GLOBAL OPTIMIZATION AND OPTIMIZATION UNDER UNCERTAINTY

Pan American Study Institute
on Process Systems Engineering
Iguazu Falls, Argentina
18 August 2005

Nick Sahinidis

University of Illinois at Urbana-Champaign
Chemical and Biomolecular Engineering

OUTLINE

GLOBAL OPTIMIZATION

1. Introduction
   - Convex analysis basics
   - Applications
2. Algorithms
   - Branch-and-bound
   - Convexification
3. Branch-and-reduce
   - Algorithms
   - Applications
   - BARON software

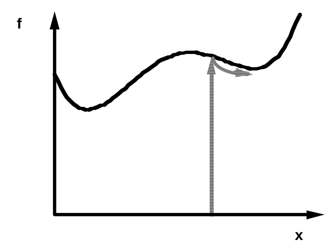
OPTIMIZATION UNDER UNCERTAINTY

1. Models of uncertainty
   - Stochastic programming
   - Probabilistic programming
   - Fuzzy programming
2. Stochastic programming
   - Linear
   - Integer
3. Approximations
   - Capacity expansion of chemical processes

PART 1: INTRODUCTION

- Optimization basics
  - Optimality conditions
  - Convex sets and functions
  - The shape of typical modeling functions
- What is global optimization?
- Why do we need it?
  - Applications

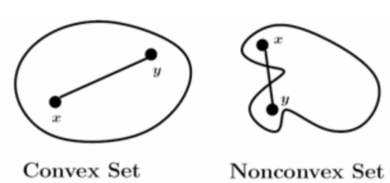
THE MULTIPLE-MINIMA DIFFICULTY



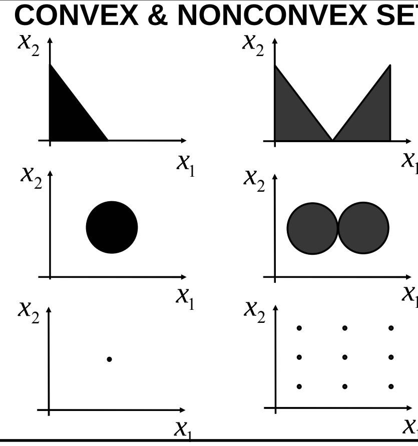
- Classical optimality conditions are necessary but not sufficient
- Classical optimization provides the local minimum “closest” to the starting point used

CONVEX AND NONCONVEX SETS

A set is convex if, for every pair of points in the set, the line segment joining these two points belongs to the set

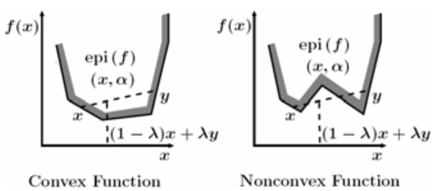


CONVEX & NONCONVEX SETS



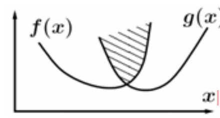
CONVEX AND NONCONVEX FUNCTIONS

A function is convex if its epigraph is convex

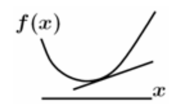


PROPERTIES OF CONVEX FUNCTIONS

- **Any positive combination of two convex functions is convex**
- **The maximum of two convex functions is convex**

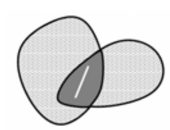


- **All tangents of a differential convex function lie below the function**

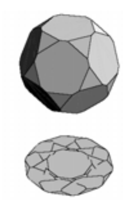


PROPERTIES OF CONVEX SETS

- **The intersection of convex sets is convex**

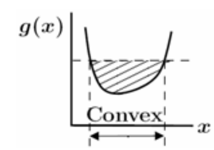


- **Linear transformations such as projection preserve convexity**

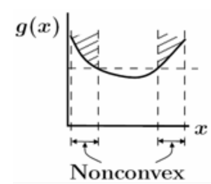


CONVEX FUNCTIONS AND SETS

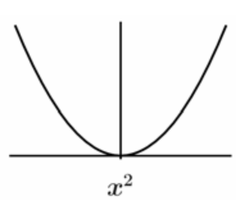
- **The lower level set *g(x)*≤*α* is convex if *g(x)* is a convex function**

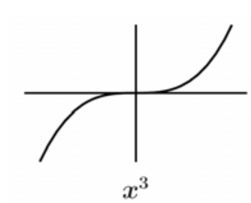


- **The upper level set *g(x)*≥*α* may be nonvonvex**

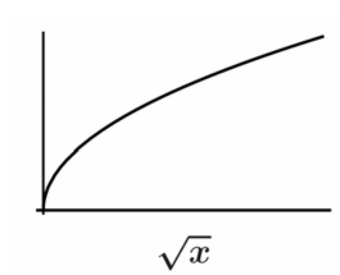


COMMON FUNCTIONS IN MODELING

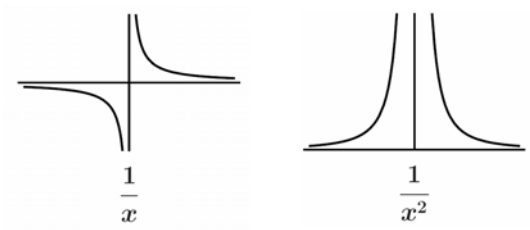




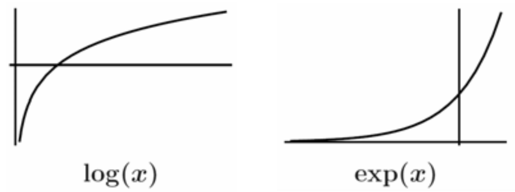
COMMON FUNCTIONS IN MODELING



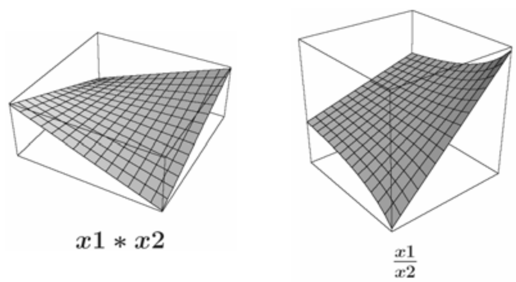
COMMON FUNCTIONS IN MODELING



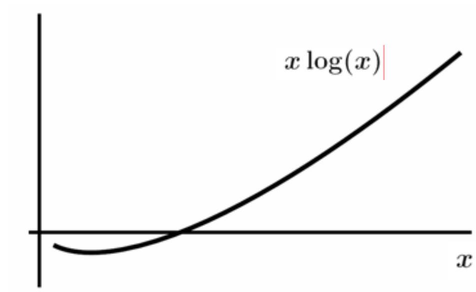
COMMON FUNCTIONS IN MODELING



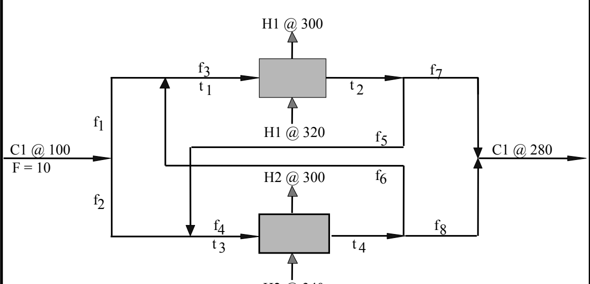
COMMON FUNCTIONS IN MODELING



A CONVEX FUNCTION



HEAT EXCHANGER NETWORK SYNTHESIS



Energy balances introduce nonconvexities (conservation of flow * temperature)

HENS MODEL

$$\min \quad 1200\left[800 \Big/ 2.5\left(2/3\sqrt{(320-t_2)(300-t_1)} + \frac{(320-t_2)+(300-t_1)}{6}\right)\right]^{0.6} +$$

$$1200\left[1000 \Big/ 0.2\left(2/3\sqrt{(340-t_4)(300-t_3)} + \frac{(340-t_4)+(300-t_3)}{6}\right)\right]^{0.6}$$

s.t.

$$f_1 + f_2 = 10$$

$$f_1 + f_6 = f_3$$

$$f_2 + f_5 = f_4$$

$$f_5 + f_7 = f_3$$

$$f_6 + f_8 = f_4$$

$$100 f_1 + t_4 f_6 = t_1 f_3$$

$$100 f_2 + t_2 f_5 = t_3 f_4$$

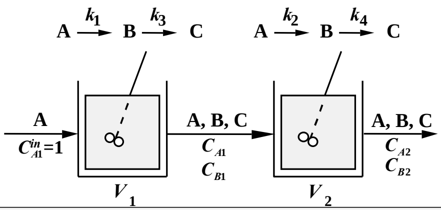
$$f_3(t_2 - t_1) = 800$$

$$f_4(t_4 - t_3) = 1000$$

$$0 \le f \le 10$$

$$(100, 100, 100, 100) \le t \le (290, 310, 290, 330)$$

REACTOR NETWORK DESIGN

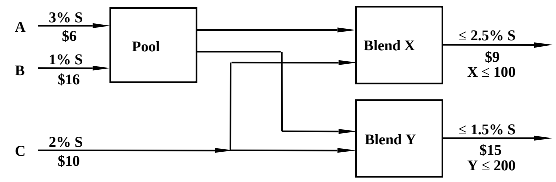


min $\quad - C_{B2}$

s.t.

$$C_{A1} - 1 + k_1 C_{A1} V_1 = 0$$
$$C_{A2} - C_{A1} + k_2 C_{A2} V_2 = 0$$
$$C_{B1} + C_{A1} - 1 + k_3 C_{B1} V_1 = 0$$
$$C_{B2} - C_{B1} + C_{A2} - C_{A1} + k_4 C_{B2} V_2 = 0$$
$$V_1^{0.5} + V_2^{0.5} \le 4$$
$$0 \le (C_{A1}, C_{A2}, C_{B1}, C_{B2}, V_1, V_2) \le (1,1,1,1,16,16)$$

BLENDING AND POOLING



Flow * Property must be preserved: nonconvex property balances

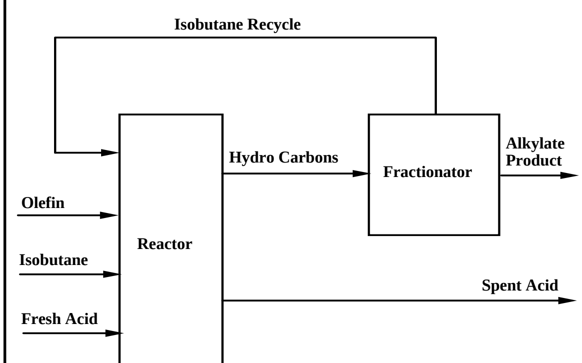
POOLING MODEL

min $\quad -9x_5 - 15x_9 + 6x_1 + 16x_2 + 10x_6$

s.t.

$$x_1 + x_2 = x_3 + x_4$$
$$x_3 + x_7 = x_5$$
$$x_4 + x_8 = x_9$$
$$x_7 + x_8 = x_6$$
$$x_{10}\, x_3 + 2x_7 \le 2.5x_5$$
$$x_{10}\, x_4 + 2x_8 \le 1.5x_9$$
$$3x_1 + x_2 = x_{10}(x_3 + x_4)$$
$$(0,0,0,0,0,0,0,0,0,1) \le x$$
$$x \le (300, 300, 100, 200, 100, 300, 100, 200, 200, 3)$$

ALKYLATION PROCESS OPTIMIZATION



Regression identifies nonconvex process models

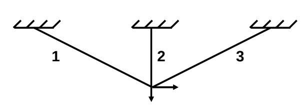
ALKYLATION PROCESS MODEL

min $\quad 5.04x_1 + 0.035x_2 + 10x_3 + 3.36x_5 - 0.063x_4 x_7$

s.t.

$$x_1 = 1.22x_4 - x_5$$
$$x_9 + 0.222x_{10} = 35.82$$
$$3x_7 - x_{10} = 133$$
$$x_7 = 86.35 + 1.098x_8 - 0.038x_8^2 + 0.325(x_8 - 89)$$
$$x_4 x_9 + 1000x_3 = 98000x_3 / x_6$$
$$x_2 + x_5 = x_1 x_8$$
$$1.12 + 0.13167x_8 - 0.00667x_8^2 \ge x_4 / x_1$$
$$(1,1,0,1,0,85,90,3,1.2,145) \le x$$
$$x \le (2000, 16000, 120, 5000, 2000, 93, 95, 12, 4, 162)$$

THREE-BAR TRUSS DESIGN PROBLEM



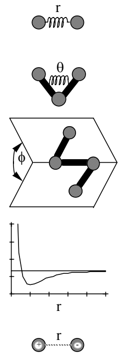
min $\quad 2x_1 + x_2 + \sqrt{2}x_1$

s.t.

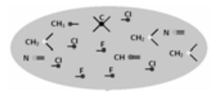
$$3x_2 + 1.932x_3 \le 1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3$$
$$0.634x_1 + 2.828x_3 \le 1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3$$
$$0.5x_1 - 2x_2 \le 1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3$$
$$(x_i - 0.1)(x_i - 0.2)(x_i - 0.3)(x_i - 0.5)(x_i - 0.8)(x_i - 1)(x_i - 1.2) = 0 \quad \text{for} \quad i = 1,2,3$$

MOLECULAR STRUCTURE PREDICTION

$$
\begin{aligned}
E = & \sum_{(ij)\in B} \left\{ k^{b}_{ij} \left(r_{ij} - r^{o}_{ij} \right)^{2} \right\} \\
& + \sum_{(ijk)\in \Theta} \left\{ k^{\theta}_{ijk} \left(\theta_{ijk} - \theta^{o}_{ijk} \right)^{2} \right\} \\
& + \sum_{(ijkl)\in \Phi} \left\{ \left| k^{\phi}_{ijkl} \right| - k^{\phi}_{ijkl} \cos\left(n\phi_{ijkl} \right) \right\} \\
& + \sum_{(ij)\in NB} \left\{ \frac{A_{ij}}{r^{12}_{ij}} - \frac{B_{ij}}{r^{6}_{ij}} \right\} \\
& + \sum_{(ij)\in NB} \frac{q_i q_j}{4\pi\varepsilon_o r_{ij}}
\end{aligned}
$$



INFORMATICS PROBLEMS IN CHEMISTRY, BIOLOGY, AND MEDICINE



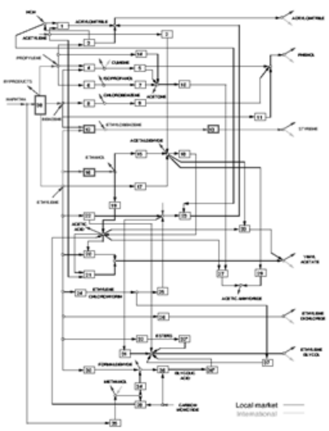
- Design of automotive refrigerants

- Molecular structure determination via X-ray crystallography

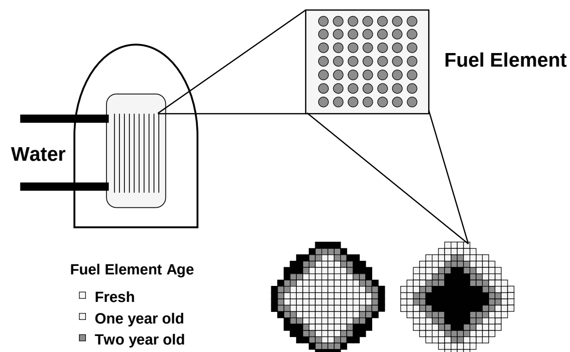
- Breast cancer diagnosis

SUPPLY CHAIN DESIGN & OPERATIONS

- Technology selection
- Facility location
- Capacity expansion
- Blending and pooling
- Uncertainty
- Portfolio optimization
- Very large-scale decision making problems

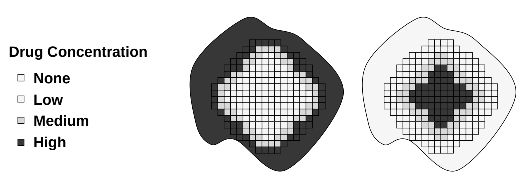


NUCLEAR REACTOR FUEL MANAGEMENT



Reactor Loading Patterns

DRUG RELEASE CONTROL



- Which parts of the tumor to cut?
  - Integer decision variables
- What is the optimal concentration profile?
  - Continuous decision variables

APPLICATIONS OF GLOBAL OPTIMIZATION

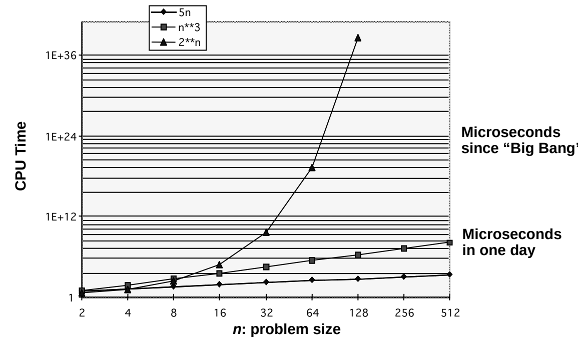
- Engineering design & manufacturing:
  - Product & process design
  - Production planning-scheduling-logistics
- Computational chemical sciences:
  - Chemical & phase equilibria
  - Molecular design
- Informatics problems in biology and medicine
  - Molecular structure prediction
  - Diagnosis

Global optimization is an empowering technology in science and engineering

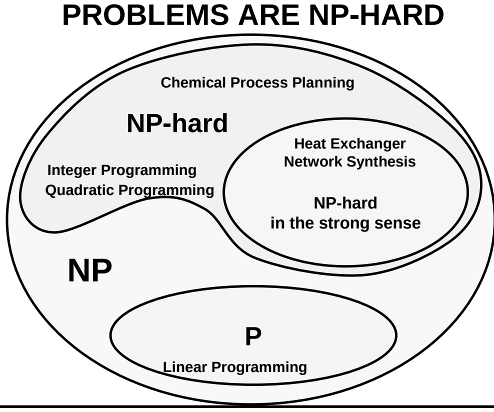
PART 2: ALGORITHMS FOR GLOBAL OPTIMIZATION

- Computational complexity
  - Is it easy or difficult to find global optima?
- Deterministic vs. stochastic
- Branch-and-bound
  - Relaxations
  - Branching
  - Finiteness issues
- Convexification
  - Convex hulls and envelopes
  - Sandwich algorithm for polyhedral relaxations
- Range reduction
  - Optimality-based
  - Feasibility-based

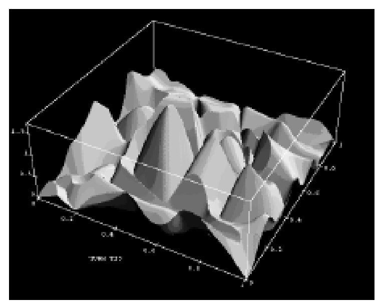
POLYNOMIAL vs. EXPONENTIAL COMPUTATIONAL REQUIREMENTS



GLOBAL OPTIMIZATION PROBLEMS ARE NP-HARD

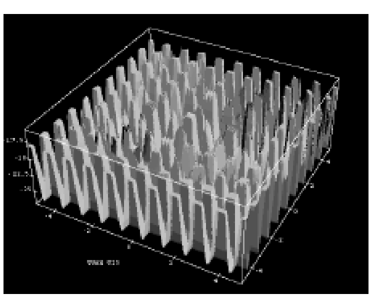


NONCONVEX FUNCTIONS—MOUNTAINS



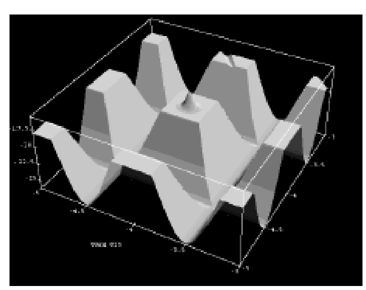
Taken from http://www.ti3.tu-harburg.de/english/globopt.html

NONCONVEX FUNCTIONS—PLATEAUS



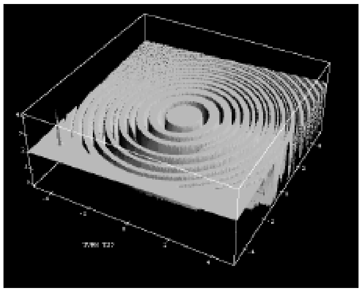
Taken from http://www.ti3.tu-harburg.de/english/globopt.html

NONCONVEX FUNCTIONS—DETAILS OF PLATEAUS



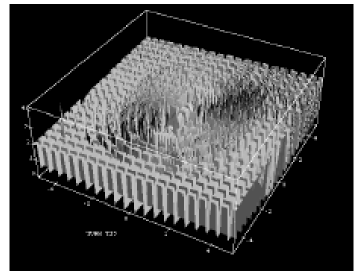
Taken from http://www.ti3.tu-harburg.de/english/globopt.html

NONCONVEX FUNCTIONS— DARTBAORD WITH ARROW



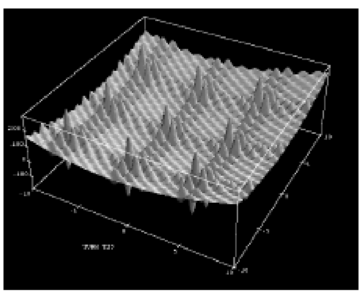
Taken from http://www.ti3.tu-harburg.de/english/globopt.html

NONCONVEX FUNCTIONS— BRYCE CANYON



Taken from http://www.ti3.tu-harburg.de/english/globopt.html

NONCONVEX FUNCTIONS— LEVY'S BENT EGG CARDBOARD



Taken from http://www.ti3.tu-harburg.de/english/globopt.html

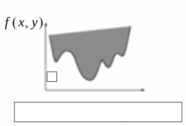
MIXED-INTEGER NONLINEAR PROGRAMMING (MINLP)

$$\begin{aligned} \min \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & x \in R^n, \; y \in Z^p \end{aligned}$$

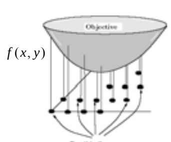
- **Integer variables**
- **Nonlinearities in the objective and constraints**
- **Nonconvexity even when integrality is relaxed**

CHALLENGES IN MIXED-INTEGER NONLINEAR PROGRAMMING

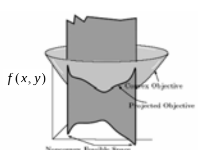
$$\begin{aligned} \min \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \le 0 \\ & x \in R^n, \; y \in Z^p \end{aligned}$$



Multimodal objective



Integrality conditions



Nonconvex constraints

GLOBAL OPTIMIZATION METHODS

- **STOCHASTIC**
  - **Involve random elements (e.g., Monte Carlo Simulations)**
    - **Do *forever*:**
      - **Generate Starting Point**
      - **Local Search**
  - **Converge to global optimum with a probability that approaches 1 as their running time approaches infinity**
- **DETERMINISTIC**
  - **No random elements required**
  - **Finite termination with ε-global optimum (ε>0)**

DETERMINISTIC GLOBAL OPTIMIZATION

- **BRANCH-AND-BOUND**
  - **Implicit enumeration via a tree search**
  - **Divide-and-conquer idea**
- **CONVEXIFICATION**
  - **Outer-approximate nonconvex space by increasingly more accurate convex programs**
- **DECOMPOSITION**
  - **Temporarily fix some variables to exploit problem structure**

⇨ **Horst & Tuy (1996)**

⇨ **Kluwer's (now Springer's) series on "Nonconvex Optimization & Its Applications"**

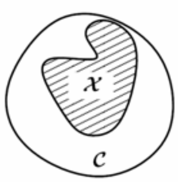
ALGORITHMIC BUILDING BLOCKS

- **Outer approximations of feasible sets**
  - **Convex hulls**
- **Under- and over-estimators of objective functions**
  - **Convex and concave envelopes**
- **Partitioning of feasible set**

CONVEX OUTER APPROXIMATION

Given a nonconvex set $\mathcal{X}$, $\mathcal{C}$ is a convex outer approximation of $\mathcal{X}$ if:

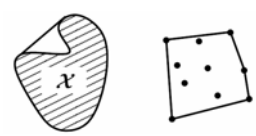
- $\mathcal{C}$ is convex
- if $x \in \mathcal{X}$, then $x \in \mathcal{C}$



CONVEX HULL

Given a nonconvex set $\mathcal{X}$, $\mathcal{C}$ is the convex hull of $\mathcal{X}$ if:

- $\mathcal{C}$ is convex outer approximation of $\mathcal{X}$
- For every $\mathcal{S}$ such that $\mathcal{S}$ is convex outer approximation of $\mathcal{X}$, $\mathcal{C} \subseteq \mathcal{S}$

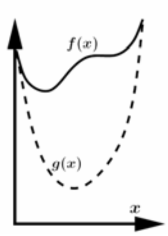


The convex hull is the tightest possible convex outer approximation of a set

CONVEX UNDERESTIMATOR

Given a nonconvex function $f(x)$, $g(x)$ is a convex underestimator of $f(x)$ for $x \in \mathcal{S}$ if

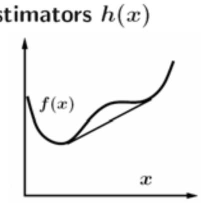
- $g(x)$ is convex
- $g(x) \leq f(x)$ for all $x \in \mathcal{S}$



CONVEX ENVELOPE

Given a nonconvex function $f(x)$, $g(x)$ is the convex envelope of $f(x)$ for $x \in \mathcal{S}$ if

- $g(x)$ is convex underestimator of $f(x)$
- $g(x) \geq h(x)$ for all convex underestimators $h(x)$



The convex envelope is the tightest possible convex underestimator of a function

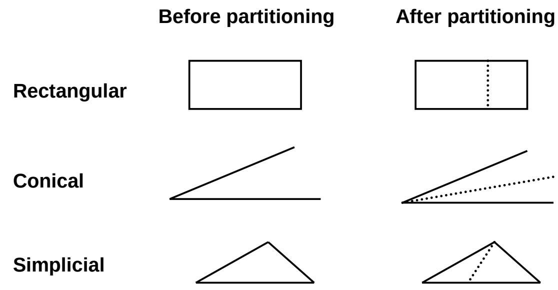
PARTITIONS

Let D be a subset of R^n and I be a finite set of indices. A set $\{D_i : i \in I\}$ of subsets of D is a partition of D if:

1. $D = \cup_{i \in I} D_i$
2. $D_i \cap D_j = \partial D_i \cap \partial D_j \; \forall i, j, \; i > j$

where ∂D_i denotes the boundary of D_i relative to D.

PARTITIONING METHODS



BRANCH-AND-BOUND

Problem to solve: $\{\min f(x) \text{ s.t. } x \in D\}$

1. Start with a relaxed feasible set $D_0 \supseteq D$ and split (partition) D_0 into finitely many subsets $D_i, \; i \in I$.
2. For each subset D_i, determine a lower bound $\beta(D_i)$ and an upper bound $\alpha(D_i)$, satisfying:

$$\beta(D_i) \leq \inf f(D_i \cap D) \leq \alpha(D_i).$$

BRANCH-AND-BOUND

Then, $\beta = \min\{\beta(D_i) : i \in I\}$ and $\alpha = \min\{\alpha(D_i) : i \in I\}$ are overall bounds, that is we have:

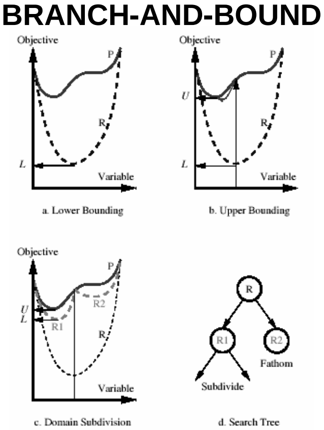
$$\beta \leq \min f(D) \leq \alpha.$$

3. If $\alpha = \beta$ (or $\alpha - \beta \leq \epsilon, \; \epsilon > 0$ prescribed), then stop

BRANCH-AND-BOUND

4. Otherwise, select some subsets D_i and partition these selected subsets further in order to obtain a refined partition of D_0. Determine new better bounds on the new partition elements and repeat the process.

BRANCH-AND-BOUND



FROM PROTOTYPE TO ALGORITHMS

- Branch-and-bound is a strategy
- To obtain a specific branch-and-bound algorithm, one must specify:
  - Relaxation technique
  - Branching strategy
  - Node selection rule

CONVERGENCE

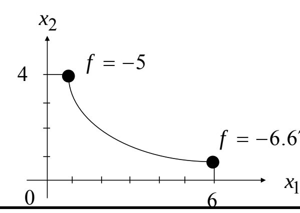
- Consistent partitioning
  - Any open partition can be further refined
  - As refinement progresses, the lower bound converges to the nonconvex problem value
- Bound improving node selection rule
  - Every finite number of steps, a node with the least lower bound is selected
- A branch-and-bound algorithm with a consistent partitioning scheme and a bound improving node selection rule, converges to a global optimum
- Exhaustiveness
  - For every sequence of partitions, the feasible region reduces to a point
  - Not necessary for convergence but most branch-and-bound algorithms satisfy it

BOUNDING METHODS

- Separable Programming
- Factorable Programming
- Lagrangian Relaxation
- Lipschitzian Bounds
- Interval Arithmetic
- D.C. Programming
- Convex Extensions/Envelopes

BOUNDS VIA SEPARABLE PROGRAMMING TECHNIQUES

$$\begin{aligned} \min \quad & f = -x_1 - x_2 \\ \text{s.t.} \quad & x_1 x_2 \le 4 \\ & 0 \le x_1 \le 6 \\ & 0 \le x_2 \le 4 \end{aligned}$$



SEPARABLE REFORMULATION

$$\begin{aligned} \min \quad & f = -x_1 - x_2 \\ \text{s.t.} \quad & x_3^2 - x_1^2 - x_2^2 \le 8 \\ & x_3 = x_1 + x_2 \\ & 0 \le x_1 \le 6 \\ & 0 \le x_2 \le 4 \\ & 0 \le x_3 \le 10 \end{aligned}$$

IN 2-D

$$\begin{aligned} \min \quad & f = -x_1 - x_2 \\ \text{s.t.} \quad & (x_1 + x_2)^2 - x_1^2 - x_2^2 \le 8 \\ & 0 \le x_1 \le 6 \\ & 0 \le x_2 \le 4 \end{aligned}$$

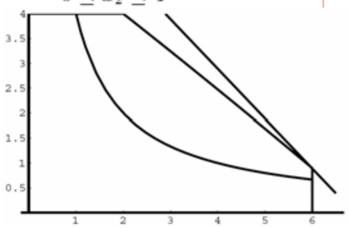
CONVEX RELAXATION

$$\min \quad -x_1 - x_2$$

$$\text{s.t.} \quad (x_1 + x_2)^2 - 6x_1 - 4x_2 \le 8$$

$$0 \le x_1 \le 6$$

$$0 \le x_2 \le 4$$

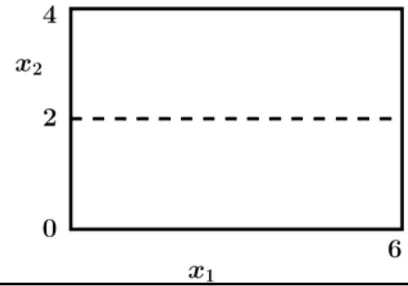


Solution: $x_1 = 6, x_2 = 0.89, L = -6.89$

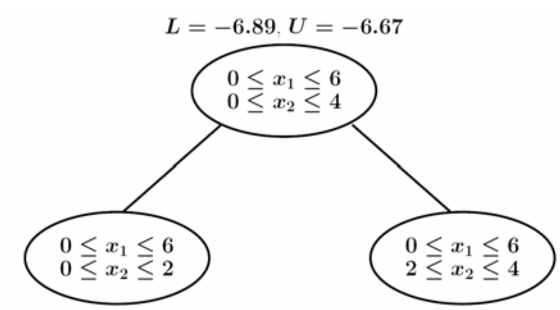
Local search from this starting point: $U = -6.67$

SELECTION OF BRANCHING VARIABLE

- **We must branch on x_2.**
- **Why?**

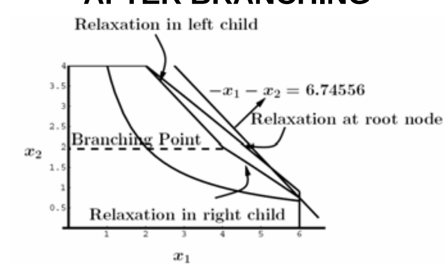


BRANCH-AND-BOUND TREE



We now solve the left as well as the right child

AFTER BRANCHING



Left Child

$\min -x_1 - x_2$

s.t. $(x_1 + x_2)^2 - 6x_1 - 2x_2 \le 8$

$0 \le x_1 \le 6$

$0 \le x_2 \le \boxed{2}$

Right Child

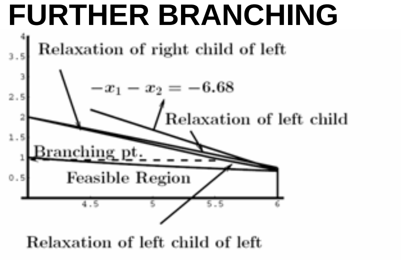
$\min -x_1 - x_2$

s.t. $(x_1 + x_2)^2 - 6x_1 - 6x_2 \le 0$

$0 \le x_1 \le 6$

$\boxed{2} \le x_2 \le 4$

FURTHER BRANCHING



Left of Left Child

$\min -x_1 - x_2$

s.t. $(x_1 + x_2)^2 - 6x_1 - x_2 \le 8$

$0 \le x_1 \le 6$

$0 \le x_2 \le \boxed{1}$

Right of Left Child

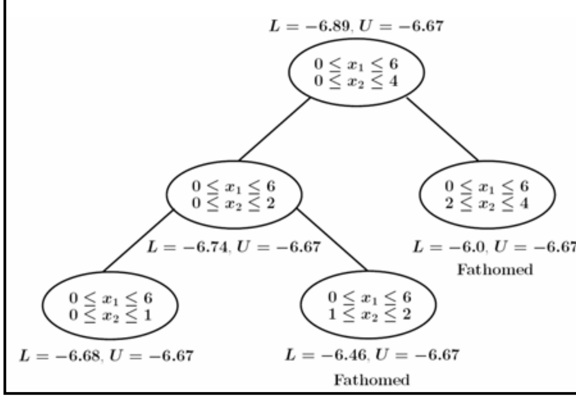
$\min -x_1 - x_2$

s.t. $(x_1 + x_2)^2 - 6x_1 - 3x_2 \le 6$

$0 \le x_1 \le 6$

$\boxed{1} \le x_2 \le \boxed{2}$

BRANCH-AND-BOUND TREE



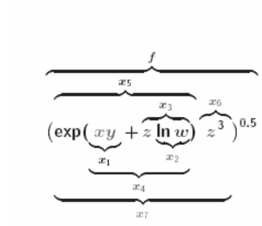
PROCEDURE FOR BOUNDING SEPARABLE PROGRAMS

- A function is separable if it can be expressed as a sum of univariate functions
- An optimization problem is separable if all its constraints and objective function are separable
- Bounding procedure:
  - Construct underestimators of each of the univariate functions
    - Often an easy task
      - A convex function underestimates itself
      - A concave function can be underestimated over an interval by a secant
      - For a concavoconvex function, first identify a point of inflection
  - Sum up these underestimators to obtain an underestimator of the original separable function

PROCEDURE FOR BOUNDING FACTORABLE PROGRAMS

Introduce variables for intermediate quantities whose envelopes are not known

Example: $f(x,y,z,w) = \sqrt{\exp(xy + z\ln w)z^3}$



$x_1 = xy$
$x_2 = \ln(w)$
$x_3 = zx_2$
$x_4 = x_1 + x_3$
$x_5 = \exp(x_4)$
$x_6 = z^3$
$x_7 = x_5 x_6$
$f = \sqrt{x_7}$

ENVELOPES FOR BILINEAR TERMS

- Convex envelope

$$xy \geq \max\left\langle \begin{matrix} x^U y + y^U x - x^U y^U \\ x^L y + y^L x - x^L y^L \end{matrix} \right\rangle$$

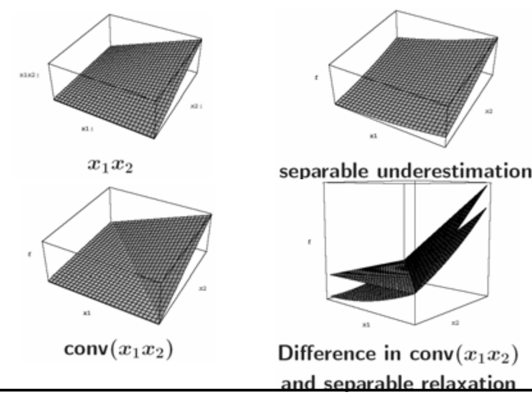
- Concave envelope

$$xy \leq \min\left\langle \begin{matrix} x^U y + y^L x - x^U y^L \\ x^L y + y^U x - x^L y^U \end{matrix} \right\rangle$$

- Separable reformulation

$$xy = \frac{1}{4}\{(x+y)^2 - (x-y)^2\}$$

RELAXING BILINEAR TERMS



MCCORMICK'S PROCEDURE FOR BOUNDING FACTORABLE PROGRAMS

- Use the above procedure for bounding factorable programs with one exception:
  - Do not introduce intermediate variables
- Leads to non-differentiable lower bounding program that may be weaker than the one obtained after introduction of intermediate variables

INTERVAL ARITHMETIC

1. addition:
   $[a,b] + [c,d] = [a+c, b+d]$
2. substraction:
   $[a,b] - [c,d] = [a-d, b-c]$
3. multiplication:
   $[a,b][c,d] = [\min(ac,ad,bc,bd), \max(ac,ad,bc,bd)]$
4. division:
   $\frac{[a,b]}{[c,d]} = \begin{cases} \infty & \text{if } 0 \in [c,d] \\ [a,b][\frac{1}{d},\frac{1}{c}] & \text{else} \end{cases}$

BOUNDS BASED ON INTERVAL ARITHMETIC

$$f = xy + z^2$$

where

$$0 \le x \le 1$$
$$1 \le y \le 10$$
$$0 \le z \le 10$$

$$f \in [0,1][1,10] + [0,10][0,10] =$$

$$[0,10] + [0,100] = [0,110]$$

$$\Rightarrow \; 0 \le f \le 110$$

D.C. PROGRAMMING BOUNDS

A function f is d.c. (difference of two convex functions) if there exist functions p and q such that $f = p - q$.

For given p and q, all that is needed to construct a convex underestimator of f is an underestimator of the concave function $-p$.

One possible d.c. decomposition of $f(x_1, x_2, \ldots, x_n)$ is:

$$f = f + \mu \sum_i x_i^2 - \mu \sum_i x_i^2$$

for a sufficiently large positive value of μ for which the eigenvalues of the Hessian of the first two terms of the above sum become positive.

Used in the QAP and αBB among others.

BOUNDS FROM LAGRANGIAN DUAL

Consider the problem:

$$\begin{array}{llll} \text{(P)} & \min & f(x) & \\ & \text{s.t.} & g_i(x) \le 0, & i = 1, m \\ & & h_j(x) = 0, & j = 1, l \\ & & x \in X & \end{array}$$

Lagrangian dual problem:

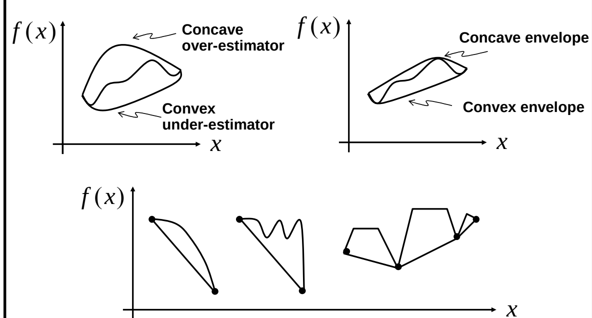
$$\begin{array}{llll} \text{(D)} & \max & \theta(u, v) & \\ & \text{s.t.} & u_i \ge 0, & i = 1, m \end{array}$$

where $\theta(u,v) = \inf_{x \in X} \{ f(x) + \sum_{i=1}^m u_i g_i(x) + \sum_{j=1}^l v_j h_j(x) \}$

Weak duality theorem

Let x be a feasible solution to (P) that is $x \in X, g(x) \le 0, h(x) = 0$. Also, let (u, v) be such that $u \ge 0$. Then $f(x) \ge \theta(u, v)$.

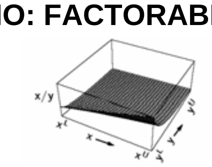
CONVEX/CONCAVE ENVELOPES OFTEN FINITELY GENERATED



TWO-STEP CONVEX ENVELOPE CONSTRUCTION VIA CONVEX EXENSIONS

1. **Identify generating set** **(Tawarmalani and Sahinidis, 2002):**
   - **Key result: A point in set X is *not* in the generating set if it is not in the generating set over a neighborhood of X that contains it**
2. **Use disjunctive programming techniques to construct epigraph over the generating set**
   - **Rockafellar (1970)**
   - **Balas (1974)**

RATIO: FACTORABLE RELAXATION



$$z \ge x/y$$
$$y^L \le y \le y^U$$
$$x^L \le x \le x^U$$

$$z \ge x/y$$
$$x^L \le x \le x^U$$
$$y^L \le y \le y^U$$

cross-multiplying →

$$zy \ge x$$
$$y^L \le y \le y^U$$
$$x^L/y^U \le z \le x^U/y^L$$
$$x^L \le x \le x^U$$

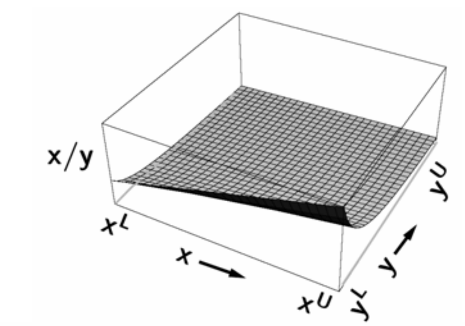
Relaxing ↓

$$zy - (z - x^L/y^U)(y - y^U) \ge x$$
$$zy - (z - x^U/y^L)(y - y^L) \ge x$$
$$y^L \le y \le y^U$$
$$x^L \le x \le x^U$$

← Simplifying

$$z \ge (xy^U - yx^L + x^L y^U)/{y^U}^2$$
$$z \ge (xy^L - yx^U + x^U y^L)/{y^L}^2$$
$$y^L \le y \le y^U$$
$$x^L \le x \le x^U$$

RATIO: THE GENERATING SET



$$G^{\text{epi}}(x/y) = \{(x,y) \mid x \in \{x^L, x^U\}\}$$

ENVELOPES OF MULTILINEAR FUNCTIONS

- Multilinear function over a box

$$M(x_1,\ldots,x_n) = \sum_t a_t \prod_{i=1}^{p_t} x_i, \; -\infty < L_i \le x_i \le U_i < +\infty, \; i = 1,\ldots,n$$

- Generating set

$$\text{vert}\left(\prod_{i=1}^{n}[L_i, U_i]\right)$$

- Polyhedral convex encloser follows trivially from polyhedral representation theorems

PRODUCT DISAGGREGATION

Consider the function:

$$\phi(x; y_1,\ldots,y_n) = a_0 + \sum_{k=1}^{n} a_k y_k + x b_0 + x\sum_{k=1}^{n} b_k y_k$$

Let

$$H = [x^L, x^U] \times \prod_{k=1}^{n}[y_k^L, y_k^U]$$

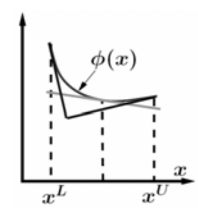
Then

$$\text{convenv}_H\, \phi = a_0 + \sum_{k=1}^{n} a_k y_k + x b_0 + \sum_{k=1}^{n} \text{convenv}_{[y_k^L, y_k^U]\times[x^L, x^U]}(b_k y_k x)$$

Disaggregated formulations are tighter

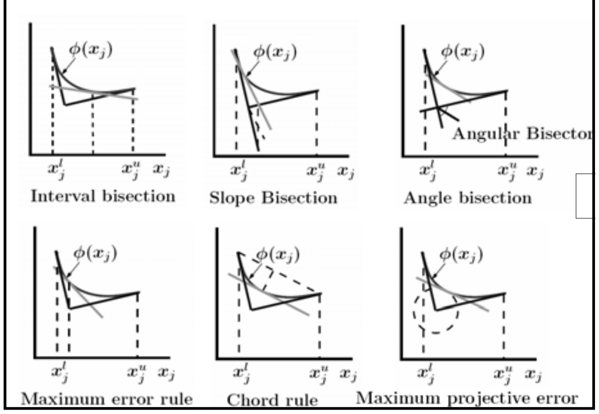
POLYHEDRAL OUTER APPROXIMATION

- Convex NLP solvers are not as robust as LP solvers
- Linear programs can be solved efficiently
- Outer-approximate convex relaxation by polyhedron

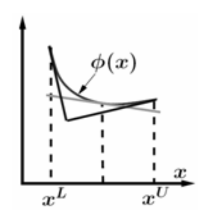


Enjoys quadratic convergence

TANGENT LOCATION RULES



EXPLOITING CONVEXITY

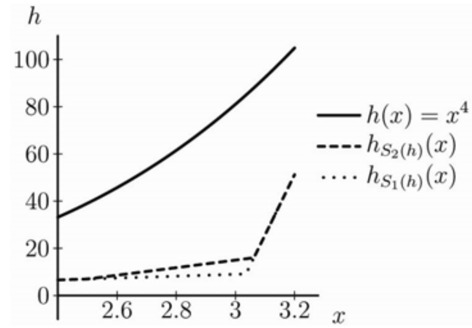


- Polyhedral relaxations of univariate functions facilitate reliable lower bounding via fast LP routines
- Outstanding issues:
  - Lower bound itself weakens
  - Effect of functional decomposition
  - Polyhedral relaxations of convex multivariate functions
    - Gruber (1993), Böröczky and Reitzner (2004)

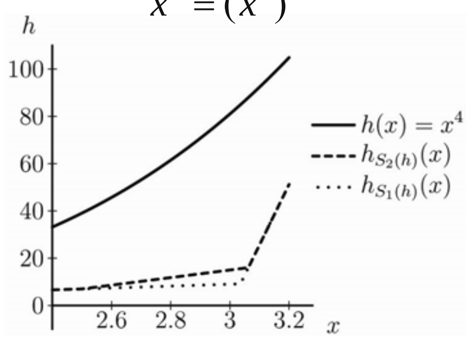
RECURSIVE FUNCTIONAL COMPOSITIONS

- **Consider *h*=*g*(*f*), where**
  - ***g* and *f* are multivariate convex functions**
  - ***g* is non-decreasing in the range of each nonlinear component of *f***
- ***h* is convex**
- **Two outer approximations of the composite function *h*:**
  - **S1: a single-step procedure that constructs supporting hyperplanes of *h* at a predetermined number of points**
  - **S2: a two-step procedure that constructs supporting hyperplanes for *g* and *f* at corresponding points**
- **Theorem: S2 is contained in S1**
  - **If *f* is affine, S2=S1**
  - **In general, the inclusion is strict**

STRICT INCLUSION

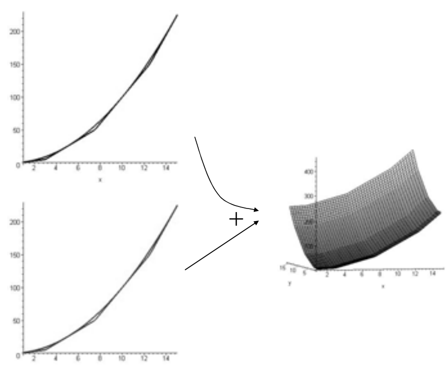


$$x^4 = (x^2)^2$$



Outer-approximating x^4 at $x = 1$ and $x = 4$

OUTER APPROXIMATION OF x^2+y^2



TWO-STEP IS BETTER

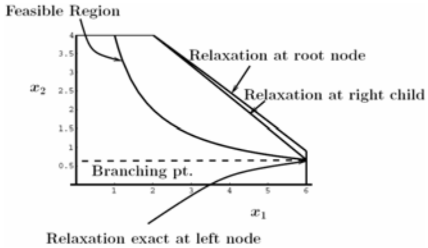
- **Theorem: An exponential number of supporting hyperplanes in S1 may be required to cover S2**

  $h = f_1(x_1) + \ldots + f_m(x_m)$ where each f_i is strictly convex
- **Separable functions are quite common in nonconvex optimization**
- **S2 has the potential of providing much tighter polyhedral outer approximations than S1 with a comparable number of supporting hyperplanes**

AUTOMATIC DETECTION AND EXPLOITATION OF CONVEXITY

- **Composition rule: *h* = *g*(*f*), where**
  - ***g* and *f* are multivariate convex functions**
  - ***g* is non-decreasing in the range of each nonlinear component of *f***
- **Subsumes many known rules for detecting convexity/concavity**
  - ***g* univariate convex, *f* linear**
  - **$g=\max\{f_1(x), \ldots, f_m(x)\}$, each f_i convex**
  - **$g=\exp(f(x))$**
  - **…**
- **Automatic exploitation of convexity is not essential for constructing polyhedral outer approximations in these cases**
  - **However, $\text{logexp}(x) = \log(e^{x_1} + \ldots + e^{x_n})$**
  - **CONVEX_EQUATIONS modeling language construct**

BRANCHING ON THE ENCUMBENT



Left Child

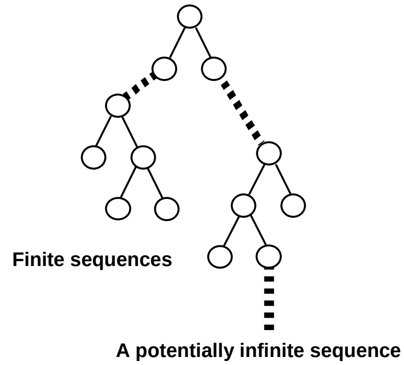
$$\min -x_1 - x_2$$
$$\text{s.t.}\ (x_1+x_2)^2 - 6x_1 - \frac{2}{3}x_2 \le 8$$
$$0 \le x_1 \le 6$$
$$0 \le x_2 \le 0.89$$

Right Child

$$\min -x_1 - x_2$$
$$\text{s.t.}\ (x_1+x_2)^2 - 6x_1 - \frac{14}{3}x_2 \le \frac{16}{3}$$
$$0 \le x_1 \le 6$$
$$0.89 \le x_2 \le 4$$

Termination after first branching

FINITE VERSUS CONVERGENT BRANCH-AND-BOUND ALGORITHMS

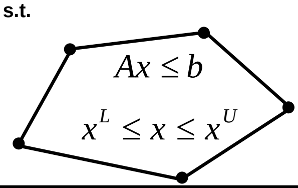


SEPARABLE CONCAVE MINIMIZATION

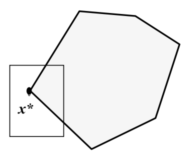
Consider $f_k(x_k)$ concave, $\forall k$.

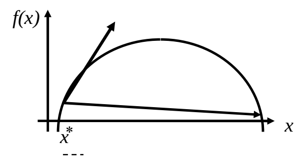
$$\min\ f(x) = \sum_k f_k(x_k)$$

s.t.



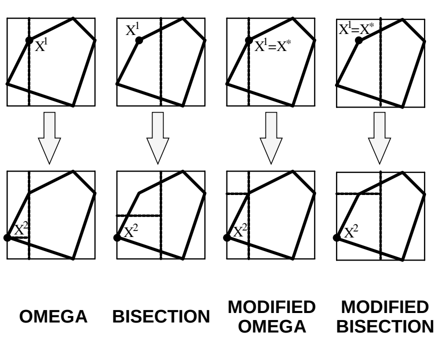
FINITE BRANCHING RULE





- **Branching variable selection:**
  - Typically, select variable with largest underestimating gap
  - Occasionally, select variable corresponding to largest edge
- **Branching point selection:**
  - Typically, at the midpoint (exhaustiveness)
  - When possible, at the best currently known solution
    - Makes underestimators exact at the candidate solutions
- **Finite isolation of global optimum**
- **Ascend directions of LP also ascend directions of QP**

COMPOUND RULES



B&B HISTORICAL NOTES

- **Branch-and-bound**
  - Land and Doig (1960); Dakin (1965)
  - Falk and Soland (1969)
  - Horst and Tuy (1996)
- **Separable programming bounds**
  - Falk and Soland (1969)
- **Factorable programming bounds**
  - McCormick (1976); non-differentiable
  - Ryoo and Sahinidis (1995, 1996); differentiable
- **Extensions, sandwich relaxations, cuts**
  - Tawarmalani and Sahinidis (2004, 2005)
- **Finite branching rules**
  - Shectman and Sahinidis (1998)

PART 3: BRANCH-AND-REDUCE

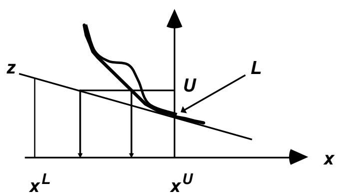
- The smaller the domain, the faster branch-and-bound converges
  - Tighter lower bounds
  - Fewer partitions
- Range reduction techniques
  - Based on marginals
  - Based on probing
    - Fixing variables at bounds, followed by marginals-based reduction
    - Solving LPs/NLPs to minimize/maximize problem variables over the feasible region or an outer approximation of it
  - Via interval arithmetic operations on the problem constraints

MARGINALS-BASED REDUCTION

- Economic interpretation of LP duals
- Economic interpretation of Lagrange multipliers
- Value function of LP
- Value function of convex NLP
- Derivation of reduction test

MARGINALS-BASED RANGE REDUCTION

Relaxed Value Function



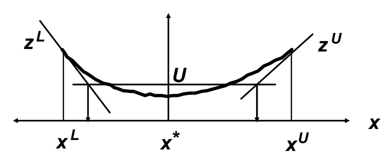
If a variable goes to its upper bound at the relaxed problem solution, this variable's lower bound can be improved

PROBING

Q. What if a variable does not go to a bound?

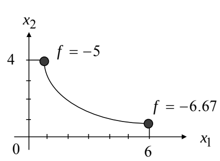
A. Use probing: temporarily fix variable at a bound or minimize/maximize this variable over the problem constraints

Relaxed Value Function



ILLUSTRATIVE EXAMPLE

$$\left.\begin{array}{ll} \min & f = -x_1 - x_2 \\ \text{s.t.} & x_1 x_2 \le 4 \\ & 0 \le x_1 \le 6 \\ & 0 \le x_2 \le 4 \end{array}\right\} (P)$$



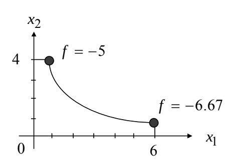
Reformulation:

$$\begin{array}{ll} \min & f = -x_1 - x_2 \\ \text{s.t.} & x_3^2 - x_1^2 - x_2^2 \le 8 \\ & x_3 = x_1 + x_2 \\ & 0 \le x_1 \le 6 \\ & 0 \le x_2 \le 4 \\ & 0 \le x_3 \le 10 \end{array}$$

Relaxation:

$$\left.\begin{array}{ll} \min & -x_1 - x_2 \\ \text{s.t.} & x_3^2 - 6x_1 - 4x_2 \le 8 \\ & x_3 = x_1 + x_2 \\ & 0 \le x_1 \le 6 \\ & 0 \le x_2 \le 4 \\ & 0 \le x_3 \le 10 \end{array}\right\} (R)$$

ILLUSTRATIVE EXAMPLE



Solution of R**:** $x_1 = 6, x_2 = 0.89, L = -6.89, \lambda_1 = 0.2$

Local Solution of P **with MINOS :** $U = -6.67$

Range reduction : $x_1^L \leftarrow x_1^U - (U - L)/\lambda_1 = 4.86$

Probing (Solve R with $x_2 \le 0$) : $L = -6, \lambda_2 = 1 \Rightarrow x_2^L = 0.67$

Update R with $x_1 \ge 4.86, x_2 \ge 0.67$

Solution is : $L = -6.67$

$\therefore$ **Proof of globality with NO Branching!!**

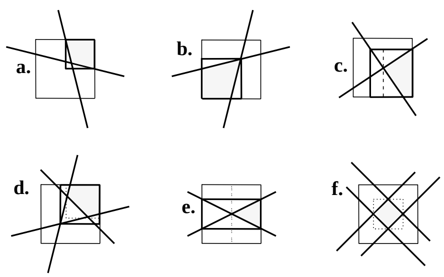
FEASIBILITY-BASED REDUCTION

$$
\begin{aligned}
\min \quad & f = -x_1 - x_2 \\
\text{s.t.} \quad & x_1 x_2 \le 4 \\
& 1 \le x_1 \le 6 \\
& 2 \le x_2 \le 4
\end{aligned}
$$

FEASIBILITY-BASED REDUCTION

- **Minimize and maximize each problem variable over the constraint set**
  - Requires the solution of $2n$ problems
  - Each problem may be a nonconvex NLP
- **Use interval arithmetic operations on one nonlinear constraint and one variable at a time ("poor man's" NLPs)**
  - Propagate bounds of variables
  - Constraint satisfaction techniques
- **Solve minimization/maximization problems over a polyhedral outer approximation of the constraint set**
  - May still be expensive
- **Solve minimization/maximization LPs approximately ("poor man's LPs")**

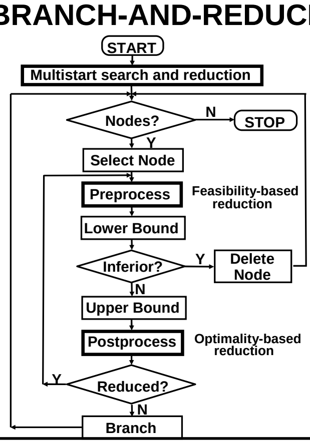
POOR MAN'S LPs AND NLPs



DOMAIN REDUCTION THEORY in T&S book (2002)

- **Draw inferences about several reduction problems from solutions of problems solved in the course of branch-and-bound**
- **Implied results:**
  - Monotone complementarity bounds for convex programs
    - Mangasarian and McLinden, 1985
  - Linearity-based tightening in integer programming
    - Andersen and Andersen, 1995
  - Marginals-based reduction
    - Ryoo and Sahinidis, 1995
  - Branch and Contract
    - Zamora and Grossmann, 1999
- **New reduction strategies**
  - Learning heuristic improves branching decisions

BRANCH-AND-REDUCE



TINY TEST PROBLEMS

Ex.	Cons.	Vars.	Source/In	Description
1	1	2	Sahinidis & Grossmann	bilinear constraint
2	3	3	Liebman et al. (GINO)	design of a water pumping system
3	7	10	Liebman et al. (GINO)	alkylation process optimization
4	1	3	Liebman et al. (GINO)	design of insulated tank
5	3	5	Liebman et al. (GINO)	heat exchanger network design
6	3	3	Liebman et al. (GINO)	chemical equilibrium
7	7	10	Liebman et al. (GINO)	pooling problem
8	2	2	Swaney	bilinear and quadratic constraints
9	1	2	Swaney	bilinear constraints and objective
10	1	2	Soland	nonlinear equality constraint
11	2	3	Westerberg & Shah	bilinearities, economies of scale
12	3	4	Stephanopoulos & Westerberg	design of two-stage process systems
13	3	2	Kocis & Grossmann	MINLP, process synthesis
14	10	7	Yuan et al.	MINLP, process synthesis
15	6	5	Kocis & Grossmann	MINLP, process synthesis
16	9	12	Floudas & Ciric	heat exchanger network synthesis
17	2	2	GINO	design of a reinforced concrete beam
18	4	2	Visweswaran & Floudas	quadratically constrained LP
19	2	2	Manousiouthakis & Sourlas	quadratically constrained QP
20	6	5	Manousiouthakis & Sourlas	reactor network design
21	6	5	Stephanopoulos & Westerberg	design of three-stage process system
22	5	2	Kalantari & Rosen	linearly constrained concave QP
23	2	2	Al-Khayyal & Falk	biconvex program
24	4	2	Thakur	linearly constrained concave QP
25	4	2	Falk & Soland	nonlinear fixed charge problem

STANDARD BRANCH-AND-BOUND

Ex.	N_{tot}	N_{opt}	N_{mem}	T
1	3	1	2	0.8
2	1007	1	200	210
3	2122*	1	113*	1245*
4	17	1	5	6.7
5	1000*	1	1000*	417*
6	1	1	1	0.3
7	205	1	37	43
8	43	1	8	1
9	2192*	1	1000*	330*
10	1	1	1	0.4
11	81	1	24	19
12	3	1	2	0.6
13	7	2	3	1.3
14	7	3	3	3.4
15	15	8	5	3.4
16	2323*	1	348*	1211*
17	1000*	1	1001*	166*
18	1	1	1	0.5
19	85	1	14	11.4
20	3162*	1	1001*	778*
21	7	1	4	1.2
22	9	1	4	1.2
23	75	6	13	11.7
24	7	3	2	1.5
25	17	9	9	2.9

N_{tot} Total number of nodes
N_{opt} Node where optimum found
N_{mem} Max. no. nodes in memory
T CPU sec (SPARC 2)

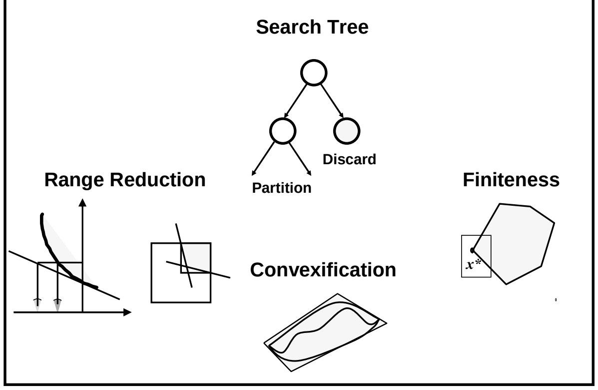
*: Did not converge within limits of
T ≤ 1200 (≈20 min) and N_{mem} ≤ 1000 nodes

- Standard branch-and-bound converges very slowly
- It is not necessarily finite
- Tighter relaxations needed

REDUCTION BENEFITS

	BRANCH-AND-BOUND				BRANCH-AND-REDUCE No probing				BRANCH-AND-REDUCE With Probing			
Ex.	N_{tot}	N_{opt}	N_{mem}	T	N_{tot}	N_{opt}	N_{mem}	T	N_{tot}	N_{opt}	N_{mem}	T
1	3	1	2	0.8	1	1	1	0.5	1	1	1	0.7
2	1007	1	200	210	1	1	1	0.2	1	1	1	0.3
3	2122*	1	113*	1245*	31	1	7	20	9	1	5	48
4	17	1	5	6.7	3	1	2	0.4	1	1	1	0.3
5	1000*	1	1000*	417*	5	1	3	1.5	5	1	3	2.4
6	1	1	1	0.3	1	1	1	0.3	1	1	1	0.3
7	205	1	37	43	25	1	8	5.4	7	1	2	5.8
8	43	1	8	10	1	1	1	0.8	1	1	1	0.8
9	2192*	1	1000*	330*	19	1	8	5.4	13	1	4	7
10	1	1	1	0.4	1	1	1	0.4	1	1	1	0.4
11	81	1	24	19	3	1	2	0.6	1	1	1	0.7
12	3	1	2	0.6	1	1	1	0.2	1	1	1	0.2
13	7	2	3	1.3	3	1	2	0.7	1	1	1	0.7
14	7	3	3	3.4	7	3	3	2.7	3	3	2	3
15	15	8	5	3.4	1	1	1	0.3	1	1	1	0.3
16	2323*	1	348*	1211*	1	1	1	2.2	1	1	1	2.4
17	1000*	1	1001*	166*	1	1	1	3.7	1	1	1	4
18	1	1	1	0.5	1	1	1	0.5	1	1	1	0.6
19	85	1	14	11.4	9	1	4	1.8	1	1	1	1.4
20	3162*	1	1001*	778*	47	1	12	16.7	23	1	5	15.4
21	7	1	4	1.2	1	1	1	0.5	1	1	1	0.5
22	9	1	4	1.2	3	1	2	0.4	3	1	2	0.5
23	75	6	13	11.7	47	1	9	6.5	7	1	4	5
24	7	3	2	1.5	3	1	2	0.5	3	1	2	0.6
25	17	9	9	2.9	5	1	3	0.8	5	1	3	1

BRANCH-AND-REDUCE



Branch-And-Reduce Optimization Navigator

Components
- Modeling language
- Preprocessor
- Data organizer
- I/O handler
- Range reduction
- Solver links
- Interval arithmetic
- Sparse matrix routines
- Automatic differentiator
- IEEE exception handler
- Debugging facilities

Capabilities
- Core module
  - Application-independent
  - Expandable
- Fully automated MINLP solver
- Application modules
  - Multiplicative programs
  - Indefinite QPs
  - Fixed-charge programs
  - Mixed-integer SDPs
  - ...
- Solve relaxations using
  - CPLEX, MINOS, SNOPT, OSL, SDPA, …

- First on the Internet in March 1995
- On-line solver between October 1999 and May 2003
  - Solved eight problems a day
- Available under GAMS and AIMMS

BARON MODELING LANGUAGE

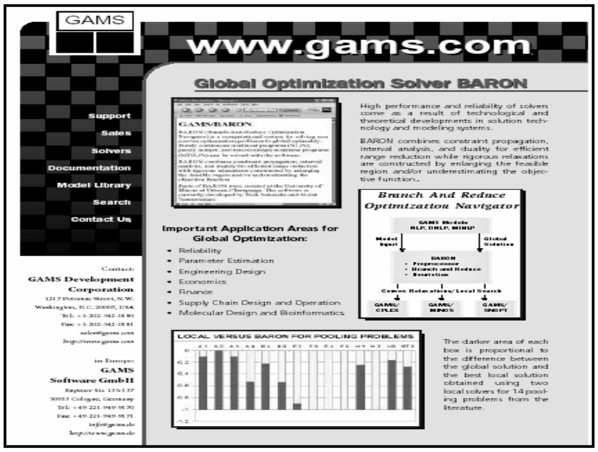
```
// Design of an insulated tank

OPTIONS{
nlpdolin: 1;
dolocal: 0; numloc: 3;
brvarstra: 1; nodesel: 0;
nlpsol: 4; lpsol: 3;
pdo: 1; pxdo: 1; mdo: 1;
}
MODULE: NLP;
```

Relaxation Strategy
Local Search Options
B&B options
Solver Links
Domain Reduction Options

```
// INTEGER_VARIABLE y1;
POSITIVE_VARIABLES x1, x2, x4;
VARIABLE x3;

LOWER_BOUNDS{x2:14.7;  x3:-459.67;}

UPPER_BOUNDS{
x1:   15.1;  x2:   94.2;
x3:   80.0;  x4: 5371.0;
}

EQUATIONS e1, e2;
e1: x4*x1 - 144*(80-x3) >= 0;
e2: x2-exp(-3950/(x3+460)+11.86) == 0 ;
OBJ: minimize 400*x1^0.9 + 1000
    + 22*(x2-14.7)^1.2+x4;
```

SAMPLE INPUT FILE

```
// mixed-integer nonlinear program
// Source:  M. Duran & I.E. Grossmann,
// "An outer approximation algorithm for a class of mixed integer nonlinear
//  programs," Mathematical Programming, 36:307-339, 1986.

MODULE: NLP;

BINARY_VARIABLES y1, y2, y3;

POSITIVE_VARIABLES x1, x2, x3;

UPPER_BOUNDS{ x1: 2 ;  x2: 2 ;  x3: 1 ; }

EQUATIONS c1, c2, c3, c4, c5, c6;

c1: 0.8*log(x2 + 1) + 0.96*log(x1 - x2 + 1) - 0.8*x3 >= 0 ;
c2: log(x2 + 1) + 1.2*log(x1 - x2 + 1) - x3 - 2*y3 >= -2 ;
c3: x2 - x1 <= 0 ;
c4: x2 - 2*y1 <= 0 ;
c5: x1 - x2 - 2*y2 <= 0 ;
c6: y1 + y2 <= 1 ;

OBJ: minimize
    5*y1 + 6*y2 + 8*y3 + 10*x1 - 7*x3 - 18*log(x2 + 1)  - 19.2*log(x1 - x2 + 1) + 10;
```

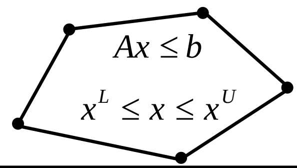


SEPARABLE CONCAVE MINIMIZATION

Consider $f_k(x_k)$ concave, $\forall k.$

$$\min f(x) = \sum_k f_k(x_k)$$

s.t.



PHILLIPS AND ROSEN PROBLEMS

			GOP (1993) E=1% (relative) HP 730		P&R (1990) E=0.1% (relative) CRAY 2 (parallel)			BARON (1996) ε=.000001 (absolute) IBM RS/6000 Power PC		
m	n	k	avg	std dev	min	avg	max	min	avg	max
20	25	0	0.5	0.01	1	2	4	0.3	0.4	0.5
20	25	50	2	2	1	1	1	1	1	1
20	25	100	17	20	1	2	3	1	2	3
20	25	200	33	28	2	7	17	2	4	6
20	25	400	82	58	7	14	32	4	10	16
20	50	0	0.6	0.01	3	6	13	1	1	1
20	50	50	17	31	1	2	3	2	2.5	3
20	50	100	47	49	2	5	14	2	4	7
20	50	200	109	80	4	9	28	4	8	19
20	50	400			20	32	45	11	20	48
40	25	0	0.5	0.02				0.3	0.4	0.4
40	25	50	1	0.6				1	1	1
40	25	100	3	4				1	2	3
40	25	200	25	26				2	4	5
40	25	400						6	15	22
50	100	0						6	7	14
50	100	50						8	12	18
50	100	100						9	17	27
50	100	200						14	65	160
50	100	400						131	345	663

- **(m, n/k) = number of constraints, concave/linear variables.**
- **HP 730 is 3-4 times faster than IBM RS/6000 Power PC.**
- **CRAY 2 is 10+ times faster than IBM RS/6000 Power PC.**

MULTIPLICATIVE PROGRAMS

$$\min \quad \sum_{j=1}^{t} \prod_{i=1}^{p_i} \left(c_{ij}^T x + c_{ij0} \right)$$

$$\text{s.t.} \quad Ax \le b$$

$$x \in \Re^n, \; A \in \Re^{m \times n}$$

PRACTICAL APPLICATIONS

- **Micro-economics (Henderson and Quandt, 1971)**
- **Plant Layout Design (Quesada and Grossman, 1994)**
- **Multi-stage Financial Planning (Maranas *et al*., 1997)**
- **Multiple Objective Decision Making (Lai and Hwang, 1994)**
- **Data Mining/Pattern Recognition (Bennet and Mangasarian, 1994)**
- **Global Tree Optimization (Bennet, 1994)**

SPECIAL CASES of GLMP

- **Linear Multiplicative Programs (LMP):**

$$f(x) = \prod_{j=1}^{p} f_j(x)$$

- **Quadratic Programs (QP):**

$$\begin{aligned} f(x) &= x^T Qx + c^T x + d \\ &= \sum_{i=1}^{n} a_{ii} x_i^2 + \sum_{i=1}^{n-1} \sum_{j>i} 2 a_{ij} x_i x_j + c^T x + d \end{aligned}$$

- **Bilinear Programs (BLP):**

$$\begin{aligned} f(x) &= p^T x + x^T Qy + q^T y + d \\ &= p^T x + \sum_{i=1}^{n} \sum_{J=1}^{M} a_{ij} x_i x_j + q^T y + d \end{aligned}$$

0-1 MULTILINEAR PROGRAMS

$$\text{min} \quad f_0(x) = \sum_{k=1}^{p_0} c_k \prod_{j \in N_{0k}} x_j$$

$$\text{s.t.} \quad f_i(x) = \sum_{k=1}^{p_i} a_{ik} \prod_{j \in N_{ik}} x_j \geq b_i, \quad i = 1, \ldots, m$$

$$x \in \{0,1\}^n,$$

$$\text{where} \quad N_{ik} \subseteq N = \{1, 2, \ldots, n\}, \quad c_k, a_{ik}, b_i \in R, \quad \forall i, k$$

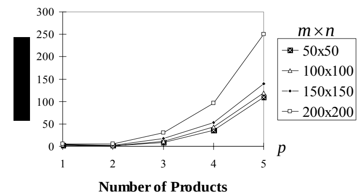
- **Applications in:**
  - Capital Budgeting (Laughlunn, 1970; Peterson and Laughlunn, 1971)
  - Media Selection (Zangwill, 1965)
  - Cluster Analysis (Rao, 1971;Torki *et al.*, 1996)
  - Asset Allocation (Perold and Sharpe, 1988)
  - Multi-project scheduling (Pritsker *et al.*, 1969)
  - Vision Loss Testing (Kolesar, 1980)

COMPUTATIONAL RESULTS FOR LMP

$$\min \quad \prod_{i=1}^{p} f_i(x) = \prod_{i=1}^{p} \left(c_i^t x + c_{i0}\right)$$

$$\text{s.t.} \quad Ax \leq b$$

$$x \in \Re_+^n, \; A \in \Re^{m \times n}$$



COMPARATIVE RESULTS FOR LMP

		Kuno *et al.*	BARON
	Platform	Sun 4/75	IBM RS/6000 43P
	Tolerance	10^{-5}	10^{-6}
p	(*m,n*)	CPU sec	CPU sec
3	(200,180)	914	110
4	(120,120)	1155	
	(150,150)		43
5	(20,30)	1170	
	(200,200)		250

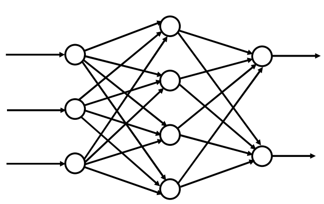
RS/6000 is 1.5 times faster than Sun 4/75

COMPARATIVE RESULTS FOR GLMP

			Konno *et al.*	BARON
		Platform	SunSPARC2	IBM RS/6000 43P
		Tolerance	10^{-5}	10^{-6}
p	*t*	(*m,n*)	CPU sec	CPU sec
2	2	(30,50)	26	2
		(70,50)	56	7
		(100,100)		28
	3	(30,50)	203	31
		(70,50)	1088	54
		(100,100)		143
	4	(30,50)	3898	69
		(70,50)		272
		(100,100)		1109
3	3	(70,50)		3497
4	2	(70,50)		2821

RS/6000 is 1.5 times faster than Sun SPARC2

TRAINING OF NEURAL NETWORKS



Given:
- The Structure of the Network
- An Error Function
- Training Sets
- Target Sets

Find:
- The values of the weights and the thresholds which minimize the error function

OPTIMIZATION MODEL

$$\min \sum_{i \in N_l} \sum_{p} (x_{lip} - t_{ip})^2$$

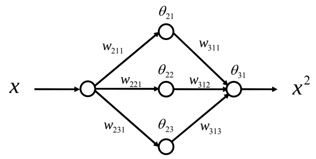
$$\text{s.t.} \; x_{lip} = \frac{1}{1 + e^{-s_{lip}}} \qquad l \in L; i \in N_l; p \in P$$

$$s_{lip} = -\theta_{li} + \sum_{j \in N_{l-1}} w_{lij} x_{l-1,jp} \quad l \in L; i \in N_l; p \in P$$

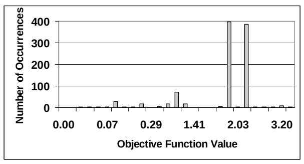
$$w_{lo} \leq w_{lij} \leq w_{up} \qquad i \in N_l; l \in L; j \in N_{l-1}$$

$$\theta_{lo} \leq \theta_{li} \leq \theta_{up} \qquad i \in N_l; l \in L$$

SIMULATION OF x^2

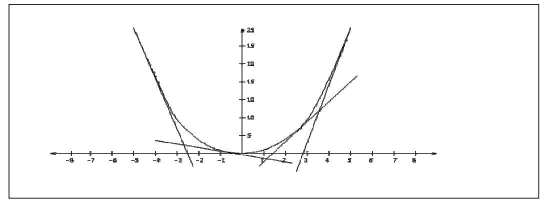


Training Set	**0.2**	**0.4**	**0.6**	**.8**	**1**
Target Set	**0.04**	**0.16**	**0.36**	**0.64**	**1**



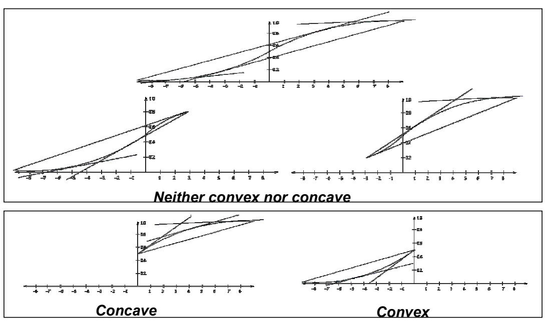
1000 Random Local Searches using MINOS

OBJECTIVE RELAXATION



Outer-linearization of the convex objective allows fast solution of descendant nodes at the expense of a weaker relaxation.

SIGMOIDAL RELAXATION



LOWER BOUNDING LP

Objective linearization

$$\min \sum_{i \in N_{\bar{l}}} \sum_{p \in P} z_{\bar{l}ip}$$

$$\text{s.t.} \quad z_{\bar{l}ip} \geq \alpha_{\bar{l}ipr} x_{\bar{l}ip} + \beta_{\bar{l}ipr} \qquad i \in N_{\bar{l}};\ p \in P;\ r \in R$$

Sigmoidal linearization

$$\left.\begin{aligned} x_{lip} &\geq \varepsilon_{lipf} s_{lip} + \zeta_{lipf} \\ x_{lip} &\geq \gamma_{lipm} s_{lip} + \delta_{lipm} \end{aligned}\right\} \qquad i \in N_l;\ l \in L;\ p \in P;\ f \in F;\ m \in M$$

$$s_{lip} = -\theta_{ij} + \sum_{j \in N_{l-1}} y_{lijp} \qquad i \in N_l;\ l \in L;\ p \in P$$

$$\left.\begin{aligned} y_{lijp} &\geq w_{lij}^{U} x_{l-1\,jp} + x_{l-1\,jp}^{U} w_{lij} - w_{lij}^{U} x_{l-1\,jp}^{U} \\ y_{lijp} &\geq w_{lij}^{L} x_{l-1\,jp} + x_{l-1\,jp}^{L} w_{lij} - w_{lij}^{L} x_{l-1\,jp}^{L} \\ y_{lijp} &\leq w_{lij}^{U} x_{l-1\,jp} + x_{l-1\,jp}^{L} w_{lij} - w_{lij}^{U} x_{l-1\,jp}^{L} \\ y_{lijp} &\leq w_{lij}^{L} x_{l-1\,jp} + x_{l-1\,jp}^{U} w_{lij} - w_{lij}^{L} x_{l-1\,jp}^{U} \end{aligned}\right\} \qquad i \in N_l;\ l \in L;\ p \in P;\ j \in N_{l-1}$$

McCormick linearization

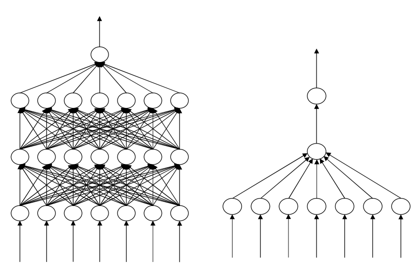
$$w_{lo} \leq w_{lij} \leq w_{up} \qquad i \in N_l;\ l \in L;\ j \in N_{l-1}$$

$$\theta_{lo} \leq \theta_{li} \leq \theta_{up} \qquad i \in N_l;\ l \in L$$

EXAMPLE: POLLUTION PREDICTION

- **Source:**
  - **"A new approach for finding the global minimum of error function of neural network",** ***Neural Networks*****, 2:367-373, 1989.**
- **Eight input nodes:**
  - **SO_2 density at 10 a.m.**
  - **(SO_2 density at 10 a.m.) - (SO_2 density at 7 a.m.)**
  - **Wind velocity at 10 a.m.**
  - **Wind velocity at 8 a.m.**
  - **SO_2 density at 9 a.m.**
  - **(SO_2 density at 9 a.m.) - (SO_2 density at 8 a.m.)**
  - **SO_2 density at noon last week.**
  - **(average SO_2 density between 8 a.m. and 10 a.m.) - (SO_2 density at 10 a.m.)**
- **One logical output node:**
  - **Value of 1 (alarm) when SO_2 density exceeds 5 pphm**
  - **Value of 0 when SO_2 density is below 5 pphm**

RESULTS FOR POLLUTION PREDICTION



Literature solution **Global solution**

CUTTING PLANE GENERATION

- Use supporting hyperplanes (outer approximation) of convex functions from:
  - Univariate convex functions of original problem
  - Univariate convex functions obtained from functional decomposition of multivariate functions
  - Convex envelopes of nonconvex functions
  - Multivariate functions identified by CONVEX_EQUATIONS modeling language construct by the user
- Supporting hyperplanes generated only if they are violated by LP solution
- Process:
  - Start with a rough outer approximation
  - Solve LP
  - Add some cuts
  - Repeat process at current node

ILLUSTRATIVE EXAMPLE 1: CUTS FROM CONVEX ENVELOPES

$$\begin{aligned} \min \quad & x^2 - 100x + y^2 - 30y + 1000\frac{x}{y} \\ \text{s.t.} \quad & 0 \le x \le 1000 \\ & 1 \le y \le 1000 \end{aligned}$$

Solution -1118 at (34.3, 31.8)

Iteration	Lower bound	Relaxation optimal solution
1	-7500.9	$x_1 = (65.3, 66.5)$
2	-3832.2	$x_2 = (33.1, 34.1)$
3	-2839.5	$x_3 = (49.2, 19.0)$
4	-2325.7	$x_4 = (41.1, 25.6)$
5	-2057.5	$x_5 = (37.1, 22.3)$
6	-2041.1	$x_6 = (39.1, 23.9)$

Cutting planes reduce root-node gap by 86%

With cuts: 7 nodes
Without: 47 nodes

ILLUSTRATIVE EXAMPLE 2: CONVEX_EQUATIONS CONSTRUCT

$$\begin{aligned} \min \quad & 100\log(e^{x_1} + e^{x_2} + e^{x_3}) + x_1^2 - 40x_1 \\ & + x_2x_3 - 10\log(x_1) + x_2^2 - 20x_2 - 50x_3 \\ \text{s.t.} \quad & 1 \le x_1 \le 10 \\ & 1 \le x_2 \le 10 \\ & 1 \le x_3 \le 10 \end{aligned}$$

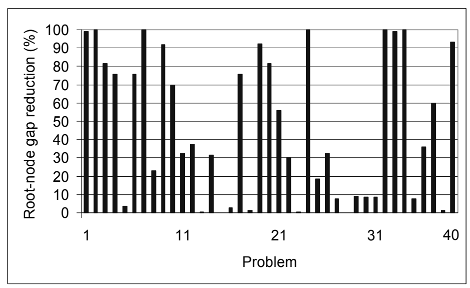
Solution 83.28

Iteration	Lower bound 1	Lower bound 2
1	-586.9	62.5
2	-514.9	78.9
3	-445.8	79.6
4	-436.2	80.2
5	-432.9	80.6

Cutting planes reduce root-node gap by 99.5%

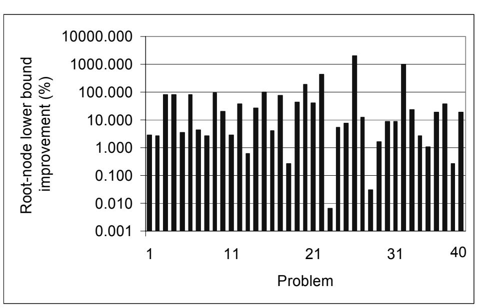
With cuts: 35 nodes
Without: 1793 nodes

ROOT-NODE GAP REDUCTIONS FOR PROBLEMS FROM `globallib`



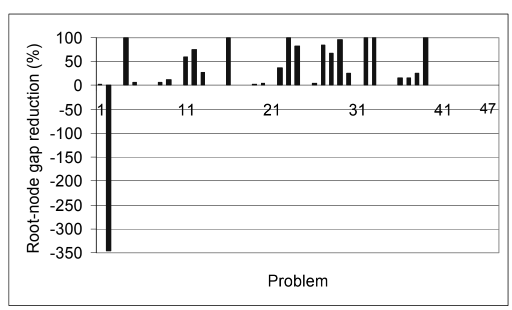
Range from 0.05% to 100%. Average 48%.

ROOT-NODE LOWER BOUND IMPROVEMENTS FOR `globallib`



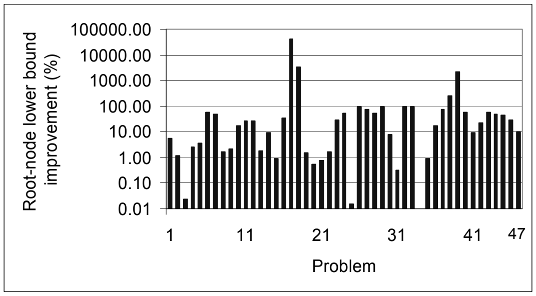
Up to 2000% improvement

ROOT-NODE GAP REDUCTIONS FOR PROBLEMS FROM `minlplib`



Average 27%
(excluding problem with worse upper bound).

ROOT-NODE LOWER BOUND IMPROVEMENTS FOR `minlplib`



Up to 41357% improvement

SOLUTION TO GLOBALITY--1

				Without cuts			With cuts		
Problem	m	n	n_d	N_t	N_m	CPU s	N_t	N_m	CPU s
arki0001	513	1030		261	157	25200	1	1	137
chakra	41	62		164238	84712	25200	1	1	0
du-opt	9	20	13	82520	44734	25200	79	25	157
du-opt5	9	20	12	106299	33954	25200	78	15	92
elf	38	54	24	866	91	31	698	89	95
enpro48	215	154	92	4215	434	165	702	92	84
enpro56	192	128	73	4609	477	182	1354	99	101
ex1233	64	52	12	28122	652	922	2198	310	1246
ex5_4_4	19	27		971	93	33	843	80	46
ex6_2_14	2	4		739	65	7	2713	115	38

SOLUTION TO GLOBALITY--2

				Without cuts			With cuts		
Problem	m	n	n_d	N_t	N_m	CPU s	N_t	N_m	CPU s
ex8_4_1	10	22		926853	58084	17864	13	4	4
ex8_4_4	12	17		2655	154	63	101	12	7
ex8_4_7	40	62		12497	7021	25200	10011	6658	1883
fac1	18	22	6	78055	8842	143	1	1	0
fac2	33	66	12	4653901	101202	25200	27	8	1
fac3	33	66	12	11516667	101172	25200	24	7	0
gsg_0001	112	78		145	22	17	89	10	21
gtm	24	63		3229450	87622	25200	1	1	0
himmel16	21	18		1211	152	27	915	116	81
linear	20	24		1904473	19706	24403	59472	955	2772

SOLUTION TO GLOBALITY--3

				Without cuts			With cuts		
Problem	m	n	n_d	N_t	N_m	CPU s	N_t	N_m	CPU s
parallel	115	205	25	651	65	198	555	72	194
ravem	186	112	53	778	134	30	246	52	18
sambal	10	17		9279	436	119	1	1	0
spectra2	73	70	30	1963	330	54	43	10	6
stockcycle	98	481	432	108260	67671	25200	1573	231	1152
tls4	64	105	89	191760	4357	4106	172015	4807	12295
Average	76	115	63	885825	23936	10583	9760	530	786

EFFECT OF CUTTING PLANES (26 problems)

Effect of adding cuts	Iterations	Memory	CPU time
Better by a factor at least two	18 (69%)	19 (73%)	15 (58%)
Between 30% and 100% better	2 (8%)	1 (4%)	3 (12%)
Difference smaller than 30%	5 (19%)	5 (19%)	2 (8%)
Between 30% and 100% worse	1 (4%)	1 (4%)	6 (23%)
Worse by a factor at least two	0 (0%)	0 (0%)	0 (0%)

26 PROBLEMS FROM `globallib` AND `minlplib`

	Minimum	Maximum	Average
Constraints	2	513	76
Variables	4	1030	115
Discrete variables	0	432	63

EFFECT OF CUTTING PLANES

	Without cuts	With cuts	% reduction
Nodes	23,031,434	253,754	99
Nodes in memory	622,339	13,772	98
CPU sec	275,163	20,430	93

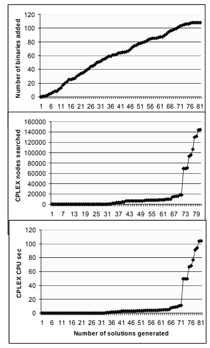
BETTER SOLUTION FOR tls4

Problem	Objective		Problem	Objective	
arki0001	40.7129	*	fac1	160912612.3500	
chakra	-179.1336	*	fac2	331837498.1770	
du-opt	3.5563		fac3	31982309.8480	
du-opt5	8.0737		gsg_0001	2378.1605	*
elf	0.1917		gtm	543.5651	*
enpro48	187276.7080		himmel16	-0.8660	
enpro56	263427.2212		linear	88.9994	*
ex1233	155010.6713		parallel	924.2956	
ex5_4_4	10077.7754	*	ravem	269589.5584	
ex6_2_14	-0.6954		sambal	3.9682	
ex8_4_1	0.6185	*	spectra2	13.9783	
ex8_4_4	0.2125		stockcycle	119948.6883	
ex8_4_7	28.6032	*	tls4	8.3000	¶

*: No solutions reported for these problems in globallib and minlplib.

¶: Better solution than the one reported earlier in minlplib.

FINDING THE *K*-BEST OR ALL FEASIBLE SOLUTIONS



Typically found through repetitive applications of branch-and-bound and generation of "integer cuts"

$$\min \quad \sum_{i=1}^{4} 10^{4-i} x_i$$
$$\text{s.t.} \quad 2 \le x_i \le 4, \quad i = 1,\ldots,4$$
$$x \quad \text{integer}$$

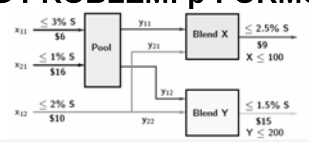
BARON finds all solutions:

- No integer cuts
- Fathom nodes that are infeasible or points
- Single search tree
- 511 nodes; 0.56 seconds
- Applicable to discrete and continuous spaces

RELAXATION-ONLY CONSTRAINTS

- **Can strengthen relaxation by adding to the model:**
  - Nonlinear reformulations (RLT)
  - First-order optimality conditions
  - Problem-specific optimality conditions and symmetry-breaking constraints
- **Traditionally, modeling languages for optimization pass single model**
- **RELAXATION_ONLY_EQUATIONS construct added to BARON's modeling language**
- **Strengthen relaxation without complicating local search**

POOLING PROBLEM: p-FORMULATION



$$\min \quad \overbrace{6x_{11} + 16x_{21} + 10x_{12}}^{\text{cost}} - \overbrace{9(y_{11} + y_{21})}^{X\text{-revenue}} - \overbrace{15(y_{12} + y_{22})}^{Y\text{-revenue}}$$

s.t. $q = \dfrac{3x_{11} + x_{21}}{y_{11} + y_{12}}$ — Sulfur Mass Balance

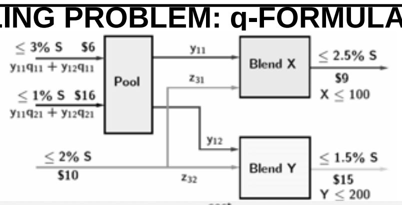
$x_{11} + x_{21} = y_{11} + y_{12}$, $x_{12} = y_{21} + y_{22}$ — Mass balance

$\dfrac{qy_{11} + 2y_{21}}{y_{11} + y_{21}} \le 2.5$, $\dfrac{qy_{12} + 2y_{22}}{y_{12} + y_{22}} \le 1.5$ — Quality Requirements

$y_{11} + y_{21} \le 100$, $y_{12} + y_{22} \le 200$ — Demands

Haverly 1978

POOLING PROBLEM: q-FORMULATION



$$\min \quad \overbrace{6\,(y_{11}q_{11} + y_{12}q_{11}) + 16\,(y_{11}q_{21} + y_{12}q_{21}) + 10\,(z_{31} + z_{32})}^{\text{cost}}$$
$$- \overbrace{9(y_{11} + y_{21})}^{X\text{-revenue}} - \overbrace{15(x_{12} + x_{22})}^{Y\text{-revenue}}$$

s.t. $q_{11} + q_{21} = 1$ — Mass Balance

$-0.5z_{31} + 3y_{11}q_{11} + y_{11}q_{21} \le 2.5y_{11}$, $0.5z_{32} + 3y_{12}q_{11} + y_{12}q_{21} \le 1.5y_{12}$ — Quality Requirements

$y_{11} + z_{31} \le 100$, $y_{12} + z_{32} \le 200$ — Demands

Ben-Tal et al. 1994

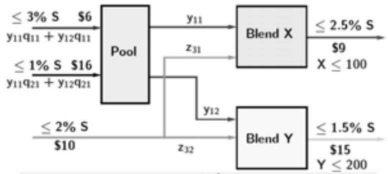
RESULTS FOR POOLING PROBLEMS FROM THE LITERATURE

Algorithm	Foulds'92		Ben-Tal'94	GOP'93		GOP'96		BARON'99	
Computer*	CDC 4340			HP9000/730		HP9000/730		RS6000/43P	
Tolerance*						**		10^{-6}	
Problem	Iter	CPU s	Iter	Iters	CPU s	Iter	CPU s	Iter	CPU s
Haverly 1	5	0.7	3	7	0.95	12	0.22	3	0.09
Haverly 2			3	19	3.19	12	0.21	8	0.09
Haverly 3			3			14	0.26	5	0.13
Foulds 2	9	3						1	0.10
Foulds 3	1	10.5						1	2.33
Foulds 4	25	125						1	2.59
Foulds 5	125	163.6						1	0.86
Ben-Tal 4			25	47	44.54	7	0.95	3	0.11
Ben-Tal 5			283	42	40.31	0		1	1.12
Adhya 1								1869	77
Adhya 2								2087	146
Adhya 3								7369	1160
Adhya 4								157	10

*: Blank entries in this table indicate t hat data were not provided or problems were not solved by prior approaches.

**: Tolerances used in GOP'96 were 0.05% for Haverly 1, 2, 3; 0.5% for Ben-Tal 4; and 1% for Ben-Tal 5.

POOLING PROBLEM: pq-FORMULATION



$$\min \quad \overbrace{6\,(y_{11}q_{11} + y_{12}q_{11}) + 16\,(y_{11}q_{21} + y_{12}q_{21}) + 10\,(z_{31} + z_{32})}^{\text{cost}} - \overbrace{9(y_{11} + y_{21})}^{X\text{-revenue}} - \overbrace{15(x_{12} + x_{22})}^{Y\text{-revenue}}$$

s.t.

$$q_{11} + q_{21} = 1 \qquad \text{Mass Balance}$$

$$-0.5z_{31} + 3y_{11}q_{11} + y_{11}q_{21} \leq 2.5y_{11}$$
$$0.5z_{32} + 3y_{12}q_{11} + y_{12}q_{21} \leq 1.5y_{12} \qquad \text{Quality Requirements}$$

$$y_{11} + z_{31} \leq 100$$
$$y_{12} + z_{32} \leq 200 \qquad \text{Demands}$$

$$y_{11}q_{11} + y_{11}q_{21} = y_{11}$$
$$y_{12}q_{11} + y_{12}q_{21} = y_{12} \qquad \text{Convexification Constraints}$$

LOCAL SEARCH WITH CONOPT

	q-formulation			pq-formulation		
Problem	Objective	CPU s	Iter	Objective	CPU s	Iter
adhya1	-68.74	0.01	9	-56.67	0.00	5
adhya2	0.00	0.01	4	0.00	0.01	3
adhya3	-65.00	0.03	12	-57.74	0.02	7
adhya4	-470.83	0.01	9	-470.83	0.02	9
bental4	0.00	0.01	3	0.00	0.00	3
bental5	-2900.00	0.02	9	-2700.00	0.03	18
foulds2	-1000.00	0.00	6	-600.00	0.01	14
foulds3	-6.50	0.04	6	-6.50	0.09	9
foulds4	-6.00	0.04	6	-6.50	0.16	23
foulds5	-7.00	0.04	7	-6.50	0.06	7
haverly1	-400.00	0.00	5	0.00	0.00	3
haverly2	-400.00	0.00	5	0.00	0.01	3
haverly3	-750.00	0.01	8	0.00	0.00	3
rt97	inf	0.00	4	-4330.78	0.01	8
sum	-6074.07*	0.22	89	-3904.74*	0.41	107

*: Not including rt97.

GLOBAL SEARCH WITH BARON

Problem	Strategy 1				Strategy 2				Strategy 3			
	N_t	N_o	N_m	CPU s	N_t	N_o	N_m	CPU s	N_t	N_o	N_m	CPU s
adhya1	573	550	50	17	30	24	7	1	28	24	7	0.5
adhya2	501	338	41	20	17	13	4	1	17	13	4	0.5
adhya3	9248*	2404	1800*	1200*	31	1	6	1.5	31	1	6	1.5
adhya4	6129*	-1	1620*	1200*	1	1	1	1.5	1	1	1	1
bental4	101	101	14	0.5	1	-1	1	0.5	1	-1	1	0.5
bental5	6445*	901	3815*	1200*	-1	-1	0	0.5	-1	-1	0	0
foulds2	1061	977	106	16	-1	-1	0	0	-1	-1	0	0
foulds3	348*	91	260*	1200*	-1	-1	0	1	-1	-1	0	5
foulds4	326*	262	246*	1200*	-1	-1	0	1	-1	-1	0	1
foulds5	389*	316	287*	1200*	-1	-1	0	1	-1	-1	0	1
haverly1	25	6	5	0	1	1	1	0	1	1	1	0
haverly2	17	1	5	0	1	1	1	0	1	1	1	0
haverly3	3	1	2	0	1	1	1	0	1	1	1	0
rt97	5629	2836	609	173.5	13	6	4	0.5	13	6	4	0.5
sum	30795	8783	8860	7427	91	42	26	10	89	42	26	12

*: Run did not terminate within 1200 seconds.

CONCLUSIONS

- **ALGORITHMS**
  - **Range reduction:**
    - **Easy to implement**
    - **Applicable within any global optimization algorithm**
  - **Finite algorithms for:**
    - **Concave minimization**
    - **Multi-stage stochastic programming**
  - **Lower Bounding:**
    - **Convex extensions provide convex envelopes**
    - **Polyhedral relaxations through sandwich algorithm**
- **SOFTWARE**
  - **Specialized codes for concave, multiplicative, polynomial, fractional, fixed-charge programs ...**
  - **Problems with up to a few thousand variables and constraints solvable on conventional computers**
- **APPLICATIONS**
  - **Engineering design and manufacturing**
  - **Informatics problems in chemistry, biology, medicine, …**
  - **Design of new Runge-Kutta methods for ODEs**

OPPORTUNITIES

- **Theory and Algorithms**
  - **Finiteness issues**
  - **Convexification**
  - **Probabilistic and worst-case analysis of approximation schemes**
  - **Guaranteed approximation schemes**
- **Applications**
  - **Global optimization of "Differential-Algebraic Systems"**
  - **Global optimization of "Black Boxes"**
- **Implementations**
  - **Supply chain management**
  - **Engineering design and manufacturing**
  - **Molecular design and analysis**
  - **Finance**
  - **Pattern recognition**
  - **…**

BARON MANUAL

- **Screen logs**
- **Missing bounds**
- **No starting point is required**
- **Many solutions (numsol) can be found**
- **Optimality tolerances**
- **Branching strategies**
- **Number of outer approximators**
- **Local search options**
- **Relaxation_only and Convex equations**

RESOURCES

- Springer's Series on Nonconvex Optimization and Its Applications (NOIA):
  - http://www.springeronline.com/sgw/cda/frontpage/0,11855,5-10044-69-33111451-0,00.html
- Journal of Global Optimization:
  - http://www.springeronline.com/sgw/cda/frontpage/0,11855,4-40012-70-35755812-0,00.html
- Neumaier's Global Optimization web page:
  - http://www.mat.univie.ac.at/~neum/glopt.html

OPTIMIZATION UNDER UNCERTAINTY

A LONG RECOGNIZED NEED

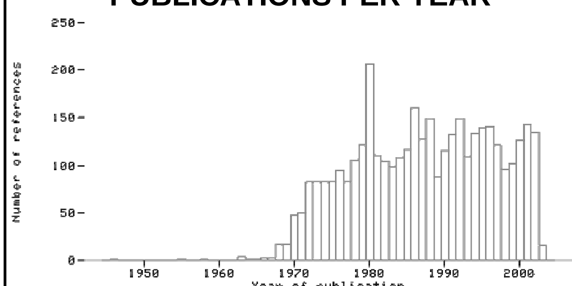
"Those of us who were doing the planning right from the very beginning understood that the real problem was to be able to do planning under uncertainty."

G. B. Dantzig, E-Optimization (2001)
Interviewed by Irv Lustig

THE FIRST PAPERS

- Stochastic Programming
  - Based on probability distributions for uncertain parameters
  - Minimize expected costs
    - » Beale (1955)
    - » Dantzig (1955)
    - » Tintner (1955)
  - Maximize system's ability to meet constraints
    - » Charnes & Cooper's chance-constraint programming (1959)
- Fuzzy Programming
  - Optimization over soft constraints
  - Bellman & Zadeh (1970)

STOCHASTIC PROGRAMMING PUBLICATIONS PER YEAR



- Maarten H. van der Vlerk. ***Stochastic Programming Bibliography.*** http://mally.eco.rug.nl/index.html?spbib.html, last updated on May 2003
- Over 3840 papers on stochastic programming
  - 100 papers per year for the past 30 years

STILL A NEED

"Planning under uncertainty. This, I feel, is the real field we should all be working on."

G. B. Dantzig, E-Optimization (2001)

PRESENTATION GOALS

- Illustrate algorithmic challenges
  - Stochastic programming
    - Expectation minimization
    - Chance-constrained
    - Linear, integer, and nonlinear programming
  - Fuzzy programming
- Review progress to date
  - Computational state-of-the-art
- Introduction to approximation schemes and probabilistic analysis

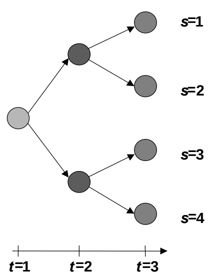
STOCHASTIC PROGRAMS

- Multi-stage optimization problems with parameter uncertainties
  - Decisions do not affect the uncertainties
  - Finite number of decision stages

- Objective: Minimize expected total cost

MODELING UNCERTAINTY

- Assume: A finite sample space
- Uncertainty is modeled as a scenario tree
- A scenario is a path from the root to a leaf



TWO-STAGE STOCHASTIC LP WITH RECOURSE

- Decide $x \Rightarrow$ Observe scenario $\Rightarrow$ Decide y
  - x is the vector of first-stage variables
  - y is the vector of second-stage variables
- Objective: E[total cost]
- Second stage problem depends on first-stage decision and scenario realized

$$\begin{aligned} \min \quad & cx + \sum_{s=1}^{S} p^s Q^s(x) \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

where $Q^s(x) = \min\{ f^s y \mid D^s y \geq h^s + T^s x \}.$

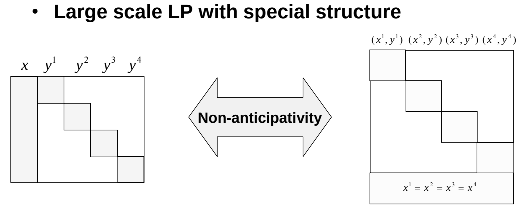
THE CHALLENGE

- Consider 100 uncertain parameters
- Each parameter can take 3 values
- Total number of possible scenarios is $3^{100} = 5\text{x}10^{47}$

- Explicit evaluation of the second-stage cost function is out of the question

STOCHASTIC LP

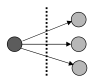
- $Q^s(x)$ is the value function of a linear program
  - Piece-wise linear and convex
  - Convex programming methods are applicable
- Properties and algorithms extend to:
  - Multi-stage stochastic LP
  - First-stage integer variables
- Large scale LP with special structure



DECOMPOSITION

Primal Methods	Dual Methods

Primal Methods

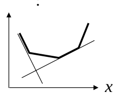


$$\min_{x\in X} cx + \sum_{s=1}^{S} p^s \theta^s$$

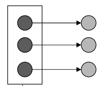
$$\text{s.t.} \quad \theta^s \ge a_1^s + b_1^s x \quad \forall s$$

$$\theta^s \ge a_2^s + b_2^s x \quad \forall s$$

$$\vdots$$



Dual Methods



Non-anticipativity $\sum_{s=1}^{S} H^s x^s = 0$

$$\max_{\lambda} L(\lambda)$$

$$L(\lambda) = \sum_{s=1}^{S} \min_{\substack{x^s \in X \\ Dy^s \ge h^s + T^s x^s}} \{(c+\lambda H^s)x^s + p^s f^s y^s\}$$

SAMPLING APPROXIMATIONS

- **“Interior” sampling methods**
  - In each decomposition iteration, sample a few only scenarios
  - Dantzig and Infanger (1992), Infanger (1994)
- **“Exterior” sampling methods**
  - First sample a few scenarios, then solve stochastic LP with sampled scenarios only
  - Shapiro (1996)
- **Desirable statistical convergence properties**

STATE-OF-THE-ART IN COMPUTATIONS

- **Exact algorithms**
  - Birge (1997)
  - Millions of variables in deterministic equivalent
    - » 1000 variables
    - » 10 uncertain parameters, each with 3 possible values
  - Parallel computers
- **Sampling-based methods**
  - Linderoth, Shapiro and Wright (2002)
  - Computational grid
  - Up to 10^{81} scenarios
  - Within an estimated 1% of optimality

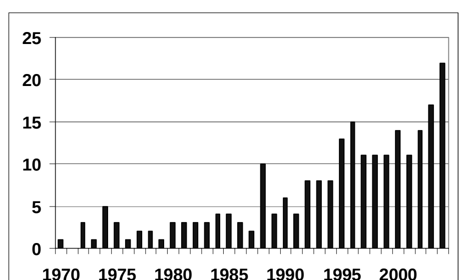
TWO-STAGE STOCHASTIC INTEGER PROGRAMMING

- **Second stage optimization problem involves combinatorial decisions**
- **Examples:**
  - Resource acquisition (Dempster et al., 1983):
    Acquire machines ⇒ Observe processing times ⇒ Schedule jobs
  - Location-Routing (Laporte et al., 1989):
    Locate depots ⇒ Observe demand ⇒ Route vehicles
  - Crew recovery:
    Assign crews ⇒ Observe breakdown ⇒ Recover crews
- $Q^s(x)$ **is the value function of an integer program**

FIRST RELATED PAPERS

- **Modeling with integer variables under uncertainty**
  - Ferguson and Dantzig, 1955-1956
    - » Allocation of aircraft to routes
    - » Number of aircraft
    - » Totally unimodular constraint matrices
- **Ettinger and Hammer (1972)**
  - Polynomial chance constraints in 0-1 variables
  - Transformation to deterministic equivalent
- **Zimmermann and Pollatschek (1972)**
  - Linear 0-1 programs with stochastic right-hand side
  - Find set of feasible right-hand-side values
- **Yudin and Tzoy (1973)**
  - Maximization of expected value of nonlinear stochastic 0-1 programs
  - Reformulation to infinite-dimensional LP
  - Relaxation for two-stage 0-1 problems
- **Wollmer (1980)**
  - Two-stage with 0-1 first-stage and continuous second-stage variables
  - Benders decomposition

STOCHASTIC INTEGER PROGRAMMING JOURNAL PAPERS PER YEAR



- **Over 250 papers on stochastic integer programming**
- **Strong growth in past two decades**

TWO-STAGE SIP WITH CONTINUOUS RECOURSE

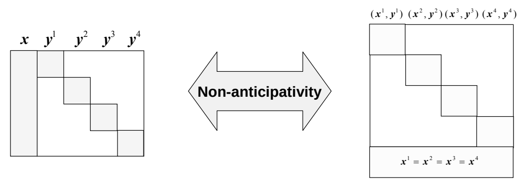
- **Decide $x \Rightarrow$ Observe scenario $\Rightarrow$ Decide y**
  - x **is the vector of first-stage variables**
  - y **is the vector of second-stage variables**
- **Objective: E[total cost]**
- **Second stage problem depends on first-stage decision and scenario realized**

$$\begin{aligned} \min \quad & cx + \sum_{s=1}^{S} p^s Q^s(x) \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \text{integer} \end{aligned}$$

where $Q^s(x) = \min\{f^s y \mid D^s y \geq h^s + T^s x\}$.

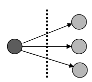
TWO-STAGE SIP WITH CONTINUOUS RECOURSE

- $Q^s(x)$ **is the value function of a linear program**
  - **Piece-wise linear and convex**
  - **Convex programming methods are applicable**
- **Large scale ILP with special structure**



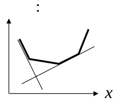
DECOMPOSITION

Primal Methods | **Dual Methods**

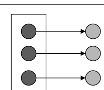


$$\min_{x \in X} cx + \sum_{s=1}^{S} p^s \theta^s$$

$$\begin{aligned} \text{s.t.} \quad & \theta^s \geq a_1^s + b_1^s x \quad \forall s \\ & \theta^s \geq a_2^s + b_2^s x \quad \forall s \\ & \vdots \end{aligned}$$



Wollmer (1980)



Non-anticipativity $\sum_{s=1}^{S} H^s x^s = 0$

$$\max_{\lambda} L(\lambda)$$

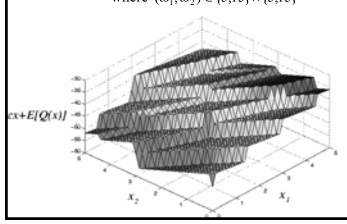
$$L(\lambda) = \sum_{s=1}^{S} \min_{\substack{x^s \in X \\ Dy^s \geq h^s + T^s x^s}} \{(c + \lambda H^s) x^s + p^s f^s y^s\}$$

Carøe and Tind (1997), Carøe and Schultz (1999)

THE CHALLENGE: INTEGER RECOURSE

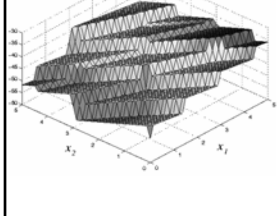
$$\begin{aligned} \min \quad & f(x_1, x_2) = -1.5x_1 - 4x_2 + \sum_{s=1}^{4} \frac{1}{4} Q^s(x_1, x_2) \\ s.t. \quad & 0 \leq x_1, x_2 \leq 5 \\ & Q^s(x_1, x_2) := \min \quad -16y_1 - 19y_2 - 23y_3 - 28y_4 \\ & \qquad s.t. \quad 2y_1 + 3y_2 + 4y_3 + 5y_4 \leq \omega_1^s - \frac{1}{2}x_1 - \frac{2}{3}x_2 \\ & \qquad\qquad 6y_1 + y_2 + 3y_3 + 2y_4 \leq \omega_2^s - \frac{2}{3}x_1 - \frac{1}{3}x_2 \\ & \qquad\qquad y_i \in \{0,1\} \quad \text{for } i = 1, \ldots, 4 \end{aligned}$$

where $(\omega_1, \omega_2) \in \{5,15\} \times \{5,15\}$



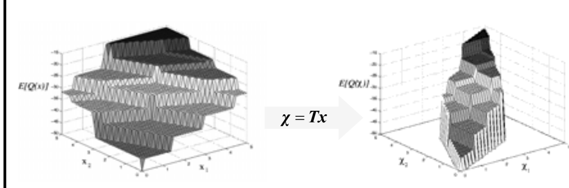
- **Discontinuous**
- **Highly non-convex**
- **Many local minima**

FINITENESS ISSUE



- **Branching on continuous first-stage variables may not be finite**
- **Consider rectangular partitioning—branching along a variable axis**
- **The polyhedral discontinuous pieces cannot be finitely isolated by rectangular partitions**

VARIABLE TRANSFORMATION



Ahmed, Tawarmalani and Sahinidis (2004)

- **Solve the problem in the space of the “tender variables”**
- **Variable transformation aligns discontinuities orthogonal to variable axes**
- **Discontinuities identified based on Blair and Jeroslow (1977) results**
- **Finite termination**

COMPUTATIONAL RESULTS

TEST PROBLEMS

Problem	Binary Variables	Continuous Variables	Constraints
SIZES3	40	260	142
SIZES5	60	390	186
SIZES10	110	715	341

	JORJANI ('95) CPLEX B&B				CAROE ('98) B&B with Lagrangian Rel.				BARON			
Problem	LB	UB	nodes	CPU¶	LB	UB	nodes	CPU¶	LB	UB	nodes	CPU *
SIZES3	218.2	224.7	20000	1859.8	224.3	224.5	-	1000	224.4	224.4 ✓	260	70.7
SIZES5	220.1	225.6	20000	4195.2	224.3	224.6	-	1000	224.5	224.5 ✓	13562	7829.1
SIZES10	218.2	226.9	250000	7715.5	224.3	224.7	-	1000	224.2	224.7	23750	10000.0

¶ Digital Alpha 500 Mhz

* IBM RS/6000 133 MHz

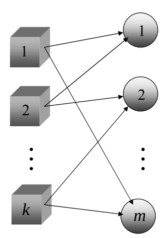
OTHER WORKS

- **Computational algebra**
  - Schultz *et al.* (1998)
  - Hemmecke and Schultz (2003)
- **Convexification/Decomposition**
  - Sherali and Fraticelli (2002)
  - Sen and Higle (2003)
  - Sen and Sherali (2004)
  - van der Vlerk (2004)
- **Superadditive dual**
  - Kong *et al.* (2004)
- **Reviews**
  - Klein Haneveld and van der Vlerk (1998)
  - Schultz (2003, 2004)
  - Louveaux and Schultz (2004)

MULTISTAGE SIP: PLANNING IN THE SUPPLY CHAIN

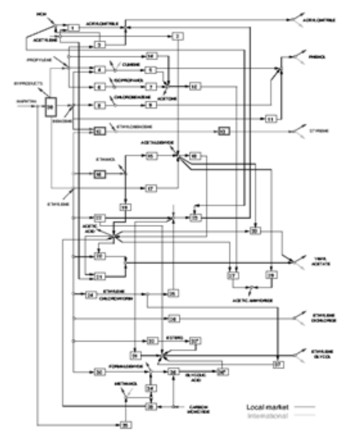
Ahmed and Sahinidis (2003)

- **Given:**
  - A network of *k* facilities
  - *m* product families
  - Forecasts of demands and costs for *n* time periods
- **Determine**
  - When and how much to expand?
  - How to allocate capacity?



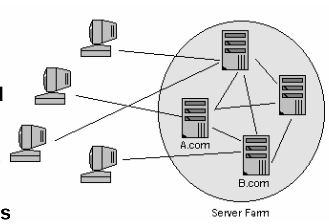
PROCESS SUPPLY CHAIN

- A network of processes, chemicals and markets
- New products and processing technology are anticipated
- When and how much new capacity to bring on-line?



SERVER FARMS

- **A network of servers hosting WebPages**
- **When and how much new technology to install to meet demand growth?**
- **Multi-billion $ industry**
- **Technology adoption is a leading strategic concern**



ASSUMPTIONS

- **Expansion involves a set-up cost $\Rightarrow$ Fixed charge cost function**
- **Linear production technology**
- **No inventories across time periods (can be relaxed)**
- **Continuous expansion units**

THE DETERMINISTIC MILP

$$\min z_n = \sum_{t=1}^{n} [\underbrace{\alpha_t X_t + \beta_t Y_t}_{\text{Expansion Costs}} + \underbrace{tr(\delta_t W_t)}_{\text{Allocation Costs}}]$$

Expansion ≤ Bound: $X_t \le U_t Y_t$

Production ≤ Capacity: $W_t e \le X_0 + \sum_{\tau=1}^{t} X_\tau$

Production = Demand: $diag(AW_t) = d_t$ $\quad t = 1, \ldots, n$

Non-negativity: $X_t \in \Re_+^k, \; W_t \in \Re_+^{k \times m}$

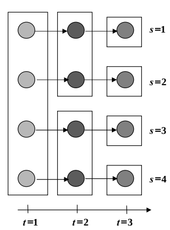
Binary Variables: $Y_t \in \{0,1\}^k$

UNCERTAINTY

- **Significant forecast uncertainty**
- **Sources:**
  - **Demands**
  - **Costs and prices**
  - **Technology**
- **Evolves over multiple time periods**
- **There are integer decision making variables in every time period/stage**

THE SCENARIO FORMULATION

$$\min \sum_{s=1}^{S} p^s f^s(x^s)$$
$$\text{s.t. } x^s \in \mathsf{X}^s \bigcap \mathsf{N} \quad \forall s = 1, \ldots, S.$$



Where

$x^s = (x_1^s, x_2^s, \ldots, x_n^s)$

$x_t^s = (X_t^s, Y_t^s, W_t^s)$

$p^s =$ **Probability of scenario** s

$f^s(\cdot) =$ **Objective function for scenario** s

$\mathsf{X}^s =$ **Constraints for scenario** s

$\mathsf{N} = \{x^s \mid x_t^{s_1} = x_t^{s_2} \; \forall (s_1, s_2) \in \mathsf{B}_t\}$ ← **Non-anticipativity**

COMPLEXITY IN THE TIME DOMAIN

- **The capacitated lot sizing problem (CLSP) is NP-hard**
- **Given any CLSP, we can construct an equivalent instance of the deterministic capacity expansion problem:**

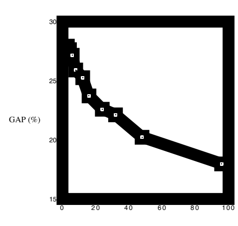
$$\min \sum_{t=1}^{n} (p_t x_t + s_t y_t) \quad \text{s.t.} \quad x_t \le C_t y_t, \quad I_{t-1} + x_t = d_t + I_t, \quad x_t, I_t \ge 0; \; y_t \in \{0,1\}$$

⟺

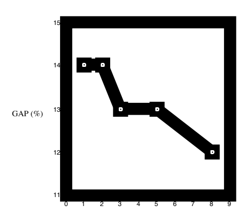
$$\min \sum_{t=1}^{n} (p_t X_t + s_t Y_t) \quad \text{s.t.} \quad X_t \le C_t Y_t, \quad W_t = \sum_{\tau=1}^{t} d_\tau, \; W_t \le I_0 + \sum_{\tau=1}^{t} X_\tau, \quad X_t, W_t \ge 0; \; Y_t \in \{0,1\}$$

The deterministic capacity expansion is NP-hard in the number of time periods but…

EMPIRICAL EVIDENCE



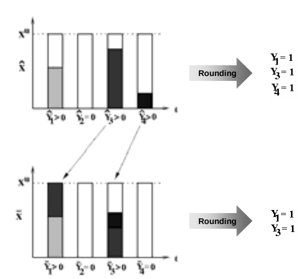
- **Liu & Sahinidis (IECR 1995)**
- **Processing Networks**
- **LP Relaxation**



- **Chang & Gavish (OR 1995)**
- **Telecommunication networks**
- **Lagrangian Relaxation**

CAPACITY SHIFTING

For the Deterministic Problem



N.B.: Naive rounding of LP solution results in too many expansions

3-PHASE HEURISTIC

For the Stochastic Problem

Construct an Implementable solution

- Relax integrality
- Solve as a multi-stage stochastic LP

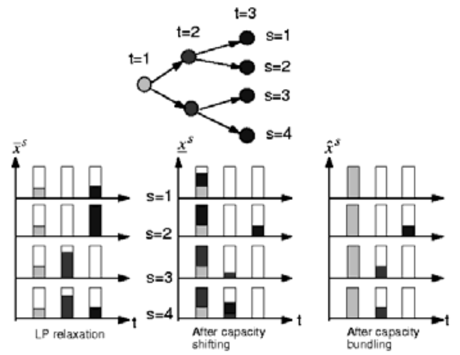
Construct an Admissible solution

- Relax non-anticipativity
- For each scenario, construct an integral solution by capacity shifting

Construct a Feasible solution

- Re-enforce non-anticipativity by capacity bundling

ILLUSTRATION



PROBABILISTIC ANALYSIS

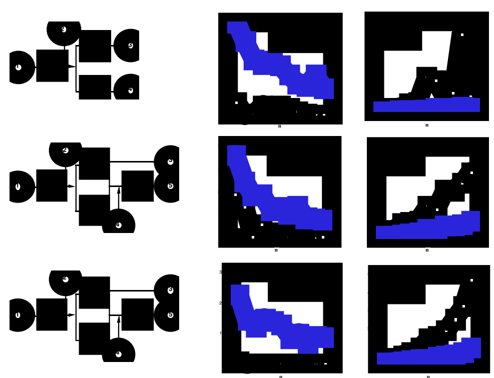
- How does the heuristic perform in "most" cases?
- Consider instances generated from the following probability model:
  - Demand in each period is independent with bounded first and second moments
  - Cost parameters have bounded distributions

Theorem:

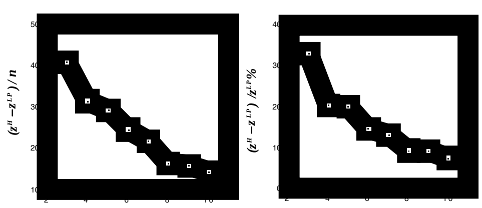
$$\lim_{n\to\infty} \frac{z_n^H - z_n^*}{z_n^*} = 0 \qquad \text{w.p.1}$$

- For "almost all," "large" sampled instances, the heuristic error vanishes asymptotically

GAPS FOR EXAMPLES 1-3



GAPS FOR EXAMPLE 4



- 38 processes, 24 chemicals
- With 10 time periods and 2^9 scenarios
  - 36,000 binaries, 184,000 continuous variables, 368,000 constraints
- Limited only by the size of the stochastic LP that can be solved

OTHER MULTISTAGE SIP APPLICATIONS

- Asset liability management
  - Drijver *et al.* (2002)
- Power production optimization
  - Takriti *et al.* (1996)
  - Carøe and Schultz (1999)
  - Nowak and Römisch (2000)
- Production planning and scheduling
  - Ahmed *et al.* (2001)
  - Lulli and Sen (2002)
  - Balasubramanian and Grossmann (2004)

ROBUSTNESS ISSUES

- **Recourse model provides first-stage solution that optimizes expected second-stage cost**
- **This solution may be very bad under certain conditions**
- **Robust solutions: remain near-optimal irrespective of uncertain outcome**
- **Mulvey, Vanderbei and Zenios (1995)**
  - May not lead to optimal second-stage decisions
  - King *et al*. (1997), Sen and Higle (1999)
  - Takriti and Ahmed (2002)
- **More recent approaches**
  - Ben-Tal and Nemirovski (2000)
  - Bertsimas (2002)

PROBABILISTIC PROGRAMMING

- **Also known as chance-constrained programming**
- **Focuses on reliability of the system**
- **LP with chance constraints:**

$$\begin{aligned} \min \quad & c^t x \\ \text{s.t.} \quad & P(Ax \geq b) \geq p \\ & x \geq 0 \end{aligned}$$

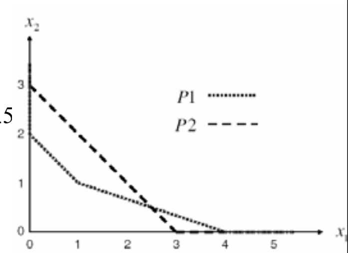
- **Consider**
  - One constraint
  - Deterministic $A=a^t$
  - Uncertain b, with $F(\beta)=p$

$$\Longrightarrow \quad \begin{aligned} \min \quad & c^t x \\ \text{s.t.} \quad & a^t x \geq \beta \\ & x \geq 0 \end{aligned}$$

THE CHALLENGE

Consider

$$\begin{aligned} \min \quad & c^t x \\ \text{s.t.} \quad & P\begin{pmatrix} x_1 + x_2 \geq b_1 \\ x_1 + 3x_2 \geq b_2 \end{pmatrix} \geq 0.5 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$



with

$$P(b_1 = 2, b_2 = 4) = 0.5$$
$$P(b_1 = 3, b_2 = 0) = 0.5$$

Probabilistic programming is a global optimization problem

FUZZY PROGRAMMING

- **Considers uncertain parameters as fuzzy numbers**
- **Treats constraints as fuzzy sets**
- **Some constraint violation is allowed**
- **Bellman and Zadeh (1970)**
  - Minimize largest constraint violation
- **Flexible programming**
  - Right-hand-side and objective uncertainty
- **Possibilistic programming**
  - Constraint coefficient uncertainty
  - Nonconvex optimization problem
    - » Liu and Sahinidis (1997)
- **Zimmermann (1991)**
- **Comparisons needed between SP and FP!**

STOCHASTIC PROGRAMMING OPPORTUNITIES

- **Global optimization algorithms and software now available**
  - Nonconvex stochastic integer programming
  - BARON: 100s to 1000s of variables
  - Subproblems within decomposition and sampling
- **Applications in systems biology and bioinformatics**
  - Metabolic pathway design
  - Protein binding site identification
  - DNA sequencing