Nested interval branch-and-bound algorithm for min-max problem

Daniel IOAN[1], Jordan NININ[1], Benoit CLÉMENT[1]

ENSTA-Bretagne, LAB-STICC, Brest, France

{daniel.ioan, jordan.ninin, benoit.clement}@ensta-bretagne.fr

Keywords : *min-max optimization, interval analysis, global optimization, branch-and-bound algorithm*

1 Introduction

Many engineering problems (e.g., structured robust control [1, 2] or optimal control [3]) reduce to optimization programs for which global resolution is desired or required. In this kind of realistic problems, the parametric uncertainties and the non-convexity are not only some artifacts but rather built-in features that cannot be eluded. Therefore, the resulting min-max problem (1) has to be solved efficiently and accurately. The output of the algorithm consists of **(a)** a reliable solution w.r.t. the numerical criteria and **(b)** an enclosure of the global minimum with a user-defined precision.

$$\left\{ \min_{x\in\mathbb{X}} \quad \sup_{y\in\mathbb{Y}} \quad f(x,y) \right. \tag{1}$$

To this end, the interval analysis [4, 5] has proved to be an elegant and adequate tool in conjunction with the branch-and-bound approach [6, 7]. However, aspects regarding the convergence and performance of the algorithms are prone to be refined by employing several techniques such as linear relaxation, constraint propagation, or warm start.

2 Implementation aspects

The proposed method resides in nesting an interval branch-and-bound algorithm inside another one. More precisely, at each iteration of the main minimization algorithm, a secondary maximization one is engaged to provide an enclosure on $f_{sup}(x) = \sup_{y\in\mathbb{Y}} f(x,y), \forall x \in \mathbb{X}_{current}$.

Since complexity plays a significant role in our approach, taking into account the field of application, some refinement techniques need to be considered in order to improve the convergence of the second interval branch-and-bound algorithm. Hence, besides the classical acceleration techniques based on linear relaxation, constraint programming, or warm start, we propose a restriction of the searching domain $\mathbb{Y}$ (in the secondary algorithm) by eliminating the redundant/irrelevant information, concentrating only on some key points and that without losing the global optimum. Our technique relies on a monotonicity test and KKT (Karush-Kuhn-Tucker) conditions; both applied on an entire sub-domain $(\mathbf{x}, \mathbf{y})$ using interval arithmetic. The idea is briefly presented in Algorithm 1, where the secondary maximization interval branch-and-bound algorithm is detailed.

Consequently, we minimize memory consumption, avoiding a quick saturation of the available memory. Several strategies are discussed in order to efficiently obtain this restriction from a numerical perspective.

As stated above, the algorithm uses interval analysis and set-theory methods. Thus, for the practical implementation, the IBEX library [8] is employed, and some proof of concept illustrations are provided.

Algorithm 1 Secondary interval branch-and-bound with monotonicity test

Input: $\mathcal{L}_{\mathbf{x}}$ ▷ it contains all boxes $\mathbf{y}$ associated with $\mathbf{x}$

Output: $[\texttt{lb}, \texttt{ub}]$ ▷ an enclosure on $f_{sup}(\mathbf{x})$

while $width([\texttt{lb}, \texttt{ub}]) \leq \epsilon$ **do** ▷ Stop criterion

  Extract $\mathbf{y}$ from $\mathcal{L}_{\mathbf{x}}$.

  Compute bounds on $f(\mathbf{x}, .)$ over $\mathbf{y}$ ▷ via interval analysis.

  Search a good solution $\tilde{y}$. ▷ with a local solver.

  Update $\texttt{lb} = f(\mathbf{x}, \tilde{y})$.

  Bisect $\mathbf{y}$ into $\mathbf{y}_1$ and $\mathbf{y}_2$,

  for $\mathbf{y} \in \{\mathbf{y}_1, \mathbf{y}_2\}$ **do** ▷ Monotonicity test for $\mathbf{y}_1$ and $\mathbf{y}_2$

    Compute $\mathbf{J}$ for $\mathbf{x}$ and $\mathbf{y}$ ▷ $\mathbf{J}$ - an enclosure box of the gradient ∇f

    For each component $\mathbf{y}_i$ of $\mathbf{y}$: ▷ w.r.t. monotonicity given by each component of $\mathbf{J}$

$$\tilde{\mathbf{y}}_i = \begin{cases} \overline{\mathbf{y}}_i & \text{if } \overline{\mathbf{J}}_i \leq 0, \\ \underline{\mathbf{y}}_i & \text{if } \underline{\mathbf{J}}_i \geq 0, \\ \mathbf{y} & \text{else.} \end{cases}$$

    Insert $\tilde{\mathbf{y}}$ in $\mathcal{L}_{\mathbf{x}}$.

  end for

  Update $\texttt{ub} = \max_{\mathbf{y} \in \mathcal{L}_{\mathbf{x}}} \overline{f(\mathbf{x}, \mathbf{y})}$

end while ▷ **Remark**: an interval $\mathbf{y} = [\underline{y}, \overline{y}]$

References

[1] Dominique Monnet. "Global minmax optimization for robust H_∞ control". PhD thesis. École nationale supérieure de techniques avancées Bretagne, 2018.

[2] Luc Jaulin and Éric Walter. "Guaranteed tuning, with application to robust control and motion planning". In: *Automatica* 32.8 (1996), pp. 1217–1221.

[3] Ion Necoara et al. "Finite-horizon min–max control of max-plus-linear systems". In: *IEEE Transactions on Automatic Control* 52.6 (2007), pp. 1088–1093.

[4] Ramon E Moore. *Interval analysis*. Prince-Hall, Englewood Cliffs, NJ, 1966.

[5] Luc Jaulin et al. *Applied Interval Analysis*. Springer London, 2001.

[6] Jordan Ninin. "Optimisation globale basée sur l'analyse d'intervalles: relaxation affine et limitation de la mémoire". PhD thesis. Institut National Polytechnique de Toulouse, 2010.

[7] Emilio Carrizosa and Frédéric Messine. "An interval branch and bound method for global Robust optimization". In: *Journal of Global Optimization* 80.3 (2021), pp. 507–522.

[8] Gilles Chabert et al. *Ibex, an interval-based explorer*. 2007.