Certified Global Optimization using Max-Plus based Templates

Joint Work with B. Werner, S. Gaubert and X. Allamigeon

Third year PhD Victor MAGRON

LIX/CMAP INRIA, École Polytechnique

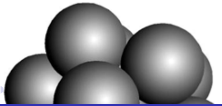
Motivation: Flyspeck-Like Problems

The Kepler Conjecture

Kepler Conjecture (1611):

The maximal density of sphere packings in 3D-space is $\frac{\pi}{\sqrt{18}}$

- It corresponds to the way people would intuitively stack oranges, as a pyramid shape
- The proof of T. Hales (1998) consists of thousands of non-linear inequalities



Contents

Motivation: Flyspeck-Like Problems

The Kepler Conjecture

Inequalities issued from Flyspeck non-linear part involve:

1. Multivariate Polynomials:

$$x_1 x_4(-x_1+x_2+x_3-x_4+x_5+x_6)+x_2 x_5(x_1-x_2+x_3+x_4-x_5+x_6)+ x_3 x_6(x_1+x_2-x_3+x_4+x_5-x_6)-x_2(x_3 x_4+x_1 x_6)-x_5(x_1 x_3+x_4 x_6)$$

2. Semi-Algebraic functions algebra $\mathcal{A}$: composition of polynomials with $|\cdot|, \sqrt{}, +, -, \times, /, \sup, \inf, \cdots$
3. Transcendental functions $\mathcal{T}$: composition of semi-algebraic functions with $\arctan, \exp, \sin, +, -, \times, \cdots$

Motivation: Flyspeck-Like Problems

Global Optimization Problems: Examples from the Literature

- *H3*: $\min_{\mathbf{x} \in [0,1]^3} - \sum_{i=1}^{4} c_i \exp \left[- \sum_{j=1}^{3} a_{ij} (x_j - p_{ij})^2 \right]$
- *MC*: $\min_{\substack{x_1 \in [-3,3] \\ x_2 \in [-1.5,4]}} \sin(x_1 + x_2) + (x_1 - x_2)^2 - 0.5 x_2 + 2.5 x_1 + 1$
- *SBT*: $\min_{\mathbf{x} \in [-10,10]^n} \prod_{i=1}^{n} \left(\sum_{j=1}^{5} j \cos((j+1) x_i + j) \right)$
- *SWF*: min $\sum^{n}$ …

Motivation: Flyspeck-Like Problems

Global Optimization Problems: a Framework

Given K a compact set, and f a transcendental function, minor $f^* = \inf_{\mathbf{x} \in K} f(\mathbf{x})$ and prove $f^* \geq 0$

1. f is underestimated by a semialgebraic function f_{sa}
2. We reduce the problem $f_{\text{sa}}^* := \inf_{\mathbf{x} \in K} f_{\text{sa}}(\mathbf{x})$ to a polynomial optimization problem in a lifted space K_{pop} (with lifting variables $\mathbf{z}$)
3. We solve the POP problem $f_{\text{pop}}^* := \inf_{(\mathbf{x},\mathbf{z}) \in K_{\text{pop}}} f_{\text{pop}}(\mathbf{x}, \mathbf{z})$ using a hierarchy of SDP relaxations by Lasserre

Classical Approach: Taylor + SOS

Semialgebraic Optimization Problems

- Polynomial Optimization Problem (POP):
  $p^* := \min_{\mathbf{x} \in K} p(\mathbf{x})$ with K the compact set of constraints:
  $$K = \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \cdots, g_m(\mathbf{x}) \geq 0\}$$
- Let $\Sigma[\mathbf{x}]$ be the cone of Sum-of-Squares (SOS) and consider the restriction $\Sigma_d[\mathbf{x}]$ to polynomials of degree at most $2d$:
  $$\Sigma_d[\mathbf{x}] = \left\{ \sum_i q_i(\mathbf{x})^2, \text{ with } q_i \in \mathbb{R}_d[\mathbf{x}] \right\}$$

Classical Approach: Taylor + SOS

Semialgebraic Optimization Problems

$$M(\mathbf{g}) = \left\{ \sum_{j=0}^{m} \sigma_j(\mathbf{x}) g_j(\mathbf{x}), \ \text{with } \sigma_j \in \Sigma[\mathbf{x}] \right\}$$

Proposition (Putinar)

Suppose $\mathbf{x} \in [\mathbf{a}, \mathbf{b}]$. $p(\mathbf{x}) - p^* > 0$ on $K \implies (p(\mathbf{x}) - p^*) \in M(\mathbf{g})$

- But the search space for $\sigma_0, \cdots, \sigma_m$ is infinite so consider the truncated module $M_d(\mathbf{g})$:

Classical Approach: Taylor + SOS

Semialgebraic Optimization Problems: examples

Lasserre Hierarchy Convergence:

- Let $k \geq k_0 := \max\{f, 1, \lceil \deg g_1/2 \rceil, \cdots, \lceil \deg g_m/2 \rceil\}$.
- The sequence $\inf(\mu_k)_{k \geq k_0}$ is non-decreasing. Under a certain (moderate) assumption, it converges to p^*.

- $\min_{\mathbf{x} \in [4, 6.3504]^6} \Delta(\mathbf{x}) = x_1 x_4(-x_1 + x_2 + x_3 - x_4 + x_5 + x_6) + x_2 x_5(x_1 - x_2 + x_3 + x_4 - x_5 + x_6) + x_3 x_6(x_1 + x_2 - x_3 + x_4 + x_5 - x_6) - x_2(x_3 x_4 + x_1 x_6) - x_5(x_1 x_3 + x_4 x_6) = \mu_2 = 128$

Classical Approach: Taylor + SOS

Semialgebraic Optimization Problems: examples

b.s.a.l. lemma [Lasserre, Putinar] :

Let $\mathcal{A}$ be the semi-algebraic functions algebra obtained by composition of polynomials with $|\cdot|$, $(\cdot)^{\frac{1}{p}} (p \in \mathbb{N}_0), +, -, \times, /, \sup, \inf$. Then every well-defined $f_{sa} \in \mathcal{A}$ has a basic semi-algebraic lifting.

Example from Flyspeck:

$$z_1 := \sqrt{4 x_1 \Delta \mathbf{x}}, \; m_1 = \inf_{\mathbf{x} \in [4, 6.3504]^6} z_1(\mathbf{x}), \; M_1 = \sup_{\mathbf{x} \in [4, 6.3504]^6} z_1(\mathbf{x}).$$

$$h_1 := z_1 - m_1 \qquad h_5 := z_1 \ldots$$

Classical Approach: Taylor + SOS

Taylor Approximation of Transcendental Functions

SWF: $\min_{\mathbf{x} \in [1,500]^n} f(\mathbf{x}) = - \sum_{i=1}^{n} (x_i + x_{i+1}) \sin(\sqrt{x_i})$

Classical idea: approximate $\sin(\sqrt{\cdot})$ by a degree-d Taylor

Polynomial f_d, solve $\min_{\mathbf{x} \in [1,500]^n} - \sum_{i=1}^{n} (x_i + x_{i+1}) f_d(x_i)$ (POP)

Issues:

- Lack of accuracy if d is not large enough $\Longrightarrow$ expensive Branch and Bound

Max-Plus Based Templates Approach

Main Purpose

Goals:

- Reduce the $O(n^{2k})$ polynomial dependency: decrease the number of lifting variables
- Reduce the $O(n^{2k})$ exponential dependency: use low degree approximations
- Reduce the Branch and Bound iterations: refine the approximations with an adaptive iterative algorithm

Max-Plus Based Templates Approach

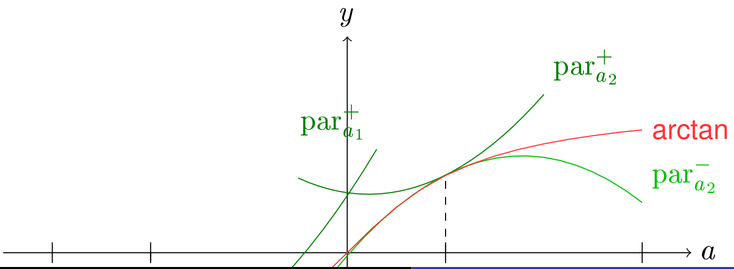
Max-Plus Estimators and Semi-convexity

- Let $\hat{f} \in \mathcal{T}$ be a transcendental univariate elementary function such as $\arctan$, $\exp$ defined on a real interval I.
- Convexity/semi-convexity properties and monotonicity of $\hat{f}$
- $\hat{f}$ is semi-convex: there exists a constant $c_j > 0$ s.t. $a \mapsto \hat{f}(a) + c_j/2(a - a_j)^2$ is convex
- By convexity:
  $\forall a \in I, \hat{f}(a) \geq -c_j/2(a - a_j)^2 + \hat{f}'(a_j)(a - a_j) + \hat{f}(a_j) = \mathrm{par}^-_{a_j}(a)$
- $\forall j, \hat{f} \geq \mathrm{par}^-_{a_j} \implies \hat{f} \geq \max_j \{\mathrm{par}^-_{a_j}\}$ **Max-Plus underestimator**

Max-Plus Based Templates Approach

Max-Plus Estimators for $\arctan$

Example with $\arctan$:



Max-Plus Based Templates Approach

Max-Plus Estimators on a Flyspeck example

- $l := -\frac{\pi}{2} + 1.6294 - 0.2213\,(\sqrt{x_2} + \sqrt{x_3} + \sqrt{x_5} + \sqrt{x_6} - 8.0) + 0.913\,(\sqrt{x_4} - 2.52) + 0.728\,(\sqrt{x_1} - 2.0)$

Lemma$_{9922699028}$ from Flyspeck:

$$\forall \mathbf{x} \in [4, 6.3504]^6, \arctan\left(\frac{\partial_4 \Delta \mathbf{x}}{\sqrt{4 x_1 \Delta \mathbf{x}}}\right) + l(\mathbf{x}) \geq 0$$

- Using semialgebraic optimization methods:

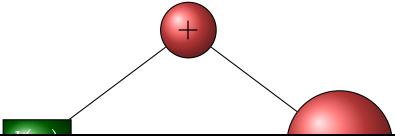
$$\forall x \in [4, 6.3504]^6,\ m \leq \frac{\partial_4 \Delta \mathbf{x}}{\ldots} \leq M$$

Max-Plus Based Templates Approach

Semialgebraic Max-Plus Approximations Algorithm

Abstract syntax tree representations of multivariate transcendental function:

- leaves are semialgebraic functions of $\mathcal{A}$
- nodes are univariate transcendental functions of $\mathcal{T}$ or binary operations



Max-Plus Based Templates Approach

Semialgebraic Max-Plus Approximations Algorithm

Recursive Algorithm `samp_approx`:

Input: tree t, box K, SDP relaxation order k, control points sequence $s = \mathbf{x}^1, \ldots, \mathbf{x}^p \in K$

Output: lower bound m, upper bound M, lower tree t^-, upper tree t^+

if $t \in \mathcal{A}$ **then**
    $t^- := t, t^+ := t$
else if $r := \mathtt{root}(t) \in \mathcal{T}$ parent of the single child c **then**
    $m_c, M_c, c^-, c^+ := \mathtt{samp\_approx}(c, K, k, s)$
    $\mathrm{par}^-, \mathrm{par}^+ := \mathtt{build\_par}(r, m_c, M_c, s)$

Max-Plus Based Templates Approach

Semialgebraic Max-Plus Optimization Algorithm

Iterative Algorithm samp_optim:

Input: tree t, box K, $iter_{\max}$ (optional argument)

Output: lower bound m, feasible solution $\mathbf{x}_{opt}$

1: $s := [\,$ argmin (randeval t) $]$ $\quad \triangleright\ s \in K$

2: $iter := 0$

3: $m := -\infty$

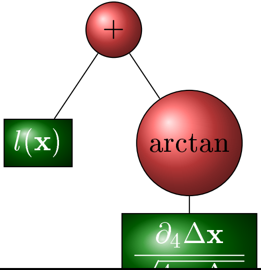
4: **while** $iter \leq iter_{\max}$ **do**

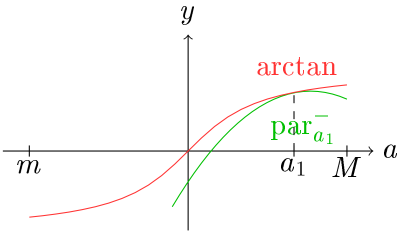
5: Choose an SDP relaxation order k

Max-Plus Based Templates Approach

Semialgebraic Max-Plus Optimization Algorithm

samp_optim First iteration:

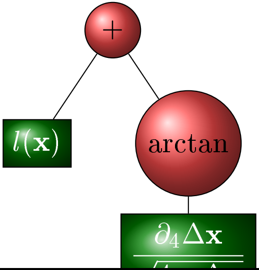


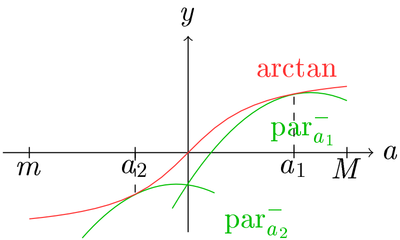


Max-Plus Based Templates Approach

Semialgebraic Max-Plus Optimization Algorithm

`samp_optim` Second iteration:

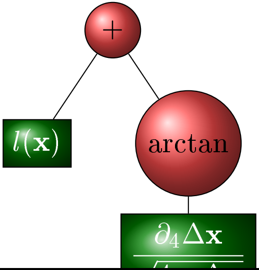


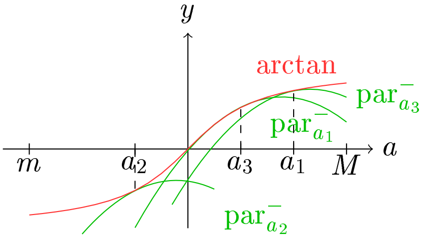


Max-Plus Based Templates Approach

Semialgebraic Max-Plus Optimization Algorithm

samp_optim Second iteration:





Max-Plus Based Templates Approach

Semialgebraic Max-Plus Optimization Algorithm

- For $k = 3, m_3 = -0.0333 < 0$, obtain a new minimizer $\mathbf{x}_{opt}^4$ and iterate again...

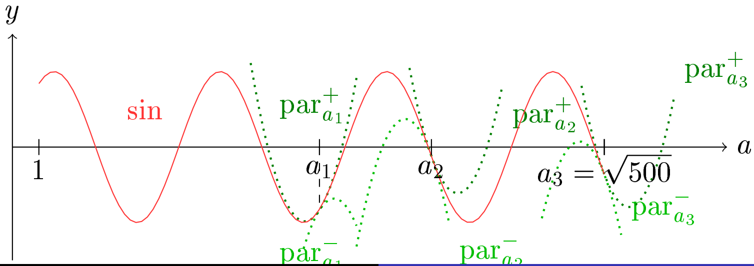
Theorem: Convergence of Semialgebraic underestimators

Let f be a multivariate transcendental function that can be represented by such syntaxic abstract trees.
Let $(\mathbf{x}_{opt}^p)_{p\in\mathbb{N}}$ be a sequence of control points obtained to define the hierarchy of underestimators in the algorithm `samp_optim` and $\mathbf{x}^*$ be an accumulation point of $(\mathbf{x}_{opt}^p)_{p\in\mathbb{N}}$.

Max-Plus Based Templates Approach

Semialgebraic Max-Plus Optimization Algorithm

Example with $\sin$:



Max-Plus Based Templates Approach

Semialgebraic Max-Plus Optimization Algorithm

SWF: $$\min_{\mathbf{x} \in [1,500]^n} - \sum_{i=1}^{n} (x_i + x_{i+1}) \sin(\sqrt{x_i}) \quad (\epsilon = 1)$$

- Use one lifting variable y_i to represent $x_i \mapsto \sqrt{x_i}$ and one lifting variable z_i to represent $x_i \mapsto \sin(x_i)$

$$\left\{ \begin{array}{ll} \min\limits_{\mathbf{x} \in [1,500]^n, \mathbf{y} \in [1, \sqrt{500}]^n, \mathbf{z} \in [-1,1]^n} & - \sum_{i=1}^{n} (x_i + x_{i+1}) z_i \\ \text{s.t.} & z_i \leq \mathrm{par}^+_{a_{ji}}(y_i), j \in \{1, 2, 3\} \end{array} \right.$$

Max-Plus Based Templates Approach

Polynomial Estimators using Semidefinite Programming

With `samp_optim`: the number of lifting variables is not bounded

Remedy: select some subcomponents of f and compute estimators involving less lifting variables

- Let t be such a subcomponent and $\mathbf{x}^j$ be a control point and suppose that t is twice differentiable.
- Define the interval matrix $\tilde{D}$ enclosing all the entries of $(\mathcal{D}^2(t)(\mathbf{x}) - \mathcal{D}^2(t)(\mathbf{x}_j))$ for $\mathbf{x} \in K$
- Define the quadratic form

Max-Plus Based Templates Approach

Polynomial Estimators using Semidefinite Programming

- Lower bound of $\min_{\mathbf{x} \in K} \{\lambda_{\min}(\mathcal{D}^2(t)(\mathbf{x}) - \mathcal{D}^2(t)(\mathbf{x}_j))\}$: $\lambda_{\min}(\tilde{D})$
  $\lambda^- := \lambda_{\min}(\tilde{D})$: minimal eigenvalue of an interval matrix
- For each interval $\tilde{D}_{ij} = [m_{ij}, M_{ij}]$, define the symmetric matrix entry $B_{ij} := \max\{| m_{ij} |, | M_{ij} |\}$
- Let $\mathcal{S}^n$ be the set of diagonal matrices of sign.
  $\mathcal{S}^n := \{\text{diag}\,(s_1, \cdots, s_n),\ s_1 = \pm 1, \cdots s_n = \pm 1\}$

Robut Optimization with Reduced Vertex Set [Calafiore, Dabbene]

Max-Plus Based Templates Approach

The Templates Algorithm

Previous Algorithm:

Input: tree t, box K, SDP relaxation order k, control points sequence $s = \mathbf{x}^1, \dots, \mathbf{x}^p \in K$

Output: lower bound m, upper bound M, lower tree t^-, upper tree t^+

1: **if** $t \in \mathcal{A}$ **then**
2: $t^- := t, t^+ := t$
3: **else if** $r := \texttt{root}(t) \in \mathcal{T}$ parent of the single child c **then**
4: $m_c, M_c, c^-, c^+ := \texttt{samp\_approx}(c, K, k, s)$
5: $\text{par}^-, \text{par}^+ := \texttt{build\_par}(r, m_c, M_c, s)$

Max-Plus Based Templates Approach

The Templates Algorithm: `template_optim`

Input: tree t, box K, SDP relaxation order k, control points sequence $s = \mathbf{x}^1, \ldots, \mathbf{x}^p \in K$

Output: lower bound m, upper bound M, lower tree t^-, upper tree t^+

if $t \in \mathcal{A}$ **then**
  return $\texttt{min}\ (t,\ k), \texttt{max}\ (t,\ k), t, t$
else if $r := \texttt{root}(t) \in \mathcal{T}$ parent of the single child c **then**
  $m_c, M_c, c^-, c^+ := \texttt{template\_optim}(c, K, k, s)$
  $\text{par}^-, \text{par}^+ := \texttt{build\_par}(r, m_c, M_c, s)$
  $t^-, t^+ := \texttt{compose}(\text{par}^-, \text{par}^+, c^-, c^+)$
  return $\texttt{min}\ (t^-,\ k), \texttt{max}\ (t^+,\ k), t^-, t^+$

Max-Plus Based Templates Approach

The Templates Algorithm: `build_template`

$\texttt{build}_{\texttt{template}}$ builds quadratic forms by solving SDP problems.

Input: tree t, box K, SDP relaxation order k, control points sequence $s = \mathbf{x}^1, \ldots, \mathbf{x}^p \in K$, lower/upper semialgebraic estimator t^-, t^+

1: **if** the number of lifting variables exceeds $n_{\text{lifting}}^{\max}$ **then**

2: **for** $\mathbf{x}^j \in s$ **do**

3: Compute the interval matrix $\tilde{D}^j$

4: $\lambda^- := \lambda_{\min}(\tilde{D}^j) \quad q_j^- := q_{\mathbf{x}^j, \lambda^-}$

5: $\lambda^+ := \lambda_{\max}(\tilde{D}^j) \quad q_j^+ := q_{\mathbf{x}^j, \lambda^+}$

6: **done**

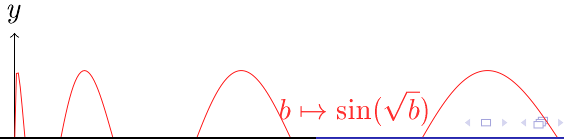
Max-Plus Based Templates Approach

The Templates Algorithm: *SWF*

When t is univariate, $\lambda^- = -c_j$ (the semi-convexity constant)

SWF: $$\min_{\mathbf{x} \in [1,500]^n} - \sum_{i=1}^{n} (x_i + x_{i+1}) \sin(\sqrt{x_i})$$

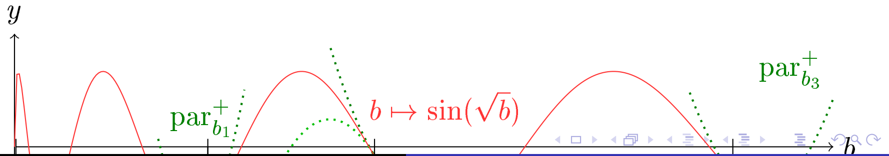
- Consider the univariate function $b \mapsto \sin(\sqrt{b})$ on $I = [1, 500]$



Max-Plus Based Templates Approach

The Templates Algorithm: *SWF*

- $\forall j, \hat{f} \geq \mathrm{par}^{-}_{b_j} \implies \hat{f} \geq \max_j \{ \mathrm{par}^{-}_{b_j} \}$: **Max-Plus underestimator**
- $\forall j, \hat{f} \leq \mathrm{par}^{+}_{b_j} \implies \hat{f} \leq \min_j \{ \mathrm{par}^{+}_{b_j} \}$: **Max-Plus overestimator**



Max-Plus Based Templates Approach

The Templates Algorithm: *SWF*

- Use a lifting variable z_i to represent $x_i \mapsto \sin(\sqrt{x_i})$
- For each i, pick points b_{ji}
- With 3 points b_{ji}, we solve the POP:

$$\begin{cases} \min\limits_{\mathbf{x} \in [1,500]^n, \mathbf{z} \in [-1,1]^n} & -\sum_{i=1}^{n} (x_i + x_{i+1}) z_i \\ \text{s.t.} & z_i \leq \text{par}^{+}_{b_{ji}}(x_i), j \in \{1, 2, 3\} \end{cases}$$

- POP with $n + n$ variables ($n_{\text{lifting}} = n$ variables), with

Max-Plus Based Templates Approach

Benchmarks: *SWF*

$$\min_{\mathbf{x} \in [1,500]^n} f(\mathbf{x}) = -\sum_{i=1}^{n} (x_i + \epsilon x_{i+1}) \sin(\sqrt{x_i})$$

n	lower bound	n_{lifting}	#boxes	time
$10 (\epsilon = 0)$	$-430n$	$2n$	16	$40\ s$
$10 (\epsilon = 0)$	$-430n$	0	827	$177\ s$
$1000 (\epsilon = 1)$	$-967n$	$2n$	1	$543\ s$

Max-Plus Based Templates Approach

Benchmarks: GO Literature

- SOS of degree $2k$
- $\#s$ control points (`template_optim` iterations)
- when $\#s = 0, n_{\text{lifting}} = 0$: interval arithmetic + SOS

Problem	n	lower bound	k	$\#s$	n_{lifting}	#boxes	time
H3	3	-3.863	2	3	4	99	101 s
				0	0	1096	247 s
				2	1	17	1.8 s

Max-Plus Based Templates Approach

Benchmarks: Flyspeck

- $n = 6$ variables, SOS of degree $2k = 4$
- $n_{\mathcal{T}}$ univariate transcendental functions
- $\#$boxes sub-problems

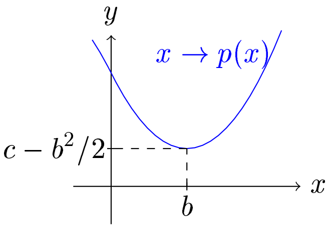
Inequality id	$n_{\mathcal{T}}$	n_{lifting}	$\#$boxes	time
9922699028	1	9	47	$241\ s$
9922699028	1	3	39	$190\ s$

Certified Global Optimization with Coq

Certification Framework: who does what?

Polynomial Optimization (POP): $\min_{x \in \mathbb{R}} p(x) = 1/2x^2 - bx + c$

1. A program written in OCaml/C provides the SOS decomposition: $1/2(x-b)^2$
2. A program written in Coq checks: $\forall x \in \mathbb{R}, p(x) = 1/2(x-b)^2 + c - b^2/2$



- Sceptical approach: obtain *certificates* of positivity with

Certified Global Optimization with Coq

Coq tactics: `field`, `interval`

Formal proofs for lower bounds of POP:

- The oracle returns floating point certificate: $\mu, \sigma_0, \cdots, \sigma_m$
- Check equality of polynomials: $f(\mathbf{x}) - \mu = \sum_{i=0}^{m} \sigma_i(\mathbf{x}) g_i(\mathbf{x})$

  with the Coq `field` tactic.
- The equality test often fails. Two solutions:
  1. Rounding and Projection of the certificate (Peyrl and Parillo, Kaltofen) until we get the equality

Certified Global Optimization with Coq

Polynomial Underestimators of Semialgebraic functions using SDP

- Let t be a semialgebraic leaf of the abstract syntaxic tree of f
- Let $\mathbf{x}^j \in K$ a control point
- Let λ denote the Lebesgue measure distributed on K

Consider the following optimization problem with optimal solution h_d^*:

$$\begin{cases} \min\limits_{h \in \mathbb{R}_d[\mathbf{x}]} & \displaystyle\int_K (t - h) d\lambda \\ \text{s.t.} & t - h \geq 0 \text{ on } K \end{cases}$$

Certified Global Optimization with Coq

Polynomial Underestimators of Semialgebraic functions using SDP

- There exist lifting variables $z_1, \cdots, z_p$ and polynomials $g_j \in \mathbb{R}[\mathbf{x}, \mathbf{z}], j = 1, \cdots, m$ defining the semialgebraic set: $K_{\text{pop}} := \{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{n+p} : \mathbf{x} \in K, g_1(\mathbf{x}, \mathbf{z}) \geq 0, \cdots, g_m(\mathbf{x}, \mathbf{z}) \geq 0\}$ such that $\Psi_t := \{(\mathbf{x}, t(\mathbf{x})) : \mathbf{x} \in K\} = \{(\mathbf{x}, z_p) : (\mathbf{x}, \mathbf{z}) \in K_{\text{pop}}\}$
- Then we can rewrite the previous optimization problem:

$$\begin{cases} \min\limits_{h \in \mathbb{R}_d[\mathbf{x}]} & \displaystyle\int_{K_{\text{pop}}} (z_p - h) d\lambda \\ \text{s.t.} & z_p - h(\mathbf{x}) \geq 0 \text{ on } K_{\text{pop}} \end{cases}$$

Certified Global Optimization with Coq

Polynomial Underestimators of Semialgebraic functions using SDP

- $K := \{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) \geq 0, \cdots, f_{2n}(\mathbf{x}) \geq 0\}$
- Let $g_0 := 1$ and $\omega_0 := \deg(g_0), \cdots, \omega_m := \deg(g_m)$
- For $k \geq k_0 = \max\{\lceil d/2 \rceil, \lceil \omega_1/2 \rceil, \cdots, \lceil \omega_m/2 \rceil\}$, introduce the following SDP relaxation F_{dk}:

$$\begin{cases} \max\limits_{h \in \mathbb{R}_d[\mathbf{x}], \sigma, \phi} & \displaystyle\int_K h d\lambda \\ \text{s.t.} & \forall \mathbf{x}, \mathbf{z}, z_p - h(\mathbf{x}) = \displaystyle\sum_{i=0}^{m} \sigma_i(\mathbf{x}, \mathbf{z}) g_i(\mathbf{x}) + \sum_{i=1}^{2n} \phi_i(\mathbf{x}, \mathbf{z}) f_i(\mathbf{x}) \end{cases}$$

Certified Global Optimization with Coq

Exploiting System Properties

- Templates preserve system properties: Sparsity / Symmetries
- Implementation in OCaml of the sparse variant of SDP relaxations (Kojima) for POP and semialgebraic underestimators
- Reducing the size of SDP input data has a positive domino effect:
  1. on the global optimization oracle to decrease the $O(n^{2d})$ complexity
  2. to check SOS with `field` and `interval` Coq tactics

End

Thank you for your attention!
Questions?