sep 2000, pp. 99–104.

[35] P. Zhang, D. Romero, J. Beck, and C. Amon, *Solving wind farm layout optimization with mixed integer programming and constraint programming*, in Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, C. Gomes and M. Sellmann, eds., vol. 7874 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 284–299.