Constraint Techniques for a Safe and Fast Implementation of Optimality-Based Reduction

Yahia Lebbah, Claude Michel and Michel Rueher

I3S (UNSA-CNRS)
We consider here the global optimization problem \( P \)

\[
\begin{align*}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & g_i(x) = 0, \quad i = 1..k \\
& g_j(x) \leq 0, \quad j = k + 1..m
\end{align*}
\] (1)

with

- \( x \in \mathbb{R}^n, \ f : \mathbb{R}^n \to \mathbb{R} \) and \( g_j : \mathbb{R}^n \to \mathbb{R}; \)
- functions \( f \) and \( g_j \) are continuously differentiable on \( x \)
Aim: rigorously solving $\mathcal{P}$

→ find an interval $[L, U] \ni \min f(x)$ such that $U - L \leq \epsilon$

where $U$, $L$ and $\epsilon$ are floating point numbers.

Means: branch and bound algorithm

- Based on a reduction approach which attempt to
  1. reduce the value of $U$ (upper bounding),
  2. increase the value of $L$ (lower bounding).

- With the help of a pruning step.
**Branch and bound algorithm**

**Function** \( \text{BB}(\text{IN } x, \epsilon; \text{OUT } S, [L, U]) \)

\( L \leftarrow \{x\}; \ S \leftarrow \emptyset; \ (L, U) \leftarrow (-\infty, +\infty); \)

**while** \( w([L, U]) > \epsilon \) **do**

\( x' \leftarrow x'' \text{ such that } f_{x''} = \min\{f_{x''} : x'' \in L\}; \quad L \leftarrow L \setminus x'; \)

\( \bar{f}_{x'} \leftarrow \min(\bar{f}_{x'}, U); \quad x' \leftarrow \text{Prune}(x'); \quad f_{x'} \leftarrow \text{Lower Bound}(x'); \)

\( (\bar{f}_{x'}, x_p, \text{Proved}) \leftarrow \text{Upper Box}(x'); \)

**if** \( \text{Proved} \) **then** \( S \leftarrow S \cup \{x_p\}; \) **endif**

**if** \( x' \neq \emptyset \) **then** \( (x_1', x_2') \leftarrow \text{Split}(x'); \quad L \leftarrow L \cup \{x_1', x_2'\}; \) **endif**

**if** \( L = \emptyset \) **then** \( (L, U) \leftarrow (+\infty, -\infty); \)

**else** \( (L, U) \leftarrow (\min\{f_{x''} : x'' \in L\}, \min\{\bar{f}_{x''} : x'' \in S\}); \)

**endif**

**endwhile**
Upper bounding

Aim: reduce the value of the global $U$

Means: compute a feasible point!

- local search → approximate feasible point $x_{\text{approx}}$
- epsilon inflation process and proof → provide a feasible box $x_{\text{proved}}$

compute $f^* = \min(f(x_{\text{proved}}), \bar{f}^*)$

Then, the global $U$ is the smallest of the local $\bar{f}^*$
Lower bounding

- **Aim:** increase the value of the global $L$
- **Means:** relaxing the problem
  - linear relaxation $R$ of $P$
  $$
  \begin{align*}
  \min \quad & d^T x \\
  \text{s.t.} \quad & Ax \leq b
  \end{align*}
  \tag{2}
  $$
- **LP solver** → $f^*$
- **Issues:**
  - efficient LP solvers work with floats
    → compute a *safe lower bound* of the min
  - relaxation coefficients are computed
    → must be *rigorously* computed

Then, the global $L$ is the smallest of the local $f^*$
Optimality Base Reduction (OBR)

- Why? there is room for improvement ...
- OBR is a way to speed up the reduction process
- How? (Ryoo & Sahinidis 96)

$$x_i' = x_i - \frac{U-L}{\lambda_i}$$

$$x_i = x_i + \frac{U-L}{\lambda_i}$$

does not modify the very branch and bound process
Theorems of OBR

The theorem says that knowing \([L, U]\) (the domain of \(f\)), if the constraint \(x_i - \overline{x}_i \leq 0\) is active at the optimal solution of \(R\) (i.e. \(x_i - \overline{x}_i = 0\) if \(x_i\) is set to this optima) and has a corresponding multiplier \(\lambda_i^* > 0\) (i.e. \(\lambda^*\) is the optimal solution of the dual of \(R\)). Then

\[
x_i \geq x'_i \text{ with } x'_i = \overline{x}_i - \frac{U - L}{\lambda_i^*}.
\] (3)

Thus, if \(x'_i > x_i\), the domain of \(x_i\) can be shrunked to \([x'_i, \overline{x}_i]\) without loss of any global optima.

similar theorems for \(\overline{x}_i - x_i \leq 0\) and \(g_i(x) \leq 0\).
OBR Issues

Main issue: the available dual solution $\lambda^*$ is an approximation ...
if used in OBR ...
... OBR might lose the global optima!

Solutions: two ways to take advantage of OBR
1. build a proved dual solution (Kearfott) ...
2. validate the reduction proposed by OBR with CP!
A proved dual solution (Kearfott)

Prove existence of a solution of a system of equations combining

- the dual of linear relaxation \( R (2) \)

\[
\begin{align*}
\max \quad & b^T y \\
\text{s.t.} \quad & A^T \lambda = d
\end{align*}
\]

(4)

- with the Kuhn-Tucker conditions which provides lower and upper bounds on \( R (2) \) and its dual (4):

\[
(\text{KT}) \begin{cases}
A^T \lambda - d = 0 \\
\lambda_i (A_i,:) x - b_i) = 0, 1 \leq i \leq m
\end{cases}
\]

(5)

where \( A_{i,:} \) is \( i \)-th row of \( A \).
Kearfott’s approach drawbacks

- Critical issue:
  - The system is overconstrained (→ cannot directly use existence theorems).

- To get a squared system
  → relax the relaxation (remove constraints) acceptable as all we need is a lower bound of $\lambda^*$

- Drawbacks:
  - Less efficient to prove the existence of a solution.
  - Bounds may be wide due to weakened relaxation.
Prove that no global solution is lost!

*Essential observation:* if the constraint system

\[
L \leq f(x) \leq U
\]

\[
g_i(x) = 0, \ i = 1..k
\]

\[
g_j(x) \leq 0, \ j = k + 1..m
\]

has *no solution* with \(x\) set to \([x_i, x'_i]\), then the reduction computed by OBR is valid.

- Task achieved by a *classical filtering process*

- Otherwise add this box to the list of boxes to process
$\mathcal{L}_r \leftarrow \emptyset$  \textbf{\%} $\mathcal{L}_r$: set of potential non-solution boxes

\textbf{for each variable} $x_i$ \textbf{do}

Apply OBR

and add the generated potential non-solution boxes to $\mathcal{L}_r$

\textbf{for each box} $B_i$ in $\mathcal{L}_r$ \textbf{do}

$B_i' \leftarrow 2B$-filtering($B_i$)

\textbf{if} $B_i' = \emptyset$ \textbf{then} reduce the domain of $x_i$

\textbf{else} $B_i'' \leftarrow \text{Quad-filtering}(B_i')$

\textbf{if} $B_i'' = \emptyset$ \textbf{then} reduce the domain of $x_i$

\textbf{else} add $B_i$ to global list of box to be handled \textbf{endif}

\textbf{endif}
Experimental Results (1)

- Compares 4 versions of the branch and bound algorithm:
  - without OBR
  - with unsafe OBR
  - with safe OBR based on Kearfott’s approach
  - with safe OBR based on CP techniques implemented with Icos using Coin/CLP and Coin/Ipopt.

- On 78 benches (from Ryoo & Sahinidis 1995, Audet thesis and the coconut library)

- All experiments have been done on PC-Notebook/1Ghz.
Experimental Results (2): Synthesis

Synthesis of the results:

<table>
<thead>
<tr>
<th></th>
<th>$\Sigma_t(s)$</th>
<th>%saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>no OBR</td>
<td>2384.36</td>
<td>-</td>
</tr>
<tr>
<td>unsafe OBR</td>
<td>881.51</td>
<td>63.03%</td>
</tr>
<tr>
<td>safe OBR Kearfott</td>
<td>1975.95</td>
<td>17.13%</td>
</tr>
<tr>
<td>safe OBR CP</td>
<td>454.73</td>
<td>80.93%</td>
</tr>
</tbody>
</table>

(with a timeout of 500s)
## Experimental Results (3): best cases

<table>
<thead>
<tr>
<th>name</th>
<th>Safe OBR CP</th>
<th>Safe OBR Kearfott</th>
<th>% saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>himmel11</td>
<td>2.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ex5_2_2_case3</td>
<td>4.49</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>c-chem7</td>
<td>10.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ex2_1_5</td>
<td>3.52</td>
<td>15.64</td>
<td>77.49%</td>
</tr>
<tr>
<td>ex7_2_5</td>
<td>2.19</td>
<td>7.14</td>
<td>69.32%</td>
</tr>
<tr>
<td>ex14_2_5</td>
<td>2.48</td>
<td>7.19</td>
<td>65.50%</td>
</tr>
<tr>
<td>ex7_2_10</td>
<td>0.16</td>
<td>0.41</td>
<td>60.97%</td>
</tr>
<tr>
<td>ex8_1_6</td>
<td>0.73</td>
<td>1.54</td>
<td>52.59%</td>
</tr>
<tr>
<td>ex14_2_2</td>
<td>3.05</td>
<td>6.38</td>
<td>52.19%</td>
</tr>
<tr>
<td>ex7_3_1</td>
<td>3.38</td>
<td>5.95</td>
<td>43.19%</td>
</tr>
</tbody>
</table>
## Experimental Results (4): worst cases

<table>
<thead>
<tr>
<th>name</th>
<th>Safe ORB CP</th>
<th>Safe OBR Kearfott</th>
<th>% saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>ex2_1_1</td>
<td>0.25</td>
<td>0.21</td>
<td>-19.04%</td>
</tr>
<tr>
<td>ex9_1_9</td>
<td>0.11</td>
<td>0.09</td>
<td>-22.22%</td>
</tr>
<tr>
<td>ex2_1_4</td>
<td>1.25</td>
<td>1.01</td>
<td>-23.76%</td>
</tr>
<tr>
<td>ex9_2_8</td>
<td>0.05</td>
<td>0.04</td>
<td>-25%</td>
</tr>
<tr>
<td>c-chem18</td>
<td>4.83</td>
<td>3.57</td>
<td>-35.29%</td>
</tr>
<tr>
<td>ex3_1_2</td>
<td>0.19</td>
<td>0.14</td>
<td>-35.71%</td>
</tr>
<tr>
<td>c-audet147</td>
<td>0.61</td>
<td>0.44</td>
<td>-38.63%</td>
</tr>
<tr>
<td>c-audet140b</td>
<td>0.13</td>
<td>0.09</td>
<td>-44.44%</td>
</tr>
<tr>
<td>alkyln</td>
<td>32.45</td>
<td>20.55</td>
<td>-57.90%</td>
</tr>
<tr>
<td>ex5_2_2_case1</td>
<td>7.75</td>
<td>2.88</td>
<td>-169.09%</td>
</tr>
</tbody>
</table>
Conclusion

Constraint programming techniques
- allow a safe and efficient implementation of OBR
- can outperform standard mathematical methods
- allow a safe embedding of OBR in a simple way thanks to refutation
- might be suitable for other unsafe methods

The End!