Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-25 (2015) 085–090

Branch and bound algorithm for the robustness analysis of uncertain systems [⋆]

L. Ravanbod [*] D. Noll [*] P. Apkarian [**]

[*] *Université de Toulouse, Institut de Mathématiques, Toulouse, France lalehravanbod@yahoo.fr, dominikus.noll@math.univ-toulouse.fr*
[**] *ONERA, Toulouse, France Pierre.Apkarian@onera.fr*

Abstract: Computing the worst-case spectral abscissa of a system with uncertain parameters allows to decide whether it is robustly stable in a given parameter range. Since this problem is NP-hard, we use a heuristic local optimization method based on a bundle trust-region strategy to compute good lower bounds. Then we employ branch-and-bound to certify the global maximum. A specific frequency sweeping technique is used to accelerate the global optimization.

Keywords: Spectral abscissa · minimum stability degree · frequency sweep · bundle trust-region

1. INTRODUCTION

We consider a parameter-dependent linear time-invariant system

$$\begin{aligned} \dot{x} &= Ax + Bp \\ q &= Cx + Dp \\ p &= \Delta q \end{aligned} \tag{1}$$

with $x(t) \in \mathbb{R}^n$, $p(t), q(t) \in \mathbb{R}^r$, where A, B, C, D are real matrices of appropriate sizes, and where Δ is an $r \times r$ diagonal matrix of the form

$$\Delta = \operatorname{diag}\left[\delta_1 I_{r_1}, \dots, \delta_m I_{r_m}\right] \tag{2}$$

with I_{r_i} an identity matrix of size r_i and $r = r_1 + \cdots + r_m$. Assuming that the matrix A is stable, we ask whether the system (1) remains stable for all choices $\delta \in [-1,1]^m$ of the uncertain real parameters. If we consider the matrices Δ in (2) in one-to-one correspondence with $\delta \in \mathbb{R}^m$, then this amounts to checking wether

$$\dot{x} = \left(A + B\Delta(I - D\Delta)^{-1}C\right)x \tag{3}$$

is stable for every $\delta \in [-1,1]^m$. We assume throughout that $I - D\Delta$ is invertible for every $\delta \in [-1,1]^m$, i.e., that (1) is robustly well-posed over $[-1,1]^m$. All rational parameter variations in a nominal system $\dot{x} = Ax$ can be represented via a suitable LFT of the form (1).

Recall that the spectral abscissa of a square matrix A is $\alpha(A) = \max\{\text{Re}(\lambda) : \lambda \text{ eigenvalue of } A\}$, and that stability of A is equivalent to $\alpha(A) < 0$. The problem of robust stability of (1) over $\delta \in [-1,1]^m$ can therefore be addressed by the optimization program

$$\alpha^* = \max_{\delta \in [-1,1]^m} \alpha\left(A(\delta)\right), \tag{4}$$

where $A(\delta) = A + B\Delta(I - D\Delta)^{-1}C$. As soon as the global optimum satisfies $\alpha^* < 0$, the system (1) is certified robustly stable over $\delta \in [-1,1]^m$, while a solution δ^* of (4) with $\alpha^* \geqslant 0$ gives a destabilizing choice of the uncertain parameters, which may represent valuable information for parametric robust synthesis, see Apkarian et al. (2014). When $\alpha^* < 0$ every solution $x(t)$ of (3) decays at least as fast as $e^{\alpha^* t}$, in which case $-\alpha^* > 0$ is also known as the *minimum stability degree* of (1).

2. BRANCH AND BOUND STRATEGY

In this section we present the main ingredients of our branch and bound algorithm for (4). The differences with earlier work by Gaston et al. (1988), Sideris et al. (1989), and Balakrishnan et al. (1991) are (i) the use of a sophisticated local solver which gives an improved lower bound, (ii) an evaluation procedure which avoids computing explicit upper bounds, and (iii) a new element which integrates frequency information in the setup. We will explain these improvements as we go.

2.1 Basic setup

For every subbox $\mathbf{\Delta} = \prod_{i=1}^m [a_i, b_i]$ of $[-1,1]^m$ with $-1 \leqslant a_i < b_i \leqslant 1$ we consider the subproblem

$$\alpha^*(\mathbf{\Delta}) = \max_{\delta \in \mathbf{\Delta}} \alpha\left(A(\delta)\right) \tag{5}$$

of (4) associated with $\mathbf{\Delta}$. During the algorithm we maintain a finite list $\mathscr{L}$ of subproblems specified by pairwise non-overlapping subboxes, called the *list of doables*. The algorithm stops as soon as the list $\mathscr{L}$ has been worked off. The list is initialized with the box $[-1,1]^m$. When a box $\mathbf{\Delta} \in \mathscr{L}$ comes up for evaluation, we call a decision procedure $\mathscr{P}$, called a *pruning test*, which decides whether or not $\mathbf{\Delta}$ can be pruned. When pruned, $\mathbf{\Delta}$ simply disappears from the list $\mathscr{L}$. When $\mathscr{P}$ decides that pruning is not possible, then $\mathbf{\Delta}$ is divided into two successor boxes $\mathbf{\Delta}'$, $\mathbf{\Delta}''$ of half volume, $\mathbf{\Delta}$ is removed from the list and $\mathbf{\Delta}'$, $\mathbf{\Delta}''$ are added, so that $\mathscr{L}$ grows by one. Usually we cut the box in two halves along a longest edge.

2.2 Lower bound

We use a local optimization method based on a bundle trust-region strategy Apkarian et al. (2015a,b) to compute a lower bound $\underline{\alpha} \leqslant \alpha^*$ of the global optimum. Suppose the

[⋆] The authors acknowledge financial support by Fondation d'Entreprise EADS (F-EADS) and Fondation de Recherche pour l'Aéronautique et l'Espace (FNRAE).

local optimum is attained at $\underline{\delta} \in [-1,1]^m$, then $\underline{\delta}$ is our candidate for the solution, called the *incumbent*. Since the local solver is fast, it is re-started within $\mathbf{\Delta}$ whenever a new subproblem $\mathbf{\Delta} \in \mathscr{L}$ is evaluated. This may lead to an improved lower bound and incumbent. The information provided within $\mathbf{\Delta}$ is also used to rank the boxes in the list of doables $\mathscr{L}$. A detailed description of the local solver can be found in Apkarian et al. (2015a,b). For the current analysis it is enough to know that when the algorithm is started with initial guess $\delta_0 \in \mathbf{\Delta}$, then it always ends with a local maximum $\underline{\delta} \in \mathbf{\Delta}$ satisfying $\alpha\left(A(\underline{\delta})\right) \geqslant \alpha\left(A(\delta_0)\right)$.

2.3 Pruning test

Following standard terminology, a function $\overline{\alpha}(\cdot)$, defined on boxes $\mathbf{\Delta}$, is called an upper bound if $\alpha^*(\mathbf{\Delta}) \leqslant \overline{\alpha}(\mathbf{\Delta})$. Given the current lower bound $\underline{\alpha}$ and a tolerance $\epsilon \geqslant 0$, the standard pruning test with upper bound $\overline{\alpha}(\cdot)$ is

$$\mathscr{P}_{\text{ub}} : \begin{cases} \text{if } \overline{\alpha}(\mathbf{\Delta}) \leqslant \underline{\alpha} + \epsilon \text{ then} & \texttt{pruning } \mathbf{\Delta} \\ \text{otherwise} & \texttt{not\_pruning} \end{cases} \qquad (6)$$

The idea is that the decision `pruning` is only issued when $\alpha^*(\mathbf{\Delta}) \leqslant \overline{\alpha}(\mathbf{\Delta}) \leqslant \underline{\alpha} + \epsilon$, in which case the present incumbent cannot be further improved within the tolerance ϵ by investigating subboxes of $\mathbf{\Delta}$. Hence $\mathbf{\Delta}$ can be eliminated.

It turns out that in order to reach the decision (6) it is not necessary to compute an upper bound. Any method which allows to certify that $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha} + \epsilon$ will be sufficient to reach the same decision. This is captured by the following

Definition 1. A decision procedure $\mathscr{P}$ which, given a box $\mathbf{\Delta}$ and a reference value $\underline{\alpha}$ on entry, and being allowed a tolerance $\epsilon \geqslant 0$, issues a decision between `pruning` $\mathbf{\Delta}$ and `not_pruning` is called a *pruning test* if the decision `pruning` $\mathbf{\Delta}$ is only issued when it is certified that $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha} + \epsilon$. □

We shall use the shorthand $\mathscr{P}(\mathbf{\Delta}, \underline{\alpha}, \epsilon) =$ `pruning`, respectively, $\mathscr{P}(\mathbf{\Delta}, \underline{\alpha}, \epsilon) =$ `not_pruning`. In order to succeed, a pruning test $\mathscr{P}$ has to satisfy the following property:

Definition 2. A pruning test $\mathscr{P}$ is *consistent* if for every $\epsilon > 0$ there exists $\eta > 0$ such that for every box $\mathbf{\Delta}$ with diameter $< \eta$ and for every $\alpha \geqslant 0$ the decision $\mathscr{P}(\mathbf{\Delta}, \alpha^*(\mathbf{\Delta}) + \alpha, \epsilon) =$ `pruning` is made. □

The explanation is that sufficiently small boxes will get pruned when the global lower bound $\underline{\alpha}$ is better than their value $\alpha^*(\mathbf{\Delta})$ within the allowed level of tolerance ϵ. For the classical pruning test (6) consistency amounts to requiring

$$\lim_{\mathbf{\Delta} \to 0} \alpha^*(\mathbf{\Delta}) - \overline{\alpha}(\mathbf{\Delta}) = 0.$$

In section 3 we shall present several consistent pruning tests, which do not require computing an upper bound $\overline{\alpha}(\cdot)$. This leads to an advantage in speed.

2.4 Presentation of the algorithm

In this section we present the algorithm by way of the pseudo-code given below. The principal property of the algorithm can be summarized by the following

Theorem 1. Suppose algorithm 1 is operated with a consistent pruning test $\mathscr{P}$ and tolerance level $\epsilon > 0$. Then it terminates with an empty list $\mathscr{L}$ after a finite number of steps, and on exit the returned lower bound $\underline{\alpha}$ satisfies $\alpha^* \leqslant \underline{\alpha} + \epsilon$. In particular, if $\underline{\alpha} + \epsilon < 0$, then a robust stability certificate for (1) is obtained.

Algorithm 1. Branch and bound for program (4).

```
1: Lower bound. Call local solver to compute lower
   bound $\underline{\alpha}$. Initialize list $\mathscr{L} = \{[-1,1]^m\}$.
2: while $\mathscr{L} \neq \emptyset$ do
3:     Choose first element $\mathbf{\Delta} \in \mathscr{L}$ for evaluation
4:     Call local solver in $\mathbf{\Delta}$ to update lower bound $\underline{\alpha}$.
5:     Call pruning test $\mathscr{P}$.
6:     if $\mathscr{P}(\mathbf{\Delta}, \underline{\alpha}, \epsilon) =$ pruning then
7:         Remove $\mathbf{\Delta}$ from $\mathscr{L}$
8:     else
9:         Remove $\mathbf{\Delta}$ and replace it by two successors
10:        $\mathbf{\Delta}'$, $\mathbf{\Delta}''$ in $\mathscr{L}$
11:    end if
12:    Update ordering of $\mathscr{L}$
13: end while
14: Return $\underline{\delta}$ and $\underline{\alpha}$.
```

Proof. 1) Let $\underline{\alpha}^{(n)}$ be the best lower bound found after iteration n. Then $\underline{\alpha}^{(n)} \leqslant \underline{\alpha}^{(n+1)} \to \underline{\alpha} \leqslant \alpha^*$. Now suppose first the algorithm ends finitely at iterate n, then at some stage $k \leqslant n$ a box $\mathbf{\Delta}$ containing a global maximum δ^* has been pruned. This box satisfies $\alpha^*(\mathbf{\Delta}) = \alpha^*$, and since the pruning test was based on $\underline{\alpha}^{(k)}$, we have $\alpha^* = \alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha}^{(k)} + \epsilon \leqslant \underline{\alpha} + \epsilon$. That gives the estimate claimed in the statement.

2) It obviously suffices to show that there exists $\eta > 0$ and an iteration counter n_0 such that for all counters $n \geqslant n_0$ boxes with $\operatorname{diam}(\mathbf{\Delta}) < \eta$ are automatically pruned when evaluated.

3) Since α is a continuous function, $\delta \mapsto \alpha\left(A(\delta)\right)$ is uniformly continuous on $[-1,1]^m$ by the hypothesis of robust well-posedness, hence there exists $\eta > 0$ such that for all all boxes with $\operatorname{diam}(\mathbf{\Delta}) < \eta$ and all $\delta, \delta' \in \mathbf{\Delta}$ we have $|\alpha\left(A(\delta)\right) - \alpha\left(A(\delta')\right)| < \frac{1}{2}\epsilon$. That means as soon as a box with $\operatorname{diam}(\mathbf{\Delta}) < \eta$ is evaluated, the local optimizer finds a value $\underline{\alpha}(\mathbf{\Delta})$ such that $|\alpha^*(\mathbf{\Delta}) - \underline{\alpha}(\mathbf{\Delta})| < \frac{1}{2}\epsilon$. If this evaluation occurs at iteration n, then $\alpha^*(\mathbf{\Delta}) \leqslant \underline{\alpha}(\mathbf{\Delta}) + \frac{1}{2}\epsilon \leqslant \underline{\alpha}^{(n)} + \frac{1}{2}\epsilon \leqslant \underline{\alpha} + \frac{1}{2}\epsilon$, because the lower bound is regularly updated.

4) Using consistency of $\mathscr{P}$, by reducing η found in 3), we can further assume that $\mathscr{P}(\mathbf{\Delta}, \alpha^*(\mathbf{\Delta}) + \alpha, \frac{1}{2}\epsilon) =$ `pruning` for every $\alpha \geqslant 0$ and every box with $\operatorname{diam}(\mathbf{\Delta}) < \eta$.

5) Now assume the algorithm does not terminate. Then there exist boxes $\mathbf{\Delta}_k$ of diameter $\leqslant \eta_k \to 0$, $\eta_k < \eta$, which are evaluated at counter n_k, but not pruned. Then $\alpha^*(\mathbf{\Delta}_k) \leqslant \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon$, hence $\mathscr{P}(\mathbf{\Delta}_k, \underline{\alpha}^{(n_k)}, \epsilon) = \mathscr{P}(\mathbf{\Delta}_k, \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon, \frac{1}{2}\epsilon) = \mathscr{P}(\mathbf{\Delta}_k, \alpha^*(\mathbf{\Delta}_k) + a_k, \frac{1}{2}\epsilon) =$ `pruning` by consistency, where $a_k = \underline{\alpha}^{(n_k)} + \frac{1}{2}\epsilon - \alpha^*(\mathbf{\Delta}_k) \geqslant 0$. This contradicts the assumption that $\mathbf{\Delta}_k$ was not pruned and completes the proof. □

3. CENTRALIZING LOOP TRANSFORMATION

In order to prepare our pruning tests we follow Balakrishnan et al. (1991) and apply a loop transformation to the system (A, B, C, D) with uncertainty $\delta \in \mathbf{\Delta}$ such that the transformed system $(\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D})$ has its uncer-