Mixed-Integer PDE-Constrained Optimization
IMA Workshop on Uncertainty Quantification

Pelin Cay, Bart van Bloemen Waanders, Drew Kouri, Anna Thuenen and Sven Leyffer

Lehigh University, Universität Magdeburg, Argonne National Laboratory, and Sandia National Laboratories

February 22, 2016
Outline

1. Introduction
   - Problem Definition and Applications
   - Source Inversion as MIP with PDE Constraints
   - Problem Classification and Challenges

2. Early Theoretical & Numerical Results
   - Eliminating the PDE & State Variables
   - Numerical Experience with Source Inversion
   - Control Regularization: Not All Norms Are Equal
   - Heat Equation: Actuator Design

3. Rounding-Based Heuristic for Cloaking

4. Conclusions
PDE-constrained MIP ... $u = u(t, x, y, z) \Rightarrow$ infinite-dimensional!

- $t$ is time index; $x, y, z$ are spatial dimensions

$$\begin{align*}
\text{minimize} & \quad F(u, w) \\
\text{subject to} & \quad C(u, w) = 0 \\
& \quad u \in U, \quad \text{and} \quad w \in \mathbb{Z}^p \quad \text{(integers)},
\end{align*}$$

- $u(t, x, y, z)$: PDE states, controls, & design parameters
- $w$ discrete or integral variables

**MIPDECO Warning**

$w = w(t, x, y, z) \in \mathbb{Z}$ may be infinite-dimensional integers!
Mixed-Integer PDE-Constrained Optimization (MIPDECO)

PDE-constrained MIP ... $u = u(t, x, y, z) \Rightarrow$ infinite-dimensional!
- $t$ is time index; $x, y, z$ are spatial dimensions

$$
\begin{aligned}
\begin{cases}
\text{minimize} & \mathcal{F}(u, w) \\
\text{subject to} & \mathcal{C}(u, w) = 0 \\
& u \in \mathcal{U}, \quad \text{and} \quad w \in \mathbb{Z}^p \quad (\text{integers}),
\end{cases}
\end{aligned}
$$

- $u(t, x, y, z)$: PDE states, controls, & design parameters
- $w$ discrete or integral variables

MIPDECO Warning

$w = w(t, x, y, z) \in \mathbb{Z}$ may be infinite-dimensional integers!

It’s a MIP, Jim, but not as we know it!
Grand-Challenge Applications of MIPDECO

- **Topology optimization** [Sigmund and Maute, 2013]
- Nuclear plant design: select core types & control flow rates [Committee, 2010]
- Well-selection for remediation of contaminated sites [Ozdogan, 2004]
- Design of next-generation solar cells [Reinke et al., 2011]
- Design of wind-farms [Zhang et al., 2013]
- Design & control of gas networks, [De Wolf and Smeers, 2000, Martin et al., 2006, Zavala, 2014]

... also as optimization under uncertainty
Uncertainty Quantification and MIPDECO

- Design of experiments, e.g. discrete sensor placement
- Akaike’s Information Criterion: parameter & structure est.
  - AIC: maximize log-likelihood & minimize nonzeros $u$

$$\text{minimize } \sum_{k=1}^{N} e_k(u)^T R^{-1} e_k(u) + \sum_{i=1}^{l} w_i \quad \text{s.t. } -Mw_i \leq u_i \leq Mw_i$$

where $R$ is known co-variance
Source Inversion as MIP with PDE Constraints

Simple Example: Locate number of sources to match observation $\bar{u}$

\[
\begin{aligned}
\text{minimize} & \quad J = \frac{1}{2} \int_{\Omega} (u - \bar{u})^2 d\Omega \\
\text{subject to} & \quad -\Delta u = \sum_{k,l} w_{kl} f_{kl} \quad \text{in } \Omega \\
& \quad \sum_{k,l} w_{kl} \leq S \quad \text{and } w_{kl} \in \{0, 1\} \\
\end{aligned}
\]

with Dirichlet boundary conditions $u = 0$ on $\partial \Omega$.

E.g. Gaussian source term, $\sigma > 0$, centered at $(x_k, y_l)$

\[
f_{kl}(x, y) := \exp \left( \frac{-\| (x_k, y_l) - (x, y) \|^2}{\sigma^2} \right),
\]

Motivated by porous-media flow application to determine number of boreholes, [Ozdogan, 2004, Fipki and Celi, 2008]
Source Inversion as MIP with PDE Constraints

Consider 2D example with $\Omega = [0, 1]^2$ and discretize PDE:

- 5-point finite-difference stencil; uniform mesh $h = 1/N$
- Denote $u_{i,j} \approx u(ih, jh)$ approximation at grid points

$$
\begin{align*}
\text{minimize } \quad & J_h = \frac{h^2}{2} \sum_{i,j=0}^{N} (u_{i,j} - \bar{u}_{i,j})^2 \\
\text{subject to } \quad & \frac{4u_{i,j} - u_{i,j-1} - u_{i,j+1} - u_{i-1,j} - u_{i+1,j}}{h^2} = \sum_{k,l=1}^{N} w_{kl} f_{kl}(ih, jh) \\
& u_{0,j} = u_{N,j} = u_{i,0} = u_{i,N} = 0 \\
& \sum_{k,l=1}^{N} w_{kl} \leq S \text{ and } w_{kl} \in \{0, 1\}
\end{align*}
$$

$\Rightarrow$ finite-dimensional (convex) MIQP
Source Inversion as MIP with PDE Constraints

Potential source locations (blue dots) on $16 \times 16$ mesh
Create target $\bar{u}$ using red square sources
Source Inversion as MIP with PDE Constraints

Target (3 sources), reconstructed sources, & error on $32 \times 32$ mesh
Outline

1 Introduction
   - Problem Definition and Applications
   - Source Inversion as MIP with PDE Constraints
   - Problem Classification and Challenges

2 Early Theoretical & Numerical Results
   - Eliminating the PDE & State Variables
   - Numerical Experience with Source Inversion
   - Control Regularization: Not All Norms Are Equal
   - Heat Equation: Actuator Design

3 Rounding-Based Heuristic for Cloaking

4 Conclusions
Mixed-Integer PDE-Constrained Optimization (MIPDECO)

\[
\begin{align*}
\text{minimize} & \quad \mathcal{F}(u, w) \\
\text{subject to} & \quad C(u, w) = 0 \\
& \quad u \in \mathcal{U}, \text{ and } w \in \mathbb{Z}^p \text{ (integers)},
\end{align*}
\]

- \(u(t, x, y, z)\): PDE states, controls, & design parameters
- \(w\) discrete or integral variables

Towards a problem characterization

- **Type of PDE:** different classes of PDEs
  - e.g. elliptic, parabolic, hyperbolic, nonlinear, ...
- **Class of Integers:** binary, general integers, etc
- **Type of Objective:** functional form of objective
- **Type of Constraints:** characterize c/s other than PDE
- **Discretization:** discretization method & CUTEr classification
Mesh-Independent & Mesh-Dependent Integers

**Definition (Mesh-Independent & Mesh-Dependent Integers)**

1. The integer variables are mesh-independent, iff number of integer variables is independent of the mesh.
2. The integer variables are mesh-dependent, iff the number of integer variables depends on the mesh.

Mesh-Independent

- Manageable tree
- Theory possible

Mesh-Dependent

- Exploding tree size
- Theory???
Theoretical Challenges of MIPDECO

Functional Analysis (mesh-dependent integers)

Denis Ridzal: What function space is \( w(x, y) \in \{0, 1\} \)?

- Consistently approximate \( w(x, y) \in \{0, 1\} \) as \( h \to 0 \)?
- Conjecture: \( \{w(x, y) \in \{0, 1\}\} \neq L_2(\Omega) \)
  ... e.g. binary support of Cantor set not integrable
- Likely need additional regularity assumptions

Coupling between Discretization & Integers

Discretization scheme (e.g. upwinding for wave equation) depends on direction of flow (integers).

- Application: gas network models with flow reversals

... open postdoc position at Argonne!
Computational Challenges of MIPDECO

- Approaches for **humongous branch-and-bound trees**
  ... e.g. 3D topology optimization with $10^9$ binary variables

- **Warm-starts** for PDE-constrained optimization (nodes)
- Guarantees for **nonconvex (nonlinear) PDE constraints**
  ... factorable programming approach hopeless for $10^9$ vars!

\[
\begin{align*}
\log^3 \log^2 x \times x^* + \\
\cdots \quad f(x_1, x_2) &= x_1 \log(x_2) + x_2^3
\end{align*}
\]
# MIPDECO: Two Cultures Collide

**Observation**

PDE-optimization & MIP developed separately
⇒ different assumptions, methodologies, and computational kernels!

## PDE-Optimization vs. Mixed-Integer Programming

<table>
<thead>
<tr>
<th>PDE-Optimization</th>
<th>Mixed-Integer Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtain good solutions efficiently</td>
<td>Deliver certificate of optimality</td>
</tr>
<tr>
<td>Nonlinear optimization:</td>
<td>Combinatorial optimization:</td>
</tr>
<tr>
<td>Newton’s method</td>
<td>branch-and-cut</td>
</tr>
<tr>
<td>Iterative Krylov solvers</td>
<td>Factors &amp; rank-one updates</td>
</tr>
<tr>
<td>Run on bleeding-edge HPC</td>
<td>Limited HPC developments</td>
</tr>
</tbody>
</table>

**Potential for Disaster, or Opportunity for Innovation!**
Outline

1 Introduction
   - Problem Definition and Applications
   - Source Inversion as MIP with PDE Constraints
   - Problem Classification and Challenges

2 Early Theoretical & Numerical Results
   - Eliminating the PDE & State Variables
   - Numerical Experience with Source Inversion
   - Control Regularization: Not All Norms Are Equal
   - Heat Equation: Actuator Design

3 Rounding-Based Heuristic for Cloaking

4 Conclusions
Find number and location of sources to match observation $\bar{u}$

$$\begin{align*}
\text{minimize} \quad & \mathcal{J} = \frac{1}{2} \int_{\Omega} (u(w) - \bar{u})^2 \, d\Omega \\
\text{subject to} \quad & -\Delta u = \sum_{k,l} w_{kl} f_{kl} \quad \text{in} \; \Omega \quad \text{Poisson equation} \\
& \sum_{k,l} w_{kl} \leq S \quad \text{and} \; w_{kl} \in \{0, 1\} \quad \text{source budget}
\end{align*}$$

- MIP with convex quadratic objective on $\Omega = [0, 1]^2$
- 5-point finite-difference stencil; uniform mesh $h = 1/N$
- Denote $u_{i,j} \approx u(ih, jh)$ approximation at grid points
Cool MIPDECO Trick: Eliminating the PDE

Discretized PDE constraint (Poisson equation)

\[
\frac{4u_{i,j} - u_{i,j-1} - u_{i,j+1} - u_{i-1,j} - u_{i+1,j}}{h^2} = \sum_{k,l} w_{kl} f_{kl}(ih,jh), \forall i,j
\]

\[\Leftrightarrow A\mathbf{u} = \sum w_{kl} f_{kl}, \text{ where } w_{kl} \in \{0, 1\} \text{ only appear on RHS!}\]

Elimination of PDE and states \(u(x, y, z)\)

\[
A\mathbf{u} = \sum_{k,l} w_{kl} f_{kl} \Leftrightarrow \mathbf{u} = A^{-1} \left( \sum_{k,l} w_{kl} f_{kl} \right) = \sum_{k,l} w_{kl} A^{-1} f_{kl}
\]

- Solve \(n^2 \ll 2^n\) PDEs: \(u^{(kl)} := A^{-1} f_{kl}\)
- Eliminate \(\mathbf{u} = \sum_{k,l} w_{kl} u^{(kl)}\) from Source Inversion
Cool MIPDECO Trick: Eliminating the PDE

Eliminating $u = \sum_{k,l} w_{kl} u^{(kl)}$ in MINLP gives:

$$
\begin{align*}
\begin{cases}
\text{minimize} & J_h = \frac{h^2}{2} \sum_{i,j=0}^{N} \left( \sum_{k,l} w_{kl} u^{(kl)}_{ij} - \bar{u}_{i,j} \right)^2 \\
\text{subject to} & \sum_{k,l=1}^{N} w_{kl} \leq S \quad \text{and} \quad w_{kl} \in \{0, 1\}
\end{cases}
\end{align*}
$$

- Eliminates the states $u$ ($N^2$ variables)
- Eliminates the PDE constraint ($N^2$ constraints)

... generalizes to other PDEs (with integer controls on RHS)

Simplified model is quadratic knapsack problem
Outline

1 Introduction
   - Problem Definition and Applications
   - Source Inversion as MIP with PDE Constraints
   - Problem Classification and Challenges

2 Early Theoretical & Numerical Results
   - Eliminating the PDE & State Variables
   - Numerical Experience with Source Inversion
   - Control Regularization: Not All Norms Are Equal
   - Heat Equation: Actuator Design

3 Rounding-Based Heuristic for Cloaking

4 Conclusions
Numerical Experience with Source Inversion

Find number and location of sources to match observation $\bar{u}$

$$
\begin{aligned}
& \text{minimize} & & \mathcal{J} = \frac{1}{2} \int_{\Omega} (u(w) - \bar{u})^2 \, d\Omega \\
& \text{subject to} & & -\Delta u = \sum_{k,l} w_{kl} f_{kl} \quad \text{in } \Omega \\
& & & \sum_{k,l} w_{kl} \leq S \text{ and } w_{kl} \in \{0, 1\}
\end{aligned}
$$

MIP with convex quadratic objective

Computational Experiments:

1. Test NLP-plus-rounding heuristic versus MINLP
2. Effect of mesh-dependent vs. mesh-independent integers
   - Mesh-independent: pick sources from 36 potential locations
   - Mesh-dependent: all nodes are potential locations
3. Effect of state-elimination trick
1st Example Mixed-Integer PDE-Constrained Optimization

Potential source locations (blue dots) on 16 × 16 mesh
Create target \( \bar{u} \) using red square sources
Approach 1: NLP-Solve, Knapsack Rounding, and MIP

**Knapsack Rounding**

1. Solve continuous relaxation using NLP solver
2. Round largest $S$ locations, $w_i$, to one & set all others to zero
Approach 1: NLP-Solve, Knapsack Rounding, and MIP

Knapsack Rounding

1. Solve continuous relaxation using NLP solver
2. Round largest $S$ locations, $w_i$, to one & set all others to zero

Knapsack-rounded NLP (left) and MINLP (right)

MINLP solution better: $\text{NLP-err} = 0.0388 > 0.0307 = \text{MIP-err}$
Mesh-Independent Source Inversion: MINLP Solvers

Number of Nodes and CPU time for Increasing Mesh Sizes

- **Number of Nodes independent of mesh size!**
- **MINLP & Minotaur**: filterSQP runs out of memory for $N \geq 32$
- **BonminOA** takes roughly 100 iterations ... quadratic objective
Mesh-Dependent (all) Source Inversion: MINLP Solvers

Number of Nodes and CPU time for Increasing Mesh Sizes

- Number of nodes explodes with mesh size!
- OA after 130,000 seconds
Elimination of States & PDEs: Source Inversion

CPU Time for Increasing Mesh Sizes: Simplified vs. Original Model

Eliminating PDEs is two orders of magnitude faster!
Elimination of States & PDEs: Source Inversion

CPU Time for Increasing Mesh Sizes: Simplified vs. Original Model

<table>
<thead>
<tr>
<th></th>
<th>$8 \times 8$</th>
<th>$16 \times 16$</th>
<th>$32 \times 32$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presolve Time</td>
<td>0.05</td>
<td>1.30</td>
<td>62.51</td>
</tr>
<tr>
<td>Simplified Model</td>
<td>0.18</td>
<td>0.50</td>
<td>2.38</td>
</tr>
<tr>
<td>Total Simplified</td>
<td>0.23</td>
<td>1.80</td>
<td>64.89</td>
</tr>
<tr>
<td>Full PDE Model</td>
<td>2.10</td>
<td>29.43</td>
<td>1013.21</td>
</tr>
</tbody>
</table>

... using NLP solve for PDE (inefficient)

Presolve is cheap ... simplified model solves much faster!
First Conclusions: Source Inversion

Numerical Results

- Solve mesh-independent problems with coarse discretization
- Mesh-dependent instances cannot be solved
- Outer Approximation (Bon-OA) inefficient for these instances
- Trick #1: elimination of states and PDE constraint
- Nonlinear solvers run into storage issues

...not surprising: MIPDECO trees grow like tribbles!
First Conclusions: Source Inversion

Numerical Results

- Solve mesh-independent problems with coarse discretization
- Mesh-dependent instances cannot be solved
- Outer Approximation (Bon-OA) inefficient for these instances
- Trick # 1: elimination of states and PDE constraint
- Nonlinear solvers run into storage issues

...not surprising: MIPDECO trees grow like tribbles!
Outline

1 Introduction
   - Problem Definition and Applications
   - Source Inversion as MIP with PDE Constraints
   - Problem Classification and Challenges

2 Early Theoretical & Numerical Results
   - Eliminating the PDE & State Variables
   - Numerical Experience with Source Inversion
   - Control Regularization: Not All Norms Are Equal
   - Heat Equation: Actuator Design

3 Rounding-Based Heuristic for Cloaking

4 Conclusions
Control Regularization: Not All Norms Are Equal

Poisson with Distributed Control [OPTPDE, 2014] & [Tröltzsch, 1984]

\[
\begin{align*}
\text{minimize} \quad & \| u - u_d \|_{L^2(\Omega)}^2 + \int_\Gamma e_{\Gamma} \ u \ ds + \alpha \| w \|_{L^x}^2 \\
\text{subject to} \quad & -\Delta u + u = w + e_{\Omega} \text{ in } \Omega \\
& \frac{\partial u}{\partial n} = 0 \text{ on boundary } \Gamma \\
& w(t) \in \{0, 1\}
\end{align*}
\]

L1 or L2 regularization term for control w(t) ∈ {0, 1}? 

Good Norms for MIPs

MIP’ers prefer polyhedral norms … promote integrality

- Old MIP trick: \( w^2(t) = |w(t)| \) for \( w(t) \in \{0, 1\} \)

\( \Rightarrow \) L1-norm same as L2-norm on binary variables!
Not All Norms Are Equal

Consider **Distributed Control** for increasing mesh-size

<table>
<thead>
<tr>
<th>Mesh</th>
<th>CPU for $L^2$ Regularization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Minotaur</td>
</tr>
<tr>
<td>8x8</td>
<td>0.04</td>
</tr>
<tr>
<td>16x16</td>
<td>6.61</td>
</tr>
<tr>
<td>32x32</td>
<td>Time</td>
</tr>
</tbody>
</table>

$L^1$ regularization is equivalent to $L^2$, but faster

Many fewer nodes in tree-searches $\Rightarrow$ solve up to $256 \times 256$
Not All Norms Are Equal

Consider Distributed Control for increasing mesh-size

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Minotaur</th>
<th>B-BB</th>
<th>B-Hyb</th>
<th>B-OA</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x8</td>
<td>0.04</td>
<td>0.80</td>
<td>2.54</td>
<td>126.81</td>
</tr>
<tr>
<td>16x16</td>
<td>6.61</td>
<td>72.21</td>
<td>1305.00</td>
<td>Time</td>
</tr>
<tr>
<td>32x32</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
<td>Time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Minotaur</th>
<th>B-BB</th>
<th>B-Hyb</th>
<th>B-OA</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x8</td>
<td>0.03</td>
<td>0.48</td>
<td>0.21</td>
<td>0.04</td>
</tr>
<tr>
<td>16x16</td>
<td>0.11</td>
<td>3.62</td>
<td>0.66</td>
<td>0.20</td>
</tr>
<tr>
<td>32x32</td>
<td>0.18</td>
<td>62.66</td>
<td>3.53</td>
<td>0.74</td>
</tr>
</tbody>
</table>

$L^1$ regularization is equivalent to $L^2$, but faster

Many fewer nodes in tree-searches $\Rightarrow$ solve up to $256 \times 256$
Outline

1 Introduction
   - Problem Definition and Applications
   - Source Inversion as MIP with PDE Constraints
   - Problem Classification and Challenges

2 Early Theoretical & Numerical Results
   - Eliminating the PDE & State Variables
   - Numerical Experience with Source Inversion
   - Control Regularization: Not All Norms Are Equal
   - Heat Equation: Actuator Design

3 Rounding-Based Heuristic for Cloaking

4 Conclusions
Problem 2: Actuator Placement and Operation [Falk Hante]

Goal: Control temperature with actuators
- Select sequence of control inputs (actuators)
- Choose continuous control (heat/cool) at locations
- Match prescribed temperature profile

... “de-mist bathroom mirror with hair-drier”

Potential Actuator Locations \( l = 1, \ldots, L \)
Problem 2: Actuator Placement and Operation

Find optimal sequence of actuators, \( w_l(t) \), and controls, \( v_l(t) \):

\[
\begin{align*}
\text{minimize} & \quad \| u(t_f, \cdot) \|_{\Omega}^2 + 2\| u \|_{T \times \Omega}^2 + \frac{1}{500} \| v \|_T^2 \\
\text{subject to} & \quad \frac{\partial u}{\partial t} - \kappa \Delta u = \sum_{l=1}^{L} v_l(t) f_l \quad \text{in} \quad T \times \Omega \\
& \quad w_l(t) \in \{0, 1\}, \quad \sum_{l=1}^{L} w_l(t) \leq W, \quad \forall t \in T \\
& \quad Lw_l(t) \leq v_l(t) \leq Uw_l(t), \quad \forall l = 1, \ldots, L, \quad \forall t \in T
\end{align*}
\]

where

\[
f_l(x, y) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-\| (x, y) - (x_l, y_l) \|^2}{2\sigma}\right)
\]

point-source for actuators at \((x_l, y_l)\) … movies!
Outline

1. Introduction
   - Problem Definition and Applications
   - Source Inversion as MIP with PDE Constraints
   - Problem Classification and Challenges

2. Early Theoretical & Numerical Results
   - Eliminating the PDE & State Variables
   - Numerical Experience with Source Inversion
   - Control Regularization: Not All Norms Are Equal
   - Heat Equation: Actuator Design

3. Rounding-Based Heuristic for Cloaking

4. Conclusions
Design of cloaking device on domain $\Omega$
- Cloak subdomain $\Omega_0$ (red dashes) by preventing (complex) wave from entering domain
- Design scatterer in subdomain $\hat{\Omega}$
  
  $w(x, y) \in \{0, 1\}$

PDE: 2D Helmholtz (over $\mathbb{C}$) with Robin boundary conditions

Incident wave is $\exp(ik_0y)$ for wavelength $k_0 = 6\pi$

where $i = \sqrt{-1}$
Topology Design of Cloaking Devices/Scatterers

Control: \( w = w(x, y) \) in \( \hat{\Omega} \)
States: \( u = u(x, y) \) in \( \Omega \)
Target: \( u_0 = u_0(x, y) \) in \( \Omega_0 \)

\[
\begin{align*}
\text{minimize} \quad & J(u) = \frac{1}{2} \| u + u_0 \|_{2, \Omega_0}^2 \\
\text{subject to} \quad & -\Delta u - k_0^2 (1 + qw) u = k_0^2 qw u_0 \quad \text{in } \Omega \\
& \frac{\partial u}{\partial n} - ik_0 u = 0 \quad \text{on } \partial \Omega \\
& w \in \{0, 1\} \quad \text{in } \hat{\Omega}.
\end{align*}
\]

Discretization: finite-differences with \( l = 3 \) nodes per scatter element, \( w(x, y) \).
Strip Rounding Heuristic

Cannot solve on reasonable mesh/domain with any MINLP solver.

**Algorithm: Strip Rounding Heuristic**
Solve continuous relaxation & initialize $i = 1$

for $i=1,...,N$ do
  Round a strip $w(x_i, y_j)$ for all $j$
  Resolve relaxation with $w(x_k, y)$ fixed for all $k \leq i$
end

Round fractional $w(x, y)$ following direction of wave
Strip Rounding Heuristic

Cannot solve on reasonable mesh/domain with any MINLP solver.

Algorithm: Strip Rounding Heuristic
Solve continuous relaxation & initialize $i = 1$
for $i=1,...,N$ do
  Round a strip $w(x_i, y_j)$ for all $j$
  Resolve relaxation with $w(x_k, y)$ fixed for all $k \leq i$
end

Round fractional $w(x, y)$ following direction of wave
Strip Rounding Heuristic

Cannot solve on reasonable mesh/domain with any MINLP solver.

**Algorithm: Strip Rounding Heuristic**
Solve continuous relaxation & initialize $i = 1$

for $i=1,\ldots,N$ do
   Round a strip $w(x_i, y_j)$ for all $j$
   Resolve relaxation with $w(x_k, y)$ fixed for all $k \leq i$
end

Round fractional $w(x, y)$ following direction of wave
Results for Strip Rounding

Scatterer, $w(x, y)$

States $u(x, y)$

... resolve PDE on finer mesh for fixed controls
... Solution Not Physical!

Coarse States

Resolved States

... not clear we’re getting the correct physics!
Conclusions

Mixed-Integer PDE-Constrained Optimization (MIPDECO)
- Class of challenging problems with important applications
  - Subsurface flow: oil recovery or environmental remediation
  - Design and operation of gas-/power-networks
- Classification: mesh-dependent vs. mesh-independent
- On-going work: Building library of test problems
  ... formulation matters: interplay of binary and continuous
- Elimination of PDE and state variables \( u(t, x, y, z) \)
- Discretized PDEs ⇒ huge MINLPs ... push solvers to limit

Outlook and Extensions
- Consider multi-level in space (network) and time
- Move toward truly multi-level approach similar to PDEs
- Interested in new UQ applications involving MIP & PDEs ...
Our five-year mission

To boldly go where no optimizer has gone before …

… to explore strange new PDEs & MIPs!


