Computation, Complementarity and Extended Mathematical Programming

Michael C. Ferris

University of Wisconsin, Madison

ICiS, Snowbird, Utah, August 3, 2010

Optimal Power Flow

$$\text{OPF}(\alpha): \quad \min_y \quad \text{energy dispatch cost } (y, \alpha)$$
$$\text{s.t.} \quad \text{conservation of power flow at nodes}$$
$$\text{Kirchoff's voltage law, and simple bound constraints}$$

α are (given) price bids, parametric optimization

Optimal Power Flow

$$\text{OPF}(\alpha): \quad \min_y \quad \text{energy dispatch cost } (y, \alpha)$$
$$\text{s.t.} \quad \text{conservation of power flow at nodes}$$
$$\text{Kirchoff's voltage law, and simple bound constraints}$$

α are (given) price bids, parametric optimization

$$\text{Leader}(\bar{\alpha}_{-i}): \quad \max_{\alpha_i, y, \lambda} \quad \text{firm } i\text{'s profit } (\alpha_i, y, \lambda)$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq \hat{\alpha}_i$$
$$y \text{ solves OPF}(\alpha_i, \bar{\alpha}_{-i})$$

Note that objective involves multiplier from OPF problem

Optimal Power Flow

$$\text{OPF}(\alpha): \quad \min_y \quad \text{energy dispatch cost } (y, \alpha)$$
$$\text{s.t.} \quad \text{conservation of power flow at nodes}$$
$$\text{Kirchoff's voltage law, and simple bound constraints}$$

α are (given) price bids, parametric optimization

$$\text{Leader}(\bar{\alpha}_{-i}): \quad \max_{\alpha_i, y, \lambda} \quad \text{firm } i\text{'s profit } (\alpha_i, y, \lambda)$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq \hat{\alpha}_i$$
$$y \text{ solves OPF}(\alpha_i, \bar{\alpha}_{-i})$$

Note that objective involves multiplier from OPF problem

$$\text{Leader}(\bar{\alpha}_{-i}): \quad \max_{\alpha_i, y, \lambda} \quad \text{firm } i\text{'s profit } (y, \lambda, \alpha)$$
$$\text{s.t.} \quad 0 \leq \alpha_i \leq \hat{\alpha}_i$$
$$y, \lambda \text{ solves KKT(OPF}(\alpha_i, \bar{\alpha}_{-i}))$$

This is an example of an MPCC since KKT form complementarity constraints

Multi-player EPEC and security constraints

- $(\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_m)$ is an equilibrium if

$$\bar{\alpha}_i \text{ solves Leader}(\bar{\alpha}_{-i}), \quad \forall i$$

- (Nonlinear) Nash Equilibrium where each player solves an MPCC

Multi-player EPEC and security constraints

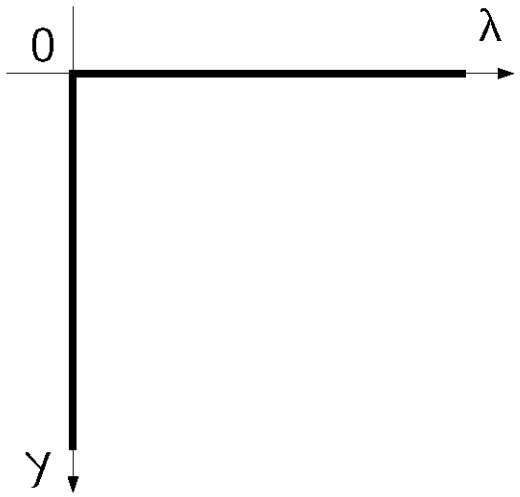
- $(\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_m)$ is an equilibrium if

$$\bar{\alpha}_i \text{ solves Leader}(\bar{\alpha}_{-i}), \quad \forall i$$

- (Nonlinear) Nash Equilibrium where each player solves an MPCC
- MPCC is hard (lacks a constraint qualification)
- Nash Equilibrium is PPAD-complete (Chen et al, Papadimitriou et al)
- In practice, also require contingency (scenario) constraints imposed in the OPF problem

- Leader/follower game: Stackleberg
- Supply chains with "market leader"

Complementarity Problems via Graphs



$$\mathcal{T} = \mathcal{N}_{\mathbf{R}_+} = (\mathbf{R}_+ \times \{0\}) \bigcup (\{0\} \times \mathbf{R}_-)$$

$$-y \in \mathcal{T}(\lambda) \iff (\lambda, -y) \in \mathcal{T} \iff 0 \leq \lambda \perp y \geq 0$$

By approximating (smoothing) graph can generate interior point algorithms for example $y\lambda = \epsilon, y, \lambda > 0$

$$-F(x) \in \mathcal{N}_{\mathbf{R}^n_+}(x) \iff (x, -F(x)) \in \mathcal{T}^n \iff 0 \leq x \perp F(x) \geq 0$$

Variational Inequality Formulation

- $F : \mathbf{R}^n \to \mathbf{R}^n$
- Ideally: $X \subseteq \mathbf{R}^n$ – constraint set
- In practice: $X \subseteq \mathbf{R}^n$ – simple bounds

$$0 \in F(x) + N_X(x)$$

- Special Cases
  - Nonlinear Equations ($X \equiv \mathbf{R}^n$)

$$F(x) = 0$$

  - Nonlinear Complementarity Problem ($X \equiv \mathbf{R}^n_+$)

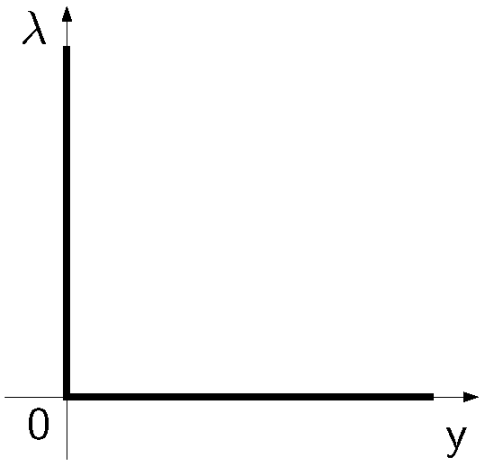
$$0 \leq F(x) \quad x \geq 0$$
$$x^T F(x) = 0$$

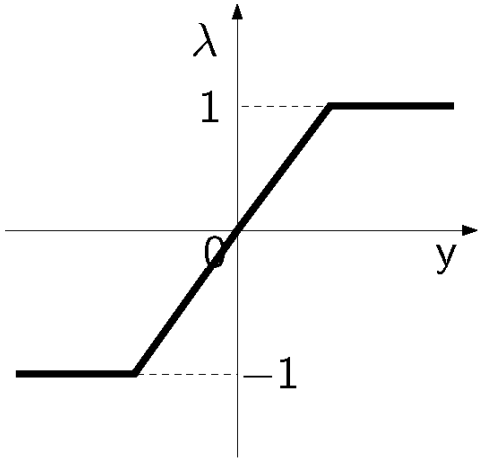
Complementarity Systems

$$\frac{dx}{dt}(t) = f(x(t), \lambda(t))$$

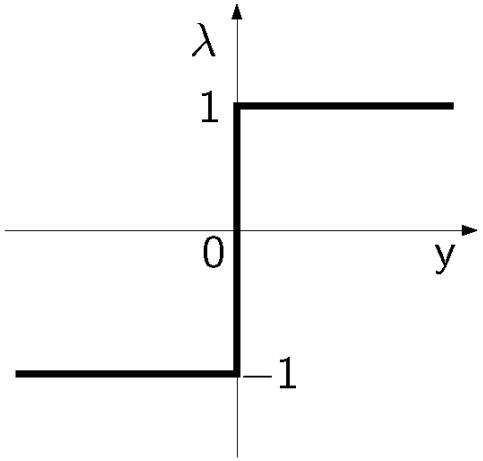
$$y(t) = h(x(t), \lambda(t))$$

$$0 \leq y(t) \perp \lambda(t) \geq 0$$

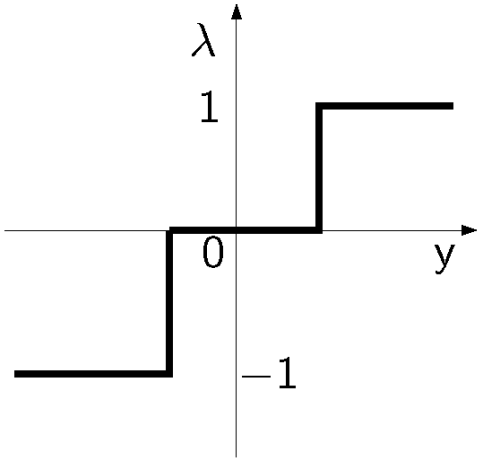




saturation



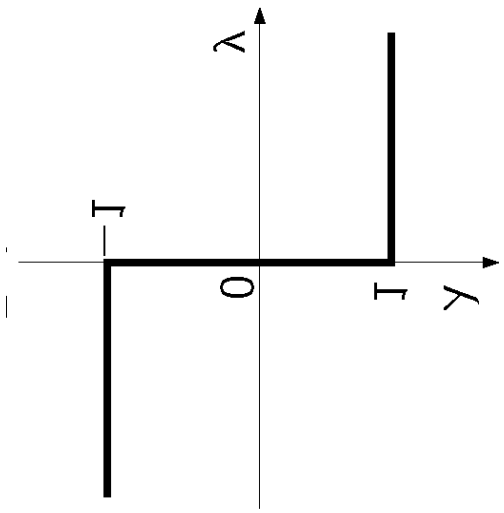
Relay



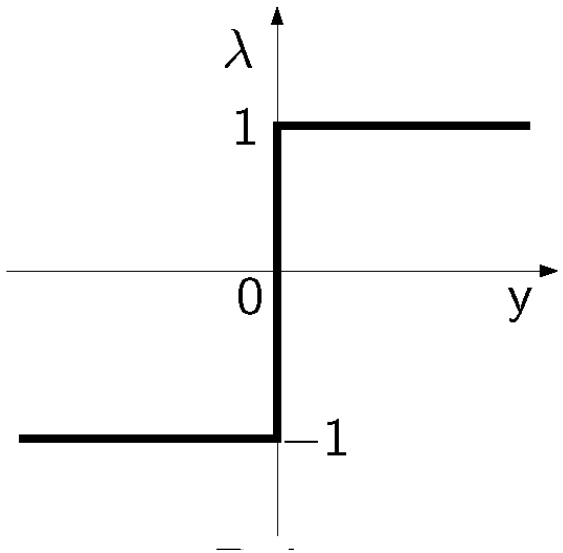
Relay with dead zone

Operators and Graphs ($X = [\ell, u]$)

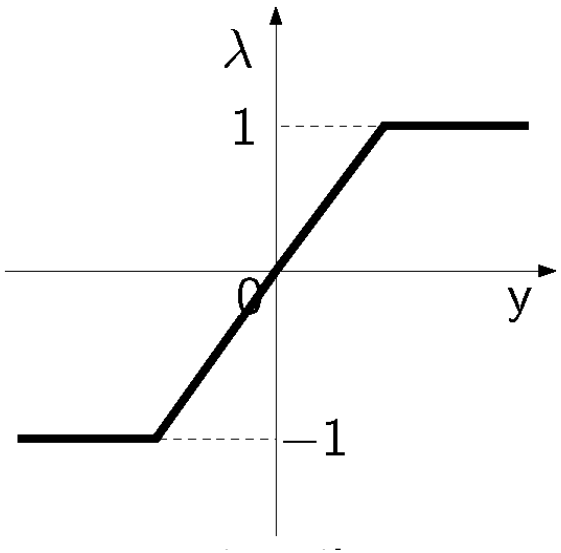
$$x_i = \ell_i, -F_i(x) \leq 0 \text{ or } x_i \in (\ell_i, u_i), -F_i(x) = 0 \text{ or } x_i = u_i, -F_i(x) \geq 0$$



$\mathcal{T}(\lambda)$



$\mathcal{T}^{-1}(y)$



$(\mathcal{I} + \mathcal{T})^{-1}(y) = P_{\mathcal{T}}(y)$

$P_{\mathcal{T}}(y)$ is the projection of y onto $[\ell, u]$

Generalized Equations

- Suppose $\mathcal{T}$ is a maximal monotone operator

$$0 \in F(z) + \mathcal{T}(z) \qquad (GE)$$

- Define $P_{\mathcal{T}} = (\mathcal{I} + \mathcal{T})^{-1}$
- If $\mathcal{T}$ is polyhedral (graph of $\mathcal{T}$ is a finite union of convex polyhedral sets) then $P_{\mathcal{T}}$ is piecewise affine (continous, single-valued, non-expansive)

$$\begin{aligned} 0 \in F(z) + \mathcal{T}(z) \quad &\iff \quad z \in F(z) + \mathcal{I}(z) + \mathcal{T}(z) \\ &\iff \quad z - F(z) \in (\mathcal{I} + \mathcal{T})(z) \iff P_{\mathcal{T}}(z - F(z)) = z \end{aligned}$$

Use in fixed point iterations (cf projected gradient methods)

Splitting Methods

- Suppose $\mathcal{T}$ is a maximal monotone operator

$$0 \in F(z) + \mathcal{T}(z) \quad (GE)$$

- Can devise Newton methods (e.g. SQP) that treat F via calculus and $\mathcal{T}$ via convex analysis
- Alternatively, can split $F(z) = A(z) + B(z)$ (and possibly $\mathcal{T}$ also) so we solve solve (GE) by solving a sequence of problems involving just

$$\mathcal{T}_1(z) = A(z) \text{ and } \mathcal{T}_2(z) = B(z) + \mathcal{T}(z)$$

  where each of these is "simpler"
- Forward-Backward splitting:

$$z^{k+1} = (I + c_k T_2)^{-1} (I - c_k T_1) \left(z^k \right),$$

Normal Map

- Suppose T is a maximal monotone operator

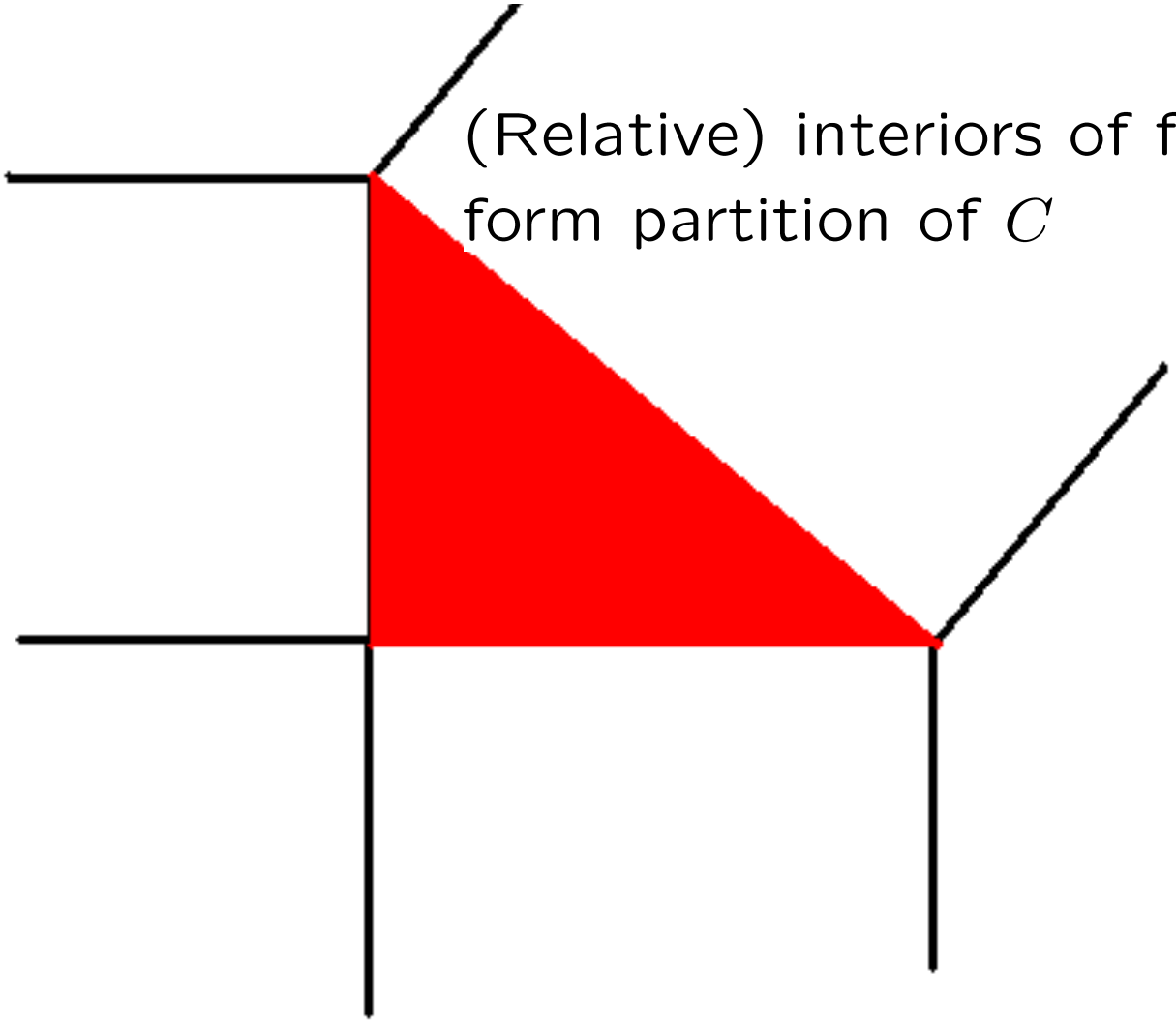
$$0 \in F(z) + T(z) \qquad (GE)$$

- Define $P_T = (I + T)^{-1}$

$$\begin{aligned}
0 \in F(z) + \mathcal{T}(z) \quad &\Longleftrightarrow \quad z \in F(z) + \mathcal{I}(z) + \mathcal{T}(z) \\
&\Longleftrightarrow \quad z - F(z) = y \text{ and } y \in (\mathcal{I} + \mathcal{T})(z) \\
&\Longleftrightarrow \quad z - F(z) = y \text{ and } P_{\mathcal{T}}(y) = z \\
&\Longleftrightarrow \quad P_{\mathcal{T}}(y) - F(P_{\mathcal{T}}(y)) = y \\
&\Longleftrightarrow \quad 0 = F(P_{\mathcal{T}}(y)) + y - P_{\mathcal{T}}(y)
\end{aligned}$$

This is the so-called Normal Map Equation

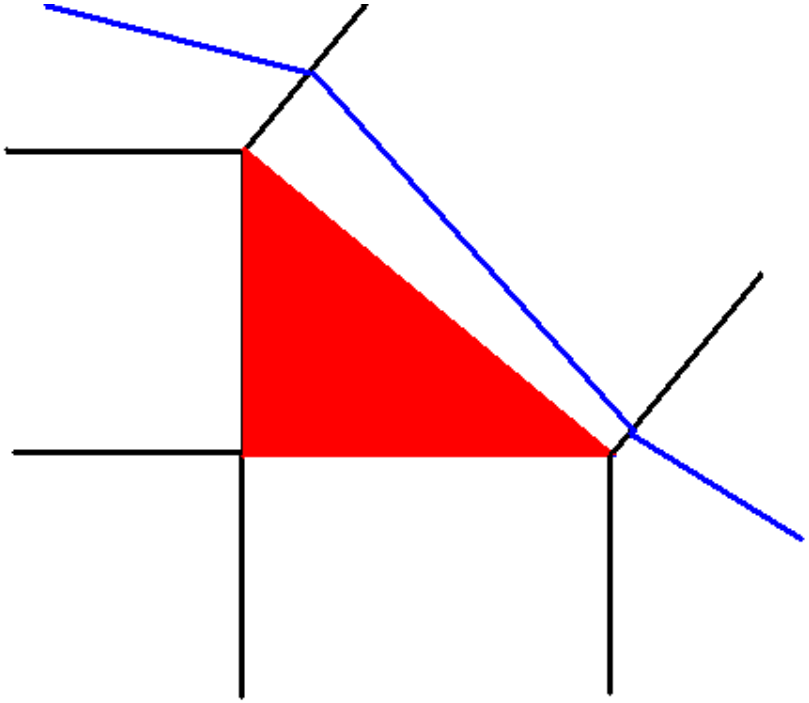
Normal manifold $= \{F_i + N_{F_i}\}$



(Relative) interiors of faces F_i form partition of C

Cao/Ferris Path (Eaves)

- Start in cell that has interior (face is an extreme point)
- Move towards a zero of affine map in cell
- Update direction when hit boundary (pivot)
- Solves or determines infeasible if M is copositive-plus on rec(C)
- Solves 2-person bimatrix games, 3-person games too, but these are nonlinear



But algorithm has exponential complexity (von Stengel et al)

Extensions and Computational Results

- Embed AVI solver in a Newton Method - each Newton step solves an AVI
- Compare performance of PathAVI with PATH on equivalent LCP
- PATH the most widely used code for solving MCP
- AVIs constructed to have solution with $M_{n\times n}$ symmetric indefinite

	PathAVI		*PATH*	
Size (m,n)	*Resid*	*Iter*	*Resid*	*Iter*
$(180, 60)$	3×10^{-14}	193	0.9	10176
$(360, 120)$	3×10^{-14}	1516	2.0	10594

- 2 - 10x speedup in Matlab using sparse LU instead of QR
- 2 - 10x speedup in C using sparse LU updates

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.
- Give three examples of this: disjunctive programming, multi-agent competitive models and bilevel programming

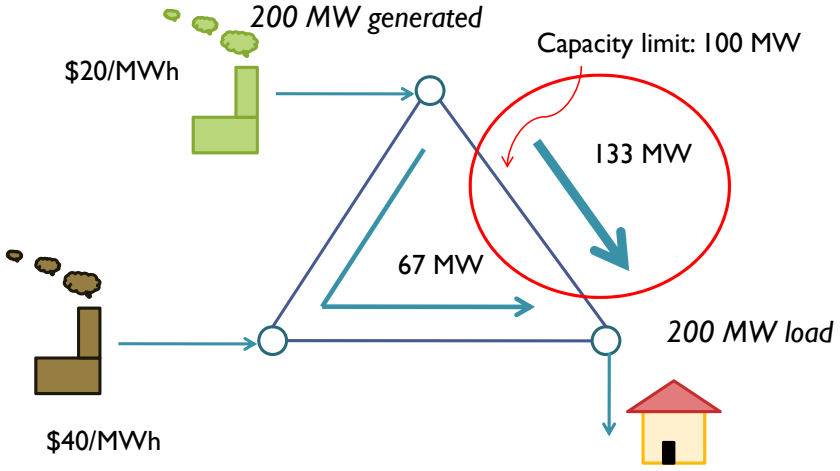
EMP(i): constraint logic

Sequencing Example to minimize makespan:

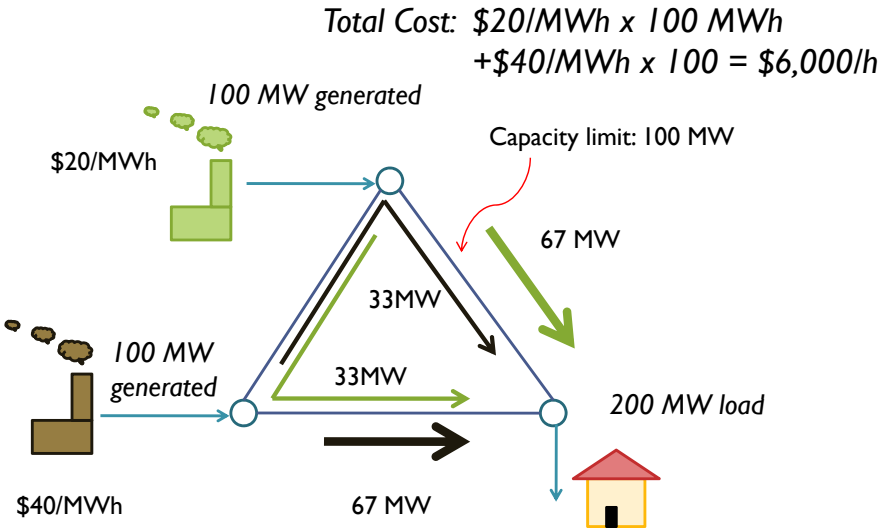
- seq(i,j): start(i) + wait(i,j) $\leq$ start(j)
- for each pair ($i \neq j$), either i before j or j before i
- empinfo: disjunction * seq(i,j) else seq(j,i)
- i.e. write down all seq equations, only enforce one of every pair
- EMP options facilitate either Big M reformulation, or Convex Hull reformulation (Grossmann et al), or CPLEX indicator reformulation
- Other logic constructs available

Transmission switching

Opening lines in a transmission network can reduce cost



(a) Infeasible due to line capacity



(b) Feasible dispatch

Need to use expensive generator due to power flow characteristics and capacity limit on transmission line

The basic model

$$
\begin{array}{lll}
\min_{g,f,\theta} & c^T g & \text{generation cost} \\
\text{s.t.} & g - d = Af, f = BA^T\theta & A \text{ is node-arc incidence} \\
 & \bar{\theta}_L \leq \theta \leq \bar{\theta}_U & \text{bus angle constraints} \\
 & \bar{g}_L \leq g \leq \bar{g}_U & \text{generator capacities} \\
 & \bar{f}_L \leq f \leq \bar{f}_U & \text{transmission capacities}
\end{array}
$$

with transmission switching (within a smart grid technology) we modify as:

$$
\begin{array}{lll}
\min_{g,f,\theta} & c^T g & \\
\text{s.t.} & g - d = Af & \\
 & \bar{\theta}_L \leq \theta \leq \bar{\theta}_U & \\
 & \bar{g}_L \leq g \leq \bar{g}_U & \\
\text{either} & f_i = (BA^T\theta)_i, \bar{f}_{L,i} \leq f_i \leq \bar{f}_{U,i} & \text{if } i \text{ closed} \\
\text{or} & f_i = 0 & \text{if } i \text{ open}
\end{array}
$$

Use EMP to facilitate the disjunctive constraints (several equivalent formulations, including LPEC)

LogMip: Generalized disjunctive programming

$$\min \quad Z = \sum_k c_k + f(x)$$ **Objective Function**

$$s.t. \quad r(x) \le 0$$ **Common Constraints**

OR operator $\longrightarrow$ $$\bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ g_{jk}(x) \le 0 \\ c_k = \gamma_{jk} \end{bmatrix}, k \in K$$

Disjunction

Constraints

Fixed Charges

$$\Omega(Y) = true$$ **Logic Propositions**

$$x \in R^n, c_k \in R^1$$ **Continuous Variables**

$$Y_{jk} \in \{true, \ false\}$$ **Boolean Variables**

EMP(ii): MPCC: complementarity constraints

$$\begin{array}{ll} \min\limits_{x,s} & f(x,s) \\ \text{s.t.} & g(x,s) \leq 0, \\ & 0 \leq s \perp h(x,s) \geq 0 \end{array}$$

- g, h model "engineering" expertise: finite elements, etc
- $\perp$ models complementarity, disjunctions
- Complementarity "$\perp$" constraints available in AIMMS, AMPL and GAMS

EMP(ii): MPCC: complementarity constraints

$$\begin{array}{ll} \min\limits_{x,s} & f(x,s) \\ \text{s.t.} & g(x,s) \leq 0, \\ & 0 \leq s \perp h(x,s) \geq 0 \end{array}$$

- g, h model "engineering" expertise: finite elements, etc
- $\perp$ models complementarity, disjunctions
- Complementarity "$\perp$" constraints available in AIMMS, AMPL and GAMS
- NLPEC: use the convert tool to automatically reformulate as a parameteric sequence of NLP's
- Solution by repeated use of standard NLP software
  - Problems solvable, local solutions, hard

Hierarchical models

- Bilevel programs:

$$\begin{array}{ll} \min\limits_{x^*,y^*} & f(x^*,y^*) \\ \text{s.t.} & g(x^*,y^*) \leq 0, \\ & y^* \text{ solves } \min\limits_{y} v(x^*,y) \text{ s.t. } h(x^*,y) \leq 0 \end{array}$$

- model bilev /deff,defg,defv,defh/;
  empinfo: bilevel min v y defv defh
- EMP tool automatically creates the MPCC

$$\begin{array}{ll} \min\limits_{x^*,y^*,\lambda} & f(x^*,y^*) \\ \text{s.t.} & g(x^*,y^*) \leq 0, \\ & 0 \leq \nabla v(x^*,y^*) + \lambda^T \nabla h(x^*,y^*) \quad \perp y^* \geq 0 \\ & 0 \leq -h(x^*,y^*) \quad \perp \lambda \geq 0 \end{array}$$

Large scale example: bioreactor (with Niebel)

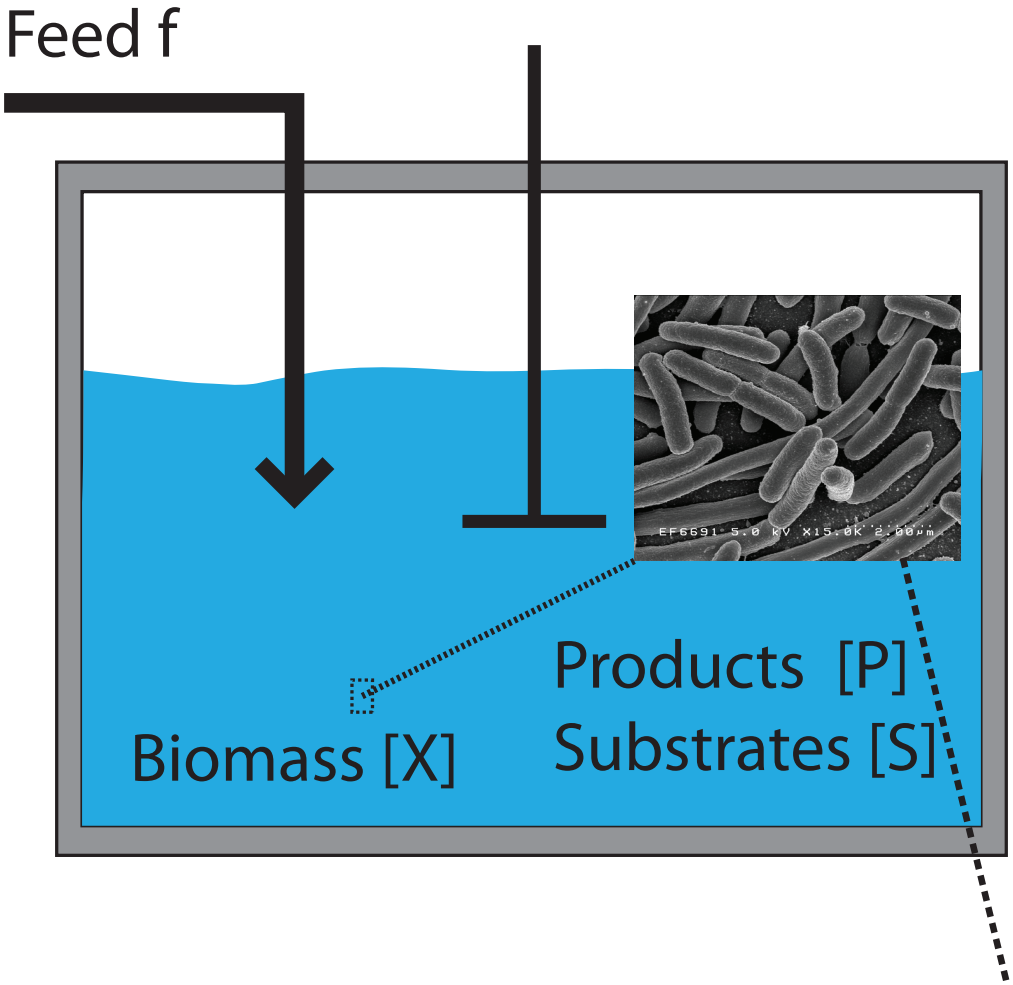
Challenge

Formulating an optimization problem that allows the estimation of the dynamic changes in intracellular fluxes based on measured external bioreactor concentrations.

Approach

Using existing constraint-based stoichiometric models of the cellular metabolism to formulate a bilevel dynamic optimization problem.

Bioreactor



from: Rocky Mountain Laboratories, NIAID, NIH

When feed then fed-batch, else batch reactor.

constant environmental conditions, such as

- temperature
- pH level
- pressure

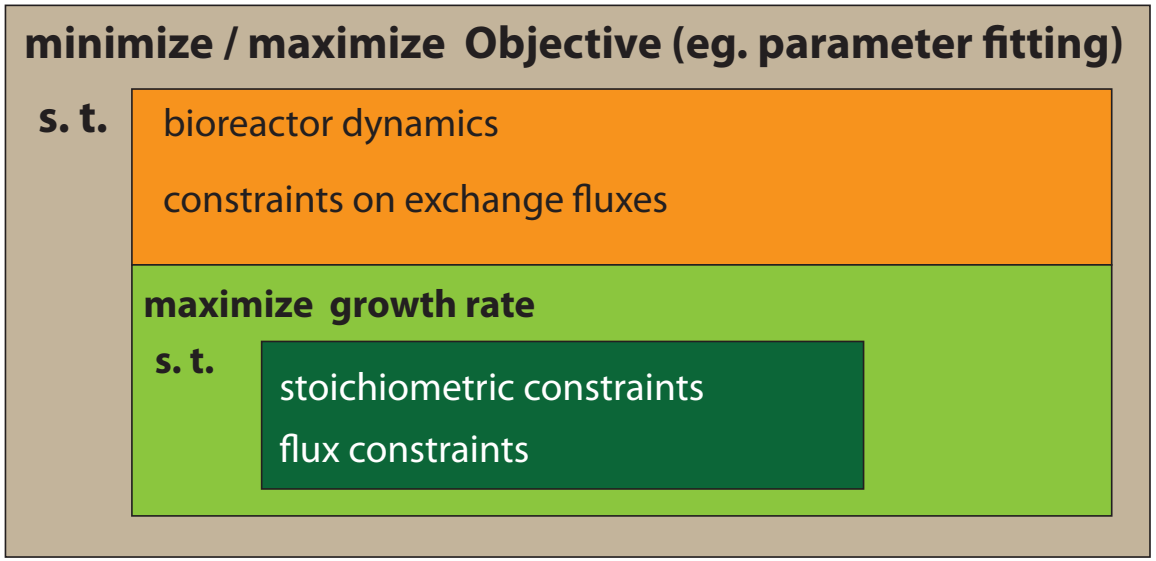
run time: **days**

most industrial applications with biological processes, such as

- fermentation
- biochemical production
- pharmaceutical protein production

Dynamic optimization

Approach:
The different timescales of the metabolism (**fast**) and the reactor growth (**slow**), allows to assume steady-state for the metabolism.

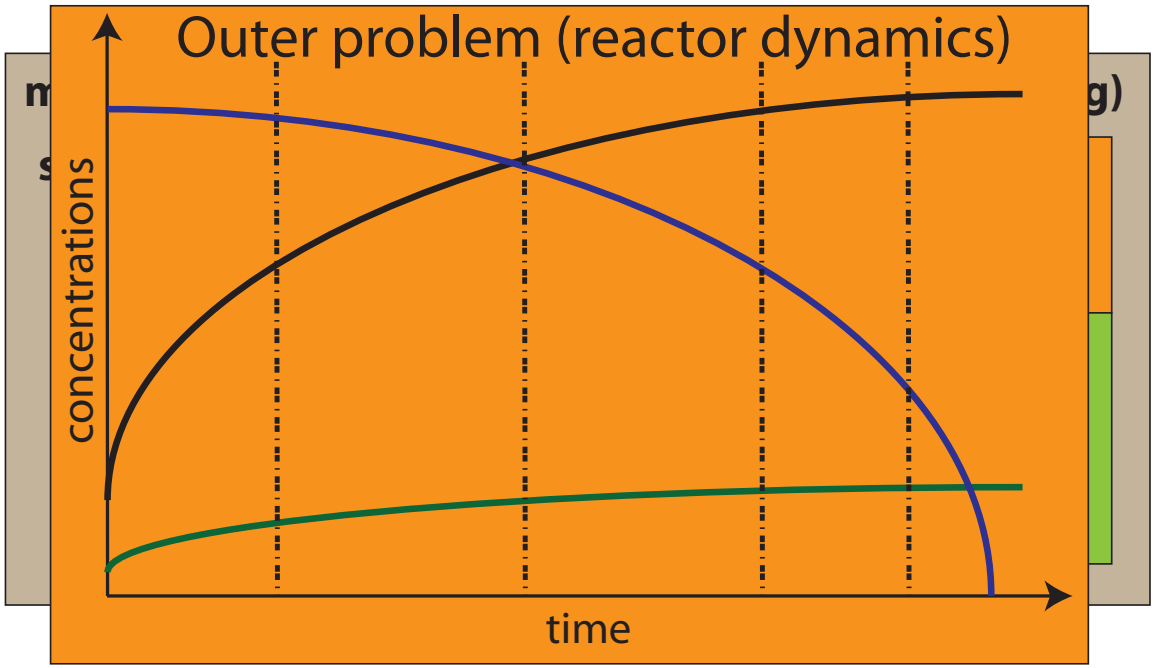


Different mathematical programming techniques are used to transform the problem to a nonlinear program. The differential equations are transformed into nonlinear constraints using collocation methods.

Dynamic optimization

Approach:

The different timescales of the metabolism (**fast**) and the reactor growth (**slow**), allows to assume steady-state for the metabolism.



Different mathematical programming techniques are used to transform the problem to a nonlinear program. The differential equations are transformed into nonlinear constraints using collocation methods.

Variational inequalities

- Find $z \in X$ such that

$$0 \in F(z) + \mathcal{N}_X(z)$$

- Many applications where F is not the derivative of some f
- model vi / F, g /;
  empinfo: vifunc F z
- Convert problem into complementarity problem by introducing multipliers on representation of X
- Can now do MPEC (as opposed to MPCC)!
- Projection algorithms, robustness (evaluate F only at points in X)

EMP(iii): Embedded models

- Model has the format:

$$\text{Agent o:} \quad \min_x \quad f(x,y)$$
$$\text{s.t.} \quad g(x,y) \leq 0 \quad (\perp \lambda \geq 0)$$

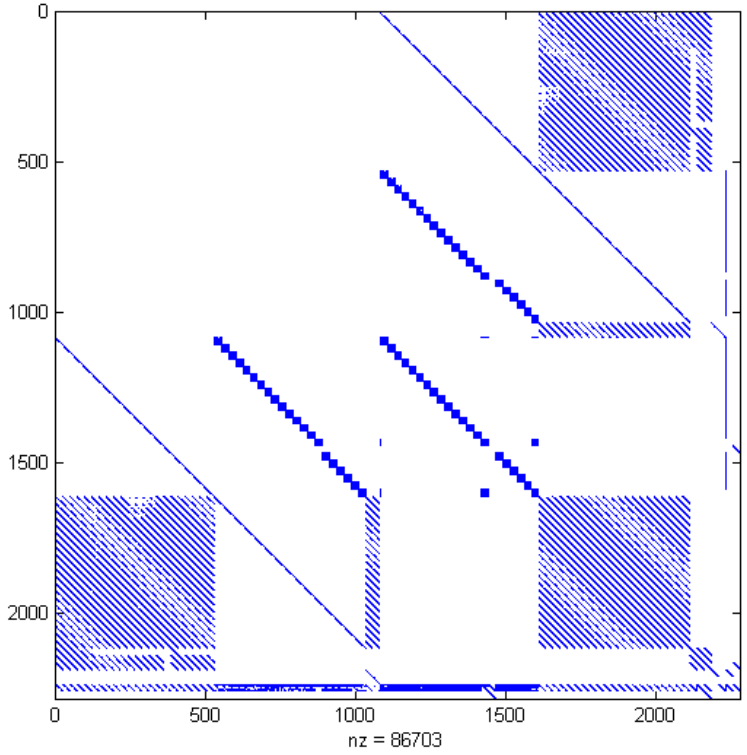
$$\text{Agent v:} \quad H(x,y,\lambda) = 0 \qquad (\perp y \text{ free})$$

- Difficult to implement correctly (multiple optimization models)
- Can do automatically - simply annotate equations
  empinfo: equilibrium
  min f x defg
  vifunc H y dualvar λ defg
- EMP tool automatically creates an MCP

$$\nabla_x f(x,y) + \lambda^T \nabla g(x,y) = 0$$
$$0 \leq -g(x,y) \perp \lambda \geq 0$$
$$H(x,y,\lambda) = 0$$

World Bank Project (Uruguay Round)

- 24 regions, 22 commodities
  - Nonlinear complementarity problem
  - Size: 2200 x 2200
- Short term gains $53 billion p.a.
  - Much smaller than previous literature
- Long term gains $188 billion p.a.
  - Number of less developed countries loose in short term
- Unpopular conclusions - forced concessions by World Bank
- Region/commodity structure not apparent to solver



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.

Key point: models generated correctly solve quickly

Here S is mesh spacing parameter

S	Var	rows	non-zero	dense(%)	Steps	RT (m:s)
20	2400	2568	31536	0.48	5	0 : 03
50	15000	15408	195816	0.08	5	0 : 19
100	60000	60808	781616	0.02	5	1 : 16
200	240000	241608	3123216	0.01	5	5 : 12

Convergence for $S = 200$ (with new basis extensions in PATH)

Iteration	Residual
0	1.56(+4)
1	1.06(+1)
2	1.34
3	2.04(−2)
4	1.74(−5)
5	2.97(−11)

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):

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

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

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

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:

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

Time 1:

$$d_{h,s,l} = c_{h,s,l} - e_{h,s,l} - \sum_k D_{s,l,k} * z_{h,k}, \quad \sum_l p_{s,l} d_{h,s,l} \leq 0$$

Additional constraints (complementarity) outside of control of agents:

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

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

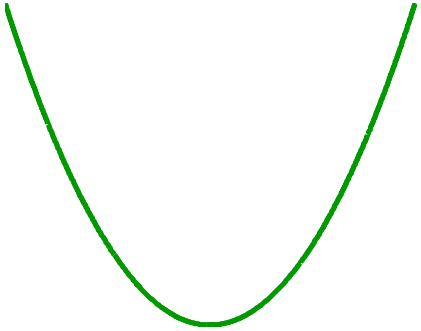
Conclusions

- 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

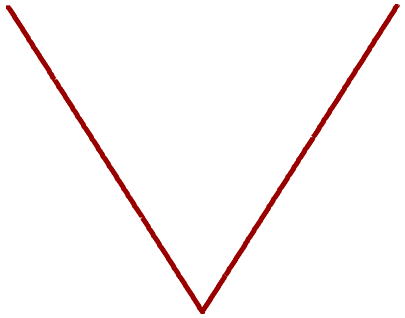
EMP(iv): Extended nonlinear programs

$$\min_{x \in X} f_0(x) + \theta(f_1(x), \ldots, f_m(x))$$

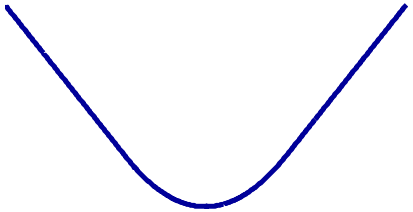
Examples of different θ



least squares,



absolute value,

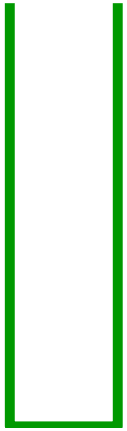


Huber function

Solution reformulations are very different

Huber function used in robust statistics.

More general θ functions



In general any piecewise linear penalty function can be used: (different upside/downside costs).

General form:

$$\theta(u) = \sup_{y \in Y} \{ y^T u - k(y) \}$$

θ nonsmooth due to the max term; θ separable in example.

θ is always convex.

First order conditions

- Solution via reformulation. One way:

$$\begin{array}{rlcl} 0 \in & \nabla_x \mathcal{L}(x,y) & + & N_X(x) \\ 0 \in & -\nabla_y \mathcal{L}(x,y) & + & N_Y(y) \end{array}$$

  $N_X(x)$ is the normal cone to the closed convex set X at x.
- Automatically creates an MCP (or a VI)
- Already available!
- To do: extend X and Y beyond simple bound sets.

Alternative Reformulations

Convert does symbolic/numeric reformulations. Alternative NLP formulations also possible.

$$k(y) = \frac{1}{2} y' Q y, \quad X = \{x : Rx \leq r\}, \quad Y = \left\{y : S' y \leq s\right\}$$

Defining

$$Q = DJ^{-1}D', \quad F(x) = (f_1(x), \ldots, f_m(x))$$

$$\begin{array}{ll} \min & f_0(x) + s'z + \frac{1}{2} wJw \\ \text{s.t.} & Rx \leq r, z \geq 0, F(x) - Sz - Dw = 0 \end{array}$$

Can set up better (solver) specific formulation.