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.

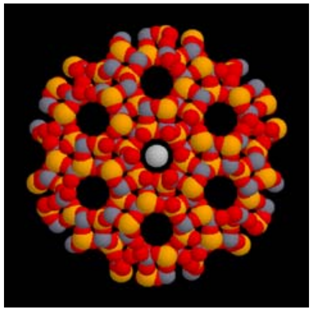
Carnegie Mellon

Process Systems Engineering (PSE)

Questions:

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

Value preservation vs. Value creation



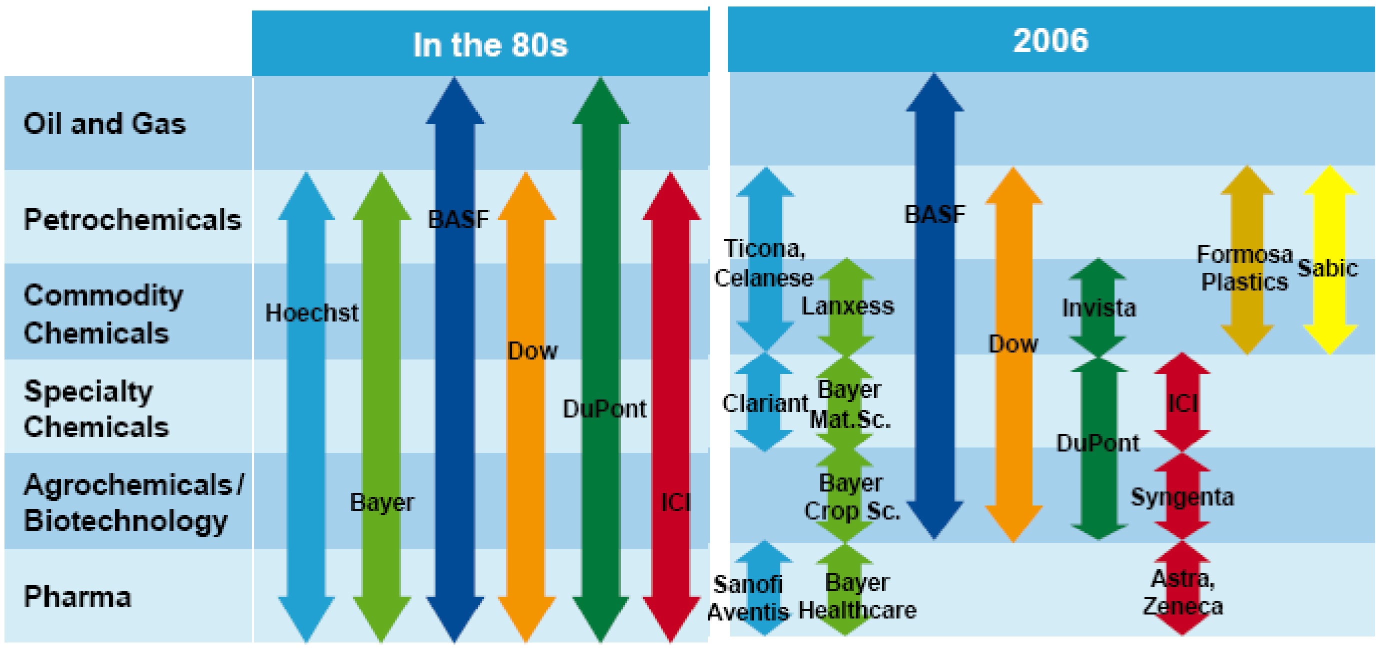
2. What is the future scope for fundamental contributions in Process Systems Engineering ?

   Science vs. Engineering

3. What are Research Challenges in Process Systems Engineering?

Changes in the Chemical Industry

From Marcinowski (2006)



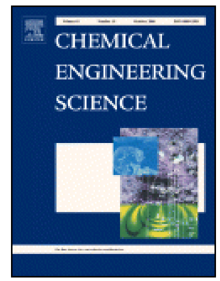
Economics of Chemical Enterprise

Value preservation vs. Value Creation

Revenues of major U.S. companies (billions)

	(2001)	*(2003)*	*(2005)*
ExxonMobil	$208.7	$237	$359
ChevronTexaco	99.7	120	193
Dow	27.8	32.6	46.3
DuPont	26.8	30.2	25.3
Procter & Gamble	39.2	43.4	56.7
Johnson & Johnson	32.3	41.8	50.5
Merck	21.2	22.4	22.0
Bristol-Myers Squibb	21.7	18.6	19.2
Amgen	4.0	8.4	12.4
Genentech	1.7	3.3	5.5

Move from Engineering to Science



⇓

Distribution Research Papers AIChE Journal 2005

	Papers	U.S.
Fluid Mechanics, Transport Phenomena	57	18
Reactors, Kinetics, Catalysis	56	21
Process Systems Engineering	51	22
Separations	37	11
Particle Technology, Fluidization	28	11
Thermodynamics	19	2
Materials, Interfaces, Electrochemical Phenomena	21	9
Environmental, Energy Engineering	16	8
Bioeng., Food, Natural Products	9	5
Total	**294**	**107**

=> 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

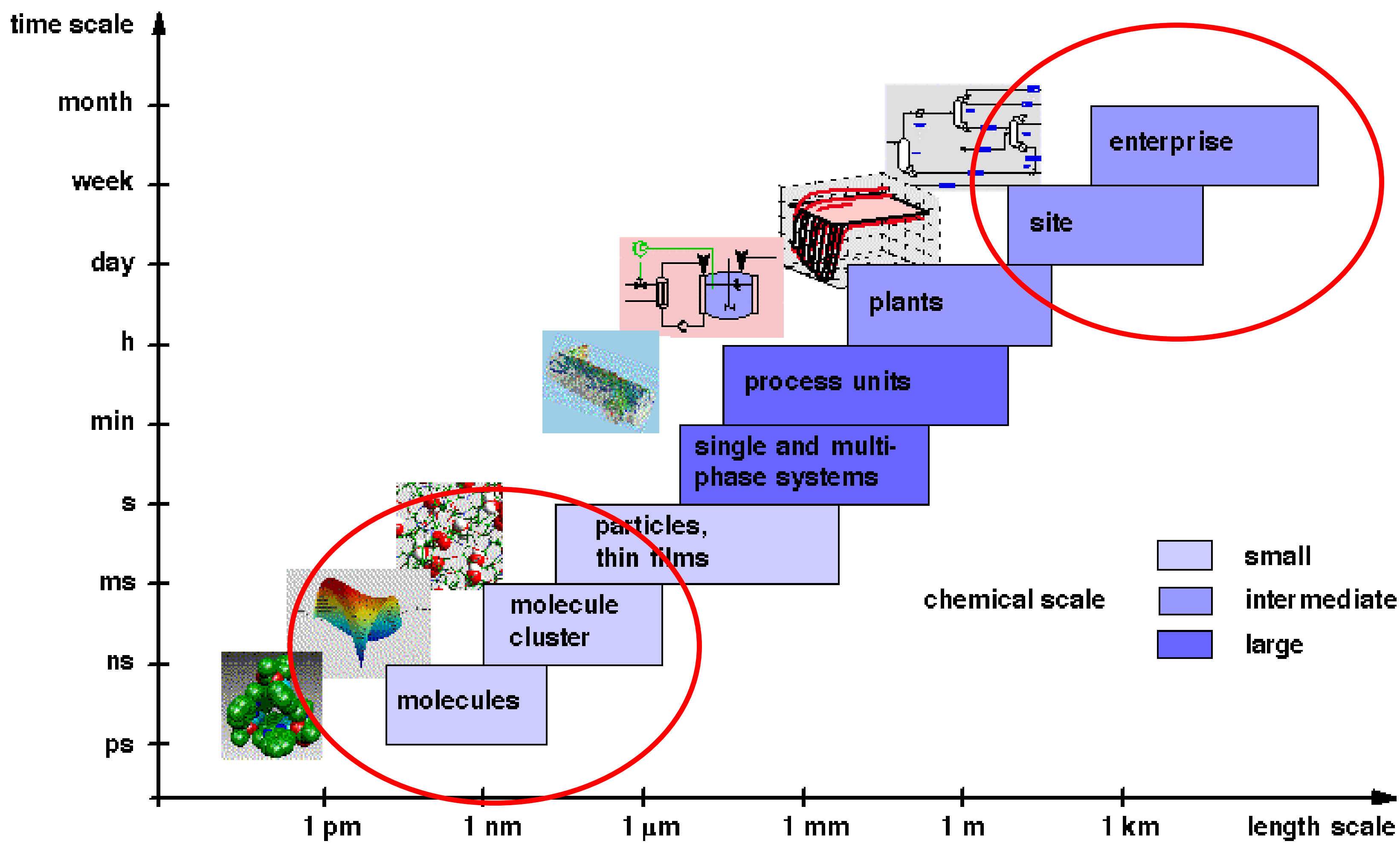
Research trend away from Chemical Engineering
Science vs Engineering

Major real world challenges
Globalization, energy, environment, health

=> Need expand scope of Process Systems Engineering

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

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?

Mathematical Programming

Rand Symposium, Santa Monica (1959)

Given a <u>space of alternatives</u> that are specified through constraints in a mathematical model <u>select decision variables</u> to optimize an objective function

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

$$s.t. \quad h(x, y) = 0$$

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

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

MINLP: Mixed-integer Nonlinear Programming Problem

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

Classical Optimization

Newton (1664)

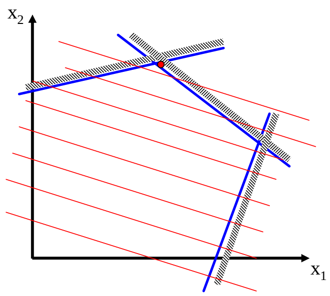
$$\min Z = f(x)$$
$$x \in R^n$$

$$\min Z = f(x)$$
$$s.t. \quad h(x) = 0$$
$$x \in R^n$$

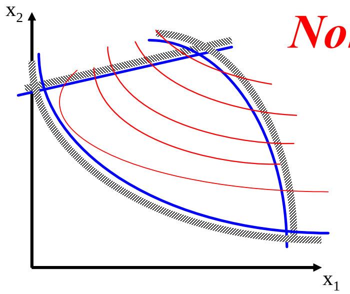
Lagrange (1778)

Solution of inequalities — **Fourier (1826)**
Solution of linear equations — **Gauss (1826)**
Solution of inequality systems — **Farkas (1902)**

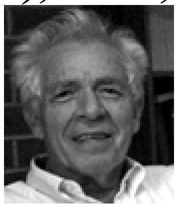
Modern Optimization

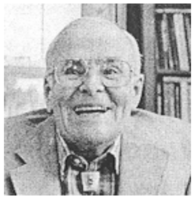


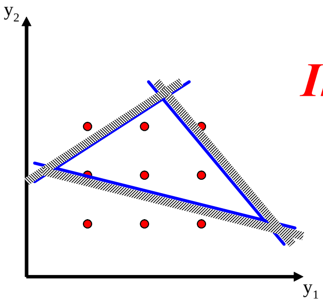
Linear Programming ***Kantorovich (1939), Dantzig (1947)***



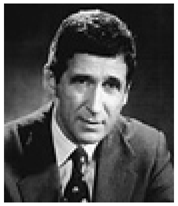
Nonlinear Programming ***Karush (1939), Kuhn, A.W.Tucker (1951)***







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**
Lovacz & Schrijver (1989), Sherali & Adams (1990),
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**

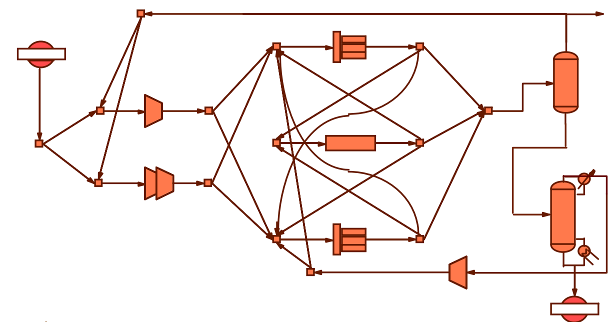
Net increase: Algorithm * Machine ~ 1 900 000x

Two million-fold increase in speed!!

Applications of Math. Programming in Chemical Engineering

Process Design

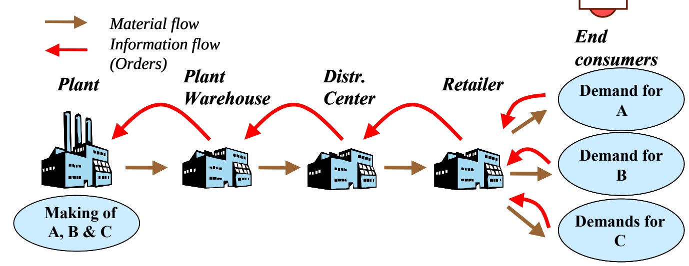
Process Synthesis



Production Planning

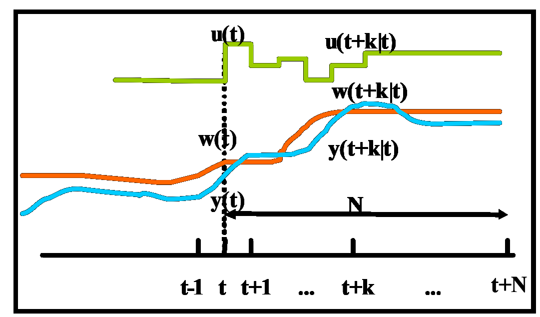
Process Scheduling

Supply Chain Management



Process Control

Parameter Estimation



LP, MILP, NLP, MINLP, Optimal Control

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

$$\min f(x, y) \quad \textit{Cost}$$

$$\text{s.t.} \quad h(x, y) = 0 \quad \textit{Process equations}$$

$$g(x, y) \leq 0 \quad \textit{Specifications}$$

$$x \in X \quad \textit{Continuous variables}$$

$$y \in \{0,1\} \quad \textit{Discrete variables}$$

Continuous optimization

Linear programming: *LP*

Nonlinear programming: *NLP*

Discrete optimization

Mixed-integer linear programming: *MILP*

Mixed-integer nonlinear programming: *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

$$\min Z = a^T y + b^T x \quad \textit{Objective function}$$

$$st \quad Ay + Bx \le d \quad \textit{Constraints}$$

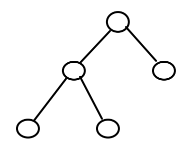
$$y \in \{0,1\}^m,\ x \ge 0$$

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! => *Small LP relaxation gap*

Drawback: exponential complexity

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

$$n = \sum_{k=1}^{N} k y_k, \quad \sum_{k=1}^{N} y_k = 1$$

also

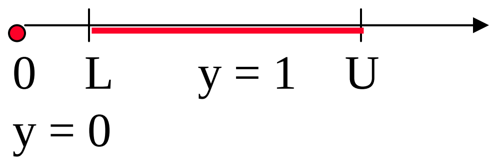
$$n = \sum_{k=1}^{M} 2^k y_k$$

Fewer 0-1 variables
Weaker relaxation

Discontinuous Functions/Domains

a) Domain

$$x = \begin{cases} 0 & \text{IF } y = 0 \\ L \le x \le U & \text{IF } y = 1 \end{cases}$$



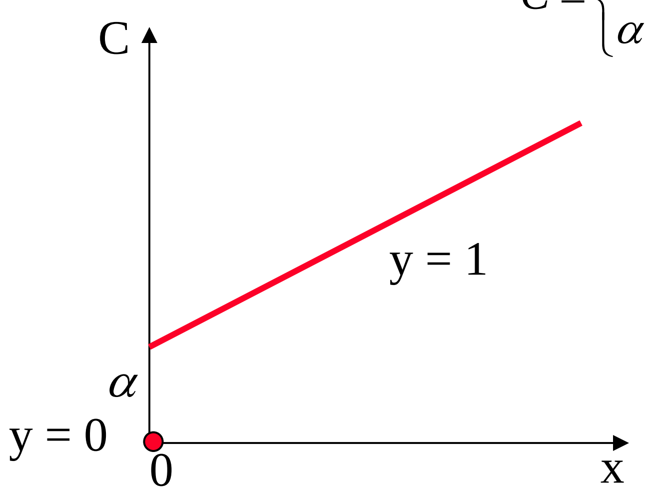
$\Rightarrow$

MIXED-INTEGER MODEL

$$Ly \le x \le Uy$$
$$y = 0, 1$$

b) Function

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



$\Rightarrow$

MIXED-INTEGER MODEL

$$C = \alpha y + \beta x$$
$$0 \le x \le U y$$
$$y = 0, 1$$

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 $0 \le y \le 1$

If solution not integer round closest

RELAXATION

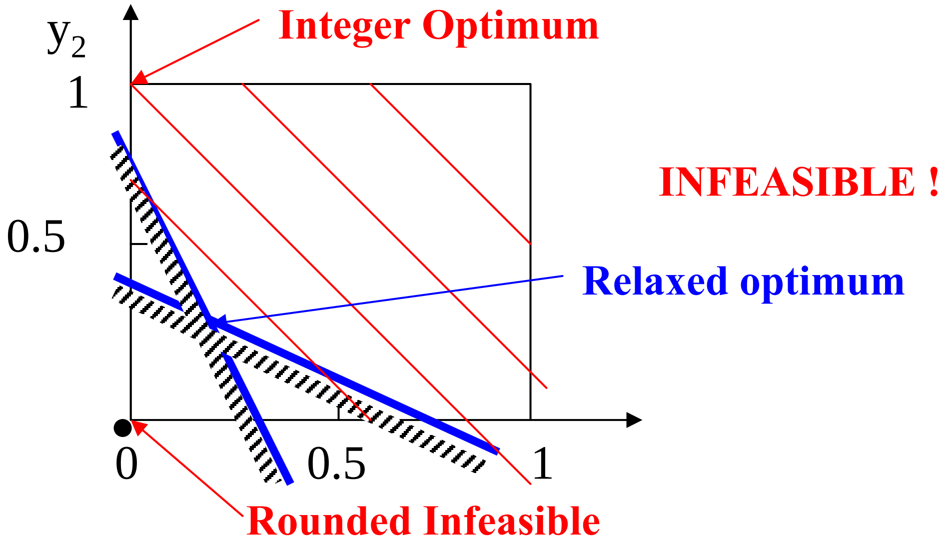
Only special cases yield integer optimum (*Assignment Problem*)

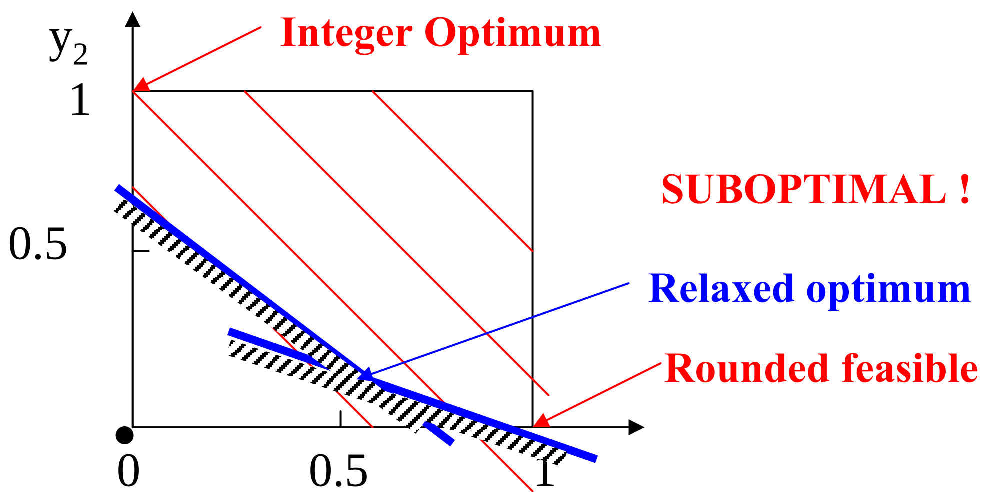
Relaxed LP provides LOWER BOUND to MILP solution

Difference: Relaxation gap

ROUNDING

May yield infeasible or suboptimal solution





Convert MIP into a Continuous NLP

Example: $\text{Min } Z = y1 + 2y2$

$\text{s.t. } 2y1 + y2 \geq 1$

$y1 = 0, 1 \quad y2 = 0, 1$

replace 0 – 1 conditions by

$0 \leq y1 \leq 1, \quad y1\,(1-y1) \leq 0$

$0 \leq y2 \leq 1 \quad y1\,(1-y2) \leq 0$

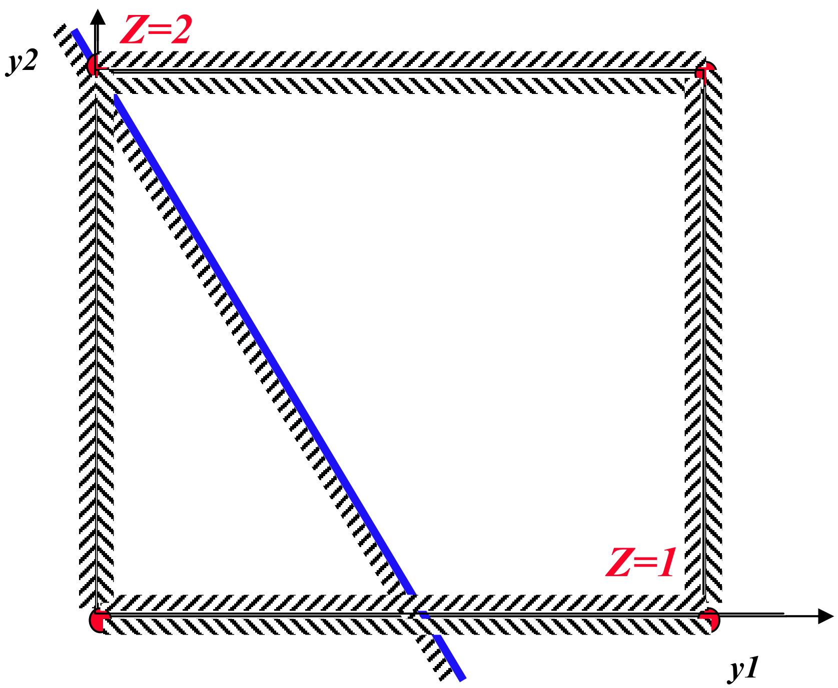
=> **Nonlinear Nonconvex!**

only feasible pts.

(1,0)

(0,1)

(1,1)



Using CONOPT2

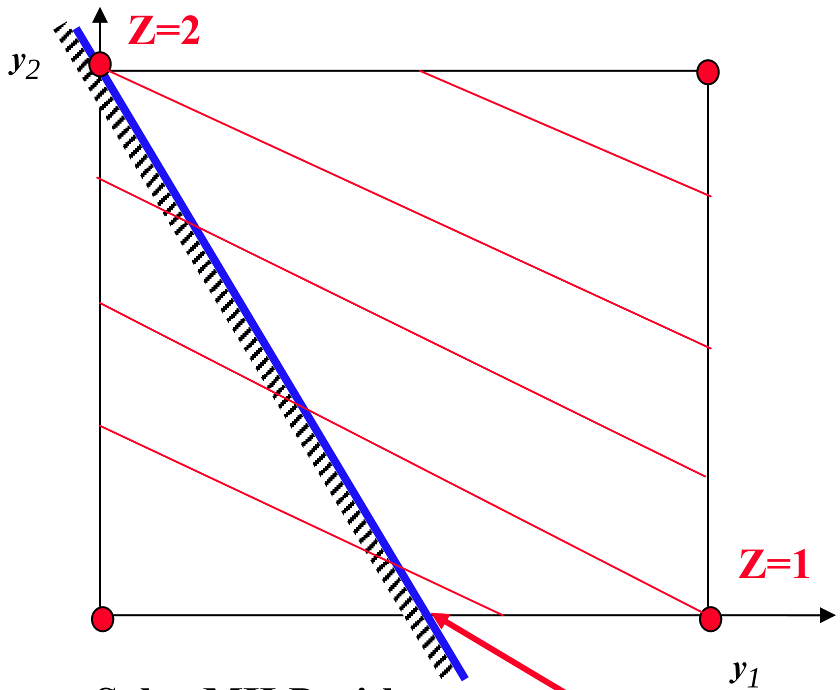
st. point $y_1 = 0, \; y_2 = 0$ => **infeasible**

st. pt. $y_1 = 0.5, \; y_2 = 0.5$ => $y_1 = 0 \; y_2 = 1$

$Z = 2$ **suboptimal**

correct solution $y_1 = 1, y_2 = 0 \;\; Z = 1$

Branch and Bound



Tree Enumeration
Solve LP At Each Node

$$\begin{aligned} &Min \quad y_1 + 2y_2 \\ &s.t. \quad 2y_1 + y_2 \geq 1 \qquad \text{(P)} \\ &y_1 = 0, 1 \quad y_2 = 0, 1 \end{aligned}$$

Solve MILP with

$0 \leq y_1 \leq 1$ $y_1 = 0.5$

$0 \leq y_2 \leq 1$ $y_2 = 0$

$Z = 0.5$ Lower Bound

1

Fix $y_1 = 0$

Fix $y_1 = 1$

$y_2 = 1$ $Z = 2$ 2

3 **$y_2 = 0$ $Z = 1$**

OPTIMUM

Carnegie Mellon

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

Gomory (1958) Crowder, Johnson, Padberg (1983), Balas, Ceria, Cornjuelos (1993)

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

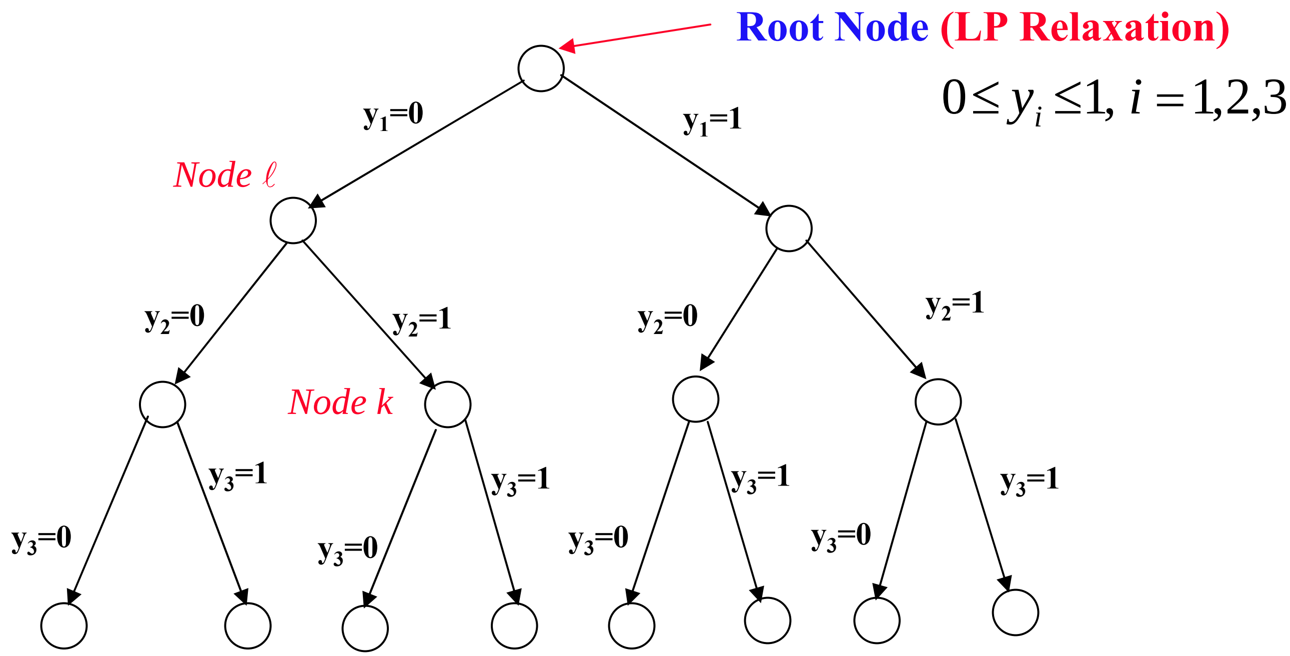
Remark

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

BRANCH-AND-CUT

Branch and Bound

Partitioning Integer Space Performed with Binary Tree



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

Node *k* descendent node *ℓ*

Chemical ENGINEERING

NODE k: LP

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

CAPD CENTER

Since node k descendent of node ℓ

1. IF LP^{ℓ} INFEASIBLE THEN LP^k INFEASIBLE

2. IF LP^k FEASIBLE $Z^{\ell} \leq Z^k$
   monotone increase objective
   Z^{ℓ} : 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

Modeling of Integer Programs

"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 $\neg$ Pi

OR $P_1 \vee P_2 \vee \ldots \vee P_r$ $y_1 + y_2 + .. + y_r \geq 1$

AND $P_1 \wedge P_2 \wedge \ldots \wedge P_r$ $y_1 \geq 1, y_2 \geq 1, \ldots y_r \geq 1$

IMPLICATION $P_1 \Rightarrow P_2$

EQUIVALENT TO $\neg P_1 \vee P_2$ $1 - y_1 + y_2 \geq 1$

OR $y_2 \geq y_1$

Systematic Procedure to Derive Linear Inequalities for Logic Propositions

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

$$Q_1 \wedge Q_2 \wedge \ldots \wedge Q_s$$

where clause Q_i : $P_1 \vee P_2 \vee \ldots \vee P_r$ *(Note: all OR)*

BASIC STEPS

1. REPLACE IMPLICATION BY DISJUNCTION

$$P1 \Rightarrow P2 \Leftrightarrow \neg P1 \vee P2$$

2. MOVE NEGATION INWARD APPLYING DE MORGAN'S THEOREM

$$\neg (P1 \wedge P2) \Leftrightarrow \neg P1 \vee \neg P2$$

$$\neg (P1 \vee P2) \Leftrightarrow \neg P1 \wedge \neg P2$$

3. RECURSIVELY DISTRIBUTE OR OVER AND

$$(P1 \wedge P2) \vee P3 \Leftrightarrow (P1 \vee P3) \wedge (P2 \vee P3)$$

EXAMPLE

flash $\Rightarrow$ dist $\vee$ abs

memb $\Rightarrow$ not abs $\wedge$ comp

$$P_F \Rightarrow P_D \vee P_A \quad (1)$$
$$P_M \Rightarrow \neg P_A \wedge P_C \quad (2)$$

(1) $\neg P_F \vee P_D \vee P_A$ *remove implication*

$$1 - y_F + y_D + y_A \geq 1$$
$$y_D + y_A \geq y_F$$

(2) $\neg P_M \vee (\neg P_A \wedge P_c)$ *remove implication*

$(\neg P_M \vee \neg P_A) \wedge (\neg P_M \vee P_c)$ *distribute OR over AND* **=> CNF!**

$$1 - y_M + 1 - y_A \geq 1 \qquad 1 - y_M + y_C \geq 1$$
$$y_M + y_A \leq 1 \qquad y_c \geq y_M$$

$$y_D + y_A \geq y_F$$
$$y_M + y_A \leq 1$$
$$y_C \geq y_M$$

Verify: $y_F = 1 \quad y_D + y_A \geq 1 \qquad y_F = 0 \quad y_D + y_A \geq 0$

$y_M = 1 \Rightarrow y_A = 0 \qquad y_C = 1$

EXAMPLE

Integer Cut

Constraint that is infeasible for integer point

$y_i = 1 \quad i \in B \qquad\qquad y_i = 0 \quad i \in N$

and feasible for all other integer points

Balas andJeroslow (1968)

Carnegie Mellon

Example: Multiperiod Problems

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

Intuitive Approach

$$y_1 + y_2 + \ldots + y_n \le n*z \qquad (1)$$

Logic Based Approach

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

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

$$(\neg y_1 \wedge \neg y_2 \wedge \ldots \wedge \neg y_n) \vee z$$

$$(\neg y_1 \vee z) \wedge (\neg y_2 \vee z) \wedge \ldots \wedge (\neg y_n \vee z)$$

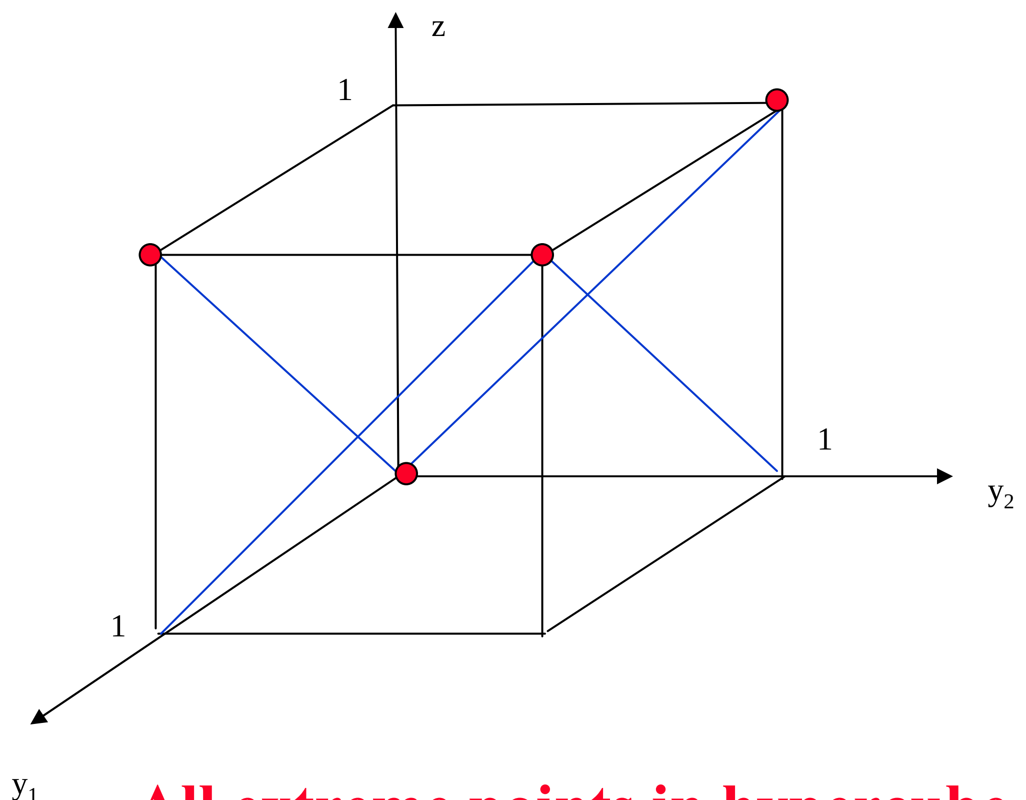
$$1 - y_1 + z \ge 1 \quad 1 - y_2 + z \ge 1 \quad 1 - y_n + z \ge 1$$

$$\begin{array}{l} y_1 \le z \\ y_2 \le z \\ \vdots \\ y_n \le z \end{array} \qquad (2)$$

Inequalities in (2) are stronger than inequalities in (1)

Geometrical interpretation

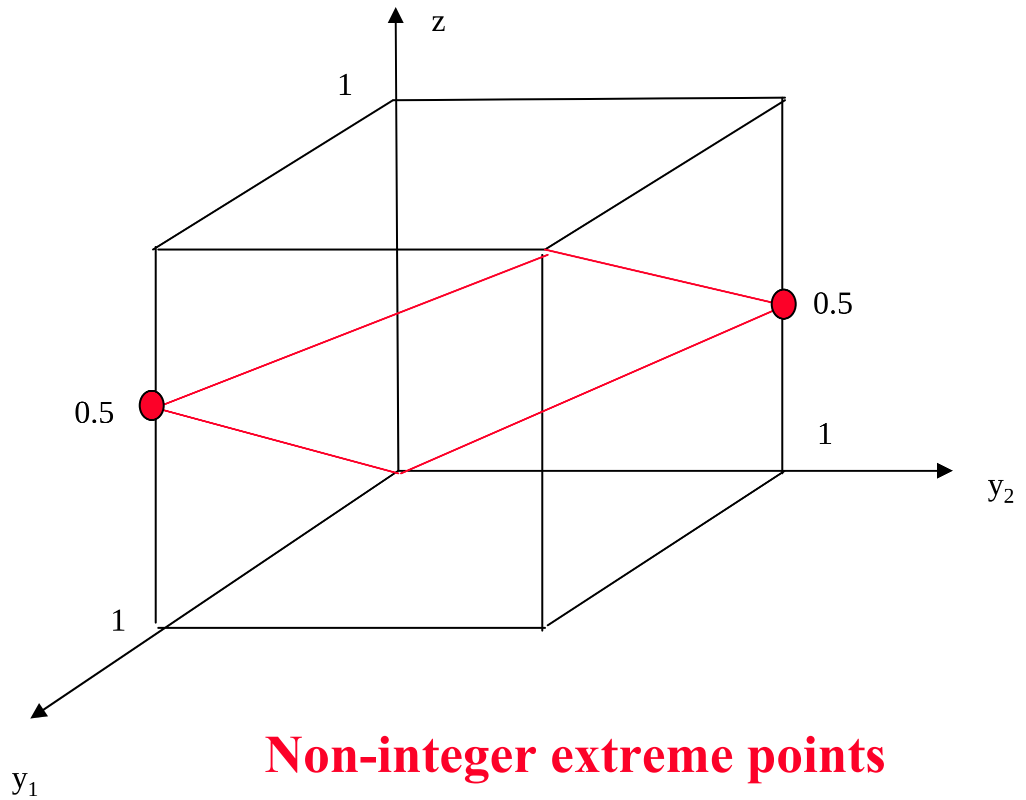
$$y_1 \leq z$$
$$y_2 \leq z$$



All extreme points in hypercube are integer!

Geometrical interpretation

$$y_1 + y_2 \leq 2z$$



Non-integer extreme points

Weaker relaxation!

Modeling of Disjunctions

$$\bigvee_{i \in D} \left[A_i x \le b_i \right]$$ one inequality must hold

Example: A before B *OR* B before A

$$\left[TS_A + pt_A \le TS_B \right] \vee \left[TS_B + pt_B \le TS_A \right]$$

Big M Formulation

$$A_i\, x \le b_i + M_i\,(1 - y_i) \qquad i \in D$$

$$\sum_{i \in D} y_i = 1$$

Difficulty: Parameter M_i

Must be sufficiently large to render inequality redundant

Large value yields poor relaxation

Convex-hull Formulation (Balas, 1985)

$$x = \sum_{i \in D} z_i$$ disaggregation vars.

$$A_i z_i \leq b_i y_i \qquad i \in D$$

$$\sum_{i \in D} y_i = 1$$

$$0 \leq z_i \leq U y_i \qquad i \in D$$ (may be removed)

$$y_i = 0, 1$$

Derivation

$$A_i \, x \, y_i \leq b_i \, y_i \qquad i \in D \qquad \text{(B)}$$ nonlinear disj. equiv.

$$\sum_{i \in D} y_i = 1$$

Carnegie Mellon

Let $z_i = x\, y_i$ disaggregated variable

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

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

to ensure $z_i = 0$ if $y_i = 0$ (C)

$$0 \le z_i \le U y_i$$

(A) $\Rightarrow$ $x = \sum_{i \in D} z_i$

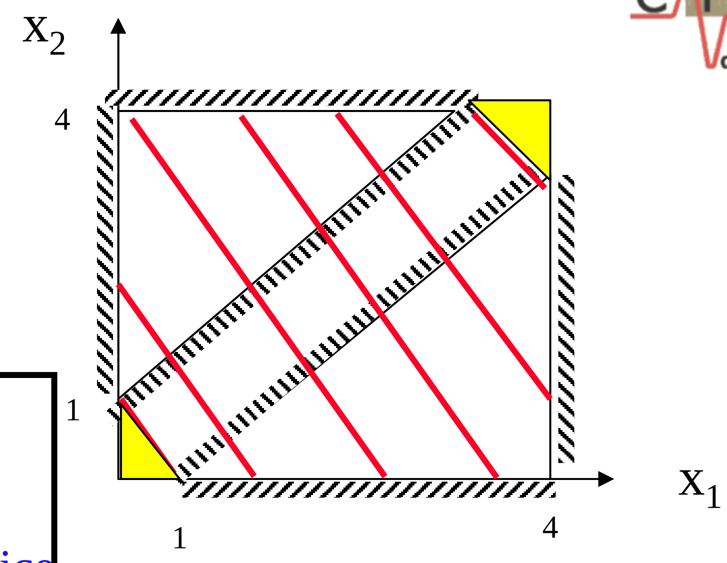
subst. (B) $A_i z_i \le b_i y_i \quad i \in D$

$$\sum_{i \in D} y_i = 1$$

(C) $\Rightarrow$ $0 \le z_i \le U y_i \quad i \in D$

Example

$$[x_1 - x_2 \le -1] \vee [-x_1 + x_2 \le -1]$$
$$0 \le x_1,\ x_2 \le 4$$



big M

$$x_1 - x_2 \le -1 + M(1 - y_1)$$
$$-x_1 + x_2 \le -1 + M(1 - y_2)$$
$$y_1 + y_2 = 1 \quad M = 10$$ possible choice

Convex hull

$$x_1 = z_1^1 + z_1^2$$
$$x_2 = z_2^1 + z_2^2$$
$$z_1^1 - z_2^1 \le -y_1 \qquad -z_1^2 + z_2^2 \le -y_2$$
$$y_1 + y_2 = 1$$
$$0 \le z_1^1 \le 4y_1$$
$$0 \le z_1^2 \le 4y_2$$
$$0 \le z_2^1 \le 4y_1$$
$$0 \le z_2^2 \le 4y_2$$

Nonlinear Programming

NLP: Algorithms

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

- Sucessive 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

MINLP

- **Mixed-Integer Nonlinear Programming**

$$\min Z = f(x, y)$$ *Objective Function*

$$s.t. \quad g(x, y) \le 0$$ *Inequality Constraints*

$$x \in X, \; y \in Y$$

$$X = \{x \mid x \in R^n, x^L \le x \le x^U, Bx \le b\}$$

$$Y = \{y \mid y \in \{0,1\}^m, Ay \le a\}$$

- *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

Solution Algorithms

- **Branch and Bound method (BB)**

Ravindran and Gupta (1985) Leyffer and Fletcher (2001)

Branch and cut: *Stubbs and Mehrotra (1999)*

- **Generalized Benders Decomposition (GBD)**

Geoffrion (1972)

- **Outer-Approximation (OA)**

Duran & Grossmann (1986), Yuan et al. (1988), Fletcher & Leyffer (1994)

- **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*

$$\begin{aligned} &\min Z_{LB}^k = f(x,y) \\ s.t.\ \ &g_j(x,y) \le 0 \quad j \in J \\ &x \in X,\ y \in Y_R \\ &y_i \le \alpha_i^k \ \ i \in I_{FL}^k \\ &y_i \ge \beta_i^k \ \ i \in I_{FU}^k \end{aligned} \qquad \text{(NLP1)}$$

b) NLP Fixed y^k *Upper bound*

$$\begin{aligned} &\min Z_U^k = f(x,y^k) \\ s.t.\ \ &g_j(x,y^k) \le 0 \quad j \in J \\ &x \in X \end{aligned} \qquad \text{(NLP2)}$$

c) Feasibility subproblem for fixed y^k.

$$\begin{aligned} &\min u \\ s.t.\ \ &g_j(x,y^k) \le u \quad j \in J \\ &x \in X,\ u \in R^1 \end{aligned} \qquad \text{(NLPF)}$$

Infinity-norm

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

$$\min Z_L^K = \alpha$$

$$\left.\begin{array}{l} st\ \ \alpha \geq f(x^k,y^k) + \nabla f(x^k,y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ \qquad g_j(x^k,y^k) + \nabla g_j(x^k,y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \leq 0 \quad j \in J \end{array}\right\} k = 1,...K$$

$$x \in X,\ y \in Y$$

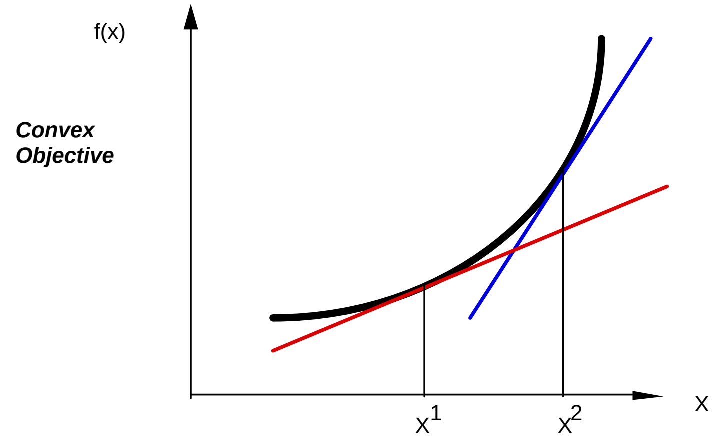
Notes:

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

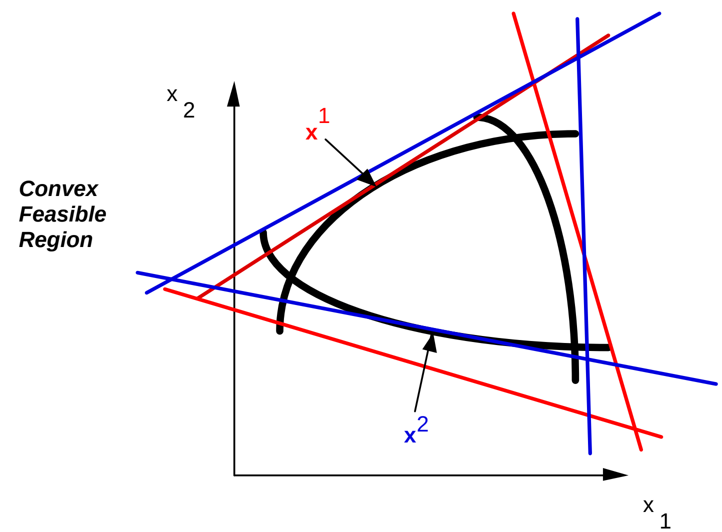
b) Linearizations *accumulated* as iterations K increase

c) Non-decreasing sequence lower bounds

Linearizations and Cutting Planes



Underestimate Objective Function



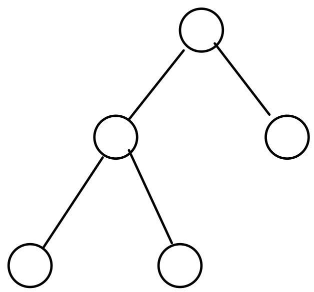
Overestimate Feasible Region

Branch and Bound

NLP1:

$$\min \; Z_{LB}^k = f(x,y)$$
$$s.t. \quad g_j(x,y) \le 0 \qquad j \in J$$
$$x \in X, \quad y \in Y_R$$
$$y_i \le \alpha_i^k \qquad i \in I_{FL}^k$$
$$y_i \ge \beta_i^k \qquad i \in I_{FU}^k$$

Tree Enumeration



Successive solution of NLP1 subproblems

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

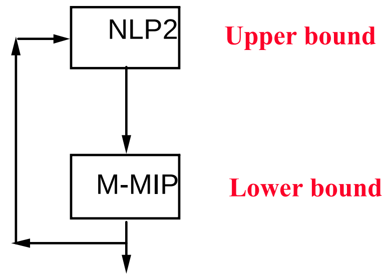
Disadvantage:
Potentially many NLP subproblems

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

Less stringent than other methods

Outer-Approximation

Alternate solution of NLP and MIP problems:



NLP2:

$$\min Z_U^k = f(x, y^k)$$
$$s.t.\ \ g_j(x, y^k) \le 0 \quad j \in J$$
$$x \in X$$

M-MIP:

$$\min \ Z_L^K = \alpha$$
$$\left.\begin{aligned} st \quad \alpha \ &\ge f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ g_j(x^k, y^k) &+ \nabla g_j(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le 0 \ \ j \in J^k \end{aligned}\right\} \ k = 1, \ldots K$$
$$x \in X, \ \ y \in Y$$

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.

Carnegie Mellon

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)

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

$$g\left(x^k,y^k\right) + \nabla g_j\left(x^k,y^k\right)^T \begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \leq 0 \quad j \in J^k \qquad (1)$$

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

$$\alpha \geq f\left(x^k,y^k\right) + \nabla_y f\left(x^k,y^k\right)^T\left(y-y^k\right) \Big\} \qquad (2)$$

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

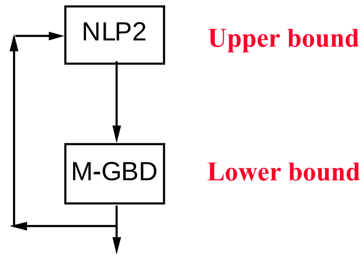
Lagrangian cut

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

$$\left(\lambda^k\right)^T\left[g\left(x^k,y^k\right) + \nabla_y g\left(x^k,y^k\right)^T\left(y-y^k\right)\right] \leq 0$$

Generalized Benders Decomposition

Alternate solution of NLP and MIP problems:



NLP2:

$$\begin{aligned} &\min Z_U^k = f(x, y^k) \\ s.t.\ \ &g_j(x, y^k) \le 0 \quad j \in J \\ &x \in X \end{aligned}$$

M-GBD:

$$\begin{aligned} &\min \quad Z_L^K = \alpha \\ st \quad &\alpha \ \ge \ f(x^k, y^k) + \nabla_y f(x^k, y^k)^T \left(y - y^k\right) \\ &+ \left(\mu^k\right)^T \left[g(x^k, y^k) + \nabla_y g\ (x^k, y^k)^T \left(y - y^k\right)\right] \quad k \in KFS \\ &\left(\lambda^k\right)^T \left[g(x^k, y^k) + \nabla_y g\ (x^k, y^k)^T \left(y - y^k\right)\right] \le 0 \quad k \in KIS \\ &y \in Y,\ \ \alpha \in R^1 \end{aligned}$$

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

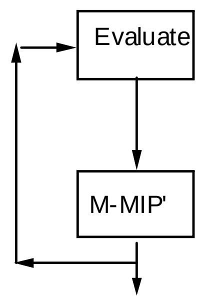
Sahinidis, Grossmann (1991)

=> Also applies to Outer-Approximation

Carnegie Mellon

Extended Cutting Plane

Westerlund and Pettersson (1995)



No NLP !

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

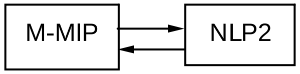
= > Convergence may be slow

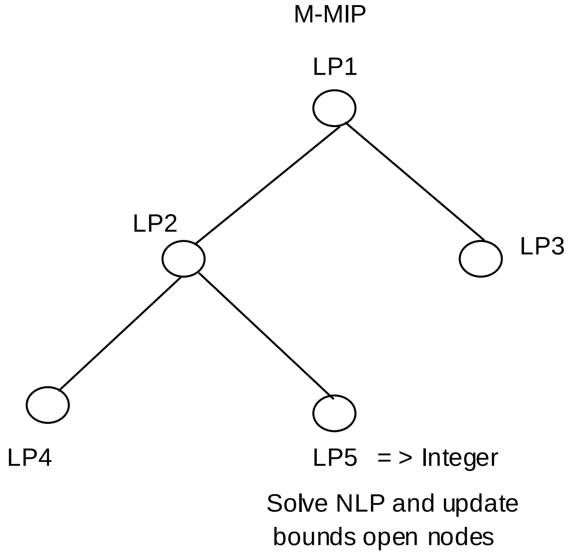
Carnegie Mellon

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

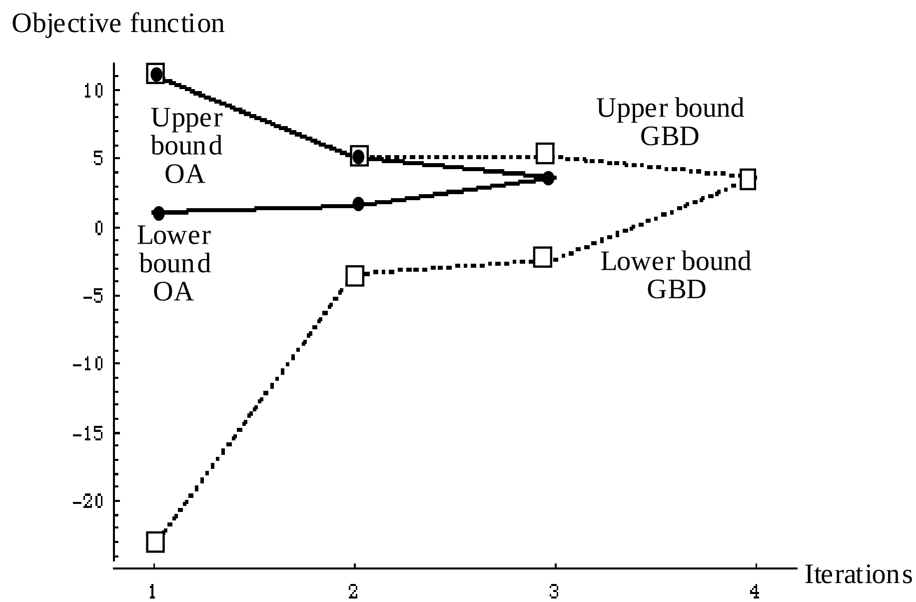
Numerical Example

$$\min Z = y_1 + 1.5y_2 + 0.5y_3 + x_1^2 + x_2^2$$

$$\begin{aligned}
\text{s.t.}\quad & (x_1 - 2)^2 - x_2 \le 0 \\
& x_1 - 2y_1 \ge 0 \\
& x_1 - x_2 - 4(1-y_2) \le 0 \\
& x_1 - (1 - y_1) \ge 0 \\
& x_2 - y_2 \ge 0 \qquad \text{(MIP-EX)} \\
& x_1 + x_2 \ge 3y_3 \\
& y_1 + y_2 + y_3 \ge 1 \\
& 0 \le x_1 \le 4, \quad 0 \le x_2 \le 4 \\
& y_1, y_2, y_3 = 0, 1
\end{aligned}$$

Optimum solution: $y_1=0$, $y_2 = 1$, $y_3 = 0$, $x_1 = 1$, $x_2 = 1$, $Z = 3.5$.

Starting point $y_1 = y_2 = y_3 = 1.$

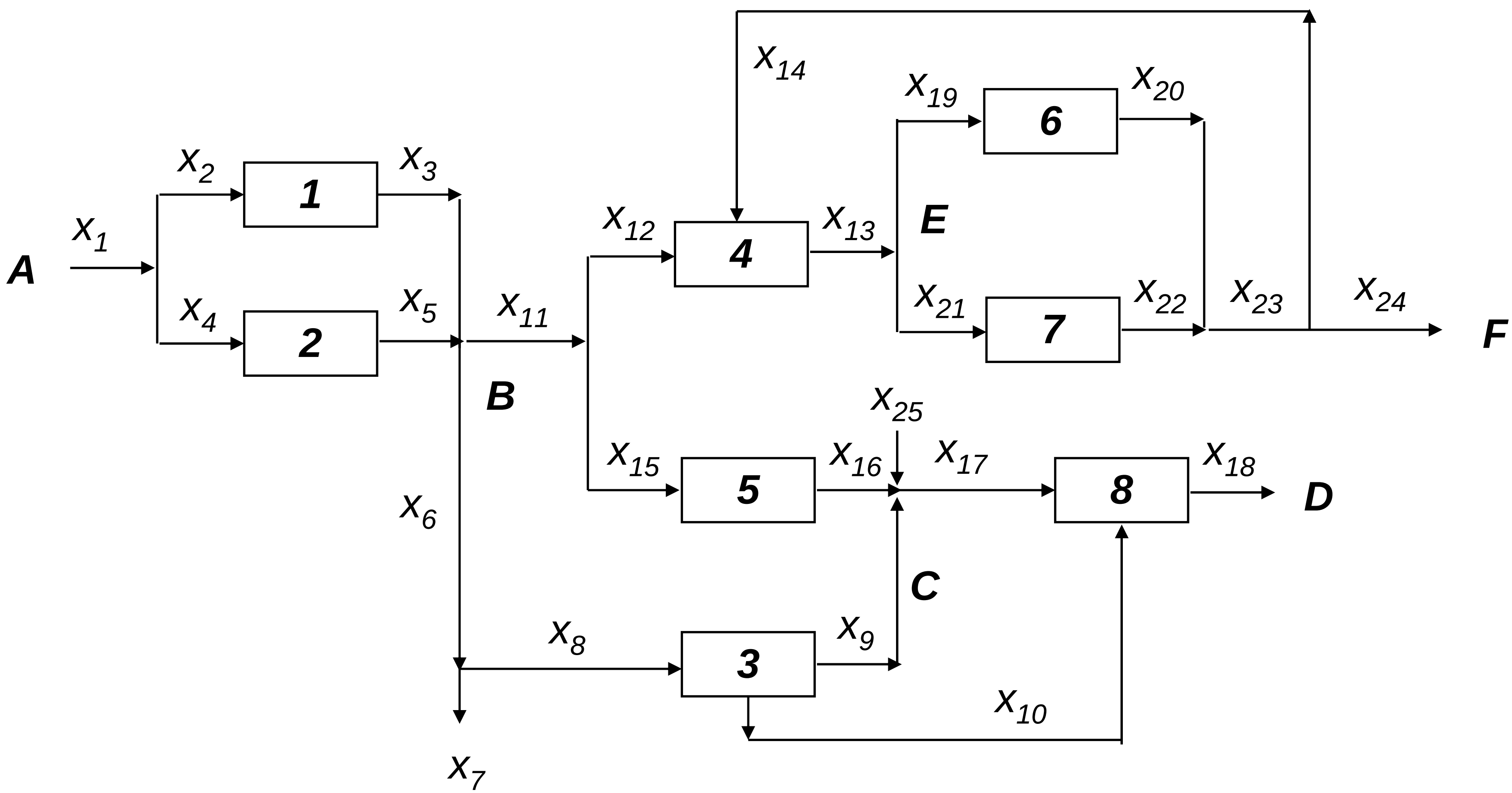


Summary of Computational Results

Method	Subproblems	Master problems (LP's solved)
BB	5 (NLP1)	
OA	3 (NLP2)	3 (M-MIP) (19 LP's)
GBD	4 (NLP2)	4 (M-GBD) (10 LP's)
ECP	-	5 (M-MIP) (18 LP's)

Example: Process Network with Fixed Charges

- *Duran and Grossmann (1986)*
  - **Network superstructure**



Example *(Duran and Grossmann, 1986)*

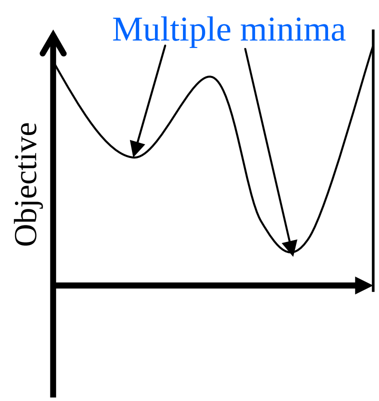
Algebraic MINLP: *linear in y, convex in x*

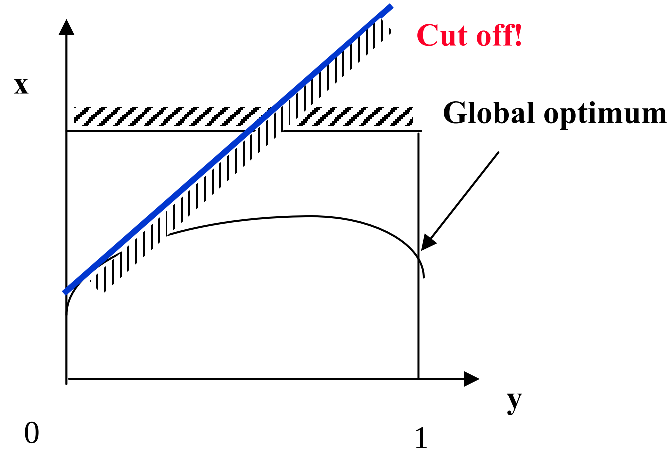
8 0-1 variables, 25 continuous, 31 constraints (5 nonlinear)

	NLP	MIP
Branch and Bound *(F-L)*	20	-
Outer-Approximation:	3	3
Generalized-Benders	10	10
Extended Cutting Plane	-	15
LP/NLP based	3	7 LP's vs 13 LP's OA

Effects of Nonconvexities

1. NLP supbroblems 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 underestimtors/convex envelopes
   Solve convex MINLP within spatial branch and bound

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

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**

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

$$T^k = \left[t_{ii}^k \right], \quad t_{ii}^k = \begin{cases} 1 \ \ if \ \ \lambda_i^k > 0 \\ -1 \ if \ \ \lambda_i^k < 0 \\ 0 \ \ if \ \ \lambda_i^k = 0 \end{cases}$$

Lower bounds may not be valid

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

MIP-Master Augmented Penalty

Viswanathan and Grossmann, 1990

Slacks: p^k, q^k with weights w^k

$$\min \quad Z^K = \alpha + \sum_{k=1}^{K} \left[w_p^k p^k + w_q^k q^k \right] \qquad \text{(M-APER)}$$

$$\left.\begin{aligned} s.t. \quad & \alpha \ge f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ & T^k \nabla h(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le p^k \\ & g(x^k, y^k) + \nabla g(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le q^k \end{aligned}\right\} \; k = 1, \ldots K$$

$$\sum_{i \in B^k} y_i - \sum_{i \in N^k} y_i \le \left| B^k \right| - 1 \quad k = 1, \ldots K$$

$$x \in X, \; y \in Y, \; \alpha \in \boldsymbol{R}^1, \; p^k, q^k \ge 0$$

If convex MINLP then slacks take value of zero
=> 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

Carnegie Mellon

Mixed-integer Nonlinear Programming

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*)*

α–ECP *Westerlund and Peterssson (1996)*
MINOPT *Schweiger and Floudas (1998)*

BARON *Sahinidis et al. (1998)*

GAMS

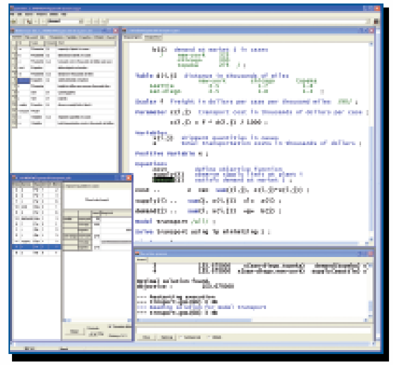
Optimization

www.gams.com

- Support
- Sales
- Solvers
- Documentation
- Model Library
- gamsworld.org

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.



GAMS Integrated Developer Environment for editing, debugging and solving models and viewing data.

Wide Range of Model Types

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

- Linear (LP) and Mixed Integer Linear (MIP)
- Quadratic Programming (QCP) and Mixed Integer QCP (MIQCP)
- Nonlinear (NLP) and Mixed Integer NLP (MINLP)
- Constrained Nonlinear Systems (CNS)
- Mixed Complementary (MCP)
- Programs with Equilibrium Constraints (MPEC)
- Conic Programming Problems
- Stochastic Linear Problems

State-of-the-Art Solvers

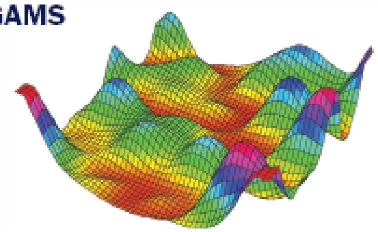
GAMS incorporates all major commercial and academic **state-of-the-art solution technologies** for a broad range of problem types, including global nonlinear optimization solvers.

MINLP and Global Solvers in GAMS

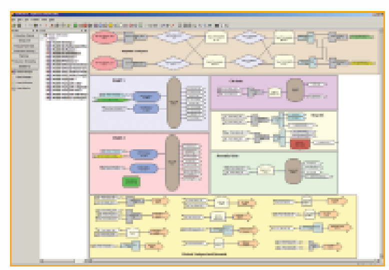
The area of Mixed Integer Nonlinear Programming (MINLP) and Global Optimization has experienced significant growth in industry and academia over the last years. More and more general purpose solution algorithms have been implemented and have matured into reliable solution systems:

- AlphaECP^Δ: Extended cutting plane method from Åbo Akademi University, Finland
- BARON: Branch-And-Reduce Optimization Navigator for proven global solutions from The Optimization Firm
- Bonmin^Δ: Hybrid outer-approximation based branch-and-cut algorithm jointly developed by a collaboration between Carnegie Mellon University and the IBM Corporation and distributed from COIN-OR
- DICOPT: Outer approximation framework from Carnegie Mellon University
- LGO: Lipschitz global optimizer from Pinter Consulting Services, Inc.
- LINDOGlobal^Δ: A branch-and-bound solver for proven global solutions from Lindo Systems, Inc.
- MSNLP/OQNLP: Multi-start method for global optimization from Optimal Methods, Inc.
- SBB: Branch-and-Bound algorithm from ARKI Consulting & Development A/S

Δ New in GAMS 22.5.



Surface of a function with multiple local optima.
© Janos D. Pinter, PCS Inc.



Screenshot from SCMart Suite deploying MINLP models from Optience Corp.

Contact:
GAMS Development Corporation
1217 Potomac Street, N.W.
Washington, D.C. 20007, USA
Tel.: +1-202-342-0180
Fax: +1-202-342-0181
sales@gams.com
http://www.gams.com

in Europe:
GAMS Software GmbH
Eupener Str. 135-137
50933 Cologne, Germany
Tel.: +49-221-949-9170
Fax: +49-221-949-9171
info@gams.de
http://www.gams.de

Chemical Engineering

CAPD CENTER

Carnegie Mellon

Bonmin-An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs

http://egon.cheme.cmu.edu/ibm/page.htm *Pierre Bonami*

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)*

Generalized Disjunctive Programming (GDP)

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

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

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

OR operator ⟶

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

Disjunction

Constraints

Fixed Charges

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

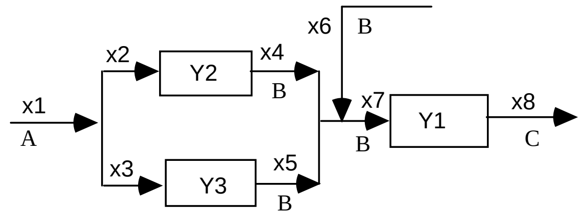
$$x \in R^n, c_k \in R^1$$ **Continuous Variables**

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

◆**Multiple Terms / Disjunctions**

Modeling Example

Process: A –> B –> C



MINLP

$$\min \quad Z = 3.5y_1 + y_2 + 1.5y_3 + x_4 + 7x_6 + 1.2x_5 + 1.8x_1 - 11x_8$$

s.t

$$x_1 - x_2 - x_3 = 0$$

$$x_7 - x_4 - x_5 - x_6 = 0$$

$$x_5 \leq 5$$

$$x_8 \leq 1$$

$$x8 - 0.9x7 = 0$$

$$x_4 = \ln(1+x_2)$$

$$x5 = 1.2 \ln(1+x3)$$

$$x_7 - 5y_1 \leq 0$$

$$x_2 - 5y_2 \leq 0$$

$$x_3 - 5y_3 \leq 0$$

$$x_i \in R \quad i = 1,...,8$$

$$y_j \in \{0,1\} \quad j = 1,2,3$$