Overview of
Mixed-integer Nonlinear Programming

Ignacio E. Grossmann
Center for Advanced Process Decision-making
Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon

Outline

- **MINLP models in planning and scheduling**
- **Overview MINLP methods**
- **Brief reference to Generalized Disjunctive Programming**
- **Overview of global optimization of MINLP models**

- **Challenges**
  - How to develop effective algorithms for nonlinear discrete/continuous?
  - How to improve relaxation?
  - How to solve nonconvex GDP problems to global optimality?

Observation for EWO problems

-Most Planning, Scheduling, Supply Chain models are linear = > MILP

Major reasons: Simplified performance models (fixed rates, processing times)

Fixed-time horizon, minimization makespan problems

See recent reviews in area:

Mendez, C.A., J. Cerdá , I. E. Grossmann, I. Harjunkoski, and M. Fahl, (2006). "State-Of-The-Art Review of Optimization Methods for Short-Term Scheduling of Batch Processes," *Computers & Chemical Engineering* 30, 913-946.

Burkard, R., & Hatzl, J. (2005). Review, extensions and computational comparison of MILP formulations for scheduling of batch processes. *Computers and Chemical Engineering*, 29, 2823–2835.

Kelley, J..D., "Formulating Production Planning Problems," *Chemical Engineering Progress*,", Jan. p.43 (2004)

Floudas, C. A., & Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. *Computers and Chemical Engineering*, 28, 2109–2129.

Kallrath, J. (2002). Planning and scheduling in the process industry. *OR Spectrum,* 24, 219–250.

When are nonlinearities required?

Cyclic scheduling problems: *infinite horizon (only nonlinear objective)*

Sahinidis, N.V. and I.E. Grossmann, "MINLP Model for Cyclic Multiproduct Scheduling on Continuous Parallel Lines," *Computers and Chemical Engineering* 15, 85 (1991).

Pinto, J. and I.E. Grossmann, "Optimal Cyclic Scheduling of Multistage Multiproduct Continuous Plants," *Computers and Chemical Engineering*, 18, 797-816 (1994)

Performance models:

Jain, V. and I.E. Grossmann, "Cyclic Scheduling and Maintenance of Parallel Process Units with Decaying Performance", *AIChE J.*, 44, pp. 1623-1636 (1998) **Exponential decay furnaces**

Van den Heever, S.A., and I.E. Grossmann, "An Iterative Aggregation/Disaggregation Approach for the Solution of a Mixed Integer Nonlinear Oilfield Infrastructure Planning Model," *I&EC Res*.39, 1955-1971 (2000). **Pressure and production curves reservoir**

Bizet, V.M., N. Juhasz and I.E. Grossmann, "Optimization Model for the Production and Scheduling of Catalyst Changeovers in a Process with Decaying Performance," *AIChE Journal*, 51, 909-921 (2005). **Reactor model**

Flores- Tlacuahuac, A. and I.E. Grossmann, "Simultaneous Cyclic Scheduling and Control of a Multiproduct CSTR," *Ind. Eng. Chem. Res*. 45, 6698-6712 (2006). **Dynamic models CSTR reactor**

Karuppiah, R., K. Furman, and I.E. Grossmann, "Global Optimization for Scheduling Refinery Crude Oil Operations," in preparation (2007). **Crude blending**

MINLP

- **Mixed-Integer Nonlinear Programming**

$$\min Z = f(x,y) \qquad \textit{Objective Function}$$

$$s.t. \quad g(x,y) \leq 0 \qquad \textit{Inequality Constraints}$$

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

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

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

- ***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**

Generalized Disjunctive Programming (GDP)

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

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

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

OR operator →

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

Disjunction

Constraints

Fixed Charges

$$\underline{\vee}_{j \in J} \; Y_{jk}$$

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

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

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

Can be transformed into MINLP or solved directly as a GDP

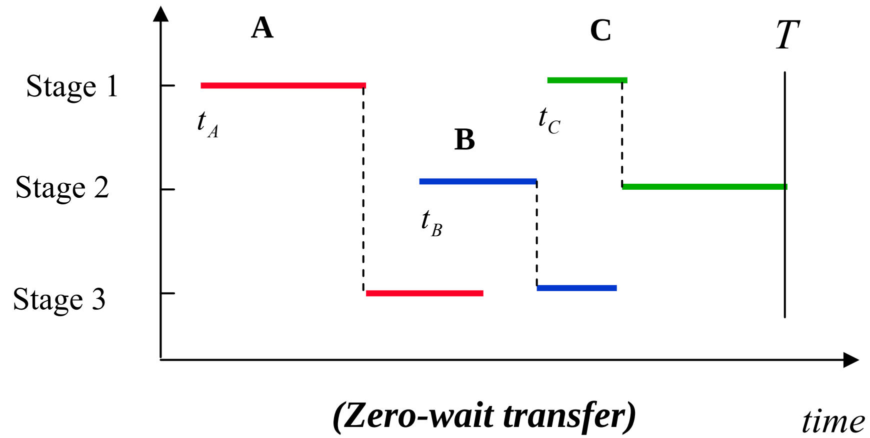
See previous work Sangbum Lee (2002) and Nick Sawaya (2006)

Jobshop Scheduling Problem

Processing times τ ***(hr)***

Jobs/Stages	1	2	3
A	5	0	3
B	0	3	2
C	2	4	0

Find sequence, times to minimize makespan T



(Zero-wait transfer)

GDP:

$$\min \quad MS = T$$

$$st \quad T \geq t_A + 8$$

$$T \geq t_B + 5$$

$$T \geq t_C + 6$$

$$(t_A - t_B \leq -5) \vee (t_B - t_A \leq 0)$$ **A before B or B before A**

$$(t_A - t_C \leq -5) \vee (t_C - t_A \leq -2)$$ **A before C or C before A**

$$(t_B - t_C \leq -1) \vee (t_C - t_B \leq -6)$$ **B before C or C before B**

$$T, t_A, t_B, t_C \geq 0$$

Carnegie Mellon

MILP:

Big-M reformulation

Parameter M “sufficiently” large

$$\min \quad MS = T$$

$$st \quad T \geq t_A + 8$$

$$T \geq t_B + 5$$

$$T \geq t_C + 6$$

$$t_A - t_B \leq -5 + M(1 - y_{AB})$$

$$t_B - t_A \leq 0 + M(1 - y_{BA})$$

$$t_A - t_C \leq -5 + M(1 - y_{AC})$$

$$t_C - t_A \leq -2 + M(1 - y_{CA})$$

$$t_B - t_C \leq -1 + M(1 - y_{BC})$$

$$t_C - t_B \leq -6 + M(1 - y_{CB})$$

$$y_{AB} + y_{BA} = 1$$

$$y_{AC} + y_{CA} = 1$$

$$y_{BC} + y_{CB} = 1$$

$$y_{AB}, y_{BA}, y_{AC}, y_{CA}, y_{BC}, y_{CB} = 0,1$$

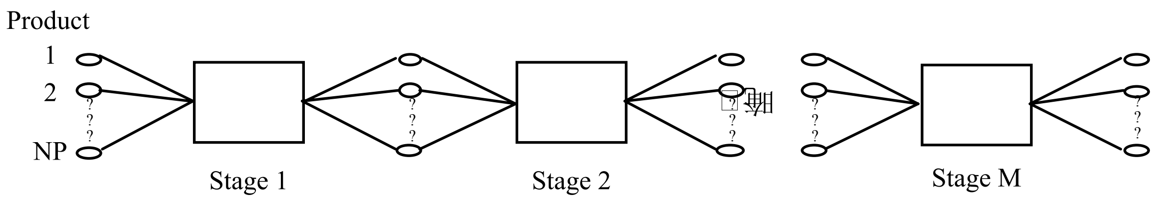
$$T, t_A, t_B, t_C \geq 0$$

Continuous multistage plants

Pinto, Grossmann (1994)

Cyclic schedules *(constant demand rates, infinite horizon)*

Intermediate storage



Given :

N Products
Transition times ***(sequence dependent)***
Demand rates

Determine :

PLANNING

Amount of products to be produced
Inventory levels

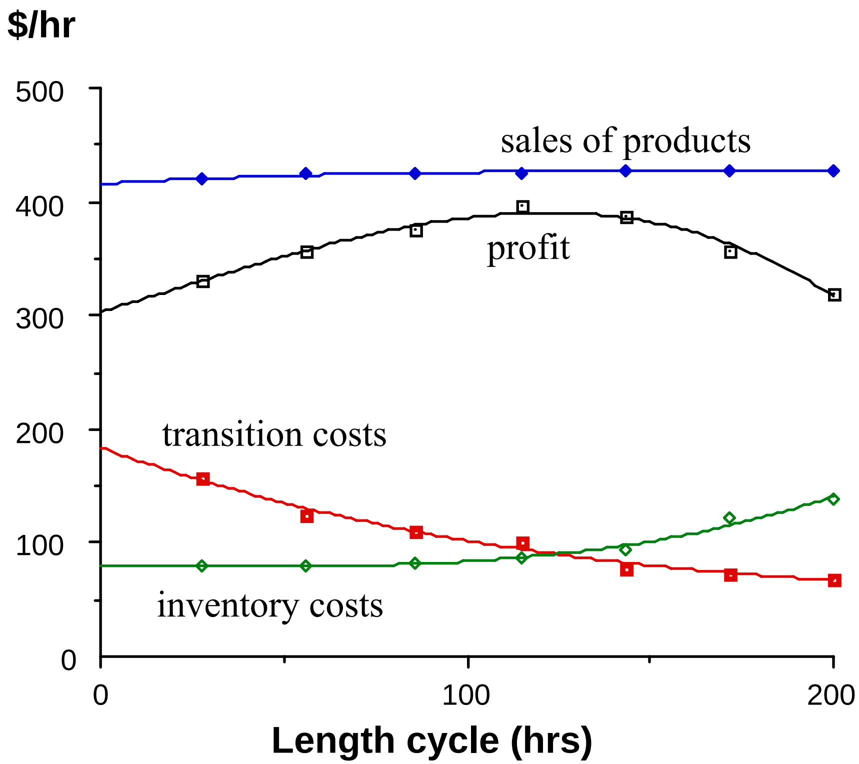
SCHEDULING

Cyclic production schedule
Sequencing
Lengths of production
Cycle time

Objective : Maximize Profit = + Sales of products - inventory costs - transition costs

Optimal Trade-offs

Optimal length of cycle determined largely by transition and inventory costs



Critical to model properly inventory levels and transition times

MINLP Model (1)

Assumption:

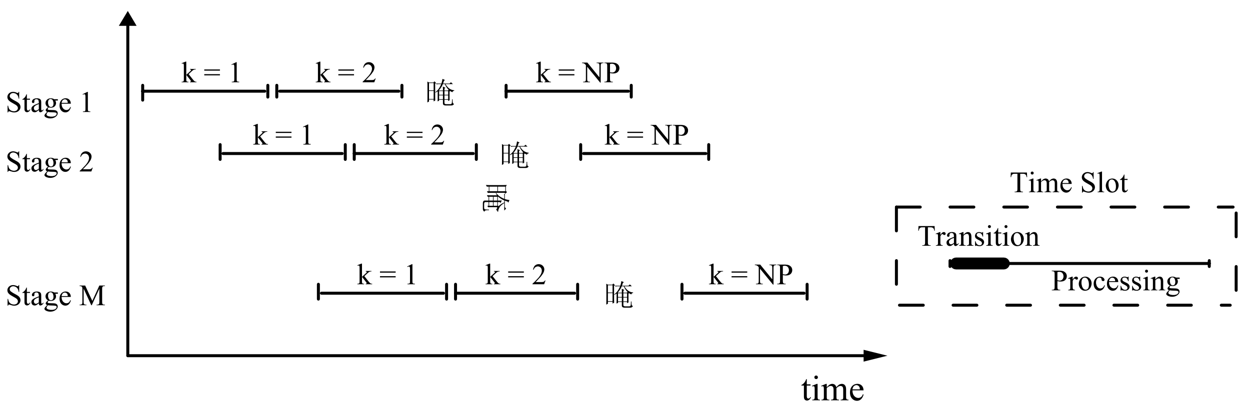
Each product is processed in the same sequence at each stage

Basic ideas :

a. NP products
b. NP time slots at each stage

Binary variable

$$y_{ik} = \begin{cases} 1 & \text{if product } i \text{ assigned to slot } k \\ 0 & \text{otherwise} \end{cases}$$



a) Assignments of products to slots and vice versa

$$\sum_k y_{ik} = 1 \qquad \forall i$$

$$\sum_i y_{ik} = 1 \qquad \forall k$$

MINLP Model (2)

b) **Definition of transition variables**

$$z_{ijk} \geq y_{ik} + y_{jk--1} - 1 \qquad \forall i, j \quad \forall k$$

z_{ijk} *treated as continuous*

c) **Processing rates, mass balances and amounts produced**

$$Wp_{ikm} = \gamma p_{im} Tpp_{ikm} \qquad \forall i \quad \forall k \quad \forall m$$

$$W_{km} = \gamma_{km} Tp_{km} \qquad \forall k \quad \forall m$$

$$Wp_{ikm} = \alpha_{im+1} Wp_{ikm+1} \qquad \forall i \quad \forall k \quad m = 1 \ldots M-1$$

d) **Timing constraints**

Processing time:

$$Tpp_{ikm} - U^T_{im} y_{ik} \leq 0 \qquad \forall i \quad \forall k \quad \forall m$$

$$Tp_{km} = \sum_i Tpp_{ikm} \qquad \forall k \quad \forall m$$

$$Tp_{km} = Te_{km} - Ts_{km} \qquad \forall k \quad \forall m$$

Transitions:

$$Ts_{k+1m} = Te_{km} + \sum_i \sum_j \tau_{ijm} z_{ijk+1} \qquad k = 1 \ldots NP-1 \quad \forall m$$

$$Ts_{11} = \sum_i \sum_j \tau_{ij1} z_{ij1}$$

$$Te_{km} \leq Te_{km+1} \qquad \forall k \quad m = 1 \ldots M-1$$

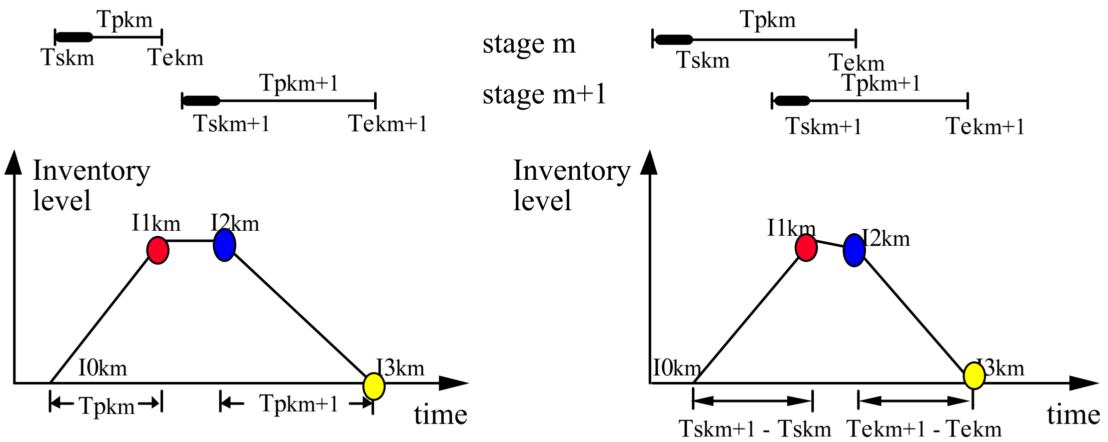
$$Ts_{km} \leq Ts_{km+1} \qquad \forall k \quad m = 1 \ldots M-1$$

Cycle time:

$$Tc \geq \sum_k \left(Tp_{km} + \sum_i \sum_j \tau_{ijm} z_{ijk} \right) \qquad \forall m$$

MINLP Model (3)

Inventories: Case 1 Case 2



e) Inventory levels for intermediates

$$I1_{km} = \gamma_{km} \min\{Tp_{km}, Ts_{km+1} - Ts_{km}\} + I0_{km} \qquad \forall k \qquad m = 1 \ldots M-1$$

$$I2_{km} = (\gamma_{km} - \alpha_{km+1}\gamma_{km+1})\max\{0, Te_{km} - Ts_{km+1}\} + I1_{km} \qquad \forall k \qquad m = 1 \ldots M-1$$

$$I3_{km} = -\alpha_{km+1}\gamma_{km+1} \min\{Te_{km+1} - Te_{km}, Tp_{km+1}\} + I2_{km} \qquad \forall k \qquad m = 1 \ldots M-1$$

Can be reformulated as 0-1 linear inequalities

$$0 \le I1_{km} \le Imax_{km} \qquad \forall k \qquad m = 1 \ldots M-1$$
$$0 \le I2_{km} \le Imax_{km} \qquad \forall k \qquad m = 1 \ldots M-1$$
$$0 \le I3_{km} \le Imax_{km} \qquad \forall k \qquad m = 1 \ldots M-1$$

$$Imax_{km} = \sum_i Ip_{ikm} \qquad \forall k \qquad \forall m$$

$$Ip_{ikm} - U^I_{im} y_{ik} \le 0 \qquad \forall i \qquad \forall k \qquad m = 1 \ldots M$$

Carnegie Mellon

MINLP Model (4)

f) Demand constraints

$$\sum_k Wp_{ikM} \geq d_i Tc \qquad \forall i$$

Note: Linear

g) Objective function : Maximize Profit

Note: Nonlinear (divide by Tc)

$$Profit = \sum_i \sum_k p_i \frac{Wp_{ikM}}{Tc} \qquad INCOME$$

$$-\sum_i \sum_j \sum_k Ctr_{ij} \frac{z_{ijk}}{Tc} \qquad TRANSITION\ \ COST$$

$$-\sum_i \sum_k \sum_m Cinv_{im} \frac{Ip_{ikm}}{Tc} \qquad INTERMEDIATE\ \ INVENTORY\ \ COST$$

$$-\frac{1}{2} \sum_i \sum_k Cinvf_i \left(\gamma p_{iM} - \frac{Wp_{ikM}}{Tc} \right) Tpp_{ikM} \qquad FINAL\ \ INVENTORY\ \ COST$$

$\Rightarrow$ **MINLP**

Solution Algorithms for MINLP

- **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*

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

b) NLP Fixed y^k *Upper bound*

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

c) Feasibility subproblem for fixed y^k.

$$\min u$$
$$s.t.\ \ g_j(x,y^k) \le u \quad j \in J$$
$$x \in X,\ u \in R^1$$

u > 0 => infeasible
(NLPF)

Infinity-norm

Carnegie Mellon

Cutting plane MILP master

(Duran and Grossmann, 1986)

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

Lower Bound

M-MIP

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

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

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

Notes:

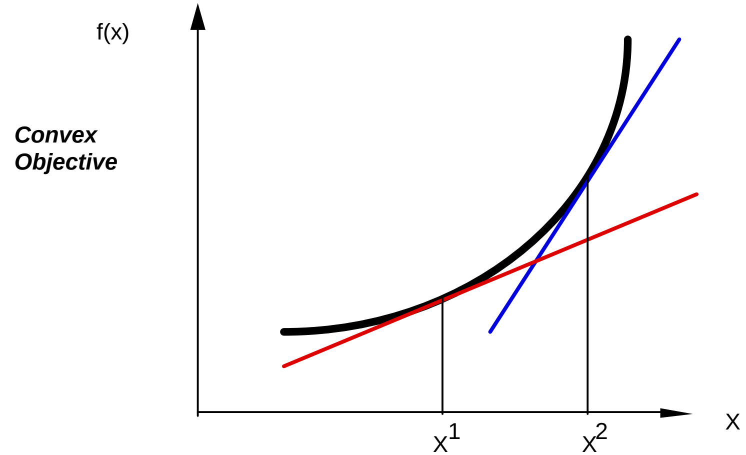
a) Point (x^k, y^k) $k=1,\ldots K$ normally from NLP2

b) Linearizations *accumulated* as iterations K increase

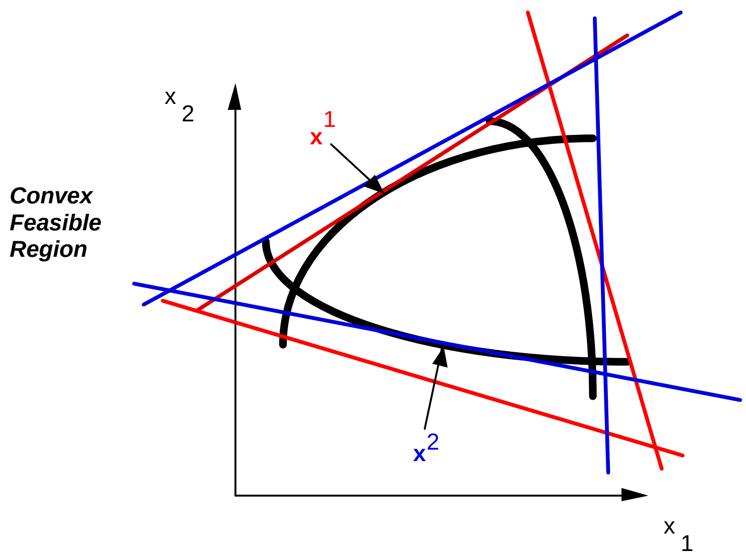
c) Non-decreasing sequence lower bounds

Linearizations and Cutting Planes

Supporting hyperplanes



Underestimate Objective Function



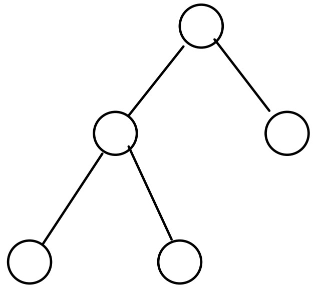
Overestimate Feasible Region

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 *($I_{FL}=I_{FU}=\emptyset$)*

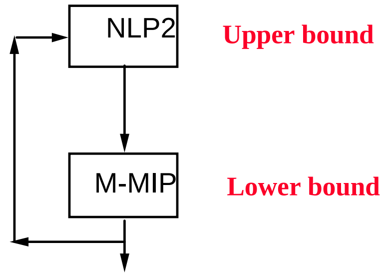
Disadvantage:
Potentially many NLP subproblems

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

Less stringent than other methods

Outer-Approximation

Alternate solution of NLP and MIP problems:



NLP2:

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

M-MIP:

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

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.

MIP Master problem need not be solved to optimality

Find new y^{k+1} such that predicted objetive lies below current upper bound UB^K:

(M-MIPF)

$$\min \quad Z_L^K = 0\alpha$$

$$s.t. \quad \boxed{\alpha \geq UB^K - \varepsilon}$$

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

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

Remark.

M-MIPF will tend to increase number of iterations

Generalized Benders Decomposition

Benders (1962), Geoffrion (1972)

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

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

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

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

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

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

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

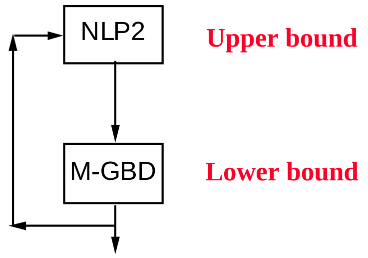
Lagrangian cut

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

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

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:

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

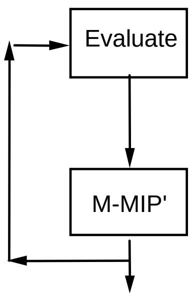
Property 1. If problem (P1) has zero integrality gap, Generalized Benders Decomposition converges in one iteration when optimal *(x^k, y^k)* are found.

Sahinidis, Grossmann (1991)

=> Also applies to Outer-Approximation

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

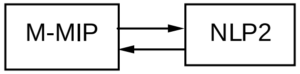
= > ***Convergence may be slow***

LP/NLP Based Branch and Bound

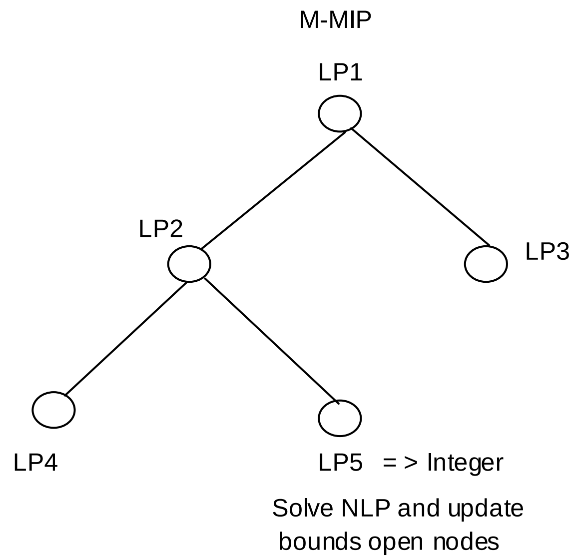
Quesada and Grossmann (1992)

(a.k.a Branch & Cut; Hybrid)

Integrate NLP and M-MIP problems



Solve NLPs at selected nodes of tree from master MILP and add cutting planes



Remark.

Fewer number branch and bound nodes for LP subproblems

May increase number of NLP subproblems

Carnegie Mellon

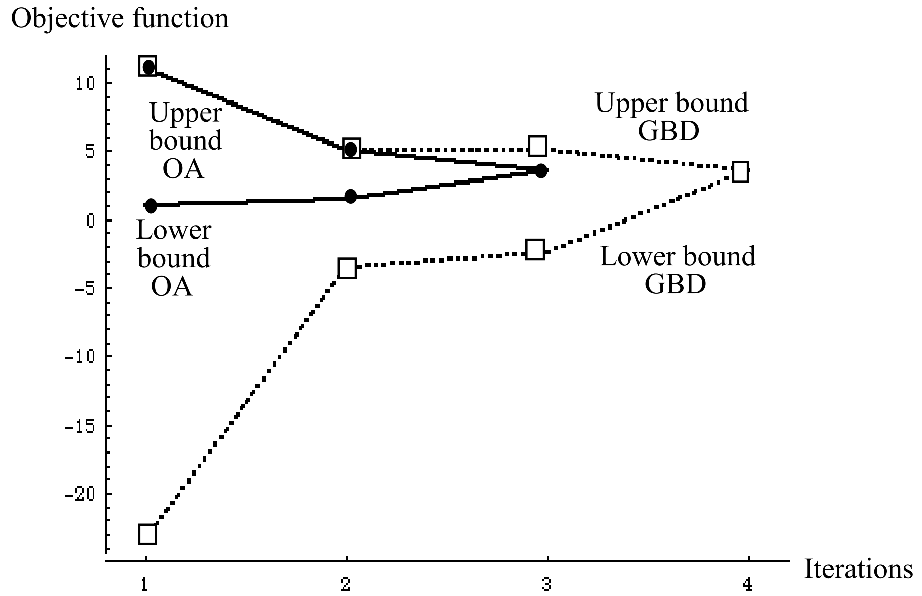
Numerical Example

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

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

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

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



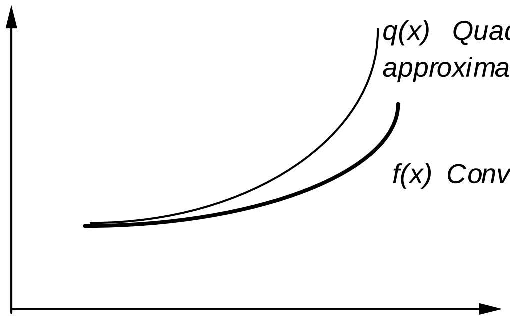
Summary of Computational Results

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)

Mixed-Integer Quadratic Programming

Fletcher and Leyffer (1994)

Quadratic-Approximation may not provide valid bounds for convex function



Add quadratic objective to feasibility M-MIPF
Proceed as OA

M-MIQP:

$$\min Z^K = \alpha + \frac{1}{2}\begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix}^T \nabla^2 L(x^k, y^k) \begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix}$$

$$s.t. \quad \alpha \le UB^K - \varepsilon$$

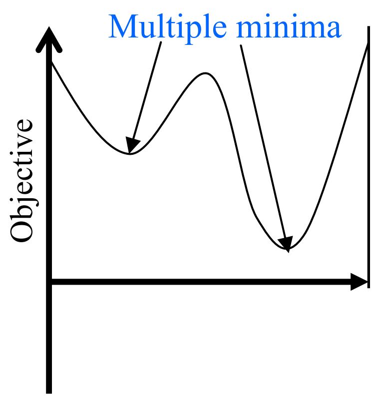
$$\left.\begin{array}{l} \alpha \ge f(x^k, y^k) + \nabla f(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ g(x^k, y^k) + \nabla g(x^k, y^k)^T \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le 0 \end{array}\right\} \quad k = 1, \ldots K$$

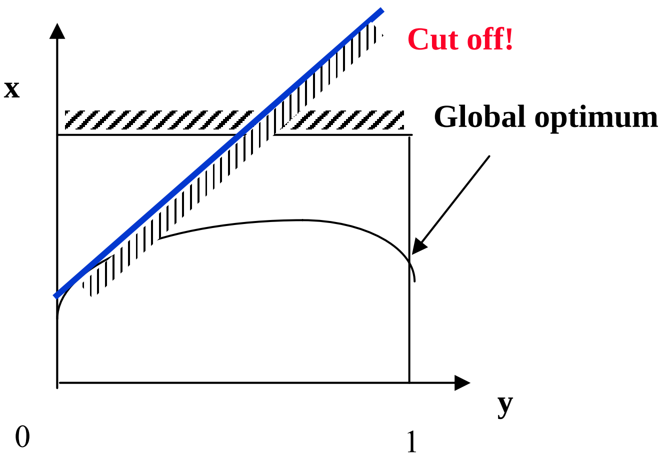
$$x \in X, \quad y \in Y, \quad \alpha \in \mathbf{R}^1$$

Remark. Faster convergence if problems nonlinear in y

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):**

   Replace nonconvex terms by underestimtors/convex envelopes
   Solve convex MINLP within spatial branch and bound

2. **Heuristic approach:**

   Assume lower bound valid
   Add slacks to linearizations MILP
   Search until no improvement 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

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

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

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

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

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

Mixed-integer Nonlinear Programming

MINLP Codes:

SBB *Bussieck, Drud (2003) (B&B)*
MINLP-BB (AMPL)*Fletcher and Leyffer (1999)(B&B-SQP)*

Bonmin (COIN-OR) *Bonami et al (2006) (B&B, OA, Hybrid)*
FilMINT Linderoth and Leyffer (2006) *(Hybrid-MINTO-FilterSQP)*

DICOPT (GAMS) *Viswanathan and Grossman (1990) (OA)*
AOA (AIMSS) *(OA)*

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

Global MINLP code:

BARON *Sahinidis et al. (1998) (Global Optimization)*

MIQP codes:

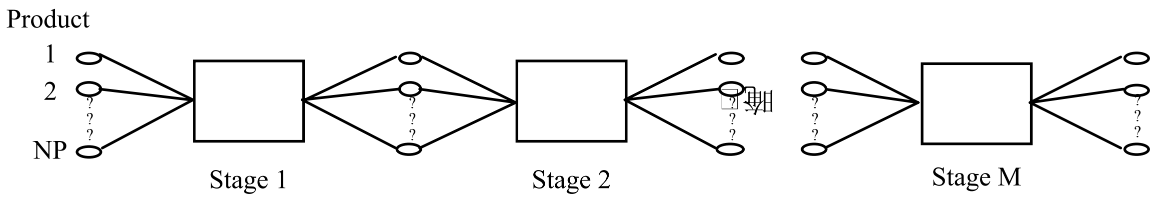
CPLEX-MIQP *ILOG (Branch and bound, cuts)*
MIQPBB *(Fletcher, Leyffer, 1999)*

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

Continuous multistage plants

Cyclic schedules *(constant demand rates, infinite horizon)*

Intermediate storage



Given :

N Products
Transition times ***(sequence dependent)***
Demand rates

Determine :

PLANNING

Amount of products to be produced
Inventory levels

SCHEDULING

Cyclic production schedule
Sequencing
Lengths of production
Cycle time

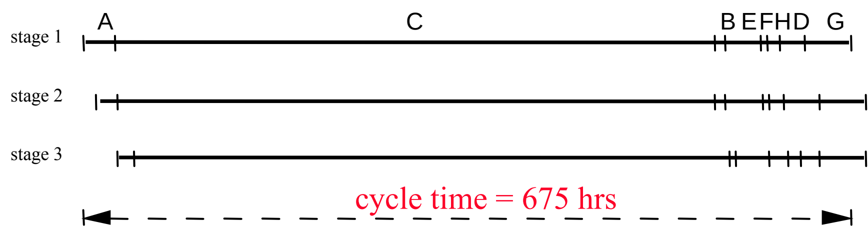
Objective : Maximize Profit = + Sales of products - inventory costs - transition costs

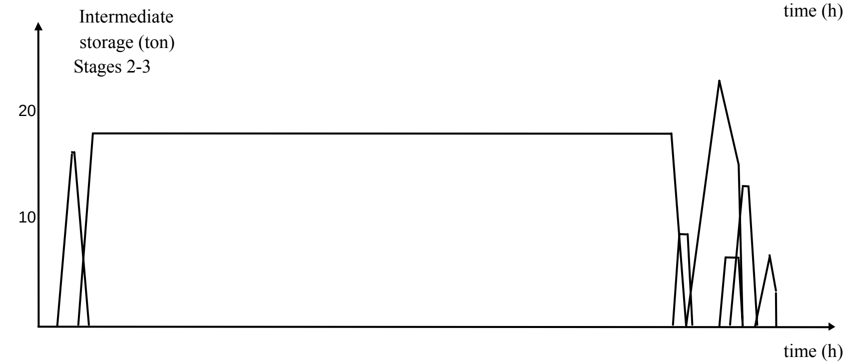
Example 3 stages, 8 products, 3

MINLP model 448 binary 0-1 variables, 2050 continuous variables, 3010 constraints

DICOPT (CONOPT/CPLEX): 38.2 secs

Optimal solution **Profit = $6609/h**





Bonmin (COIN-OR)

Bonami, Biegler, Conn, Cornuejols, Grossmann, Laird, Lee, Lodi, Margot, Sawaya, Wächter

Single computational framework (C++) 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 (extensions)

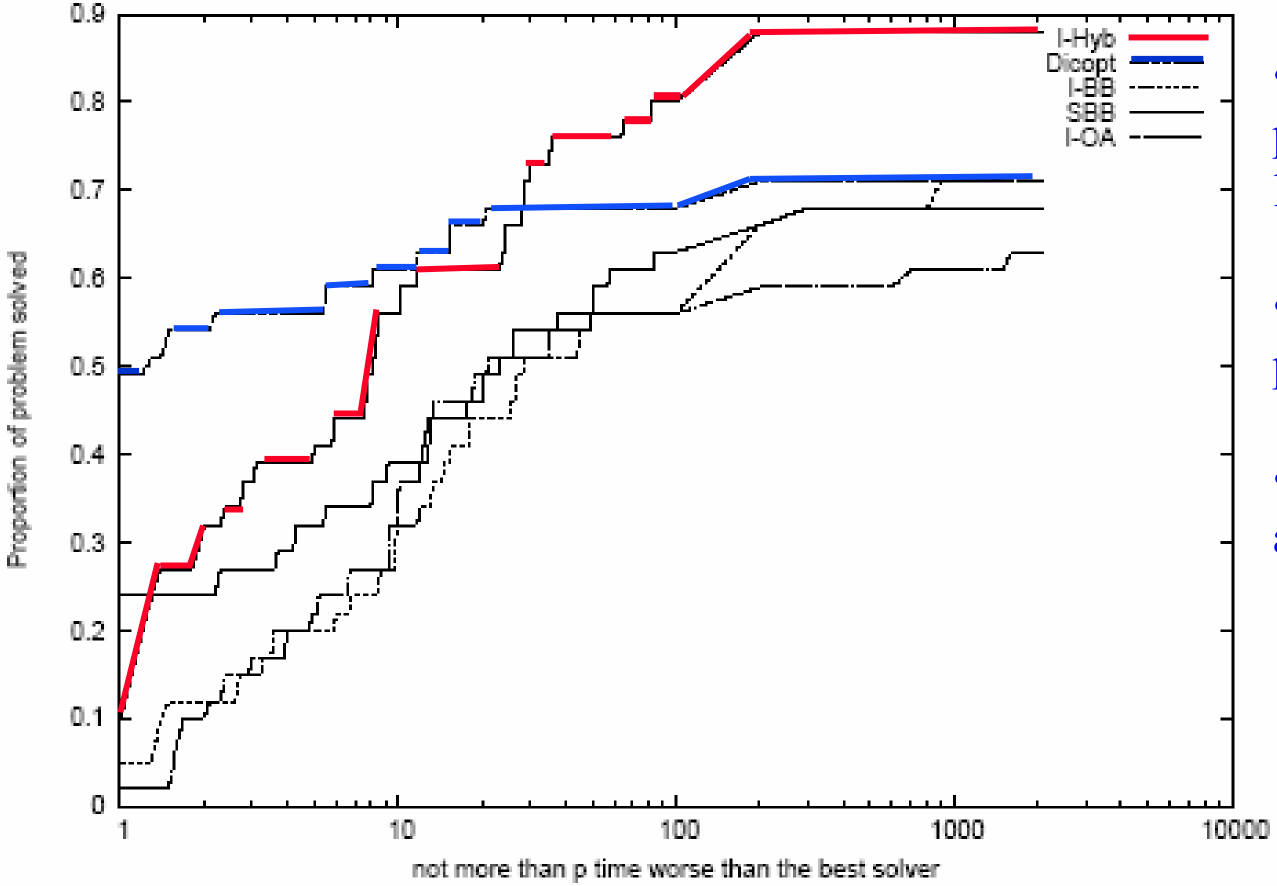
http://projects.coin-or.org/Bonmin

Convex MINLP Test Problems

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

Computational performance: Comparison of 3 variants with DICOPT and SBB

Largest: 586 0-1 vars, 2720 cont., 4980 constr.



- DICOPT solves 20 of the 38 problems the fastest and in less than 3minutes
- I-Hyb solves the most problem in the time limit
- I-Hyb always dominate I-BB and I-OA

Safety Layout Problem

MIQP

Sawaya (2006)

Determine placement a set of rectangles with fixed width and length such that the Euclidean distance between their center point and a pre-defined "safety point" is minimized.

$$Min \qquad Q = \sum_i \sum_j c_{ij}(delx_{ij} + dely_{ij}) + \sum_i Cost_i((x_i - x^0{}_i)^2 + (y_i - y^0{}_i)^2)$$

$s.t.$

$$delx_{ij} \geq x_i - x_j \qquad \forall i,j \in N, i<j$$

$$delx_{ij} \geq x_j - x_i \qquad \forall i,j \in N, i<j$$

$$dely_{ij} \geq y_i - y_j \qquad \forall i,j \in N, i<j$$

$$dely_{ij} \geq y_j - y_i \qquad \forall i,j \in N, i<j$$

$$\begin{bmatrix} Z^1{}_{ij} \\ x_i + L_i/2 \leq x_j - L_j/2 \end{bmatrix} \vee \begin{bmatrix} Z^2{}_{ij} \\ x_j + L_j/2 \leq x_i - L_i/2 \end{bmatrix} \vee \begin{bmatrix} Z^3{}_{ij} \\ y_i + H_i/2 \leq y_j - H_j/2 \end{bmatrix} \vee \begin{bmatrix} Z^4{}_{ij} \\ y_j + H_j/2 \leq y_i - H_i/2 \end{bmatrix} \quad \forall i,j \in N, i<j$$

$$x_i \leq UB^1{}_i \qquad \forall i \in N$$

$$x_i \geq LB^1{}_i \qquad \forall i \in N$$

$$y_i \leq UB^2{}_i \qquad \forall i \in N$$

$$y_i \geq LB^2{}_i \qquad \forall i \in N$$

$$delx_{ij}, dely_{ij} \in \mathbb{R}^1{}_+ , Z^1{}_{ij}, Z^2{}_{ij}, Z^3{}_{ij}, Z^4{}_{ij} \in \{True, False\} \qquad \forall i,j \in N, i<j$$

Numerical results MIQP

Small instance: 5 rectangles

40 0-1 vars, 31 cont. vars., 91 constr.

SBB (CONOPT)	**281 nodes, 2.4 sec**
DICOPT (CONOPT/CPLEX)	**19 major iterations, 5.2 sec**
CPLEX-MIQP	**18 nodes, 0.06 secs**

Larger instance: 10 rectangles

180 0-1 vars, 111 cont. vars., 406 constr.

Bonmin-BB (IPOT)	**16,072 nodes, 514.6 sec**
Bonmin-Hybrid (Cbc, IPOPT)	**6,548 nodes (*563 NLPs*) 197.9 sec**
CPLEX-MIQP	**1,093 nodes, 1.6 secs**

Global Optimization Algorithms

- **Most algorithms are based on spatial branch and bound method** *(Horst & Tuy, 1996)*

- **Nonconvex NLP/MINLP**
  - **αBB** (Adjiman, Androulakis & Floudas, 1997; 2000)
  - **BARON (Branch and Reduce)** (Ryoo & Sahinidis, 1995, Tawarmalani and Sahinidis (2002))
  - **OA for nonconvex MINLP** (Kesavan, Allgor, Gatzke, Barton, 2004)
  - **Branch and Contract** (Zamora & Grossmann, 1999)

- **Nonconvex GDP**
  - **Two-level Branch and Bound** (Lee & Grossmann, 2001)

Nonconvex MINLP

$$\min Z = f(x,y)$$
$$s.t. \quad g(x,y) \le 0$$
$$x \in X,\ y \in Y$$

$f(x,y),\ g(x,y)$ *nonconvex*

⇓

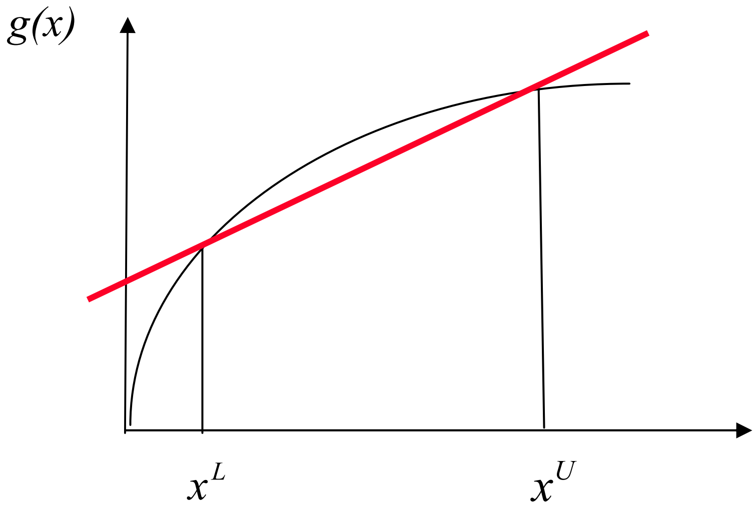
Convex MINLP *(relaxation)*

$$\min Z = \hat{f}(x,y)$$
$$s.t. \quad \hat{g}(x,y) \le 0$$
$$x \in X,\ y \in Y$$

=> ***Lower bound***

$\hat{f}(x,y),\ \hat{g}(x,y)$ *convex underestimators, convex envelopes*

Concave function *g(x)*



Convex envelope: secant

$$\hat{g} = g(x^L) + \left(\frac{g(x^U) - g(x^L)}{x^U - x^L} \right) (x - x^L)$$

$$x^L \le x \le x^U$$

Bilinear terms $w = xy$

Convex envelopes
bilinear term

$$w = xy \qquad x^L \le x \le x^U, \; y^L \le y \le y^U$$

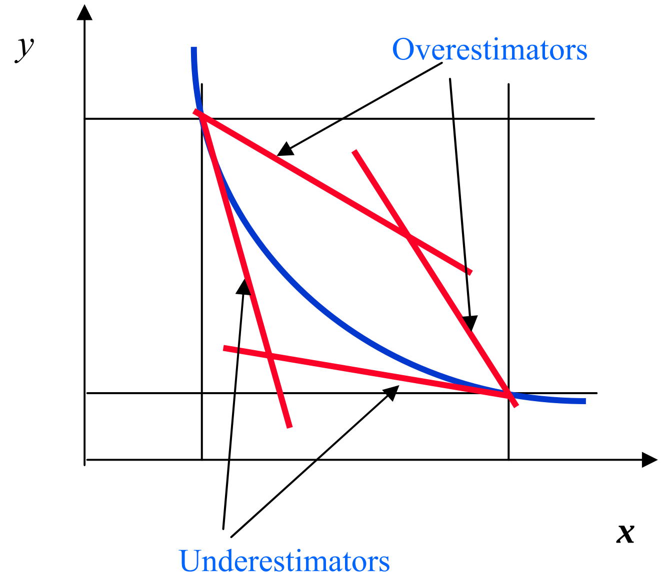
McCormick (1976) under/over estimators

$$w \ge x^L y + y^L x^i - x^L y^{iL}$$

$$w \ge x^U y + y^U x - x^U y^U$$

$$w \le x^L y + y^U x - x^L y^U$$

$$w \le x^U y + y^L x - x^U y^L$$

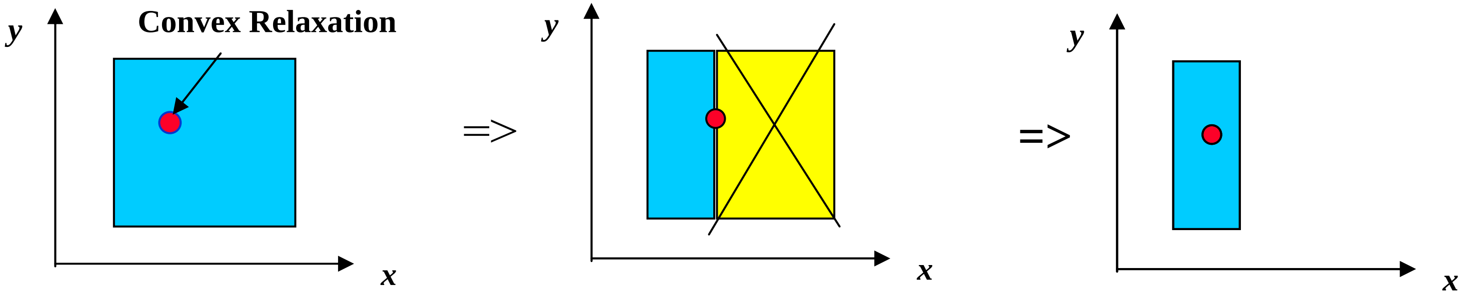


For other convex envelopes/underestimators see:

Tawarmalani, M. and N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, Vol. 65, *Nonconvex Optimization And Its Applications* series, Kluwer Academic Publishers, Dordrecht, 2002

Spatial Branch and Bound Method

1. Generate subproblems by branching on continuous variables (subregions)



2. Compute lower bound from convex MINLP (relaxation)

3. Compute upper bound from local solution to nonconvex MINLP

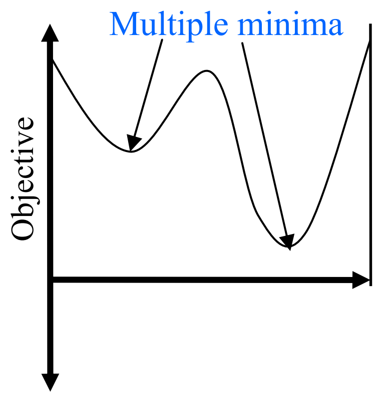
Continue until tolerance of bounds within tolerance

► *Guaranteed to converge to global optimum given a certain tolerance between lower and upper bounds*

good upper bound, generation cutting planes

Example spatial branch and bound

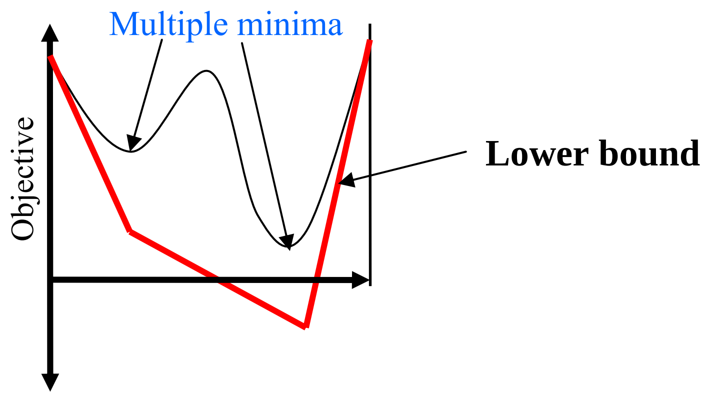
Global optimum search



Branch and bound tree

Example spatial branch and bound

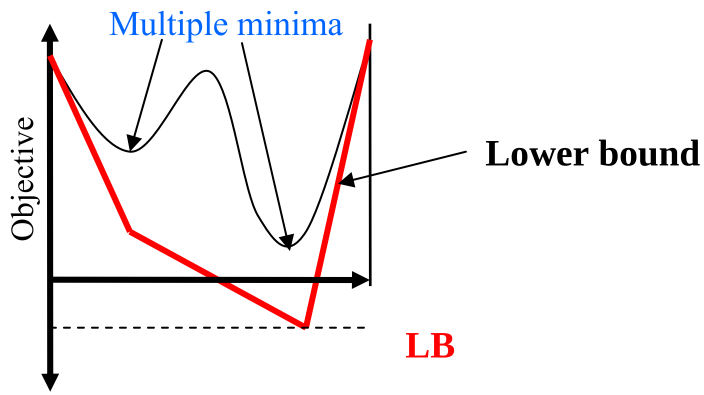
Global optimum search



Branch and bound tree

Example spatial branch and bound

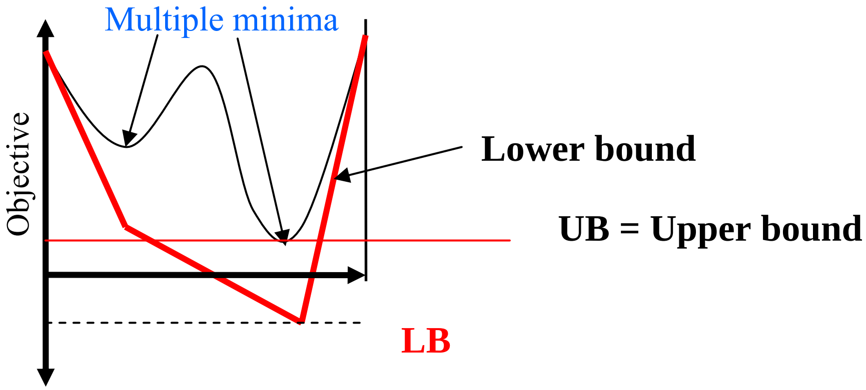
Global optimum search



Branch and bound tree

Example spatial branch and bound

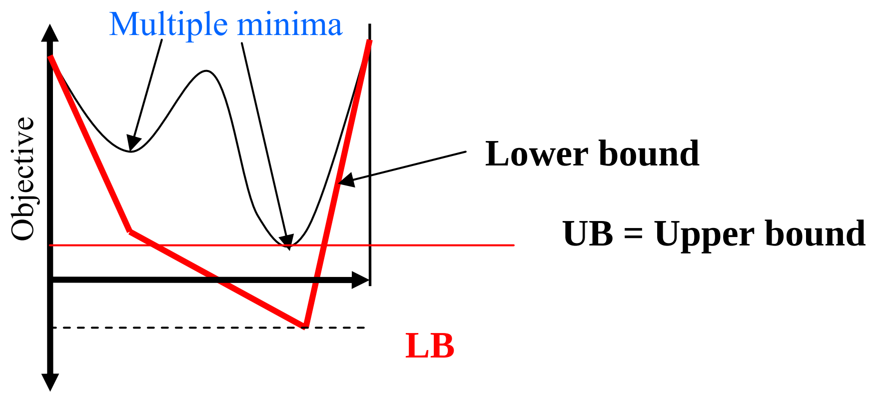
Global optimum search



Branch and bound tree

Example spatial branch and bound

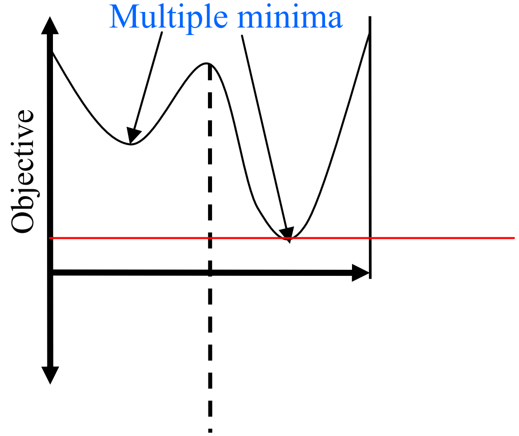
Global optimum search



Branch and bound tree

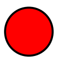
Example spatial branch and bound

Global optimum search



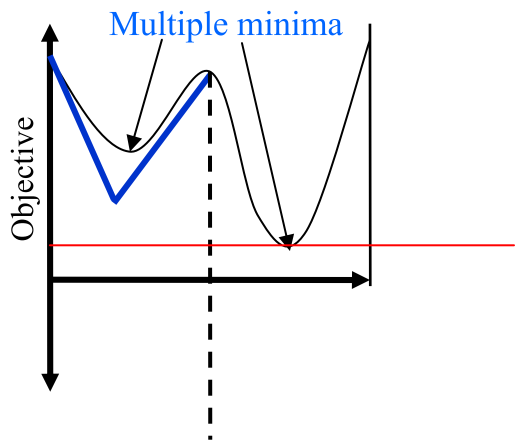
UB = Upper bound

Branch and bound tree



Example spatial branch and bound

Global optimum search

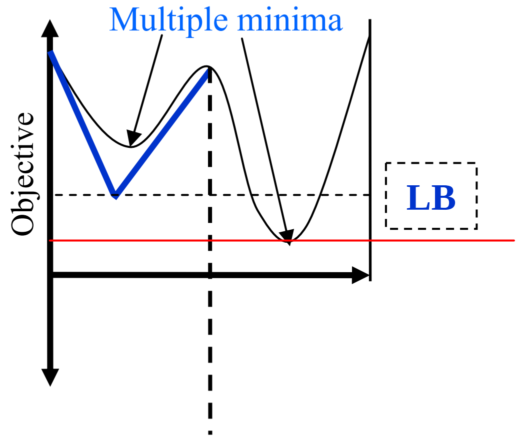


UB = Upper bound

Branch and bound tree

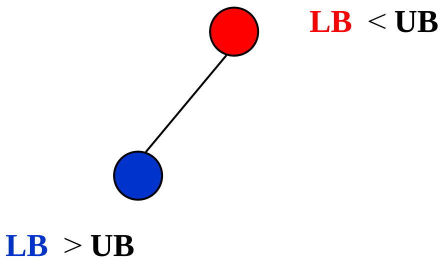
Example spatial branch and bound

Global optimum search



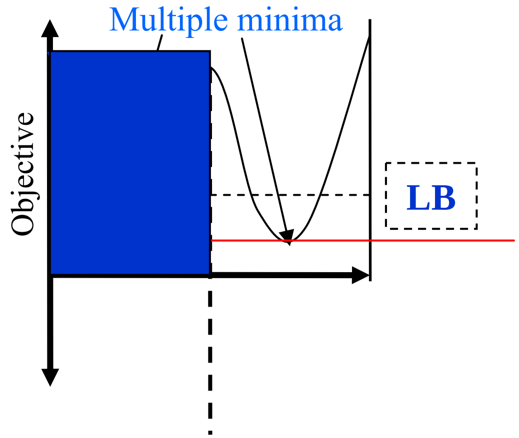
UB = Upper bound

Branch and bound tree



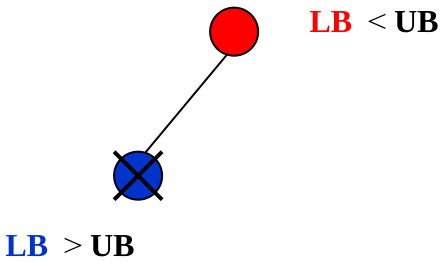
Example spatial branch and bound

Global optimum search



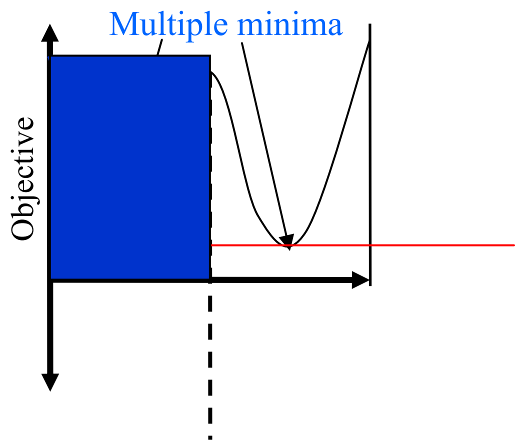
UB = Upper bound

Branch and bound tree



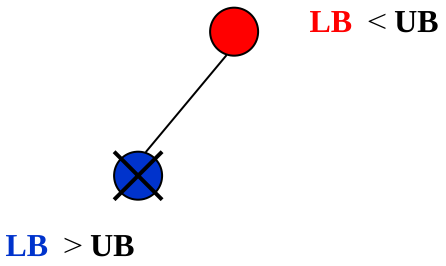
Example spatial branch and bound

Global optimum search



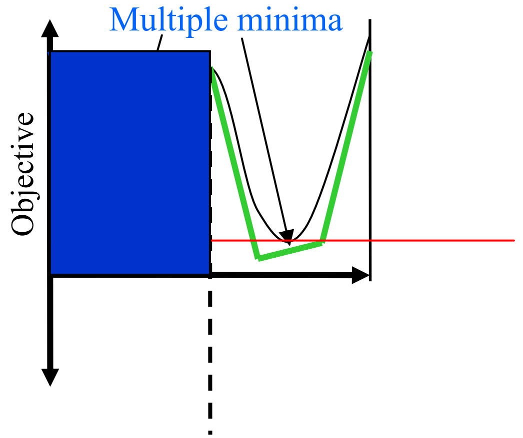
UB = Upper bound

Branch and bound tree



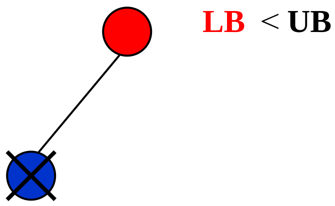
Example spatial branch and bound

Global optimum search



UB = Upper bound

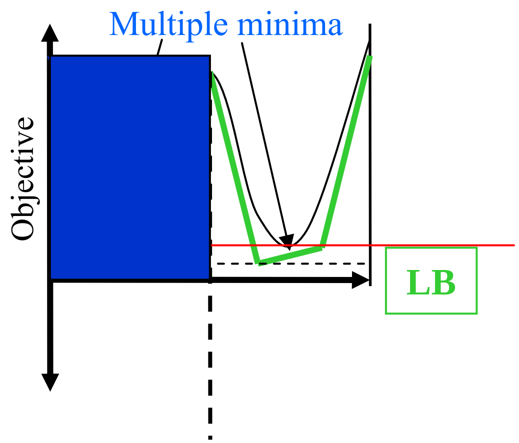
Branch and bound tree



LB > **UB**

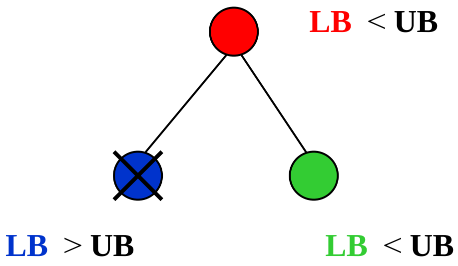
Example spatial branch and bound

Global optimum search



UB = Upper bound

Branch and bound tree



Recent development in BARON *Sahinidis (2005)*

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

Cutting planes:

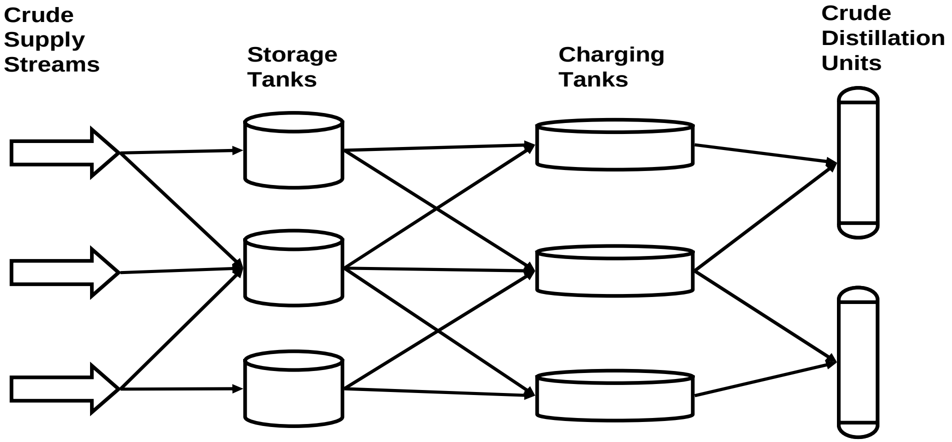
Supporting hyperplanes (outer-approximations) of convex functions

Application Global MINLP in Planning/Scheduling

Currently limited due to problem size
=> Need special purpose methods

Example: *Karuppiah, Furman, Grossmann (2007)*

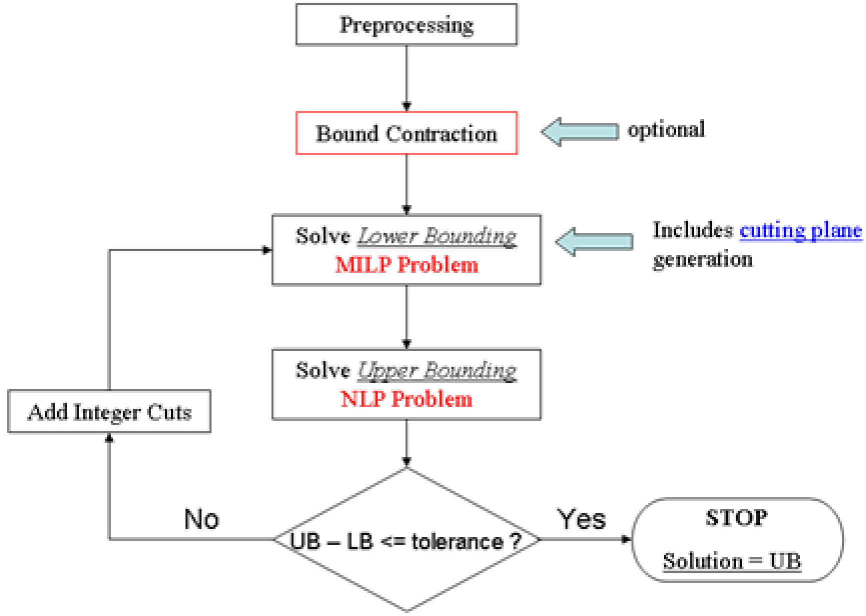
Scheduling Refinery Crude Oil Operations

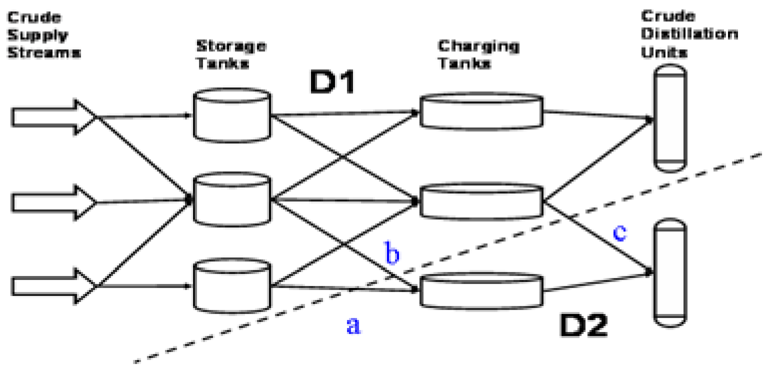


Source non-convexities: bilinearities in mass balances

Carnegie Mellon

- Variant of **Outer-Approximation** (Duran and Grossmann, 1986)





1. **Decompose using Lagrangean Decomposition**
2. **Globally optimize each subsystem with BARON**
3. **Derive Lagrangean cutting planes**

$$z_n^* \le w_n s(x,y) + r_n(u_n,v_n) + (\bar{\lambda}_n^x - \bar{\lambda}_{n-1}^x)^T(x) + (\bar{\lambda}_n^y - \bar{\lambda}_{n-1}^y)^T(y)$$

Iterate on Lagrange multipliers

- Network is split into two decoupled sub-structures D1 and D2
  - Physically interpreted as cutting some pipelines (Here ***a***, ***b*** and ***c***)
  - Set of split streams denoted by p $\in$ {a , b, c }

Carnegie Mellon

Numerical Results

Size MINLP Problems

3 Supply streams – 3 Storage tanks – 3 Charging tanks – 2 Distillation units

3 Supply streams – 3 Storage tanks – 3 Charging tanks – 2 Distillation units

3 Supply streams – 6 Storage tanks – 4 Charging tanks – 3 Distillation units

Example	Original MINLP model (P)		
	Number of Binary Variables	Number of Continuous Variables	Number of Constraints
1	**48**	**300**	**946**
2	**42**	**330**	**994**
3	**57**	**381**	**1167**

Results for Proposed Algorithm Root Node

Example	**Lower bound** [obtained by solving relaxation (RP)] (z^{RP})	**Upper bound** [on solving (P-NLP) using BARON] ($z^{P\text{-}NLP}$)	**Relaxation gap (%)**	**Total time** taken for one iteration[1] of algorithm (CPUsecs)	Local optimum (using DICOPT)
1	**281.14**	**282.19**	**0.37**	**827.7**	**291.93**
2	**351.32**	**359.48**	**2.27**	**6913.9**	**361.63**
3	**383.69**	**383.69**	**0**	**8928.6**	**383.69**

[1] Total time includes time for generating a pool of cuts, updating Lagrange multipliers, solving the relaxation (RP) using CPLEX and solving (P-NLP) using BARON

Conclusions

1. MINLP Optimization

Not widespread in planning/scheduling but increasing interest

Significant progress has been made

More software is available: *commercial, open-source*

MINLP problems of significant size can be solved

Convex case: rigorous global optimality

Modeling, efficiency and robustness still issues

2. Global Optimization Nonconvex MINLP

Convex MINLP used as a basis

Key: spatial branch and bound

Main issue is scaling

Special purpose techniques may be required

Open-source: Work is under way CMU-IBM: Margot, Belotti (Tepper)

Practical approach: ignore nonconvexities, or use heuristics