Global Optimization based on Contractor Programming

Jordan Ninin[1]

[1] ENSTA-Bretagne, Lab-STICC,
2 rue Francois Verny, 29806 Brest, France
jordan.ninin@ensta-bretagne.fr

Keywords: Global Optimization, Contractor Programming

Introduction

In this talk, we will present a general pattern based on contractor programming for designing a global optimization solver. This approach allows to solve problems with a wide variety of constraints. The complexity and the performance of the algorithm rely on the construction of contractors which characterize the feasible region. We illustrate the methodology on a H_∞ control synthesis under structural constraints.

General pattern to designing global optimization solvers

Contractor Programming is a methodology which allows to enclose each algorithm in a unify framework, in order to interact heterogeneous formulations or techniques. This approach is based on Interval Analysis.Using Contractor Programming, we will show a user-friendly way to solve problems with non-smooth functions, disjunctive constraints, non-mathematical constraints (such as "stay in an area defined by a polygon") and constraints with quantifiers (such as ForAll and Exists). This approach allows to design, in a single step, a model and a solver for a given problem.

Given a physical problem, the user can construct a contractor for the feasible region $\mathbb{X}$ of his problem. We denote this contractor $\mathcal{C}_{out}$. Moreover, using the counterparts of set-membership operators for contractors, we can construct in the same way a contractor for the negation of $\mathbb{X}$. This contractor is denoted by $\mathcal{C}_{in}$. The only required mathematical expression is the objective function, f_{cost}.

Given a box $[\mathbf{x}] \in \mathbb{R}^n$, $\mathcal{C}_{out}([\mathbf{x}])$ removes from $[\mathbf{x}]$ a part that does not contain a feasible solution. In the same way, $\mathcal{C}_{in}([\mathbf{x}])$ removes from $[\mathbf{x}]$ parts which are entire feasible; i.e. $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}])) \subseteq \mathbb{X}$. Thus, $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}]))$ is a feasible subset and we can perform a global optimization without constraint on it. If this step succeeds, this set can be discarded: indeed, if a new best current solution is found, we save it and it is proved that this set does not contain a better solution; else it is directly proved that no better solution can be found in this set $([\mathbf{x}]/\mathcal{C}_{in}([\mathbf{x}]))$.

The following algorithm describes a simple implementation pattern for a global optimization solver based on contractors. This algorithm is inspired from the *SIVIA* Algorithm (Set-Inversion Via Interval Analysis), which is used to compute the feasible set in a domain.

The inputs are an initial domain $[\mathbf{x}] \in \mathbb{IR}^n$, $\mathcal{C}_{out}$ a contractor for $\mathbb{X}$, $\mathcal{C}_{in}$ a contractor for $\overline{\mathbb{X}}$ and f_{cost} an objective function. The outputs are $\tilde{f}$, the global minimum value found and $\tilde{x}$, a global minimum. A boolean variable b is added for each element of $\mathcal{L}$ to indicate if the element is included in the feasible region.

H_∞ control synthesis under structural constraints

We will illustrate this new approach on a example on the control of a periodic second order system G with a PID controller K subject to two frequency constraints on the error e and on the command u of the closed loop system. The objective is to find $k = (k_p, k_i, k_d)$ minimizing the H_∞ norm of the controlled system.

$$G(s) = \frac{k\omega_n}{s^2 + 2\xi\omega_n s + \omega_n^2}, \quad K(s) = k_p + \frac{k_i}{s} + k_d s.$$

$(\tilde{x},\ \tilde{f}) =$ **OptimCtc** $([\mathbf{x}],\ \mathcal{C}_{out},\ \mathcal{C}_{in},\ f_{cost})$:

$\tilde{f} := +\infty$, denotes the current upper bound for the global minimum;
$\mathcal{L} := \{([\mathbf{x}], false)\}$, initialization of the data structure of the stored elements;
Let $\mathcal{C}_f$ a contractor based on the constraint $\{x\ :\ f_{cost}(x) \leq \tilde{f}\}$;
Repeat until a stopping criterion is fulfilled:
    Extract from $\mathcal{L}$ an element $([\mathbf{y}], b)$,
    Bisect the considered box $[\mathbf{y}]$: $[\mathbf{y}_1], [\mathbf{y}_2]$,
    for $j = 1$ to 2 :
        if $(b =$ false$)$ *then*
            Contract $[\mathbf{y}_i]$ with $\mathcal{C}_{out} \cap \mathcal{C}_f$,
            $[\mathbf{y}_{tmp}] := [\mathbf{y}_i]$,
            Contract $[\mathbf{y}_i]$ with $\mathcal{C}_{in}$,
            Add $([\mathbf{y}_i]$,false$)$ in $\mathcal{L}$.
            $[\mathbf{y}_{tmp}] := [\mathbf{y}_{tmp}]/[\mathbf{y}_i]$,
        else
            Contract $[\mathbf{y}_i]$ with $\mathcal{C}_f$,
            $[\mathbf{y}_{tmp}] := [\mathbf{y}_i]$,
        Try to **find** the global optimum without constraint in $[\mathbf{y}_{tmp}]$,
        if the search succeeds in a limited time *then*
            Update $\tilde{f}$ and $\tilde{x}$.
        else
            Add $([\mathbf{y}_{tmp}], true)$ in $\mathcal{L}$.
end.

Table 1: General pattern for a global optimization algorithm based on contractor programming.

The feasible region $\mathbb{K}_{in}$ of our global optimization problem have the following form, with Re_1 and Im_1 the real and imaginary part of the transfer function C_1 corresponding to the first constraint on the error, and Re_2 and Im_2 of C_2 corresponding to the second constraint on the command. The objective function consists to minimizing γ.

$$\begin{aligned}\mathbb{K}_1 &= \{(k,\gamma) \ : \ \|C_1\,(G\star K)\|_\infty \leq \gamma\} \\ &= \{(k,\gamma) \ : \ \forall\omega, \sqrt{Re_1^2(k,\omega)+Im_1^2(k,\omega)} \leq \gamma\},\end{aligned}$$

$$\begin{aligned}\mathbb{K}_2 &= \{(k,\gamma) \ : \ \|C_2\,(G\star K)\|_\infty \leq \gamma\} \\ &= \{(k,\gamma) \ : \ \forall\omega, \sqrt{Re_2^2(k,\omega)+Im_2^2(k,\omega)} \leq \gamma\},\end{aligned}$$

$$\mathbb{K}_{in} = \mathbb{K}_1 \cap \mathbb{K}_2.$$

This algorithm is implemented in the library IBEX which is free available. The goal of this library is to give all tools to the users for designing easily the best solver for its own problem.

References

[1] IBEX : *a C++ numerical library based on interval arithmetic and constraint programming.*
http ://www.ibex-lib.org

[2] P. Apkarian and D. Noll, Nonsmooth H_∞ Synthesis", *IEEE Transaction on Automatic Control*, 51(1):71–86, 2006.

[3] G. Chabert and L. Jaulin, Contractor Programming, *Artificial Interlligence*, 173:1079–1100, 2009.

[4] L. Jaulin and E. Walter, Set inversion via interval analysis for nonlinear bounded-error estimation, *Automatica* 29(4):1053–1064, 1993.

[5] J. Ninin, F. Messine and P. Hansen, A reliable affine relaxation method for global optimization, *Quarterly Journal of Operations Research*, to appear, 2015.