On structural similarities of combinatorial integral approximation and binary trust-region steepest descent
Paul Manns†  Mirko Hahn‡  Christian Kirches§  Sven Leyffer¶  Sebastian Sager†
February 17, 2022

Abstract

Combinatorial integral approximation and binary trust-region steepest descent are two approaches for the solution of optimal control problems with discrete (binary in the latter case) control inputs. While the former method approximates solutions of a continuous relaxation with discrete controls, the latter produces a sequence of discrete and objective-decreasing controls by manipulating the level sets of the control inputs. The main difference in their analyses is an additional compactness assumption on the reduced objective functional for the combinatorial integral decomposition that implies an approximation of stationary points or (local) minimizers of the continuous relaxation and their objective function values. We prove that by imposing a suitable compactness assumption on the derivative of the reduced objective, the iterates produced by binary trust-region steepest descent have the same convergence properties. The methods achieve comparable results under comparable structural assumptions on the problem. We provide computational results to illustrate qualitative and quantitative similarities and differences of the iterates of both methods.

Keywords  Mixed-integer optimal control, trust-region methods, relaxation-based methods

Mathematics Subject Classifications  49J45,49M05,90C30

1 Introduction

For bounded domains $\Omega \subseteq \mathbb{R}^d$ we are interested in optimization problems of the form

$$\inf_x J(x) \quad \text{s.t.} \quad x(s) \in V \text{ for almost all (a.a.) } s \in \Omega \text{ and } x \in L^2(\Omega), \quad (P)$$

where $V \subset \mathbb{Z}$ with $|V| < \infty$ and $J$ is a map from $L^2(\Omega)$ to $\mathbb{R}$. For this problem class we study solutions of corresponding continuous relaxations of the form

$$\min_x J(x) \quad \text{s.t.} \quad x(s) \in \text{conv } V \text{ for a.a. } s \in \Omega \text{ and } x \in L^2(\Omega) \quad (R)$$

and their relation to problem (P). The inf in the formulation of (P) and the min in the formulation of (R) are deliberately chosen to highlight that problem (R) (in contrast to (P)) admits a minimizer under mild assumptions [12, 19]. We restrict ourselves to this setting in the interest of a concise presentation. One can,

---

*Submitted to the editors DATE. This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through the Advanced Computing (SciDAC) Program through the FASTMath Institute under Contract No. DE-AC02-06CH11357 and from the German Research Foundation under GRK 2297 MathCoRe (project No. 314838170) and SPP 1962 (projects No. SA 2016/1-2, KI 1839/1-2) and from the German Federal Ministry of Education and Research within the program “Mathematics for Innovations” under the project “Power to Chemicals. Argonne National Laboratory Preprint ANL/MCS-P9652-0222.

†Faculty of Mathematics, TU Dortmund University (paul.manns@tu-dortmund.de).
‡Faculty of Mathematics, Otto von Guericke University Magdeburg.
§Carl-Friedrich-Gauß-Faculty, Technical University of Braunschweig.
¶Mathematics and Computer Science Division, Argonne National Laboratory.
however, extend our analysis in different directions, for example, to the case where an $L^1$-regularization term is added to the objective function, $J(x)$, by following the arguments in [17].

A rich class of instances of (P) are mixed-integer PDE-constrained optimization problems, where $J = j \circ S$, where $j$ is the objective of the optimization and $S$ the control-to-state operator of an underlying partial differential equation (PDE). Such problems arise in many different areas such as topology optimization [14, 10], optimum experimental design [28], and gas network optimization [7, 9].

We analyze the relationship between two algorithmic solution approaches of (P) that have been investigated in recent years: the combinatorial integral approximation (CIA) and the binary trust-region steepest descent (BTR) method. Because the BTR method has been formulated only for $V = \{0, 1\}$ so far, we restrict our analysis to $V = \{0, 1\}$.

**Combinatorial integral approximation** [8, 12, 23, 22] The idea that underlies CIA is to split the solution process of (P) into solving the continuous relaxation (R) and then computing a $V$-valued approximation of the relaxed solution. The approximation process can be analyzed in the weak-*-topology of $L^\infty(\Omega)$ [18]. Relying on compactness properties of an underlying control-to-state operator (e.g., $J = j \circ S$, where $j \in C(L^2(\Omega), \mathbb{R})$ and $S : L^2(\Omega) \to L^2(\Omega)$ is a compact operator), a tight approximation of the optimal objective value by the resulting approximants can be proved [12]. If stationary points are computed in the first step, the approximation properties generalize accordingly.

**Binary trust-region steepest descent** [5, 24, 27] The BTR method applies to the binary case $V = \{0, 1\}$ (generalizations to the case $V = \{0, 1\}^k$ for $k \in \mathbb{N}$ are conceivable but have not been considered in the literature so far). It solves trust-region subproblems in which the level sets of the control $x$, corresponding to the values in $V$, are manipulated to greedily improve the linearized objective. A trust-region constraint limits the volume of the level sets, which is the $L^1$-norm of the control function, and can change from one accepted iterate to the next. The analysis in [5] shows that BTR iterates eventually satisfy a condition called $\varepsilon$-stationarity under a regularity assumption used to obtain sufficient decrease of the aggregated volume of level set manipulations.

The concept $\varepsilon$-stationarity as introduced in [5] resembles first-order necessary optimality conditions for (R). Moreover, similar to CIA, the structural assumptions on the quantities that appear in (P) are in general not able to prevent a fine microstructure from developing over the iterations. In fact, the weak-*-closure of the feasible set of (P) in $L^\infty(\Omega)$ is the feasible set of its relaxation (R) [15, 16]. However, it is not known whether the iterates generated by BTR converge to a limit point that satisfies a first-order optimality condition of (R) if the termination tolerance of BTR is driven to zero. This lack of a convergence result is in contrast to CIA, which has stationary limits under a suitable compactness assumption.

**Contributions** We close this important theoretical gap between BTR and CIA. In particular, we introduce a suitable compactness assumption on the derivative of $J$, and we show that the BTR iterates produced by Algorithm 2 in [5] converge weakly-* in $L^\infty(\Omega)$ to a point that is feasible for and satisfies a first-order optimality condition for the continuous relaxation (R). Moreover, we perform several computational experiments on an example problem that is governed by an elliptic PDE to show how the iterates produced by both methods behave in practice.

**Structure of the paper** In §2 we formally introduce and describe the BTR algorithm. In §3 we relate it to [5] and state our main convergence result. The proof is presented to §4. We provide a computational validation of our findings and demonstrate how the behavior of the two approaches differs in §5.

**Notation** Let $d \in \mathbb{N}$ denote a dimension. For a measurable set $A \subset \Omega$, $\lambda(A)$ denotes the Borel–Lebesgue measure of $A$ in $\mathbb{R}^d$. The function $\chi_A$ denotes the $\{0, 1\}$-valued characteristic function of the set $A$. Let $\mathcal{B}$ denote the Borel $\sigma$-algebra on $\Omega$. For a set $A \subset \Omega$, the set $A^c$ denotes its complement in $\Omega$. For sets $A, B \subset \Omega$ the expression $A \triangle B$ denotes the symmetric difference between $A$ and $B$, that is, $A \triangle B := (A \cup B) \setminus (A \cap B)$. The inner product of the Hilbert space $L^2(\Omega)$ is denoted by $\langle \cdot, \cdot \rangle_{L^2}$. For a space $X$ and its topological dual $X^*$ we denote the pairing that puts $X$ and $X^*$ in duality by $\langle \cdot, \cdot \rangle_{X^*, X}$. 
2 Binary Trust-Region Steepest Descent

The BTR algorithm operates on characteristic functions induced by measurable sets. We introduce the inputs and the trust-region subproblem. Then we describe the iterations of Algorithm 1 step by step. We relate the quantities in our variant of the algorithm to the one introduced as Algorithm 2 in [5], which purely takes the point of view of measurable sets.

The BTR algorithm is given as Algorithm 1 and can be interpreted as a variant of [5, Algorithm 2]. In particular, Algorithm 1 corresponds to [5, Algorithm 2] with the choices \( J(A) := J(\chi_A) \) for \( A \in \mathcal{B} \). We also choose \( \varepsilon = 0 \) because we aim to study the asymptotics of the algorithm when it is not stopped early.

\textbf{Algorithm 1} BTR algorithm to optimize (P) and (R) for \( V = \{0, 1\} \).

\begin{algorithm}
\begin{algorithmic}
  \State \textbf{Input:} \( J : L^2(\Omega) \to \mathbb{R}, \Delta_{\text{max}} \in (0, \lambda(\Omega)), 0 < \sigma_1 < \sigma_2 \leq 1, \omega \in (0, 1) \).
  \State \textbf{Input:} \( U^0 \in \mathcal{B}, \Delta^0 \in (0, \Delta_{\text{max}}) \).
  \For {\( n = 0, \ldots \)}
  \State \( g^n \leftarrow \nabla J(\chi_{U^n})\chi_{(U^n)^c} - \chi_{U^n} \)
  \State \( D^n \leftarrow \text{FindStep} \left(g^n, \Delta^n, \min\{\omega, 5C_1(\chi_{U^n})/\lambda(\Omega), \Delta^n\}\right) \) // [5, Proc. 1]
  \If {\( J(\chi_{U^n \Delta D^n}) - J(\chi_{U^n}) \leq \sigma_1(\nabla J(\chi_{U^n}) \chi_{D^n \setminus U^n} - \chi_{U^n \cap D^n}) \chi_{R^n} \)}
    \State \( U^{n+1} \leftarrow U^n \Delta D^n \)
  \ElsIf {\( J(\chi_{U^n \Delta D^n}) - J(\chi_{U^n}) \leq \sigma_2(\nabla J(\chi_{U^n}) \chi_{D^n \setminus U^n} - \chi_{U^n \cap D^n}) \chi_{R^n} \)}
    \State \( \Delta^{n+1} \leftarrow \min\{2\Delta^n, \Delta_{\text{max}}\} \)
  \Else
    \State \( \Delta^{n+1} \leftarrow \Delta^n \)
  \EndIf
  \State \( U^{n+1}, \Delta^{n+1} \leftarrow U^n, 0.5\Delta^n \)
  \EndFor
\end{algorithmic}
\end{algorithm}

\textbf{Inputs of Algorithm 1} The algorithm requires the objective function \( J \) as well as its gradient \( \nabla J \) as inputs. Using the latter requires assuming differentiability of \( J \) with respect to the \( L^2 \)-norm. For the acceptance criterion of the computed descent step and the update of the trust-region radius, the algorithm requires a maximal trust-region radius \( \Delta_{\text{max}} \) and control parameters \( \sigma_1 \) and \( \sigma_2 \) as inputs. To compute a descent step, the algorithm uses the subroutine \text{FindStep}, which can be found as Procedure 1 in [5]. The subroutine also requires the parameter \( \omega \), which ensures that the volume of the returned set is always bounded from below by a fraction of \( \Delta^n \) that is smaller than 1. Moreover, the criticality measure \( C_1 \) is defined for a measurable set \( A \in \mathcal{B} \) and its characteristic function \( \chi_A \) as

\[ C_1(\chi_A) := \| \min\{g(\chi_A), 0\} \|_{L^1} = \int_\Omega |\min\{g(\chi_A)(s), 0\}| \, ds, \]

where the integrand \( g(\chi_A) \) is defined as

\[ g(\chi_A) := \nabla J(\chi_A)(\chi_{A^c} - \chi_A). \]

This implies the characterization of the integrand of \( C_1 \) for a.a. \( s \in \Omega \) as

\[ \min\{g(\chi_A)(s), 0\} = \begin{cases} -\nabla J(\chi_A)(s) & \text{if } s \in A \text{ and } \nabla J(\chi_A)(s) > 0, \\
\nabla J(\chi_A)(s) & \text{if } s \in A^c \text{ and } \nabla J(\chi_A)(s) < 0, \\
0 & \text{else}.\end{cases} \]

The criticality measure \( C_1 \) is induced by a local optimality condition for characteristic functions, which is shown in Proposition A.1 in the interest of completeness. It differs slightly from the norm of the gradient projected onto the box constraint—the common criticality measure for (R)—and their relationship is analyzed below (for the case of the \( L^1 \)-norm) too.
Trust-region subproblem The subroutine \textit{FindStep} in Line 3 of Algorithm 1 approximately solves the trust-region subproblem

$$\min_d \langle \nabla J(\chi_U), d \rangle_{L^2} \quad \text{s.t.} \quad \begin{cases} \chi_U(s) + d(s) \in \{0, 1\} \text{ for a.a. } s \in \Omega, \\ \|d\|_{L^1} \leq \Delta \end{cases} \quad \text{(TR(\Delta))}$$

for a given characteristic function \(\chi_U\) for \(U \in \mathcal{B}\) and trust-region radius \(\Delta > 0\). Because of the constraint \(\chi_U(s) + d(s) \in \{0, 1\}\) for a.a. \(s \in \Omega\) and the use of the \(L^1\)-norm in the trust-region constraint, it follows that optimizing for \(d\) in (TR(\Delta)) is equivalent to optimizing for a set \(D \in \mathcal{B}\) such that \(\chi_U \Delta D = \chi_U + d\) and \(\lambda(D) \leq \Delta\). Using min in the definition of (TR(\Delta)) is justified because (TR(\Delta)) indeed admits a minimizer, which is shown in Proposition A.3.

Description of steps of Algorithm 1 The for-loop starting in Line 1 of Algorithm 1 computes candidates for improvements of the objective function that are either accepted or rejected and then updates the trust-region radius accordingly.

Line 2 computes the function \(g^n\) such that it is equal to \(-\nabla J(\chi_{U^n})\) on \(U^n\), where \(\chi_{U^n}\) may be decreased, and such that it is equal to \(\nabla J(\chi_{U^n})\) on \((U^n)^c\), where \(\chi_{U^n}\) may be increased. Thus \(\{s \in \Omega \mid g^n(s) \leq 0\}\) is the set on which \(U^n\) can be changed to obtain a first-order decrease of \(J\). Because of the use of the \(L^1\)-norm, the trust-region radius \(\Delta^n\) limits the volume of the set that can be changed in the current iteration.

The subroutine \textit{FindStep} in Line 3 computes a set \(D^n \subset (g^n)^{-1}((-\infty, 0])\) that approximates

$$\inf_D \int_D g^n(s) \, ds \quad \text{s.t.} \quad D \subset (g^n)^{-1}((-\infty, 0])$$

The candidate for improving \(J\) is then the modification of the characteristic function of the set \(U^n\), where the values on \(D^n\) are flipped, or formally \(\chi_{U^n \Delta D^n}\). From Line 4 onward, the for-loop resembles common trust-region methods. Line 4 determines whether the reduction achieved by \(\chi_{U^n \Delta D^n}\) is at least a fraction of the reduction predicted by the linear model, in which case the step is accepted. A second (larger) ratio is used to determine whether the trust-region radius should be increased (doubled) or left unchanged after acceptance. The trust-region radius is reduced (halved) after rejection of a candidate step.

3 Convergence of BTR to First-Order Optimal Points

We introduce our main result. Algorithm 1 operates on iterates that are feasible for the integer problem \((P)\). We prove that our algorithm generates a sequence of integer feasible points whose limits are first-order optimal points of the relaxation \((R)\) and hence are optimal for \((P)\) if \((R)\) is a convex problem. We describe and discuss standing assumptions, followed by a suitable criticality measure, before presenting the main theorem.

3.1 Standing Assumptions

To state our main convergence results, we introduce a set of standing assumptions and discuss their respective roles in our analysis. We then show how these assumptions are reflected in the set-based arguments from [5] on BTR and also relate to their role in CIA [12].

Assumption 3.1.

1. Let \(J : L^2(\Omega) \to \mathbb{R}\) be bounded from below.
2. Let \(J : L^1(\Omega) \to \mathbb{R}\) be Fréchet differentiable.
3. Let \(\nabla J : L^1(\Omega) \to L^2(\Omega)\) be Lipschitz continuous.
4. Let \(\nabla J : L^2(\Omega) \to L^2(\Omega)\) be weak-norm continuous (completely continuous).
We note that assuming continuity with respect to the codomain in $L^2(\Omega)$ is well defined for our purpose because all feasible points and iterates are also $L^\infty$-functions.

Assumption 3.1 1 corresponds to the boundedness below of $J$ in the assumptions of [5, Theorem 2]. In our analysis this assumption can be weakened to asking for boundedness from below on the feasible set of (P). Assumption 3.1 2 corresponds to Assumption 1.4 from [5]. The local first-order Taylor expansion in [5, Theorem 1] for objective functions defined on measurable sets follows for the natural construction

$$J_s(A) := J(\chi_A), \quad J_s'(A)D := \langle J'(\chi_A), \chi_{D\setminus A} - \chi_{D\cap A} \rangle_{L^\infty(\Omega), L^1(\Omega)}$$

for $A, D \in \mathcal{B}$. We give a short proof in Proposition A.2.

Assumption 3.1 3 resembles Assumption 4.1 in [13] and implies the assumptions (5) in Lemma 3 and (10) in Theorem 2 in [5]. Assumptions of this type are required for the analysis of descent algorithms that manipulate binary control functions for the following reason. All binary control functions $v$ satisfy $\|v\|_{L^1} = \|v\|^2_{L^2}$. Therefore, bounding the error term of the Taylor expansion by the squared $L^2$-norm is not sufficient to obtain a sufficient decrease condition because the linear predicted reduction is bounded from below only by a fraction of the maximal $L^1$-norm of the trust-region step. Thus we cannot prove that the linear predicted reduction dominates the remainder terms for small trust-region radii without this further assumption. Because the trust-region subproblem does not allow fractional-valued control functions, a greedy strategy can always be used to approximate the infimal value of the trust-region subproblem regardless of the $L^p$-norm ($p \in [0, \infty]$) that is used for the trust-region radius. Consequently, this assumption cannot be avoided by choosing a different $L^p$-norm for the trust-region radius. The problem also arises for if $V$ is a finite subset of $\mathbb{R}$ or $\mathbb{Z}$, where one obtains $\|v\|_{L^1} = \Theta(\|v\|^2_{L^2})$ using Landau notation.

Assumption 3.1, 4 is a compactness assumption on the derivative of the objective function, which allows us to infer the first-order necessary optimality condition of an accumulation point. It resembles Assumption 1.4 for CIA in [12], where such an assumption is used on the objective functional itself. The reason for this subtle difference is that we need to pass to the limit in the derivative of the objective functional in the norm when analyzing the first-order necessary optimality condition.

Specifically, Assumption 3.1, 4 with Assumption 3.1, 2 yields weak continuity of $J : L^2(\Omega) \to \mathbb{R}$ (in particular on the feasible set of (R)), thereby establishing that (P) and (R) may be treated by CIA. This follows directly from the following proposition.

Proposition 3.1. Let Assumptions 3.1, 2 and 4 hold. Then $J : L^2(\Omega) \to \mathbb{R}$ is weakly continuous.

Proof. Let $x^n, x \in L^2(\Omega)$ be such that $x^n \to x$, meaning $x^n$ converging weakly to $x$, in $L^2(\Omega)$. We need to show $J(x^n) \to J(x)$. Assumption 3.1, 2 and the mean value theorem imply that $J(x^n) - J(x) = (\nabla J(\xi^n), x^n - x)_{L^2}$ for some $\xi^n \in L^2(\Omega)$ in the line segment between $x^n$ and $x$ for all $n \in \mathbb{N}$. Because $(\xi^n)_n$ is bounded, there exists a weakly convergent subsequence $\xi^{n_k} \to \xi$ for some $\xi \in L^2(\Omega)$. Assumption 3.1, 4 implies $\nabla J(\xi^{n_k}) \to \nabla J(\xi)$ in $L^2(\Omega)$, and consequently $J(x^{n_k}) - J(x) = (\nabla J(\xi^{n_k}), x^{n_k} - x)_{L^2} \to 0$. Passing to subsubsequences proves the claim.

Remark 3.1. Our setting and assumptions are more restrictive than the ones in [5], which is set in terms of general atomless measure spaces; see, e.g., [3, Definition 1.12.7], and an additional $L^1$-regularization of the control. We note, however, that our arguments do not hinge on the particular choice of the Lebesgue–Borel measure and generalize to more general Lebesgue spaces. Moreover, while the $L^1$-regularization does not satisfy Assumption 3.1 at first appearance, the required assertions actually hold for $L^1$-regularization of the feasible set of (R) with the choice $V = \{0, 1\}$.

3.2 Criticality Measure

Because we are interested in proving a first-order necessary optimality condition for (R), we define a usual criticality measure with respect to the $L^1$-norm for (R) as

$$C_2(f) := \|f - P_{[0,1]}(f - \nabla J(f))\|_{L^1} = \int_{\Omega} \left| f(s) - P_{[0,1]}(f(s) - \nabla J(f)(s)) \right| \, ds$$
for measurable functions $f : \Omega \to \mathbb{R}$, where $P_{[0,1]}$ denotes the superposition operator that is induced by the orthogonal projection of real scalars to the unit interval. If $f = \chi_A$ for some $A \in \mathcal{B}$, then we deduce that $C_2(\chi_A) = \int_{\Omega} |h(\chi_A)|$ holds with

$$h(\chi_A)(s) = \begin{cases} \min\{\nabla J(\chi_A)(s), 1\} & \text{if } s \in A \text{ and } \nabla J(\chi_A)(s) > 0, \\ \max\{\nabla J(\chi_A)(s), -1\} & \text{if } s \in A^c \text{ and } \nabla J(\chi_A)(s) < 0, \\ 0 & \text{else} \end{cases}$$

for a.a. $s \in \Omega$. Inspecting the definitions of $C_1$ and $C_2$, we observe that the relationship

$$C_2(\chi_A) \leq C_1(\chi_A) \quad (1)$$

holds for all measurable sets $A \in \mathcal{B}$.

**Definition 3.1.** A function $f$ that is feasible for (R) is called stationary if $C_2(f) = 0$ or if $f = \chi_A$ for some $A \in \mathcal{B}$ and $C_1(f) = 0$.

**Remark 3.2.** It is known that $C_2(f) = 0$ implies that $f$ satisfies a first-order necessary optimality condition for (R); see, e.g., [11, Lemma 1.12]. Moreover, the definition imply for $f = \chi_A$ for some $A \in \mathcal{B}$ that $C_1(f) = 0$ if and only if $C_2(f) = 0$.

### 3.3 Main Result

Having introduced the necessary notation, concepts, and assumptions, we now state our main convergence results.

**Theorem 3.1.** Let Assumptions 3.1 1-2 hold. Let $(U^n)_n \subset \mathcal{B}$, $(D^n)_n \subset \mathcal{B}$, and $(\Delta^n)_n \subset (0, \Delta_{\text{max}}]$ be the sequences of sets and trust-region radii produced by Algorithm 1. Then the sequence of objective values $(J(\chi_{U^n}))_n$ is monotonously nonincreasing. Moreover, one of the following mutually exclusive outcomes holds:

1. There exists $n_0 \in \mathbb{N}$ such that $U^n = U^\infty$ a.e. holds for all $n \geq n_0$. Then $\chi_{U^n}$ is stationary for (R).
2. For all $n_0 \in \mathbb{N}$ there exists $n_1 > n_0$ such that $\lambda(U^{n_1} \Delta U^{n_0}) > 0$. The sequence $(\chi_{U^n})_n \subset L^\infty(\Omega)$ admits a weak-*-accumulation point. Every weak-*-accumulation point $f$ of $(\chi_{U^n})_n$ is feasible for (R).

If additionally Assumption 3.1 3 holds, then

$$\liminf_{n \to \infty} C_1(\chi_{U^n}) = \liminf_{n \to \infty} C_2(\chi_{U^n}) = 0.$$ 

If additionally Assumption 3.1 4 holds and if a subsequence $(\chi_{U^{n_k}})_k \subset (\chi_{U^n})_n$ satisfies $C_2(\chi_{U^{n_k}}) \to 0$, then every weak-*-accumulation point $f$ of $(\chi_{U^{n_k}})_k$ satisfies $C_2(f) = 0$. There exists at least one weak-*-accumulation point $f$ of $(\chi_{U^n})_n$ such that $C_2(f) = 0$, $f$ is stationary.

For Outcome 2, we can strengthen these results and obtain convergence of the whole sequence.

**Theorem 3.2.** Let Assumptions 3.1 1-2 hold. Let $(U^n)_n \subset \mathcal{B}$, $(D^n)_n \subset \mathcal{B}$, and $(\Delta^n)_n \subset (0, \Delta_{\text{max}}]$ be the sequences of sets and trust-region radii produced by Algorithm 1. We consider the case that Outcome 2 of Theorem 3.1 holds for the produced sequence of iterates. Let Assumption 3.1 3 hold. Then

$$\lim_{n \to \infty} C_1(\chi_{U^n}) = \lim_{n \to \infty} C_2(\chi_{U^n}) = 0.$$ 

Theorems 3.1 and 3.2 are proven in §4. We obtain the following corollary that shows that Algorithm 1 produces a sequence of binary iterates that converge weakly-* to stationary points of the continuous relaxation (R) of (P). Thus BTR yields results comparable to those produced by CIA because solution algorithms for (R) cannot be expected to perform better than producing a stationary point of (R) in practice.

**Corollary 3.1.** Let Assumption 3.1 hold. Let $(U^n)_n \subset \mathcal{B}$ be the sequences of sets produced by Algorithm 1. Then all weak-*-accumulation points of $(\chi_{U^n})_n$ are feasible and stationary for (R).

**Proof.** The claim follows by combining Theorems 3.1 and 3.2. □
4 Proof of the Main Theorems

In this section we prove Theorems 3.1 and 3.2. We first prove preparatory results on the sufficient reduction condition (Algorithm 1, line 4) for binary-valued control functions and trust-region steps in §4.1. Then we employ these results to analyze the asymptotics of Algorithm 1 in §4.2, finishing with the proofs of Theorems 3.1 and 3.2.

4.1 Sufficient Decrease with a Characteristic Function

The first step of the proof is to show that if $\chi_{U^n}$ for a given iterate $U^n \in B$ of Algorithm 1 is not stationary, then there exists a set $D^n$ such that $\chi_{U^n \Delta D^n}$ satisfies a sufficient decrease condition with respect to $\nabla J(\chi_{U^n})$ for sufficiently small trust-region radii. We briefly recap the well-known result on existence of a descent direction in Lemma 4.1, which we then use to obtain a corresponding direction yielding a new binary-valued function in Lemma 4.2. This in turn implies acceptance of a new iterate after finitely many steps, as shown in Lemma 4.3.

Lemma 4.1. Let Assumption 3.1 2 hold. Let $f$ be feasible but not stationary for (R). Then there exist $\epsilon > 0$ and $\Delta_0 > 0$ such that for all $0 < \Delta \leq \Delta_0$ there exists $d \in L^2(\Omega)$ such that $f + d$ is feasible for (R), $\|d\|_L^2 \leq \Delta$, and $(\nabla J(f), d)_{L^2} \leq -\epsilon \Delta$.

Proof. Because $f$ is not stationary for (R), $C_2(f) > 0$, where $J^*(f) \in (L^1(\Omega))^*$ and $(L^1(\Omega))^* \cong L^\infty(\Omega)$ imply that $\nabla J(f)$ is well defined. Thus there exist $\epsilon > 0$ and $B \in B$ with $\lambda(B) > 0$ such that $|J^*(f) - \nabla J(f)(s)| > \sqrt{\epsilon}$ and $|\nabla J(f)(s)| > \sqrt{\epsilon}$ for a.a. $s \in B$. Then $d := (P_{[0,1]}(f - \nabla J(f)) - f)\chi_B$ satisfies $\int_{\Omega} \nabla J(f)(s) d(s) ds \leq -\epsilon \Delta_0$, where $\Delta_0 = \lambda(B)$. This also implies that $f + d$ is feasible for (R). Moreover, for all $0 < \Delta \leq \Delta_0$, we can choose a set $B_\Delta \subset B$ with $\lambda(B_\Delta) = \Delta$ and $d_\Delta := (P_{[0,1]}(f - \nabla J(f)) - f)\chi_{B_\Delta}$ to obtain that $f + d$ is feasible for (R) and $(\nabla J(f), d_\Delta)_{L^2} \leq -\epsilon \Delta$.

Lemma 4.2. Let Assumption 3.1 2 hold. Let $\chi_U$ for $U \in B$ be not stationary for (R). Then there exist $\epsilon > 0$ and $\Delta_0 > 0$ such that for all $0 < \Delta \leq \Delta_0$, there exists $d \in L^2(\Omega)$ that is feasible for (TR(\Delta)) and satisfies $(\nabla J(\chi_U), d)_{L^2} \leq -\epsilon \Delta$.

Proof. We apply Lemma 4.1 and obtain that there exist $\epsilon_1 > 0$ and $\Delta_1 > 0$ such that for all $0 < \Delta \leq \Delta_1$ there exist $d_\Delta \in L^2(\Omega)$ with $\|d_\Delta\|_{L^1} \leq \Delta$, $\chi_U(s) + d_\Delta(s) \in [0,1]$ for a.a. $s \in \Omega$, and $(\nabla J(\chi_U), d_\Delta)_{L^1} \leq -\epsilon_1 \Delta$.

We consider a sequence of $[-1,1]$-valued simple functions $(d_\Delta^n)_n$ that are defined on a uniformly refined uniform grid such that $d_\Delta^n \rightarrow d_\Delta$ in $L^2(\Omega)$. For all refined grids in this sequence, we may arrange the grids’ cells such the sequence of cells is an order-conserving domain dissection; see Definition A.1. Moreover, we define the coefficient functions

$$\alpha^n(s) := \begin{cases} (d_\Delta^n(s), 1 - d_\Delta^n(s), 0)^T & \text{if } d_\Delta^n(s) \geq 0, \\ (0, 1 + d_\Delta^n(s), -d_\Delta^n(s))^T & \text{otherwise} \end{cases}$$

for a.a. $s \in \Omega$. We then apply the sum-up rounding algorithm, which is defined in Algorithm 2, to this sequence of ordered sets of grid cells and the functions $\alpha^n$. The analysis in [19] implies that Algorithm 2 produces a sequence of $\{0,1\}^3$-valued functions $\omega^n$ that satisfy $\sum_{i=1}^3 \omega^n_i = 1$ a.e. such that the definition

$$d_\Delta^n := \omega^n_1 - \omega^n_3$$

yields $d_\Delta^n(s) \in \{-1,0,1\}$ for $s \in \Omega$, and $d_\Delta^n \rightarrow d_\Delta$ in $L^1(\Omega)$. Moreover, the construction of $\alpha^n$ implies that $\max\{d_\Delta^n, 0\}_{L^1} \rightarrow \max\{d_\Delta, 0\}_{L^1}$, and $\min\{d_\Delta^n, 0\}_{L^1} \rightarrow \min\{d_\Delta, 0\}_{L^1}$, which we prove in Lemma A.1. We give a description of Algorithm 2 and order-conserving domain dissections in §A.1.

Without affecting these properties, we may further modify $d_\Delta^n$ by setting $d_\Delta^n(s) := 0$ for $s$ if $\chi_U(s) + d_\Delta^n(s) \notin \{0,1\}$ because $d_\Delta^n \rightarrow d_\Delta$ in $L^1(\Omega)$, and $\chi_A + d_\Delta$ is feasible for (R). Combining this implies that $\chi_U + d_\Delta$ is feasible for (P) for all $n \in \mathbb{N}$ and

$$\|d_\Delta^n\|_{L^1} \rightarrow \|d_\Delta\|_{L^1} \text{ and } (\nabla J(\chi_U), d_\Delta^n)_{L^2} \rightarrow (\nabla J(\chi_U), d_\Delta)_{L^2}.$$
We use these insights to prove the claim. It remains to choose $\varepsilon$ and $\Delta_0$ such that the claim holds. Let $\delta < \frac{1}{4}$. We set $\Delta_0 := \Delta_1(1 + \delta)$ and $\varepsilon := \frac{3}{4(1 + \delta)}$. For all $0 < \Delta \leq \Delta_1$, we choose $\Delta = \Delta/(1 + \delta)$ by the reasoning above. Then there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$

$$\langle \nabla J(\chi_U), d^\Delta \rangle_{L^2} \leq -\frac{3}{4} \varepsilon_1 \Delta \leq -\frac{3}{4(1 + \delta)} \varepsilon_1 \Delta = -\varepsilon \Delta$$

and

$$\|d^\Delta\|_{L^1} \leq \|d\|_{L^1} + \Delta \frac{\delta}{1 + \delta} \leq \Delta \frac{\delta}{1 + \delta} + \Delta \frac{\delta}{1 + \delta} = \tilde{\Delta}.$$ 

This proves the claim.

We employ this result to prove that Algorithm 1 accepts a step after finitely many iterations if the current iterate is not stationary for (R).

**Lemma 4.3.** Let Assumptions 3.1, 2–3 be satisfied. Let $(U^n) \subset B$, $(D^n) \subset B$, and $(\Delta^n) \subset (0, \Delta_{\text{max}}]$ be the sequences of sets and trust-region radii produced by Algorithm 1. Let $\chi_{U^n}$ not be stationary for (R). Then the output of Algorithm 1, Line 3, is accepted after $k \in \mathbb{N}$ steps: specifically, $U^n = U^{n+j}$ for all $0 \leq j < k$ and $J(\chi_{U^n \Delta D^{n+j}}) - J(\chi_{U^n}) \leq \sigma_1(\nabla J(\chi_{U^n}), \chi_{D^{n+k} \setminus U^n} - \chi_{U^n \cap D^{n+k}})_{L^2}.$

**Proof.** For $m \in \mathbb{N}$, we define the optimal linear predicted reduction as

$$L^m := \inf_{d^m \in B} \left\{ \langle \nabla J(\chi_{U^m}), d^m \rangle_{L^2} \middle| d^m = \chi_{D^m \setminus U^m} - \chi_{D^m \cap U^m} \text{ and } \|d^m\|_{L^1} \leq \Delta^m \right\}.$$ 

By design of Algorithm 1, we have $\Delta^{n+j+1} = 0.5 \Delta^{n+j}$ for all $j \geq 0$ until the step $D^{n+j}$ is accepted.

We prove the claim by contradiction and assume that the step $D^{n+k}$ is not accepted for all $k \in \mathbb{N}$. Because $U^n$ is not stationary for (R) and $\Delta^{n+k} \to 0$ for $k \to \infty$, Lemma 4.2 implies that there exist $\varepsilon > 0$ and $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$ the estimate $L^{n+k} \geq \Delta^{k+j} \varepsilon$ holds.

We apply Taylor’s theorem and obtain that

$$J(\chi_U) - J(\chi_{U \Delta D^{n+k}}) = -\langle \nabla J(\chi_U), \chi_{D^{n+k} \setminus U} - \chi_{U \cap D^{n+k}} \rangle_{L^2} + o(\Lambda(D^{n+k}))$$

for all $D^{n+k} \in B$. This is satisfied in Algorithm 1, Line 3, because the parameter $\delta$ of FindStep, the third argument of the procedure, is given a value that is less than or equal to the trust-region radius in all iterations.

Because $L^{n+k} \geq \varepsilon \Delta^{n+k}$ holds for all $k \geq k_0$ and the two latter terms in (4) are $o(\Delta^{n+k})$ there exists $k_1 \in \mathbb{N}$ such that for all $k \geq k_1$ it holds that

$$J(\chi_U) - J(\chi_{U \Delta D^{n+k}}) \geq \sigma_1 L^{n+k}.$$ 

By definition of $L^{n+k}$ it follows that

$$J(\chi_U) - J(\chi_{U \Delta D^{n+k}}) \geq \sigma_1 L^{n+k} \geq -\sigma_1(\nabla J(\chi_U), \chi_{D^{n+k} \setminus U} - \chi_{U \cap D^{n+k}})_{L^2},$$

and thus the step $D^{n+k_1}$ is accepted in Algorithm 1. This contradicts the assumption that the step $D^{n+k}$ is not accepted for all $k \in \mathbb{N}$ and completes the proof.

**4.2 Asymptotics of Algorithm 1**

Before finalizing the proofs of Theorems 3.1 and 3.2, we show two further preparatory lemmas. Lemma 4.4 states that the sequence of iterates produced by Algorithm 1 has a corresponding sequence of monotonically nonincreasing objective values. Lemma 4.5 shows that if one of the criticality measures, $C_1$ or $C_2$, stays bounded away from zero over the iterations of Algorithm 1, then the trust-region radius contracts to zero.
Lemma 4.4. Let Assumption 3.1, 2 hold. Let \((U_n)_n, (D_n)_n \subset B\), and \((\Delta_n)_n \subset (0, \Delta_{\max})\) be the sequences of sets and trust-region radii produced by Algorithm 1. Then the sequence of objective values \((J(U_n))_n\) is monotonically nonincreasing.

Proof. By construction of \(D^n\) with FindStep, Procedure 1 of [5], it holds that \(D^n \subset \{s \in \Omega \mid g_{U^n}(s) < 0\}\). A step that is accepted in Algorithm 1 Line 4 satisfies \(J(\chi_{U^n, D^n}) < J(\chi_{U^n})\) because

\[
(\nabla J(\chi_{U^n}), \chi_{D^n \setminus U^n} - \chi_{U^n \cap D^n})_L^2 = \int_{D^n} g_{U^n} \ ds < 0,
\]

while \(J(\chi_{U^n})\) remains unchanged for rejected steps. Thus, the sequence of objective values \((J(\chi_{U^n}))_n\) is monotonically nonincreasing. \(\Box\)

Lemma 4.5. Let Assumptions 3.1, 1–2 hold. Let \((U_n)_n \subset B, (D_n)_n \subset B, and (\Delta_n)_n \subset (0, \Delta_{\max})\) be the sequences of sets and trust-region radii produced by Algorithm 1. If there exists \(\epsilon > 0\) and \(n_0 \in \mathbb{N}\) such that \(C_1(\chi_{U^n}) > \epsilon\) or \(C_2(\chi_{U^n}) > \epsilon\) for all \(n \geq n_0\), then \(\Delta^n \to 0\).

Proof. It is sufficient to show the claim for the case \(C_1(\chi_{U^n}) > \epsilon\) because of (1). We use the notation \(L^n\) for the optimal predicted reduction in iteration \(m \in \mathbb{N}\) as in the proof of Lemma 4.3. From Proposition A.3 and the definition of \(C_1\) it follows that \(L^n \geq C_1(U^n) \Delta^n_{U^n}\) for all iterations \(m \in \mathbb{N}\). This can be seen by using the greedily constructed set.

From Algorithm 1 Line 3, and \(\Delta_{\max} \leq \lambda(\Omega)\) it follows that the third parameter \(\delta\) of the subroutine FindStep satisfies \(\delta \leq C_1(U^n)/(2\Delta^n)\). The analysis of FindStep, specifically [5, Lemma 6], implies that

\[
-J(\chi_{U^n}), \chi_{D^n \setminus U^n} - \chi_{U^n \cap D^n})_L^2 \geq L^n - \min\{\omega, C_1(U^n)/(2\lambda(\Omega)), \Delta^n\} \Delta^n \\
\geq C_1(\chi_{U^n}) \frac{\Delta^n}{\lambda(\Omega)} - \frac{1}{2} C_1(\chi_{U^n}) \frac{\Delta^n}{\lambda(\Omega)} \geq \frac{\epsilon \Delta^n}{2\lambda(\Omega)}
\]

for all \(n \in \mathbb{N}\) and thus all \(n \geq n_0\).

We close the proof with a contradictory argument and assume that \(\Delta^n \neq 0\). We deduce that there exists an infinite subsequence \((\Delta^n)_k\) of \((\Delta^n)_n\) such that \(\liminf_{k \to \infty} \Delta^n_k \geq \Delta\) for some \(\Delta > 0\). Consequently, there exists an infinite subsequence \((n_k)_\ell\) of accepted iterates with trust-region radii \(\Delta^n_{\ell} \geq \Delta\).

Combining these insights on the accepted iterates with the lower bound on the linearly predicted reductions, we obtain that

\[
J(\chi_{U^n_{\ell}}) - J(\chi_{U^n_{\ell}, D^n_{\ell}}) \geq \sigma \frac{\epsilon \Delta}{4\lambda(\Omega)}.
\]

Because the sequence of objective values is monotonically nonincreasing by virtue of Lemma 4.4, this implies \(J(\chi_{U^n_{\ell}}) \to -\infty\), which contradicts Assumption 3.1 1 and thus the assumption that \(\Delta^n \neq 0\). This closes the proof. \(\Box\)

We are now ready to finish the proof of the two main results.

Proof of Theorem 3.1. Lemma 4.4 proves the claim that the sequence of objective values is monotonically nonincreasing.

We first analyze Outcome 1. Because there exists \(n_0 \in \mathbb{N}\) such that \(U^{n_0} = U^n\) holds a.e. for all \(n \geq n_0\), the acceptance criterion in Algorithm 1, Line 4, is violated for all \(n \geq n_0\). Then the claim of Outcome 1 follows from Lemma 4.3.

If there is no \(n_0 \in \mathbb{N}\) such that \(U^{n_0} = U^n\) holds a.e. for all \(n \geq n_0\), then Outcome 1 does not hold true, and for all \(n \in \mathbb{N}\) there exists \(n_1 > n_0\) such that \(\lambda(U^{n_1}, D,U^{n_0}) > 0\). It follows that Outcomes 1 and 2 are mutually exclusive. Moreover, the sequence \((\chi_{U^n})_n \subset L^\infty(\Omega)\) is bounded and thus admits a weak-* cluster point. By virtue of, for example, [25, Theorem 3], every weak-* cluster point of \((\chi_{U^n})_n\) is feasible for \((R)\).

Next we assume that Assumptions 3.1 1–3 are satisfied and prove the claim \(\liminf_{n \to \infty} C_1(\chi_{U^n}) = \liminf_{n \to \infty} C_2(\chi_{U^n}) = 0\) for Outcome 2. Because of (1) it suffices to show that \(\liminf_{n \to \infty} C_1(\chi_{U^n}) = 0\).
We consider the subsequence of accepted iterates (successful steps) \((n_k)_k\) of Algorithm 1. Using the fundamental theorem of calculus and the notation \(d^n := \chi_{U^n \Delta D^n} - \chi_{U^n}\) for \(n \in \mathbb{N}\), we may rewrite the decrease in the objective as

\[
J(\chi_{U^n}) - J(\chi_{U^n \Delta D^n}) = -(\nabla J(\chi_{U^n}), d^n)_{L^2} - \int_0^1 (\nabla J(\chi_{U^n} + td^n) - \nabla J(\chi_{U^n}), d^n)_{L^2} \, dt.
\]

We observe that \(d^n(s) \in \{-1,0,1\}\) for a.a. \(s \in \Omega\) implies \(\sqrt{||d^n||_{L^2}} = ||d^n||_{L^2}\). The Lipschitz continuity of \(\nabla J : L^1(\Omega) \rightarrow L^2(\Omega)\) (Assumption 3.1 3) with Lipschitz constant \(L > 0\) implies

\[
J(\chi_{U^n}) - J(\chi_{U^n \Delta D^n}) \geq -(\nabla J(\chi_{U^n}), d^n)_{L^2} - \frac{L}{2} ||d^n||_{L^1} \sqrt{||d^n||_{L^1}}.
\]

As in the proof of Lemma 4.5 we observe that the estimate

\[-(\nabla J(\chi_{U^n}), d^n) \geq C_1(\chi_{U^n}) \frac{\Delta^n}{2\lambda(\Omega)}\]

holds for the steps \(d^n\). Inserting this estimate yields

\[
J(\chi_{U^n}) - J(\chi_{U^n \Delta D^n}) \geq -(\nabla J(\chi_{U^n}), d^n)_{L^2} - (1 - \sigma_1) \frac{C_1(\chi_{U^n})\Delta^n}{2\lambda(\Omega)} - \frac{L}{2} (\Delta^n)^{\frac{3}{2}}.
\]

We show \(\liminf_{n \to \infty} C_1(\chi_{U^n}) = 0\) by contradiction. If \(\liminf_{n \to \infty} C_1(\chi_{U^n}) > 0\), then Lemma 4.5 implies that \(\Delta^n \to 0\) and \(r^n \geq 0\) holds for all \(n \geq n_2\) for some \(n_2 \in \mathbb{N}\). But \(r^n \geq 0\) implies that the acceptance criterion in Algorithm 1.4 is satisfied for all \(n \geq n_2\) and the trust-region radius is not decreased further from iteration \(n_2\) on. This contradicts \(\Delta^n \to 0\) and proves the claim.

For the remainder of the proof, we restrict ourselves to a subsequence of \((\chi_{U^n})_n\), for ease of notation denoted by the same symbol, which satisfies \(C_2(\chi_{U^n}) \to 0\). It remains to show that every weak-* cluster point \(f\) of \((\chi_{U^n})_n\) satisfies

\[
0 = C_2(f) = \int_{\Omega} \left| f(s) - P_{[0,1]}(f(s) - \nabla J(f)(s)) \right| \, ds
\]

if Assumptions 3.1 1–4 hold. We show the claim by verifying that the integral in (5) is zero when integrating over suitable measurable subsets of \(\Omega\) that correspond to level sets of \(f\). We start by considering the sets \(A_0 = f^{-1}(\{0\})\) and \(A_1 = f^{-1}(\{1\})\). The convergence \(\chi_{U^n} \rightharpoonup^* f\) yields

\[
\|\chi_{U^n} - f\|_{L^1(A_0)} = \int_{\Omega} \chi_{A_0}(s) \chi_{U^n}(s) \, ds \to 0
\]

and

\[
\|f - \chi_{U^n}\|_{L^1(A_1)} = \int_{\Omega} \chi_{A_1}(s)(1 - \chi_{U^n}(s)) \, ds \to 0.
\]

Because the projection \(P_{[0,1]}\) is Lipschitz continuous, its induced superposition operator is also Lipschitz continuous (from \(L^1\) to \(L^1\)). Moreover, \(\nabla J\) is completely continuous, and thus we obtain the convergence \(\chi_{U^n} - \nabla J(\chi_{U^n}) \rightharpoonup f - \nabla J(f)\) in \(L^1(A_0 \cup A_1)\), which gives

\[
\int_{A_0 \cup A_1} \left| \chi_{U^n}(s) - P_{[0,1]}(\chi_{U^n}(s) - \nabla J(\chi_{U^n}))(s) \right| \, ds \to \int_{A_0 \cup A_1} \left| f(s) - P_{[0,1]}(f(s) - \nabla J(f)(s)) \right| \, ds.
\]

For \(\delta > 0\), we consider \(A_{\delta} := f^{-1}(\{\delta, 1 - \delta\})\). Let \(\varepsilon > 0\) and \(B_{\varepsilon} \subset A_{\delta}\) be such that \(\nabla J(f)(s) > \varepsilon\) for a.a. \(s \in B_{\varepsilon}\). We consider the part of \(C_2(f)\) corresponding to \(B_{\varepsilon}\), \(\int_{B_{\varepsilon}} \left| \chi_{U^n}(s) - P_{[0,1]}(\chi_{U^n}(s) - \nabla J(\chi_{U^n}))(s) \right| \, ds.\)
Because \( \nabla J(\chi_{U^n}) \to \nabla J(f) \) holds by virtue of Assumption 3.1, 4, we may restrict ourselves to a pointwise a.e. convergent subsequence (for ease of notation denoted by the same symbol). Then we apply Egorov’s theorem [3, Theorem 2.2.1] to deduce that there exists \( n_0 \in \mathbb{N} \) such that for all \( n \geq n_0 \) it holds that

\[
\operatorname{ess sup}_{s \in B_\varepsilon \setminus S^n} P_{[0,1]}(\chi_{U^n}(s) - \nabla J(\chi_{U^n})(s)) \leq \operatorname{ess sup}_{s \in B_\varepsilon \setminus S^n} P_{[0,1]} \left( \chi_{U^n}(s) - \frac{\varepsilon}{2} \right),
\]

where \( S^n \subset B_\varepsilon \) with \( \lambda(S^n) \to 0 \) and \( \operatorname{ess sup} \) denotes the essential supremum [3, Supplement 2.12.83]. Thus without loss of generality it holds that that \( \lambda(S^n) < \frac{\varepsilon^2}{8} \lambda(B_\varepsilon) \). Because \( f(s) \geq \delta \) for a.a. \( s \in \Omega \), we can choose \( n_0 \) large enough such that \( \int_{B_\varepsilon} \chi_{U^n}(s) \, ds > \frac{\delta}{8} \lambda(B_\varepsilon) \). We combine these estimates to deduce

\[
\int_{B_\varepsilon} \left| \chi_{U^n}(s) - P_{[0,1]}(\chi_{U^n}(s) - \nabla J(\chi_{U^n})(s)) \right| \, ds \geq 0,
\]

for all \( n \geq n_0 \). The same arguments apply for subsets \( B_\varepsilon \subset A_3 \) such that \( \nabla J(f) < -\varepsilon \).

Now assume that we operate on a subsequence such that \( C_2(\chi_{U^n}) \to 0 \). Then, the arguments above imply

\[
\int_{A_0 \cup A_1} \left| f(s) - P_{[0,1]}(f(s) - \nabla J(f)(s)) \right| \, ds = 0.
\]

Moreover, if \( \lambda((A_0 \cup A_1) \setminus \Omega) > 0 \) holds, then there exists \( \delta > 0 \) such that \( \lambda(A_3) > 0 \). The estimate above implies that an \( \varepsilon > 0 \) such that \( |\nabla J(f)(s)| > \varepsilon \) holds a.e. on such a subset of \( A_3 \) cannot exist. This implies that \( \nabla J(f)(s) = 0 \) a.e. on \( A_3 \), which in turn yields \( C_2(f) = 0 \).

**Proof of Theorem 3.2.** To prove the claim, we follow the proof strategy of [26, Theorem 6]. We say that \( A_0 \) is a successful iteration of Algorithm 1 if the acceptance test in Line 4 is successful. Let \( S \subset \mathbb{N} \) denote the set of successful iterations of Algorithm 1. We observe that any successful iteration satisfies

\[
J(\chi_{U^n}) - J(\chi_{U^{n+1}}) \geq \frac{1}{2\lambda(\Omega)} C_1(\chi_{U^n}) \Delta^n
\]

because of the properties of the subroutine \textbf{FindStep} (see the proof of Lemma 4.5).

This implies

\[
J(\chi_{U^0}) - J(\chi_{U^{n+1}}) \geq \frac{1}{2\lambda(\Omega)} \sum_{\ell=0}^n C_1(\chi_{U^\ell}) \Delta^\ell
\]

for all \( n \in \mathbb{N} \). We seek for a contradiction to the claim and assume that there exists a subsequence \( (n_k)_k \subset S \) such that

\[
C_1(\chi_{U^{n_k}}) \geq 2\varepsilon > 0 \tag{6}
\]

for some \( \varepsilon > 0 \). Let \( K := \{ n \in S \mid C_1(\chi_{U^n}) \geq \varepsilon \} \). It follows that

\[
J(\chi_{U^n}) - J(\chi_{U^{n+1}}) \geq \frac{1}{2\lambda(\Omega)} \sum_{\ell=0}^n C_1(\chi_{U^\ell}) \Delta^\ell \geq \frac{1}{2\lambda(\Omega)} \varepsilon \sum_{\ell=0}^n \Delta^\ell
\]

for all \( n \in \mathbb{N} \). Let \( n_0 \in \mathbb{N} \). Then we obtain for all \( n \geq n_0 \) that

\[
\sum_{\ell=n_0}^n \Delta^\ell \leq \frac{2\lambda(\Omega)}{\varepsilon} \left( J(\chi_{U^n}) - J(\chi_{U^{n+1}}) \right) \leq \frac{2\lambda(\Omega)}{\varepsilon} \left( J(\chi_{U^0}) - \min(R) \right) < \infty,
\]

11
which implies that \( \sum_{k=0}^{\infty} \Delta^k < \kappa < \infty \) for some \( \kappa > 0 \) for all \( n_0 \in \mathbb{N} \). From \( \liminf_{n \to \infty} C_1(\chi_{U_n}) = 0 \) it follows for all \( k \in \mathbb{N} \) that there exists a smallest \( \ell(k) > n_k \) with \( \ell(k) \in S \setminus K \). We obtain

\[
\|\chi_{U^{(k)}} - \chi_{U^{\ell(k)}}\|_{L^1} \leq \sum_{j=n_k}^{\ell(k)-1} \|\chi_{U^{j+1}} - \chi_{U^j}\|_{L^1} \\
\leq \sum_{j=n_k}^{\ell(k)-1} \Delta^j = \sum_{j=n_k}^{\ell(k)-1} \Delta^j \leq \sum_{j=n_k}^{\infty} \Delta^j < \kappa.
\]

This implies that for \( k \to \infty \) we obtain that

\[
\|\chi_{U^{(k)}} - \chi_{U^{n_k}}\|_{L^1} \to 0.
\]

By virtue of Fatou’s lemma and the fact that every sequence that converges in \( L^1 \) has a pointwise a.e. convergent subsequence, we obtain

\[
\|\chi_{U^{(k)}} - \chi_{U^{n_k}}\|_{L^2} \to 0
\]

for a subsequence of \( (n_k)_k \), which we denote with the same symbol for ease of notation. We conclude that

\[
|C_1(\chi_{U^{(k)}}) - C_1(\chi_{U^{n_k}})| \to 0 \quad \text{(for } k \to \infty),
\]

which violates our assumption (6). Hence \( C_1(\chi_{U^{n_k}}) \to 0 \). The claim for \( C_2 \) follows from (1).

\[
\square
\]

5 Computational Validation

We now report results of computational experiments to validate our theoretical results.

5.1 Example Satisfying Assumption 3.1

We consider the case \( d = 2 \) and a bounded Lipschitz domain \( \Omega \subset \mathbb{R}^2 \). For (P) we choose \( J(x) := j(S(x)) \) and \( V = \{0,1\} \), where \( j \) is a so-called tracking-type objective, specifically \( j(y) = 0.5\|y - y_d\|_{L^2}^2 \) for a given \( y_d \in L^2(\Omega) \), and \( S \) is the solution operator of the Poisson problem

\[
- \Delta y = x, \quad y|_{\partial \Omega} = 0
\]

for a given control input \( x \in L^1(\Omega) \). This yields the following instance of (P):

\[
\inf_x \frac{1}{2}\|y - y_d\|_{L^2}^2 \quad \text{s.t.} \quad y = S(x) \quad \text{and} \quad x(s) \in \{0,1\} \quad \text{for a.a.} \ s \in \Omega.
\]

(8)

For a Poisson problem with right-hand side in \( L^1(\Omega) \) we have that the weak solution \( y \) is an element of the Sobolev space of \( q \)-integrable functions with a \( q \)-integrable distributional derivative that vanish at the boundary, \( W^1_0(\Omega) \), for bounded, Lipschitz domains \( \Omega \), where we have the estimate \( \|u\|_{W^1_0(q)(\Omega)} \leq c\|u\|_{L^1(\Omega)} \) for some \( c > 0 \) if \( q < 2 \); see, for example, [2, Theorem 1] and [6] for the case of mixed boundary conditions.

Combining these considerations with the chain rule for Banach spaces and the Riesz representation theorem implies that Assumptions 3.1 1-3 are satisfied for this example. To see that Assumption 3.1 4 is also satisfied, we consider the compact embedding \( W^1_0(\Omega) \hookrightarrow W^{1,1}(\Omega) \hookrightarrow L^2(\Omega) \) (for \( d = 2 \)), where the compactness is due to the second embedding. Because \( S'(\bar{x}) \) is linear and bounded for \( \bar{x} \in L^1(\Omega) \), it maps weakly convergent sequences to weakly convergent sequences in \( W^1_0(\Omega) \) and, by compactness, to norm convergent sequences in \( L^2(\Omega) \), which implies that \( S' \) and in turn \( \nabla J \) are weak-norm continuous.

5.2 Experimental Setup

In all computational experiments the problem domain is \( \Omega := (0,1)^2 \) and \( y_d : [0,1]^2 \to \mathbb{R} \) is given by

\[
y_d(x) := \frac{2}{5} x_1 x_2 \cdot (1 - x_1) \cdot (1 - x_2) \cdot \sin(\pi \cdot \|x - (0.5, 0.5)^T\|) \quad \text{for all} \ x \in [0,1]^2.
\]
We start all iterative optimization methods with the constant zero function, \( f = 0 \), as an initial solution. We solve the boundary value problem (7) numerically using a finite element method on a conforming unstructured triangle mesh subdividing the domain \( \Omega \). We optimize the control function \( f \) in the space of cellwise constant discontinuous functions. The solution \( y = S(f) \) of (7) is computed in the space of cellwise linear continuous functions.

We apply four algorithms to solve (8): a general-purpose algorithm for nonlinear optimization ("NLP"), where the constraint \( f(x) \in \{0, 1\} \) is relaxed to \( f(x) \in [0, 1] \); the sum-up rounding procedure ("SUR"); the binary trust-region algorithm ("BTR"), where the mesh is refined in the procedure \( \text{FindStep} \) in Line 3 as described in [5]; and a simplified version of BTR ("BSTR") that does not refine the mesh and terminates when its trust-region radius becomes less than the volume of the smallest cell with an associated negative gradient. Moreover, we run each algorithm for a sequence of uniformly refined starting meshes. For each starting mesh, these methods are applied in the following order: (1) NLP on the starting mesh; (2) SUR on the starting mesh to approximate solution from Step 5.2; (3) BSTR on the starting mesh; (4) BTR initialized with the starting mesh, ; (5) NLP on the refined mesh (= final mesh of BTR), ; (6) SUR on the refined mesh (= final mesh of BTR) to approximate solution from Step 5.2; and (7) BSTR on the refined mesh (= final mesh of BTR). These choices ensure that the mesh refinement is consistent for all methods.

We note that we do not control the objective and adjoint error through adaptive mesh refinements but instead compare the outcomes for starting meshes ranging from \( 1 \times 1 \) to \( 256 \times 256 \) to assess whether the numerical accuracy is sufficient for our results to be meaningful.

To make our computational experiment repeatable, we provide detailed implementation information in §B.

5.3 Results

We list the objective function values and final \( C_2 \) values yielded by each algorithm in table 1. While BTR is started on the starting meshes, it computes a refined mesh in its iterations. Therefore, we report BTR’s objective and final \( C_2 \) value with respect to the refined mesh because the final result of BTR is achieved on the refined mesh.

For initial meshes \( 16 \times 16 \) and finer, the objective function value produced by the BTR algorithm varies by less than \( 5 \cdot 10^{-9} \). A similar stability in the objective function is reached by the relaxed solver for initial meshes \( 32 \times 32 \) and finer. We take this behavior as an indication that at those resolutions, meshes are sufficiently fine for the results to be largely unaffected by numerical error.

The objective function values achieved by both NLP and SUR on refined meshes and initial meshes \( 32 \times 32 \) and finer are lower than the ones achieved by BTR. This result is unsurprising because the NLP solver produces an optimal solution of the continuous relaxation for the fixed mesh, which is then approximated by SUR with a guaranteed approximation quality, while the BTR solution is not guaranteed to be optimal for the given discretization. This can also be observed visually in fig. 1, which shows the controls produced by all algorithms on the \( 256 \times 256 \) starting mesh.

We also observe in fig. 1 that SUR produces a chattering control with little discernible structure and a granularity close to that of the mesh, while BTR and BSTR produce microstructures at a scale much larger than that of the individual mesh cells.

The results in table 1 also show that BTR produces lower objective values on both initial and refined meshes to resolution \( 32 \times 32 \). At resolution \( 64 \times 64 \), BSTR produces a better objective but achieves worse \( C_2 \) values. In fig. 2 we can see that BSTR does not reach the \( C_1 \) threshold used to stop BTR in these instances.

To better compare the performance of both algorithms, we turn our attention to fig. 2, which compares the progression of \( C_1 \) over the iterations.

We first note that for the instances shown, BTR is very consistent in the number of iterations required to reach its \( C_1 \) threshold. However, starting as early as the refined \( 8 \times 8 \) and \( 16 \times 16 \) meshes, BSTR achieves lower \( C_1 \) values in fewer iterations. This is in addition to the fact that individual iterations are much faster in BSTR because mesh refinement and the associated reassembly of large linear operators are not needed. We take this behavior as an indication that significant room remains for improvement in BTR’s mesh refinement and the criteria by which it is triggered.

Nevertheless, we note that mesh refinement does allow BTR to produce plausible objective function values from extremely coarse meshes such as the \( 1 \times 1 \) mesh, whereas BSTR requires either a prerefined
Table 1: Final objective function values and $C_2$ values for $n \times n$ starting meshes and the refined meshes generated by BTR. We use $C_2$ values instead of $C_1$ because they are applicable to both relaxed and binary solutions. Note that none of the algorithms’ stopping criteria are directly based on $C_2$.

| $n$  | NLP     | SUR     | BSTR    | BTR     |
|------|---------|---------|---------|---------|
|      | Obj. Initial | 1.694e-05 | 1.701e-05 | 1.701e-05 |
|      | Obj. Refined | 2.236e-06 | 2.237e-06 | 2.293e-06 | 2.240e-06 |
|      | $C_2$ Initial | 2.776e-16 | 4.673e-06 | 4.673e-06 |
|      | $C_2$ Refined | 6.06e-09 | 3.405e-07 | 8.421e-07 | 4.810e-08 |
| 2    | Obj. Initial | 8.606e-06 | 9.151e-06 | 1.988e-05 |
|      | Obj. Refined | 2.235e-06 | 2.236e-06 | 2.275e-06 | 2.246e-06 |
|      | $C_2$ Initial | 0.000e+00 | 3.881e-05 | 3.212e-05 |
|      | $C_2$ Refined | 5.06e-09 | 1.486e-07 | 4.566e-07 | 4.820e-08 |
| 4    | Obj. Initial | 9.158e-06 | 8.559e-06 | 1.008e-05 |
|      | Obj. Refined | 2.248e-06 | 2.251e-06 | 2.279e-06 | 2.256e-06 |
|      | $C_2$ Initial | 4.891e-05 | 9.737e-06 | 2.457e-05 |
|      | $C_2$ Refined | 4.211e-09 | 7.669e-07 | 3.919e-07 | 4.805e-08 |
| 8    | Obj. Initial | 2.354e-06 | 2.433e-06 | 2.829e-06 |
|      | Obj. Refined | 2.282e-06 | 2.282e-06 | 2.295e-06 | 2.292e-06 |
|      | $C_2$ Initial | 3.963e-09 | 2.783e-06 | 2.333e-06 |
|      | $C_2$ Refined | 4.523e-09 | 1.156e-07 | 2.047e-07 | 4.813e-08 |
| 16   | Obj. Initial | 2.324e-06 | 2.330e-06 | 2.483e-06 |
|      | Obj. Refined | 2.308e-06 | 2.309e-06 | 2.327e-06 | 2.319e-06 |
|      | $C_2$ Initial | 4.023e-09 | 4.422e-07 | 1.200e-06 |
|      | $C_2$ Refined | 3.066e-09 | 3.354e-07 | 2.902e-07 | 4.818e-08 |
| 32   | Obj. Initial | 2.315e-06 | 2.315e-06 | 2.330e-06 |
|      | Obj. Refined | 2.313e-06 | 2.313e-06 | 2.330e-06 | 2.323e-06 |
|      | $C_2$ Initial | 4.735e-09 | 9.249e-08 | 2.607e-07 |
|      | $C_2$ Refined | 4.276e-09 | 1.046e-07 | 2.442e-07 | 4.774e-08 |
| 64   | Obj. Initial | 2.313e-06 | 2.313e-06 | 2.317e-06 |
|      | Obj. Refined | 2.313e-06 | 2.313e-06 | 2.318e-06 | 2.324e-06 |
|      | $C_2$ Initial | 3.989e-09 | 8.170e-08 | 5.612e-08 |
|      | $C_2$ Refined | 5.016e-09 | 1.868e-08 | 6.898e-08 | 4.816e-08 |
| 128  | Obj. Initial | 2.312e-06 | 2.312e-06 | 2.313e-06 |
|      | Obj. Refined | 2.312e-06 | 2.312e-06 | 2.313e-06 | 2.323e-06 |
|      | $C_2$ Initial | 4.265e-09 | 7.120e-09 | 1.563e-08 |
|      | $C_2$ Refined | 3.628e-09 | 3.682e-09 | 1.317e-08 | 4.812e-08 |
| 256  | Obj. Initial | 2.312e-06 | 2.312e-06 | 2.312e-06 |
|      | Obj. Refined | 2.312e-06 | 2.312e-06 | 2.312e-06 | 2.323e-06 |
|      | $C_2$ Initial | 3.055e-09 | 1.814e-09 | 2.986e-09 |
|      | $C_2$ Refined | 3.455e-09 | 3.094e-09 | 2.974e-09 | 4.759e-08 |
Figure 1: Controls produced by our new algorithm BTR (1a) and, for comparison, NLP, SUR, and BSTR for the uniform initial $256 \times 256$ mesh (1b) and the refined output mesh of BTR (1c). BTR and BSTR produce solutions with similar structures whose granularity is coarser than that produced by SUR. Those structures remain mesh-dependent, as we can see by comparing the BSTR solutions in (1b) and (1c).
Figure 2: Progression of $C_1$ over iterations for both the BTR and BSTR algorithms for different starting meshes. BSTR achieves worse $C_1$ values than does BTR up to the $64 \times 64$ starting mesh despite not being subject to a $C_1$-based termination criterion. For $256 \times 256$, BSTR clearly outperforms BTR even in early iterations, indicating that BTR’s refinement strategy leaves room for improvement. The dashed horizontal line indicates the stopping threshold of $5 \cdot 10^{-8}$ for BTR.
input mesh or a comparatively fine starting mesh with, in our case, a $64 \times 64$ mesh of squares divided into four triangles each, giving $64 \times 64 \times 4 = 16,384$ cells to achieve the same or better results.

Acknowledgments

We thank Peter Bella and Christian Meyer (both TU Dortmund University) for helpful discussions on the topic. This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through the Advanced Computing (SciDAC) Program through the FASTMath Institute under Contract No. DE-AC02-06CH11357 and by the German Research Foundation under GRK 2297 MathCoRe (project No. 314838170) and SPP 1962 (projects No. SA 2016/1-2, KI 1839/1-2) and by the German Federal Ministry of Education and Research within the program “Mathematics for Innovations” under the project “Power to Chemicals.”

A Auxiliary Results

For the sake of completeness, this appendix reviews the SUR algorithm and its properties, establishes relationships between set-based and characteristic function, and proves the existence of minimizers of the trust-region subproblem.

A.1 Sum-Up Rounding Algorithm and Its Basic Properties

Here we describe the multidimensional variant [19] of the sum-up rounding algorithm [21, 23] that is used for constructing the sequence $\tilde{d}^N_\Delta$ in the proof of Lemma 4.2.

The algorithm starts from a $[0,1]^m$-valued simple function $\alpha$ such that $\sum_{i=1}^m \alpha_i = 1$ a.e. and that is defined on an ordered sequence of grid cells that partition the domain $\Omega$. The algorithm iterates over the grid cells in the given order, identifies an entry $i \in \{1,\ldots,m\}$ such that the cumulative difference up to the current grid cell to a rounded function $\omega$, which is $\{0,1\}^m$-valued, satisfies $\sum_{i=1}^m \omega_i = 1$ a.e., and is defined on the same grid as $\alpha$, is maximal. Then the algorithm sets $\omega_i$ to one on the current grid cell and the other entries to zero on the respective grid cell. It is stated as Algorithm 2 below.

Algorithm 2 Sum-Up Rounding (multidimensional variant) [19]

Input: Ordered grid cells $S_1, \ldots, S_N \subset \Omega$ that partition $\Omega$.

Input: Simple function $\alpha \in L^1(\Omega,\mathbb{R}^m)$ with coefficients $a_{k,i}$ such that $\alpha_i = \sum_{k=1}^N a_{k,i} \chi_{S_k}$ for all $i \in \{1,\ldots,m\}$, and $\sum_{i=1}^m a_{k,i} = 1$ for all $k \in \{1,\ldots,N\}$.

1: $\phi_0 := 0_{\mathbb{R}^m}$
2: for $k = 1,\ldots,N$ do
3: $\gamma_k \leftarrow \phi_{k-1} + a_k \lambda(S_k)$ (where $\lambda$ is short for the vector $(a_{k,1},\ldots,a_{k,m})^T$)
4: $w_{k,i} \leftarrow \begin{cases} 1: & i \in \arg \max \{\gamma_k,j \mid j \in \{1,\ldots,m\}\} \\ 0: & \text{else} \end{cases}$ for all $i \in \{1,\ldots,m\}$
5: $\phi_k \leftarrow \sum_{i=1}^k (a_k - w_k) \lambda(S_k)$ (where $w_k$ is short for the vector $(w_{k,1},\ldots,w_{k,m})^T$)
6: end for
7: return $\omega := \sum_{k=1}^N w_k \chi_{S_k}$

In the construction in the proof of Lemma 4.2 we execute Algorithm 2 repeatedly on an order-conserving domain dissection, which allows us to obtain approximations in the weak-$^*$-topology of $L^\infty(\Omega)$ [19] and is defined next.

Definition A.1 (Order-conserving domain dissection, Definition 4.3 in [19]). Let $\Omega \subset \mathbb{R}^d$. Then we call a sequence $\{S^1_n, \ldots, S^N_n\}_n \subset 2^\mathbb{R}^d$ an order-conserving domain dissection if

1. $\{S^1_n, \ldots, S^N_n\}$ is a finite partition of $\Omega$ for all $n \in \mathbb{N}$,
2. $\max\{\lambda(S^i_n) \mid i \in \{1,\ldots,N^n\}\} \to 0$, 

17
3. for all $n \in \mathbb{N}$ for all $i \in \{1, \ldots, N^{n-1}\}$ there exists $1 \leq j < k \leq N^n$ such that $\bigcup_{i=1}^{k} S^n_i = S^n_{j-1}$, and

4. the cells $S^n_j$ shrink regularly (there exists $C > 0$ such that for each $S^n_j$ there exists a ball $B^n_j$ with $S^n_j \subset B^n_j$ and $\lambda(S^n_j) \geq C\lambda(B^n_j)$).

We prove the convergence of $L^1$-norm of the positive and negative parts of the functions $d_{\Delta}$ as is required in the proof of Lemma 4.2.

**Lemma A.1.** Let $d_{\Delta} \in L^1(\Omega)$ be an a.e. $[-1, 1]$-valued function. Let $(d_{\Delta}^n)_n \subset L^1(\Omega)$ be a sequence of a.e. $[-1, 1]$-valued simple functions that are defined on a sequence of ordered partitions of $\Omega$ that are an order-conserving domain dissection. Let $d_{\Delta}^n \to d_{\Delta}$ and $(\alpha^n)_n \subset L^1(\Omega, \mathbb{R}^N)$ be defined as in (2) in the proof of Lemma 4.2. Let $\omega^n$ be computed from $\alpha^n$ with Algorithm 2 on the same ordered partitions of $\Omega$. Let $\tilde{d}_{\Delta}^n$ be defined as in (3). Then it follows that

$$\|\max\{d_{\Delta}^n, 0\}\|_{L^1} \to \|\max\{d_{\Delta}, 0\}\|_{L^1},$$

and

$$\|\min\{d_{\Delta}^n, 0\}\|_{L^1} \to \|\min\{d_{\Delta}, 0\}\|_{L^1}.$$  

**Proof.** We observe that $\|f\|_{L^1} = \|\max\{f, 0\}\|_{L^1} + \|\min\{f, 0\}\|_{L^1}$ holds for all $f \in L^1(\Omega)$, which implies that the second claim follows directly from the first. To prove the first claim, we follow the ideas in the proof of [17, Proposition 2.1].

Because $d_{\Delta}^n \to d_{\Delta}$ in $L^1(\Omega)$, it is sufficient to prove that

$$\int_{\Omega} \max\{0, \tilde{d}_{\Delta}^n(s)\} \, ds - \int_{\Omega} \max\{0, d_{\Delta}(s)\} \, ds \to 0,$$

which is equivalent to

$$\int_{\Omega} \max\{0, \omega^n - \omega^n_{\Delta}\} \, ds - \int_{\Omega} \max\{0, \omega^n\} \, ds \to 0$$

by virtue of the constructions in (2) and (3). Because $\omega^n$ is computed by Algorithm 2, it follows that $\omega^n \geq 0$ for all $i \in \{1, 2, 3\}$ and $\sum_{i=1}^{3} \omega^n_i(s) = 1$ a.e. Moreover, $\alpha^n_i \geq 0$ a.e. by construction for all $i \in \{1, 2, 3\}$, implying that we need to show

$$\int_{\Omega} \omega^n_i \, ds - \int_{\Omega} \alpha^n_i \, ds \to 0.$$

This follows from the convergence properties of Algorithm 2, in particular [19, Corollary 4.2 & Lemma 4.4].

### A.2 Relationship between set-based and characteristic function points of view

**Optimality condition for (R) leading to C**

Let $J : L^1(\Omega) \to \mathbb{R}$ be Fréchet differentiable such that the derivative can be represented with the function $\nabla J : L^1(\Omega) \to L^2(\Omega)$. For a characteristic function $\chi_A$, the criticality measure $C_1$ is the $L^1$-norm of the projection of $g(\chi_A) = \nabla J(\chi_A)(\chi_A^c - \chi_A)$ to the positive half plane. The following proposition shows that $C_1(\chi_A) = 0$ if and only if a necessary optimality condition of (R) is satisfied for $\chi_A$. This corresponds to Corollary 1 and Lemma 5 in the set-based point of view analyzed in [5].

**Proposition A.1.** If $\chi_A$ is a local minimizer for (R), then $g(\chi_A) \geq 0$ a.e.

**Proof.** Let $\varepsilon > 0$ and $B \in \mathfrak{B}$ with $\lambda(B) > 0$ be such that $g(\chi_A)(s) < -\varepsilon$ for a.a. $s \in B$. Then $\int_B g(\chi_A) < -\lambda(B)\varepsilon$. Let $d := \chi_{B \setminus A} - \chi_A \cap B$. Then $(\nabla J(\chi_A), d)_{L^2} = \int_B g(\chi_A)$, and we consider the Taylor expansion

$$J(\chi_A + \alpha d) = J(\chi_A) + \alpha(\nabla J(\chi_A), d)_{L^2} + o(\alpha\|d\|_{L^1})$$

$$< J(\chi_A) - \alpha\varepsilon\lambda(B) + o(\alpha)$$

for $\alpha > 0$. Thus $J(\chi_A + \alpha d) < J(\chi_A)$ for all $\alpha > 0$ that are chosen small enough, and $\chi_A$ is not a local minimizer. The claim follows by contraposition. 

\[18\]
Taylor expansion for sets and characteristic functions
Let $J, J', J_s, J'_s$ be given as in §3.1. We say that $J_s$ is Fréchet differentiable if
\[
J_s(A \triangle D) = J_s(A) + J'_s(A)D + o(\lambda(D)),
\]
which is the assertion of [5, Theorem 1].

**Proposition A.2.** $J_s$ is Fréchet differentiable.

**Proof.** Let $A, D \in B$. We use the defined identity $J_s(A \triangle D) = J(\chi_{A \triangle D})$, Taylor’s theorem for $J$, and the identities $A \triangle D = (A \setminus D) \cup (D \setminus A)$, and $A = (A \setminus D) \cup (A \cap D)$—where both unions are disjoint—to deduce
\[
J_s(A \triangle D) = J(\chi_A) + \langle J'(\chi_A), \chi_{A \triangle D} - \chi_A \rangle_{L^\infty,L^1} + o(\|\chi_{A \triangle D} - \chi_A\|_{L^1})
\]

\[
= J(\chi_A) + \langle J'(\chi_A), \chi_{D \setminus A} - \chi_{A \cap D} \rangle_{L^\infty,L^1} + o(\|\chi_{D \setminus A} - \chi_{A \cap D}\|_{L^1}).
\]
We observe that $\|\chi_{D \setminus A} - \chi_{A \cap D}\|_{L^1} = \|\chi_D\|_{L^1}$, and $\|\chi_D\|_{L^1} = \lambda(D)$. This implies
\[
J_s(A \triangle D) = J(\chi_A) + \langle J'(\chi_A), \chi_{D \setminus A} - \chi_{A \cap D} \rangle + o(\lambda(D)).
\]
Inserting the definitions of $J_s(A)$ and $J'_s(A)D$ yields the claim. □

A.3 Existence of Minimizer for (TR($\Delta$))

**Proposition A.3.** Let $\Omega$ be a bounded domain. Let $U \in B$. Let $\Delta \in [0, \infty)$. Let $\nabla J \in L^1(\Omega)$. Then (TR($\Delta$)) admits a minimizer $\bar{d}$ with $(\nabla J(\chi_U), \bar{d})_{L^2} \leq -\frac{1}{\lambda(\Omega)}C_1(\chi_U)$.

**Proof.** Let $f_U := \nabla J(\chi_U)(\chi_{U^c} - \chi_U)$, and let $D : \mathbb{R} \rightarrow B$ be defined as $D(x) := f_U^{-1}((\infty, x))$ for $x \in \mathbb{R}$. Then $D(x) \subset D(y)$ for all $x \leq y$, implying that $\|\chi_{U \Delta D(x)} - \chi_U\|_{L^1} = \lambda(D(x))$ is monotone in $x$. Let $d(x) := \chi_{U \Delta D(x)} - \chi_U$. Then the $d(x)$ are greedy solution candidates for (TR($\Delta$)) with $\lim_{x \rightarrow -\infty} \|d(x)\|_{L^1} = 0$. Specifically, $d(0)$ minimizes (TR($\Delta$)) if $\Delta = \infty$.

Let $\Delta < \infty$. If $\|d(0)\|_{L^1} \leq \Delta$, then $d(0)$ minimizes (TR($\Delta$)). We restrict to $\|d(0)\|_{L^1} > \Delta$. Because of the greedy construction, $\bar{d}(\bar{x})$ is optimal if $\|d(\bar{x})\|_{L^1} = \Delta$ for some $\bar{x} < 0$. We consider the case where there is no such $\bar{x} < 0$. We consider $\bar{x} : = \sup\{x \mid \|d(x)\|_{L^1} \leq \Delta\}$. If $d$ is continuous at $\bar{x}$, then $\|d(\bar{x})\|_{L^1} = \Delta$, and $d(\bar{x})$ minimizes (TR($\Delta$)). We distinguish two situations.

**Situation 1.** If $d$ is only left continuous at $\bar{x}$, then we have $\|d(\bar{x})\|_{L^1} \leq \Delta < \lim_{y \uparrow \bar{x}} \|d(y)\|_{L^1}$. Thus there exists a set $A \in B$ satisfying $A \cap D(\bar{x}) = \emptyset$, $A \subset D(y)$ for all $y > \bar{x}$, and $\lambda(A) = \lim_{y \uparrow \bar{x}} \|d(y)\|_{L^1} - \|d(\bar{x})\|_{L^1}$. Such a set also exists if $\|d(\bar{x})\|_{L^1} \leq \Delta$ and $d$ is neither left nor right continuous at $\bar{x}$.

**Situation 2.** If $d$ is only right continuous at $\bar{x}$, then we have $\lim_{y \downarrow \bar{x}} \|d(y)\|_{L^1} \leq \Delta < \|d(\bar{x})\|_{L^1}$. Thus there exists a set $A \in B$ satisfying $A \subset D(\bar{x})$, $A \cap D(y)$ for all $y < \bar{x}$, and $\lambda(A) = \|d(\bar{x})\|_{L^1} - \lim_{y \downarrow \bar{x}} \|d(y)\|_{L^1}$. Such a set also exists if $\Delta < \|d(\bar{x})\|_{L^1}$ and $d$ is neither left nor right continuous at $\bar{x}$.

Because of the monotony of $\|d(\cdot)\|_{L^1}$ and the fact that the limits $\lim_{y \uparrow \bar{x}} \|d(y)\|_{L^1}$ and $\lim_{y \downarrow \bar{x}} \|d(y)\|_{L^1}$ always exist by virtue of continuity from below and above of the Lebesgue measure, this distinction is exhaustive. Because of the mean value property of the Lebesgue measure [3, Cor. 1.12.10], there exists $B \supset A$ with $\lambda(B) = \Delta - \|d(\bar{x})\|_{L^1}$ (Situation 1) or $\lambda(B) = \|d(\bar{x})\|_{L^1} - \Delta$ (Situation 2). In Situation 1 we set $\hat{D} := D(\bar{x}) \cup B$, and in Situation 2 we set $\hat{D} := D(\bar{x}) \setminus B$. In both situations, $\hat{d} : = \chi_{U \Delta \hat{D}} - \chi_U$ minimizes (TR($\Delta$)).

The greedy construction of $D(\bar{x})$ and thus $\hat{D}$ with respect to $f_U$ imply
\[
\frac{1}{\Delta} \int_{\hat{D}} f_U \, ds \leq \frac{1}{\|\lambda(f_U^{-1}((\infty, 0)))\|} \int_{f_U^{-1}((\infty, 0))} f_U \, ds
\]

\[
= \frac{1}{\|\lambda(f_U^{-1}((\infty, 0)))\|} C_1(\chi_U) \leq \frac{1}{\lambda(\Omega)} C_1(\chi_U),
\]
where we have used the definition of $C_1$ for the equality and that the integrand is negative for the second inequality. Then the identity $(\nabla J(\chi_U), \hat{d})_{L^2} = \int_{\hat{D}} f_U \, ds$ yields the claimed inequality. □
B Implementation Information on §5.2

We automate the solution process for the boundary value problem as well as the assembly of the objective function and its gradient with respect to the degrees of freedom (DOFs) of the PDE solution using the FEniCS framework.\footnote{Version 2019.1.0; obtained from \url{https://fenicsproject.org} under GNU LGPLv3} The gradient with respect to the control function is then determined by solving the adjoint problem, which is separately constructed by using FEniCS’ \texttt{adjoint} function. Since the DOFs of cellwise constant functions in FEniCS are their pointwise values for each cell, we can approximate the set gradient for a given cell by dividing the gradient with respect to its DOF by the cell’s volume and adjusting the sign where appropriate. This is equivalent to the approximation method described in \cite[Sec. 3.3.2]{5}.

Relaxed Optimization Algorithm (NLP) The relaxed problem is produced by replacing \{0, 1\} with its convex hull \([0, 1]\) in (8). The NLP resulting from discretization using FEniCS is then solved using SciPy’s\footnote{Version 1.3.3; obtainable from \url{https://scipy.org} under a BSD-3-clause license} implementation of L-BFGS-B.

To obtain comparable instationarity tolerances and thus termination criteria between (NLP) and (BTR), we scale variables and gradient tolerances with the associated cell volumes. In order to avoid very small numeric values, each variable is scaled by the ratio between its associated cell volume and the largest cell volume in the mesh. The tolerance is then scaled by the largest cell volume in the mesh to make the effective tolerance threshold roughly comparable with that of the binary trust-region algorithm. We note that even after scaling, the termination criteria are not equivalent, because the NLP solver uses the $\ell_2$ norm whereas the trust-region algorithm uses the $\ell_1$ norm. However, we believe that this yields a fairer comparison of the optimization results of both algorithms.

SciPy’s implementation of L-BFGS-B takes an additional parameter $\texttt{ftol}$ that is used to terminate the optimization loop when the improvement in objective function value becomes too small. In order to effectively deactivate this criterion, $\texttt{ftol}$ is set to $10^{-16}$ and scaled along with the gradient tolerance parameter.

Sum-Up Rounding (SUR) Sum-up rounding requires the mesh cells to be ordered; see the discussion in §A.1. We use an approximation of the Sierpinski curve to enumerate the cells of the initial mesh. However, this is no longer possible for the unevenly refined meshes produced by the binary trust-region algorithm. We therefore require a method by which to refine a cell order. We do this by a recursive procedure using \texttt{k-means++} clustering \cite{1} and Christofides’ algorithm \cite{4} for the traveling salesman problem (TSP).

First, we group cells based on the parent cell. We detect the parent cell based on the location of the child cell’s midpoint. The resulting clusters are then arranged according to the parent cell order. The midpoint of each cluster is defined to be the average of the midpoints of all cells contained therein.

The cell order of the child mesh is determined by concatenating the cell orders of these clusters in their predetermined order. When encountering a cluster with $N = 50$ or more children we replace the cluster with a cluster of $k = \lceil N/25 \rceil$ subclusters generated by applying the \texttt{k-means++} algorithm to the midpoints of its immediate children. For clusters with less than 50 children, an approximate TSP solution is used to order the cluster’s elements.

The approximate TSP solution is determined on a metric graph that is generated by taking a complete graph of the midpoints of the cluster’s children with edges weighted by their respective Euclidean distances from one another. We add three dummy nodes to this graph: origin, entry, and exit. The entry node is associated with the midpoint of the last cell preceding the current cluster. The exit node is associated with the midpoint of the cluster or cell immediately following the current cluster. The entry and exit nodes are connected to each of the current cluster’s elements by an edge whose weight corresponds to the Euclidean distance between the entry or exit point and that element’s midpoint.

If either the entry or the exit node does not correspond to the midpoint of a cluster or mesh cell, then it is associated with a fictional point whose distance from all cluster element midpoints is equal to the maximum over all intrachannel distances. The entry and exit node are connected through the origin node whose distance from both nodes is equal to the maximal distance between any two connected points in the graph thus far.

An approximate minimal cost Hamilton cycle is constructed by using Christofides’ algorithm as implemented by the Python package NetworkX.\footnote{Version 2.6.3, obtainable from \url{https://github.com/networkx/networkx} under a 3-clause BSD license} Because the resulting cycle is constructed on a shortest-distance
Figure 3: Step-by-step illustration of cell order determination upon refinement. All parent cell clusters are divided exactly once with \( k = 4 \) to illustrate the treatment of subclusters. In practice, recursive subclustering may be necessary.

The order refinement method may appear complicated. We illustrate the steps of the algorithm using a simplified example in Figure 3. We note that in this illustration we assume that each of the parent cell clusters is divided exactly once into \( k = 4 \) subclusters and that no recursive clustering is required.

**Binary Simplified Trust-Region (BSTR)** The simplified trust-region algorithm is largely equal to the binary trust-region algorithm, except that it does not follow the same \texttt{FindStep} procedure. Rather than finding a cutoff level by bisection, we sort the mesh cells with negative set gradient by ascending gradient value, calculate the cumulative cell volume up to each index, and find a cutoff point by searching for the largest index in the list of cumulative volumes. The algorithm does not terminate based on the original optimality criterion. Rather, it terminates if and only if the step-finding procedure returns an empty step.

We note that this means that BSTR can run beyond the point where the optimality threshold is satisfied.
We use BSTR as a benchmark to determine if the result of BTR could easily be improved without refinement.

**Binary Trust-Region (BTR)** Our implementation of BTR follows exactly the procedures laid out in [5]. The FindStep procedure is implemented by using the bisection method described therein. After bisection has reached the desired accuracy in its cutoff level, the step is filled up to the trust-region radius. The method by which this is done is deliberately left open in [5].

To make step determination more comparable between BTR and BSTR, the step is filled with mesh cells in order of ascending set gradient value. If the result is not sufficiently close to the trust-region radius, the remaining cells between the upper and lower cutoff level are sorted by descending volume, and the smallest number of candidate cells accounting for at least half of the candidate volume are refined by using FEniCS’ refine method, which internally uses the refinement algorithm described in [20]. Upon refinement, the set gradient is not re-calculated but rather interpolated to the new mesh. In this way, refinement is iteratively performed until the step reaches the desired volume range.

We recalculate the objective function on the new mesh for the purposes of step quality calculation. If the step is not accepted, we discard the refinement and continue using the old mesh for the next iteration. Otherwise, we use the new mesh for subsequent iterations.

**Initial Meshes and Their Cell Order** We generate starting meshes using FEniCS’ UnitSquareMesh constructor with crossed diagonals, the result of which is illustrated in fig. 4. We always select an equal power of two as the number of subdivisions on each axis because this enables the use of cell orders based on the Sierpinski curve for sum-up rounding. We denote these meshes by the number of subdivisions. A “256 × 256” mesh is therefore a uniform triangle mesh subdividing the unit square with 256 subdivisions along each axis and crossed diagonals.

To determine the starting cell order, we use the Sierpinski curve. The Sierpinski curve is a space-filling curve that fills the unit square. Figure 5 illustrates how the Sierpinski curve is generated and how the vertices of its nth approximation fall precisely into the cells of a 2n × 2n mesh with crossed diagonals.

**References**

[1] D. Arthur and S. Vassilvitskii. K-Means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’07, pages 1027—1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

[2] L. Boccardo and T. Gallouët. Non-linear elliptic and parabolic equations involving measure data. *Journal of Functional Analysis*, 87(1):149–169, 1989.

[3] V. I. Bogachev. *Measure Theory*, volume 1. Springer, 2007.

[4] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon University, Pittsburgh, PA, Management Sciences Research Group, 1976.
Figure 5: Illustration of cell ordering via the Sierpinski curve.

[5] M. Hahn, S. Sager, and S. Leyffer. Binary optimal control by trust-region steepest descent. Optimization Online Peprint 7589, 2020.

[6] R. Haller-Dintelmann, C. Meyer, J. Rehberg, and A. Schiela. Hölder continuity and optimal control for nonsmooth elliptic problems. Applied Mathematics and Optimization, 60(3):397–428, 2009.

[7] F. M. Hante. Mixed-integer optimal control for PDEs: Relaxation via differential inclusions and applications to gas network optimization. In Mathematical Modelling, Optimization, Analytic and Numerical Solutions, pages 157–171. Springer, 2020.

[8] F. M. Hante and S. Sager. Relaxation methods for mixed-integer optimal control of partial differential equations. Computational Optimization and Applications, 55(1):197–225, 2013.

[9] F. M. Hante, G. Leugering, A. Martin, L. Schewe, and M. Schmidt. Challenges in optimal control problems for gas and fluid flow in networks of pipes and canals: From modeling to industrial applications. In Industrial mathematics and complex systems, pages 77–122. Springer, 2017.

[10] J. Haslinger and R. A. Mäkinen. On a topology optimization problem governed by two-dimensional Helmholtz equation. Computational Optimization and Applications, 62(2):517–544, 2015.

[11] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. Optimization with PDE constraints, volume 23. Springer Science & Business Media, 2008.

[12] C. Kirches, P. Manns, and S. Ulbrich. Compactness and convergence rates in the combinatorial integral approximation decomposition. Mathematical Programming, pages 1–30, 2020.

[13] S. Leyffer and P. Manns. Sequential linear integer programming for integer optimal control with total variation regularization. arXiv preprint arXiv:2106.13453, 2021.

[14] S. Leyffer, P. Manns, and M. Winckler. Convergence of sum-up rounding schemes for cloaking problems governed by the Helmholtz equation. Computational Optimization and Applications, 79(1):193–221, 2021.

[15] J. Lindenstrauss. A short proof of Liapounoff’s convexity theorem. Journal of Mathematics and Mechanics, 15(6):971–972, 1966.

[16] A. A. Lyapunov. On completely additive vector functions. Izv. Akad. Nauk SSSR, 4:465–478, 1940.

[17] P. Manns. Relaxed multibang regularization for the combinatorial integral approximation. SIAM Journal on Control and Optimization, 59(4):2645–2668, 2021.

[18] P. Manns and C. Kirches. Improved regularity assumptions for partial outer convexification of mixed-integer PDE-constrained optimization problems. ESAIM: Control, Optimisation and Calculus of Variations, 26:32, 2020.
[19] P. Manns and C. Kirches. Multidimensional sum-up rounding for elliptic control systems. *SIAM Journal on Numerical Analysis*, 58(6):3427–3447, 2020.

[20] A. Plaza and G. Carey. Local refinement of simplicial grids based on the skeleton. *Applied Numerical Mathematics*, 32(2):195–218, 2000. ISSN 0168-9274.

[21] S. Sager. *Numerical methods for mixed-integer optimal control problems*. Citeseer, 2005.

[22] S. Sager, M. Jung, and C. Kirches. Combinatorial integral approximation. *Mathematical Methods of Operations Research*, 73(3):363–380, 2011.

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

[24] M. Sharma, M. Hahn, S. Leyffer, L. Ruthotto, and B. van Bloemen Waanders. Inversion of convection–diffusion equation with discrete sources. *Optimization and Engineering*, pages 1–39, 2020.

[25] L. Tartar. Compensated compactness and applications to partial differential equations. In *Nonlinear analysis and mechanics: Heriot–Watt symposium*, volume 4, pages 136–212, 1979.

[26] P. L. Toint. Non-monotone trust-region algorithms for nonlinear optimization subject to convex constraints. *Mathematical Programming*, 77(3):69–94, 1997.

[27] R. H. Vogt, S. Leyffer, and T. Munson. A mixed-integer PDE-constrained optimization formulation for electromagnetic cloaking. *Preprint ANL/MCS-P9277-0120, Mathematics and Computer Science Division, Argonne National Laboratory*, pages 1–21, 2020.

[28] J. Yu and M. Anitescu. Multidimensional sum-up rounding for integer programming in optimal experimental design. *Mathematical Programming*, 185(1):37–76, 2021.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan [http://energy.gov/downloads/doe-public-access-plan](http://energy.gov/downloads/doe-public-access-plan).