Toward a rigorous and efficient global optimizer

Yahia LEBBAH[*,§], Claude MICHEL[§] and Michel RUEHER[§]

[*]**Computer Science Dept, Université d'Oran Es-Sénia, Algeria**

[§]**Projet CeP, I3S-CNRS, Université de Nice - Sophia Antipolis, France**

Outline

- Motivations
- Branch & Bound algorithm
- Lower bounding
- Upper bounding
- Filtering (reduction/pruning)
- Splitting strategy
- Experiments
- Conclusion and perspectives

Motivation: example of a non safe approach

$$\begin{array}{ll} \textbf{mininmize} & z = x_1 + x_2 \\ \textbf{subject to} & 2 - x_1^2 - x_2^2 \geq 0; x_2^2 \leq 1.1 \\ & x_1, x_2 \in [-100, +100]. \end{array} \tag{1}$$

The global minimum: $x_1^* = \sqrt{0.9}, x_1^* = \sqrt{1.1}, z^* = \sqrt{0.9} + \sqrt{1.1}$**.**
Local solvers Minos and IpOpt, and the global solver Baron find

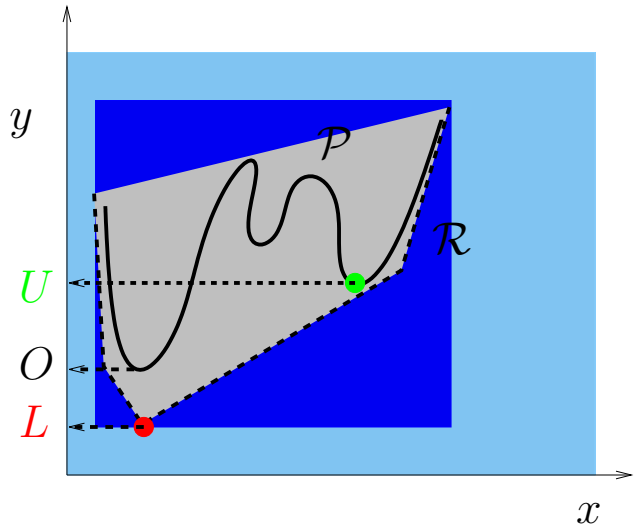
$$x_1' = -0.948683, x_2' = -1.04881, z' = -1.997492146$$

By using a rigorous approach:

$\mathbf{z}$	$=$	$[-1.99749214,$	$-1.99749211]$
$\mathbf{x}_1$	$=$	$[-0.948683299,$	$-0.948683298]$
$\mathbf{x}_2$	$=$	$[-1.048808849,$	$-1.048808848]$
...			

Branch & Bound schema

- **While $\mathcal{L} \neq \emptyset$ do** %$\mathcal{L}$ initialized with the input box
  - Select a box B from the set of current boxes $\mathcal{L}$
  - Reduction (filtering or tightening) of B
  - Lower bounding and Upper bounding of f in the box B
  - Splitting of B if it is not proved empty; put the new boxes in $\mathcal{L}$



Branch & Bound algorithm

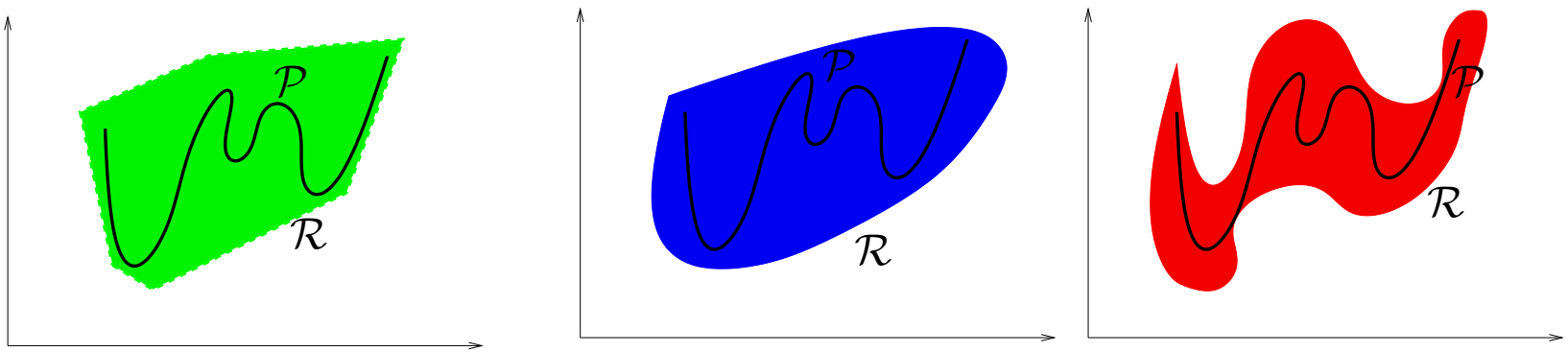
Function BB(IN $\mathbf{x}$, ϵ; OUT $\mathcal{S}$, $[L, U]$)

% $\mathcal{S}$: set of proved feasible points
% $\mathbf{f}_{\mathbf{x}}$ denotes the set of possible values for f in $\mathbf{x}$
$\mathcal{L} := \{\mathbf{x}\}$; $\mathcal{S} := \emptyset$; $(L, U) := (-\infty, +\infty)$;
while $w([L, U]) > \epsilon$ **do**
  $\mathbf{x}' := \mathbf{x}''$ **such that** $\underline{\mathbf{f}}_{\mathbf{x}''} = min\{\underline{\mathbf{f}}_{\mathbf{x}''} : \mathbf{x}'' \in \mathcal{L}\}$; $\mathcal{L} := \mathcal{L} \backslash \mathbf{x}'$;
  $\overline{\mathbf{f}}_{\mathbf{x}'} := min(\overline{\mathbf{f}}_{\mathbf{x}'}, U)$;
  $\mathbf{x}' := Prune(\mathbf{x}')$;
  $\underline{\mathbf{f}}_{\mathbf{x}'} := LowerBound(\mathbf{x}')$;
  $(\overline{\mathbf{f}}_{\mathbf{x}'}, \mathbf{x}_p, Proved) := UpperBox(\mathbf{x}')$;
  if $Proved$ **then** $\mathcal{S} := \mathcal{S} \cup \{\mathbf{x}_p\}$; **endif**
  if $\mathbf{x}' \neq \emptyset$ **then** $(\mathbf{x}'_1, \mathbf{x}'_2) := Split(\mathbf{x}')$; $\mathcal{L} := \mathcal{L} \cup \{\mathbf{x}'_1, \mathbf{x}'_2\}$;
  if $\mathcal{L} = \emptyset$ **then** $(L, U) := (+\infty, -\infty)$;
  else $(L, U) := (min\{\underline{\mathbf{f}}_{\mathbf{x}''} : \mathbf{x}'' \in \mathcal{L}\}, min\{\overline{\mathbf{f}}_{\mathbf{x}''} : \mathbf{x}'' \in \mathcal{S}\})$;
endwhile

Lower bounding: Relaxation of $\mathcal{P}$

$$\mathcal{R} \equiv \begin{cases} \textbf{minimize} & f(x) \\ \textbf{subject to} & g_i(x) = 0, \ i \in \mathcal{RE} \\ & g_i(x) \geq 0, \ i \in \mathcal{RI} \\ & \underline{x_i} \leq x_i \leq \overline{x_i}, x_i \in \mathcal{RX} \end{cases}$$

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

Embed

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

into a linear relaxation

$$\mathcal{LR} \equiv \begin{cases} \textbf{minimize} & cx \\ \textbf{subject to} & A_E x = b_E, \\ & A_I x \geq b_I, \\ & \underline{x_i} \leq x_i \leq \overline{x_i}, x_i \in \mathcal{RX} \end{cases}$$

Solving $\mathcal{LR}$ provides a rigorous lower bound of the objective function value at the optimum of $\mathcal{P}$, provided that $\mathcal{LR}$ is rigorously generated and solved. The precision of the lower bound depends on the tightness of $\mathcal{LR}$.

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, $x_1 x_2 ... x_n$, $exp(x)$, $log(x)$, **...) 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**

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

Lower bounding: monomial terms $x_1 x_2 ... x_n$

1. **Decomposition into a set of quadratic and monovariate terms**
2. **Relax every quadratic term with classical relaxations**
3. **Relax every convex term with the Sandwich algorithm**

Consider the term $x_1 x_2 x_3 x_4 \exp(x_5)$**:**

$$\overbrace{x_1 x_2 x_3 x_4 \exp(x_5)}^{y_1} = \overbrace{x_1 x_2 x_3}^{y_2} \times \overbrace{x_4 \exp(x_5)}^{y_3}$$

$$y_2 = \overbrace{x_1 x_2}^{y_4} \times \overbrace{x_3}^{x_3}$$

$$y_3 = \overbrace{x_4}^{x_4} \times \overbrace{\exp(x_5)}^{y_5}$$

$$y_4 = \overbrace{x_1}^{x_1} \times \overbrace{x_2}^{x_2}$$

$$y_5 = \exp(\overbrace{x_5}^{x_5})$$

$$...$$

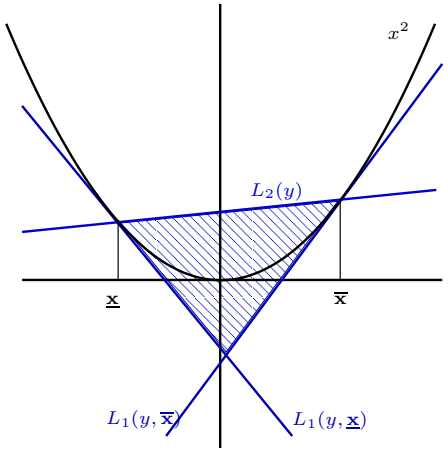
Lower bounding: power terms

A power term x^n can be approximated by $n+1$ inequalities [Sherali & al. 1997]:

$$[x^n]_R = \{[(x-\underline{x})^i(\overline{x}-x)^{n-i} \geq 0]_l, i = 0..n\}$$

1. **This relaxation generates a tight relations at lower and upper bounds.**
2. **It generalizes x^2 relaxation on the bounds to x^n.**

$$[x^2]_R = \{[(x-\underline{x})^2 \geq 0]_l, [(x-\underline{x})(\overline{x}-x) \geq 0]_l, [(\overline{x}-x)^2 \geq 0]_l\}$$



Lower bounding

- **Monomial term** $T \equiv x_1 x_2 \ldots x_n$ **of degree** n**: quadrification.**
  - **The middle heuristic:**

$$T = T_1 T_2, \ldots$$

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

  - **The Left hand side heuristic:**

$$T = T_1 T_2, \ldots$$

$$T_1 = x_1 \text{ and } T_2 = x_2 \ldots x_n$$

  - $4(n-1)$ **inequalities**
- **Convex terms: The sandwich algorithm [Rote]**
- **(Trilinear terms: Floudas's algorithm)**
- **(nonconvex terms: Sahinidis's polyhedral relaxations), ...**

Upper Bounding

1. **Call a local search method to find an approximate feasible point P_{loc}, possibly nearly optimal (COIN/IpOpt)**

$$\mathcal{P} \equiv \begin{cases} \textbf{minimize} & h(x) \\ \textbf{subject to} & c_i(x) = 0, \ i \in \mathcal{E} \\ & c_i(x) \geq 0, \ i \in \mathcal{I} \\ & x \in \mathcal{D} \\ & x^0 := P_{lb} \quad \%\textbf{where } P_{lb} \textbf{ is the underestimator} \end{cases}$$

2. **Prove the feasibility of a box $\mathcal{D}'$ containing P_{loc} on the constraints**

$$\begin{aligned} & g_i(x) = 0, i \in \{1, \ldots, k\} \\ & g_j(x) \leq 0, j \in \{k+1, \ldots, m\} \end{aligned} \tag{2}$$

Upper Bounding: Feasibility proof (ONGOING WORK)

Loop on

1. $\mathcal{D}' = [P_{loc} - \Delta, P_{loc} + \Delta]$
2. **Separation of the constraints into two sets: active set and inactive set**
   - **Inactive set: set of inequalities that are completely satisfied**
   - **Active set: set of equalities and inequalities that are approximately satisfied (inequalities are considered equalities)**
3. **We obtain the set of equalities** $g_i(x) = 0, i \in \mathcal{AE}$
4. **Prove the feasibility of active constraints**

Example: Let be $x, y \in [1, 10]$ **and the constraint** $x + y \geq 0$**. With IA, we have** $[1, 10] + [1, 10] = [2, 20] \geq 0]$**; the constraint is inactive !**

Let be the constraint $x - y \geq 0$**. With IA, we have** $[1, 10] - [1, 10] = [-9, 9]$**.**

Upper Bounding: proving the feasibility of active constraints (ON-GOING WORK)

1. The Taylor form of the equations is then used to generate the linear system
$$h(c) + \sum_{i=1}^{n} nat(\frac{\partial h}{\partial x_i})(\mathcal{D}')(X_i - c_i) = 0$$
which is equivalent to the $l \times n$ linear interval system
$$A(X - c) = b \tag{3}$$
2. Total pivoting is applied on (3).
3. The last remaining $l - n$ variables are then fixed to their midpoints in order to obtain the squared interval linear system
$$A'(X' - c') = b' \tag{4}$$
4. An existence test is then applied (Krawczyck, Miranda, Borsuk).

Filtering/Reduction techniques

Two main classes:

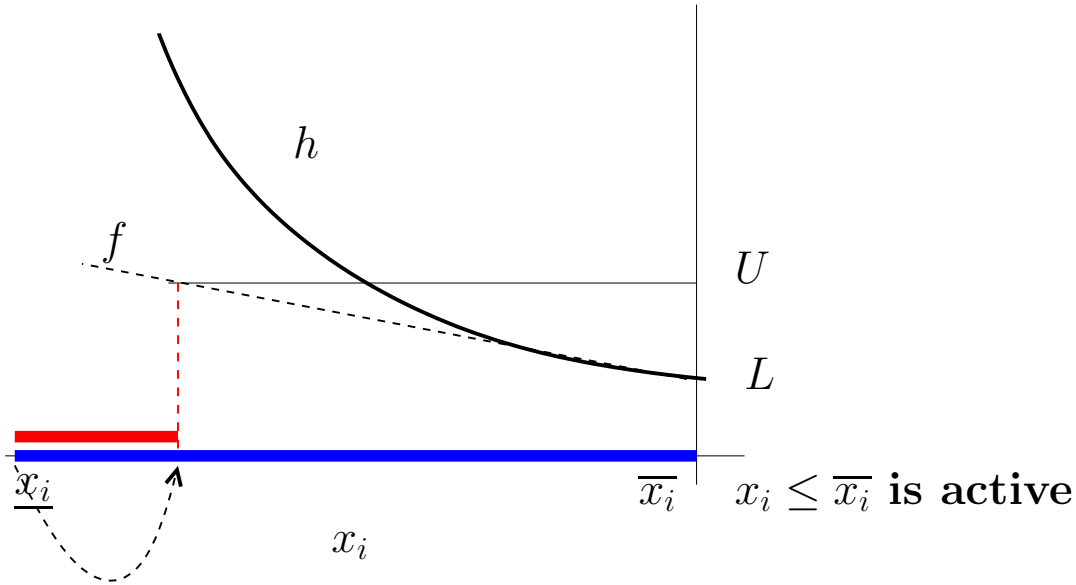
- **Feasibility based reduction**
  $\Longrightarrow$ discards subdomains where the constraints are not satisfied
  $\Longrightarrow$ optimization, constraint satisfaction (solving equations)
- **Optimality based reduction**
  $\Longrightarrow$ discards subdomains where the objective function takes values above the current upper bound
  $\Longrightarrow$ optimization

Filtering: Optimality based reduction

Optimality techniques principle

- **Consider $[L, U]$ the current bounds of the objective function**
- **Variable x_i is consistent with $[L, U]$ if**

$$f(..., \underline{x_i}, ...) \in [L, U]$$
$$f(..., \overline{x_i}, ...) \in [L, U]$$



Filtering: Optimality based reduction

Optimality techniques implementation [Sahinidis et al.]: Let be λ and γ the dual multipliers at the solution of the relaxation

Active Constraint	*Requirement*	*Valid Inequality*
$x_i \leq \overline{\mathbf{x}_i}$	$\lambda_i > 0$	$x_i \geq \overline{\mathbf{x}_i} - (U - L)/\lambda_i$
$x_i \geq \underline{\mathbf{x}_i}$	$\lambda_i > 0$	$x_i \leq \underline{\mathbf{x}_i} + (U - L)/\lambda_i$
$a_j^t x \geq b_i$	$\gamma_j > 0$	$a_j^t x \geq b_j - (U - L)/\gamma_j$
$g_j(x) \leq 0$	$\gamma_j > 0$	$g_j(x) \geq -(U - L)/\gamma_j$

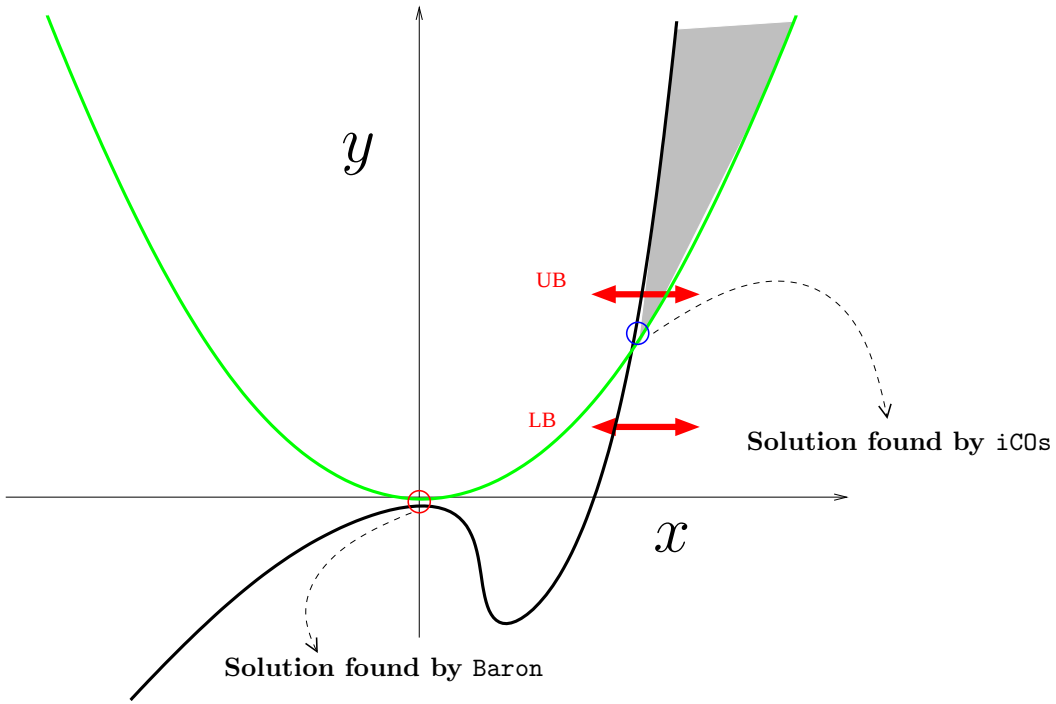
Critical issue: safe computation of multipliers $\lambda_1, \lambda_2, ..., \lambda_n, \gamma_1, ..., \gamma_m$.

Two approaches

- **Kearfott: solving KT system with interval Newton**
- **Simple approach: no proof; use OBR bounds as branching points !**

Branching strategy (ONGOING WORK)

Choosing l variables having the largest gap between the underestimator (lower bound) and the feasible set, and then Multisplit on these l variable.



Branching strategy (ONGOING WORK)

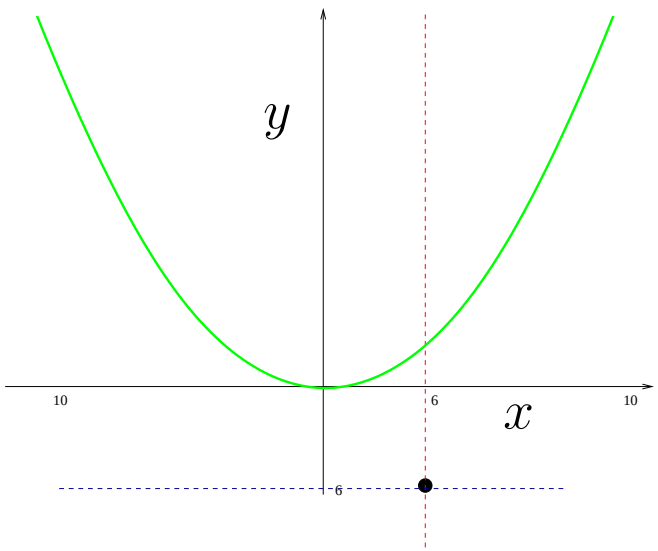
Interval methods: Largest width, Hansen rule, Ratz rule, ...

- **Let be P_{lb} the current lower bound;**
- **Let be B a box initialized with P_{lb};**
- **Inflate B by propagation until reaching feasibility;**
- **Choose the l variables mostly inflated;**
- **Multisplit on these l variables.**

Branching strategy: illustration (ONGOING WORK)

$$\begin{aligned} \textbf{mininmize} \quad & z = x \\ \textbf{subject to} \quad & y = x^2 \\ & x, y \in [0, 10]. \end{aligned} \tag{5}$$

Let be the infeasible point $(x_0, y_0) = (6, -6)$**.**



Branching strategy: illustration (ONGOING WORK)

$$\begin{array}{ll} \textbf{mininmize} & z = x \\ \textbf{subject to} & y = x^2 \\ & x, y \in [0, 10] \\ & (x_0, y_0) = (6, -6) \end{array}$$

By propagation on the constraints, we get:

$$y = x_0^2 = [6, 6]^2 \cup [-6, -6] = [-6, 36]$$

$$x = \sqrt{y_0} = \sqrt{[-6, -6]} \cup [6, 6] = \emptyset \cup [6, 6] = [6, 6]$$

These intervals extend the domain to the feasible region.

`iCOs` Solver

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

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

- **http://www.essi.fr/~lebbah/icos/**

Experimental results

The strategy:

```
optimize(1.0e-3,
   b2box(1e-8, pinat),
   quadopt(quad_mid quad_xn, 1000),
   ipopt,
   large_infeas
   borsuk)
```

1. **BB depends on upper-bounding and local search with feasibility (Borsuk seems a good existence test);**
2. **BB depends on lower-bounding and the tightness of the relaxation;**

Tuning is required to enhance solving some problems !

Conclusion

Chemical problem coming from [Hansen & Jaumard, Audet, ...]

```
var x1 >= 1, <= 8.03773157; var x2 >= 4.5, <= 9.71853961;
var x3 >= 9, <= 10; var x4 >= 0.001, <= 10;
var x5 >= 1, <= 4.68381588; var x6 >= 1, <= 9;
var x7 >= 0.01, <= 100; var x8 >= 0.01, <= 54.91813487;
var x9 >= 50, <= 100; var x10 >= 0.01, <= 50;
minimize obj  <= 6.64865039370471322400e+01 :
  12.62626*(x8 + x9 + x10) - 1.231059*(x1*x8 + x2*x9 + x3*x10);
subject to
c1a: 12.62626*(x8 + x9 + x10) - 1.231059*(x1*x8 + x2*x9 + x3*x10) >= 50;
c1b: 12.62626*(x8 + x9 + x10) - 1.231059*(x1*x8 + x2*x9 + x3*x10) <= 250;
c2a: 12.62626*x8 - 1.231059*x1*x8 <= 150;
c2b: 12.62626*x9 - 1.231059*x2*x9 <= 150;
c2c: 12.62626*x10 - 1.231059*x3*x10 <= 150;
c3a: -3.475*x1 + 100*x4 + 0.0975*x1^2 - 9.75*x1*x4 = 0;
c3b: -3.475*x2 + 100*x5 + 0.0975*x2^2 - 9.75*x2*x5 <= 0;
c3c: -3.475*x3 + 100*x6 + 0.0975*x3^2 - 9.75*x3*x6 <= 0;
c4: -x4*x7 + x5*x7 - x1*x8 + x4*x8 >= 0;
c5: 50*x5 + 50*x6 - x1*x8 + x5*x8 + x2*x9 - x6*x9 = 500;
c5a: -50*x2 + 50*x6 + x2*x9 - x6*x9 + x2*x10 - x3*x10 >= 0;
c6: x7 - x8 >= 0; c7: x7 - x8 <= 50; c8: x5 <= x1;
c9: x1 <= x2; c10: x6 <= x2; c11: x2 <= x3;
```

In a paper results with a non-rigorous global optimization approach, $(z = -1161.34)$**.**

But, we find $z \in [-9.6903e + 02, -9.6845e + 02]$ **!**

Perspectives

- **Integrating other optimization techniques in a rigorous way: polyhedral relaxations for nonconvex terms, multilinear terms, ...**
- **Symbolic methods**
- **SDP relaxations**
- **Bernestein extension**
- **...**
- **Large scale optimization !**