Research Challenges in Process Systems Engineering and Overview of Mathematical Programming

Ignacio E. Grossmann
Center for Advanced Process Decision-making (CAPD)
Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
Questions:

1. What is the role of Process Systems Engineering in “commodity” industry vs. “new emerging” technologies?

Value preservation vs. Value creation

2. What is the future scope for fundamental contributions in Process Systems Engineering?

Science vs. Engineering

3. What are Research Challenges in Process Systems Engineering?

Carnegie Mellon
Changes in the Chemical Industry

From Marcinowski (2006)

<table>
<thead>
<tr>
<th>Sector</th>
<th>In the 80s</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oil and Gas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Petrochemicals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commodity Chemicals</td>
<td>Hoechst</td>
<td>BASF, Formosa</td>
</tr>
<tr>
<td>Specialty Chemicals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agrochemicals/Biotechnology</td>
<td>Bayer</td>
<td>BASF, Bayer Mat.Sc., ICI</td>
</tr>
<tr>
<td>Pharma</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The diagram illustrates changes in the chemical industry from the 1980s to 2006, showing the evolution of major players in different sectors.
## Economics of Chemical Enterprise

### Value preservation vs. Value Creation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ExxonMobil</td>
<td>$208.7</td>
<td>$237</td>
<td>$359</td>
</tr>
<tr>
<td>ChevronTexaco</td>
<td>99.7</td>
<td>120</td>
<td>193</td>
</tr>
<tr>
<td>Dow</td>
<td>27.8</td>
<td>32.6</td>
<td>46.3</td>
</tr>
<tr>
<td>DuPont</td>
<td>26.8</td>
<td>30.2</td>
<td>25.3</td>
</tr>
<tr>
<td>Procter &amp; Gamble</td>
<td>39.2</td>
<td>43.4</td>
<td>56.7</td>
</tr>
<tr>
<td>Johnson &amp; Johnson</td>
<td>32.3</td>
<td>41.8</td>
<td>50.5</td>
</tr>
<tr>
<td>Merck</td>
<td>21.2</td>
<td>22.4</td>
<td>22.0</td>
</tr>
<tr>
<td>Bristol-Myers Squibb</td>
<td>21.7</td>
<td>18.6</td>
<td>19.2</td>
</tr>
<tr>
<td>Amgen</td>
<td>4.0</td>
<td>8.4</td>
<td>12.4</td>
</tr>
<tr>
<td>Genentech</td>
<td>1.7</td>
<td>3.3</td>
<td>5.5</td>
</tr>
</tbody>
</table>

---

Carnegie Mellon
Move from Engineering to Science
## Distribution Research Papers AIChE Journal 2005

<table>
<thead>
<tr>
<th>Topic</th>
<th>Papers</th>
<th>U.S.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluid Mechanics, Transport Phenomena</td>
<td>57</td>
<td>18</td>
</tr>
<tr>
<td>Reactors, Kinetics, Catalysis</td>
<td>56</td>
<td>21</td>
</tr>
<tr>
<td>Process Systems Engineering</td>
<td>51</td>
<td>22</td>
</tr>
<tr>
<td>Separations</td>
<td>37</td>
<td>11</td>
</tr>
<tr>
<td>Particle Technology, Fluidization</td>
<td>28</td>
<td>11</td>
</tr>
<tr>
<td>Thermodynamics</td>
<td>19</td>
<td>2</td>
</tr>
<tr>
<td>Materials, Interfaces, Electrochemical Phenomena</td>
<td>21</td>
<td>9</td>
</tr>
<tr>
<td>Environmental, Energy Engineering</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Bioeng., Food, Natural Products</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>294</td>
<td>107</td>
</tr>
</tbody>
</table>

=> Only 36% papers involve US-based authors

Only 15% from top 25 U.S. schools
Observations

Trade-offs: Value preservation vs. Value growth
Chemicals/Fuels vs. Pharmaceutical/Biotechnology

Research trend away from Chemical Engineering
Science vs Engineering

Major real world challenges
Globalization, energy, environment, health

=> Need expand scope of Process Systems Engineering

Carnegie Mellon
Expanding the Scope of Process Systems Engineering

(Grossmann & Westerberg, 2000; Marquardt et al, 1998)
Major Research Challenges

I. Product and Process Design

II. Energy and Sustainability

III. Enterprise-wide Optimization
What is science base for PSE?

Numerical analysis => Simulation

Mathematical Programming => Optimization

Systems and Control Theory => Process Control

Computer Science => Advanced Info./Computing

Management Science => Operations/Business

*Math Programming & Control Theory “competitive” advantage*

Carnegie Mellon
Motivation Math Programming

Motivation

Design, operations and control problems involve decision-making over a large number of alternatives for selecting “best” solution
- Configurations (discrete variables)
- Parameter values (continuous variables)

Challenges:

How to model optimization problems?
How to solve large-scale models?
How to avoid local solutions?
How to handle uncertainties?
Given a space of alternatives that are specified through constraints in a mathematical model, select decision variables to optimize an objective function:

$$\min Z = f(x, y)$$

subject to:

$$h(x, y) = 0$$

$$g(x, y) \leq 0$$

$$x \in \mathbb{R}^n, \quad y \in \{0,1\}^m$$

**MINLP:** Mixed-integer Nonlinear Programming Problem

**Major challenges:**
- **Combinatorics:** scalability (NP-hard)
- **Nonconvexities:** global optima
Classical Optimization

\[ \min Z = f(x) \]
\[ x \in \mathbb{R}^n \]

Newton (1664)

\[ \min Z = f(x) \]
\[ s.t. \quad h(x) = 0 \]
\[ x \in \mathbb{R}^n \]

Lagrange (1778)

Solution of inequalities
Solution of linear equations
Solution of inequality systems

Fourier (1826)
Gauss (1826)
Farkas (1902)

Carnegie Mellon
Modern Optimization

Linear Programming  Kantorovich (1939), Dantzig (1947)

Nonlinear Programming  Karush (1939), Kuhn, A.W.Tucker (1951)

Integer Programming  R. E. Gomory (1958)
Evolution of Mathematical Programming

*Egon Balas: Preface to Minoux (1983)*

1950's  Linear Programming  
         Nonlinear Programming  

1960's  Network Theory  
         Integer Programming  
         Dynamic Programming  

1970's  Nondifferentiable Optimization  
         Combinatorial Optimization: Graph Theory  
         Theory Computational Complexity  

1980’s  Interior Point Methods *Karmarkar (1984)*  

1990’s  Convexification of Mixed-Integer Linear Programs  
         *Balas, Ceria, Cornuejols (1993)*  

2000’s  **MINLP**  
         Global Optimization  
         Logic-based optimization  
         Search techniques *(tabu, genetic algorithms)*  
         Hybrid-systems  

**Computational progress: much faster algorithms/much faster computers**
Progress in Linear Programming

Increase in computational speed from 1987 to 2002

_Bixby-ILOG (2002)_

For 50,000 constraint LP model

Algorithms

Primal simplex in 1987 (XMP) versus

Best(primal,dual,barrier) 2002 (CPLEX 7.1) 2400x

Machines

Sun 3/150

Pentium 4, 1.7GHz 800x

Net increase: Algorithm * Machine ~ 1 900 000x

_Two million-fold increase in speed!!_

_Carnegie Mellon_
Applications of Math. Programming in Chemical Engineering

Process Design

Process Synthesis

Production Planning

Process Scheduling

Supply Chain Management

Process Control

Parameter Estimation

LP, MILP, NLP, MINLP, Optimal Control

Carnegie Mellon
Contributions by Chemical Engineers to Mathematical Programming

Large-scale nonlinear programming
SQP algorithms
Interior Point algorithms

Optimal control problems
NLP-based strategies

Mixed-integer nonlinear programming
Outer-approximation algorithm
Extended-Cutting Plane Method
Generalized Disjunctive Programming

Global optimization
α-Branch and Bound
Spatial branch and bound methods

Optimization under Uncertainty
Sim-Opt
Parametric programming
Mathematical Programming

\[
\begin{align*}
\text{min } f(x, y) & \quad \text{Cost} \\
\text{s.t. } h(x, y) &= 0 \quad \text{Process equations} \\
g(x, y) &\leq 0 \quad \text{Specifications} \\
x &\in X \quad \text{Continuous variables} \\
y &\in \{0,1\} \quad \text{Discrete variables}
\end{align*}
\]

Continuous optimization

- Linear programming: \textit{LP}
- Nonlinear programming: \textit{NLP}

Discrete optimization

- Mixed-integer linear programming: \textit{MILP}
- Mixed-integer nonlinear programming: \textit{MINLP}
Modeling systems

Mathematical Programming

**GAMS** *(Meeraus et al, 1997)*

**AMPL** *(Fourer et al., 1995)*

**AIMSS** *(Bisschop et al. 2000)*

1. Algebraic modeling systems => pure equation models
2. Indexing capability => large-scale problems
3. Automatic differentiation => no derivatives by user
4. Automatic interface with
   LP/MILP/NLP/MINLP solvers
Linear Programming

LP: Algorithms:
    Simplex  (Dantzig, 1949; Kantorovich, 1938)
    Interior Point (Karmarkar, 1988, Marsten at al, 1990)

Major codes:
    CPLEX (ILOG)    (Bixby)
    XPRESS (Dash Optimization)   (Beale, Daniel)
    OSL (IBM)     (Forrest, Tomlin)

Simplex:  *up to 100,000 rows (constraints), 1,000,000 vars*

Interior Point:
    *up to 1,000,000 rows (constraints), 10,000,000 vars*
    *typically 20-40 Newton iterations regardless size*
    *Only limitation very large problems >500,000 constr*
MILP

\[
\begin{align*}
\text{min } Z &= a^T y + b^T x \\
\text{st } &\quad Ay + Bx \leq d \\
&\quad y \in \{0,1\}^m, \ x \geq 0
\end{align*}
\]

Objective function

Constraints

Theory for Convexification

Lovacz & Schrijver (1989), Sherali & Adams (1990),
Balas, Ceria, Cornuejols (1993)

Branch and Bound

Beale (1958), Balas (1962), Dakin (1965)

Cutting planes

Gomory (1959), Balas et al (1993) LP (simplex) based

Branch and cut

Johnson, Nemhauser & Savelsbergh (2000)

"Good" formulation crucial! \(\Rightarrow\) Small LP relaxation gap

Drawback: exponential complexity

Carnegie Mellon
Modeling with MILP  

**Note: linear constraints**

1. **Multiple choice**
   - *At least one*\[\sum_{i \in I} y_i \geq 1\]
   - *Exactly one*\[\sum_{i \in I} y_i = 1\]
   - *At most one*\[\sum_{i \in I} y_i \leq 1\]

2. **Implication**
   - If select i then select k\[y_i - y_k \leq 0\]
   - Select i if and only if select k\[y_i - y_k = 0\]

3. **Integer numbers**
\[n = \sum_{k=1}^{N} ky_k, \quad \sum_{k=1}^{N} y_k = 1\]  
also\[n = \sum_{k=1}^{M} 2^k y_k\]  
Fewer 0-1 variables  
Weaker relaxation

**Carnegie Mellon**
Discontinuous Functions/Domains

a) **Domain**

\[ x = \begin{cases} 
0 & \text{IF } y = 0 \\
L \leq x \leq U & \text{IF } y = 1 
\end{cases} \]

\[ 0 \quad L \quad y = 1 \quad U \]
\[ y = 0 \]

**MIXED-INTEGER MODEL**

\[ L_y \leq x \leq U_y \]
\[ y = 0, 1 \]

b) **Function**

\[ C = \begin{cases} 
0 & \text{IF } y = 0 \\
\alpha + \beta x & \text{IF } y = 1 
\end{cases} \]

**MIXED-INTEGER MODEL**

\[ C = \alpha y + \beta x \]
\[ 0 \leq x \leq U_y \]
\[ y = 0, 1 \]
Simple Minded Approaches

**Exhaustive Enumeration**

SOLVE LP’S FOR ALL 0-1 COMBINATIONS ($2^m$)

- IF $m = 5$ 32 COMBINATIONS
- IF $m = 100$ $10^{30}$ COMBINATIONS
- IF $m = 10,000$ $10^{3000}$ COMBINATIONS

**Relaxation and Rounding**

SOLVE MILP WITH $0 \leq y \leq 1$

If solution not integer round closest

RELAXATION

Only special cases yield integer optimum (*Assignment Problem*)

Relaxed LP provides **LOWER BOUND** to MILP solution

Difference: Relaxation gap

**Carnegie Mellon**
ROUNDING
May yield infeasible or suboptimal solution
Convert MIP into a Continuous NLP

*Example:* Min $Z = y_1 + 2y_2$

s.t. $2y_1 + y_2 \geq 1$

$y_1 = 0, 1 \quad y_2 = 0, 1$

replace $0 - 1$ conditions by

$0 \leq y_1 \leq 1, \quad y_1 (1-y_1) \leq 0$  =>  Nonlinear  only feasible pts.

$0 \leq y_2 \leq 1, \quad y_1 (1-y_2) \leq 0$  =>  Nonconvex!

(1,0)
(0,1)
(1,1)

Using CONOPT2

st. point  $y_1 = 0, \quad y_2 = 0$  =>  infeasible

st. pt.  $y_1 = 0.5, \quad y_2 = 0.5$  =>  $y_1 = 0 \quad y_2 = 1$  $Z = 2$  suboptimal

correct solution  $y_1 = 1, \quad y_2 = 0$  $Z = 1$
Branch and Bound

Tree Enumeration
Solve LP At Each Node

\[
\begin{align*}
\text{Min} & \quad y_1 + 2y_2 \\
\text{s.t.} & \quad 2y_1 + y_2 \geq 1 \quad (P) \\
\end{align*}
\]

\[
y_1 = 0, 1 \quad y_2 = 0, 1
\]

Solve MILP with
\[
\begin{align*}
0 \leq y_1 \leq 1 & \quad y_1 = 0.5 \quad Z = 0.5 \quad \text{Lower Bound} \\
0 \leq y_2 \leq 1 & \quad y_2 = 0
\end{align*}
\]

Fix \( y_1 = 0 \)

\( y_2 = 1 \quad Z = 2 \)

Fix \( y_1 = 1 \)

\( y_2 = 0 \quad Z = 1 \quad \text{OPTIMUM} \)
Major Solution Approaches MILP

I. Enumeration

Branch and bound

Land, Doig (1960) Dakin (1965)

Basic idea: partition successively integer space to determine whether subregions can be eliminated by solving relaxed LP problems

II. Convexification

Cutting planes


Basic idea: solve sequence relaxed LP subproblems by adding valid inequalities that cut-off previous solutions

Remark

- Branch and bound most widely used
- Recent trend to integrate it with cutting planes

BRANCH-AND-CUT

Carnegie Mellon
Branch and Bound
Partitioning Integer Space Performed with Binary Tree

Note: 15 nodes for $2^3=8$ 0-1 combinations

Node $k$ descendent node $\ell$
NODE k: LP

\[ \min Z = c^T x + b^T y \]
\[ \text{s.t.} \quad Ax + By \leq d \]
\[ x \geq 0 \quad 0 \leq y \leq 1 \]
\[ y_i = 0 \text{ or } 1 \quad i \in I_k \]

Since node k descendent of node \( \ell \)

1. IF LP\( \ell \) INFEASIBLE THEN LP\( k \) INFEASIBLE

2. IF LP\( k \) FEASIBLE \( Z \ell \leq Z^k \)
   monotone increase objective
   \( Z \ell : \) LOWER BOUND

3. IF LP\( k \) INTEGER \( Z^k \leq Z^* \)
   \( Z^k : \) UPPER BOUND

**FATHOMING RULES:** If node is infeasible
   If Lower Bound exceeds Upper Bound

*Carnegie Mellon*
"If sufficient care is exercised, it is now possible to solve MILP models of size approaching ‘large’ LP’s. Note, however, that ‘sufficient care’ is the operative phrase”.

JOHN TOMLIN (1983)

HOW TO MODEL INTEGER CONSTRAINTS?
Propositional Logic
Disjunctions
Mathematical Modeling of Boolean Expressions

Williams (1988)

LITERAL IN PROPOSITIONAL LOGIC  \( P_i \)  TRUE
NEGATION  \( \neg P_i \)  FALSE

Example  \( P_i \): select unit i, execute task j

PROPOSITION: set of literals  \( P_i \) separated by **OR, AND, IMPLICATION**

**Representation Linear 0-1 Inequalities**

ASSIGN binary  \( y_i \) to  \( P_i \)   \( (1 - y_i) \) to  \( \neg P_i \)

**OR**  \( P_1 \lor P_2 \lor \ldots \lor P_r \)  \( y_1 + y_2 + \ldots + y_r \geq 1 \)

**AND**  \( P_1 \land P_2 \land \ldots \land P_r \)  \( y_1 \geq 1, y_2 \geq 1, \ldots y_r \geq 1 \)

**IMPLICATION**  \( P_1 \Rightarrow P_2 \)

**EQUIVALENT TO**  \( \neg P_1 \lor P_2 \)  \( 1 - y_1 + y_2 \geq 1 \)

\[ \text{OR} \quad y_2 \geq y_1 \]

Carnegie Mellon
Systematic Procedure to Derive Linear Inequalities for Logic Propositions

Goal is to Convert Logical Expression into Conjunction Normal Form (CNF)

\[ Q_1 \land Q_2 \land \ldots \land Q_s \]

where clause \( Q_i : P_1 \lor P_2 \lor \ldots \lor P_r \) (Note: all OR)

**BASIC STEPS**

1. REPLACE IMPLICATION BY DISJUNCTION

\[ P_1 \implies P_2 \iff \neg P_1 \lor P_2 \]

2. MOVE NEGATION INWARD APPLYING DE MORGAN’S THEOREM

\[ \neg (P_1 \land P_2) \iff \neg P_1 \lor \neg P_2 \]

\[ \neg (P_1 \lor P_2) \iff \neg P_1 \land \neg P_2 \]

3. RECURSIVELY DISTRIBUTE OR OVER AND

\[ (P_1 \land P_2) \lor P_3 \iff (P_1 \lor P_3) \land (P_2 \lor P_3) \]
EXAMPLE
flash ⇒ dist ∨ abs
memb ⇒ not abs ∧ comp

PF ⇒ PD ∨ PA  (1)
PM ⇒ ¬ PA ∧ PC  (2)

(1) ¬ PF ∨ PD ∨ PA remove implication

1 − yF + yD + yA ≥ 1
yD + yA ≥ yF

(2) ¬ PM ∨ (¬ PA ∧ Pc) remove implication
(¬ PM ∨ ¬ PA) ∧ (¬ PM ∨ Pc) distribute OR over AND => CNF!

1 − yM + 1 − yA ≥ 1  1 − yM + yC ≥ 1
yM + yA ≤ 1  yC ≥ yM

Verify:

yF = 1  yD + yA ≥ 1  yF = 0  yD + yA ≥ 0
yM = 1 ⇒ yA = 0  yC = 1

Carnegie Mellon
EXAMPLE

Integer Cut

Constraint that is infeasible for integer point

\[ y_i = 1 \quad i \in B \quad y_i = 0 \quad i \in N \]

and feasible for all other integer points

*Balas and Jeroslow (1968)*
Example: Multiperiod Problems

“If Task $y_i$ is performed in any time period $i = 1, ..n$ select Unit $z$”

**Intuitive Approach**

$$y_1 + y_2 + \ldots + y_n \leq n*z \quad (1)$$

**Logic Based Approach**

$$y_1 \lor y_2 \lor \ldots \lor y_n \Rightarrow z$$

$$\neg(y_1 \lor y_2 \lor \ldots \lor y_n) \lor z$$

$$\neg y_1 \lor \neg y_2 \lor \ldots \lor \neg y_n \lor z$$

$$\neg y_1 \lor z \land \neg y_2 \lor z \land \ldots \land \neg y_n \lor z$$

$$1 - y_1 + z \geq 1 \quad 1 - y_2 + z \geq 1 \quad 1 - y_n + z \geq 1$$

\[\begin{align*}
  y_1 & \leq z \\
  y_2 & \leq z \\
  \vdots \\
  y_n & \leq z
\end{align*}\]

**Inequalities in (2) are stronger than inequalities in (1)**
Geometrical interpretation

\[ y_1 \leq z \]
\[ y_2 \leq z \]

All extreme points in hypercube are integer!
Geometrical interpretation

\[ y_1 + y_2 \leq 2z \]

Non-integer extreme points
Weaker relaxation!
Modeling of Disjunctions

\[ \bigvee_{i \in D} A_i x \leq b_i \] one inequality must hold

Example: A before B \textit{OR} B before A

\[ [TS_A + pt_A \leq TS_B] \lor [TS_B + pt_B \leq TS_A] \]

**Big M Formulation**

\[ A_i x \leq b_i + M_i (1 - y_i) \quad i \in D \]

\[ \sum_{i \in D} y_i = 1 \]

**Difficulty: Parameter \( M_i \)**

Must be sufficiently large to render inequality redundant

Large value yields poor relaxation

Carnegie Mellon
Convex-hull Formulation  (Balas, 1985)

\[
x = \sum_{i \in D} z_i \quad \text{disaggregation vars.}
\]

\[
A_i z_i \leq b_i y_i \quad i \in D
\]

\[
\sum_{i \in D} y_i = 1
\]

\[
0 \leq z_i \leq U y_i \quad i \in D \quad \text{(may be removed)}
\]

\[
y_i = 0, 1
\]

**Derivation**

\[
A_i \ x y_i \leq b_i y_i \quad i \in D \quad \text{(B) \ nonlinear disj. equiv.}
\]

\[
\sum_{i \in D} y_i = 1
\]
Let $z_i = x y_i$ be the disaggregated variable

$$\sum_{i \in D} z_i = \sum_{i \in D} x y_i = x \sum_{i \in D} y_i$$

since $\sum_{i \in D} y_i = 1 \Rightarrow \sum_{i \in D} z_i = x$ \hspace{1cm} (A)

to ensure $z_i = 0$ if $y_i = 0$

$$0 \leq z_i \leq U y_i$$ \hspace{1cm} (C)

\begin{align*}
(A) & \Rightarrow x = \sum_{i \in D} z_i \\
& \text{subst. (B)} \hspace{1cm} A_i z_i \leq b_i y_i \hspace{1cm} i \in D \\
& \sum_{i \in D} y_i = 1 \\
(C) & \Rightarrow 0 \leq z_i \leq U y_i \hspace{1cm} i \in D
\end{align*}
Example

\[ x_1 - x_2 \leq -1 \] \lor \ [ -x_1 + x_2 \leq -1 ]

\[ 0 \leq x_1, x_2 \leq 4 \]

**big M**

\[ x_1 - x_2 \leq -1 + M (1 - y_1) \]
\[ -x_1 + x_2 \leq -1 + M (1 - y_2) \]
\[ y_1 + y_2 = 1 \]
\[ M = 10 \text{ possible choice} \]

Convex hull

\[ x_1 = z_1^1 + z_1^2 \]
\[ x_2 = z_2^1 + z_2^2 \]
\[ z_1^1 - z_2^1 \leq -y_1 \]
\[ -z_1^2 + z_2^2 \leq -y_2 \]
\[ y_1 + y_2 = 1 \]
\[ 0 \leq z_1^1 \leq 4 y_1 \]
\[ 0 \leq z_2^1 \leq 4 y_2 \]
\[ 0 \leq z_1^2 \leq 4 y_1 \]
\[ 0 \leq z_2^2 \leq 4 y_2 \]
Nonlinear Programming

NLP: Algorithms

(variants of Newton's method for solving KKT conditions)

- Successive quadratic programming (SQP) (Han 1976; Powell)
- Reduced gradient
- Interior Point Methods

Major codes:

- MINOS (Murtagh, Saunders, 1978, 1982)
- CONOPT (Drud, 1994)
- SQP: SNOPT (Murray, 1996) OPT (Biegler, 1998)
- IP: IPOPT (Wachter, Biegler, 2002) www.coin-or.org

Typical sizes: 50,000 vars, 50,000 constr. (unstructured)
500,000 vars (few degrees freedom)

Convergence: Good initial guess essential (Newton's)

Nonconvexities: Local optima, non-convergence

Carnegie Mellon
Mixed-Integer Nonlinear Programming

\[
\begin{align*}
\min Z &= f(x, y) \\
\text{s.t.} & \quad g(x, y) \leq 0 \\
& \quad x \in X, \ y \in Y \\
X &= \{x \mid x \in \mathbb{R}^n, x^l \leq x \leq x^u, Bx \leq b\} \\
Y &= \{y \mid y \in \{0,1\}^m, Ay \leq a\}
\end{align*}
\]

- \( f(x,y) \) and \( g(x,y) \) - assumed to be convex and bounded over \( X \).
- \( f(x,y) \) and \( g(x,y) \) commonly linear in \( y \)

Carnegie Mellon
Solution Algorithms

- **Branch and Bound method (BB)**
  Branch and cut: Stubbs and Mehrotra (1999)

- **Generalized Benders Decomposition (GBD)**
  Geoffrion (1972)

- **Outer-Approximation (OA)**

- **LP/NLP based Branch and Bound**
  Quesada and Grossmann (1992)

- **Extended Cutting Plane (ECP)**
  Westerlund and Pettersson (1995)
Basic NLP subproblems

a) NLP Relaxation *Lower bound*

\[
\min Z^k_{LB} = f(x, y) \\
\text{s.t. } g_j(x, y) \leq 0 \quad j \in J \\
x \in X, y \in Y_R \\
y_i \leq \alpha_i^k \quad i \in I^k_{FL} \\
y_i \geq \beta_i^k \quad i \in I^k_{FU} 
\]  
(NLP1)

b) NLP Fixed \(y^k\) *Upper bound*

\[
\min Z^k_U = f(x, y^k) \\
\text{s.t. } g_j(x, y^k) \leq 0 \quad j \in J \\
x \in X 
\]  
(NLP2)

c) Feasibility subproblem for fixed \(y^k\).

\[
\min u \\
\text{s.t. } g_j(x, y^k) \leq u \quad j \in J \\
x \in X, u \in R^1 
\]  
(NLPF)

*Infinity-norm*

Carnegie Mellon
Cutting plane MILP master
(Duran and Grossmann, 1986)

Based on solution of K subproblems \((x^k, y^k) \ k=1,...K\)

**Lower Bound**

M-MIP

\[
\min Z_L^K = \alpha \\
\text{st} \quad \alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\
g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \quad j \in J \\
\]

\(x \in X, y \in Y\)

**Notes:**

a) Point \((x^k, y^k) \ k=1,...K\) normally from NLP2

b) Linearizations *accumulated* as iterations K increase

c) Non-decreasing sequence *lower bounds*
Linearizations and Cutting Planes

Convex Objective

Underestimate Objective Function

Convex Feasible Region

Overestimate Feasible Region
Branch and Bound

**NLP1:**
\[
\min \ Z_{LB}^k = f(x, y)
\]

s.t.
\[
g_j(x, y) \leq 0 \quad j \in J
\]
\[
x \in X, \quad y \in Y_R
\]
\[
y_i \leq \alpha_i^k \quad i \in I_{FL}^k
\]
\[
y_i \geq \beta_i^k \quad i \in I_{FU}^k
\]

Successive solution of NLP1 subproblems

**Advantage:**
Tight formulation may require one NLP1 ($I_{FL}=I_{FU}=$ \(\emptyset\))

**Disadvantage:**
Potentially many NLP subproblems

**Convergence global optimum:**
Uniqueness solution NLP1 *(sufficient condition)*

\[\text{Less stringent than other methods}\]
**Outer-Approximation**

Alternate solution of NLP and MIP problems:

NLP2: \[
\min Z_U^k = f(x, y^k)
\]

s.t. \[
g_j(x, y^k) \leq 0 \quad j \in J
\]
\[x \in X\]

M-MIP: \[
\min Z_L^k = \alpha
\]

s.t. \[
\begin{align*}
\alpha & \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\
g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} & \leq 0 \quad j \in J^k
\end{align*}
\]
\[x \in X, \quad y \in Y\]

*Property.* Trivially converges in **one iteration** if \(f(x,y)\) and \(g(x,y)\) are **linear**

- If infeasible NLP solution of feasibility NLP-F required to guarantee convergence.
Generalized Benders Decomposition

Benders (1962), Geoffrion (1972)

Particular case of Outer-Approximation as applied to (P1)

1. Consider Outer-Approximation at \((x^k, y^k)\)

\[
\alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix}
\]

\[
g(x^k, y^k) + \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \quad j \in J_k
\]

2. Obtain linear combination of (1) using Karush-Kuhn-Tucker multipliers \(\mu^k\) and eliminating \(x\) variables

\[
\alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T (y - y^k)
\]

\[
+ (\mu^k)^T \left[ g(x^k, y^k) + \nabla g(x^k, y^k)^T (y - y^k) \right]
\]

\[\text{Lagrangian cut}\]

Remark. Cut for infeasible subproblems can be derived in a similar way.

\[
(\lambda^k)^T \left[ g(x^k, y^k) + \nabla g(x^k, y^k)^T (y - y^k) \right] \leq 0
\]
Generalized Benders Decomposition

Alternate solution of NLP and MIP problems:

NLP2: \[
\min_{x, y} Z_u = f(x, y^k) \\
\text{s.t. } g_j(x, y^k) \leq 0 \quad j \in J \\
x \in X
\]

M-GBD: \[
\min \alpha \quad Z^k_L = \alpha \\
\text{s.t. } \alpha \geq f(x^k, y^k) + \nabla_y f(x^k, y^k)^T (y - y^k) \\
\quad + \left( \mu^k \right)^T \left[ g(x^k, y^k) + \nabla_y g (x^k, y^k)^T (y - y^k) \right] \quad k \in KFS \\
\quad \left( \lambda^k \right)^T \left[ g(x^k, y^k) + \nabla_y g (x^k, y^k)^T (y - y^k) \right] \leq 0 \quad k \in KIS \\
\quad y \in Y, \quad \alpha \in R^l
\]

**Property 1.** If problem (P1) has zero integrality gap, Generalized Benders Decomposition converges in one iteration when optimal \((x^k, y^k)\) are found.

=> Also applies to Outer-Approximation

*Sahinidis, Grossmann (1991)*
Extended Cutting Plane

Westerlund and Pettersson (1995)

Add linearization most violated constraint to M-MIP

\[ J^k = \{ \hat{j} \in \arg \{ \max_{j \in J} g_j(x^k, y^k) \} \} \]

Remarks.

- Can also add full set of linearizations for M-MIP
- Successive M-MIP's produce non-decreasing sequence lower bounds
- Simultaneously optimize \( x^k, y^k \) with M-MIP

\[ \Rightarrow \text{Convergence may be slow} \]
LP/NLP Based Branch and Bound (*Branch & Cut*)
Quesada and Grossmann (1992)

Integrate NLP and M-MIP problems

Remark.
Fewer number branch and bound nodes for LP subproblems
May increase number of NLP subproblems
Numerical Example

\[ \text{min } Z = y_1 + 1.5y_2 + 0.5y_3 + x_1^2 + x_2^2 \]

s.t. \[ (x_1 - 2)^2 - x_2 \leq 0 \]
\[ x_1 - 2y_1 \geq 0 \]
\[ x_1 - x_2 - 4(1-y_2) \leq 0 \]
\[ x_1 - (1 - y_1) \geq 0 \]
\[ x_2 - y_2 \geq 0 \]
\[ x_1 + x_2 \geq 3y_3 \]
\[ y_1 + y_2 + y_3 \geq 1 \]
\[ 0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 4 \]
\[ y_1, y_2, y_3 = 0, 1 \]

(MIP-EX)

Optimum solution: \[ y_1 = 0, y_2 = 1, y_3 = 0, x_1 = 1, x_2 = 1, Z = 3.5. \]
**Starting point** \( y_1 = y_2 = y_3 = 1. \)

### Summary of Computational Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Subproblems</th>
<th>Master problems (LP's solved)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB</td>
<td>5 (NLP1)</td>
<td></td>
</tr>
<tr>
<td>OA</td>
<td>3 (NLP2)</td>
<td>3 (M-MIP) (19 LP's)</td>
</tr>
<tr>
<td>GBD</td>
<td>4 (NLP2)</td>
<td>4 (M-GBD) (10 LP's)</td>
</tr>
<tr>
<td>ECP</td>
<td>-</td>
<td>5 (M-MIP) (18 LP's)</td>
</tr>
</tbody>
</table>
Example: Process Network with Fixed Charges

- Duran and Grossmann (1986)
  - Network superstructure
Example  (Duran and Grossmann, 1986)

Algebraic MINLP:  \textit{linear in } y, \textit{convex in } x

8 0-1 variables, 25 continuous, 31 constraints (5 nonlinear)

<table>
<thead>
<tr>
<th>Method</th>
<th>NLP</th>
<th>MIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch and Bound (F-L)</td>
<td>20</td>
<td>-</td>
</tr>
<tr>
<td>Outer-Approximation</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Generalized-Benders</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Extended Cutting Plane</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>LP/NLP based</td>
<td>3</td>
<td>7 LP's vs 13 LP's OA</td>
</tr>
</tbody>
</table>
Effects of Nonconvexities

1. NLP subproblems may have local optima
2. MILP master may cut-off global optimum

Handling of Nonconvexities

1. Rigorous approach (global optimization - See Chris Floudas):
   Replace nonconvex terms by underestimators/convex envelopes
   Solve convex MINLP within spatial branch and bound

2. Heuristic approach:
   Add slacks to linearizations
   Search until no improvement in NLP

Carnegie Mellon
Handling nonlinear equations

\[ h(x,y) = 0 \]

1. In branch and bound no special provision—simply add to NLPs

2. In GBD no special provision—cancels in Lagrangian cut

3. In OA equality relaxation

\[
T^k \nabla h(x^k, y^k) \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0
\]

\[
T^k = [t^k_i], \quad t^k_i = \begin{cases} 1 & \text{if } \lambda_i^k > 0 \\ -1 & \text{if } \lambda_i^k < 0 \\ 0 & \text{if } \lambda_i^k = 0 \end{cases}
\]

Lower bounds may not be valid

Rigorous if eqtn relaxes as \( h(x,y) \leq 0 \) \( h(x,y) \) is convex

Carnegie Mellon
MIP-Master Augmented Penalty
Viswanathan and Grossmann, 1990

Slacks: \( p^k, q^k \) with weights \( w^k \)

\[
\begin{align*}
\text{min} \quad & Z^K = \alpha + \sum_{k=1}^{K} \left[ w^k p^k + w^k q^k \right] \\
\text{s.t.} \quad & \alpha \geq f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \\
& T^k \nabla h(x^k, y^k)^T \begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \leq p^k \\
& g(x^k, y^k) + \nabla g(x^k, y^k)^T \begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \leq q^k \\
& \sum_{i \in B^k} y_i - \sum_{i \in N^k} y_i \leq |B^k| - 1 \quad k = 1, \ldots, K \\
& x \in X, \; y \in Y, \; \alpha \in \mathbb{R}^1, \; p^k, q^k \geq 0
\end{align*}
\]

If convex MINLP then slacks take value of zero
\( \Rightarrow \) reduces to OA/ER

Basis DICOPT (nonconvex version)
1. Solve relaxed MINLP
2. Iterate between MIP-APER and NLP subproblem
   until no improvement in NLP
MINLP:

**Algorithms**
- Branch and Bound (BB) *Leyffer (2001), Bussieck, Drud (2003)*
- Generalized Benders Decomposition (GBD) *Geoffrion (1972)*
- Outer-Approximation (OA) *Duran and Grossmann (1986)*
- Extended Cutting Plane (ECP) *Westerlund and Pettersson (1992)*

**Codes:**
- SBB *GAMS simple B&B*
- MINLP-BB (AMPL) *Fletcher and Leyffer (1999)*
- Bonmin (COIN-OR) *Bonami et al (2006)*
- FilMINT *Linderoth and Leyffer (2006)*
- DICOPT (GAMS) *Viswanathan and Grossman (1990)*
- AOA (AIMSS)
- $\alpha$–ECP *Westerlund and Peterssson (1996)*
- MINOPT *Schweiger and Floudas (1998)*
- BARON *Sahinidis et al. (1998)*

Carnegie Mellon
High-Level Modeling

The General Algebraic Modeling System (GAMS) is a high-level modeling system for mathematical programming problems. GAMS is tailored for complex, large-scale modeling applications, and allows you to build large maintainable models that can be adapted quickly to new situations. Models are fully portable from one computer platform to another.

Wide Range of Model Types

GAMS allows the formulation of models in many different problem classes, including:

- Linear (LP) and Mixed Integer Linear (MILP)
- Quadratic Programming (QCP)
- and Mixed integer QCP (MINLP)
- Nonlinear (NLP) and Mixed Integer NLP (MINLP)
- Constrained Nonlinear Systems (CNS)
- Mixed Complementarity (MCP)
- Programs with Equilibrium Constraints (MPEC)
- Cone Programming Problems
- Stochastic Linear Problems

MINLP and Global Solvers in GAMS

The area of Mixed Integer Nonlinear Programming (MINLP) and Global Optimization has experienced significant growth in industry and academia over the last years. More and more general purpose solution algorithms have been implemented and have matured into reliable solution systems:

- AlphaECP*: Extended cutting plane method from Aalto Akademi University, Finland
- BARON Branch-and-Reduce optimization Navigator for proven global solutions from the Optimization Firm
- Bonmin*: Hybrid outer approximation based branch-and-cut algorithm jointly developed by a collaboration between Carnegie Mellon University and the IBM Corporation and distributed from CONOPT
- DOPTIP: outer approximation framework from Carnegie Mellon University
- LGO: Lipschtz global optimizer from Primo Consulting Services, Inc.
- MINLP/QGTL: Multistart method for global optimization from Optima Methods, Inc.
- SEB: Branch-and-Bound algorithm from ASK Consulting & Development A/S

* New in GAMS 22.5.
Bonmin - An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs

http://egon.cheme.cmu.edu/ibm/page.htm

IBM
A.R. Conn, J. Lee, A. Lodi, A. Wächter

Carnegie Mellon University
L. T. Biegler, I.E. Grossmann, C.D. Laird, N. Sawaya (Chemical Eng.)
G. Cornuéjols, F. Margot (OR-Tepper)

Software in COIN-OR

COIN-OR is a set of open-source codes for operations research.
Contains codes for:
Linear Programming (CLP)
Mixed Integer Linear Programming (CBC, CLP, CGL)
Non Linear Programming (IPOPT)

Goal: Produce new MINLP software

Carnegie Mellon
Bonmin  http://projects.coin-or.org/Bonmin

Single computational framework that implements:

- NLP based branch and bound (Gupta & Ravindran, 1985)
- Outer-Approximation (Duran & Grossmann, 1986)
- LP/NLP based branch and bound (Quesada & Grossmann, 1994)

a) Branch and bound scheme
b) At each node LP or NLP subproblems can be solved
   NLP solver: IPOPT  MIP solver: CLP
c) Various algorithms activated depending on what subproblem is solved at given node
   I-OA     Outer-approximation
   I-BB     Branch and bound
   I-Hyb    Hybrid  LP/NLP based B&B
Logic-based Optimization

Motivation

1. Facilitate modeling of discrete/continuous optimization problems through use symbolic expressions
2. Reduce combinatorial search effort
3. Improve handling nonlinearities

Emerging techniques

1. Constraint Programming
   Van Hentenryck (1989)
2. Generalized Disjunctive Programming
   Raman and Grossmann (1994)
3. Mixed-Logic Linear Programming
   Hooker and Osorio (1999)

Carnegie Mellon
Generalized Disjunctive Programming (GDP)

- Raman and Grossmann (1994)  
  (Extension Balas, 1979)

\[
\min \quad Z = \sum_k c_k + f(x)
\]

\[
s.t. \quad r(x) \leq 0
\]

\[
Y_{jk} \leq 0 \quad , k \in K
\]

\[
c_k = \gamma_{jk}
\]

\[
\Omega(Y) = true
\]

\[
x \in R^n, c_k \in R^1
\]

\[
Y_{jk} \in \{true, false\}
\]

\[\mbox{OR operator} \quad \bigvee_{j \in J_k}
\]

Objective Function

Common Constraints

Disjunction

Constraints

Fixed Charges

Logic Propositions

Continuous Variables

Boolean Variables

\[\mbox{Multiple Terms / Disjunctions}\]
MINLP

\[
\begin{align*}
\text{min } & \quad Z = 3.5y_1 + y_2 + 1.5y_3 + x_4 \\
& + 7x_6 + 1.2x_5 + 1.8x_1 \\
& - 11x_8 \\
\text{s.t} & \quad x_1 - x_2 - x_3 = 0 \\
& \quad x_7 - x_4 - x_5 - x_6 = 0 \\
& \quad x_5 \leq 5 \\
& \quad x_8 \leq 1 \\
& \quad x_8 - 0.9x_7 = 0 \\
& \quad x_4 = \ln(1 + x_2) \\
& \quad x_5 = 1.2 \ln(1 + x_3) \\
& \quad x_7 - 5y_1 \leq 0 \\
& \quad x_2 - 5y_2 \leq 0 \\
& \quad x_3 - 5y_3 \leq 0 \\
& \quad x_i \in \mathbb{R}, i = 1, \ldots, 8 \\
& \quad y_j \in \{0,1\}, j = 1, 2, 3
\end{align*}
\]