An Extended Mathematical Programming Framework

Michael C. Ferris
Steven Dirkse, Jan Jagla, Alex Meeraus

University of Wisconsin, Madison

Informs Computing Society Meeting, January 9, 2011

Extended Mathematical Programs

- Optimization models improve understanding of underlying systems and facilitate operational/strategic improvements under resource constraints
- Problem format is old/traditional

$$\min_{x} f(x) \text{ s.t. } g(x) \leq 0, h(x) = 0$$

- Extended Mathematical Programs allow annotations of constraint functions to augment this format.
- This talk will give several examples of how to use this modeling framework

But who cares?

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

But who cares?

- Why aren't you using my *********** algorithm?
  (Michael Ferris, Boulder, CO, 1994)
- Show me on a problem like mine
- Must run on defaults
- Must deal graciously with poorly specified cases
- Must be usable from my environment (Matlab, R, GAMS, ...)
- Must be able to model my problem easily

EMP provides annotations to an existing optimization model that convey new model structures to a solver

NEOS is soliciting case studies that show how to do the above, and will provide some tools to help

The PIES Model (Hogan)

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax = q(p) \\ & Bx = b \\ & x \geq 0 \end{aligned}$$

- Issue is that p is the multiplier on the dembal constraint of LP
- Can solve the problem by writing down the KKT conditions of this LP, forming an LCP and exposing p to the model
- EMP: dualvar p dembal

Example: Bimatrix Games

- Nash game: two players have I and J pure strategies.
- p and q (strategy probabilities) belong to unit simplex $\triangle_I$ and $\triangle_J$ respectively.
- Payoff matrices $A \in R^{J \times I}$ and $B \in R^{I \times J}$, where $A_{j,i}$ is the profit received by the first player if strategy i is selected by the first player and j by the second, etc.
- The expected profit for the first and the second players are $q^T Ap$ and $p^T Bq$ respectively.
- A Nash equilibrium is reached by the pair of strategies (p^*, q^*) if and only if

$$p^* \in \arg \min_{p \in \triangle_I} \langle Aq^*, p \rangle \quad \text{and} \quad q^* \in \arg \min_{q \in \triangle_J} \langle B^T p^*, q \rangle$$

- EMP: facilitates modeling of Nash Equilibria

Complementarity Problems in Economics (MCP)

- p represents prices, x represents activity levels
- System model: given prices, (agent) i determines activities x_i

$$G_i(x_i, x_{-i}, p) = 0$$

  x_{-i} are the decisions of other agents.
- Walras Law: market clearing

$$0 \leq S(x, p) - D(x, p) \perp p \geq 0$$

- Key difference: optimization assumes you control the complete system
- Complementarity determines what activities run, and who produces what

Nash Equilibria

- Nash Games: x^* is a Nash Equilibrium if

$$x_i^* \in \arg\min_{x_i \in X_i} \ell_i(x_i, x_{-i}^*, q), \forall i \in \mathcal{I}$$

  x_{-i} are the decisions of other players.
- Quantities q given exogenously, or via complementarity:

$$0 \leq H(x, q) \perp q \geq 0$$

- empinfo: equilibrium
  min loss(i) x(i) cons(i)
  vifunc H q
- Applications: Discrete-Time Finite-State Stochastic Games. Specifically, the Ericson & Pakes (1995) model of dynamic competition in an oligopolistic industry.

General Equilibrium models

$$(C): \max_{x_k \in X_k} U_k(x_k) \text{ s.t. } p^T x_k \leq i_k(y,p)$$

$$(I): i_k(y,p) = p^T \omega_k + \sum_j \alpha_{kj} p^T g_j(y_j)$$

$$(P): \max_{y_j \in Y_j} p^T g_j(y_j)$$

$$(M): \max_{p \geq 0} p^T \left(\sum_k x_k - \sum_k \omega_k - \sum_j g_j(y_j) \right) \text{ s.t. } \sum_l p_l = 1$$

General Equilibrium models

$$(C) : \max_{x_k \in X_k} U_k(x_k) \text{ s.t. } p^T x_k \leq i_k(y, p)$$

$$(I) : i_k(y, p) = p^T \omega_k + \sum_j \alpha_{kj} p^T g_j(y_j)$$

$$(P) : \max_{y_j \in Y_j} p^T g_j(y_j)$$

$$(M) : \max_{p \geq 0} p^T \left(\sum_k x_k - \sum_k \omega_k - \sum_j g_j(y_j) \right) \text{ s.t. } \sum_l p_l = 1$$

Can reformulate as embedded problem (Ermoliev et al):

$$\max_{x \in X, y \in Y} \quad \sum_k \frac{t_k}{\beta_k} \log U_k(x_k)$$

$$\text{s.t. } \sum_k x_k \leq \sum_k \omega_k + \sum_j g_j(y_j)$$

$t_k = i_k(y, p)$ where p is multiplier on NLP constraint

Sequential Joint Maximization

$$\max_{x \in X, y \in Y} \quad \sum_k \frac{t_k}{\beta_k} \log U_k(x_k)$$
$$\text{s.t.} \quad \sum_k x_k \leq \sum_k \omega_k + \sum_j g_j(y_j)$$

$t_k = i_k(y, p)$ where p is multiplier on NLP constraint

- Embedded model often solves faster as an MCP than the original MCP from Nash game
- Can exploit structure to improve computational performance further

Sequential Joint Maximization

$$\max_{x \in X, y \in Y} \quad \sum_k \frac{t_k}{\beta_k} \log U_k(x_k)$$
$$\text{s.t.} \quad \sum_k x_k \leq \sum_k \omega_k + \sum_j g_j(y_j)$$

$t_k = i_k(y, p)$ where p is multiplier on NLP constraint

- Embedded model often solves faster as an MCP than the original MCP from Nash game
- Can exploit structure to improve computational performance further
- Can iterate (on m) $t_k^m = i_k(y^m, p^m)$, and solve sequence of NLP's

$$\max_{x \in X, y \in Y} \quad \sum_k \frac{t_k^m}{\beta_k} \log U_k(x_k)$$
$$\text{s.t.} \quad \sum_k x_k \leq \sum_k \omega_k + \sum_j g_j(y_j)$$

Stochastic competing agent models (with Wets)

- Competing agents (consumers, or generators in energy market)
- Each agent maximizes objective independently (utility)
- Market prices are function of all agents activities
- Additional twist: model must "hedge" against uncertainty
- Facilitated by allowing contracts bought now, for goods delivered later
- Conceptually allows to transfer goods from one period to another (provides wealth retention or pricing of ancilliary services in energy market)

The model details: c.f. Brown, Demarzo, Eaves

Each agent maximizes:

$$u_h = -\sum_s \pi_s \left(\kappa - \prod_l c_{h,s,l}^{\alpha_{h,l}} \right)$$

Time 0:

$$\sum_l p_{0,l} c_{h,0,l} + \sum_k q_k z_{h,k} \leq \sum_l p_{0,l} e_{h,0,l}$$

Time 1:

$$\sum_l p_{s,l} c_{h,s,l} \leq \sum_l p_{s,l} \sum_k D_{s,l,k} * z_{h,k} + \sum_l p_{s,l} e_{h,s,l}$$

Additional constraints (complementarity) outside of control of agents:

$$0 \leq -\sum_h z_{h,k} \perp q_k \geq 0$$

$$0 \leq -\sum_h d_{h,s,l} \perp p_{s,l} \geq 0$$

Stochastic programming and risk measures

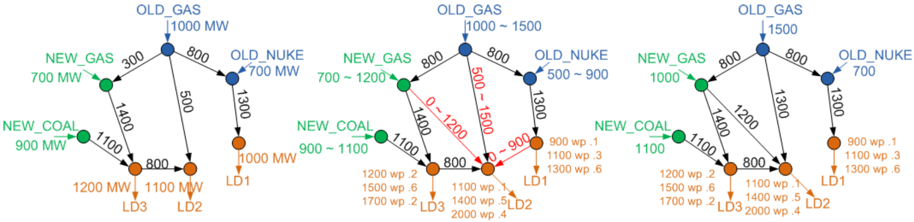
$$
\begin{aligned}
\text{SP:}\ \min \quad & c^\top x + \mathbb{R}[d^\top y] \\
\text{s.t.} \quad & Ax = b \\
& T(\omega)x + W(\omega)y(\omega) \geq h(\omega), \qquad \text{for all } \omega \in \Omega, \\
& x \geq 0, \quad y(\omega) \geq 0, \qquad \text{for all } \omega \in \Omega.
\end{aligned}
$$

Annotations are slightly more involved but straightforward:

- Need to describe probability distribution
- Define (multi-stage) structure (what variables and constraints belong to each stage)
- Define random parameters and process to generate scenarios
- Can also define risk measures on variables

Automatic reformulation (deterministic equivalent), solvers such as DECIS, etc.

Transmisson Line Capacity Expansion



Transmission Line Planning (1)

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

(budget constraints) $$s.t. \quad Ax \le b$$

$$x \ge 0$$

Generator Expansion (2)

$$\forall f \in F: \quad \min_{y_f} \sum_{\omega} \pi_{\omega} \sum_{j \in G_f} C_j(y_j, q_j^{\omega})$$

(budget constraints) $$s.t. \quad \sum_{j \in G_f} y_j \le h_f$$

$$y \ge 0$$

Day Ahead Market Clearing (3)

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

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

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

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

(gen capacity) $$\underline{u}_j \le q_j \le \overline{u}_j$$

$$\theta, z_{ij} \ free$$

Sets:

- N: Set of all buses
- X: Set of line expansions
- F: Set of firms
- G_f: Set of generators belonging to firm f

Variables:

- x: Investment in line x
- y: Investment in generator j
- z_{ij}: Real power flowing along line i-j
- q_j: Real power generated at bus j
- θ_i: Voltage phase angle at bus i
- p_i^{ω}: LMP at node i in ω

Parameters:

- ω: Demand scenarios
- π_{ω}: Probability of scenario ω
- d_i^{ω}: Demand at node i in ω
- C_j: Cost function of generator j
- Ω_{ij}: Susceptance of line i-j
- b_{ij}: Line capacity
- $\underline{u}_j$: Min generation at j
- $\bar{u}_j$: Max generation at j

Solution method

- Use deriviative free method for the upper level problem (1)
- Constraints (2) and (3) form an MCP (via EMP)
- Can show (due to specific problem structure that there is a (convex) NLP whose KKT conditions are that MCP
- Useful for theoretical analysis
- Resulting problem is too large for NLP solvers
- Can show that "Gauss-Seidel/Jacobi" method on problems in (2) and (3) converges in this case - decoupling makes problem tractable for large scale instances

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