A Branch-and-Bound Algorithm for Unconstrained Global Optimization

Laurent Granvilliers and Alexandre Goldsztejn

Université de Nantes – LINA – CNRS

- Interval-based BB framework
- New hybrid algorithm
- Experiments using Realpaver

Unconstrained Global Optimization

- We consider
  - a class $\mathcal{C}^2$ function $f : \mathbb{R}^n \to \mathbb{R}$ to be minimized
  - a box $B \subseteq \mathbb{R}^n$ (bound constraints, search space)

- We define
  - a global minimizer as an element $x^* \in B$ such that

$$\forall x \in B: \ f(x^*) \leqslant f(x)$$

  - the global minimum $f^* = f(x^*)$ for every global minimizer x^*

- Our goal is to compute within given tolerances
  - a set of boxes $\mathcal{S}$ enclosing the set of all global minimizers
  - an interval $[l, u]$ enclosing f^*

Branch-and-Bound

- Basic principles
  - the initial box B is recursively split into sub-boxes, forming a tree, until given tolerances are reached
  - the best upper bound u of f^* is maintained
  - the lower bound of f is calculated on every sub-box; this box can be rejected if this lower bound is strictly greater than u
- Complexity: worst case exponential in the depth of the tree
- An efficient algorithm requires many other techniques
  - constraint-based reduction methods
  - graph decomposition techniques
  - use of optimality conditions
  - local optimization
  - adaptive splitting techniques

Interval Enclosures

Definition

An interval extension of f is an interval function $\boldsymbol{f}$ such that

$$\forall \boldsymbol{x} \subseteq B : \ \{f(x) : x \in \boldsymbol{x}\} \subseteq \boldsymbol{f}(\boldsymbol{x}).$$

Interval Enclosures

Definition

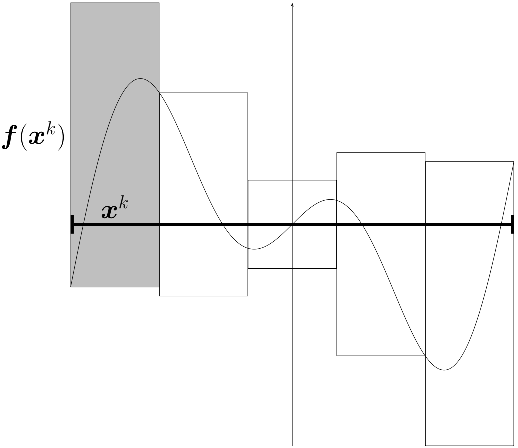
An interval extension of f is an interval function $\boldsymbol{f}$ such that

$$\forall \boldsymbol{x} \subseteq B: \ \{f(x) : x \in \boldsymbol{x}\} \subseteq \boldsymbol{f}(\boldsymbol{x}).$$

Consider a subdivision $\{\boldsymbol{x}^k\}_{k \in K}$ of B.

We use:

- natural form
- mean value form



Definition

An interval extension of f is an interval function $\boldsymbol{f}$ such that

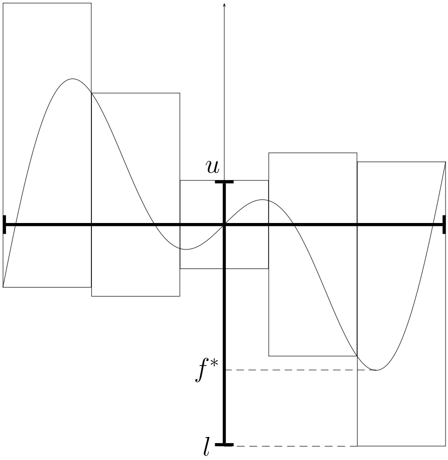
$$\forall \boldsymbol{x} \subseteq B: \ \{f(x) : x \in \boldsymbol{x}\} \subseteq \boldsymbol{f}(\boldsymbol{x}).$$

Consider a subdivision $\{\boldsymbol{x}^k\}_{k \in K}$ of B.

We have $f^* \in [l, u]$ such that:

$$l \leftarrow \min_{k \in K} \underline{\boldsymbol{f}(\boldsymbol{x}^k)}$$

$$u \leftarrow \min_{k \in K} \overline{\boldsymbol{f}(\boldsymbol{x}^k)}$$



Interval Enclosures

Definition

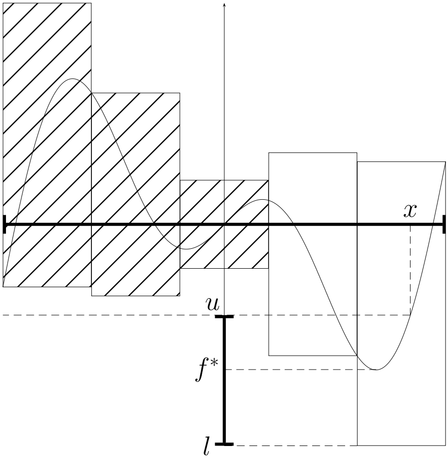
An interval extension of f is an interval function $\boldsymbol{f}$ such that

$$\forall \boldsymbol{x} \subseteq B: \ \{f(x) : x \in \boldsymbol{x}\} \subseteq \boldsymbol{f}(\boldsymbol{x}).$$

Consider a subdivision $\{\boldsymbol{x}^k\}_{k \in K}$ of B.

For all $x \in B$, $f(x)$ is an upper bound of f^*.

Improving u leads to reject sub-boxes (basic principle of BB).



Algorithm

$\mathcal{S} \leftarrow \varnothing$ % output boxes
$\mathcal{L} \leftarrow \{\langle B, \boldsymbol{f}(B)\rangle\}$ % boxes to be considered
$u \leftarrow +\infty$ % upper bound

while $\mathcal{L}$ is not empty **do**
  $\langle \boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x})\rangle \leftarrow \mathrm{pop}\,(\mathcal{L})$ % box $\boldsymbol{x}$ s.t. $\underline{\boldsymbol{f}(\boldsymbol{x})}$ is minimum
  $a \leftarrow \mathrm{Minimize}\,(f, \boldsymbol{x})$ % minimization of f within $\boldsymbol{x}$
  $u \leftarrow \min(u, \overline{\boldsymbol{f}(a)})$ % update the upper bound
  $\boldsymbol{x} \leftarrow \mathrm{Contract}\,(\boldsymbol{x}, B, f, u)$ % use constraints to contract $\boldsymbol{x}$
  $\mathrm{Branch}\,(\boldsymbol{x}, \varepsilon, \mathcal{L}, \mathcal{S})$ % split $\boldsymbol{x}$ if it is large enough
  update $\mathcal{L}$ and $\mathcal{S}$ according to u
end while

$l \leftarrow \min\{\underline{\boldsymbol{f}(\boldsymbol{x})} : \langle \boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x})\rangle \in \mathcal{S}\}$
return $\langle \mathcal{S}, [l, u]\rangle$

Branching Step

- $\text{Branch}\,(\boldsymbol{x}, \varepsilon, \mathcal{L}, \mathcal{S})$

  if $\boldsymbol{x}$ is not empty **then**
      if $\text{wid}\,(\boldsymbol{x}) \leqslant \varepsilon$ **then**
          $\mathcal{S} \leftarrow \mathcal{S} \cup \{\langle \boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}) \rangle\}$
      else
          $\{\boldsymbol{x}^k\}_{k \in K} \leftarrow \text{Split}\,(\boldsymbol{x}, \varepsilon)$
          $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle \boldsymbol{x}^k, \boldsymbol{f}(\boldsymbol{x}^k) \rangle : k \in K\}$
      end if
  end if

- Splitting heuristics
  - round robin, largest domain, smear function
  - bisection, trisection

Minimization Step

- The goal is to find a minimizer of f within the given box $\boldsymbol{x}$ in order to update the upper bound of f^*

- Some techniques
  - local optimization
  - grid-based exploration
  - metaheuristics
  - hybrid methods

- Our algorithm
  - line search based Newton's method: $x^0, x^1, x^2, \ldots$ such that

$$x^{k+1} \leftarrow x^k - \alpha_k H(x^k)^{-1} \nabla f(x^k),\ k \geqslant 0$$

  - greedy strategy with a grid of n points (linear) stopping when the upper bound is not improved enough
  - hybrid algorithm

Contraction Step

- Constraints that must be verified by every global minimizer x^*

$$\begin{array}{rl} i. & f(x) \leqslant u \\ ii. & x \notin \partial B \implies \nabla f(x) = 0 \end{array}$$

Contraction Step

- Constraints that must be verified by every global minimizer x^*

$$
\begin{array}{rl}
i. & f(x) \leqslant u \\
ii. & x \notin \partial B \implies \nabla f(x) = 0
\end{array}
$$

- Contracting operator for a constraint c: $\forall\, \boldsymbol{x}, \boldsymbol{x}' \subseteq B$

$$
\begin{array}{rl}
i. & \theta(\boldsymbol{x}) \subseteq \boldsymbol{x} \\
ii. & \forall x \in \boldsymbol{x} \setminus \theta(\boldsymbol{x}) : \neg c(x) \\
iii. & \boldsymbol{x} \subseteq \boldsymbol{x}' \implies \theta(\boldsymbol{x}) \subseteq \theta(\boldsymbol{x}')
\end{array}
$$

Contraction Step

- Constraints that must be verified by every global minimizer x^*

$$\begin{array}{rl} i. & f(x) \leqslant u \\ ii. & x \notin \partial B \implies \nabla f(x) = 0 \end{array}$$

- Contracting operator for a constraint c: $\forall\, \boldsymbol{x}, \boldsymbol{x}' \subseteq B$

$$\begin{array}{rl} i. & \theta(\boldsymbol{x}) \subseteq \boldsymbol{x} \\ ii. & \forall x \in \boldsymbol{x} \setminus \theta(\boldsymbol{x}) : \neg c(x) \\ iii. & \boldsymbol{x} \subseteq \boldsymbol{x}' \implies \theta(\boldsymbol{x}) \subseteq \theta(\boldsymbol{x}') \end{array}$$

- Constraint propagation given $\theta_1, \ldots, \theta_k$: $\forall \boldsymbol{x} \subseteq B$

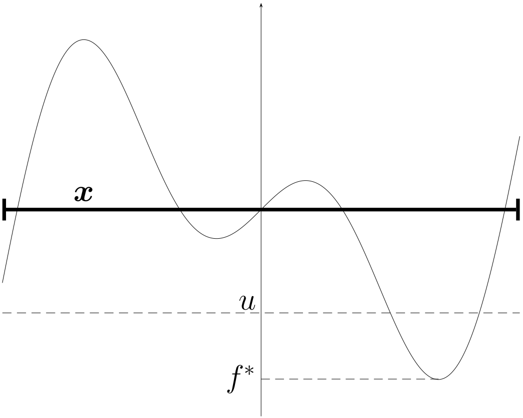
$$\boldsymbol{x} \mapsto \left(\bigcap_{i=1}^{k} \theta_i \right)^{\omega} (\boldsymbol{x})$$

Note: greatest common fixed-point of the θ_i included in $\boldsymbol{x}$

Inequality Constraint

$$f(x) \leqslant u \quad \text{s.t.} \quad f^* \leqslant u$$

$$\begin{array}{lcl} f(x) & = & x\cos(x) \\ \boldsymbol{x} & = & [-5, 5] \\ u & = & -2 \end{array}$$



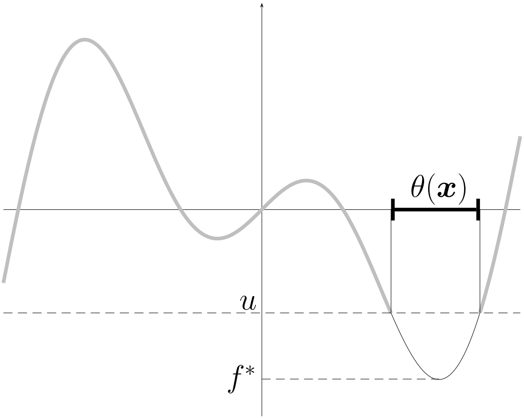
Inequality Constraint

$$f(x) \leqslant u \quad \text{s.t.} \quad f^* \leqslant u$$

$$\begin{array}{rcl} f(x) & = & x\cos(x) \\ \theta(\boldsymbol{x}) & = & [2.4, 4.3] \end{array}$$

Projection onto x

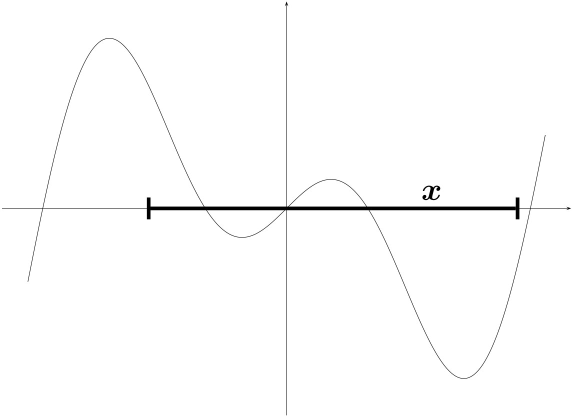
- 2B consistency
- box consistency



System of Equations

$$x \notin \partial B \implies \nabla f(x) = 0$$

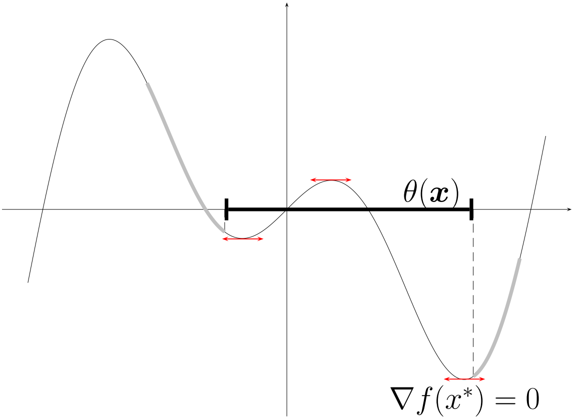
$$\begin{aligned} f(x) &= x\cos(x) \\ \boldsymbol{x} &= [-2.7, 4.5] \end{aligned}$$



System of Equations

$$x \notin \partial B \implies \nabla f(x) = 0$$

$$f(x) = x\cos(x)$$
$$\theta(\boldsymbol{x}) = [-1.2, 3.6]$$



Solving methods

- interval Newton
- consistency techniques

Experiments

- Implementation in Realpaver 1.1 (2010)
  - constraint solving and optimization library
  - high-level modeling language
  - interval arithmetic: gaol 3.1.1 (F. Goualard)
- Initial box: $[-10000, 1000]^n$ (not centered around 0)
- Tolerance $\varepsilon = 10^{-12}$
- Time Out (TO) : 600 seconds
- Processor: Intel Core2 Duo T7700 2.40GHz

Problems (1)

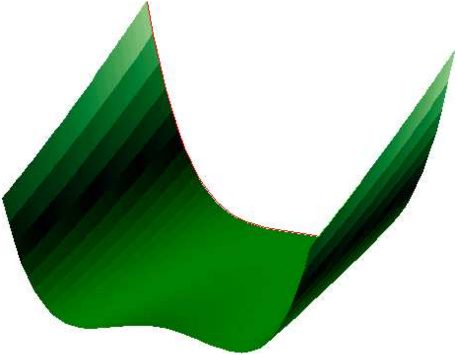
Schwefel function

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_i\right)^2$$

- polynomial, long narrow valley
- unimodal

Dixon-Price function

$$(x_1-1)^2+\sum_{i=2}^{n}\left(i\cdot(2x_i^2-x_{i-1})^2\right)$$

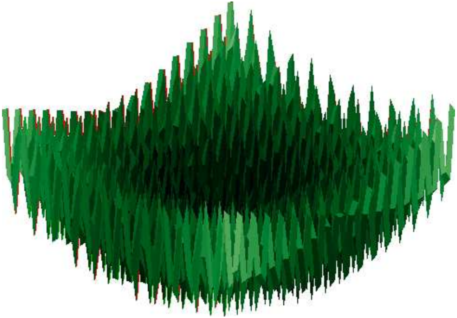


- polynomial, large valley
- two global minimizers

Problems (2)

Griewank function

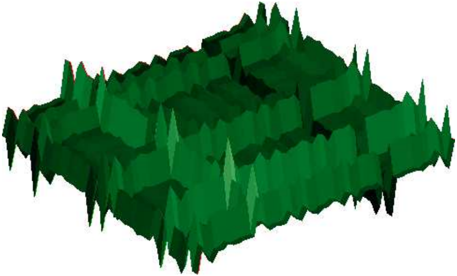
$$1 + \frac{1}{4000} \cdot \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$



- trigonometric, multimodal
- regular distribution of minimizers

Michalewicz function

$$-\sum_{i=1}^{n} \sin(x_i) \sin\left(i \cdot \frac{x_i^2}{\pi}\right)^{20}$$



- multimodal, high degree
- narrow peaks, large valleys

General Results

Problem	n	occ	sol	$u-l$	split	time
Dixon Price	20	39	2	10^{-29}	66	9.95
	40	79	2	10^{-29}	959	582.45
Rosenbrock	40	117	1	10^{-27}	1	1.40
	80	237	1	10^{-27}	4	18.70
Schwefel	20	210	1	10^{-323}	0	1.90
	40	820	1	10^{-323}	4	24.25
Griewank	40	80	1	10^{-15}	0	6.45
	80	160	1	10^{-15}	0	42.70
Michalewicz	5	10	1	10^{-14}	6	0.15
	10	20	1	10^{-14}	88	5.30
Rastrigin	20	40	1	10^{-13}	28	2.40
	40	80	1	10^{-13}	260	80.30

Contraction Step

We compare two algorithms for solving $\nabla f(x) = 0$

- Full: best algorithm
  - interval Newton method
  - consistency techniques and constraint propagation
- Newton: only the interval Newton method

Problem	n	Full		Newton	
		$u-l$	*time*	$u-l$	*time*
Dixon Price	20	10^{-29}	**9.95**	230.7	TO
Rosenbrock	40	10^{-27}	1.40	10^{-27}	**1.30**
Schwefel	20	10^{-323}	1.90	10^{-323}	**1.55**
Griewank	40	10^{-15}	6.45	10^{-15}	**2.05**
Michalewicz	10	10^{-14}	**5.30**	0.05	TO
Rastrigin	20	10^{-13}	**2.40**	755.6	TO

Minimization Step

We compare three algorithms for improving the upper bound

- Newton descent method (local optimization)
- Grid-based algorithm
- Hybrid Grid+Newton

Problem	n	Descent	Grid	Hybrid
Dixon Price	20	**9.35**	9.90	9.95
Rosenbrock	40	**1.40**	7.35	1.40
Schwefel	20	**1.90**	TO	1.90
Griewank	40	**6.40**	28.60	6.45
Michalewicz	10	68.20	**1.20**	5.30
Rastrigin	20	8.85	**0.10**	2.40

Conclusions

- Constraint-based contracting techniques are used to enclose precisely and rigorously the global minimizers, resulting in a precise enclosure of the global minimum.

- Our algorithm is generic, described by an object model, and the components can vary independently from each others, allowing to test easily many combinations
  - interval extension
  - minimization step
  - contraction step
  - branching step

- Realpaver has been compared with two rigorous solvers: GlobSol (R.B. Kearfott) and iCOs (Y. Lebbah)
  - the first results show good performances
  - it should also be compared with other solvers

A Branch-and-Bound Algorithm for Unconstrained Global Optimization

Laurent Granvilliers and Alexandre Goldsztejn

Université de Nantes – LINA – CNRS