Strengthening dual bounds in branch-and-bound by SONC certificates

Ksenia Bestuzheva, Ambros Gleixner, Helena Völker

SCIP 20th Anniversary Workshop
November 4, 2022

Definitions

Consider a **polynomial**

$$f(x) = \sum_{\alpha \in \mathcal{A}} f_\alpha x^\alpha = \sum_{\alpha \in \mathcal{A}} f_\alpha x_1^{\alpha_1} \cdot \ldots x_n^{\alpha_n},\ x \in \mathbb{R}^n.$$

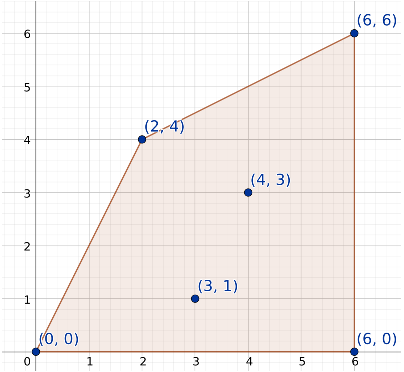
- $\mathcal{A} \subset \mathbb{N}^n$ – support of f
- $New(f) = conv(\{\alpha \in \mathcal{A} \mid f_\alpha \neq 0\})$ – Newton polytope of f (assume $New(f) = conv(\mathcal{A})$)
- $V(\mathcal{A})$ – vertices of $New(f)$
- $\Delta f = \{\alpha \in \mathcal{A} \mid f_\alpha \neq 0$ and $\{f_\alpha < 0$ or $\alpha \notin (2\mathbb{N})^n\}\}$ – non monomial square terms

Necessary condition for nonnegativity: **all vertices must be monomial squares**.

Example

$$f(x,y) = 3x^6y^6 + 2x^6 - x^4y^3 - x^3y + 5x^2y^4 - 5$$

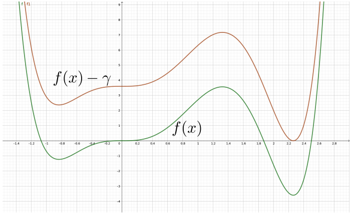
- The figure shows $New(f)$
- $\mathcal{A} = \{(6,6),(6,0),(4,3),(3,1),(2,4),(0,0)\}$
- 4 vertices: $(6,6),(6,0),(2,4),(0,0)$
- 2 points in the interior: $(4,3),(3,1)$
- $\Delta(f) = (4,3),(3,1)$



Finding Dual Bounds via Nonnegativity Certificates

$$f(x) - \gamma \geq 0 \;\Leftrightarrow\; f(x) \geq \gamma$$

Can find dual bound by maximising γ subject to $f(x) - \gamma \geq 0$.

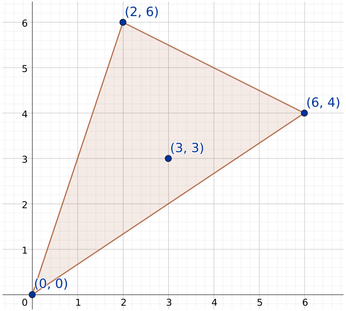


- Sum of squares (SOS) is a well-known nonnegativity certificate
- SOS scales poorly with the degree of the polynomials
- An alternative: sum of nonnegative circuit polynomials (SONC) certificates
- Cost of SONC depends on number of terms, not on degree

Circuit Polynomials

$$f(x) = \sum_{\alpha \in V(\mathcal{A})} f_\alpha x^\alpha + f_\beta x^\beta$$

- $\alpha \in V(\mathcal{A})$ are affinely independent and are monomial squares.
- β can be written as convex combination of α: $\beta = \sum_{\alpha \in V(\mathcal{A})} \lambda_\alpha \alpha, \ \ \lambda_\alpha \geq 0, \ \ \sum_{\alpha \in V(\mathcal{A})} \lambda_\alpha = 1.$



Nonnegativity of circuit polynomials:

- **Circuit number** $\Theta_f = \prod_{\alpha \in V(\mathcal{A})} \left(\frac{f_\alpha}{\lambda_\alpha}\right)^{\lambda_\alpha}$.
- $f(x) \geq 0 \ \forall x$ if and only if: $\beta \notin (2\mathbb{N})^n$ and $|f_\beta| \leq \Theta_f$ or $\beta \in (2\mathbb{N})^n$ and $f_\beta \geq -\Theta_f$

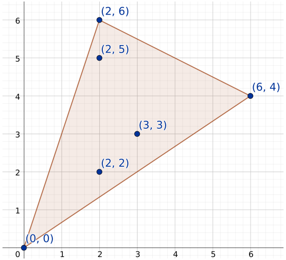
ST-Polynomials

$$f(x) = \sum_{\alpha \in V(\mathcal{A})} f_\alpha x^\alpha + \sum_{\beta \in \Delta f} f_\beta x^\beta$$

is an ST-polynomial if:

- $\alpha \in V(A)$ are affinely independent and
- every $\beta \in \Delta f$ can be written as convex combination of α: $\beta = \sum_{\alpha \in V(\mathcal{A})} \lambda_\alpha \alpha$, $\lambda_\alpha \geq 0$, $\sum_{\alpha \in V(\mathcal{A})} \lambda_\alpha = 1$.

ST-polynomials are similar to circuit polynomials, except that there are several “inner” terms.



Nonnegativity of ST-Polynomials

$$f(x) = \sum_{\alpha \in V(\mathcal{A})\setminus 0} f_\alpha x^\alpha + f_{\alpha(0)} + \sum_{\beta \in \Delta f} f_\beta x^\beta, \;\; \alpha(0) = 0$$

Write $f(x)$ as a sum of circuit polynomials with unknown vertex coefficients:

$$f(x) = \sum_{\beta \in \Delta f} \left(\sum_{\alpha \in nz(\beta)} a_{\beta,\alpha} x^\alpha + f_\beta x^\beta \right)$$

$f(x)$ is **nonnegative** if for every $(\beta, \alpha) \in \Delta f \times V(\mathcal{A})$ there exists $a_{\beta,\alpha} \geq 0$ such that:

$$|f_\beta| \leq \prod_{\alpha \in nz(\beta)} \left(\frac{a_{\beta,\alpha}}{\lambda_\alpha^{(\beta)}} \right)^{\lambda_\alpha^\beta}.$$

$$f_\alpha \geq \sum_{\beta \in \Delta f} a_{\beta,\alpha}.$$

Geometric Programming

- Monomial function $q: \mathbb{R}^n_{>0} \to \mathbb{R}$: $q(z) = cz_1^{\alpha_1} \cdot \dots \cdot z_n^{\alpha_n}$ $(c > 0)$
- Posynomial function $p: \mathbb{R}^n_{>0} \to \mathbb{R}$: $p(z) = \sum_i c_i z_1^{\alpha_{1,i}} \cdot \dots \cdot z_n^{\alpha_{n,i}}$ $(c_i > 0)$

A **geometric program** has the form:

$$\min p_0(z) \text{ subject to:}$$
$$p_i(z) \leq 1 \text{ for all } 1 \leq i \leq m,$$
$$q_j(z) = 1 \text{ for all } 1 \leq j \leq l,$$

where p_i are posynomials and q_j are monomial functions.

Geometric programs **can be transformed into convex programs**.

Nonnegativity of ST-Polynomials: Reformulation

Rewrite the conditions for nonnegativity of $f(x) - \gamma$ by:

- grouping together everything **not** involving any $a_{\beta,0}$,
- using the left out conditions to formulate a lower bound on $f_0 - \gamma$:

$$\lambda_\alpha^{(\beta)} > 0 \iff a_{\beta,\alpha} > 0 \qquad \leftarrow \text{nonzero coefficients correspond to nonzero } \lambda$$

$$|f_\beta| \leq \prod_{\alpha \in nz(\beta) \setminus 0} \left(\frac{a_{\beta,\alpha}}{\lambda_\alpha^{(\beta)}} \right)^{\lambda_\alpha^{(\beta)}}, \ \forall \beta \text{ with } \lambda_0^\beta = 0 \qquad \leftarrow \text{circuit number condition}$$

$$f_\alpha \geq \sum_{\beta \in \Delta f} a_{\beta,\alpha}, \ \forall \alpha \in V(\mathcal{A}) \setminus 0 \qquad \leftarrow \text{cannot exceed original value of coefficients}$$

$$f_0 - \gamma \geq \sum_{\beta \in \Delta f, 0 \in nz(\beta)} \lambda_0^{(\beta)} |f_\beta|^{1/\lambda_0^{(\beta)}} \prod_{\alpha \in nz(\beta) \setminus 0} \left(\frac{\lambda_\alpha^{(\beta)}}{a_{\beta,\alpha}} \right)^{\lambda_\alpha^{(\beta)}/\lambda_0^{(\beta)}} \qquad \leftarrow \text{above two conditions for } \alpha = 0$$

Finding Lower Bounds on ST-Polynomials

Optimisation problem directly built upon reformulated nonnegativity conditions:

$$\min \sum_{\beta \in \Delta f, 0 \in nz(\beta)} \lambda_0^{(\beta)} |f_\beta|^{1/\lambda_0^{(\beta)}} \prod_{\alpha \in nz(\beta) \setminus 0} \left(\frac{\lambda_\alpha^{(\beta)}}{a_{\beta,\alpha}} \right)^{\lambda_\alpha^{(\beta)}/\lambda_0^{(\beta)}}$$

subject to

$$|f_\beta| \leq \prod_{\alpha \in nz(\beta) \setminus 0} \left(\frac{a_{\beta,\alpha}}{\lambda_\alpha^{(\beta)}} \right)^{\lambda_\alpha^{(\beta)}}, \ \forall \beta \text{ with } \lambda_0^\beta = 0$$

$$f_\alpha \geq \sum_{\beta \in \Delta f} a_{\beta,\alpha}, \ \forall \alpha \in V(\mathcal{A}) \setminus 0$$

This program is a GP, i.e. can be directly transformed into a convex program!

Constrained Optimisation

Constrained polynomial optimisation problem:

$$
\begin{aligned}
& min\ f(x) \\
& \text{s.t. } g_i(x) \geq 0\ \forall i = 1, \ldots, m, \\
& x \in \mathbb{R}^n,
\end{aligned}
$$

where f, g_i are polynomials.

Constrained Optimisation: Lagrangian Approach

$$G(x,\mu) = -\sum_{i=0}^{m} \mu_i g_i(x),$$

where $g_0(x) = -f(x)$.
Separate into monomial squares and tail terms:

$$G(x,\mu) = \sum_{\alpha \in V(G)} G_{\alpha,\mu} x^{\alpha} + \sum_{\beta \in \Delta G} G_{\beta,\mu} x^{\beta}$$

Support depends on μ. We will assume it to be the union of exponents existing for any $\mu \geq 0$.

Goal: find $\max_{\mu} \min_{x} G(x,\mu)$.

Lagrangian Approach: Problem Formulation

Optimisation problem over μ and a simultaneously:

$$\min \sum_{i=1}^{m} \mu_i g_{i,\alpha(0)} + \sum_{\beta \in \Delta G, 0 \in nz(\beta)} \lambda_0^{(\beta)} |G_{\beta,\mu}|^{1/\lambda_0^{(\beta)}} \prod_{\alpha \in nz(\beta) \setminus 0} \left(\frac{\lambda_\alpha^{(\beta)}}{a_{\beta,\alpha}} \right)^{\lambda_\alpha^{(\beta)} / \lambda_0^{(\beta)}}$$

subject to

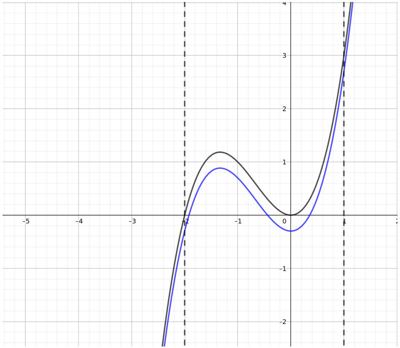
$$|G_{\beta,\mu}| \leq \prod_{\alpha \in nz(\beta) \setminus 0} \left(\frac{a_{\beta,\alpha}}{\lambda_\alpha^{(\beta)}} \right)^{\lambda_\alpha^{(\beta)}}, \ \forall \beta \text{ with } \lambda_0^{\beta} = 0$$

$$G_{\alpha,\mu} \geq \sum_{\beta \in \Delta G} a_{\beta,\alpha}, \ \forall \alpha \in V(G) \setminus 0$$

Can be relaxed into a **geometric program**, which can be transformed into a **convex program**.

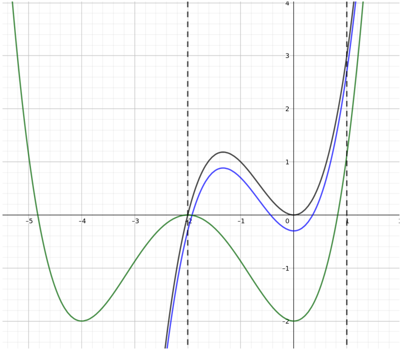
How to Utilise Variable Bounds?

- **Optimisation problem**:
  min $x^3 + 2x^2$ s.t. $-2 \leq x \leq 1$
- **Optimal value** is 0
- $G(x, \mu) = x^3 + 2x^2 - \mu_1(x+2) - \mu_2(1-x)$
- $G(x, \mu)$ **is unbounded** for all $\mu_1, \mu_2 \geq 0$
- The issue is that **variable bounds only add tail terms** to the Lagrangian



Reformulated Bound Constraints

- **Relaxation of the optimisation problem**: min $x^3 + 2x^2$ s.t. $x^4 \leq 32$
- **Optimal value** is 0
- $G(x, \mu) = x^3 + 2x^2 - \mu(16 - x^4)$
- **Lower bound:** $\max_\mu \min_x G(x, \mu) = -2$
- A **SONC certificate can be obtained** proving the lower bound of -2



Choosing Reformulations

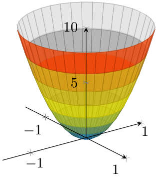
Reformulated constraints: $x_i^{\hat{\alpha}} \leq \max\{|l_i|, |u_i|\}^{\hat{\alpha}}$, $\hat{\alpha} \in \hat{\mathcal{A}}$

- **Requirement on $\hat{\mathcal{A}}$**: each $\beta \in \Delta G$ must be representable as a convex combination of exponents from $\mathcal{A} \cup \hat{\mathcal{A}}$
- Let $\hat{\mathcal{A}} = \{\hat{\alpha}^{(j)} \mid \hat{\alpha}_i^{(j)} = 0$ if $i \neq j$ and $\hat{\alpha}_i^{(j)} = (n + (n \mod 2)) \cdot \max_{\beta \in \Delta G} \beta, \ \forall j = 1, \ldots, n\}$
- Choose exponents to add such that the resulting polynomial is an ST-polynomial

MINLPs in SCIP

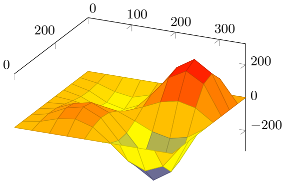
$$\begin{aligned} &\min\ c^T x \\ &\text{s.t. } g_k(x,z) \le 0 \quad && \forall k \in \mathcal{C} \\ &\quad x_i \in [\ell_i, u_i] && \forall i \in \mathcal{I} \\ &\quad z_j \in \mathbb{Z} && \forall j \in \mathcal{J} \end{aligned}$$

- The functions $g_k : [\ell, u] \to \mathbb{R}$ can be



convex

or



nonconvex

and are given in **algebraic form**.

- SCIP solves MINLPs by **spatial Branch & Bound**.

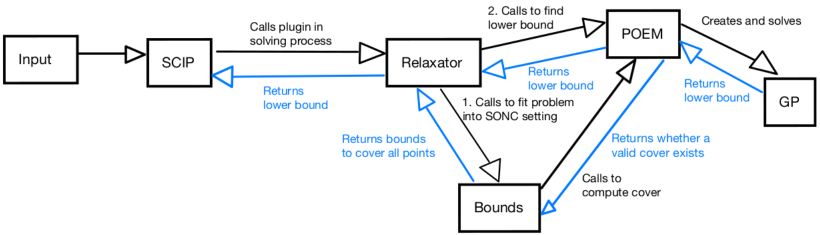
Relaxation Handler Plugins in SCIP

- By default, SCIP solves LP relaxations
- Custom relaxations can be implemented in **relaxation handler** plugins (**relaxators**)
- One fundamental callback: **RELAXEXEC**, which can:
  - Provide a dual bound or detect infeasibility of the node
  - Provide primal solution candidates
  - Reduce variable domains
  - Add branching candidates and cutting planes

Algorithm: Key Steps

- Revert the moving of nonlinear terms from the objective to constraints (otherwise, the introduced objective variable is a term in $\Delta(f)$ without cover)
- Construct the Lagrangian $G(x, \mu)$
- Compute $V(\mathcal{A})$, simplex vertices/simplex cover, add polynomial bound constraints
- Add bounds on Lagrangian multipliers to ensure nonnegativity of vertex coefficients
- Create and solve the GP
- Retrieve the dual bound

Implementation with SCIP and POEM



- Implemented as a relaxator plugin, called during SCIP's branch and bound
- GPs are constructed by POEM, software for polynomial optimization via SONC certificates

Computational results: Existence of a Solution

- Testset: 180 instances randomly generated by Dalkiran and Sherali, 169 instances from MINLPLib
- Time limit 1 hour

Test Run	*Solution Status of the Relaxation*		
	optimal	*decomposition not found*	*did not run/timeout*
Standard SONC	9	333	7
Strengthened SONC	330	0	19

Table: Comparison of the relaxator statuses at root node.

Computational results: Bound Quality

- SONC bound improved upon LP bound only on 8 instances
- We show 6 instances where the SONC root node bound was used

Instance	*LP bound*	*SONC bound*	*Primal bound*	*Gap closed*
ex4_1_1	-385.31	-97.88	0.1	74.5%
ex4_1_4	-3125	-27	0	89.5%
ex4_1_6	-256	-250	250	85.2%
ex4_1_7	-166.64	-44.17	0	73.5%
mathopt5_8	-41.14	-8.24	0	80.0%
waterund01	-307.56	-262.89	∞	-

Table: Dual bounds at the root node

Summary and Outlook

- We developed a method that guarantees a finite SONC bound if all variables are bounded
- A considerable improvement is observed on few instances, but LP bound is stronger on remaining instances
- The computational cost of SONC relaxations is high compared to LP relaxations

Future work:

- Further develop methods for utilizing variable bounds to strengthen SONC relaxations
- Develop branching rules, presolving methods, etc.
- Can a SONC-based branch and bound algorithm be proven to converge?
- Interesting polynomial optimization problems? (high degrees, non-trivial instance sizes)