MINLP applications, part I: Hydro Unit Commitment and Pooling Problem

Claudia D'Ambrosio

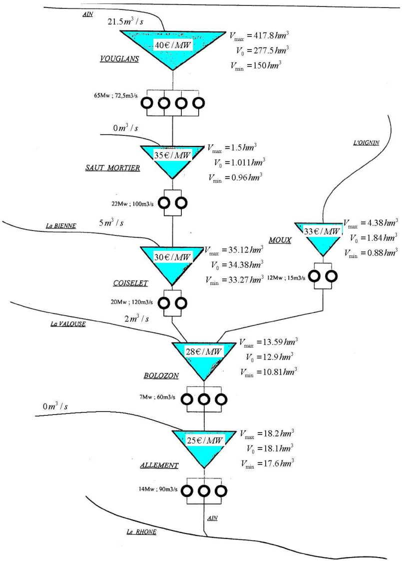
CNRS & LIX, École Polytechnique

5th Porto Meeting on MATHEMATICS for INDUSTRY, April 10, 2014

Hydro UC: Introduction

- Production planning of hydro valley (short-term: from 1 day to 1 week).
- Crucial problem in energy management: **hydro valley management**.
- Combinatorial elements leads to far tougher hydro valley problems.
- French Hydro valleys.

Example



AIN
21.5 m³ / s
40€ / MW
VOUGLANS
Vmax = 417.8 hm³
V0 = 277.5 hm³
Vmin = 150 hm³
65Mw ; 72,5m3/s
0 m³ / s
35€ / MW
SAUT MORTIER
Vmax = 1.5 hm³
V0 = 1.011 hm³
Vmin = 0.96 hm³
22Mw ; 100m3/s
L'OIGNIN
La BIENNE
5 m³ / s
33€ / MW
MOUX
Vmax = 4.38 hm³
V0 = 1.84 hm³
Vmin = 0.88 hm³
30€ / MW
COISELET
Vmax = 35.12 hm³
V0 = 34.38 hm³
Vmin = 33.27 hm³
12Mw ; 15m3/s
20Mw ; 120m3/s
2 m³ / s
La VALOUSE
28€ / MW
BOLOZON
Vmax = 13.59 hm³
V0 = 12.9 hm³
Vmin = 10.81 hm³
7Mw ; 60m3/s
0 m³ / s
25€ / MW
ALLEMENT
Vmax = 18.2 hm³
V0 = 18.1 hm³
Vmin = 17.6 hm³
14Mw ; 90m3/s
AIN
Le RHONE

Mathematical Model

Decision variables:

- Activation of turbines/pumps: binary
- Flow through turbines/pumps: continuous

Maximize profit: depends on produced power (non-linear function of the flow and dependent variables)

Constraints:

- Bounds and target on water volume in the reservoirs
- Flow conservation
- If turbine is active, minimum and maximum flow
- Flow variation limit from one time period to the other
- Either turbines or pumps on in the same period/unit

Solution Approaches

Practically difficult: **complicated constraints**, and **large size of real instances.**

Looking for **provable high accuracy in a limited amount of time.**

Efficient Modeling: formulation strengthening, cuts, decomposition methods, and approximations to efficiently provide effective lower bounds on the optimal value

Decompositions

The subproblem might be itself decomposed into smaller sub-subproblems.
For example, the only constraints that link the different hydro plants are the
Flow conservation constraint ($\forall n \in \mathcal{N}, t \in \mathcal{T}$):

$$
\begin{aligned}
v_{nt} &= v_{n(t-1)} \\
&+ \sum_{m \in \mathcal{F}_n : D_{(m,n)} \leq t} \; \sum_{u \in \mathcal{U} : \mu_u = (m,n)} x_{u(t-D_{(m,n)})} T \\
&- \sum_{m \in \mathcal{F}_n} \; \sum_{u \in \mathcal{U} : \mu_u (n,m)} x_{ut} T \\
&+ \sum_{m \in \mathcal{D}_n : D_{(m,n)} \leq t} \; \sum_{p \in \mathcal{P} : (n,m)} y_{p(t-D_{(m,n)})} T \\
&- \sum_{m \in \mathcal{D}_n} \; \sum_{p \in \mathcal{P} : \mu'_p = (m,n)} y_{pt} T + I_{nt} T
\end{aligned}
$$

Approximations

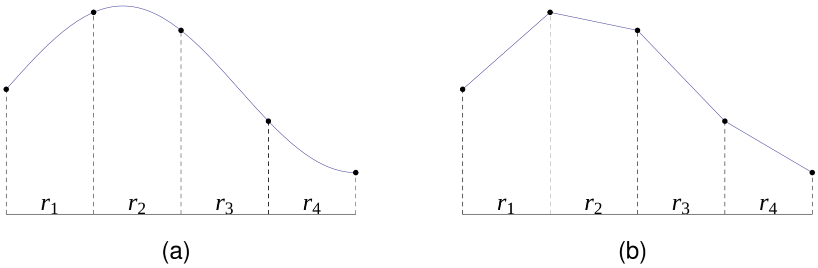
- Real-world optimization problem can be often modeled as a MINLP problem.
- What makes MINLP problem difficult?
  1. non-linear functions;
  2. integer variables.
- MILP solvers more efficient than MINLP ones and handle large-scale instances.
- Trying to get rid of the non-linear functions $\rightarrow$ "linearize" and use MILP solvers!!!!
- **Piecewise linear approximation**: Beale & Tomlin, 1970 (*Special Ordered Sets*).

For the moment, focus on MINLP with non-linear objective function and linear constraints .

Starting simple: univariate function

Consider a function $f(x)$ and construct its piecewise linear approximation.

- Divide the domain of f in $n-1$ intervals of coordinates $x_1, \ldots, x_n$.
- Sample f at each point x_i with $i = 1, \ldots, n$.
- The piecewise linear approximation of f is given by the convex combination of the samples.



Function of 2 variables: Method 1

1. Simply fix the value of one of the 2 variables and obtain a univariate function: $f(x, \tilde{y})$.
2. Apply methods for approximating univariate functions (previous slide).

The quality of the approximation depends on the function at hand.

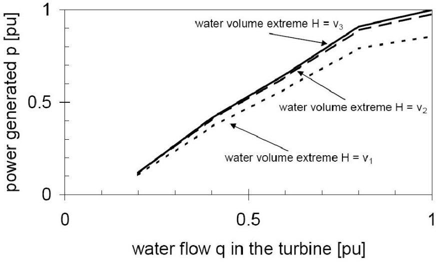
Choose to fix the “less non-linear” variable.

Function of 2 variables: Method 2

In Conejo et al. (2002) the function $f^a = f(x, y)$ was approximated by considering three prefixed water volumes, say $\widetilde{y}^1$, $\widetilde{y}^2$, $\widetilde{y}^3$ and interpolating, for each $\widetilde{y}^r$, the resulting function

$$f^a = f(x, \widetilde{y}^r)$$

by piecewise linear approximation.

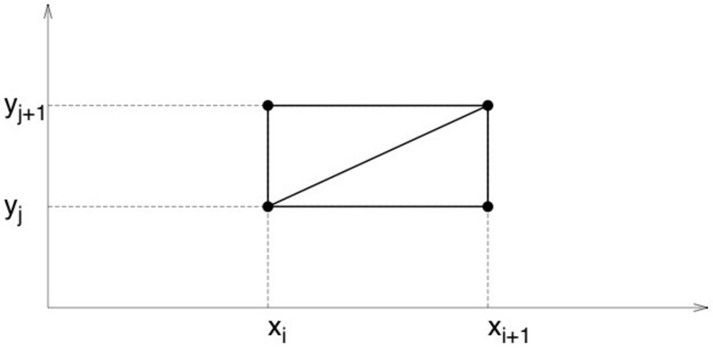


It can be **generalized** by approximating a prefixed number *m* of values of y.

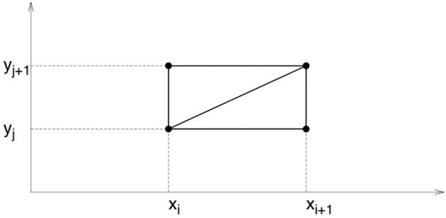
Function of 2 variables: Method 3

Consider a function $f(x, y)$ and construct its piecewise linear approximation.

- Divide the domain of f in a $(n-1) \times (m-1)$ grid of coordinates $x_1, \ldots, x_n$, $y_1, \ldots, y_m$.
- Divide the rectangles in the (x, y)-space in triangles .
- Sample f at each point (x_i, y_j) with $i = 1, \ldots, n$ and $j = 1, \ldots, m$.



Function of 2 variables: Method 3 (cont.d)



Any point $(\tilde{x}, \tilde{y})$

- belongs to one of the triangles;
- can be written as a convex combination of its vertices with weights α_{ij}; and
- the value of function f at $(\tilde{x}, \tilde{y})$ is approximated as

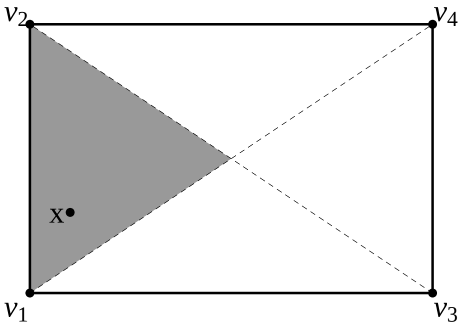
$$f^a = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{ij} f(x_i, y_j).$$

1 triangle $\leftrightarrow$ 1 binary variable $\rightarrow$ $O(n \times m)$ binaries.

Method 3: Standard Triangulation

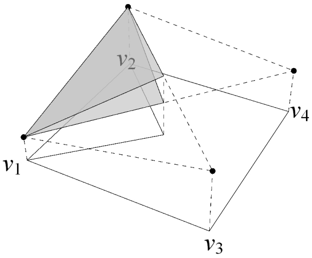
Given a rectangle identified by the four points v_1, v_2, v_3, v_4 we can divide it in 2 triangles in 2 different ways by selecting:

1. diagonal $[v_1, v_4]$; or
2. diagonal $[v_2, v_3]$.



Non-linear $f(x, y) \rightarrow$ 2 different f^a for choice 1 and 2 !

Method 3: Standard Triangulation



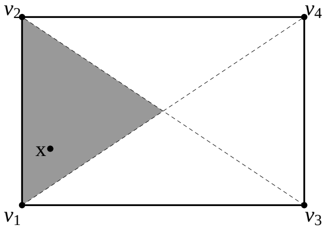
Diagonal $[v_1, v_4]$:

$$\alpha_{v_1} \leq \beta_{[v_1,v_2,v_4]} + \beta_{[v_1,v_3,v_4]}$$
$$\alpha_{v_2} \leq \beta_{[v_1,v_2,v_4]}$$
$$\alpha_{v_3} \leq \beta_{[v_1,v_3,v_4]}$$
$$\alpha_{v_4} \leq \beta_{[v_1,v_2,v_4]} + \beta_{[v_1,v_3,v_4]}$$
$$\beta_{[v_1,v_2,v_4]} + \beta_{[v_1,v_3,v_4]} = 1$$

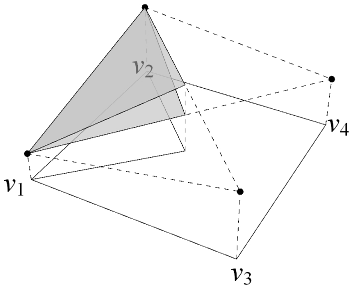
Diagonal $[v_2, v_3]$:

$$\alpha_{v_1} \leq \beta_{[v_1,v_2,v_3]}$$
$$\alpha_{v_2} \leq \beta_{[v_1,v_2,v_3]} + \beta_{[v_2,v_3,v_4]}$$
$$\alpha_{v_3} \leq \beta_{[v_1,v_2,v_3]} + \beta_{[v_2,v_3,v_4]}$$
$$\alpha_{v_4} \leq \beta_{[v_2,v_3,v_4]}$$
$$\beta_{[v_1,v_2,v_3]} + \beta_{[v_2,v_3,v_4]} = 1$$

Method 4: Optimistic Approximation



(c)



(d)

Observation is simple:

> *Why do we need to decide the triangle "offline"?*

Let the point $(\tilde{x}, \tilde{y})$ be a convex combination of all the 4 vertices of the rectangle and the MILP solver (optimistically) decide based on the objective function!

Method 4: Optimistic Approximation (cont.d)

Let the MILP (optimistically) decide based on the objective function!

In each region:

$$\check{f}(x) = \min \sum_{j=1}^{\nu} \alpha_j f(v_j) \quad \text{or} \quad \hat{f}(x) = \max \sum_{j=1}^{\nu} \alpha_j f(v_j)$$

subject to

$$\begin{aligned} \alpha_j &\geq 0 \\ \sum_{j=1}^{\nu} \alpha_j &= 1 \\ \sum_{j=1}^{\nu} \alpha_j x(v_j) &= x \\ \sum_{j=1}^{\nu} \alpha_j y(v_j) &= y \end{aligned}$$

where ν is the number of vertices that characterize the region.

Method 4: Optimistic Approximation Properties

Theorem

The approximations $\check{f}$ and $\hat{f}$ are such that

- $\check{f}$ *(resp. $\hat{f}$) is piecewise convex (resp. concave).*
- $\check{f}$ *and $\hat{f}$ are continuous.*
- *if f is linear then $\check{f} = \hat{f} = f$.*

Method 4: Optimistic Approximation Properties

Theorem

The approximations $\check{f}$ and $\hat{f}$ are such that

- $\Delta_r\left(f, \check{f}\right) \leq D_{\max}(r)$ *and* $\Delta_r\left(f, \hat{f}\right) \leq D_{\max}(r)$ $(\forall r \in \mathcal{R})$.
- *if f is convex (resp. concave) in any $r \in \mathcal{R}$, then $\check{f}$ (resp. $\hat{f}$) is the* ***best possible linear interpolation*** *of the samples $f(v_j)$ in the sense of $\Delta_r(f, \cdot)$.*

where

$\mathcal{R}$ is the collection of rectangles,

$\Delta_r(f, g) = \max_{(x,y)\in r} |f(x,y) - g(x,y)|$, and

$D_{\max}(r)$ is the maximum $\Delta_r\left(f, \tilde{f}\right)$ among all the possible linear interpolations $\tilde{f}$.

Standard vs Optimistic Approach: MILP size

Besides the nice properties, the optimistic approximation provides huge advantages when modeled with a MILP.

- Standard triangulation: 1 binary variable for each triangle $O(n \times m)$.
- Optimistic approximation: 1 binary variable for each rectangle.
- Note: Each axis treated separately, i.e.,
  n binaries for the x axis, and
  m binaries for the y axis. $\rightarrow O(n + m)$.
- For example, 3×3 grid $\rightarrow$ 6 vs 18 binaries
  10×10 grid $\rightarrow$ 20 vs 200 binaries!

Hydro UC: $f^a = f(x, y)$: MILP size

n	m	optimistic approximation # var.s all	optimistic approximation # var.s binary	optimistic approximation # con.s	optimistic approximation # nzs	standard approximation # var.s all	standard approximation # var.s binary	standard approximation # con.s	standard approximation # nzs
9	9	17,471	3,192	5,208	107,515	41,999	27,720	15,624	185,803
17	17	55,103	5,880	7,896	360,187	146,831	97,608	50,568	666,955
33	33	194,879	11,256	13,272	1,317,115	550,031	366,408	184,968	2,532,427
65	65	732,479	**22,008**	**24,024**	5,037,307	2,130,575	1,420,104	711,816	9,876,043

For $n = m =$65:

- Number of binary variables: 22,008 vs 1,420,104.
- Number of constraints: 24,024 vs 711,816.

Hydro UC: $f^a = f(x, y)$: Solving the MILP

Single processor of an Intel Core2 CPU 6600, 2.40 GHz, 1.94 GB of RAM under Linux.

Cplex 10.0.1.

Time limit of 1 hour.

		optimistic approximation				standard approximation				
		solution	%	CPU	#	solution	%	final	CPU	#
n	m	value	error	time	nodes	value	error	%gap	time	nodes
9	9	31,565.40	-2.34	14.71	1,507	31,565.40	-2.34	—	169.30	9,837
17	17	31,577.20	-2.31	755.96	36,507	31,577.20	-2.31	0.19	3,600.00	73,401
33	33	31,626.20	-2.35	277.13	2,567	n/a	n/a	n/a	3,600.00	5,500
65	65	31,640.30	-2.33	2,003.18	2,088	n/a	n/a	n/a	failure	failure

- Number of solved instances: 4 vs 2.

Hydro UC: $f^a = f(x, y)$: Going Logarithmic

Vielma & Nemhauser, 2011 : MILP model for the standard triangulations with a logarithmic number of variables (binary tree structure).

Doable also for the Optimistic approximation.

		optimistic approximation				logarithmic standard approximation			
		# var.s		# con.s	# nzs	# var.s		# con.s	# nzs
n	m	all	binary			all	binary		
9	9	17,471	3,192	5,208	107,515	16,127	1,848	4,368	142,963
17	17	55,103	5,880	7,896	360,187	51,407	2,184	5,040	578,419
33	33	194,879	11,256	13,272	1,317,115	186,143	2,520	5,712	2,501,683
65	65	732,479	22,008	24,024	**5,037,307**	713,327	2,856	6,384	11,056,243

		optimistic approximation				logarithmic standard approximation			
n	m	solution value	% error	CPU time	# nodes	solution value	% error	CPU time	# nodes
9	9	31,565.40	-2.34	14.71	1,507	31,538.70	-2.26	18.69	1,723
17	17	31,577.20	-2.31	755.96	36,507	31,577.20	-2.31	20.84	369
33	33	31,626.20	-2.35	277.13	2,567	31,624.10	-2.35	231.99	1,531
65	65	31,640.30	-2.33	2,003.18	2,088	31,640.30	-2.34	530.56	435

Hydro UC: $f^a = f(x, y)$: Going Logarithmic (cont.d)

n	m	logarithmic optimistic approximation # var.s	all binary	# con.s	# nzs	logarithmic standard approximation # var.s	all binary	# con.s	# nzs
9	9	16,127	1,848	4,032	135,907	16,127	1,848	4,368	142,963
17	17	51,407	2,184	4,704	553,891	51,407	2,184	5,040	578,419
33	33	186,143	2,520	5,376	2,409,955	186,143	2,520	5,712	2,501,683
65	65	713,327	2,856	6,048	10,701,091	713,327	2,856	6,384	11,056,243

		log optimistic approximation					log standard approximation				
n	m	solution value	% error	initial %gap	CPU time	# nodes	solution value	% error	initial %gap	CPU time	# nodes
9	9	31,565.40	-2.34	1.13	17.87	1,734	31,538.70	-2.26	1.14	18.69	1,723
17	17	31,577.20	-2.31	1.35	21.08	450	31,577.20	-2.31	1.35	20.84	369
33	33	31,626.20	-2.35	1.24	263.88	2,195	31,624.10	-2.35	1.25	231.99	1,531
65	65	31,640.30	-2.33	1.20	664.15	796	31,640.30	-2.34	1.20	530.56	435

Why? $\log(nm) = \log(n) + \log(m)$

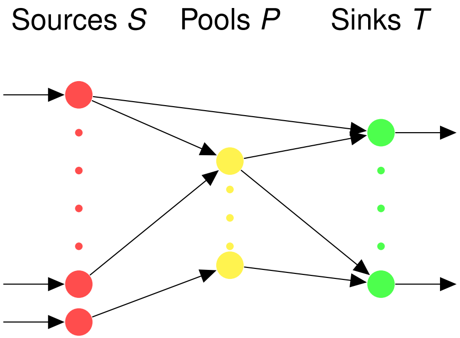
Advantages of the optimistic approximation: MILP model of limited size (tractable) and easy to implement .

Hydro UC: Summarizing...

- Several methods for approximating MINLPs through piecewise linear approximation
- From univariate to general functions
- Trade-off between tractability and approximation quality
- Best choice depends on the problem at hand
- What if we know the characteristics of the non-linear functions?

The Pooling Problem

The Pooling Problem



- Nodes $N = S \cup P \cup T$
- Arcs A $(i,j) \in (S \times P) \cup (P \times T) \cup (S \times T)$ on which materials flow
- Material attributes: K
- Arc capacities: u_{ij}
- Pool capacities: b_i
- Quality requirements $\beta_{kt} \forall k \in K, t \in T$

Quality Blending

- Product quality : weighted average of the quality of its inputs
- y_{ki}: Quality of attribute k at node $i \in N$

$$y_{ki} = \lambda_{ki} \quad \forall i \in S$$
$$y_{ki} = \frac{\sum_{j \in \delta^+(i)} y_{kj} x_{ji}}{\sum_{j \in \delta^+(i)} x_{ji}} \quad \forall i \in N \setminus S$$

- Upper bound on product quality

$$\sum_{i \in \delta^+(t)} y_{ki} x_{it} \leq \beta_{kt} \sum_{i \in \delta^+(t)} x_{it} \quad \forall k \in K, \forall t \in T$$

Bilinear inequalities!

Different Pooling Formulations

- P-formulation : Variables x_{ij} for flows on arcs
- Q-formulation (Ben-Tal et al. 94): Variables for proportion of flow coming from source:

$$q_{si} = \frac{x_{si}}{\sum_{t \in \delta^-(i)} x_{it}}$$

$(x_{si} = q_{si} \sum_{t \in \delta^-(i)} x_{it})$

- PQ-formulation : stronger! (Sahinidis and Tawarmalani (2005)). RLT technique to Q formulation.

$$\sum_{s \in S} q_{si} x_{it} = x_{it} \quad \forall i \in P, \forall t \in T$$

$$\sum_{t \in T} q_{si} x_{it} \leq q_{si} u_i \quad \forall s \in S, \forall i \in P$$

Relaxation

- Some notation:

$$H = \{(s \in S, i \in P, t \in T) : (s,i) \in A, (i,t) \in A\}$$
$$A_1 = \{(s \in S, i \in P) : (s,i) \in A\}$$

- Reformulate bilinear terms $q_{si}x_{it}$ in the "standard" way introducing auxiliary variables $w_{sit} = q_{si}x_{it}\ \forall (s,i,t) \in H$
- Relax nonconvex equality using McCormick relaxation . Additional constraints $\forall (s,i,t) \in H$

$$w_{sit} \leq \min(b_t, u_{it})q_{si}$$
$$w_{sit} \leq x_{it}$$
$$w_{sit} \geq 0$$
$$w_{sit} \geq \min(b_t, u_{it})q_{si} + x_{it} - \min(b_t, u_{it})$$

xy when x, y continuous

- Get bilinear term xy where $x \in [x^L, x^U]$, $y \in [y^L, y^U]$
- We can construct a **relaxation**:
  - Replace each term xy by an added variable w
  - Adjoin following constraints:

$$
\begin{aligned}
w &\geq x^L y + y^L x - x^L y^L \\
w &\geq x^U y + y^U x - x^U y^U \\
w &\leq x^U y + y^L x - x^U y^L \\
w &\leq x^L y + y^U x - x^L y^U
\end{aligned}
$$

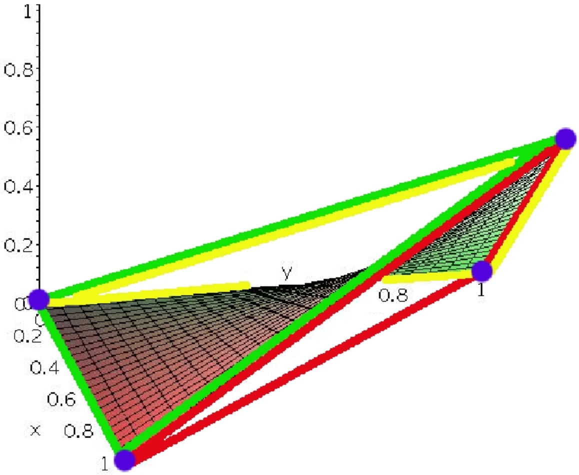
- These are called **McCormick's envelopes**
- Get an LP relaxation (solvable in polynomial time)

xy when x is binary

- If $\exists$ bilinear term xy where $x \in \{0, 1\}$, $y \in [0, 1]$
- We can construct an **exact reformulation**:
  - Replace each term xy by an added variable w
  - Adjoin Fortet's reformulation constraints:

$$\begin{aligned} w &\geq 0 \\ w &\geq x + y - 1 \\ w &\leq x \\ w &\leq y \end{aligned}$$

  - Get a MILP reformulation
  - Solve reformulation using CPLEX: more effective than solving MINLP


1
0.8
0.6
0.4
0.2
y
0.8
1
0.2
0.4
0.6
x
0.8
1

Summarizing...

Linearizing non-linear function might be a way. Two possibilities:

- Approximation: no guarantee, several possibilities
- Relaxation: guaranteee a bound, exploit characteristics of the non-linear function

For bilinear terms:

- If binary variable: Fortet refomulation (exact)
- If continuous variables: Mc Cormick relaxation

Important: formulation strengthening (RLT: reformulation-linearization technique, cuts, etc).