MOPEC: multiple optimization problems with equilibrium constraints

Michael C. Ferris

Joint work with: Michael Bussieck, Steven Dirkse, Jan Jagla and Alexander Meeraus

Supported partly by AFOSR, DOE and NSF

University of Wisconsin, Madison

MOPTA, Lehigh University
August 18, 2011

Optimization beyond our discipline

- So you can solve an LP, MIP, NLP, SOCP, SDP, MCP, SP
  - N times faster
  - M times larger
  - or with K times better optimality guarantee

  than at MOPTA 2010
- Why aren't you using my *********** algorithm?
  (Michael Ferris, Boulder, CO, 1994)

Optimization beyond our discipline

- So you can solve an LP, MIP, NLP, SOCP, SDP, MCP, SP
  - N times faster
  - M times larger
  - or with K times better optimality guarantee

  than at MOPTA 2010
- Why aren't you using my *********** algorithm?
  (Michael Ferris, Boulder, CO, 1994)
- Should I care?
- Does it make a difference (how large is K, M or N)?

Optimization beyond our discipline

- So you can solve an LP, MIP, NLP, SOCP, SDP, MCP, SP
  - N times faster
  - M times larger
  - or with K times better optimality guarantee

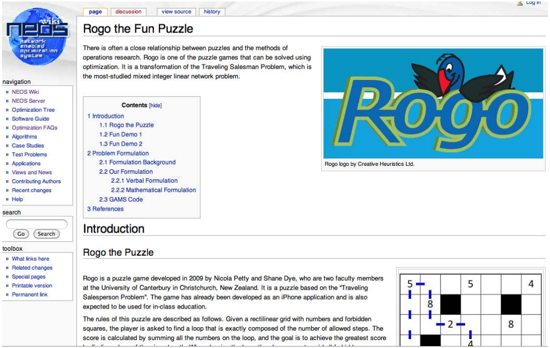
  than at MOPTA 2010
- Why aren't you using my *********** algorithm?
  (Michael Ferris, Boulder, CO, 1994)
- Should I care?
- Does it make a difference (how large is K, M or N)?
- Just solving a single problem isn't the real value of optimization
  - (Ceria): optimization finds "holes" in the model
- Optimization is part of a larger process
  - (Stein): use dual to allow online solution of primal
  - (Zhu/Chen): solve SDP relaxations

I: Show me on a problem like mine

- Repeated solutions of multiple (different) problems enables “understanding” of the solution space (or sensitivity)
- NEOS wiki (www.neos-guide.org) or try out NEOS solvers (www.neos-solvers.org) for extensive examples

Building a class of case studies:

- JAVA api to NEOS
- Web description of problem
- Solution on NEOS
- Ability to modify and resolve
- Comparison of results



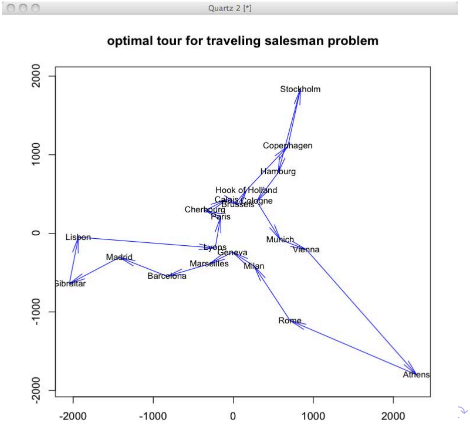
- Needs more examples from you!

II: Make it work in/enhance my environment

- In practice: need (large scale) data, problem/model transformations, access to solution features
- Modeling systems (AIMMS, AMPL, ... , GAMS, ...) provide some of these needs from an optimization perspective
- Open source, libraries, interfaces to Excel/Matlab/R

Traveling salesman in R:

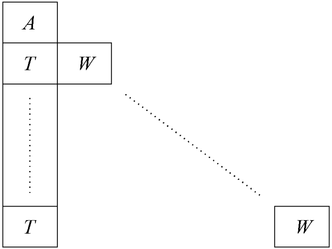
- ...
- wgdx (fnData, data)
- gams("tspDSE.gms")
- stat ← list(name='modelstat')
- v ← rgdx(fnSol, stat)
- ...
- R commands for graphics output



III: Allow new features in natural/reliable manner

- Two stage stochastic programming, x is here-and-now decision, recourse decisions y depend on realization of a random variable
- $\mathbb{R}$ is a risk measure (e.g. expectation, CVaR)

$$
\begin{aligned}
\text{SP:}\ \min \quad & c^\top x + \mathbb{R}[d^\top y] \\
\text{s.t.} \quad & Ax = b, \quad x \geq 0, \\
\forall \omega \in \Omega : \quad & T(\omega)x + W(\omega)y(\omega) \geq h(\omega), \\
& y(\omega) \geq 0.
\end{aligned}
$$



Models with explicit random variables

- Model transformation:
  - Write a core model as if the random variables are constants
  - Identify the random variables and decision variables and their staging
  - Specify the distributions of the random variables
- Solver configuration:
  - Specify the manner of sampling from the distributions
  - Determine which algorithm (and parameter settings) to use
- Output handling:
  - Optionally, list the variables for which we want a scenario-by-scenario report

Example: Farm Model (core model)

- Allocate land (L) for planting crops x to max (p/wise lin) profit
- Yield rate per crop c is $F{*}Y(c)$
- Can purchase extra crops b and sell s, but must have enough crops d to feed cattle

$$
\begin{array}{ll}
\max_{x,b,s\geq 0} & \mathit{profit} = p(x, b, s) \\
\text{s.t.} & \sum_c x(c) \leq L, \\
& F{*}Y(c) * x(c) + b(c) - s(c) \geq d(c)
\end{array}
$$

- Random variables are F, realized at stage 2: structured $T(\omega)$
- Variables x stage 1, b and s stage 2.
- landuse constraints in stage 1, requirements in stage 2.

Can now generate the *deterministic equivalent* problem or pass on directly to specialized solver

Stochastic Programming as an EMP

Three separate pieces of information (extended mathematical program) needed

1. emp.info: model transformation

```
randvar F 2 discrete 0.25 0.8 // below
                     0.50 1.0 // avg
                     0.25 1.2 // above

stage 2 b s req
```

2. solver.opt: solver configuration (benders, sampling strategy, etc)

```
4 "ISTRAT"  * solve universe problem (DECIS/Benders)
```

3. dictionary: output handling (where to put all the "scenario solutions")

How does this help?

- Clarity/simplicity of model
- Separates solution process from model description
- Models can be solved by deterministic equivalent, exisiting codes such as LINDO and DECIS, or decomposition approaches such as Benders, ATR, etc
- Allows description of compositional (nonlinear) random effects in generating ω

$$\text{i.e. } \omega = \omega_1 \times \omega_2, \;\; T(\omega) = f(X(\omega_1), Y(\omega_2))$$

- Easy to write down multi-stage problems
- Automatically generates "COR", "TIM" and "STO" files for Stochastic MPS (SMPS) input

Other EMP information

- emp.info: model transformation

```
expected_value EV_r r
cvarlo         CVaR_r r alpha
stage          2 r defr
jrandvar       v("att") v("gmc") v("usx") 2 discrete
   table of probabilities and outcomes
```

- Variables are assigned to $\mathbb{E}(r)$ and $\underline{CVaR}_\alpha(r)$; can be used in model (appropriately) for objective, constraints, or be bounded
- Problem transformation: theory states this expression can be written as convex optimization using:

$$\underline{CVaR}_\alpha(r) = \max_{a \in \mathbb{R}} \left\{ a - \frac{1}{\alpha} \sum_{j=1}^{N} Prob_j * (a - r_j)_+ \right\}$$

Example: Clear Lake Model (core model)

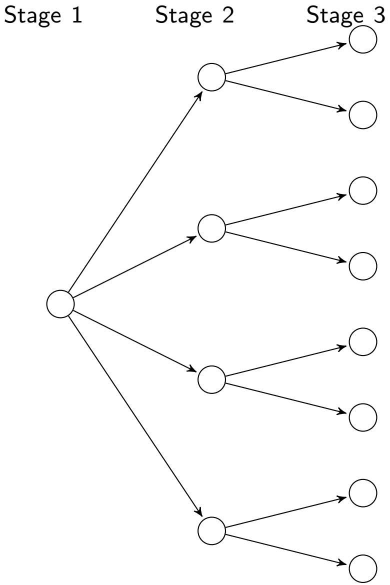
- Water levels $l(t)$ in dam for each month t
- Determine what to release normally $r(t)$, what then floods $f(t)$ and what to import $z(t)$
- minimize cost of flooding and import
- Change in reservoir level in period t is $\delta(t)$

$$\begin{aligned} &\max cost = c(f, z) \\ &\text{s.t. } l(t) = l(t-1) + \delta(t) + z(t) - r(t) - f(t) \end{aligned}$$

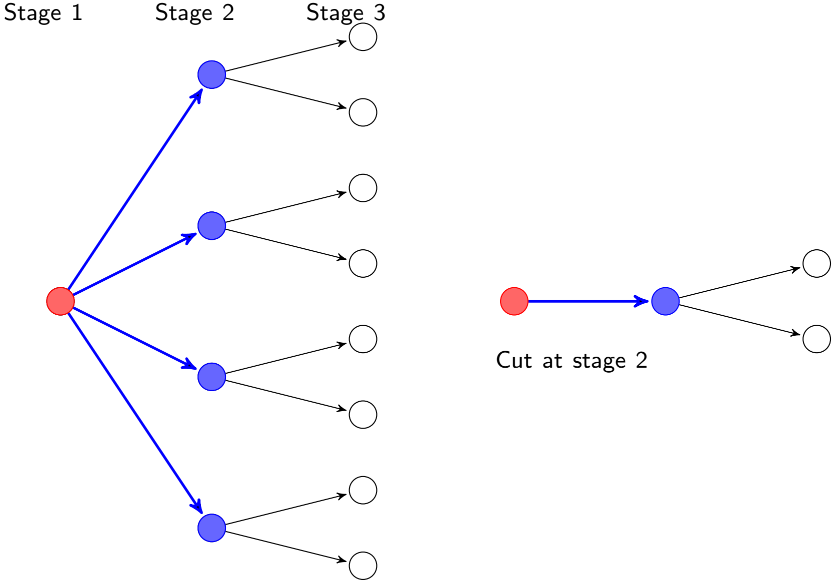
- Random variables are δ, realized at stage t, $t \geq 2$.
- Variables l, r, f, z in stage t, $t \geq 2$.
- balance constraint at t in stage t.

Example of a multi-stage stochastic program.

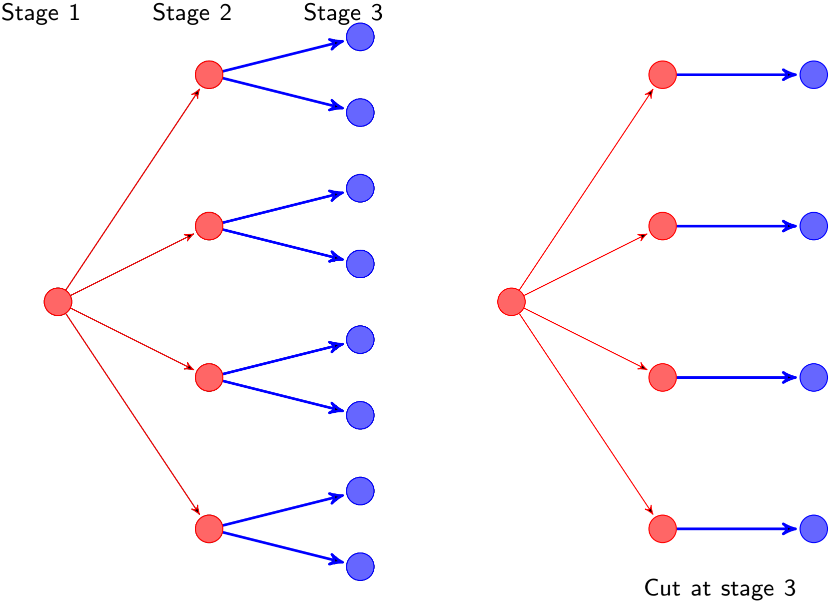
Multi to 2 stage reformulation



Multi to 2 stage reformulation



Multi to 2 stage reformulation



Solution options

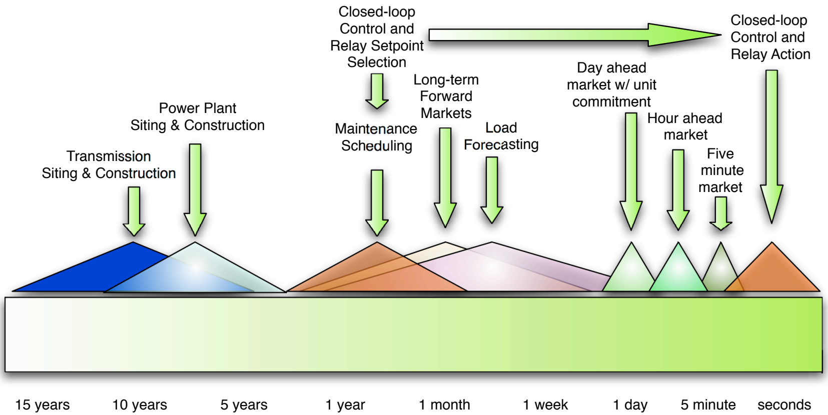
- Form the deterministic equivalent
- Solve using LINDO api (stochastic solver)
- Convert to two stage problem and solve using DECIS or any number of competing methods
- Problem with $3^{40} \approx 1.2 * 10^{19}$ realizations in stage 2
  - DECIS using Benders and Importance Sampling: < 1 second (and provides confidence bounds)
  - CPLEX on a presampled deterministic equivalent:

sample	samp. time(s)	CPLEX time(s) for solution			cols (mil)
500	0.0	5	(4.5 barrier,	0.5 xover)	0.25
1000	0.2	18	(16 barrier,	2 xover)	0.5
10000	28	195	(44 barrier,	151 xover)	5
20000	110	1063	(98 barrier,	965 xover)	10

IV: Build coupled/structured models that engage the decision maker

- The next generation electric grid will be more dynamic, flexible, constrained, and more complicated.
- Decision processes (in this environment) are predominantly hierarchical.
- Models to support such decision processes must also be layered or hierachical.
- Optimization and computation facilitate adaptivity, control, treatment of uncertainties and understanding of interaction effects.
- Developing interfaces and exploiting hierarchical structure using computationally tractable algorithms will provide overall solution speed, understanding of localized effects, and value for the coupling of the system.

Representative decision-making timescales in electric power systems

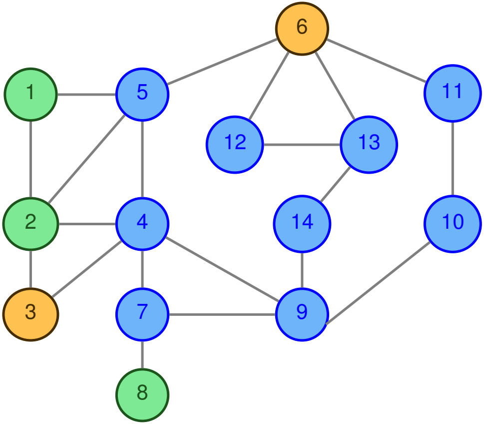


A monster model is difficult to validate, inflexible, prone to errors.

Example: Transmission Line Expansion Model (1)

$$\min_{x \in X} \sum_{\omega} \pi_\omega \sum_{i \in N} d_i^\omega p_i^\omega(x)$$

$$\text{s.t.} \quad Ax \leq b$$



N:	The set of nodes
X:	Line expansion set
x:	Amount of investment in given line
ω:	Demand scenarios
π_ω:	Scenario prob
d_i^ω:	Demand (load at i in scenario ω)
$p_i^\omega(x)$:	Price (LMP) at i in scenario ω as a function of x

Generator Expansion (2): $\forall f \in F$:

$$\min_{y_f} \sum_{\omega} \pi_\omega \sum_{j \in G_f} C_j(y_j, q_j^\omega) - r(h_f - \sum_{j \in G_f} y_j)$$

$$\text{s.t.} \sum_{j \in G_f} y_j \leq h_f, y_f \geq 0$$

G_f: Generators of firm $f \in F$
y_j: Investment in generator j
q_j^ω: Power generated at bus j in scenario ω
C_j: Cost function for generator j
r: Interest rate

Market Clearing Model (3): $\forall \omega$:

$$\min_{z,\theta,q^\omega} \sum_f \sum_{j \in G_f} C_j(y_j, q_j^\omega) \quad \text{s.t.}$$

$$q_j^\omega - d_j^\omega = \sum_{i \in I(j)} z_{ij} \qquad \forall j \in N(\perp p_j^\omega)$$

$$z_{ij} = \Omega_{ij}(\theta_i - \theta_j) \qquad \forall (i,j) \in A$$

$$-b_{ij}(x) \leq z_{ij} \leq b_{ij}(x) \qquad \forall (i,j) \in A$$

$$\underline{u}_j(y_j) \leq q_j^\omega \leq \overline{u}_j(y_j)$$

z_{ij}: Real power flowing along line ij
q_j^ω: Real power generated at bus j in scenario ω
θ_i: Voltage phase angle at bus i
Ω_{ij}: Susceptance of line ij
$b_{ij}(x)$: Line capacity as a function of x
$\underline{u}_j(y)$, $\overline{u}_j(y)$: Generator j limits as a function of y

How to combine: Nash Equilibria

- Non-cooperative game: collection of players $a \in \mathcal{A}$ whose individual objectives depend not only on the selection of their own strategy $x_a \in C_a = \mathrm{dom} f_a(\cdot, x_{-a})$ but also on the strategies selected by the other players $x_{-a} = \{x_a : o \in \mathcal{A} \setminus \{a\}\}$.
- Nash Equilibrium Point:

$$\bar{x}_{\mathcal{A}} = (\bar{x}_a, a \in \mathcal{A}) : \forall a \in \mathcal{A} : \bar{x}_a \in \mathrm{argmin}_{x_a \in C_a} f_a(x_a, \bar{x}_{-a}).$$

1. for all $x \in \mathcal{A}$, $f_a(\cdot, x_{-a})$ is convex
2. $C = \prod_{a \in \mathcal{A}} C_a$ and for all $a \in \mathcal{A}$, C_a is closed convex.

VI reformulation

Define

$$G : \mathbb{R}^N \mapsto \mathbb{R}^N \text{ by } G_a(x_\mathcal{A}) = \partial_a f_a(x_a, x_{-a}), a \in \mathcal{A}$$

where ∂_a denotes the subgradient with respect to x_a. Generally, the mapping G is set-valued.

Theorem

Suppose the objectives satisfy (1) and (2), then every solution of the variational inequality

$$x_\mathcal{A} \in C \text{ such that } -G(x_\mathcal{A}) \in N_C(x_\mathcal{A})$$

is a Nash equilibrium point for the game.
Moreover, if C is compact and G is continuous, then the variational inequality has at least one solution that is then also a Nash equilibrium point.

Solution approach (Tang)

- Use derivative free method for the upper level problem (1)
- Requires $p_i^\omega(x)$
- Construct these as multipliers on demand equation (per scenario) in an Economic Dispatch (market clearing) model
- But transmission line capacity expansion typically leads to generator expansion, which interacts directly with market clearing
- Interface blue and black models using Nash Equilibria (as EMP):

empinfo: equilibrium
forall f: min expcost(f) y(f) budget(f)
forall ω: min scencost(ω) q(ω) ...

Flow of information

$$\min_{x \in X} \quad \sum_{\omega} \pi_\omega \sum_{i \in N} d_i^\omega p_i^\omega(x)$$

$$\text{s.t.} \quad \min_{y_f \in Y} \sum_{\omega} \pi_\omega \sum_{j \in G_f} C_j(y_j, q_j^\omega) - r(h_f - \sum_{j \in G_f} y_j) \qquad \forall f \in F$$

$$\min_{z,\theta,q^\omega} \sum_{f} \sum_{j \in G_f} C_j(y_j, q_j^\omega) \qquad \forall \omega$$

$$s.t. \quad q_j^\omega - d_j^\omega = \sum_{i \in I(j)} z_{ij} \qquad \forall j \in N (\perp p_j^\omega(x))$$

$$z_{ij} = \Omega_{ij}(\theta_i - \theta_j) \qquad \forall (i,j) \in A$$

$$-b_{ij}(x) \leq z_{ij} \leq b_{ij}(x) \qquad \forall (i,j) \in A$$

$$\underline{u}_j(y_j) \leq q_j^\omega \leq \overline{u}_j(y_j) \qquad \forall j \in N$$

Feasibility

$$\text{KKT of } \min_{y_f \in Y} \sum_{\omega} \pi_\omega \sum_{j \in G_f} C_j(y_j, q_j^\omega) - r(h_f - \sum_{j \in G_f} y_j) \quad \forall f \in F \quad (2)$$

$$\text{KKT of } \min_{(z,\theta,q^\omega) \in Z(x,y)} \sum_{f} \sum_{j \in G_f} C_j(y_j, q_j^\omega) \quad \forall \omega \quad (3)$$

- Models (2) and (3) form an MCP/VI (via EMP)
- Solve (3) as NLP using global solver (actual $C_j(y_j, q_j^\omega)$ are not convex), per scenario (SNLP) this provides starting point for MCP
- Solve (KKT(2) + KKT(3)) using EMP and PATH, then repeat
- Identifies MCP solution whose components solve the scenario NLP's (3) to global optimality

Scenario	ω_1	ω_2
Probability	0.5	0.5
Demand Multiplier	8	5.5

SNLP (1):

Scenario	q_1	q_2	q_3	q_6	q_8
ω_1	3.05	4.25	3.93	4.34	3.39
ω_2		4.41	4.07	4.55	

EMP (1):

Scenario	q_1	q_2	q_3	q_6	q_8
ω_1	2.86	4.60	4.00	4.12	3.38
ω_2		4.70	4.09	4.24	

Firm	y_1	y_2	y_3	y_6	y_8
f_1	167.83	565.31			266.86
f_2			292.11	207.89	

Scenario	ω_1	ω_2
Probability	0.5	0.5
Demand Multiplier	8	5.5

SNLP (2):

Scenario	q_1	q_2	q_3	q_6	q_8
ω_1	0.00	5.35	4.66	5.04	3.91
ω_2		4.70	4.09	4.24	

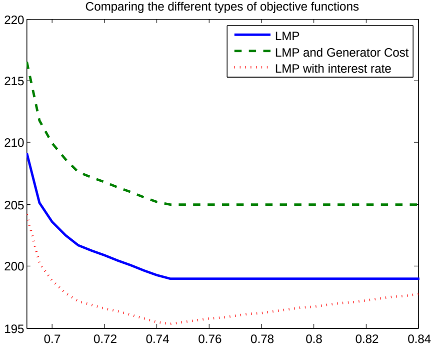
EMP (2):

Scenario	q_1	q_2	q_3	q_6	q_8
ω_1	0.00	5.34	4.62	5.01	3.99
ω_2		4.71	4.07	4.25	

Firm	y_1	y_2	y_3	y_6	y_8
f_1	0.00	622.02			377.98
f_2			283.22	216.79	

Observations

- But this is simply one function evaluation for the outer "transmission capacity expansion" problem
- Number of critical arcs typically very small
- But p_j^{ω} can be very volatile
- Outer problem is small scale, objectives are open to debate, possibly ill conditioned



- Economic dispatch should use AC power flow model
- Structure of market open to debate
- Types of "generator expansion" also subject to debate
- Suite of tools is very effective in such situations

What is EMP?

Annotates existing equations/variables/models for modeler to provide/define additional structure

- equilibrium
- vi (agents can solve min/max/vi)
- bilevel (reformulate as MPEC, or as SOCP)
- disjunction (or other constraint logic primitives)
- randvar
- dualvar (use multipliers from one agent as variables for another)
- extended nonlinear programs (library of plq functions)

Currently available within GAMS

Conclusions

- Modern optimization within applications requires multiple model formats, computational tools and sophisticated solvers
- EMP model type is clear and extensible, additional structure available to solver
- Extended Mathematical Programming available within the GAMS modeling system
- Able to pass additional (structure) information to solvers
- Embedded optimization models automatically reformulated for appropriate solution engine
- Exploit structure in solvers
- Extend application usage further
- Slides available at http://www.cs.wisc.edu/~ferris/talks