Mixed-Integer PDE-Constrained Optimization

US-México Workshop on Optimization and Applications

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

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

January 7, 2016

Outline

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

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

- $u(t, x, y, z)$: 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

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

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

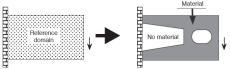
MIPDECO Warning

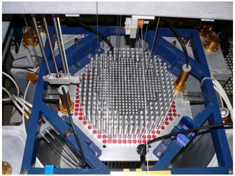
$w = w(t, x, y, z) \in \mathbb{Z}$ may be infinite-dimensional integers!

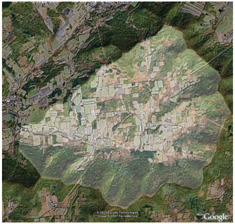
It's a MIP, Jim,
but not as we know it!

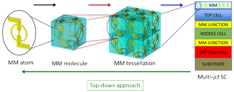
Grand-Challenge Applications of MIPDECO

- 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]
- Scheduling for disaster recovery: oil-spills [You and Leyffer, 2010] & wildfires [Donovan and Rideout, 2003]
- Design & control of gas networks, [De Wolf and Smeers, 2000, Martin et al., 2006, Zavala, 2014]
- Design of accelerators ... many more









Source Inversion as MIP with PDE Constraints

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

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

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

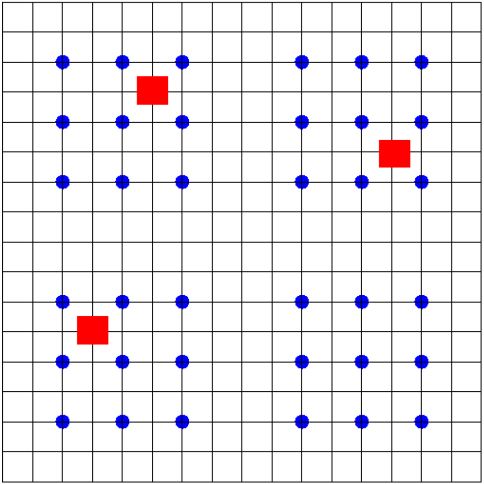
Consider 2D example with $\Omega = [0,1]^2$ and discretize PDE:

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

$$
\begin{cases}
\underset{u,w}{\text{minimize}} \quad J_h = \dfrac{h^2}{2} \displaystyle\sum_{i,j=0}^{N} \left(u_{i,j} - \bar{u}_{i,j}\right)^2 \\
\text{subject to } \dfrac{4u_{i,j} - u_{i,j-1} - u_{i,j+1} - u_{i-1,j} - u_{i+1,j}}{h^2} = \displaystyle\sum_{k,l=1}^{N} w_{kl} f_{kl}(ih, jh) \\
\qquad u_{0,j} = u_{N,j} = u_{i,0} = u_{i,N} = 0 \\
\qquad \displaystyle\sum_{k,l=1}^{N} w_{kl} \le S \text{ and } w_{kl} \in \{0,1\}
\end{cases}
$$

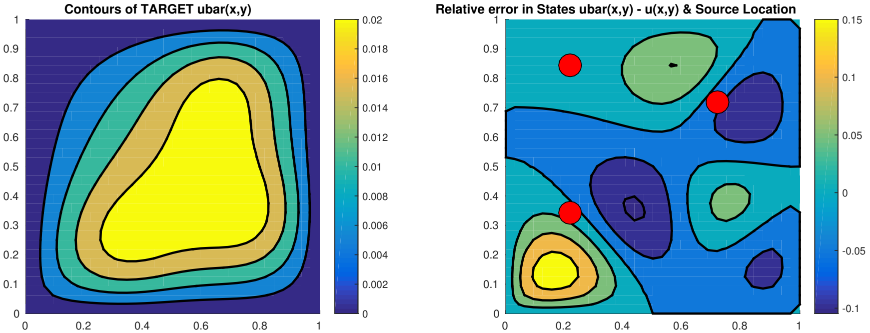
$\Rightarrow$ finite-dimensional (convex) MIQP

Source Inversion as MIP with PDE Constraints



Potential source locations (blue dots) on 16×16 mesh
Create target $\bar{u}$ using red square sources

Source Inversion as MIP with PDE Constraints



Target (3 sources), reconstructed sources, & error on 32 × 32 mesh

Outline

Mixed-Integer PDE-Constrained Optimization (MIPDECO)

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

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

Towards a problem characterization

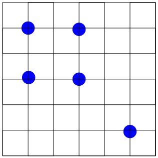
- **Type of PDE:** different classes of PDEs
  e.g. elliptic, parabolic, hyperbolic, nonlinear, ...
- **Class of Integers:** binary, general integers, etc
- **Type of Objective:** functional form of objective
- **Type of Constraints:** characterize c/s other than PDE
- **Discretization:** discretization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

Mesh-Independent & Mesh-Dependent Integers

Definition (Mesh-Independent & Mesh-Dependent Integers)

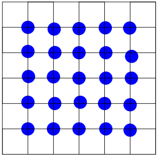
1. The integer variables are mesh-independent, iff number of integer variables is independent of the mesh.
2. The integer variables are mesh-dependent, iff the number of integer variables depends on the mesh.

Mesh-Independent



- Manageable tree size
- Theory possible

Mesh-Dependent



- Exploding tree size
- Theory???

... also mixed: mesh-dependent in time, t, but not space

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

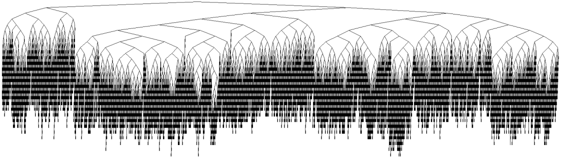
Discretization scheme (e.g. upwinding for wave equation) depends on direction of flow (integers).

- Application: gas network models with flow reversals

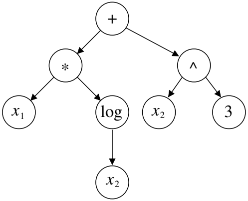
  ... open postdoc position at Argonne!

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!

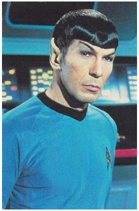


... $f(x_1, x_2) = x_1 \log(x_2) + x_2^3$

MIPDECO: Two Cultures Collide

Observation

PDE-optimization & MIP developed separately
$\Rightarrow$ different assumptions, methodologies, and computational kernels!



PDE-Optimization	**Mixed-Integer Programming**
Obtain good solutions efficiently	Deliver certificate of optimality
Nonlinear optimization: Newton's method	Combinatorial optimization: branch-and-cut
Iterative Krylov solvers	Factors & rank-one updates
Run on bleeding-edge HPC	Limited HPC developments

Potential for Disaster, or Opportunity for Innovation!

Outline

Source Inversion as MIP with PDE Constraints

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

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

- 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)

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

$\Leftrightarrow A\mathbf{u} = \sum w_{kl}\mathbf{f}_{kl}$, where $w_{kl} \in \{0,1\}$ only appear on RHS!

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

- $A\mathbf{u} = \sum_{k,l} w_{kl}\mathbf{f}_{kl} \Leftrightarrow \mathbf{u} = A^{-1}\left(\sum_{k,l} w_{kl}\mathbf{f}_{kl}\right) = \sum_{k,l} w_{kl}A^{-1}\mathbf{f}_{kl}$
- Solve $n^2 \ll 2^n$ PDEs: $\mathbf{u}^{(kl)} := A^{-1}\mathbf{f}_{kl}$
- Eliminate $\mathbf{u} = \sum_{k,l} w_{kl}\mathbf{u}^{(kl)}$ from Source Inversion

Cool MIPDECO Trick: Eliminating the PDE

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

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

- Eliminates the states $\mathbf{u}$ (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

Numerical Experience with Source Inversion

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

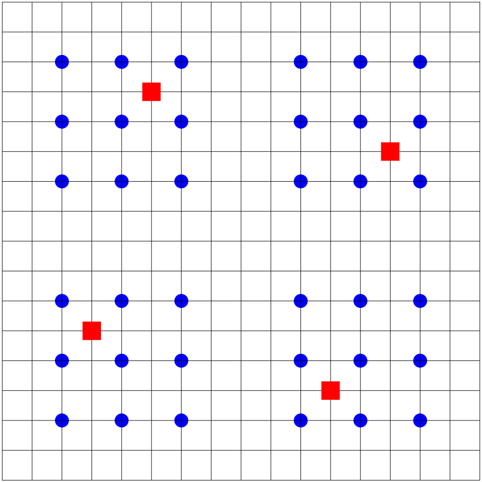
$$\begin{cases} \underset{u,w}{\text{minimize}} \quad \mathcal{J} = \frac{1}{2}\int_{\Omega} (u(w) - \bar{u})^2 d\Omega & \text{least-squares fit} \\ \text{subject to } -\Delta u = \sum_{k,l} w_{kl} f_{kl} \quad \text{in} \quad \Omega & \text{Poisson equation} \\ \qquad \sum_{k,l} w_{kl} \leq S \text{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

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

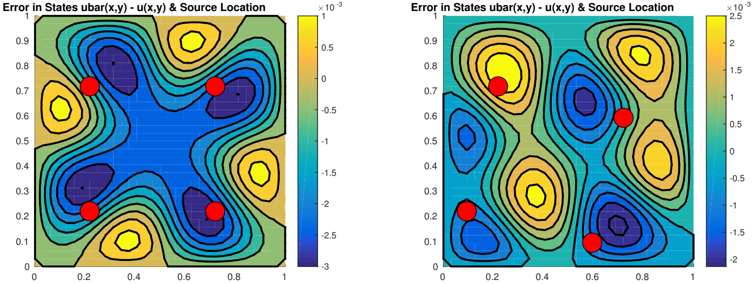
Knapsack Rounding

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

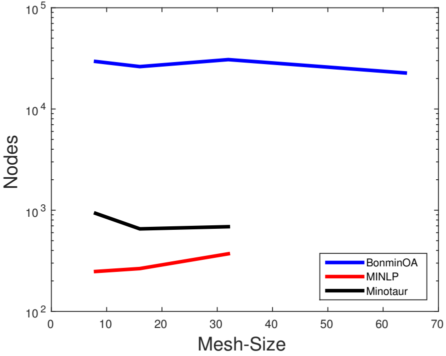


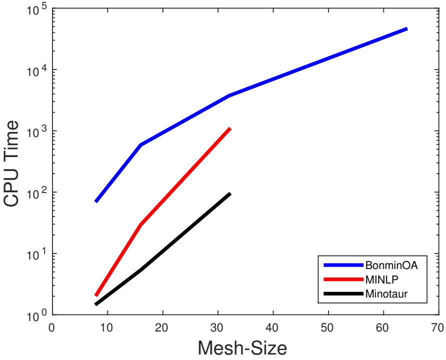
Knapsack-rounded NLP (left) and MINLP (right)

MINLP solution better: NLP-err $= 0.0388 > 0.0307 =$ MIP-err

Mesh-Independent Source Inversion: MINLP Solvers

Number of Nodes and CPU time for Increasing Mesh Sizes

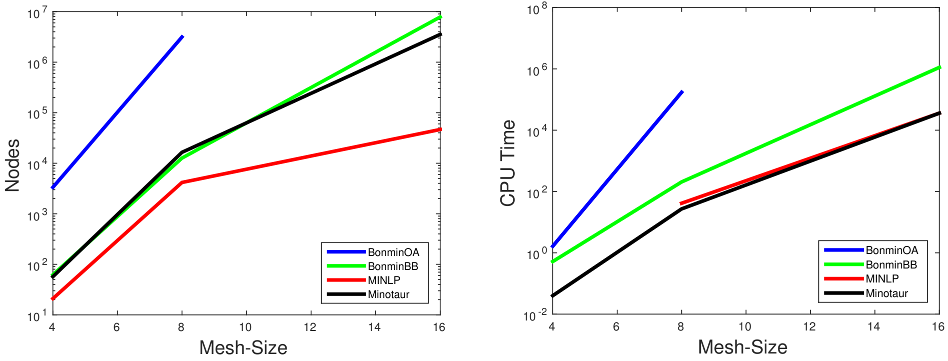




- Number of Nodes independent of mesh size!
- MINLP & Minotaur: filterSQP runs out of memory for $N \geq 32$
- BonminOA takes roughly 100 iterations ... quadratic objective

Mesh-Dependent (all) Source Inversion: MINLP Solvers

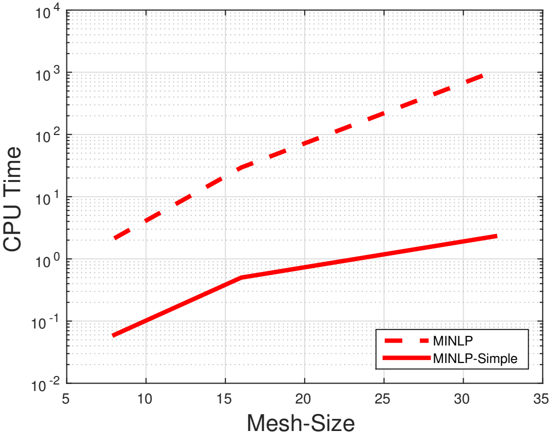
Number of Nodes and CPU time for Increasing Mesh Sizes



- Number of nodes explodes with mesh size!
- OA <BREAK> 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

	8×8	16×16	32×32
Presolve Time	0.05	1.30	62.51
Simplified Model	0.18	0.50	2.38
Total Simplified	0.23	1.80	64.89
Full PDE Model	2.10	29.43	1013.21

... 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

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

Control Regularization: Not All Norms Are Equal

Robin Boundary Control [OPTPDE, 2014] & [Tröltzsch, 1984]

$$
\begin{cases}
\underset{u,w}{\text{minimize}} & \|u - u_d\|^2_{L^2(\Omega)} + \alpha \|w\|^2_{L^x} \\
\text{subject to} & u_t - \Delta u = 0 \quad \text{in} \quad [0, T] \times \Omega \\
& u(0,x) = 0 \quad \text{in} \ \Omega \ \text{and} \ \frac{\partial u}{\partial x}(t,0) = 0 \ \text{in} \ (0, T) \\
& \frac{\partial u}{\partial x}(t,1) = b(w(t) - u(t,1)) \quad \text{in} \ (0, T) \\
& w(t) \in \{-1, 0, 1\}
\end{cases}
$$

L^1 or L^2 regularization term for control $w(t) \in \{-1, 0, 1\}$?

Good Norms for MIPs

MIP'ers prefer polyhedral norms ... promote integrality

- Old MIP trick: $w^2(t) = |w(t)|$ for $w(t) \in \{-1, 0, 1\}$
  $\Rightarrow$ L^1-norm same as L^2-norm on binary variables!

Not All Norms Are Equal

Consider Robin Boundary Control for increasing (x, t)-mesh

	CPU for L^2 Regularization			
Mesh	Minotaur	B-BB	B-Hyb	B-OA
8x8	0.04	0.80	2.54	126.81
16x16	6.61	72.21	1305.00	Time
32x32	Time	Time	Time	Time

Not All Norms Are Equal

Consider Robin Boundary Control for increasing (x, t)-mesh

	CPU for L^2 Regularization			
Mesh	Minotaur	B-BB	B-Hyb	B-OA
8x8	0.04	0.80	2.54	126.81
16x16	6.61	72.21	1305.00	Time
32x32	Time	Time	Time	Time

	CPU for L^1 Regularization			
Mesh	Minotaur	B-BB	B-Hyb	B-OA
8x8	0.03	0.48	0.21	0.04
16x16	0.11	3.62	0.66	0.20
32x32	0.18	62.66	3.53	0.74

- L^1 regularization is equivalent to L^2, but faster
- Many fewer nodes in tree-searches $\Rightarrow$ solve up to 128×128

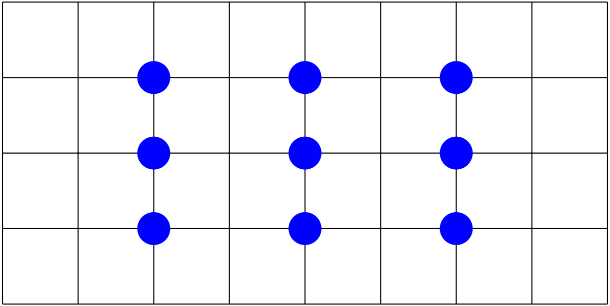
Outline

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 $l = 1, \ldots, L$

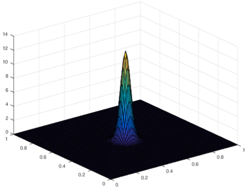
Problem 2: Actuator Placement and Operation

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

$$\begin{aligned}
\underset{u,v,w}{\text{minimize}} \quad & \|u(t_f,\cdot)\|_\Omega^2 + 2\|u\|_{T\times\Omega}^2 + \frac{1}{500}\|v\|_T^2 \\
\text{subject to } & \frac{\partial u}{\partial t} - \kappa \Delta u = \sum_{l=1}^{L} v_l(t) f_l \;\; \text{in } T \times \Omega \\
& w_l(t) \in \{0,1\}, \quad \sum_{l=1}^{L} w_l(t) \le W, \; \forall t \in T \\
& L w_l(t) \le v_l(t) \le U w_l(t), \; \forall l = 1,\ldots,L, \; \forall t \in T
\end{aligned}$$

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!

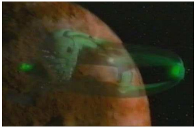
Outline

Topology Design of Cloaking Devices/Scatterers

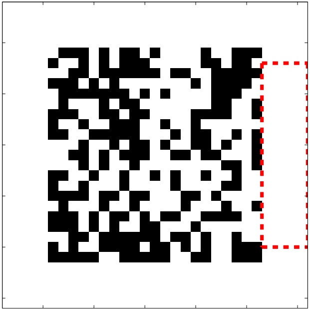
Design of cloaking device on domain Ω

- Cloak subdomain Ω_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 $\exp(ik_0 y)$ for wavelength $k_0 = 6\pi$

where $i = \sqrt{-1}$

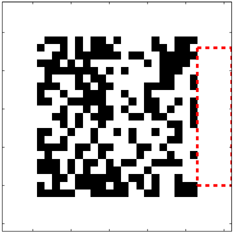


Romulan Warbird



Scatterer

Topology Design of Cloaking Devices/Scatterers



Control: $w = w(x,y)$ in $\hat{\Omega}$
States: $u = u(x,y)$ in Ω
Target: $u_0 = u_0(x,y)$ in Ω_0

$$
\begin{aligned}
\underset{u,v,w}{\text{minimize}} \quad & J(u) = \tfrac{1}{2}\|u + u_0\|^2_{2,\Omega_0} \\
\text{subject to} \quad & -\Delta u - k_0^2(1 + qw)u = k_0^2 q w u_0 \quad \text{in } \Omega \\
& \tfrac{\partial u}{\partial n} - ik_0 u = 0 \quad \text{on } \partial\Omega \\
& w \in \{0,1\} \quad \text{in } \hat{\Omega}.
\end{aligned}
$$

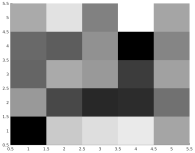
Discretization: finite-differences with $l = 3$ nodes per scatter element, $w(x,y)$.

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

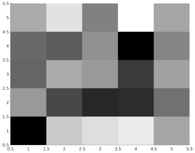


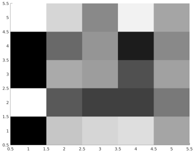
Strip Rounding Heuristic

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



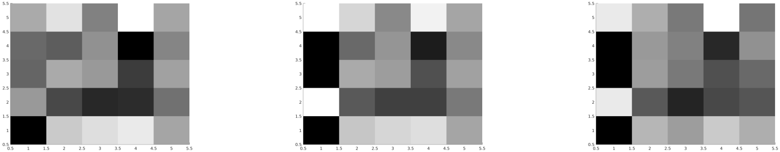


Strip Rounding Heuristic

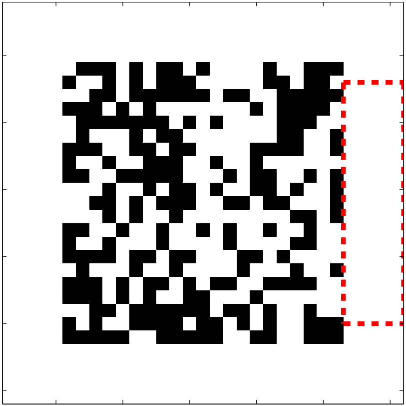
Cannot solve on reasonable mesh/domain with any MINLP solver.

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

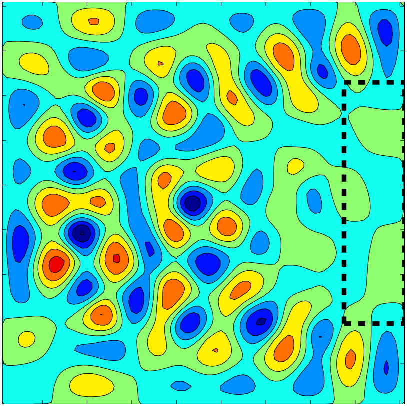
Round fractional $w(x, y)$ following direction of wave



Results for Strip Rounding



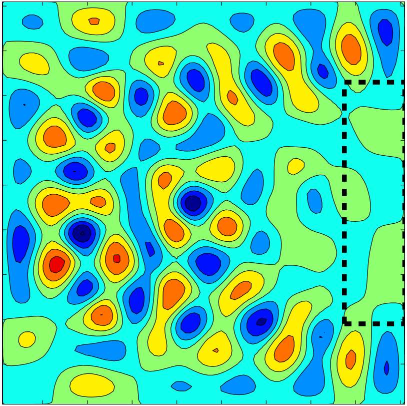
Scatterer, $w(x, y)$



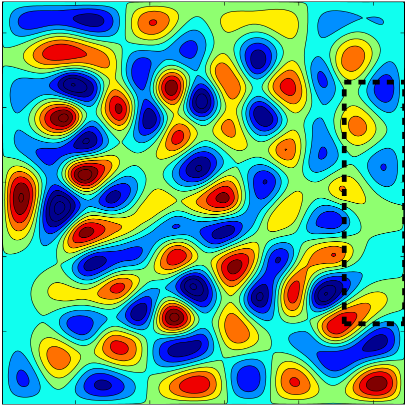
States $u(x, y)$

... 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
- On-going work: Building library of test problems
- Classification: mesh-dependent vs. mesh-independent
- Elimination of PDE and state variables $u(t, x, y, z)$
- Discretized PDEs $\Rightarrow$ huge MINLPs ... push solvers to limit
- Need new ideas, solvers, software for real applications

Outlook and Extensions

- Consider multi-level in space (network) and time
- Move toward truly multi-level approach similar to PDEs

... our five-year mission ...

To boldly go where no optimizer has gone before ...

... to explore strange new PDEs & MIPs!

Committee, T. (2010).
Advanced fueld pellet materials and fuel rod design for water cooled reactors.
Technical report, International Atomic Energy Agency.

De Wolf, D. and Smeers, Y. (2000).
The gas transmission problem solved by an extension of the simplex algorithm.
Management Science, 46:1454–1465.

Donovan, G. and Rideout, D. (2003).
An integer programming model to optimize resource allocation for wildfire contrainment.
Forest Science, 61(2).

Fipki, S. and Celi, A. (2008).
The use of multilateral well designs for improved recovery in heavy oil reservoirs.
In *IADV/SPE Conference and Exhibition*, Orlanda, Florida. SPE.

Martin, A., Möller, M., and Moritz, S. (2006).
Mixed integer models for the stationary case of gas network optimization.
Mathematical Programming, 105:563–582.

OPTPDE (2014).
OPTPDE — a collection of problems in PDE-constrained optimization.
`http://www.optpde.net`.

Ozdogan, U. (2004).
Optimization of well placement under time-dependent uncertainty.
Master's thesis, Stanford University.

Reinke, C. M., la Mata Luque, T. M. D., Su, M. F., Sinclair, M. B., and El-Kady, I. (2011).
Group-theory approach to tailored electromagnetic properties of metamaterials: An inverse-problem solution.
Physical Review E, 83(6):066603–1–18.

Sigmund, O. and Maute, K. (2013).
Topology optimization approaches: A comparative review.
Structural and Multidisciplinary Optimization, 48(6):1031–1055.

Tröltzsch, F. (1984).
The generalized bang-bang-principle and the numerical solution of a parabolic boundary-control problem with constraints on the control and the state.
Zeitschrift für Angewandte Mathematik und Mechanik, 64(12):551–556.

You, F. and Leyffer, S. (2010).
Oil spill response planning with MINLP.
SIAG/OPT Views-and-News, 21(2):1–8.

Zavala, V. M. (2014).
Stochastic optimal control model for natural gas networks.
Computers & Chemical Engineering, 64:103–113.

Zhang, P., Romero, D., Beck, J., and Amon, C. (2013).
Integration of AI and OR Techniques in Contraint Programming for Combinatorial Optimization Problems, chapter Solving Wind Farm Layout Optimization with Mixed Integer Programming and Constraint Programming.
Springer Verlag.