Mathematical Models & Optimization?

Mankind always sets itself only such problems as it can solve...
-Karl Marx (1859)

Why build them?
• To study real-world phenomena where analytic techniques alone don’t work!
• To investigate relationships among parameters affecting the functioning of complex processes

Where does optimization come in?
• The model is not usually an end in itself
• Model contains free parameters whose values have to be determined to produce an optimum measure of “goodness”
The Data Modeling Process

... who can tell
Which of her forms has shown her substance right?
- William Butler Yeats, in A Bronze Head (1933)

Gather DATA

Create alternative MODELS

Fit each MODEL to the DATA

Gather more data

Create new models

Assign preferences to the alternative MODELS

Choose what data to gather next

Choose future actions

Decide whether to create new models

Choose future actions
Example

Phenomenon:
Residential demand for electricity

Influencing factors:
• Prices of electricity & competing energy sources
• Socio-economic indicators (e.g. income)
• Weather indicators (e.g. heating degree days)

Possible models:
• Econometric (linear or nonlinear)
• End-Use Model
• Artificial Neural Network
Example from your world

Phenomenon: Runoff

Influencing factors:
• Slope
• Cover
• Rainfall

Possible models:
• Linear model
• Nonlinear model
• Artificial Neural Network
Linear Model:

Runoff \( (x, y, t) = \alpha + \beta_1 f_1 (\text{slope}(x, y)) + \beta_2 f_2 (\text{cover}(x, y)) \\
+ \beta_3 f_3 (\text{rainfall}(x, y)) \\
+ \beta_4 f_4 (\text{slope}(x, y), \text{cover}(x, y)) + \ldots + \varepsilon \)

where,

\( x \): position on the east - west coordinate

\( y \): position on the north - south coordinate

\( t \): point in the time horizon

\( f_j \): any function of the inputs: slope, cover and rainfall

\( \alpha, \beta_i \): model parameters

\( \varepsilon \): error term with mean zero and variance \( \sigma^2 \)
Given that the linear model is the right model to account for runoff:

Values for slope, cover, rainfall and the corresponding observes runoff are collected at different (x,y) locations and over time

Model parameters are estimated to minimize the discrepancies between the predicted runoff (model output) and the observed runoff over the collected sample
This is what you call “modern calibration”
This estimation problem could be formulated as:

\[
\begin{align*}
\text{Find } & \alpha, \beta_1, \beta_2, \ldots, \beta_m \\
\text{Minimizing } & \lambda(\text{observed runoff, predicted runoff}) \\
\text{Subject to possible constraints on slope, cover, and rainfall}
\end{align*}
\]

where,
\[
\lambda: \text{ some measure of discrepancy (e.g. Least - Squares)}
\]
The Optimization Problem

Min $f(x)$

$x \in R^N$

s.t.

$g_i(x) \leq 0 \quad i = 1, 2, \ldots, l'$

$g_i(x) = 0 \quad i = l', l' + 1, \ldots, l$
Mapping the jargon

- **Function being optimized** → **Objective function**
- **Model parameters** → **Decision variables**
- **Constraints on model parameters** → **constraints**
- **Simple constraints on** → **simple bounds or**
- **The range of model parameters** → **box constraints**

Objective function describes a **response surface**

Intersection of constraints in the decision variable space

Defines the **feasible region** or the **search space**
Let's take a function:

\[ f(x_1, x_2) = \frac{1}{2} \sum_{k=1}^{2} x_k^4 - 16x_k^2 + 5x_k \]
Classification of optimization problems

Properties of $f(x)$

*Functional form exists:*

- Linear
- Sum of squares of linear functions
- Quadratic function
- Sum of squares of linear functions
- Quadratic function
- Sum of squares of nonlinear functions
- Smooth nonlinear function
- Non-smooth nonlinear function

*Unimodal:*

The exists only one optimum

*Multimodal:*

There exist more than one optimum at different objective function values

*Functional form does not exist:*

Output of a black box
## Classification of optimization problems Cont.

<table>
<thead>
<tr>
<th>Properties of $g_i(x)$</th>
<th>Nature of $x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>No constraints</td>
<td>Continuous</td>
</tr>
<tr>
<td>Simple bounds</td>
<td>Discrete</td>
</tr>
<tr>
<td>Linear functions</td>
<td>Mixed</td>
</tr>
<tr>
<td>Sparse linear functions</td>
<td></td>
</tr>
<tr>
<td>Smooth nonlinear functions</td>
<td></td>
</tr>
<tr>
<td>Sparse nonlinear functions</td>
<td></td>
</tr>
<tr>
<td>Non-smooth non-linear functions</td>
<td></td>
</tr>
</tbody>
</table>
Stopping Criteria: Necessary conditions for a minimum at $x^*$

In the univariate case:
- $f'(x^*) = 0$, and
- $f''(x^*) \geq 0$

In the multivariate case:
- $\|g(x^*)\| = 0$
- $H(x^*)$ is p.s.d.
What if no gradient info. Is available?
• How hard we want to work on the problem (e.g. computing time)
• No improving for the last $N_6$ iterations

What if there are more than one minimum?
• Only under restricted conditions can we guarantee the “global” minimum!
Overview of existing solution methods

<table>
<thead>
<tr>
<th>Local Search Methods</th>
<th>Global Search Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e.g. gradient descent)</td>
<td>Deterministic</td>
</tr>
<tr>
<td></td>
<td>Probabilistic</td>
</tr>
</tbody>
</table>
More on probabilistic global search methods

Possibilities:

- Tabu Search
- Simulated Annealing
- Genetic Algorithms
Stimulated Annealing
(kirkpatrick, Gelatt, and Vecchi, 1983)

- Motivated by the behavior of systems in thermal equilibrium at a finite temperature (statistical mechanics)

- Was initially designed for optimization problems with discrete decision variables. Extensions to continuous problems have been introduced (e.g. Goffe et al., 1994)

- Allows non improving moves with a non-zero probability
General outline of Algorithm
(for a minimization problem)

Pick an initial point \( x^0 \) & initial “temperature” \( T^0 \), \( k = 0 \), \( n = 0 \), set \( x^* = x^0 \)

Loop until CONVERGENCE

1. Generate randomly \( x^{k+1} \) in the neighborhood of \( x^k \)
2. If \( f(x^{k+1}) < f(x^k) \), accept \( x^{k+1} \)
   
   Else (use the metropolis criterion) \( = \frac{f(x^{KH}) - f(x^K)}{T^n} \)

   Accept \( x^{k+1} \) with probability \( p = e \)
   
   O.W. \( x^{k+1} = x^k \)
3. If \( f(x^{k+1}) < f(x^\triangledown) \) then set \( x^\triangledown = x^{k+1} \)
4. Increment \( K \)
5. After each \( N \) moves, adjust temp. \( T^{n+1} = \sigma \cdot T^n \) or \( < \sigma < 1 \)

END LOOP

Report \( x^\triangledown \)

Cooling schedule
Important issues:

• Choice of Initial temperature and cooling schedule
• Stopping criteria
• Diversification Vs. intensification of search
Where can you go for more info?

- **Literature?**
  - Numerous books & textbooks on optimization and mathematical modeling (not as much on global optimization!)
  - A host of publications. Check journals by INFORMS and the European Society of Operational Research

- **Software packages & computer code on the net**
  - Cplex for linear programming, GAMS as an interface to a variety of linear and nonlinear solvers (available on ECN)
  - Libraries like IMSL
  - Various codes written for SA & GAs available free of charge on the internet

- **Resources on the internet**
  - Do a keyword search and you’ll locate quite a few links.