Mixed-Integer Nonlinear Optimization: Applications, Algorithms, and Computation VI

Sven Leyffer

Mathematics & Computer Science Division
Argonne National Laboratory

Graduate School in
Systems, Optimization, Control and Networks
Université catholique de Louvain
February 2013

Outline

1. Heuristics for MINLP
   - Search Heuristics for MINLP
   - Improvement Heuristics for MINLP
2. Mixed-Integer Optimal Control [Christian Kirches]
   - Mixed-Integer Optimal Control Applications
   - Methods and Theory for MI Optimal Control
3. Software Tools and Online Resources
4. Beyond Mixed-Integer Nonlinear Optimization
5. Summary, Conclusions, and Thanks

Outline

Heuristics for Solving MINLPs

Some real-world applications cannot be solved

- Problems are too large, generate huge search tree
- Problems must be solved in real time (e.g. control later)

Use heuristics to get feasible point without optimality guarantees

Heuristics accelerate deterministic techniques

- Identify good incumbent $\Rightarrow$ upper bound
- Upper bound helps prune search tree
- Incumbent also used in guided dives
- More effective bounds tightening for nonconvex MINLP

Two Classes of Heuristics for Solving MINLPs

Probabilistic Methods ... random choice at every iteration

- Simulated annealing [Kirkpatrick et al., 1983]
- Genetic algorithms [Goldberg, 1989]
- Tabu search [Glover, 1989]
- Particle-swarm optimization [Kennedy and Eberhart, 1995]
- Ant colony optimization [Dorigo et al., 1996]

... not discussed here (require less info on problem)

Deterministic Methods ... discussed here

- Simple heuristic: run any algorithm for fixed time limit
- Search heuristics: search without any known solutions
- Improvement heuristics: improve upon a given solution

... may also include random elements

Heuristics for Solving MINLPs: Notation

Mixed-Integer Nonlinear Program (MINLP)

$$(P) \quad \min_x \; f(x) \quad \text{s.t. } c(x) \le 0, \; x \in X, \; x_i \in \mathbb{Z} \; \forall \, i \in I$$

NLP subproblem for fixed integers $x_I^{(j)}$

$$\text{NLP}(x_I^{(j)}) \quad \min_x \; f(x) \quad \text{s.t. } c(x) \le 0 \quad x \in X \quad \text{and } x_I = x_I^{(j)}$$

Notation

- x^*: current incumbent, feasible in (P)
- x': (local) solution of continuous relaxation of (P)
- $x^\diamond$: solution to polyhedral relaxation of (P)
- $x^{(j)}$: (local) solution of NLP($x_I^{(j)}$)

Heuristics for Solving MINLPs: Notation

Mixed-Integer Nonlinear Program (MINLP)

$$(P) \quad \min_x \; f(x) \quad \text{s.t. } c(x) \le 0,\; x \in X,\; x_i \in \mathbb{Z}\; \forall\, i \in I$$

Example of polyhedral relaxation: OA master $M(\mathcal{X}^k)$

$$M(\mathcal{X}^k) \begin{cases} \underset{\eta,x}{\text{minimize}} & \eta, \\ \text{subject to} & \eta \le U^k - \epsilon \\ & \eta \ge f^{(j)} + \nabla f^{(j)^T}(x - x^{(j)}),\; \forall x^{(j)} \in \mathcal{X}^k \\ & 0 \ge c^{(j)} + \nabla c^{(j)^T}(x - x^{(j)}),\; \forall x^{(j)} \in \mathcal{X}^k \\ & x \in X, \quad x_i \in \mathbb{Z},\; \forall i \in I. \end{cases}$$

... or (MI)LP from reformulation of nonconvex factorable MINLP

Outline

MILP-Based Rounding

Given solution x' of continuous relaxation of MINLP, (P)

- Round fractional integer values: $\bar{x} := \text{round}(x')$
- Unfortunately, $\bar{x}$ usually not feasible in MINLP

Example of Rounding Failure

Consider SOS-1 variables

$$x_i \in \{0,1\},\ i = 1, \ldots, p,\ \sum_{i=1}^{p} x_i = 1$$

then often $x'_i \simeq 1/p$ in NLP/LP relaxation

$$\Rightarrow \text{round}(x') = \bar{x} = 0 \quad \text{and} \quad \sum \bar{x}_i = 0 \neq 1$$

is infeasible in MINLP

Could randomize the rounding, but won't help

Better remedy: solve MILP to round [Nannicini and Belotti, 2012]

MILP-Based Rounding

Define polyhedral (MILP) relaxation:

$$(R) \quad \underset{x}{\text{minimize}} \ \|x - x'\|_1 \quad \text{s.t. } x \in \ \text{polyhedral relaxn of MINLP}$$

MILP-Based Rounding for Nonconvex MINLP

$x' \leftarrow$ solution of NLP relaxation

$(R_0) \leftarrow (R)$ initialize MILP relaxation ($k \leftarrow 0$)

while *no feasible solution found & we have time* **do**

- $x^\diamond \leftarrow$ (approx.) solution of relaxation (R_k)
- $x' \leftarrow$ (local) solution of NLP($x_I^\diamond$)
- Add "no-good" cut to (R_k); set $k \leftarrow k+1$

No-good cuts ... are no good (i.e. weak)

$$\sum_{i \in I} |x_i - x_i^\diamond| \geq 1 \ \Leftrightarrow \ \sum_{i \in I : x_i^\diamond = 0} x_i + \sum_{i \in I : x_i^\diamond = 1} (1 - x_i) \geq 1$$

for binary variables (if $x_i \in \mathbb{Z}$ then add SOS-1 to (R))

MINLP Feasibility Pump

[Fischetti et al., 2005] for convex MINLP

Polyhedral relaxation (R) is ℓ_1-OA $M_1(\mathcal{X}^k)$:

$$M_1(\mathcal{X}^k) \begin{cases} \underset{\eta,x}{\text{minimize}} & \|x_I - x_I'\|_1 \\ \text{subject to} & \eta \le U^k - \epsilon \\ & \eta \ge f^{(j)} + \nabla f^{(j)^T}(x - x^{(j)}), \ \forall x^{(j)} \in \mathcal{X}^k \\ & 0 \ge c^{(j)} + \nabla c^{(j)^T}(x - x^{(j)}), \ \forall x^{(j)} \in \mathcal{X}^k \\ & x \in X, \quad x_i \in \mathbb{Z}, \ \forall i \in I. \end{cases}$$

... standard OA with ℓ_1 distance objective (for $x_I \in \{0,1\}^p$)

- Update $\mathcal{X}^k$ with new outer approximations
- Can prove finite termination at optimum
- See [Bonami et al., 2009] & BONMIN solver

MINLP Feasibility Pump

Define nonlinear ℓ_2 subproblem for fixed $x_I^{(j)}$

$$\text{NLP}(x_I^{(j)}) \quad \min_x \ \|x - x^\diamond\|_2 \quad \text{s.t. } c(x) \le 0 \quad x \in X \quad \text{and } x_I = x_I^{(j)}$$

Feasibility Pump for Convex MINLP

$x' \leftarrow$ solution of NLP relaxation

Initialize MILP relaxation $M_1(\mathcal{X}^0)$; set $k \leftarrow 0$

while *no feasible solution found & we have time* **do**

- $x^{(k)} := x^\diamond \leftarrow$ (approx.) solution of relaxation $M_1(\mathcal{X}^k)$
- $x_C^{(k)} := x'_C \leftarrow$ (local) solution of NLP$(x_I^{(k)})$
- Add outer approximations to $M_1(\mathcal{X}^k)$; set $k \leftarrow k+1$

MILP-based rounding extends MINLP-FP to nonconvex problems
... without improving MILP

Undercover Heuristic

Undercover also tries to apply MILP techniques

1. Linearize the (nonconvex) MINLP by fixing variables $\Rightarrow$ MILP
2. Solve MILP (exactly) $\Rightarrow$ solution will be feasible in MINLP

Goal is to fix as few variables as possible

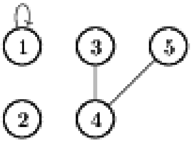
- Sparsity pattern of H, Hessian of the Lagrangian,
  ... identifies nonlinear variables
- Define graph $G(V, E)$ such that:
  - Vertex $v_i \in V, i = 1, \ldots, n$ denote variables x_i
  - $\exists$ edge e_{ij} iff $H_{ij} \neq 0$ nonzero entry

$\Rightarrow$ minimal vertex cover of H gives minimum number of variables that must be fixed

Undercover Heuristic

Example:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x_1^3 + x_2 + x_3 x_4 + x_5 \\ \text{subject to} \quad & x_4 x_5 \geq 1 \\ & x_1, x_2, x_3, x_4, x_5 \in [0, 10] \end{aligned}$$



Hessian Graph

- Linearized by fixing x_1, x_3, and x_5 ... not optimal
- Vertex cover $\Rightarrow$ fix x_1 and x_4

Extending Undercover Heuristic?

Possible Extensions:

- Are there other ways to "linearize" the MINLP?
- Can we exploit generalizations of "sparsity", e.g. group partial separability?

$$f(x) = \sum_{j=1}^{q} g_j \left(a_j^T x + b_j + \sum_{i \in \mathcal{E}_j} f_i(x_{[i]}) \right)$$

  where $g_j : \mathbb{R} \to \mathbb{R}$ univariate and $x_{[i]} \in \mathbb{R}^{n_i}$ with $n_i \ll n$

- Can we use semi-definite relaxations or QP?

... gps is potentially interesting area of research

Exercise: Code Undercover and its variations in MINOTAUR!

Diving Heuristics

Basic idea: Perform quick depth-first exploration of MINLP

1. Solve NLP relaxation $\to x'$
2. Fix (one/some) integer variables x_i to $\lfloor x'_i \rfloor$ or $\lceil x'_i \rceil$
3. Re-solve NLP, and repeat until all x_I integral

[Bonami and Gonçalves, 2012] diving heuristics for MINLP

- Fractional diving: dive on smallest factional value

$$i = \underset{j}{\operatorname{argmin}} \; |x'_j - [x'_j]|$$

  where $[x'_j] = \min(x'_j - \lfloor x'_j \rfloor \, , \; \lceil x'_j \rceil - x'_j)$ closest integer
- Vector-length diving chooses variable with small ∇f that appears in as many nonlinear functions as possible

Structured Diving

Snag: Rounding-based heuristics destroy feasibility

Simple idea: round integer structure, e.g. SOS-1

$$x_i \in \{0, 1\}, \quad 1 = \sum_{i=1}^{p} x_i, \quad z = \sum_{i=1}^{p} \alpha_i x_i$$

Rounding SOS-1, given x' fractional

1. Compute $z' = \sum_{i=1}^{p} \alpha_i x'_i$ weighted sum
2. Find i_0 such that $\alpha_{i_0} \leq z' < \alpha_{i_0+1}$
3. Round to $z \geq \alpha_{i_0+1}$ or $z \leq \alpha_{i_0}$ gives

$$\sum_{i \leq i_0} x_i = 0 \quad \bigvee \quad \sum_{i \geq i_0+1} x_i = 0$$

Preliminary Results with Structured Diving

Student implementation in MINOTAUR

	Structured		Standard	
Instance	Time	Objective	Time	Objective
space25	0.48	487.07	4.16	528.12
space25a	0.25	485.57	0.43	623.83
space-25	1.41	485.57	3.57	615.56
space-25-r	0.48	487.07	1.45	625.56
stockcycle	0.53	125953	0.67	140561
feedlock	0.12	0	0.35	0

$\Rightarrow$ find better objective value faster!

- Can detect and exploit other structure: SOS-2, ...
- Surely CPLEX, GuRoBi et al already do this!

Outline

Local Branching

Heuristic for binary MINLP: $x_i \in \{0,1\}, \ \forall i \in I$
Given feasible incumbent x^* find a better solution
For $k \in \mathbb{Z}$ consider disjunction

$$\|x_I - x_I^*\|_1 \leq k \quad \bigvee \quad \|x_I - x_I^*\|_1 \geq k+1$$

... like "trust-region" in NLP; equivalent to

$$\sum_{i \in I : x_i^* = 0} x_i + \sum_{i \in I : x_i^* = 1} (1 - x_i) \leq k \quad \vee \quad \sum_{i \in I : x_i^* = 0} x_i + \sum_{i \in I : x_i^* = 1} (1 - x_i) \geq k+1$$

- Construct "easy ... $\leq k$ branch" for $10 \leq k \leq 20$ & solve it!
- Rigorous, if we explore "hard ... $\geq k+1$ branch too.

RINS: Relaxation Induced Neighbourhood Search

MILP RINS [Danna et al., 2005] heuristic:

- Given x^* MILP feasible and x' solution of LP relaxation
- Fix variable x_i for $i \in I$ iff $x'_i = x^*_i$
- Solve reduced MILP

Extension to convex MINLP [Bonami and Gonçalves, 2012]

- Given x^* MINLP feasible and x' solution of NLP relaxation
- Fix variable x_i for $i \in I$ iff $x'_i = x^*_i$
- Solve small MINLP with LP/NLP-BnB (see Lecture II)

Look for nearby integer solutions

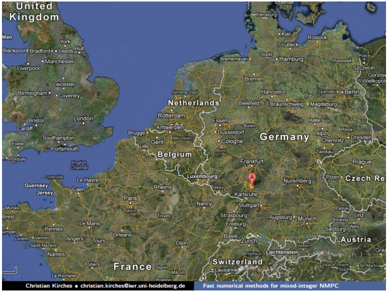
Outline

All Slides Courtesy of Christian Kirches

Christian Kirches

Universität Heidelberg

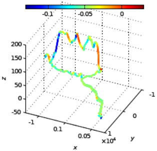
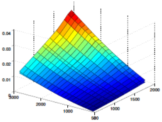
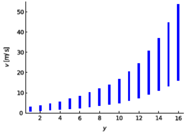
UNIVERSITÄT
HEIDELBERG
Zukunft. Seit 1386.



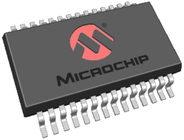
Outline

Predictive Cruise Control for Trucks

Minimize Time/Energy when driving with automatic gear choice

Online computation of MI feedback controls on moving horizon

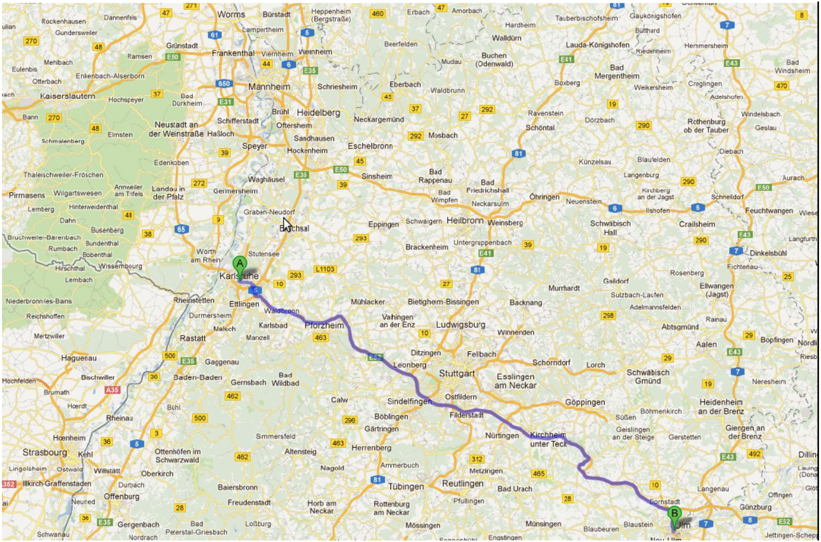


$\leftarrow$ MIP Control

State Estimation $\rightarrow$

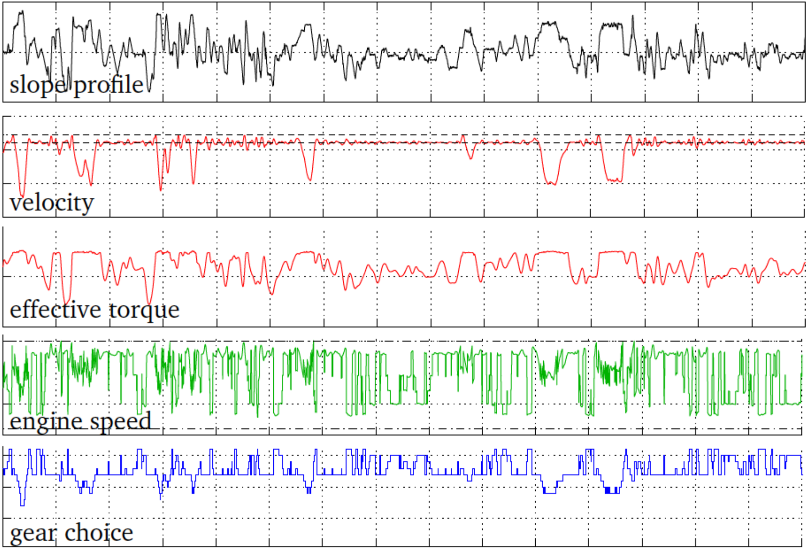
8 available gears, 20 possible shifts $\Rightarrow$ $> 10^{18}$ continuous problems

Predictive Cruise Control for Trucks



Autobahn (A8)

Predictive Cruise Control for Trucks

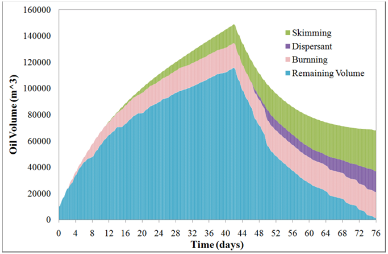


Solution

Optimal Response to Catastrophic Oil Spills

- Predict trajectories of oil slick & optimal cleanup schedule
- Combines optimal control & mixed-integer model:

- Time-dependent properties:
  - oil physiochemical properties
  - amount & rate of spill
  - hydrodynamics
  - weather conditions
- Discrete decisions (response): clean-up schedule and type
- Regulatory constraints



Outline

Mixed-Integer Optimal Control Problems (MIOCPs)

Dynamic & switched process control problem on horizon $[0, T]$:

$$
\begin{aligned}
\min_{x,z,\,u,\,v} \quad & \int_0^T l(x(t), z(t), u(t), v(t), p)\,\mathrm{d}t + m(x(T), z(t), p) & \\
\text{s.t.} \quad \dot{x}(t) &= f(x(t), z(t), u(t), v(t), p) & t \in [0, T] \\
0 &= g(x(t), z(t), u(t), v(t), p) & t \in [0, T] \\
0 &= x(0) - \hat{x}_0 & \\
0 &\leq c(x(t), z(t), u(t), v(t), p) & t \in [0, T] \\
0 &\leq d(x(t), z(t), u(t), p) & t \in [0, T] \\
0 &\leqq r(\{x(t_i), z(t)\}_{0 \leq i \leq N}, p) & \{t_i\} \in [0, T] \\
v(t) & \qquad \in \Omega & t \in [0, T]
\end{aligned}
$$

Objective: typically economic/tracking part l and terminal weight part m

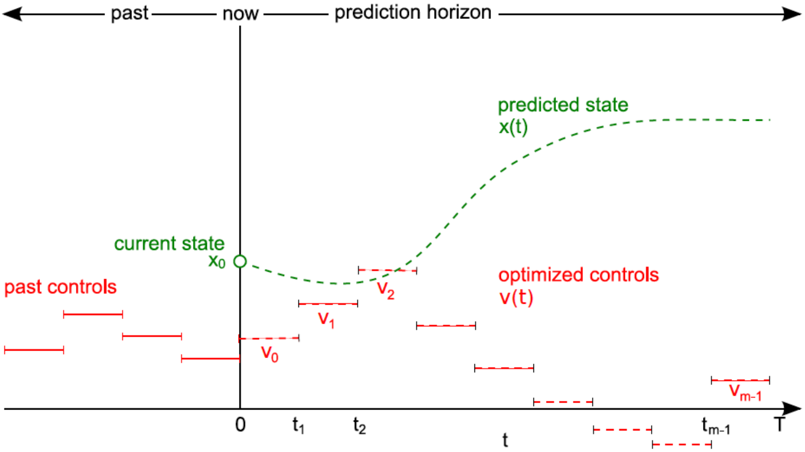
Constraints: Initial value, path constraints c, d, point constraints r at t_i

Dynamic process $(x(\cdot), z(\cdot))$ modeled by an ODE/DAE system f

Continuous controls $u(\cdot)$ from set $\mathcal{U} \subset \mathbb{R}^{n_u}$,

Controls $v(\cdot)$ from discrete set $\Omega := \{v^1, \ldots, v^{n_\Omega}\} \subset \mathbb{R}^{n_v}$

Model-Predictive Control Scheme



Challenges of Mixed-Integer Optimal Control

Solve MIOCP in real-time (moving horizon)

- Direct discretization of DAEs leads to huge MINLPs
- Multiple shooting gives simulation constraints
- Enormous combinatorial space:
  - Can switch discrete control at every $t \in [0, T]$
  - Single on/off control with n time-steps has 2^n complexity
- OCPs readily modeled in AMPL/GAMS ... or are they???

AMPL Discretized Optimal Control Problem

```
param nc > 0, integer;                # number of collocation points
param nd > 0, integer;                # order of the differential equation
param nh > 0, integer;                # number of partition intervals

param rho {1..nc};                    # roots of k-th degree Legendre polynomial
param bc {1..2,1..2};                 # boundary conditions
param tf;                             # ODEs defined in [0,tf]
param h := tf/nh;                     # uniform interval length
param t {i in 1..nh+1} := (i-1)*h;    # partition

param fact {j in 0..nc+nd} := if j = 0 then 1 else (prod{i in 1..j} i);

param R >= 0;                         # Reynolds number

var v {i in 1..nh,j in 1..nd};
var w {1..nh,1..nc};

var uc {i in 1..nh, j in 1..nc, s in 1..nd} =
  v[i,s] + h*sum {k in 1..nc} w[i,k]*(rho[j]^k/fact[k]);

var Duc {i in 1..nh, j in 1..nc, s in 1..nd} =
  sum {k in s..nd} v[i,k]*((rho[j]*h)^(k-s)/fact[k-s]) + h^(nd-s+1)*
  sum {k in 1..nc} w[i,k]*(rho[j]^(k+nd-s)/fact[k+nd-s]);
```

AMPL Discretized Optimal Control Problem

```
minimize constant_objective: 1.0;

subject to bc_1: v[1,1] = bc[1,1];
subject to bc_2: v[1,2] = bc[2,1];
subject to bc_3:
  sum {k in 1..nd} v[nh,k]*(h^(k-1)/fact[k-1]) + h^nd*
  sum {k in 1..nc} w[nh,k]/fact[k+nd-1] = bc[1,2];
subject to bc_4:
  sum {k in 2..nd} v[nh,k]*(h^(k-2)/fact[k-2]) + h^(nd-1)*
  sum {k in 1..nc} w[nh,k]/fact[k+nd-2] = bc[2,2];

subject to continuity {i in 1..nh-1, s in 1..nd}:
  sum {k in s..nd} v[i,k]*(h^(k-s)/fact[k-s]) + h^(nd-s+1)*
  sum {k in 1..nc} w[i,k]/fact[k+nd-s] = v[i+1,s];

subject to collocation {i in 1..nh, j in 1..nc}:
  sum {k in 1..nc} w[i,k]*(rho[j]^(k-1)/fact[k-1]) =
  R*(Duc[i,j,2]*Duc[i,j,3] - Duc[i,j,1]*Duc[i,j,4]);
```

Fluid flow in a channel problem with AMPL

TACO: Toolkit for AMPL Control Optimization

Extensions to AMPL for DAEs and optimal control

- Reads AMPL `stub.nl` file & detects DAE structure
- Open-source solver interface
- 1st & 2nd derivatives & presolve readily available
- Interfaced to MUSCOD-II; available on NEOS server

Optimal Control Problem	TACO/AMPL model
$\min\limits_{x,u,p} \int_0^3 x(t)^2 + u(t)^2 \, \mathrm{d}t$	$\texttt{min Lagrange} : \texttt{integral}(\texttt{x}^2 + \texttt{u}^2, 3);$
s.t. $\dot{x}(t) = (x(t) + p)x(t) - u(t),$	$\texttt{s.t. ODE} : \texttt{diff}(\texttt{x}, \texttt{t}) = (\texttt{x} + \texttt{p}) * \texttt{x} - \texttt{u};$
$x(0) = -0.05,$	$\texttt{IC} : \texttt{eval}(\texttt{x}, 0) = -0.05;$

TACO: Toolkit for AMPL Control Optimization

```
include OptimalControl.mod

var t;                  # independent time
param tf;               # ODEs defined in [0,tf]
var u{1..4} := 0;       # differential states
param R >= 0;           # Reynolds number

subject to

d1: diff(u[1],t) = u[2];
d2: diff(u[2],t) = u[3];
d3: diff(u[3],t) = u[4];
d4: diff(u[4],t) = R*(u[2]*u[3] - u[1]*u[4]);

u1s: eval(u[1],0) = bc[1,1];
u2s: eval(u[2],0) = bc[2,1];
u1e: eval(u[1],tf) = bc[1,2];
u2e: eval(u[2],tf) = bc[2,2];
```

Fluid flow in a channel problem with TACO

Challenges of Mixed-Integer Optimal Control

Solve MIOCP in real-time (moving horizon)

- Direct discretization of DAEs leads to huge MINLPs
- Multiple shooting gives "black-box" constraints
- Enormous combinatorial space:
  - Can switch discrete control at every $t \in [0, T]$
  - Single on/off control with n time-steps has 2^n complexity
- OCPs readily modeled in AMPL/GAMS ... with TACO!

How tractable are mixed-integer OCPs?

Partial Outer Convexification for MIOCP (Function Space)

Introduce *convex multipliers* $\omega_j(\cdot) \in \{0,1\}$ for $v(\cdot) = v^j \in \Omega$, $\forall j$:

$$\text{bijection}: \quad v(t) = v^j \in \Omega \quad \Longleftrightarrow \quad \omega_j(t) = 1, \; \sum_{k=1}^{n_\Omega} \omega_k(t) = 1$$

Model of MIOCP as partially convexified optimal control problem:

$$\begin{aligned}
\min_{x(\cdot),\; u(\cdot),\; \omega(\cdot)} \quad & \int_0^T \sum_{j=1}^{n_\Omega} \omega_j(t) \cdot l(x(t), u(t), v^j, p) \,\mathrm{d}t + m(x(T), p) \\
\text{s.t.} \quad & \dot{x}(t) = \sum\nolimits_{j=1}^{n_\Omega} \omega_j(t) \cdot f(x(t), u(t), v^j, p) && t \in [0, T] \\
& 0 = x(0) - \hat{x}_0(\tau) \\
& 0 \le \omega_j(t) \cdot c(x(t), u(t), v^j, p), \quad \forall j, && t \in [0, T] \\
& 0 \le d(x(t), u(t), p), && t \in [0, T] \\
& \omega(t) \in \{0,1\}^{n_\Omega}, \quad 1 = \sum\nolimits_{j=1}^{n_\Omega} \omega_j(t) && t \in [0, T]
\end{aligned}$$

Dirty secret: n_Ω is huge ... enumerate combinatorial space!

Partial Outer Convexification for MIOCP (Function Space)

Introduce *convex multipliers* $\omega_j(\cdot) \in \{0,1\}$ for $v(\cdot) = v^j \in \Omega$, $\forall j$:

$$\text{bijection}: \quad v(t) = v^j \in \Omega \quad \Longleftrightarrow \quad \omega_j(t) = 1, \; \sum_{k=1}^{n_\Omega} \omega_k(t) = 1$$

Relaxation yields continuous, larger optimal control problem:

$$\begin{aligned}
\min_{x(\cdot),\; u(\cdot),\; \alpha(\cdot)} \quad & \int_0^T \sum_{j=1}^{n_\Omega} \alpha_j(t) \cdot l(x(t), u(t), v^j, p)\, \mathrm{d}t + m(x(T), p) \\
\text{s.t.} \quad & \dot{x}(t) = \sum\nolimits_{j=1}^{n_\Omega} \alpha_j(t) \cdot f(x(t), u(t), v^j, p) && t \in [0, T] \\
& 0 = x(0) - \hat{x}_0(\tau) \\
& 0 \le \alpha_j(t) \cdot c(x(t), u(t), v^j, p), \quad \forall j, && t \in [0, T] \\
& 0 \le d(x(t), u(t), p) && t \in [0, T] \\
& \alpha(t) \in [0,1]^{n_\Omega}, \quad 1 = \sum\nolimits_{j=1}^{n_\Omega} \alpha_j(t) && t \in [0, T]
\end{aligned}$$

Continuous Relaxation is large ... but OK!

Approximation Theorem for Outer Convexification

Theorem (Approximation Theorem)

$(x^*(\cdot), u^*(\cdot), \alpha^*(\cdot))$ optimal solution of relaxation of convexified MIOCP with objective Φ_{CR}.
$\Rightarrow \forall \varepsilon > 0 \; \exists \; \omega_\varepsilon \in \{0,1\}^{n_\Omega}$, $x_\varepsilon(\cdot)$ such that $(x_\varepsilon(\cdot), u^*(\cdot), \omega_\varepsilon(\cdot))$ feasible solution of convexified MIOCP with objective Φ_{CB} and

$$(\Phi_{CR} \leq) \; \Phi_{CB} \leq \Phi_{CR} + \varepsilon.$$

Consequences:

- Relaxed solution arbitrarily close to integer feasible solution
- Control $\omega_\varepsilon(\cdot)$ and hence $v(\cdot)$ may switch infinitely often
- $x_\varepsilon(\cdot)$ violates pure state constraints $d(\cdot)$ by at most $L_d \varepsilon$.

Outer Convexification Example

Energy optimal control of rocket car

Standard MIOCP

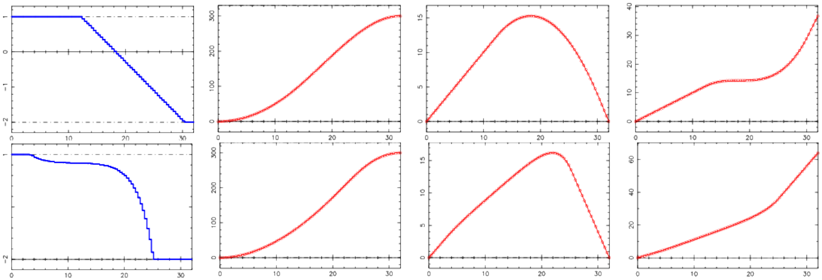
$$
\begin{aligned}
\min_{x,v}\ & x_2(32) \\
\text{s.t.}\ & \dot{x}_0(t) = x_1(t) && \forall t \\
& \dot{x}_1(t) = v(t) && \forall t \\
& \dot{x}_2(t) = v^2(t) && \forall t \\
& x(0) = (0,0,0) \\
& x_0(32) = 300 \\
& v(t) \in \{-2, 1\} && \forall t
\end{aligned}
$$

Outer Convexification

$$
\begin{aligned}
\min_{x,\omega}\ & x_2(32) \\
\text{s.t.}\ & \dot{x}_0(t) = x_1(t) && \forall t \\
& \dot{x}_1(t) = -2\omega_1(t) + \omega_2(t) && \forall t \\
& \dot{x}_2(t) = 4\omega_1(t) + \omega_2(t) && \forall t \\
& x(0) = (0,0,0) \\
& x_0(32) = 300 \\
& \omega(t) \in \{0, 1\}^2 && \forall t \\
& \omega_1(t) + \omega_2(t) = 1 && \forall t
\end{aligned}
$$

... can eliminate $\omega_1(t)$ from SOS-1 constraint

Outer Convexification Example



Top: relaxed solution and bottom outer convexification

... but can we construct an integer control $\omega_2(t) \in \{0, 1\}$?

Constructing the Integer Control (Sum-Up Rounding)

Theorem (Sum-Up Rounding)

Given a relaxed optimal solution $\alpha^*(t)$ *for* $t \in [0, T]$, *define* $\omega(t) := p_i \in \{0,1\}^{n_\Omega}$ *for* $t \in [t_i, t_{i+1})$, $0 \le i < N$, $1 \le j \le n_\Omega$ *by*

$$\hat{p}_{ij} := \int_0^{t_{i+1}} \alpha_j^*(t)\, dt - \sum_{k=0}^{i-1} p_{kj},$$

$$p_{ij} := \begin{cases} 1 \text{ if } (\forall k : \hat{p}_{ij} \ge \hat{p}_{ik}) \wedge (\forall k,\ \hat{p}_{ij} = \hat{p}_{ik} : j < k) \\ 0 \text{ otherwise.} \end{cases}$$

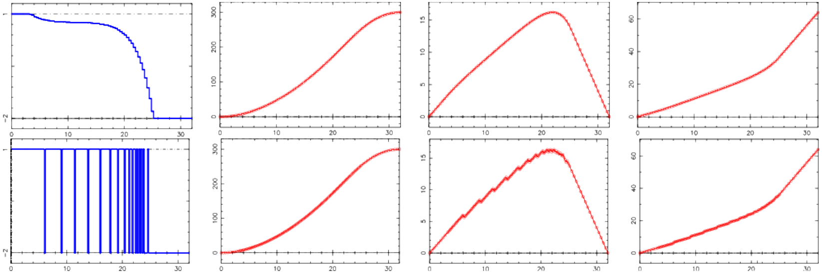
Then ω *satisfies*

$$\left\| \int_0^T \omega(t) - \alpha^*(t)\, dt \right\| \le c(n^v) \max_{0 \le i < N} \{t_{i+1} - t_i\} =: \varepsilon.$$

where we the best we can currently prove is $c(n^v) = n^v - 1$.

- $c(n^v)$ likely not tight. Conjecture: $c(n^v) \in O(\log n^v)$.

Rocket Car Example Continued



Convex relaxation and integer feasible sum-up rounding solution.

Outline

Solvers for Convex MINLP

Name	Algorithm(s)
α-ECP	Extended cutting-plane
BONMIN	BnB, LP/NLP-BnB, OA, Hybrid
DICOPT	OA
FilMINT	LP/NLP-BnB
KNITRO	BnB, LP/NLP-BnB
MILANO	BnB, OA
MINLPBB	BnB
MINOPT	OA
MINOTAUR	BnB, QP-Diving
SBB	BnB

Open-source and commercial solvers

Solvers for Nonconvex MINLP

Name	Algorithm(s)
α-BB	α-BB
BARON	LP-BB
COCONUT	LP-BB
COUENNE	LP-BB
GloMIQO	LP-BB
LGO	Sampling, Heursitics
LindoGlobal	LP-BB
SCIP	LP-BB

Open-source and commercial solvers

Optimization Modeling Languages

Advantages of DSL's

- Express optimization languages in natural algebraic form
- Access to automatic differentiation & presolve techniques

Modeling Languages for Optimization

- AIMMS [Bisschop and Entriken, 1993]
- AMPL [Fourer et al., 1993]
- GAMS [Brooke et al., 1992]
- MOSEL [Colombani and Heipcke, 2002]
- TomLab [Holmström and Edvall, 2004]

... TomLab is built on top of Matlab

Online Resources

MINLP Test Problem Libraries

- wiki.mcs.anl.gov/leyffer/index.php/MacMINLP
- minlp.org/ ... models & descriptions
- www.gamsworld.org/minlp/

Good source of modeling tricks and practices!

Online Solvers

- www.neos-server.org/neos/

Computational Infrastructure for Operations Research

- www.coin-or.org

Solvers & Resources for MIOCPs

Solvers fort MIOCPs

- MUSCOD-II `www.neos-server.org/neos/solvers/miocp:MUSCOD-II/AMPL.html`
- MINOPT [Schweiger, 1999]

Online Resources for MIOCPs

- MINTOC `mintoc.de` models & solvers

Outline

Some Open Research Question in MINLP

A very personal view of open research in MINLP

- Structure-exploiting algorithm ... nonlinear structure
- Models and methods beyond traditional MINLP
- Implementation & software design issues

Other important areas

- Modeling with integers and nonlinear functions
- Robust and fast nonlinear solvers ... hot-starts

... discuss some of these items in more detail

Exploiting Structure in MINLP

Moving beyond factorable functions

- Group partial separability

$$f(x) = \sum_{j=1}^{q} g_j \left(a_j^T x + b_j + \sum_{i \in \mathcal{E}_j} f_i(x_{[i]}) \right)$$

  - Better ways to linearizing MINLPs ... undercover heuristic
  - Alternative to factorable functions ... larger "atoms"

- Looking at more macroscopic structure, not just $x_i x_j$-terms
- Explore alternative formulations in computational graph

Models and Methods Beyond Traditional MINLP

Models and methods for traditional MINLPs

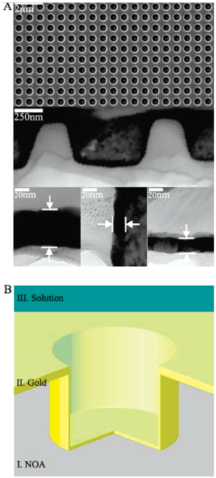
- Models described in algebraic form
- Finite-dimensional problems (variables and constraints)

Increasing interest in moving beyond traditional MINLPs

- MIPs with black-box functions or simulations
- MIPs with partial-differential equation (PDE) constraints
- Bilevel MINLP; e.g. leader-follower games
- Stochastic MINLP and optimization under uncertainty

... motivated by complex applications

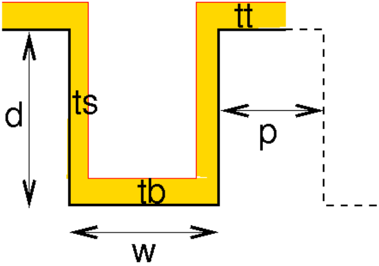
Example: Design of Nano-Photonic Devices



- nano-structures for chemical sensing
  optical response at certain wave-lengths
- Top: scanning electron micrograph
  Middle: cross section of crystal
  Bottom: gold thickness
- 3D FEM analysis simulation; no gradients
  periodicity, gold thickness, depth, & width of nano-wells
- derivative-free optimization
  objective function evaluation: 12 hrs
  ... on 125 nodes of an Apple G5 X-server
- optimization of a $\boxed{\text{black-box}}$
  ... simulation-based optimization
  ... derivative-free optimization

Inside the Black Box

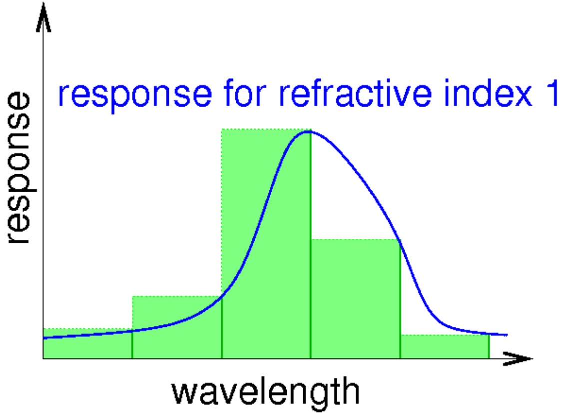
Design Parameters:



- d = depth of nano-well
- p = periodicity of design
- w = width/diameter of nano-well
- t = thickness (side/bottom/top) gold layer

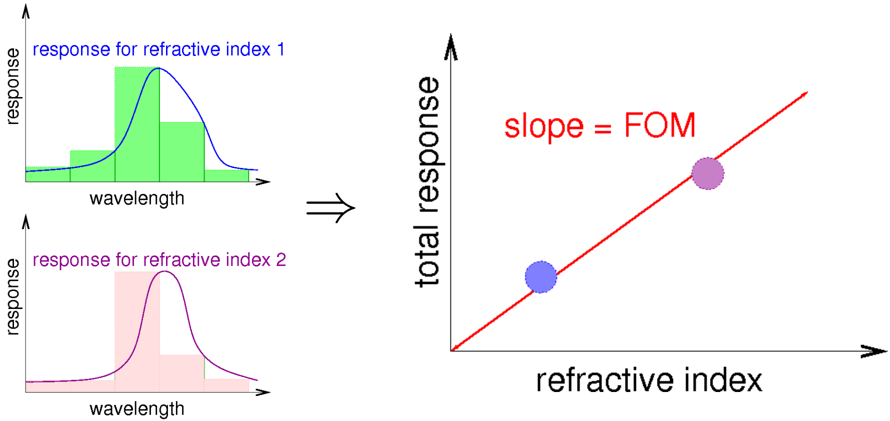
Inside the Black Box

Given values of design parameters (d, p, w, ts, tb, tt)
... perform FDFT simulation for refractive index



... get total response

Objective = Figure of Merit (FOM)



... combine responses for different refractive indices

... maximize slope (sensitivity) of design

The Initial Task: Proving Optimality

Goal:

$$\underset{d,p,w,t_s,t_b,t_t}{\text{maximize}} \quad f(d,p,w,t_s,t_b,t_t)$$

where f evaluation/simulation takes 12 hours on 125 nodes

Stephen: "We have evaluated the structure at 64 points. Can you prove that our design is optimal?"

Sven: "Sure, we just check $\nabla f(d,p,w,t_s,t_b,t_t) = 0$!"

Boyana: "They don't have gradients, you dim-witted optimizer!!!"

... Boyana should know: she is Dr. ADIC!

Our Grand Scheme to Prove Optimality

Given putative optimum, x_* and $x_k \in \mathbb{R}^6, k = 1, \ldots, 63$
... prove that $x_* = \operatorname{argmax} f(x)$ ... without using $\nabla f(x)$

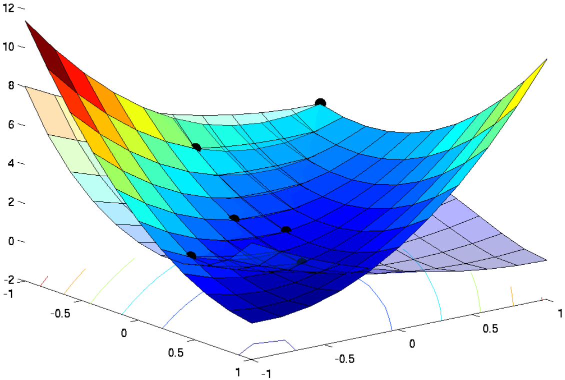
A simple idea:

- approximate $f(x) \simeq q(x)$ by a quadratic surrogate model
- **IF** $\nabla q(x_*) = 0$ **THEN** declare optimality

Building a quadratic surrogate model:

- quadratics are easy & all functions look like quadratics
- interpolate $f(x)$ at x_* and some $x_k, k = 1, \ldots, 63$
- get $q(x) = c + g^T x + \frac{1}{2} x^T H x$
- optimality: $\nabla q(x) = 0 \;\Leftrightarrow\; Hx = -g$ & H neg. definite
  ... solve a 6×6 linear system (easy)

Quadratic Interpolation in 2D at 6 Points



Building a Quadratic Surrogate Model

Given $x_k \in \mathbb{R}^6$ and $f(x_k)$ for $k = 1, \ldots, 64$
... approximate $f(x) \simeq q(x) = c + g^T x + \frac{1}{2} x^T H x$

Quadratic surrogate, $q(x) = c + g^T x + \frac{1}{2} x^T H x$
... has $m = (6+1)(6+2)/2 = 28$ parameters (c, g, H)

Interpolation conditions ($q(x)$ in $\mathbb{R}^n$ has $m = (n+1)(n+2)/2$ DOFs)

$$(L) \quad \begin{bmatrix} 1 & x_{11} \ldots x_{1n} & x_{11}^2 \ldots x_{1n}^2 \\ 1 & x_{21} \ldots x_{2n} & x_{21}^2 \ldots x_{2n}^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{m1} \ldots x_{mn} & x_{m1}^2 \ldots x_{mn}^2 \end{bmatrix} \begin{pmatrix} c \\ g \\ \mathrm{vec}(H) \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_m) \end{pmatrix}$$

Theorem: Linear system (L) is nonsingular with probability one.

Building a Quadratic Surrogate Model

Given $x_k \in \mathbb{R}^6$ and $f(x_k)$ for $k = 1, \ldots, 64$
... approximate $f(x) \simeq q(x) = c + g^T x + \frac{1}{2} x^T H x$

Quadratic $q(x)$ in $\mathbb{R}^6$ has $m = (6+1)(6+2)/2 = 28$ DOFs

Snag: Too many (64) interpolation conditions

Solution 1: Choose 28 points closest to the putative solution x_*
$\Rightarrow$ failure: singular interpolation matrix!!!

Solution 2: Compute the rank of 64×28 full interpolation matrix
$\Rightarrow$ surprise: rank $= 21 < 28$???

Building a Quadratic Surrogate Model

Given $x_k \in \mathbb{R}^6$ and $f(x_k)$ for $k = 1, \ldots, 64$
... approximate $f(x) \simeq q(x) = c + g^T x + \frac{1}{2} x^T H x$

Snag: too many (64) dependent interpolation conditions

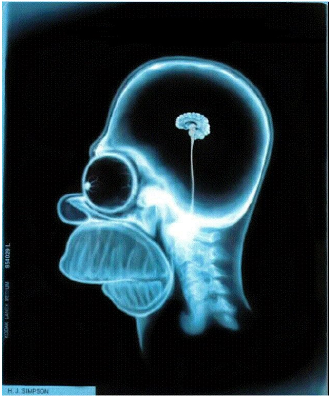
1. find 21 (=rank of (L)) independent rows of (L) closest to x_*
2. find quadratic with minimum curvature

$$\begin{array}{ll} \underset{c,g,H}{\text{minimize}} & \|H\|_F \\ \text{subject to} & \begin{bmatrix} 1 & x_{11} \ldots x_{1n} & x_{11}^2 \ldots x_{1n}^2 \\ 1 & x_{21} \ldots x_{2n} & x_{21}^2 \ldots x_{2n}^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{m1} \ldots x_{mn} & x_{m1}^2 \ldots x_{mn}^2 \end{bmatrix} \begin{pmatrix} c \\ g \\ \mathrm{vec}(H) \end{pmatrix} = \ldots \end{array}$$

that interpolates $m = 21$ linearly independent rows of (L)

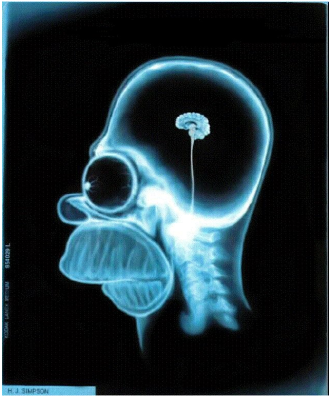
... of course $\nabla q(x_*) \neq 0$ and H has positive eigenvalues!

Trust-Region Methods

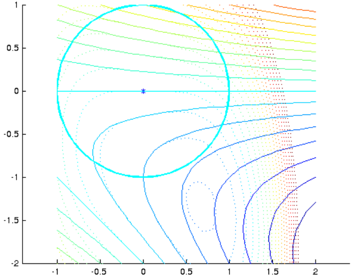


trust quadratic model $q(x)$
in small neighborhood of x_*

Trust-Region Methods



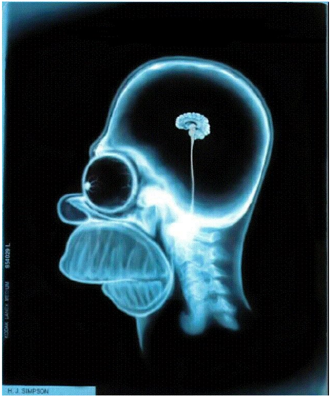
trust quadratic model $q(x)$
in small neighborhood of x_*



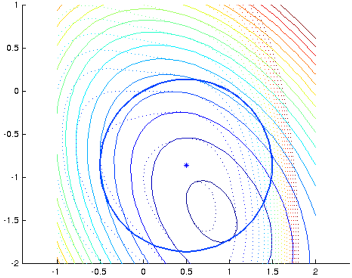
$$\max q(x) \text{ s.t.} \|x_* - x\| \le \rho$$

- handle indefinite H
- solve approximately
- convergence theory

Trust-Region Methods



trust quadratic model $q(x)$
in small neighborhood of x_*



$\max q(x)$ s.t. $\|x_* - x\| \leq \rho$

- handle indefinite H
- solve approximately
- convergence theory

Derivative-Free Optimization Approach

Aim: maximize a 6D function (sensitivity) $f(x)$ given 64 points
Idea: use (quadratic) interpolating models to find next point
Snag: interpolation conditions are linearly dependent (rank 21)

1. start at best solution found so far
2. sort remaining 63 points in order of distance
3. build up set of 21 linearly independent interpolation conditions
4. fit quadratic model $q(x) = q_k + g_k^T x + \frac{1}{2} x^T H_k x$ to data
   ... by solving QP (28 parameters, 7 degree of freedom)
   - minimize $\|H_{k-1} - H_k\|_F$ ... curvature change
   - subject to independent interpolation conditions
5. maximize $q(x)$ quadratic mode in trust-region $\|x - x_k\| \leq \rho_k$

Implementation: 3. in `matlab`; 4.&5. in `ampl`
Communication with Objective Function: `jmaria@uiuc.edu`

Optimization by Email [Paul Tseng, 2009]

Just over a day later ...

```
Date: Wed, 11 Jun 2008 22:22:33 -0500 (CDT)
From: Sven Leyffer <leyffer@mcs.anl.gov>
To: Joana Maria <jmaria@uiuc.edu>

Hi Joanna,

I built my first quadratic model from interpolation.
Here are two new points you could try:
x1 = (310,514,410,106,86,86)
x2 = (315,519,405,101,91,91)
```

... trust-region radius $\rho_1 = 5, \ \rho_2 = 10$

Optimization by Email

```
Date: Mon, 16 Jun 2008 10:18:49 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Dear Sven,

The calculated FOM for the x1 point is 7349.8 and for
point x2 is 15263.2.

I'm sorry it took so long but the cluster I use was
having some problems and they had to take it down to
fix it.
```

... she is still thinking that we are almost finished!

Optimization by Email

Just over an hour later (I needed some coffee)

```
Date: Mon, 16 Jun 2008 11:33:02 -0500 (CDT)
From: Sven Leyffer <leyffer@mcs.anl.gov>
To: Joana Maria <jmaria@uiuc.edu>

Hi Joana,

My next point is x3 = (325,  519, 405, 101, 91, 91)

The model is getting very ill-conditioned, but hopefully,
we'll be able to see more soon. At the very least, I hope
to be able to tell you that you are locally optimal.
```

... I am still thinking that we are almost finished!

Optimization by Email

Date: Tue, 17 Jun 2008 11:12:11 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Dear Sven,

Yes, this way the sensitivity improved:
The FOM for point x3 is 20040.7.

Woohoo 25% improvement;
... optimization is sooooo coooool!

In terms of fabricating the actual samples the model with 4 parameters (with the same gold thickness everywhere) is the one we are most likely to achieve though ...

... wait, we have only 4 optimization variables now???

Optimization by Email

Generating a new point x_4 ... we get

```
Date: Tue, 17 Jun 2008 20:37:01 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Dear Sven

I got a FOM of 19209.8 for the x4 point - a little worse
than point x3 but better than what I had originally.

I am not sure if I can trust point 3 completely; most of the
sensitivity comes from one sharp peak in the transmission
spectra so I am repeating the same simulation but for more
points to see if I get the same.
```

... but you can't just change the objective values!!!

Optimization by Email

Oh yes, we can ...

```
Date: Wed, 18 Jun 2008 10:25:33 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Dear Sven,

I repeated the simulation for the x3 point and I did
get a different figure of merit. This time I obtained
an FOM = 16018.2
```

... so $f(x_3) = 16018.2 < 20040.7$ got worse (nearly 25% error)

Optimization by Email

Still we keep on going ... a few iterations later

```
Date: Mon, 23 Jun 2008 10:01:00 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Hi Sven,

I got point x7. I'm sorry it took so long but the
cluster I use keeps having problems after an update
they did to the system :-(

For point x7 I got an FOM of 18859.7.
```

Our novel cut'n paste optimization techniques creates and solves a new model in about two minutes, including sending a reply email ...

Optimization by Email

A few iterations later ...

```
Date: Mon, 23 Jun 2008 12:34:02 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Hi Sven,

This point has a gold thickness at the bottom of
the nano-well that is ticker than the one at the top.
Experimentally we can’t do this sample :-(
```

You just added a constraint half way through the iteration ...

Optimization by Email

Switching to minimizing over four variables only ...

```
Date: Thu, 3 Jul 2008 11:13:43 -0500 (CDT)
From: Sven Leyffer <leyffer@mcs.anl.gov>
To: Joana Maria <jmaria@uiuc.edu>

Hi Joana,
Here is my new point: x001 = (340, 544, 409.303, 97.7334).
```

... now working through the night ...

```
Date: Fri, 4 Jul 2008 09:52:09 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Hi Sven,

Got sensitivity of 12048.6 for x001=(340, 544, 410, 98).
```

Wait, this problem is really a MIP???

Optimization by Email

Not a problem, we can do DFO-MINLP ...

```
Date: Sat, 12 Jul 2008 19:18:27 -0500 (CDT)
From: Sven Leyffer <leyffer@mcs.anl.gov>
To: Joana Maria <jmaria@uiuc.edu>

Hi Joana,

Interesting. The Hessian matrix is almost negative definite.
Its eigenvalues are: [-107.7409, -65.5983, -21.2956, 3.5465]

... anyway, you'll be more interested in the new point:
x008 = (330, 530,  419,  100)

This is NOT the same as the rounding of the continuous solution
(330, 529.3, 418.7, 100.2), which justifies the extra 2 seconds
CPU time spent solving the discrete problem ...
```

Optimization by Email

Joana is getting impatient after only a month ...

```
Date: Thu, 17 Jul 2008 10:34:39 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Hi Sven,

I got a FOM of 14419.4 for point x010.

Let me know what the next point should be :-)

Also, do you have an estimate of how many more points
we need to calculate?
```

I have no idea ... and the Hessian is still indefinite!!!

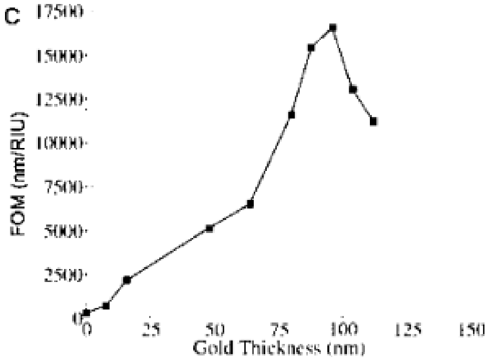
Optimization by Email

Many iterations (and a conference trip) later ...

```
Date: Mon, 21 Jul 2008 11:50:52 -0500
From: Joana Maria <jmaria@uiuc.edu>
To: Sven Leyffer <leyffer@mcs.anl.gov>

Hi, Sven

I already have a defense date: August 19th!! I guess we can
keep trying more points until the end of this week at which
point I'll need to stop and include the result we got into
one of my chapters... Hopefully we will get a stable system
before then :-)

I got a FOM of 13392.8 for point x012.
```

... we now have a practical stopping criterion (end of the week)

Results



- improved initial design by order of magnitude
- optimization gave another 25% improvement
- did not show that solution is optimal ...

... but I now have a paper in J. Phys. Chem. C!

Models and Methods Beyond Traditional MINLP

Models and methods for traditional MINLPs

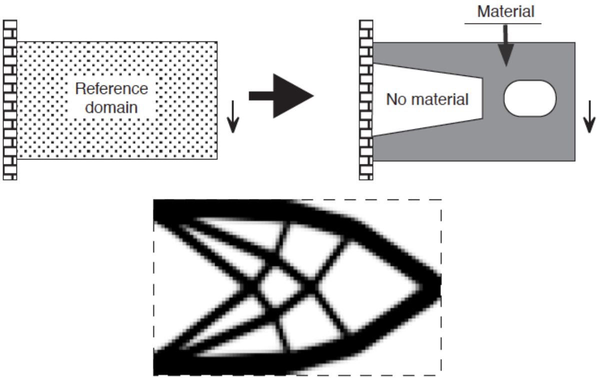
- Models described in algebraic form
- Finite-dimensional problems (variables and constraints)

Increasing interest in moving beyond traditional MINLPs

- MIPs with black-box functions or simulations
- MIPs with partial-differential equation (PDE) constraints
- Bilevel MINLP; e.g. leader-follower games
- Stochastic MINLP and optimization under uncertainty

... motivated by complex applications

Mixed-Integer PDE Constrained Optimization



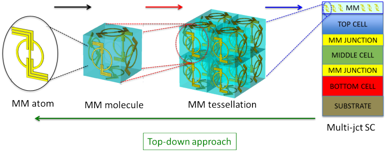
Topology Optimization [Bendsoe and Sigmund, 1999]

Topology optimization

- Minimization of compliance for statically loaded structures
- `www.mcs.anl.gov/~leyffer/MacMINLP/problems/top.mod`

Mixed-Integer PDE Constrained Optimization

Design of non-reciprocal optical metamaterial for solar cells



$$\nabla \times \mathbf{H} = -i\omega(\chi \mathbf{H} + \epsilon \mathbf{E}) + \mathbf{J}_e,$$
$$\nabla \times \mathbf{E} = i\omega(\mu \mathbf{H} + \zeta \mathbf{E}) + \mathbf{J}_m,$$

- Maxwell's equation gives **E** and **H** electric and magnetic field
- Objective is to maximize power inside solar cell

$$\frac{1}{2}\int_\omega I_{\text{solar}}(\omega)\int_V \Im(\epsilon(x,z))|\mathbf{E}(x,z;\omega)|^2 + \Im(\mu(x,z))|\mathbf{H}(x,z;\omega)|^2 \, dV \, d\omega$$

- $z_{i,j,k} = 1$ if orientation i chosen on face j of molecule k
- $z_{i,j,k}$ impact permittivities and permeabilities in Maxwell's

$$\widetilde{\epsilon_{j,k}} = \sum_{i \in \mathcal{O}} z_{i,j,k}\epsilon_i$$

Bilevel MINLPs

Leader-follower games with discrete decisions:

$$\begin{cases} \underset{x,y}{\text{minimize}} & f(x,y) \\ \text{subject to} & x \in \mathbb{B}^n \cap X \\ & y \in \begin{cases} \underset{v}{\text{argmin}} & g(x,v) \\ \text{subject to} & v \in \mathbb{B}^m \cap Y, \end{cases} \end{cases}$$

where $\mathbb{B} = \{0, 1\}$

Stackelberg game:

- Leader's variables are x
- Follower's variables are y

Bilevel MINLPs

Challenges of bilevel MINLP

- Cannot apply usual approach based on optimality
  1. Write down first-order conditions of follower
  2. Add FO conditions as constraints to leader

  $\Rightarrow$ Mathematical Program with Equilibrium Constraints
- Relaxing follower's integrality restricts leader
- How do we construct suitable relaxation and tightening?

... many applications would benefit from good methods!

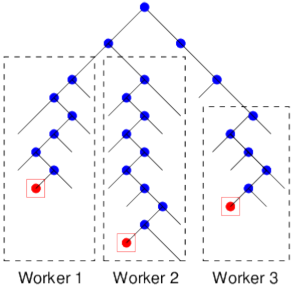
See PhD thesis by DeNegre (2011).

Implementation and Software Issues

Parallel BnB (scales to 1k cores)

First Strategy: 1 worker $\equiv$ 1 NLP
$\Rightarrow$ grain-size *too small*
... NLPs solve in seconds

Better Strategy:
1 worker $\equiv$ 1 subtree (MINLP)
... "streamers" running down tree



The Exascale Revolution [John Shalf, LBNL]

Systems	2009	2015 +1/-0	2018 +1/-0
System peak	**2 Peta**	**100-300 Peta**	**1 Exa**
Power	**6 MW**	**~15 MW**	**~20 MW**
System memory	0.3 PB	5 PB	64 PB (+)
Node performance	125 GF	0.5 TF or 7 TF	1-2 or 10TF
Node memory BW	25 GB/s	1-2TB/s	2-4TB/s
Node concurrency	12	O(100)	O(1k) or 10k
Total Node Interconnect BW	3.5 GB/s	100-200 GB/s 10:1 vs memory bandwidth 2:1 alternative	200-400GB/s (1:4 or 1:8 from memory BW)
System size (nodes)	18,700	50,000 or 500,000	O(100,000) or O(1M)
Total concurrency	225,000	O(100,000,000) *O(10)-O(50) to hide latency	O(billion) * O(10) to O(100) for latency hiding
Storage	15 PB	150 PB	500-1000 PB (>10x system memory is min)
IO	0.2 TB	10 TB/s	60 TB/s (how long to drain the machine)
MTTI	days	O(1day)	O(1 day)

Slide 51

Implementation and Software Issues

Opportunities and challenges of exascale systems

- Exploit massive concurrency: Billion-way parallelism
- Data movement are prohibitive in terms of energy
- Systems fail due to cosmic rays
  $\Rightarrow$ develop algorithms that are inherently resilient

Using exascale systems for MINLP?

- BnB won't scale to a billion-way concurrency
- Look at MIPDECO or other models for additional concurrency
  $\Rightarrow$ multiplicative effect?
- Distributed tree-search alone is not enough

Outline

Summary and Key Points

Key Points

- Heuristics for MINLP
  - Get solutions quickly (hard or real-time problems)
  - Help with standard techniques (better bounds)
- Mixed-integer optimal control ... cool relaxation properties
- Research Opportunities beyond traditional MINLP
  - Exploiting nonlinear structure
  - `MIP++`: PDEs, uncertainty, DFO, ...
  - Computer science challenges & opportunities

Final Exam for Course Credit: Have a beer with Sven today!

Thanks to my Collaborators ...





Pietro Belotti, Ashutosh Mahajan, Christian Kirches,
Jeff Linderoth, and Jim Luedtke

... and thanks to SOCN and Its Great Students!

Bisschop, J. and Entriken, R. (1993).
AIMMS The Modeling System.
Paragon Decision Technology.

Bonami, P., Cornuéjols, G., Lodi, A., and Margot, F. (2009).
A feasibility pump for mixed integer nonlinear programs.
Mathematical Programming, 119:331–352.

Bonami, P. and Gonçalves, J. P. M. (2012).
Heuristics for convex mixed integer nonlinear programs.
Computational Optimization and Applications, 51:729–747.

Brooke, A., Kendrick, D., Meeraus, A., and Raman, R. (1992).
GAMS, A User's Guide.
GAMS Development Corporation.

Colombani, Y. and Heipcke, S. (2002).
Mosel: An extensible environment for modeling and programming solutions.
In Jussien, N. and Laburthe, F., editors, *Proceedings of the Fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems (CP-AI-OR'02)*, pages 277–290.

Danna, E., Rothberg, E., and LePape, C. (2005).
Exploring relaxation induced neighborhoods to improve MIP solutions.
Mathematical Programming, 102:71–90.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996).
The ant system: optimization by a colony of cooperating agents.
IEEE Transactions on Systems, Man and Cybernetics - Part B, 26(1):1–13.

Fischetti, M., Glover, F., and Lodi, A. (2005).
The feasibility pump.
Mathematical Programming, 104:91–104.

Fourer, R., Gay, D. M., and Kernighan, B. W. (1993).
AMPL: A Modeling Language for Mathematical Programming.
The Scientific Press.

Glover, F. (1989).
Tabu search: part I.
ORSA Journal on Computing, 1(3):190–206.

Goldberg, D. E. (1989).
Genetic algorithms in search, optimization, and machine learning.
Addison-Wesley, Boston.

Holmström, K. and Edvall, M. (2004).
The tomlab optimization environment.
In Kallrath, J., editor, *Modeling languages in mathematical optimization*, pages 369–378. Kluwer Academic Publishers, Boston, MA.
`http://tomopt.com/tomlab/`.

Kennedy, J. and Eberhart, R. (1995).
Particle swarm optimization.
In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983).
Optimization by simulated annealing.

Science, 220(4598):671–680.

Nannicini, G. and Belotti, P. (2012).
Rounding-based heuristics for nonconvex MINLPs.
Mathematical Programming Computation, 4:1–31.

Schweiger, C. A. (1999).
Process Synthesis, Design, and Control: Optimization with Dynamic Models and Discrete Decisions.
PhD thesis, Princeton University, Princeton, NJ.