Solving Rigorously Equations and Optimization Problems

Yahia LEBBAH[1,2]

(Joint work with Claude MICHEL[1] **and Michel RUEHER**[1]**)**

[1] **Coprin (INRIA/Univ-Nice/I3S/CERTIS), France**

[2] **CS Dept. University of Oran Es-Senia, Algeria**

Outline

- Motivations
- Branch & Reduce to solve nonlinear equations
  - Reduction
  - Proving feasibility
  - Branching
- Branch & Bound for optimization
  - Reduction
  - Lower bounding
  - Upper bounding
  - Proving feasibility
  - Branching
- Experimental results
- Discussion

Motivations : the problem

We consider the general constrained optimization (no objective→ equations) problem

$$\mathcal{P} \equiv \begin{cases} \textbf{minimize} & h(x) \\ \textbf{subject to} & c_i(x) = 0, \;\; i \in \mathcal{E} \\ & c_i(x) \geq 0, \;\; i \in \mathcal{I} \\ & \underline{x_i} \leq x_i \leq \overline{x_i}, x_i \in \mathcal{X} \end{cases}$$

Critical issue: Efficient and rigorous computation of the global minimum (equations→all solutions)

- Most successful systems (Baron, αBB, ...) use relaxations

  → complete but not rigorous !

- Rigorous systems (GlobSol, Numerica, ...) are complete/rigorous but not efficient !

- Challenge: to combine both approaches !

Motivation: system of equations

$$
\begin{aligned}
&x_1^2 + x_2^2 - 1 = 0\\
&x_1^2 - x_2 = 0\\
&x_1, x_2 \in [-10, 10]
\end{aligned}
$$

We take the initial vector

$$x^0 = < 0, 1 >,$$

Newton-Raphson converges to

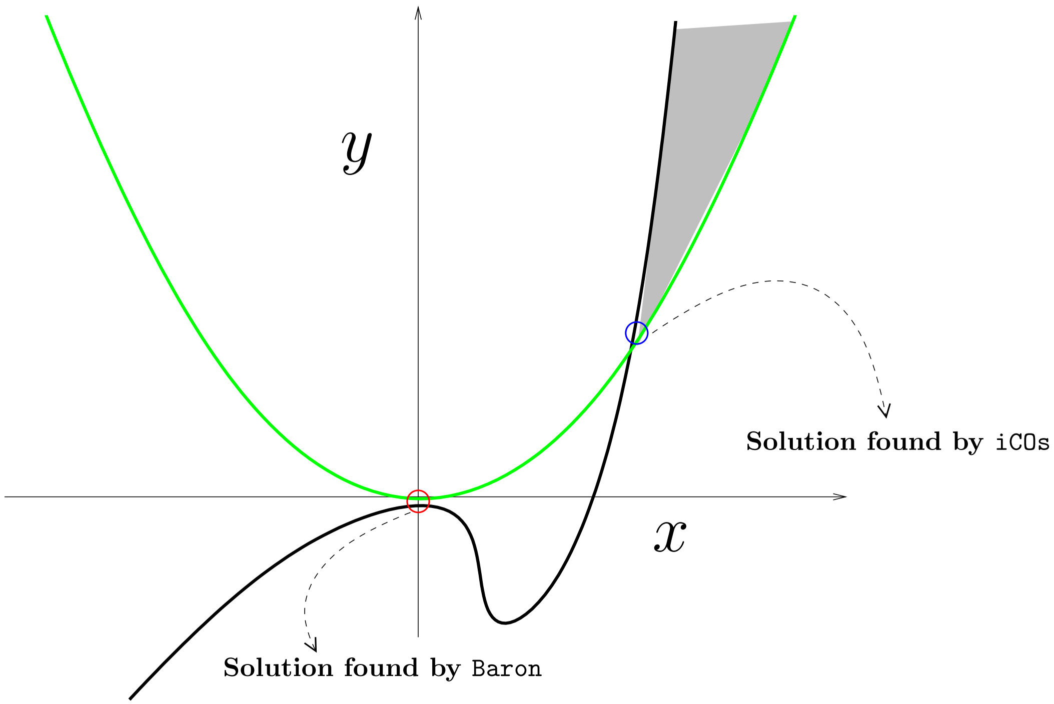
$$x^3 = < 0.78615, 0.61803 > .$$

But, is it the unique solution ? Is it *really* a solution ?
By calling a complete and rigorous solver, we get two solutions :

$x_1 = [$	$+0.786151377757422720194,$	$+0.786151377757423830417]$
$x_2 = [$	$+0.618033988749894125370,$	$+0.618033988749895457637]$
$x_1 = [$	$-0.786151377757423830417,$	$-0.786151377757422720194]$
$x_2 = [$	$+0.618033988749894125370,$	$+0.618033988749895457637]$

Motivation: optimization example

$$\begin{array}{ll} \textbf{minimize} & x \\ \textbf{subject to} & y - x^2 \geq 0; \quad -y + x^2(x-2) - 10^{-5} \geq 0 \\ & x, y \in [-10, +10] \end{array}$$

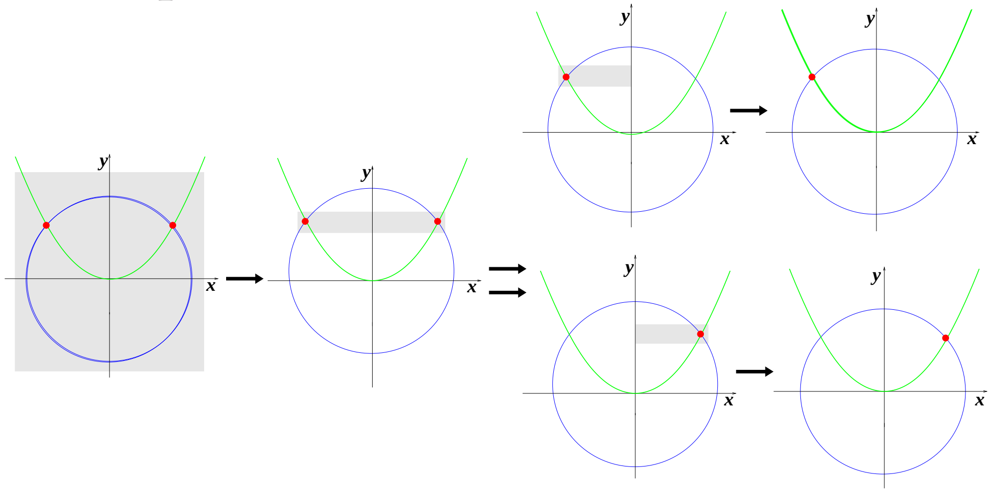


Branch and Reduce for solving nonlinear equations

Loop on [Van-Hentenryck 1997]

- **Select a box B from the set of current boxes $\mathcal{L}$**
- **Reduction (pruning, filtering or tightening) of B**
  - **Interval analysis: Interval Newton Algorithm**
  - **CP+Optimization: consistencies ($2B$, $3B$, Linearisation, ...)**
- **Splitting of $B \neq \emptyset$ and put the produced boxes in $\mathcal{L}$**

$$\left\{\begin{array}{l} x_1^2 + x_2^2 - 1 = 0 \\ x_1^2 - x_2 = 0 \\ x_1, x_2 \in [-10^8, 10^8] \end{array}\right)$$



Reduction: Interval Newton Algorithm

Let be

$$f_j(x_1, ..., x_n) = 0, j \in 1..n$$

Compute the Taylor form:

$$f_j(c) + \sum_{i=1}^{n} nat(\frac{\partial f_j}{\partial x_i})(\mathcal{D}) * (X_i - c_i) = 0, j \in 1..n$$

where $\mathcal{D} =< [...], [...], ..., [...] >$**,** c **is the center of the box** $\mathcal{D}$**, and** $nat(\frac{\partial f_j}{\partial x_i})(\mathcal{D})$ **is the interval evaluation of the Jacobian with the natural interval extension.**

We obtain an interval linear system

$$A(X - c) = b$$

Preconditionning: $PA(X - c) = Pb$**.**
Usually $P = m(A)^{-1}$**.**
We have the final interval linear system $A'X = c'$**.**

Reduction: Interval Newton Algorithm ...(proof)

Apply interval Gauss-Seidel (Gauss, Krawczyck, ...) on

$$A'X = c'$$

Proposition 1 (Miranda, ...) ***Let be*** $< \mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^n, ... >$ ***the sequence of the boxes generated by the Gauss-Seidel iterations, then if***

$$\exists k, \mathcal{D}^k \subseteq \mathcal{D}^{k-1}$$

then there is a solution. (unique solution if $\subset$***).***

Other similar results from existence theory: Brouwer, Krawczyck, ...
Reference: Hansen 2004, Kearfott 1996, Neumaier 1990, ...

Reduction: Interval Newton Algorithm (example)

Let be the system [Kearfott 1996]

$$
\begin{array}{l}
x_1^2 - 4x_2 = 0 \\
x_2^2 - 2x_1 + 4x_2 \\
x_1 \in [-0.1, 0.1], x_2 \in [-0.1, 0.3]
\end{array}
$$

We have $nat(\frac{\partial f_j}{\partial x_i})(\mathcal{D}) = \begin{pmatrix} [-0.2, 0.2] & -4 \\ -2 & [3.8, 4.6] \end{pmatrix}$**.**

$$P = m(A)^{-1} = \begin{pmatrix} -0.525 & -0.5 \\ -0.25 & 0 \end{pmatrix}$$

At the first iteration of Gauss-Seidel, we have

$$\mathcal{D}^1 = \begin{pmatrix} \textbf{[-0.0455, 0.0355]} \\ \textbf{[-0.005,0.005]} \end{pmatrix} \subset \mathcal{D}$$

There is a unique solution in the produced box.

Reduction: Constraint Programming

- **Consistency principle: Every constraint is consistent with every variable bounds ($2B$-consistency)**
- **Implementation of consistency techniques: Extract relations from the constraints that expresses every variable with other variables ($2B$-filtering)**

$$\mathcal{C} \Longrightarrow \{x_i = f_j(x_1, x_2, ..., x_n) | i \in 1..n\}$$

Example:

$$x + y - 1 = 0; \quad y - rz = 0$$
$$x \in [4, 5]; y \in [3, 100]; z \in [1, 2]; r \in [2, 2]$$

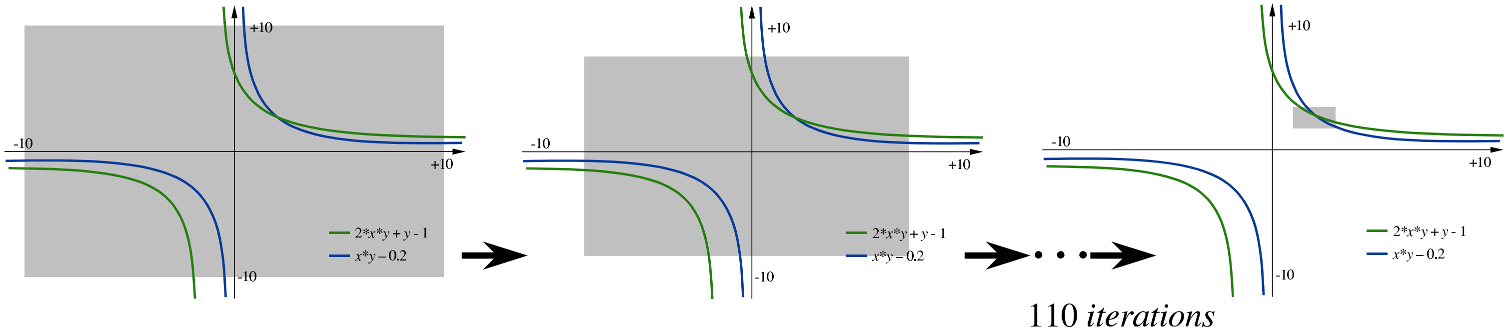
We can generate the following relations

$$x = 1 - y; \quad y = 1 - x; \quad y = rz; \quad z = y/r; \quad r = y/z$$

Interval evaluation of these relations yields

$$x \in [3, 5]; \quad y \in [1.5, 2]; \quad z \in [-3, -2]; ...$$

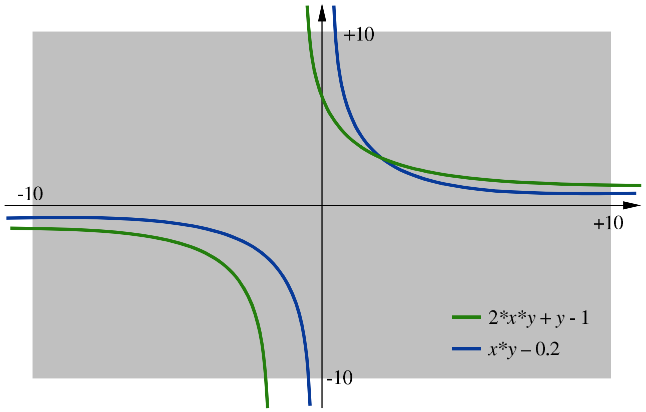
Reduction: Limitation ...

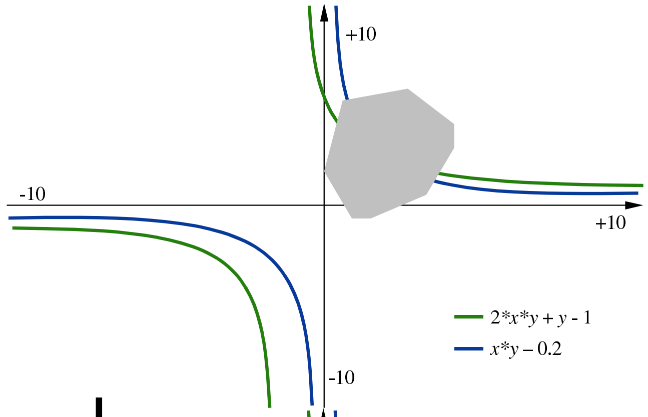


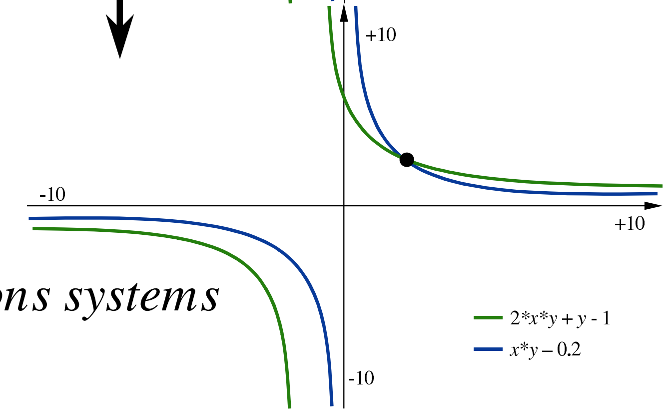
- *Standard reduction techniques are interval based*

$$\begin{pmatrix} 2xy + y = 1 \\ xy = 0.2 \\ x, y \in [-10, 10] \end{pmatrix}$$

Reduction: Linearisation [Lebbah 2005]







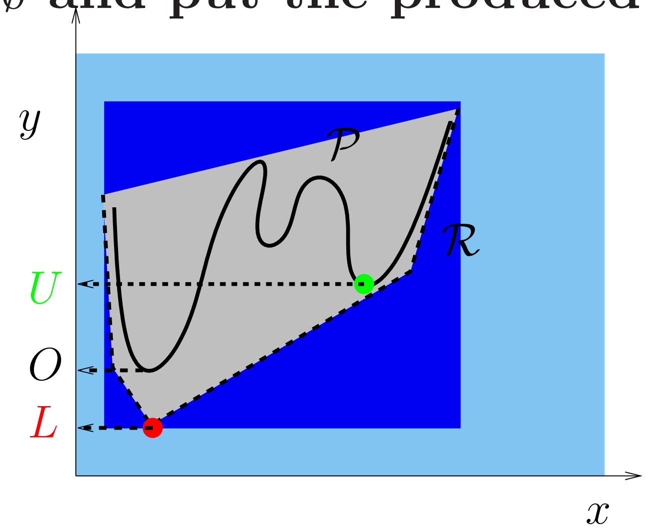
$$\begin{pmatrix} 2xy + y = 1 \\ xy = 0.2 \\ x, y \in [-10, 10] \end{pmatrix}$$

- *Quad : 3 generated linear inequations systems*
  - *12 simplex callings*
  - *Exact solution*
- *Note that Quad-filtering is convex-set based*

Branch & Bound schema for optimization

Loop on [Sahinidis 2002, Floudas 2000, Hansen 2004]

- **Select a box** B **from the set of current boxes** $\mathcal{L}$
- **Reduction (pruning, filtering or tightening) of** B
- **Lower bounding of** f **in the box** B
- **Upper bounding of** f **in the box** B
- **Splitting of** $B \neq \emptyset$ **and put the produced boxes in** $\mathcal{L}$



Reduction techniques

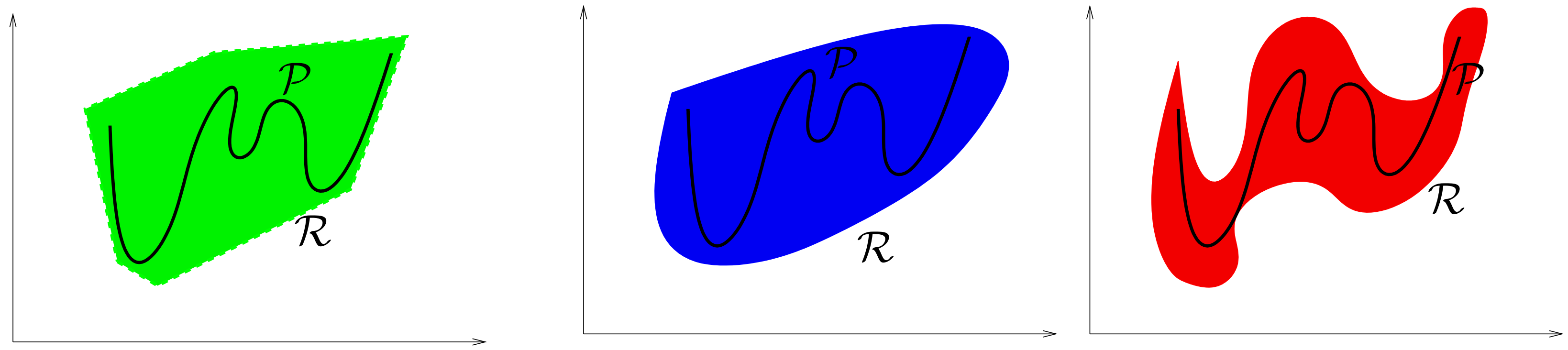
Two main classes [Sahinidis et al]:

- **Feasibility based reduction**
  $\Longrightarrow$ **discards subdomains where the constraints are not satisfied**
  $\Longrightarrow$ **optimization, constraint programming, interval analysis**
- **Optimality based reduction**
  $\Longrightarrow$ **discards subdomains where the objective function takes values above the current upper bound**
  $\Longrightarrow$ **optimization**
  $\Longrightarrow$ **A very successful technique in Baron !**

Lower bounding: Relaxation

$$\mathcal{R} \equiv \begin{cases} \textbf{minimize} & f(x) \\ \textbf{subject to} & g_i(x) = 0, \;\; i = 1..k \\ & g_i(x) \geq 0, \;\; i = 1..m \\ & \underline{x_i} \leq x_i \leq \overline{x_i}, i = 1..n. \end{cases}$$

The objective value at the global minimum of $\mathcal{R}$ is lower than the objective value of the global minimum of $\mathcal{P}$.



Lower bounding: Linear relaxation

- **Construction of the relaxation with RLT (Reformulation Linearisation Techniques) applied to** $\mathcal{C} = h(X) \cup c_{i,i\in\mathcal{E}\cup\mathcal{I}}(X)$
  - **Let be** $\mathcal{I}$ **(e.g.,** x^2**,** x^n**,** xy**,** $x_1x_2...x_n$**,** $exp(x)$**,** $log(x)$**, ...) the set of nonlinear terms occurring in** $\mathcal{C}$**.**
  - **Reformulation: adding new (redundant) constraints** $\mathcal{R}$ **to take into account the specific semantics of the nonlinear terms** I
  - **Linearisation: generating a set of linear inequalities** LR **for the whole augmented system** $\mathcal{C} \cup \mathcal{R}$ **by replacing nonlinear terms by new variables.**
- **Rigorous call to the Simplex algorithm on**

$$\mathcal{R} \equiv \begin{cases} \textbf{minimize} & [f(x)]_l \\ \textbf{subject to} & LR \\ & \underline{x_i} \le x_i \le \overline{x_i}, i = 1..n. \end{cases}$$

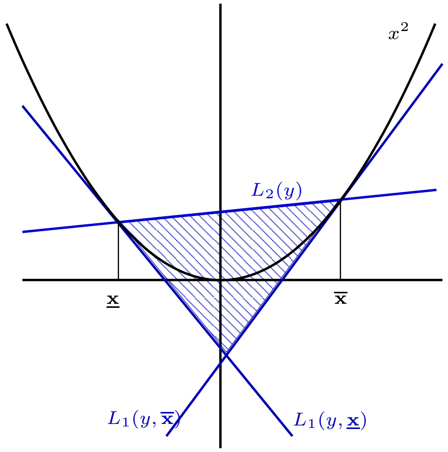
Lower bounding: linearisation of x^2

- $f(x) = x^2$ **with** $\underline{x} \leq x \leq \overline{x}$ **is approximated by :**

$$L_1(y, \alpha) \equiv [(x-\alpha)^2 \geq 0]_l \textbf{ where } \alpha \in [\underline{x}, \overline{x}] \tag{1}$$

$$L_2(y) \equiv (\underline{x} + \overline{x})x - y - \underline{x} * \overline{x} \geq 0 \tag{2}$$

- **Example: relaxation of** $y = x^2$ **with** $x \in [-4, 5]$
  - $L_1(y, -4) : y \geq -8x - 16$**;** $L_1(y, 5) : y \geq 10x - 25$**;** $L_2(y) : y \leq x + 20$**;**

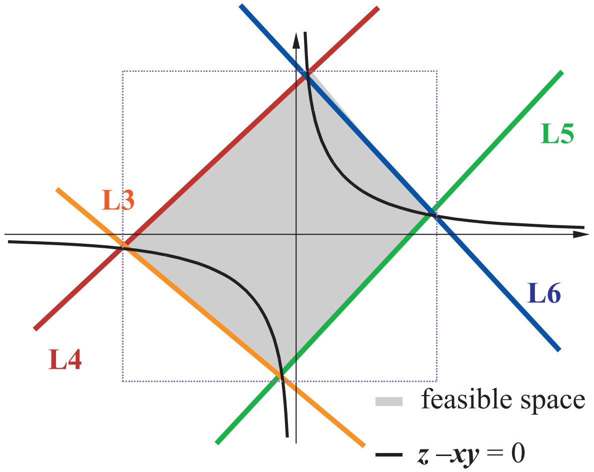


Lower bounding: linearisation of xy

Let be $z = x * y$**.**

$$L3(z) \equiv [(x - \underline{x_i})(y - \underline{x_j}) \geq 0]_l; L4(z) \equiv [(x - \underline{x_i})(\overline{x_j} - y) \geq 0]_l;$$
$$L5(z) \equiv [(\overline{x_i} - x)(y - \underline{x_j}) \geq 0]_l; L6(z) \equiv [(\overline{x_i} - x)(\overline{x_j} - y) \geq 0]_l$$

Example: $z = xy, x \in [-5, +5], y \in [-5, +5]$



Lower bounding: other cases

- **Monomial term $T \equiv x_1 x_2 \dots x_n$ of degree n: quadrification.**
  - **Apply the middle heuristic:**

$$T = T_1 T_2,$$

$$T_1 = x_1 x_2 \dots x_{n \div 2} \text{ and } T_2 = x_{n \div 2 + 1} \dots x_n$$

  - **We get a set of quadratic equations containing only the bilinear term $x_i x_j$ and the convex term x_i^2**
  - **Apply RLT on these quadratic equations**
- **Convex term: the Sandwich algorithm**
- **Power term x^n: Sherali's linearisation with RLT**
- **Trilinear terms: Floudas's algorithm**
- **...**

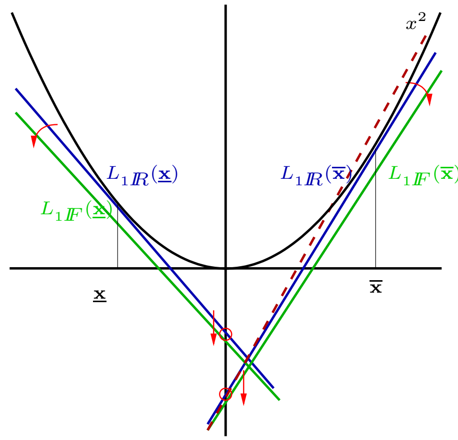
A general correction to n-ary linearisations

Let $\sum_{i=1}^n a_i x_i + b \geq 0$

then $\forall x_i \in \mathbf{x}_i.$ **:**

$$\sum_{i=1}^{n} \overline{a}_i x_i + \sup(\overline{b} + \sum_{i=1}^{n} \sup(\sup(\mathbf{a}_i \underline{x}_i) - \overline{a}_i \underline{x}_i)) \geq \sum_{i=1}^{n} a_i x_i + b \geq 0$$

Proof: SIAM Sinum 2005



Lower bounding: a rigorous procedure for the Simplex

Let be the LP:

$$\min \; c^T x \tag{3}$$
$$\textbf{s.t.} \quad \underline{b} \leq Ax \leq \overline{b} \tag{4}$$
$$\textbf{and} \quad \underline{x} \leq x \leq \overline{x} \tag{5}$$

THE problem : $c^T xr \geq c^T xf$ **?**
The dual is:

$$\max \underline{b}^T z' + \overline{b}^T z'' \tag{6}$$
$$\textbf{s.t.} \quad A^T(z' - z'') = c \tag{7}$$

Let bet $y = z' - z''$ **and the residue** $r = A^T y - c$**.**
The fundamental result [Neumaier 2004]:

$$c^T xr \in (y^T[\underline{b}, \overline{b}] - r^T[\underline{x}, \overline{x}])$$

Thus

$$c^T xr \geq \lfloor y^T[\underline{b}, \overline{b}] - r^T[\underline{x}, \overline{x}] \rfloor \tag{8}$$

Upper bounding

- **Use local optimization (SLP, SQP, IP, ...) to get an approximate feasible solution x'**
- **Initialize a box D to x'**
- **Inflation:**
  Loop on Inflating D: $D_i = [\underline{\mathbf{D}_i} - \epsilon, \overline{\mathbf{D}_i} + \epsilon], i \in 1..n$
  Until reaching some box D where the feasibility is proved.

Upper bounding: Feasibility proof

- **Let be an optimization problem with equalities.**
- **Take the Taylor form of the constraints within the current box**

$$h(c) + \sum_{i=1}^{n} nat(\frac{\partial h}{\partial x_i})(\mathcal{D}) * (X_i - c_i) = 0$$

or simply

$$A(X - c) = b$$

- **Fix, with an heuristic, the last remaining $l - n$ variables, and obtain a squared interval linear system**

$$A'(X' - c') = b'$$

- **Apply interval Gauss-Seidel. The feasibility is proved when one Gauss-Seidel iterate gets a box which is contained (not necessarily strictly) in the previous box.**

Implementation in iCOs Solver

- **General and rigorous solver: fully rigorous**
- **Techniques: CP techniques (week and strong propagation), Interval techniques (Interval Newton), Optimization (Branch and bound, linear relaxations), LP solvers interface (Cplex, Coin-Clp), AMPL local solvers interface (ipopt, minos, ...), ...**
- **AMPL language interface**

```
icos example.mod C=strategy.cfg
```

- **http://www-sop.inria.fr/coprin**

Experimentation for optimization

iCOs/PIII/1Ghz

With: Cplex for lower-bounding, IpOpt for upper-bounding, and 2B for reduction.

Name	(n, m)	ICOS	GlobSol
ex14_1_1	(3,4)	1.01	100.98
ex14_1_2	(6,9)	2.95	7200.56
ex14_2_5	(4,5)	2.53	107.74
ex9_2_5	(8,7)	0.22	7212.43
house	(8,8)	2.25	7204.24
mhw4d	(5,3)	1.01	4.81
audet140a	(5,4)	0.1	4.5
audet140b	(4,2)	0.07	0.17
audet141	(6,4)	1.09	2.52
audet145	(7,8)	0.66	48.57
audet147	(16,19)	0.30	∞
himmel16	(18,21)	1858.21	∞

Experimental results for solving equations

	Kin2			Didrit		
	P-choix	Nb Sols	T (s)	P-choix	Nb Sols	T (s)
Ilog solver	3485	10*	31.45	50739	4*	157.09
Box+Newton	3585	8	$> 300^{\dagger}$	8105	1	$> 300^{\dagger}$
Quad	11	10	69.30	24	4	185.44

	Lee			Dietmaier		
	P-choix	Nb Sols	T (s)	P-choix	Nb Sols	T (s)
Ilog solver	30904886	4*	266764.95	100294739	40*	3134026.38
Box+Newton	66369	0	$> 3600^{\dagger}$	22453	0	$> 3600^{\dagger}$
Quad	52	4	461.11	-	-	-

† time out

* solutions proved

Discussion

- Rigour/Safety: Interval techniques
- Efficiency: Linear relaxations + CP-techniques
- Solution-existence: Interval Newton

Systems: Baron, GlobSol, αBB, iCOs, ...
Books: Hansen 2004, Sahinidis 2002, Kearfott 1996, Neumaier 1990, ...
Articles: Van-Hantenryck 1997, Lebbah 2005, ...

Perpectives

- Integrating other optimization techniques in a rigorous way !
- Improve performances !
- Symbolic methods ...