Extended Mathematical Programming

Michael C. Ferris

University of Wisconsin, Madison

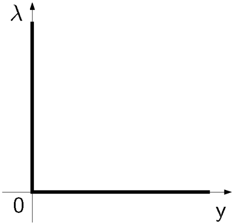
Nonsmooth Mechanics Summer School, June 15, 2010

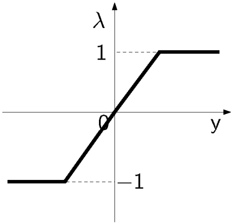
Complementarity Systems

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

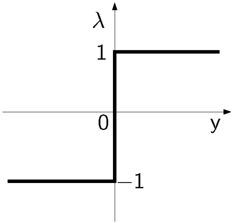
$$y(t) = h(x(t), \lambda(t))$$

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

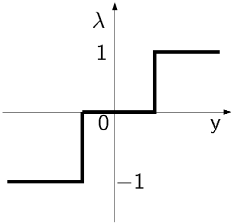




saturation



Relay



Relay with dead zone

Generalized Generalized Equations

- Suppose T is a maximal monotone operator

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

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

$$0 = F(P_T(x)) + x - P_T(x)$$

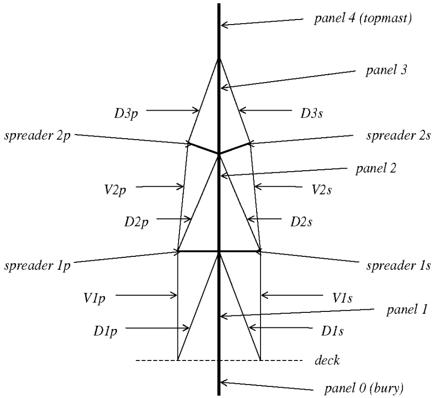
and the "path following" algorithm can be defined similarly to the variational inequality case.

Optimal Yacht Rig Design

- Current mast design trends use a large primary spar that is supported laterally by a set of tension and compression members, generally termed the rig
- Reduction in either the weight of the rig or the height of the VCG will improve performance
- Design must work well under a variety of weather conditions

Complementarity feature

- Stays are tension only members (in practice) - Hookes Law
- When tensile load becomes zero, the stay goes slack (low material stiffness)
- $0 \geq s \perp s - k\delta \leq 0$
  - s axial load
  - k member spring constant
  - δ member extension
- Either $s_i = 0$ or $s_i = k\delta_i$



MPCC: complementarity constraints

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

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

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
  - Southern Spars Company (NZ): improved from 5-0 to 5-2 in America's Cup!

How to use it?

- Download "gams" system: google "download gams distribution"
- Evaluation license provided for "full versions" of PATH, CONOPT, MINOS, MOSEK, NLPEC, MILES, EMP
- License files available at:
  http://www.cs.wisc.edu/~ferris/windows.txt
  or
  http://www.cs.wisc.edu/~ferris/linux.txt
  or
  http://www.cs.wisc.edu/~ferris/mac.txt

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, bilevel programming and multi-agent competitive models

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

LogMip: Generalized disjunctive programming

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

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

OR operator $\longrightarrow$

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

Disjunction

Constraints

Fixed Charges

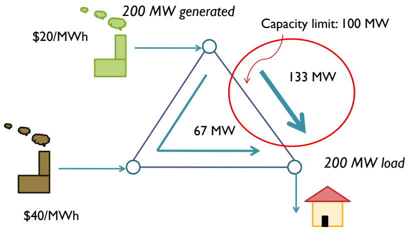
$$\Omega(Y) = true$$ **Logic Propositions**

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

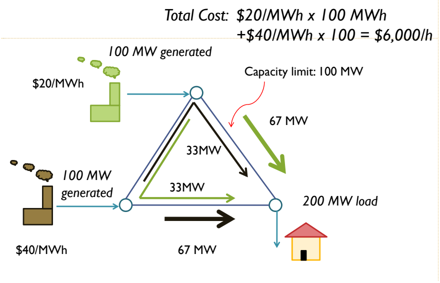
$$Y_{jk} \in \{true, \ false\}$$ **Boolean Variables**

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)

Example: Robust Linear Programming

Data in LP not known with certainty:

$$\min c^T x \text{ s.t. } a_i^T x \leq b_i, i = 1, 2, \ldots, m$$

Suppose the vectors a_i are known to be lie in the ellipsoids

$$a_i \in \varepsilon_i := \{\bar{a}_i + P_i u : \|u\|_2 \leq 1\}$$

where $P_i \in \mathbf{R}^{n \times n}$ (and could be singular, or even 0).
Conservative approach: robust linear program

$$\min c^T x \text{ s.t. } a_i^T x \leq b_i, \text{ for all } a_i \in \varepsilon_i, i = 1, 2, \ldots, m$$

Robust Linear Programming as SOCP/ENLP

The constraints can be rewritten as:

$$\begin{aligned} b_i &\geq \sup\left\{a_i^T x : a_i \in \varepsilon_i\right\} \\ &= \bar{a}_i^T x + \sup\left\{u^T P_i^T x : \|u\|_2 \leq 1\right\} = \bar{a}_i^T x + \left\|P_i^T x\right\|_2 \end{aligned}$$

Thus the robust linear program can be written as

$$\min c^T x \text{ s.t. } \bar{a}_i^T x + \left\|P_i^T x\right\|_2 \leq b_i, i = 1, 2, \ldots, m$$

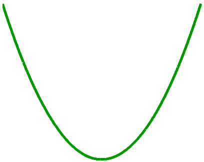
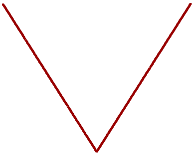
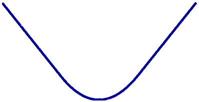
$$\min c^T x + \sum_{i=1}^{m} \psi_C(b_i - \bar{a}_i^T x, P_i^T x)$$

where C represents the second-order cone. Our extension allows automatic reformulation and solution (as SOCP) by Mosek or Conopt.

EMP(ii): 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:

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

First order conditions for optimality are an MCP!

ENLP (Rockafellar): Primal problem

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

"Classical" problem:

$$\begin{array}{ll} \min\limits_{x_1, x_2, x_3} & \exp(x_1) \\ \text{s.t.} & \log(x_1) = 1 \\ & x_2^2 \leq 2 \\ & x_1/x_2 = \log(x_3), 3x_1 + x_2 \leq 5, x_1 \geq 0, x_2 \geq 0 \end{array}$$

Soft penalization of red constraints:

$$\begin{array}{ll} \min\limits_{x_1, x_2, x_3} & \exp(x_1) + 5 \left\| \log(x_1) - 1 \right\|^2 + 2 \max(x_2^2 - 2, 0) \\ \text{s.t.} & x_1/x_2 = \log(x_3), 3x_1 + x_2 \leq 5, x_1 \geq 0, x_2 \geq 0 \end{array}$$

ENLP: Primal problem

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

$$X = \left\{ x \in \mathbf{R}^3 : 3x_1 + x_2 \leq 5, x_1 \geq 0, x_2 \geq 0 \right\}$$

$$f_1(x) = \log(x_1) - 1, f_2(x) = x_2^2 - 2, f_3(x) = x_1/x_2 - \log(x_3)$$

$$\theta_1(u) = 5 \left\| u \right\|^2, \theta_2(u) = 2 \max(u, 0), \theta_3(u) = \psi_{\{0\}}(u)$$

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

θ is always convex.

Specific choices of k and Y

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

- L_2: $k(y) = \frac{1}{4\lambda} y^2$, $Y = (-\infty, +\infty)$
- L_1: $k(y) = 0$, $Y = [-\rho, \rho]$
- L_∞: $k(y) = 0$, $Y = \Delta$, unit simplex
- Huber: $k(y) = \frac{1}{4\lambda} y^2$, $Y = [-\rho, \rho]$
- Second order cone constraint: $k(y) = 0$, $Y = C^\circ$

Elegant Duality

For these θ (defined by $k(\cdot)$, Y), duality is derived from the Lagrangian:

$$\mathcal{L}(x,y) = f_0(x) + \sum_{i=1}^m y_i f_i(x) - k(y)$$
$$x \in X, y \in Y$$

- Dual variables in Y not simply ≥ 0 or free.
- Saddle point theory, under convexity.
- Dual Problem and Complete Theory.
- Special case: ELQP - dual problem is also an ELQP.

Implementation: convert tool

```
e1.. obj =e= exp(x1);
e2.. log(x1)-1 =e= 0;
e3.. sqr(x2)-2 =e= 0;
e4.. x1/x2 =e= log(x3);
e5.. 3*x1 + x2 =l= 5;

$onecho > emp.info
strategy mcp
adjustequ
e2 sqr 5
e3 maxz 2
$offecho

solve mod using emp min obj;
```

Library of different θ functions implemented.

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.

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

Defining

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

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

Can set up better (solver) specific formulation.

EMP(iii): Variational inequalities

- Find $z \in C$ such that

$$\langle F(z), y - z \rangle \geq 0, \quad \forall y \in C$$

- 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 C
- Can now do MPEC (as opposed to MPCC)!
- Projection algorithms, robustness (evaluate F only at points in C)

Bimatrix Games

- AVI can be used to formulate many standard problem instances corresponding to special choices of M and $\mathcal{C}$.
- 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

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

Formulation using complemetarity

The optimality conditions for the above problems are:

$$-Aq^* \in N_{\triangle_I}(p^*) \text{ and } -B^T p^* \in N_{\triangle_J}(q^*)$$

Therefore the corresponding VI is affine and can be written as:

$$0 \in \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} + N_{\triangle_I \times \triangle_J}\left(\begin{bmatrix} p \\ q \end{bmatrix}\right). \tag{1}$$

EMP(iv): Embedded models

- Model has the format:

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

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

- 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

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

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)

Competing agent models

- Competing agents (consumers)
- 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)

The model details: 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:

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

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

EMP(v): Heirarchical 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

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

Biological Pathway Models

Opt knock (a bilevel program)

max	bioengineering objective (through gene knockouts)	
s.t.	max cellular objective (over fluxes)	
	s.t.	fixed substrate uptake
		network stoichiometry
		blocked reactions (from outer problem)
	number of knockouts $\leq$ limit	

Biological Pathway Models

Opt knock (a bilevel program)

max bioengineering objective (through gene knockouts)

s.t. max cellular objective (over fluxes)

s.t. fixed substrate uptake

network stoichiometry

blocked reactions (from outer problem)

number of knockouts $\leq$ limit

Also prediction models of the form:

$$\begin{aligned} \min \sum_{i,j} \|w_i - v_j\| \\ \text{s.t. } Sv = w \\ -\bar{v}_L \leq v \leq \bar{v}_U,\ w_j = \bar{w}_j \end{aligned}$$

Can be modeled as an SOCP.

The overall scheme!

- Collection of algebraic equations
- Form a bilevel program via emp
- EMP tool automatically creates the MPCC (model transformation)
- NLPEC tool automatically creates (a series of) NLP models (model transformation)
- GAMS automatically rewrites NLP models for global solution via BARON (model transformation)
- Is this global? What's the hitch?

The overall scheme!

- Collection of algebraic equations
- Form a bilevel program via emp
- EMP tool automatically creates the MPCC (model transformation)
- NLPEC tool automatically creates (a series of) NLP models (model transformation)
- GAMS automatically rewrites NLP models for global solution via BARON (model transformation)
- Is this global? What's the hitch?
- Note that heirarchical structure is available to solvers for analysis or utilization

Large scale example: bioreactor

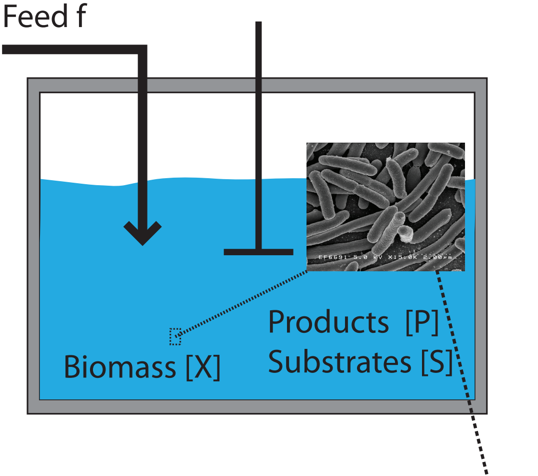
Challenge

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 model of a bioreactor

Assumptions: well stirred, one phase!
Biomass:

$$\frac{d[X]}{dt} = (\mu - \frac{f}{V})[X]$$

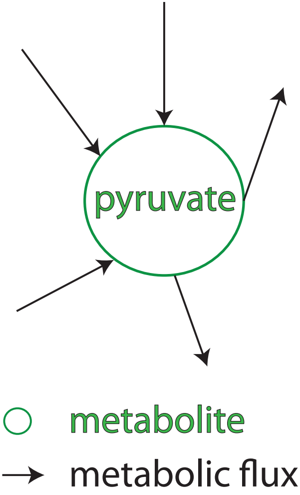
μ: growth rate
Product [P] or substrate [S] concentrations [C]:

$$\frac{d[C]}{dt} = q_{[C]}[X] + (f[C]^{\text{feed}} - \frac{f}{V}[C])$$

$q_{[C]}$: specific uptake or production rate of [C].
Volume V:

$$\frac{d[V]}{dt} = f$$

Stoichiometric constraints



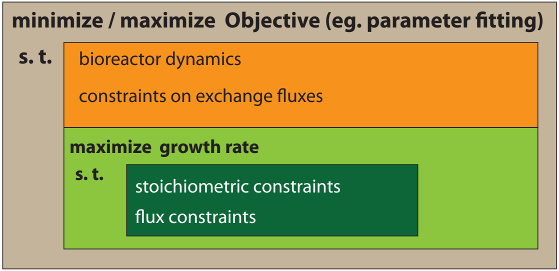
The stochiometry of the cellular metabolism is described by a stoichiometric matrix S.
S constrains steady-state flux distributions.

$$S \cdot v = 0$$

The above relation can be used in a linear programming problem, which maximizes for a cellular objective function
(**flux balance analysis).**

Dynamic optimization

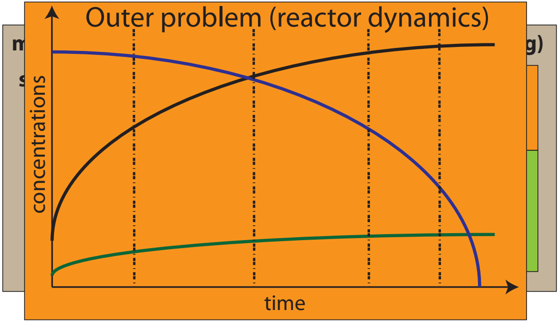
Approach:
The different timescales of the metabolism (**fast**) and the reactor growth (**slow**), allows to assume steady-state for the metabolism.



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.

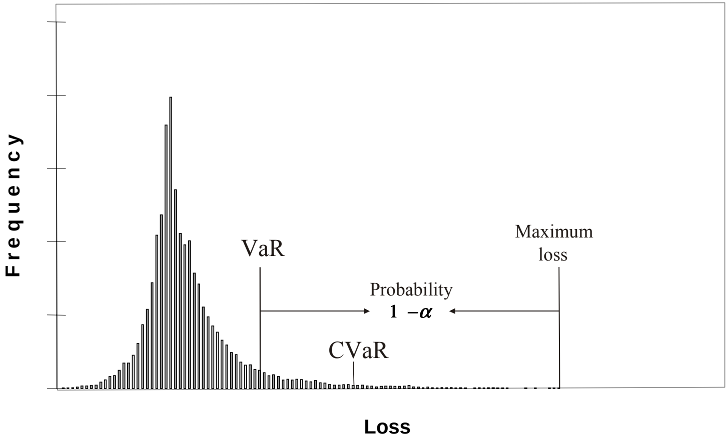
Dynamic optimization

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.

CVaR constraints: mean excess dose (radiotherapy)



Move mean of tail to the left!

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

References

S. C. Billups and M. C. Ferris.
Solutions to affine generalized equations using proximal mappings.
Mathematics of Operations Research, 24:219–236, 1999.

M. C. Ferris, S. P. Dirkse, J.-H. Jagla, and A. Meeraus.
An extended mathematical programming framework.
Computers and Chemical Engineering, 33:1973–1982, 2009.

M. C. Ferris, S. P. Dirkse, and A. Meeraus.
Mathematical programs with equilibrium constraints: Automatic reformulation and solution via constrained optimization.
In T. J. Kehoe, T. N. Srinivasan, and J. Whalley, editors, *Frontiers in Applied General Equilibrium Modeling*, pages 67–93. Cambridge University Press, 2005.

M. C. Ferris and T. S. Munson.
Complementarity problems in GAMS and the PATH solver.
Journal of Economic Dynamics and Control, 24:165–188, 2000.

R. T. Rockafellar.