Lower Bounds for Polynomials in Exact Arithmetic

Henning Seidler

(joint with Victor Magron and Timo de Wolff)

July 12, 2019

SIAM Conference on Applied Algebraic Geometry,
Bern

The Problem

Task

Global minimisation of multivariate polynomial

Let $p \in \mathbb{R}[x_1, \ldots, x_n]$

$$\min \{p(x) : x \in \mathbb{R}^n\}$$

The Problem

Task

Global minimisation of multivariate polynomial

Let $p \in \mathbb{R}[x_1, \ldots, x_n]$

$$\min \{p(x) : x \in \mathbb{R}^n\}$$

equivalent

$$\min \{\lambda \in \mathbb{R} : \forall x \in \mathbb{R}^n \,.\, p(x) + \lambda \geq 0\}$$

The Problem

Task

Global minimisation of multivariate polynomial

Let $p \in \mathbb{R}[x_1, \ldots, x_n]$

$$\min \{p(x) : x \in \mathbb{R}^n\}$$

equivalent

$$\min \{\lambda \in \mathbb{R} : \forall x \in \mathbb{R}^n \,.\, p(x) + \lambda \geq 0\}$$

The Problem

Task

Global minimisation of multivariate polynomial

Let $p \in \mathbb{R}[x_1, \ldots, x_n]$

$$\min \{p(x) : x \in \mathbb{R}^n\}$$

equivalent

$$\min \{\lambda \in \mathbb{R} : \forall x \in \mathbb{R}^n \,.\, p(x) + \lambda \geq 0\}$$

want to prove non-negativity

Main Result – Software POEM

Main Result: "Effective Methods in Polynomial Optimisation"

- Version 0.2.1.0 just released
- Numerically compute lower bound for polynomial in polynomial time

Main Result – Software POEM

Main Result: "Effective Methods in Polynomial Optimisation"

- Version 0.2.1.0 just released
- Numerically compute lower bound for polynomial in polynomial time
- Symbolic Postprocessing
  Rounding to feasible rational solution
  - idea independent of future software improvements
  - even applicable for constrained optimisation

Non-Negativity

New Problem

Given sparse $p \in \mathbb{Q}[x]$. How to show $\forall x \in \mathbb{R}^n \,.\, p(x) \geq 0$?

Non-Negativity

New Problem

Given sparse $p \in \mathbb{Z}[x]$. How to show $\forall x \in \mathbb{Q}^n \,.\, p(x) \geq 0$?

Non-Negativity

New Problem

Given sparse $p \in \mathbb{Z}[x]$. How to show $\forall x \in \mathbb{Q}^n \,.\, p(x) \geq 0$?

- notation: t summands, n variables, degree $2d$
- input: list of (exponent-vector $\times$ coefficient)
- size: $\mathcal{O}\left(t \cdot n \cdot \log d + t \cdot \max\left\{\log |c| : c \in \text{coeff}(p)\right\}\right)$

Non-Negativity

New Problem

Given sparse $p \in \mathbb{Z}[x]$. How to show $\forall x \in \mathbb{Q}^n \,.\, p(x) \geq 0$?

- notation: t summands, n variables, degree $2d$
- input: list of (exponent-vector $\times$ coefficient)
- size (unit cost): $\mathcal{O}(tn)$

Non-Negativity

New Problem

Given sparse $p \in \mathbb{Z}[x]$. How to show $\forall x \in \mathbb{Q}^n \,.\, p(x) \geq 0$?

- notation: t summands, n variables, degree $2d$
- input: list of (exponent-vector $\times$ coefficient)
- size (unit cost): $\mathcal{O}(tn)$
- coNP-hard for $2d = 4$

Non-Negativity

New Problem

Given sparse $p \in \mathbb{Z}[x]$. How to show $\forall x \in \mathbb{Q}^n \,.\, p(x) \geq 0$?

- notation: t summands, n variables, degree $2d$
- input: list of (exponent-vector $\times$ coefficient)
- size (unit cost): $\mathcal{O}(tn)$
- coNP-hard for $2d = 4$
- use sufficient condition
- obvious way: sum of squares (SOS)
    - semi-definite programme with linear equations

Non-Negativity

New Problem

Given sparse $p \in \mathbb{Z}[x]$. How to show $\forall x \in \mathbb{Q}^n \,.\, p(x) \geq 0$?

- notation: t summands, n variables, degree $2d$
- input: list of (exponent-vector $\times$ coefficient)
- size (unit cost): $\mathcal{O}(tn)$
- coNP-hard for $2d = 4$
- use sufficient condition
- obvious way: sum of squares (SOS)
  - semi-definite programme with linear equations
  - size of problem is $\mathcal{O}\left(\binom{n+d}{d}^2\right)$

    $\rightsquigarrow$ exponential in input size

A New Approach

Circuit Polynomial: $$q = \sum_{i=0}^{n} b_i x^{a_i} + c x^y \qquad b_i > 0$$

points $a_i \in (2\mathbb{N})^n$ span simplex $\quad y \in \mathrm{int}(\mathrm{hull}(\{a_0, \ldots, a_n\}))$

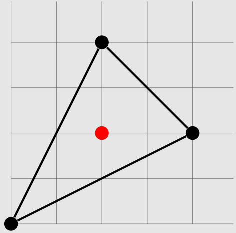
A New Approach

Circuit Polynomial: $$q = \sum_{i=0}^{n} b_i x^{a_i} + c x^y \qquad b_i > 0$$

points $a_i \in (2\mathbb{N})^n$ span simplex $\quad y \in \operatorname{int}(\operatorname{hull}(\{a_0, \ldots, a_n\}))$

Example (Motzkin polynomial)

$$q = 1 + x^2y^4 + x^4y^2 - 3x^2y^2$$



A New Approach

Circuit Polynomial: $q = \sum_{i=0}^{n} b_i x^{a_i} + c x^y \qquad b_i > 0$

points $a_i \in (2\mathbb{N})^n$ span simplex $\qquad y \in \text{int}(\text{hull}(\{a_0, \ldots, a_n\}))$

Theorem (Iliman, de Wolff)

Let $y = \sum \lambda_i a_i$ unique convex combination.
Define Circuit Number $\Theta = \prod \left(\frac{b_i}{\lambda_i}\right)^{\lambda_i}$

- $|c| \leq \Theta$ *or*
- $c \geq -\Theta$ *and y all even*

$\iff \forall x. q(x) \geq 0$

A New Approach

Circuit Polynomial: $q = \sum_{i=0}^{n} b_i x^{a_i} + c x^y \qquad b_i > 0$

points $a_i \in (2\mathbb{N})^n$ span simplex $\quad y \in \operatorname{int}(\operatorname{hull}(\{a_0, \ldots, a_n\}))$

Theorem (Iliman, de Wolff)

Let $y = \sum \lambda_i a_i$ unique convex combination.
Define Circuit Number $\Theta = \prod \left(\frac{b_i}{\lambda_i}\right)^{\lambda_i}$

- $|c| \leq \Theta$ *or*
- $c \geq -\Theta$ *and y all even*

$\iff \forall x . q(x) \geq 0$

$\Big\} \; c = -\Theta \implies \forall x \,.\, q(x) \geq 0$

A New Approach

Circuit Polynomial: $$q = \sum_{i=0}^{n} b_i x^{a_i} + c x^y \qquad b_i > 0$$

points $a_i \in (2\mathbb{N})^n$ span simplex $\quad y \in \mathrm{int}(\mathrm{hull}(\{a_0, \ldots, a_n\}))$

Decompose into

$$p \geq_{\min} \mathrm{relax(p)} = \sum q_k + C \geq C$$

where $$q_k = \sum_{i=0}^{n} b_{ik} x^{a_{ik}} + c_k x^{y_k}$$ as above

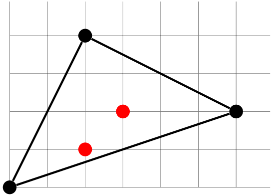
$$c_k = -\prod \left(\frac{b_{ik}}{\lambda_{ik}} \right)^{\lambda_{ik}}$$

SONC Optimisation in Simple Case

Let $p = b_0 + b_1 \cdot x_1^6 x_2^2 + b_2 \cdot x_1^2 x_2^4 - c_1 \cdot x_1^2 x_2 - c_2 \cdot x_1^3 x_2^2$.

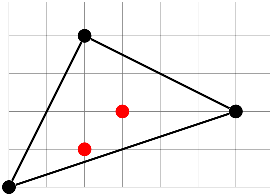
SONC Optimisation in Simple Case

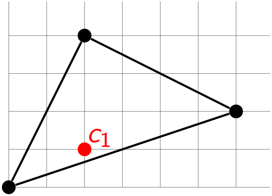
Let $p = b_0 + b_1 \cdot x_1^6 x_2^2 + b_2 \cdot x_1^2 x_2^4 - c_1 \cdot x_1^2 x_2 - c_2 \cdot x_1^3 x_2^2$.



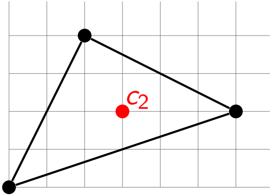
SONC Optimisation in Simple Case

Let $p = b_0 + b_1 \cdot x_1^6 x_2^2 + b_2 \cdot x_1^2 x_2^4 - c_1 \cdot x_1^2 x_2 - c_2 \cdot x_1^3 x_2^2$.



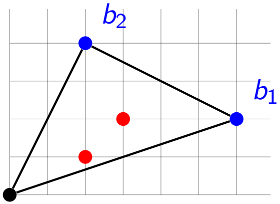


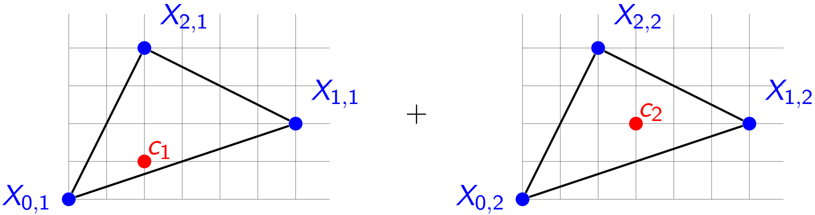
$+$



SONC Optimisation in Simple Case

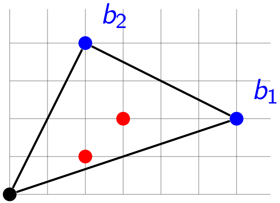
Let $p = b_0 + b_1 \cdot x_1^6 x_2^2 + b_2 \cdot x_1^2 x_2^4 - c_1 \cdot x_1^2 x_2 - c_2 \cdot x_1^3 x_2^2$.





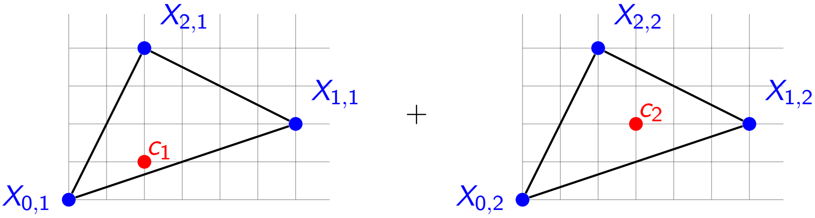
SONC Optimisation in Simple Case

Let $p = b_0 + b_1 \cdot x_1^6 x_2^2 + b_2 \cdot x_1^2 x_2^4 - c_1 \cdot x_1^2 x_2 - c_2 \cdot x_1^3 x_2^2$.



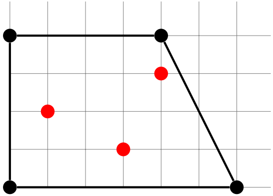
$$\min X_{0,1} + X_{0,2}$$
$$b_1 \geq X_{1,1} + X_{1,2}$$
$$b_2 \geq X_{2,1} + X_{2,2}$$

each non-negative



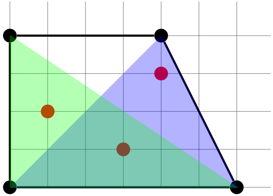
Decompose into Circuits

- write red as convex combination of black $\rightsquigarrow$ solve LPs
  $\rightsquigarrow$ covering of red points (no triangulation)



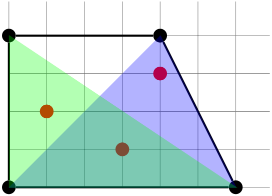
Decompose into Circuits

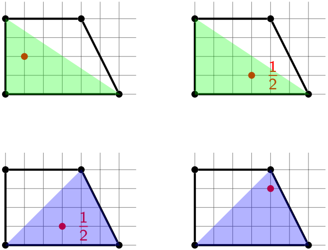
- write red as convex combination of black $\rightsquigarrow$ solve LPs
  $\rightsquigarrow$ covering of red points (no triangulation)



Decompose into Circuits

- write red as convex combination of black $\rightsquigarrow$ solve LPs
  $\rightsquigarrow$ covering of red points (no triangulation)
- distribute negative coefficients





Optimisation Programme

$$\text{relax}(p) = \sum_{i=0}^{m} b_i x^{a_i} - \sum_{k=1}^{r} c_k x^{y_k} \qquad b_i, c_k > 0$$

compute circuits with $\overline{c}_k x^{\overline{y}_k}$ as negative term
write each $\overline{y}_k = \sum \lambda_{ik} a_i$

Optimisation Programme

$$\text{relax}(p) = \sum_{i=0}^{m} b_i x^{a_i} - \sum_{k=1}^{r} c_k x^{y_k} \qquad b_i, c_k > 0$$

compute circuits with $\overline{c}_k x^{\overline{y}_k}$ as negative term
write each $\overline{y}_k = \sum \lambda_{ik} a_i$

$$\min \sum_{k=1}^{\overline{r}} X_{0k} \quad \text{s.t.} \quad b_i \geq \sum_{k=1}^{\overline{r}} X_{ik} \qquad (i = 1, \ldots, m)$$

$$\bar{c}_k = \prod_{i \in \text{circuit}_k} \left(\frac{X_{ik}}{\lambda_{ik}} \right)^{\lambda_{ik}} \qquad (k = 1, \ldots, \overline{r})$$

Optimisation Programme

$$\text{relax}(p) = \sum_{i=0}^{m} b_i x^{a_i} - \sum_{k=1}^{r} c_k x^{y_k} \qquad b_i, c_k > 0$$

compute circuits with $\overline{c}_k x^{\overline{y}_k}$ as negative term
write each $\overline{y}_k = \sum \lambda_{ik} a_i$

$$\min \sum_{k=1}^{\overline{r}} X_{0k} \quad \text{s.t.} \quad b_i \geq \sum_{k=1}^{\overline{r}} X_{ik} \qquad (i = 1, \ldots, m)$$

called Geometric Programme
$\rightsquigarrow$ good solvers

$$\bar{c}_k = \prod_{i \in \text{circuit}_k} \left(\frac{X_{ik}}{\lambda_{ik}} \right)^{\lambda_{ik}} \qquad (k = 1, \ldots, \overline{r})$$

Optimisation Programme

$$\text{relax}(p) = \sum_{i=0}^{m} b_i x^{a_i} - \sum_{k=1}^{r} c_k x^{y_k} \qquad b_i, c_k > 0$$

compute circuits with $\overline{c}_k x^{\overline{y}_k}$ as negative term
write each $\overline{y}_k = \sum \lambda_{ik} a_i$

$$\min \sum_{k=1}^{\overline{r}} X_{0k} \quad \text{s.t.} \quad b_i \geq \sum_{k=1}^{\overline{r}} X_{ik} \qquad (i = 1, \ldots, m)$$

$$\bar{c}_k = \prod_{i \in \text{circuit}_k} \left(\frac{X_{ik}}{\lambda_{ik}} \right)^{\lambda_{ik}} \qquad (k = 1, \ldots, \overline{r})$$

called Geometric Programme
$\rightsquigarrow$ good solvers

Observation: Size independent of degree!

Symbolic Certificates/Exact Arithmetic

$$\min \sum_{k=1}^{\bar{r}} X_{0k} \quad \text{s.t.} \quad b_i \geq \sum_{k=1}^{\bar{r}} X_{ik} \qquad (i = 1, \ldots, m)$$

$$\bar{c}_k = \prod_{i \in \text{circuit}_k} \left(\frac{X_{ik}}{\lambda_{ik}} \right)^{\lambda_{ik}} \qquad (k = 1, \ldots, \bar{r})$$

Observations

- each feasible solution yields lower bound

Symbolic Certificates/Exact Arithmetic

$$\min \sum_{k=1}^{\bar{r}} X_{0k} \quad \text{s.t.} \quad b_i \geq \sum_{k=1}^{\bar{r}} X_{ik} \qquad (i = 1, \dots, m)$$

$$\bar{c}_k = \prod_{i \in \text{circuit}_k} \left(\frac{X_{ik}}{\lambda_{ik}} \right)^{\lambda_{ik}} \qquad (k = 1, \dots, \bar{r})$$

Observations

- each feasible solution yields lower bound
- inequalities are independent
- X_{0k} does not appear in inequalities
- can solve equalities for X_{0k} (if $\mathbf{0} \in \text{circuit}_k \Leftrightarrow \lambda_{0k} \neq 0$)

Symbolic Certificates/Exact Arithmetic

$$\min \sum_{k=1}^{\bar{r}} X_{0k} \quad \text{s.t.} \quad b_i \geq \sum_{k=1}^{\bar{r}} X_{ik} \qquad (i = 1, \ldots, m)$$

$$\bar{c}_k = \prod_{i \in \text{circuit}_k} \left(\frac{X_{ik}}{\lambda_{ik}} \right)^{\lambda_{ik}} \qquad (k = 1, \ldots, \bar{r})$$

Observations

- each feasible solution yields lower bound
- inequalities are independent
- X_{0k} does not appear in inequalities
- can solve equalities for X_{0k} (if $\mathbf{0} \in \text{circuit}_k \Leftrightarrow \lambda_{0k} \neq 0$)

$\Rightarrow$ can refine any assignment to feasible solution

1. Compute the exact cover, including exact values for $\lambda_{i,k}$
   solve LPs exactly over $\mathbb{Q}$
2. Numerically solve the GP
3. Apply symbolic post-processing method

1. Compute the exact cover, including exact values for $\lambda_{i,k}$
   solve LPs exactly over $\mathbb{Q}$
2. Numerically solve the GP
3. Apply symbolic post-processing method

$$\widehat{X}_{i,k} := \left\lfloor \widetilde{X}_{i,k} \right\rceil \qquad \text{make rational}$$

$$X_{i,k} := \frac{b_i \cdot \widehat{X}_{i,k}}{\sum_k \widehat{X}_{i,k}} \qquad \text{scale, to match constraints}$$

$$X_{0,k} := \left\lceil \lambda_{0,k} \cdot \left(\overline{c}_k \cdot \prod_{\substack{i \in \text{cover}_k \\ i > 0}} \left(\frac{\lambda_{i,k}}{X_{i,k}} \right)^{\lambda_{i,k}} \right)^{\frac{1}{\lambda_{0,k}}} \right\rceil$$

1. Compute the exact cover, including exact values for $\lambda_{i,k}$ solve LPs exactly over $\mathbb{Q}$
2. Numerically solve the GP
3. Apply symbolic post-processing method

$$\widehat{X}_{i,k} := \left\lfloor \widetilde{X}_{i,k} \right\rceil \qquad \text{make rational}$$

$$X_{i,k} := \frac{b_i \cdot \widehat{X}_{i,k}}{\sum_k \widehat{X}_{i,k}} \qquad \text{scale, to match constraints}$$

$$X_{0,k} := \left\lceil \lambda_{0,k} \cdot \left(\overline{c}_k \cdot \prod_{\substack{i \in \text{cover}_k \\ i > 0}} \left(\frac{\lambda_{i,k}}{X_{i,k}} \right)^{\lambda_{i,k}} \right)^{\frac{1}{\lambda_{0,k}}} \right\rceil$$

Similar idea works for SAGE

Recap

Current state

software POEM: lower bound of polynomial and minimisers, in polynomial time; improved bounds in FPT, bounds in exact arithmetic

POEM homepage is linked on my homepage

Recap

Current state

software POEM: lower bound of polynomial and minimisers, in polynomial time; improved bounds in FPT, bounds in exact arithmetic
POEM homepage is linked on my homepage

Planned/Ongoing

- improve (computation of)
  - covering
  - distribution of coefficients (own iterative method)

Recap

Current state

software POEM: lower bound of polynomial and minimisers, in polynomial time; improved bounds in FPT, bounds in exact arithmetic

POEM homepage is linked on my homepage

Planned/Ongoing

- improve (computation of)
  - covering
  - distribution of coefficients (own iterative method)
- handle case $\lambda_{k,0} = 0$

Recap

Current state

software POEM: lower bound of polynomial and minimisers, in polynomial time; improved bounds in FPT, bounds in exact arithmetic

POEM homepage is linked on my homepage

Planned/Ongoing

- improve (computation of)
  - covering
  - distribution of coefficients (own iterative method)
- handle case $\lambda_{k,0} = 0$
- constrained polynomial optimisation (long term)

Questions

You have an answer? Please talk to me.

Complexity of deciding nonnegativity?

- (co)NP-problems with $\exists/\forall$ over unbounded space

Questions

You have an answer? Please talk to me.

Complexity of deciding nonnegativity?

- (co)NP-problems with $\exists/\forall$ over unbounded space

Is SOS $\in$ PSPACE, i.e. can we avoid the large matrix?

Questions

You have an answer? Please talk to me.

Complexity of deciding nonnegativity?

- (co)NP-problems with $\exists/\forall$ over unbounded space

Is SOS $\in$ PSPACE, i.e. can we avoid the large matrix?

Solving LPs: same coefficient matrix, small entries

- complexity, preprocessing, implementations?

Questions

You have an answer? Please talk to me.

Complexity of deciding nonnegativity?

- (co)NP-problems with $\exists/\forall$ over unbounded space

Is SOS $\in$ PSPACE, i.e. can we avoid the large matrix?

Solving LPs: same coefficient matrix, small entries

- complexity, preprocessing, implementations?

Implementations of exact methods

- Tarski's quantifier elimination
- Algorithm for existential theory over the reals

Questions

You have an answer? Please talk to me.

Complexity of deciding nonnegativity?

- (co)NP-problems with $\exists/\forall$ over unbounded space

Is SOS $\in$ PSPACE, i.e. can we avoid the large matrix?

Solving LPs: same coefficient matrix, small entries

- complexity, preprocessing, implementations?

Implementations of exact methods

- Tarski's quantifier elimination
- Algorithm for existential theory over the reals

Your questions to me?