Separating Hyperplanes for Mixed-Integer Polynomial Optimization Problems

Carl Eggen*† Jan Kronqvist‡ Andreas Lundell§
Stefan Volkwein*

February 25, 2026

Abstract

Algorithms based on polyhedral outer approximations provide a powerful approach to solving mixed-integer nonlinear optimization problems. An initial relaxation of the feasible set is strengthened by iteratively adding linear inequalities and separating infeasible points. However, when the constraints are nonconvex, computing such separating hyperplanes becomes challenging. In this article, the moment-/sums-of-squares hierarchy is used in the case that the objective and the constraints are polynomial. Although the calculated hyperplanes are not optimal in general, convergence to an optimal separating hyperplane can be shown under some general conditions. Then, this idea is embedded in an algorithmic framework for mixed-integer nonlinear problems. The algorithm is modified with two heuristic approaches inspired by sparsity structures to tackle instances with a high number of variables and constraints. A numerical study on a large set of problems from the MINLPLib demonstrates the potential of our method.

Keywords. Mixed-integer nonlinear programming, polyhedral outer approximation, polynomial optimization, semidefinite programming

1 Introduction

Mixed-integer nonlinear minimization problems are optimization problems that involve both continuous and discrete variables as well as linear and nonlinear constraints. Due to their inherent complexity, solving mixed integer problems

*Department of Mathematics and Statistics, University of Konstanz, Konstanz, Germany, {carl.eggen, stefan.volkwein}@uni-konstanz.de

†corresponding author

‡Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden, jankr@kth.se

§Information Technology, Åbo Akademi University, Vaasa, Finland, andreas.lundell@abo.fi

to global optimality can be computationally very challenging, sometimes even intractable [Jer73; KM78].

One fundamental concept for solving such problems is the relaxation (i.e., enlargement) of the feasible set to obtain lower bounds on the optimal value [Bel+13]. These relaxations should be much easier to solve than the original problem. To strengthen an initial relaxation, solution methods rely on iteratively adding linear inequalities. Some of these methods are called outer approximation [DG86], extended cutting plane method [WP95] or extended supporting hyperplane algorithm [KLW16]. As long as the solution of the relaxation is infeasible for the original problem, an attempt is made to compute a new linear inequality that excludes the infeasible solution, in the following also referred to as a separating hyperplane. This attempt may fail if the feasible set is nonconvex. Nevertheless, it is still possible to obtain tight lower bounds on the optimal value of the original problem, if the objective is linear, which can be always ensured by an epigraph reformulation. This is shown by the following basic proposition (e.g., [KBG25, Theorem 2]).

Proposition 1.1. *Let $M \subseteq \mathbb{R}^n$. Then the optimal value of a linear function f over M is equal to the optimal value of f over the convex hull* $\operatorname{conv}(M)$ *of M.*

Therefore, our aim is to approximate the convex hull of the feasible set using hyperplanes. If M is already a convex set, those hyperplanes can be calculated by using the first-order Taylor series expansion of the constraints as in [DG86; WP95; KLW16], but the same method cannot be directly transferred to the nonconvex case. For more details on such algorithms we also refer to [Bel+13; KBG25].

However, if the objective and the constraints are all polynomials, which we then call a mixed-integer polynomial optimization problem (MIPOP), the moment-/sums-of-squares-hierarchy from polynomial optimization introduced by Lasserre in [Las01] can be applied. It relies on sums-of-squares (SOS) certificates for nonnegative polynomials and the dual theory of moments. For a self-contained introduction and for an overview about the topic, the reader is referred to [Las09; Las15; Lau09; Nie23]. If some mild conditions hold, the moment-/SOS-hierarchy can be used to approximate a polynomial optimization problem arbitrarily well. In practice, solving this hierarchy of approximations may be rather computationally expensive, since each problem in the hierarchy is formulated as a semidefinite program (SDP) and the sizes of the involved semidefinite matrices grow rapidly for problems further up the hierarchy. In the case that some sparsity structure appears, the issue of growing complexity can be partially tackled and even instances with several thousand variables and constraints can be solved [Wak+06; WML21b; WML21a; WML22; Wan+22; MW23].

There have been some approaches in the literature to calculate valid linear inequalities for polynomial optimization problems by using the moment-/SOS-hierarchy. The first one is in [GVA16]. Therein, valid inequalities are generated to improve an SOS relaxation by solving higher-order SOS relaxations. Secondly in [BCM20], S-free sets and intersection cuts are calculated in a lifted space. Finally in [BS20], linear inequalities close to points in the feasible set are generated.

Our approach is a mixture of the ideas in [GVA16] and [BS20]. But instead of adding linear inequalities to the SOS relaxation, we add them to relaxations that are used in standard mixed-integer optimization algorithms, which usually means (mixed-integer) linear relaxations. Further, we do not need a feasible point to calculate the linear equalities, but we use infeasible points generated by relaxations and cut them off by separating hyperplanes. This approach makes the moment-/SOS-hierarchy accessible to common mixed-integer nonlinear programming (MINLP) solvers, what we illustrate by an integration into the `SHOT` solver [LKW22].

The outline of the article is as follows: Firstly, the definitions and notations for MIPOPs, separating hyperplanes and the moment-/SOS-hierarchy are introduced in Section 2. Then we show how separating hyperplanes can be calculated by a modification of the original moment-/SOS-hierarchy (cf. Section 3). Although the calculated hyperplanes are not optimal in general, convergence to an optimal separating hyperplane can be shown (cf. Theorem 3.11). Under some convexity assumptions, optimal hyperplanes can already be generated in the first step of the modified hierarchy (cf. Section 4). In Section 5, an algorithmic framework is presented illustrated by an example. The algorithm is then modified with two heuristic approaches inspired by sparsity structures to tackle larger instances in Section 6. Finally, the performance of the sparse algorithm is shown on a large set of examples from the MINLPLib [MIN25] by an integration into the `SHOT` solver (cf. Section 7).

2 Preliminaries

2.1 Definitions and Notation

Let $\mathbb{R}[x]$ denote the polynomial ring in n variables $x = (x_1, \dots, x_n)$ for some $n \in \mathbb{N}$. Further, let $r \in \mathbb{N}_0$. Then, $\mathbb{R}[x]_r$ defines the vector space of real polynomials up to degree r in the variables x.

Throughout the article, we consider *mixed-integer polynomial optimization problems* of the form

$$\begin{aligned} f_{\min} = \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t} \quad & g_j(x) \geq 0 \quad \text{for } j = 1, \dots, k, \\ & x^l \leq x \leq x^u, \\ & x_i \in \mathbb{Z} \quad \text{for } i \in I, \end{aligned} \tag{MIPOP}$$

where the objective f and the constraints $g_1, \dots, g_k$ are polynomials in $\mathbb{R}[x]$. Without loss of generality, we assume that f is a linear polynomial with $f(0) = 0$. All variables are bounded by lower bounds $x^l \in \mathbb{R}^n$ and upper bounds $x^u \in \mathbb{R}^n$. The set $I \subseteq \{1, \dots n\}$ contains all indices of variables that attain only integer values. Furthermore, let M denote the feasible set of (MIPOP).

Note that there is an ambiguity of the adjective *linear*. Although we refer to a polynomial as linear, which means that its degree equals 1, the corresponding polynomial function is only affine linear.

Next, we introduce the notion of a relaxation for (MIPOP).

Definition 2.1. Let M be the feasible set of (MIPOP) and $\Omega \subseteq \mathbb{R}^n$ with $\Omega \supseteq M$. Then

$$f_{\min}^{\text{relax}} = \min_{x \in \Omega} f(x)$$

is called a *relaxation* of (MIPOP) with *relaxed set* Ω.

We refer to the *continuous relaxation* of (MIPOP), when all integer constraints are omitted, as

$$\begin{aligned} f_{\min}^{\text{POP}} = \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_j(x) \geq 0 \quad \text{for } j = 1, \ldots, k, \\ & x^l \leq x \leq x^u \end{aligned} \tag{POP}$$

with the relaxed feasible set

$$S = \left\{ x \in \mathbb{R}^n \,\middle|\, g_1(x) \geq 0, \ldots, g_k(x) \geq 0 \text{ and } x^l \leq x \leq x^u \right\}. \tag{1}$$

Moreover, we consider the initial *mixed-integer linear relaxation* of (MIPOP)

$$\begin{aligned} f_{\min}^{\text{MILP}} = \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & l_j(x) \geq 0 \quad \text{for } j = 1, \ldots, k_{\text{lin}}, \\ & x^l \leq x \leq x^u, \\ & x_i \in \mathbb{Z} \quad \text{for } i \in I \end{aligned} \tag{MILP}$$

with $k_{\text{lin}} \in \mathbb{N}_0$ linear constraints l_j. The linear constraints can be easily obtained by taking only the linear constraints from (MIPOP). Alternatively, by introducing additional variables, a tighter linear relaxation can be derived by using McCormick relaxations [McC76], for example.

2.2 Separating Hyperplanes

We recall some fundamental concepts from the theory of separating hyperplanes, largely based on [Sch14, Section 1.3]. We first start by defining hyperplanes and their associated halfspaces in the real vector space $\mathbb{R}^n$.

Definition 2.2. Let $w \in \mathbb{R}^n \setminus \{0\}$ and $b \in \mathbb{R}$. Then the set $H = \{x \in \mathbb{R}^n \mid w^\top x = b\}$ is called a *hyperplane* in $\mathbb{R}^n$ and bounds the two closed *halfspaces*

$$H^- = \{x \in \mathbb{R}^n \mid w^\top x \leq b\} \quad \text{and} \quad H^+ = \{x \in \mathbb{R}^n \mid w^\top x \geq b\}.$$

Remark 2.3. Let H and H' be the hyperplanes defined by the two vectors $(w, b), (u, c) \in \mathbb{R}^n \backslash \{0\} \times \mathbb{R}$. Then $H = H'$ if and only if $(w, b) = (\lambda u, \lambda c)$ for some $0 \neq \lambda \in \mathbb{R}$. Hence, the vector w in Definition 2.2 can be restricted to the unit ball ($\|w\| \leq 1$) or even the unit sphere ($\|w\| = 1$).

A hyperplane H can be represented by the zero set of the linear (i.e., degree-one) polynomial $h \in \mathbb{R}[x]_1$ defined by $h(x) = w^\top x - b$, that means $H = \{x \in \mathbb{R}^n \mid h(x) = 0\}$. Hence, all hyperplanes can be represented by linear polynomials and vice versa. If $h \in \mathbb{R}[x]_1$ is a linear polynomial, we denote by $w_h \in \mathbb{R}^n \backslash \{0\}$ the coefficients of the variables and by $b_h \in \mathbb{R}$ the constant coefficient. Using this notion, we introduce separating hyperplanes and further state a separation theorem.

Definition 2.4. Let $h \in \mathbb{R}[x]_1$ represent a hyperplane. Then h *(strictly) separates* two sets $A, B \subseteq \mathbb{R}^n$, if $h \leq 0$ ($h < 0$) on A and $h \geq 0$ ($h > 0$) on B, or vice versa. The sets A and B are *strongly separated* by h, if there exists $\varepsilon > 0$ such that $h \leq -\varepsilon$ on A and $h \geq \varepsilon$ on B or vice versa.

Theorem 2.5 ([Sch14, Theorem 1.3.7])**.** *Let $A, B \subseteq \mathbb{R}^n$ be nonempty and convex with $A \cap B = \emptyset$. Then A and B can be separated. If A is compact and B is closed, then A and B can be strongly separated.*

In the following, we usually want to separate a single point $\bar{x}$ from a closed set B, which means $A = \{\bar{x}\}$ in Theorem 2.5. If a separating hyperplane exists, we are interested in a hyperplane that lies as far away as possible from $\bar{x}$. This purpose leads to the definition of the distance between a point and a hyperplane.

Definition 2.6. Let $H \subseteq \mathbb{R}^n$ be a hyperplane, $h \in \mathbb{R}[x]_1$ its polynomial representation and $\bar{x} \in \mathbb{R}^n$. Then the *distance* between $\bar{x}$ and H is defined by

$$\operatorname{dist}(\bar{x}, H) \coloneqq \inf_{x \in H} \|\bar{x} - x\|$$

for some norm $\|\cdot\|$ in $\mathbb{R}^n$. If a specific norm (e.g., 2-norm) is used, then this is indicated by a subscript (e.g., $\operatorname{dist}_2(\bar{x}, H)$). In the following, we usually replace the hyperplane H by its polynomial representation h in the definition, that means $\operatorname{dist}(\bar{x}, h) \coloneqq \operatorname{dist}(\bar{x}, H)$.

The distance can be directly calculated using the dual norm and evaluating the polynomial representation.

Lemma 2.7. *Let $h \in \mathbb{R}[x]_1$ be a linear polynomial and $\bar{x} \in \mathbb{R}^n$. Then*

$$\operatorname{dist}(\bar{x}, h) = \frac{|h(\bar{x})|}{\|w_h\|_*},$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

Proof. From the duality of the norms, we know that h is Lipschitz continuous with Lipschitz constant $\|w\|_*$, that means

$$|h(x) - h(y)| = |w_h^\top (x - y)| \leq \|w_h\|_* \|x - y\|.$$

for all $x, y \in \mathbb{R}^n$.
"$\geq$": Let $y \in \mathbb{R}^n$ with $h(y) = 0$. Then

$$\frac{|h(\bar{x})|}{\|w_h\|_*} = \frac{1}{\|w_h\|_*}|h(\bar{x}) - h(y)| \leq \|\bar{x} - y\|$$

using the Lipschitz continuity of h. Since y was arbitrarily chosen, this holds for all $y \in \mathbb{R}^n$ with $h(y) = 0$ and thus

$$\frac{|h(\bar{x})|}{\|w_h\|_*} \leq \inf\left\{\|\bar{x} - y\| \;\middle|\; y \in \mathbb{R}^n \text{ and } h(y) = 0\right\} = \operatorname{dist}(\bar{x}, h).$$

"$\leq$": By definition of the dual norm as well as compactness and symmetry of the closed unit ball, there exists $y \in \mathbb{R}^n$ with $\|y\| = 1$ such that $w_h^\top y = \|w_h\|_*$. Thus, from $h \in \mathbb{R}[x]_1$ we infer that

$$h\big(\bar{x} - \tfrac{h(\bar{x})}{\|w_h\|_*}\, y\big) = w_h^\top \bar{x} - \tfrac{h(\bar{x})}{\|w_h\|_*}\, w_h^\top y - b = h(\bar{x}) - h(\bar{x}) = 0,$$

so that $\bar{x} - ({h(\bar{x})}/{\|w_h\|_*})\, y$ is a point on the hyperplane. Now, we can conclude

$$\operatorname{dist}(\bar{x}, h) \leq \big\|\bar{x} - (\bar{x} - \tfrac{h(\bar{x})}{\|w_h\|_*}\, y)\big\| = \tfrac{|h(\bar{x})|}{\|w_h\|_*}\|y\| = \tfrac{|h(\bar{x})|}{\|w_h\|_*}$$

which gives the claim. □

2.3 The moment-/SOS-hierarchy

To formulate the moment-/sums-of-squares-hierarchy from [Las01] (cf. [Lau09]), we introduce some algebraic definitions.

Let $\Sigma[x]$ be the convex cone of sums of squares of real polynomials and $\Sigma[x]_r$ the subcone of sums of squares (SOS) of real polynomials up to degree $2r$ for some $r \in \mathbb{N}$.

$$\Sigma[x] := \left\{ \sum_{i=1}^{k} q_i^2 \;\middle|\; k \in \mathbb{N} \text{ and } q_i \in \mathbb{R}[x] \right\},$$
$$\Sigma[x]_r := \left\{ \sum_{i=1}^{k} q_i^2 \;\middle|\; k \in \mathbb{N} \text{ and } q_i \in \mathbb{R}[x]_r \right\}.$$

A polynomial in $\Sigma[x]$ is called SOS polynomial. The *quadratic module* generated by polynomials $p_1, \ldots, p_k \in \mathbb{R}[x]$ is defined by

$$Q(p_1, \ldots, p_k) := \left\{ \sigma_0 + \sum_{j=1}^{k} \sigma_j p_j \;\middle|\; \sigma_0, \sigma_j \in \Sigma[x] \right\} \subseteq \mathbb{R}[x]$$

and the corresponding *r-truncated quadratic module* by

$$Q_r(p_1, \ldots, p_k) := \left\{ \sum_{j=0}^{k} \sigma_j p_j \mid \sigma_j \in \Sigma[x], \deg(\sigma_j g_j) \leq 2r \right\} \subseteq \mathbb{R}[x]_{2r}$$

with $p_0 := 1$. For later proofs, we need the following basic result about the degree of SOS polynomials.

Lemma 2.8 ([Lau09, Lemma 3.1])**.** *If $p \in \mathbb{R}[x]$ is an* SOS *polynomial, then* $\deg(p)$ *is even and any decomposition $p = \sum_{j=1}^{m} \sigma_j^2$ with $\sigma_j \in \mathbb{R}[x]$ satisfies* $\deg(\sigma_j) \leq \deg(p)/2$ *for all j.*

Furthermore, let $\zeta := (\zeta_\alpha)_{\alpha \in \mathbb{N}_0^n} \subseteq \mathbb{R}$ be a sequence indexed by the monomial basis $(x^\alpha)_{\alpha \in \mathbb{N}_0^n}$ and $L_\zeta : \mathbb{R}[x] \to \mathbb{R}$ the associated Riesz functional given as

$$f = \sum_\alpha f_\alpha x^\alpha \mapsto L_\zeta(f) = \sum_\alpha f_\alpha \zeta_\alpha.$$

Then the *moment matrix* $\mathcal{M}(\zeta)$ is the (infinite) matrix defined by

$$\mathcal{M}(\zeta)_{\alpha,\beta} := L_\zeta(x^{\alpha+\beta}) = \zeta_{\alpha+\beta}, \quad \alpha, \beta \in \mathbb{N}_0^n$$

with columns and rows indexed by the monomial basis $(x^\alpha)_{\alpha \in \mathbb{N}_0^n}$. Similarly, given $r \in \mathbb{N}$, the *r-truncated moment matrix* $\mathcal{M}_r(\zeta)$ is defined by

$$\mathcal{M}_r(\zeta)_{\alpha,\beta} := L_\zeta(x^{\alpha+\beta}) = \zeta_{\alpha+\beta}, \quad \alpha, \beta \in \mathbb{N}_{0,r}^n$$

with columns and rows indexed by the monomial basis $(x^\alpha)_{\alpha \in \mathbb{N}_{0,r}^n}$ and $\mathbb{N}_{0,r}^n := \{\alpha \in \mathbb{N}_0^n \mid |\alpha| \leq r\}$ ($|\alpha| = \sum_i \alpha_i$). For $g = \sum_\gamma g_\gamma x^\gamma \in \mathbb{R}[x]$ we define the *localizing matrix*

$$\mathcal{M}(g\zeta)_{\alpha,\beta} := L_\zeta(g(x)x^{\alpha+\beta}) = \sum_\gamma g_\gamma \zeta_{\alpha+\beta+\gamma} \quad \text{for } \alpha, \beta \in \mathbb{N}_0^n$$

and its *r-truncation*

$$\mathcal{M}_r(g\zeta)_{\alpha,\beta} := L_\zeta(g(x)x^{\alpha+\beta}) = \sum_\gamma g_\gamma \zeta_{\alpha+\beta+\gamma} \quad \text{for } \alpha, \beta \in \mathbb{N}_{0,r}^n.$$

For the truncated moment and localizing matrices we can use the following calculation rules.

Lemma 2.9 ([Lau09, Lemma 4.1])**.** *Let $\zeta \in \mathbb{R}^{\mathbb{N}_0^n}$, L_ζ the associated linear functional and let $p, q, r \in \mathbb{R}[x]$. Then*

$$\begin{aligned} L_\zeta(pq) &= \operatorname{vec}(p)^\top \mathcal{M}(y) \operatorname{vec}(q) \text{ and} \\ L_\zeta(pqr) &= \operatorname{vec}(p)^\top \mathcal{M}(y) \operatorname{vec}(qr) = \operatorname{vec}(pq)^\top \mathcal{M}(y) \operatorname{vec}(r) \\ &= \operatorname{vec}(p)^\top \mathcal{M}(qy) \operatorname{vec}(r), \end{aligned}$$

where $\operatorname{vec}(\cdot)$ *yields the vector of coefficients of a polynomial.*

3 Separating Hyperplanes for Polynomial Optimization Problems

We first introduce our separating hyperplane method for the continuous relaxation (POP) of (MIPOP), that means the integer conditions on the variables are ignored. This does not automatically mean that variables cannot be set to integer values. In the case that the integer conditions can be expressed by polynomial constraints, these polynomial constraints can be added to the description of the relaxed feasible set S. For performance reasons those polynomials should be of low degree. This is the case for binary variables (or integer variables that behave like binary variables). If x_i is a binary variable, the two polynomial inequalities $\pm(x_i^2 - x_i) \geq 0$ can be added to the constraints. Further, if some variables are integer, but nonbinary, the cuts calculated by $(\mathrm{SOS}_r^{\mathrm{HYP}})$ may be refined by rounding methods. Instead of explicitly stating results in this direction, we will leave the rounding to the MILP solvers.

Let the relaxed feasible set S of (POP) be nonempty and $\bar{x}$ a solution of a relaxation, for example a solution of the mixed-integer linear relaxation (MILP), with $f(\bar{x}) < f_{\min}$. Then, $\bar{x}$ is not in the set S. Moreover, from Proposition 1.1 it follows that $\bar{x} \notin \mathrm{conv}(S)$. Since the convex hull of compact sets in $\mathbb{R}^n$ is compact again (cf. [Rud73, Theorem 3.25 b)]), Theorem 2.5 can be applied to $\mathrm{conv}(S)$ and $\{\bar{x}\}$. Thus, $\bar{x}$ and $\mathrm{conv}(S)$ can be strongly separated, that means there exists a linear polynomial $h \in \mathbb{R}[x]_1$ such that $h(\bar{x}) \geq \varepsilon$ for some $\varepsilon > 0$ and $h \leq 0$ on $\mathrm{conv}(S)$.

Finding such a hyperplane can be formulated as the infinite-dimensional linear optimization problem

$$\begin{aligned} \sup_{h \in \mathbb{R}[x]_1} \quad & h(\bar{x}) \\ \text{s.t.} \quad & h(x) \leq 0 \quad \text{for all } x \in S, \\ & \|w_h\|_\infty \leq 1. \end{aligned} \tag{HYP}$$

For implementation reasons, the maximum norm is used for bounding the vector w_h, since this norm constraint results in linear constraints. Other norms (e.g., 2-norm) can also be used as long as there are optimization methods that can efficiently handle them. We now provide some simple observations about (HYP).

Remark 3.1. 1) To ensure feasibility, $w_h = 0$ is allowed in (HYP), although we are only interested in linear polynomials (i.e., $w_h \neq 0$). If $w_h = 0$, h must be a nonpositive constant, since S is nonempty. If h is an optimal solution with $w_h = 0$, h must be the zero polynomial. Conversely, if h is an optimal solution and $h \neq 0$, then h must be a linear polynomial.

2) Quantifying over the set S or its convex hull results in an equivalent optimization problem: Let h be a linear polynomial. Then $h \leq 0$ on S, if and only if $h \leq 0$ on $\mathrm{conv}(S)$, since the closed halfspace of a hyperplane is convex.

3) If $\bar{x} \notin \mathrm{conv}(S)$, then (HYP) has always a feasible linear polynomial with positive optimal value by Theorem 2.5. If the optimal value of (HYP) is 0 (i.e., $h = 0$ is an optimal solution), then $\bar{x}$ cannot be strongly separated from $\mathrm{conv}\, S$, that is $\bar{x} \in \mathrm{conv}\, S$.

4) Bounding the coefficients w_h of h in the last line of (HYP) is not a significant restriction: Let h be a linear polynomial with $h \leq 0$ on S, then $\|w_h\|_\infty^{-1} h$ is feasible for (HYP) with $\|w_{\|w_h\|_\infty^{-1} h}\|_\infty = 1$ (see also Remark 2.3). Note that $w_h \neq 0$, since h is a linear polynomial.

5) If a linear polynomial h is feasible for (HYP), then also $\|w_h\|_\infty^{-1} h$ is feasible with $h(\bar{x}) \leq \|w_h\|_\infty^{-1} h(\bar{x})$. So, if h is even an optimal solution, then we have $\|w_h\|_\infty = 1$.

6) The objective gives also a measure for the distance between the calculated hyperplane h and the point $\bar{x}$. By Lemma 2.7, if h is a linear polynomial, then $h(\bar{x}) = {}^{\mathrm{dist}_1(\bar{x},h)}/_{\|w_h\|_\infty}$. If h is an optimal solution and $h \neq 0$, then $h(\bar{x}) = \mathrm{dist}_1(\bar{x}, h)$.

As the feasible set S is assumed to be nonempty, (HYP) yields a separating hyperplane, which is as far away as possible from $\bar{x}$. Since the maximum norm is used in (HYP), the distance is measured in the dual 1-norm.

Proposition 3.2. *Let $\bar{x} \in \mathbb{R}^n$. Then there exists an optimal solution $h \in \mathbb{R}[x]_1$ for* (HYP) *with optimal value*

$$h(\bar{x}) = \mathrm{dist}_1(\bar{x}, \operatorname{conv} S).$$

Proof. Let $\tilde{x} \in \operatorname{conv} S$ and $h \in \mathbb{R}[x]_1$ feasible for (HYP) with $h \neq 0$ and $h(\bar{x}) \geq 0$. Since $h(\tilde{x}) \leq 0$ holds by Remark 3.1-2), there exists $\lambda \in [0,1]$ such that $x = \lambda \bar{x} + (1-\lambda)\tilde{x}$ and $h(x) = 0$ by the Intermediate Value Theorem. Then

$$\mathrm{dist}_1(\bar{x}, h) \leq \|\bar{x} - x\|_1 = \|\bar{x} - (\lambda \bar{x} + (1-\lambda)\tilde{x})\|_1 = (1-\lambda)\|\bar{x} - \tilde{x}\|_1 \leq \|\bar{x} - \tilde{x}\|_1$$

holds. Since $\tilde{x} \in \operatorname{conv} S$ is arbitrarily chosen and one can assume $\|w_h\|_\infty = 1$ without loss of generality (cf. Remark 3.1-5)),

$$0 \leq h(\bar{x}) = \mathrm{dist}_1(\bar{x}, h) \leq \mathrm{dist}_1(\bar{x}, \operatorname{conv} S)$$

holds by Lemma 2.7.

Let $\mathfrak{d} := \mathrm{dist}_1(\bar{x}, \operatorname{conv} S)$. If $\bar{x} \in \operatorname{conv} S$, then $h = 0$ is an optimal solution. So assume $\bar{x} \notin \operatorname{conv} S$ and define

$$B_1(\bar{x}, \mathfrak{d}) := \{z \in \mathbb{R}^n \mid \|\bar{x} - z\|_1 \leq \mathfrak{d}\}.$$

Then $\mathfrak{d} > 0$ holds and the interior of $B_1(\bar{x}, \mathfrak{d})$ can be separated from $\operatorname{conv}(S)$ by Theorem 2.5. Thus, there exists a linear polynomial $h \in \mathbb{R}[x]_1$ such that $h \leq 0$ on $\operatorname{conv} S$ and $h \geq 0$ on $B_1(\bar{x}, \mathfrak{d})$. Scaling of h yields $\|w_h\|_\infty = 1$. Now, h is feasible for (HYP) with

$$h(\bar{x}) = \mathrm{dist}_1(\bar{x}, h) \geq \mathfrak{d} = \mathrm{dist}_1(\bar{x}, \operatorname{conv} S) > 0.$$

Together with the first part, $h(\bar{x}) = \mathrm{dist}_1(\bar{x}, h) = \mathrm{dist}_1(\bar{x}, \operatorname{conv} S)$ follows and h is an optimal solution. □

Proposition 3.3. *Let $\bar{x} \in \mathbb{R}^n$ hold and $h \neq 0$ be an optimal solution of* (HYP)*. Then h is an optimal separating hyperplane with respect to the distance in the 1-norm, that means*

$$\operatorname{dist}_1(\bar{x}, h') \leq \operatorname{dist}_1(\bar{x}, h)$$

for all $h' \neq 0$ feasible for (HYP) *separating $\bar{x}$ from S (i.e., $h'(\bar{x}) \geq 0$).*

Proof. Let h' be feasible for (HYP) with $h'(\bar{x}) \geq 0$. Then also $h'' = \|w_{h'}\|_\infty^{-1} h'$ is feasible by Remark 3.1-5). Since $h''(\bar{x}) \leq h(\bar{x})$ and $\|w_h\|_\infty = 1$ (cf. Remark 3.1 (5)), it follows from Lemma 2.7 and $h'(\bar{x}) \geq 0$

$$\operatorname{dist}_1(\bar{x}, h') = \frac{h'(\bar{x})}{\|w_{h'}\|_\infty} = h''(\bar{x}) \leq h(\bar{x}) = \operatorname{dist}_1(\bar{x}, h).$$

which yields the claim. □

Problem (HYP) can have multiple optimal solutions, even uncountably many as the following example shows.

Example 3.4. Let $S = \{(0,1)\} \subseteq \mathbb{R}^2$ and $\bar{x} = (0,0)$. Then $h_a = ax_1 - x_2 + 1$ is an optimal solution of (HYP) for any for $a \in [-1,1]$: Let $a \in [-1,1]$. Then h_a is feasible, since $h_a(0,1) = 0$ and $\|w_{h_a}\|_\infty = 1$. Further, $h_a(\bar{x}) = 1 = \operatorname{dist}_1(\bar{x}, S)$ and thus optimal by Proposition 3.2.

A solution of (HYP) yields the best separating hyperplane in the sense of Proposition 3.3. But solving (HYP) is in general not easy. To find a separating hyperplane, we invoke the moment-/SOS-hierarchy introduced in [Las01] (cf. [Lau09]). Replacing the box constraint $x^l \leq x \leq x^u$ by the quadratic conditions $\varphi_i(x) \geq 0$ with

$$\varphi_i(x) \coloneqq (x_i - x_i^l)(x_i^u - x_i) \in \mathbb{R}[x] \quad \text{for } i = 1, \ldots, n,$$

we gain a polynomial description of the feasible set S of (POP). Notice that we could also use the linear box constraints for describing the feasible set S, but this linear description yields a slightly weaker hierarchy as shown in [ESV25, Lemma 4.5]. On the polynomial side, we can now relax (HYP) to

$$\begin{aligned} \rho_r = \sup_{h \in \mathbb{R}[x]_1} \quad & h(\bar{x}) \\ \text{s.t.} \quad & -h \in Q_r(g_1, \ldots, g_k, \varphi_1, \ldots, \varphi_n), \\ & \|w_h\|_\infty \leq 1, \end{aligned} \tag{$\mathrm{SOS}_r^{\mathrm{HYP}}$}$$

where $r \in \mathbb{N}$ with $r \geq \max_{j=1,\ldots,k}\{\lceil \deg(g_j)/2 \rceil\}$ denotes the order of the relaxation and $Q_r(g_1, \ldots, g_k, \varphi_1, \ldots, \varphi_n)$ is the r-truncated quadratic module generated by the polynomials $g_1, \ldots, g_k, \varphi_1, \ldots, \varphi_n$. In ($\mathrm{SOS}_r^{\mathrm{HYP}}$) the semi-infinite "$\leq$"-condition from (HYP) is relaxed to a sums-of-squares constraint bounded by degree $2r$, yielding a hierarchy of semidefinite programs. As for (HYP), if the solution of ($\mathrm{SOS}_r^{\mathrm{HYP}}$) is not the zero polynomial, then the solution is a hyperplane, which separates $\bar{x}$ from S.

Proposition 3.5. *Let $h \in \mathbb{R}[x]_1$ with $h \neq 0$ be an optimal solution of* $(\text{SOS}_r^{\text{HYP}})$*. Then h separates $\bar{x}$ and S.*

Proof. From $-h \in Q_r(g_1, \ldots, g_k, \varphi_1, \ldots, \varphi_n)$ it follows that $-h \geq 0$ on S and thus $h \leq 0$ on S. Since h is an optimal solution, we have $h(\bar{x}) \geq 0$. ☐

Now, if the polynomial $h \in \mathbb{R}[x]_1$ is an optimal solution of $(\text{SOS}_r^{\text{HYP}})$ with $h(\bar{x}) > 0$, then h is a separating hyperplane by Proposition 3.5 and the linear inequality $h(x) \leq 0$ can be added to (MILP) cutting away the relaxed solution $\bar{x}$.

We can also formulate the dual of (HYP), which is the infinite-dimensional optimization problem

$$\begin{aligned} \inf_{\mu,t} \quad & e^\top t \\ \text{s.t.} \quad & \langle \mu, 1\rangle = 1, \\ & -t_i \leq \langle \mu,\ x_i\rangle - \bar{x}_i \leq t_i \quad \text{for } i = 1, \ldots, n, \\ & \mu \in \mathfrak{M}(S)_+,\ t \in \mathbb{R}^n_{\geq 0}, \end{aligned} \tag{HYP_{dual}}$$

where $\mathfrak{M}(S)_+$ is the convex cone of nonnegative, finite Borel measures on $\mathbb{R}^n$, $e^\top = (1, \ldots, 1) \in \mathbb{R}^n$ and

$$\langle \mu, p\rangle := \int_S p \; d\mu \quad \text{for } p \in \mathbb{R}[x] \text{ on } S.$$

By projecting onto the first moments and using the 1-norm, HYP_{dual} appears in the following simple form:

$$\inf \left\{ \|z - \bar{x}\|_1 \,\middle|\, z \in \operatorname{conv}(S) \right\}. \tag{2}$$

Thus, finding $z \in \operatorname{conv} S$, which is closest to $\bar{x}$, is the dual side of separating $\bar{x}$ from $\operatorname{conv} S$ by a farthest hyperplane. This is due to the well-known relationship between the convex hull and the first moments of probability measures with support in S (cf. [Rud73, Theorem 3.28]):

Proposition 3.6. *Let $S \subseteq \mathbb{R}^n$ be compact. Then*

$$\operatorname{conv}(S) = \left\{ z \in \mathbb{R}^n \,\middle|\, \exists \mu \in \mathfrak{M}(S)_+ : \langle \mu, 1\rangle = 1, \langle \mu, x_i\rangle = z_i \text{ for all } i \in \{1, \ldots, n\} \right\}.$$

Using the Riesz functional as well as the truncated moment and localizing matrices (cf. Section 2.3), we can derive a hierarchy of moment relaxations in the sense of [Las01], which are the duals of the semidefinite problems $(\text{SOS}_r^{\text{HYP}})$:

$$\begin{aligned} \rho_r^* = \inf_{\zeta,t} \quad & e^\top t \\ \text{s.t.} \quad & \zeta_0 = 1,\ \mathcal{M}_r(\zeta) \succeq 0 \\ & \mathcal{M}_{r-\nu_j}(g_j \zeta) \succeq 0 \qquad \text{for } j = 1, \ldots, k, \\ & \mathcal{M}_{r-1}(\varphi_i \zeta) \succeq 0 \qquad \text{for } i = 1, \ldots, n, \\ & -t_i \leq L_\zeta(x_i) - \bar{x}_i \leq t_i \quad \text{for } i = 1, \ldots, n, \\ & \zeta \in \mathbb{R}^{\mathbb{N}^n_{0,2r}},\ t \in \mathbb{R}^n_{\geq 0}, \end{aligned} \tag{$\text{MOM}_r^{\text{HYP}}$}$$

where $\nu_j = \lceil \deg(g_j)/2 \rceil$. Since this is a relaxation of (HYP_{dual}), we have

$$\rho_r^* \leq \text{dist}_1(\bar{x}, S) \tag{3}$$

and also weak duality holds. The proof is similar to the one in [Lau09, Section 4.2].

Proposition 3.7 (Weak Duality)**.** *Let h be feasible for* ($\text{SOS}_r^{\text{HYP}}$) *and* $(\zeta, t) \in \mathbb{R}^{\mathbb{N}^n_{0,2r}} \times \mathbb{R}^n_{\geq 0}$ *feasible for* ($\text{MOM}_r^{\text{HYP}}$)*. Then*

$$h(\bar{x}) \leq e^\top t.$$

Proof. Since $-h \in Q_r(g_1, \ldots, g_k, \varphi_1, \ldots, \varphi_n)$ holds, we have $L_\zeta(-h) \geq 0$. For $x^\zeta := (L_\zeta(x_1), \ldots, L_\zeta(x_n))$ we see that

$$\begin{aligned} h(\bar{x}) &\leq h(\bar{x}) - L_\zeta(h) = h(\bar{x}) - h(x^\zeta) \\ &= w_h^\top (\bar{x} - x^\zeta) \leq \|\bar{x} - x^\zeta\|_1 \leq e^\top t \end{aligned}$$

which gives the claim. □

To prove our main theorem, we rely on some auxiliary results. The most prominent among them is Putinar's Positivstellensatz [Put93], which requires the *Archimedean property* for the quadratic module: There must exist some $N \in \mathbb{N}$ such that

$$N - \sum_{i=1}^n x_i^2 \in Q(g_1, \ldots, g_k, \varphi_1, \ldots, \varphi_n).$$

Such a ball-constraint polynomial is even contained in $Q_1(\varphi_1, \ldots, \varphi_n)$ for some $N \in \mathbb{N}$, because the quadratic descriptions of the box constraints are used (cf. [ESV25, Proof of Lemma 4.2]).

Theorem 3.8 (Putinar's Positivstellensatz, as cited in [Lau09, Theorem 3.20])**.** *Let $f, g_1, \ldots, g_k \in \mathbb{R}[x]$, $S = \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_k(x) \geq 0\}$ and assume that the quadratic module $Q(g_1, \ldots, g_k)$ is Archimedean. If $f > 0$ on S, then $f \in Q(g_1, \ldots, g_k)$.*

The next auxiliary result shows that the entries of a truncated moment matrix can be bounded.

Proposition 3.9 ([Las09, Proposition 3.6])**.** *Let $r \geq 1$, $\zeta = (\zeta_\alpha)_{\alpha \in \mathbb{N}_0^n} \subseteq \mathbb{R}$ be a sequence indexed by the monomials $(x^\alpha)_{\alpha \in \mathbb{N}_0^n}$ and $L_\zeta : \mathbb{R}[x] \to \mathbb{R}$ the corresponding Riesz functional such that $M_r(\zeta) \succeq 0$. Then*

$$\forall \alpha \in \mathbb{N}^n_{0,2r} : |\zeta_\alpha| \leq \max\left(\zeta_0, \max_{i=1,\ldots,n} L_\zeta(x_i^{2r})\right).$$

Proof. For a proof we refer to [LN07, Lemma 4.3]. □

The last result needed is a strengthening of the classical Extreme Value Theorem (e.g., [Ste24, Theorem 1.2.13]).

Theorem 3.10. *Let $X \subseteq \mathbb{R}^n$, $J : X \to \mathbb{R}$ be a continuous function and $x_0 \in X$ such that the level set*

$$\{x \in X \mid J(x) \leq J(x_0)\}$$

is compact. Then there exists at least one global minimizer of J in X.

Now we can prove our main result: Both ($\mathrm{SOS}_r^{\mathrm{HYP}}$) and its dual ($\mathrm{MOM}_r^{\mathrm{HYP}}$) have an optimal solution, whenever S has nonempty interior. Furthermore, the optimal value of ($\mathrm{SOS}_r^{\mathrm{HYP}}$) asymptotically converges to the optimal value of (HYP). The proof follows very closely the arguments in [Las01] and [Lau09].

Theorem 3.11. *Let S be as in* (1) *with nonempty interior and $\bar{x} \in \mathbb{R}^n$. Then there is no duality gap between the semidefinite program* ($\mathrm{SOS}_r^{\mathrm{HYP}}$) *and its dual* ($\mathrm{MOM}_r^{\mathrm{HYP}}$)*. Moreover,* ($\mathrm{SOS}_r^{\mathrm{HYP}}$) *(resp.* ($\mathrm{MOM}_r^{\mathrm{HYP}}$)*) has an optimal solution $h_r \in \mathbb{R}[x]_{2r}$ (resp. $(\zeta, t) \in \mathbb{R}^{\mathbb{N}^n_{0,2r}} \times \mathbb{R}^n_{\geq 0}$) and*

$$h_r(\bar{x}) \to \operatorname{dist}_1(\bar{x}, \operatorname{conv} S) \quad \text{as } r \to \infty.$$

Proof. We show, that there exists a Slater point for ($\mathrm{MOM}_r^{\mathrm{HYP}}$). As S has nonempty interior, there exists a nonempty ball $B \subseteq S$. Define the probability measure

$$\mu(A) := \tfrac{1}{\lambda(B)} \lambda(A \cap B) \quad \text{for all Borel sets } A \subseteq \mathbb{R}^n,$$

where λ is the Lebesgue measure on $\mathbb{R}^n$. The sequence $\zeta = (\zeta_\alpha)_{\alpha \in \mathbb{N}_0^n}$ of moments is given as

$$\zeta_\alpha := \int_{\mathbb{R}^n} x^\alpha \; d\mu(x) \quad \text{for all } \alpha \in \mathbb{N}_0^n.$$

Note that $\zeta_0 = 1$. Let $u \in \mathbb{R}[x]_{\leq r}$ with $u \neq 0$ and $\operatorname{vec}(u)$ be the vector of coefficients of u. Then

$$\begin{aligned}
\langle \operatorname{vec}(u), \mathcal{M}_r(\zeta) \operatorname{vec}(u) \rangle &= \int_{\mathbb{R}^n} u(x)^2 \; d\mu(x) \\
&= \frac{1}{\lambda(B)} \int_B u(x)^2 \; d\lambda(x) > 0, \\
\langle \operatorname{vec}(u), \mathcal{M}_{r-\nu_j}(g_j \zeta) \operatorname{vec}(u) \rangle &= \int_{\mathbb{R}^n} g_j(x) u(x)^2 \; d\mu(x) \\
&= \frac{1}{\lambda(B)} \int_B g_j(x) u(x)^2 \; d\lambda(x) > 0,
\end{aligned}$$

which shows $\mathcal{M}_r(\zeta) \succ 0$ and $\mathcal{M}_{r-\nu_j}(g_j \zeta) \succ 0$ for all $j \in \{1, \ldots, k\}$. Similarily, $\mathcal{M}_{r-1}(\varphi_i \zeta) \succ 0$ can be shown for all $i \in \{1, \ldots, n\}$. Finally, choose each $t_i \in \mathbb{R}^n$

sufficient large, such that $|\zeta_{e_i} - \bar{x}_i| < t_i$ for all $i \in \{1, \ldots, n\}$, where $e_i \in \mathbb{R}^n$ is the i-th unit vector.

Hence (ζ, t) is a strictly feasibly solution, that is a Slater point, for the semidefinite program ($\text{MOM}_r^{\text{HYP}}$). If the optimal value of ($\text{MOM}_r^{\text{HYP}}$) is finite, then there is no duality gap by the strong duality result from convex optimization (e.g., [Ber08, Proposition 5.3.1]) and ($\text{SOS}_r^{\text{HYP}}$) has an optimal solution. For this, let (ζ', t') be any feasible solution of ($\text{MOM}_r^{\text{HYP}}$). Note that $\zeta'_0 = 1$. Since

$$N - \sum_{i=1}^{n} x_i^2 \in Q_1(\varphi_1, \ldots, \varphi_n) \subseteq Q_1(g_1, \ldots, g_k, \varphi_1, \ldots, \varphi_n)$$

for some $N \in \mathbb{N}$ and $\mathcal{M}_r(\zeta') \succeq 0$, $\mathcal{M}_{r-\nu_j}(g_j\zeta') \succeq 0$, $\mathcal{M}_{r-1}(\varphi_i\zeta') \succeq 0$, we can verify

$$L_{\zeta'}\left(x_l^{2(k-1)}(N - \sum_{i=1}^{n} x_i^2)\right) \geq 0 \quad \text{for } 1 \leq k \leq r$$

for all $l \in \{1, \ldots, n\}$ using the calculation rules from Lemma 2.9. For $k = 1$, we can conclude $N \geq L_{\zeta'}(x_l^2)$ and inductively $N^r \geq L_{\zeta'}(x_l^{2r})$. Proposition 3.9 implies that the absolute value of all ζ'_α is bounded by N^r. Therefore the level set with respect to the Slater point (ζ, t) of the feasible set of ($\text{MOM}_r^{\text{HYP}}$) is compact. By Theorem 3.10, this implies that ($\text{MOM}_r^{\text{HYP}}$) has an optimal solution with finite optimal value.

Finally, the convergence is shown. Since S is nonempty, we choose an optimal solution $h^* \in \mathbb{R}[x]$ of (HYP) as in Proposition 3.2. It suffices to show

$$h_r(\bar{x}) \to h^*(\bar{x}) \quad \text{as } r \to \infty.$$

If $\bar{x} \in \text{conv}(S)$, we have $h^*(\bar{x}) = 0 = h_r(\bar{x})$ for all $r \in \mathbb{N}$. So assume $\bar{x} \notin \text{conv}(S)$. Then $h^* \neq 0$ and $-h^* + \varepsilon > 0$ on S for $0 < \varepsilon \leq \text{dist}_1(\bar{x}, \text{conv}(S)) = h^*(\bar{x})$. Now, Putinar's Positivstellensatz 3.8 holds, that means

$$-h_\varepsilon := -h^* + \varepsilon \in M_R(g_1, \ldots, g_k, \varphi_1, \ldots, \varphi_n)$$

with $R \in \mathbb{N}$ large enough. Thus, h_ε is feasible for all ($\text{SOS}_r^{\text{HYP}}$) with $r \geq R$ and by definition $h_\varepsilon(\bar{x}) \geq 0$. Let h_r denote the optimal solution of ($\text{SOS}_r^{\text{HYP}}$) with $r \geq R$. Then $h_\varepsilon(\bar{x}) \leq h_r(\bar{x}) \leq h^*(\bar{x})$, where the last inequality holds by Proposition 3.5. Now,

$$0 \leq h^*(\bar{x}) - h_r(\bar{x}) \leq h^*(\bar{x}) - h_\varepsilon(\bar{x}) = \varepsilon$$

and the convergence follows. □

4 Convex Case

The moment-/SOS-hierarchy is able to recognize a large class of convex problems. For so-called SOS-convex problems, the first step of the classical moment-/SOS-hierarchy is exact. In the following section, we show that this behavior naturally extends to the case of calculating separating hyperplanes by adapting the results of [Las15, Chapter 13] and references therein. We begin with the notion of SOS-convexity of a polynomial, which was introduced in [HN10].

Definition 4.1 ([Las15, Definition 13.13])**.** Let $r \in \mathbb{N}$. A polynomial $f \in \mathbb{R}[x]_{2r}$ is said to be SOS-*convex* if its Hessian $\nabla^2 f$ is an SOS matrix-polynomial, that is, $\nabla^2 f = LL^\top$ for some real matrix-polynomial $L \in \mathbb{R}[x]^{n\times s}$ and some $s \in \mathbb{N}$.

Remark 4.2. From Definition 4.1 it follows that all SOS-convex polynomials are convex since the Hessian is positive semidefinite. In the quadratic case also the converse is true: Every convex polynomial up to degree 2 is SOS-convex. This implies in particular that the polynomials $-\varphi_i$, $i \in \{1, \dots, n\}$, which represent the box constraints, are SOS-convex.

In the SOS-convex case, we can consider simplified versions of the semidefinite programs ($\mathrm{SOS}_r^{\mathrm{HYP}}$) and its dual ($\mathrm{MOM}_r^{\mathrm{HYP}}$): That is, the order $r \in \mathbb{N}$ can be set to $r = \max(1, \max_{j=1,\dots,k}\lceil \deg(g_j)/2\rceil)$. This yields the following two semidefinite programs:

$$
\begin{aligned}
\rho^{\mathrm{c}} = \sup_{h,\lambda,\sigma_0} \quad & h(\bar{x}) \\
\text{s.t.} \quad & -h = \sigma_0 + \sum_{j=1}^{k} \lambda_j g_j + \sum_{i=1}^{n} \lambda_{i+k}\varphi_i, \\
& \|w_h\|_\infty \le 1, \\
& h \in \mathbb{R}[x]_1,\ \lambda \in \mathbb{R}^{n+k}_{\ge 0},\ \sigma_0 \in \Sigma[x]_r
\end{aligned}
\tag{$\mathrm{SOS}_r^{\mathrm{HYP,c}}$}
$$

and

$$
\begin{aligned}
\rho^{\mathrm{c},*} = \inf_{\zeta,t} \quad & e^\top t \\
\text{s.t.} \quad & \zeta_0 = 1,\ \mathcal{M}_r(\zeta) \succeq 0 \\
& L_\zeta(g_j) \ge 0 && \text{for } j = 1, \dots, k, \\
& L_\zeta(\varphi_i) \ge 0 && \text{for } i = 1, \dots, n, \\
& -t_i \le L_\zeta(x_i) - \bar{x}_i \le t_i && \text{for } i = 1, \dots, n, \\
& \zeta \in \mathbb{R}^{\mathbb{N}^n_{0,2r}},\ t \in \mathbb{R}^n_{\ge 0}.
\end{aligned}
\tag{$\mathrm{MOM}_r^{\mathrm{HYP,c}}$}
$$

Before we prove that the program ($\mathrm{SOS}_r^{\mathrm{HYP,c}}$) yields the optimal separating hyperplane, some auxiliary results are needed. For polynomials $p_1, \dots, p_k \in \mathbb{R}[x]$ define

$$
Q_c(p_1, \dots, p_k) \coloneqq \left\{ \sigma_0 + \sum_{j=1}^{k} \lambda_j p_j \,\middle|\, \lambda_j \ge 0,\ \sigma_0 \in \Sigma[x],\ \sigma_0 \text{ convex} \right\}
$$

as a specialization of the quadratic module $Q(p_1, \dots, p_k)$. We have the following certificate of nonnegativity in the SOS-convex case and an extension of Jensen's inequality.

Theorem 4.3 ([Las15, Theorem 13.19]). *Let $p, p_1, \dots, p_k \in \mathbb{R}[x]$ and $S \coloneqq \{x \in \mathbb{R}^n \mid p_1(x) \geq 0, \dots, p_k(x) \geq 0\}$ have a nonempty interior. In addition, let $p_{\min} \coloneqq \inf\{p(x) \mid x \in S\} = p(x^*)$ for some $x^* \in S$. If p and $-p_1, \dots, -p_k$ are* SOS-*convex then $p - p_{\min} \in Q_c(p_1, \dots, p_k)$.*

Theorem 4.4 ([Las15, Theorem 13.21]). *Let $p \in \mathbb{R}[x]_{2r}$ be* SOS-*convex and let $\zeta \in \mathbb{R}^{\mathbb{N}^n_{0,2r}}$ such that $\zeta_0 = 1$ and $\mathcal{M}_r(\zeta) \succeq 0$. Then*

$$L_\zeta(p) \geq p(L_\zeta(x_1), \dots, L_\zeta(x_n)).$$

Following the proof in [Las15, Theorem 13.32], we can now show that in the SOS-convex case ($\mathrm{SOS}_r^{\mathrm{HYP,c}}$) is exact. That means the hyperplane calculated by ($\mathrm{SOS}_r^{\mathrm{HYP,c}}$) is an optimal solution for the original separating hyperplane problem (HYP). From the solution of the dual ($\mathrm{MOM}_r^{\mathrm{HYP,c}}$), we can even derive a point in the feasible set S, which minimizes the distance to the point $\bar{x}$, that means we obtain an optimal solution for the optimization problem in (2).

Theorem 4.5. *Let S be as in* (1) *with nonempty interior and $\bar{x} \in \mathbb{R}^n$. Moreover, let all $-g_j$ be* SOS-*convex. Then* ($\mathrm{SOS}_r^{\mathrm{HYP,c}}$) *(resp.* ($\mathrm{MOM}_r^{\mathrm{HYP,c}}$)*) has an optimal solution $h^{\mathrm{c}} \in \mathbb{R}[x]_{2r}$ (resp. $(\zeta^{\mathrm{c}}, t^{\mathrm{c}}) \in \mathbb{R}^{\mathbb{N}^n_{0,2r}} \times \mathbb{R}^n_{\geq 0}$) and*

$$h^{\mathrm{c}}(\bar{x}) = \rho^{\mathrm{c}} = \mathrm{dist}_1(\bar{x}, S).$$

In addition, $x^* \coloneqq (L_{\zeta^{\mathrm{c}}}(x_1), \dots, L_{\zeta^{\mathrm{c}}}(x_n))$ lies in S and solves $\min_{z \in S} \|z - \bar{x}\|_1$.

Proof. Since S is nonempty, there exists an optimal solution of (HYP) by Proposition 3.2. Denote such an optimal solution by h. Since S is compact, there exists a minimizer $\tilde{x} \in S$ for $-h$. Furthermore, $-h$, $-g_j$ and $-\varphi_i$ are all SOS-convex. Thus, Theorem 4.3 can be applied, and $-h + h(\tilde{x}) \in Q_c(g_1, \dots, g_k, \varphi_1, \dots, \varphi_n)$. Since $-h \geq 0$ on S, we have $-h \in Q_c(g_1, \dots, g_k, \varphi_1, \dots, \varphi_n)$. With the degree bound for the SOS decomposition from Lemma 2.8 we obtain that $h^{\mathrm{c}} \coloneqq h$ is feasible for ($\mathrm{SOS}_r^{\mathrm{HYP,c}}$). By Proposition 3.2, $h^{\mathrm{c}}(\bar{x}) = \mathrm{dist}_1(\bar{x}, S)$. Since weak duality (cf. Proposition 3.7) and inequality (3) hold for ($\mathrm{SOS}_r^{\mathrm{HYP,c}}$) and ($\mathrm{MOM}_r^{\mathrm{HYP,c}}$) as well, we have

$$h^{\mathrm{c}}(\bar{x}) = \rho^{\mathrm{c}} = \rho^{\mathrm{c},*} = \mathrm{dist}_1(\bar{x}, S).$$

As $\inf\{\|x - \bar{x}\|_1 \mid x \in S\}$ is a convex optimization problem and S compact, there exists an optimal solution $\hat{x} \in \mathrm{argmin}\{\|x - \bar{x}\|_1 \mid x \in S\}$. Thus, we can define $\zeta^{\mathrm{c}} \coloneqq (\hat{x}^\alpha)_{\alpha \in \mathbb{N}^n_{0,2r}}$ by using the Dirac measure at the point $\hat{x}$, and $t^{\mathrm{c}}_i \coloneqq |\hat{x}_i - \bar{x}_i|$ for all $i \in \{1, \dots, n\}$. Now, $(\zeta^{\mathrm{c}}, t^{\mathrm{c}})$ is optimal for ($\mathrm{MOM}_r^{\mathrm{HYP,c}}$). Let (ζ, t) be any optimal solution to ($\mathrm{MOM}_r^{\mathrm{HYP,c}}$). Then $|L_\zeta(x_i) - \bar{x}_i| = t_i$ holds for all $i \in \{1, \dots, n\}$ by optimality and we have

$$\sum_{i=1}^n |L_\zeta(x_i) - \bar{x}_i| = \sum_{i=1}^n t_i = \rho^{\mathrm{c},*} = \mathrm{dist}_1(\bar{x}, S) = \inf_{x \in S} \|x - \bar{x}\|_1$$

holds. Define $x^* \coloneqq (L_\zeta(x_1), \ldots, L_\zeta(x_n))$. Since $-g_j$ and $-\varphi_i$ are SOS-convex, we have with Theorem 4.4

$$0 \geq L_\zeta(-g_j) \geq -g_j(x^*) \quad \text{and} \quad 0 \geq L_\zeta(-\varphi_i) \geq -\varphi_i(x^*).$$

Thus, x^* lies in S and hence solves $\min_{z \in S} \|z - \bar{x}\|_1$. □

5 A Separating Hyperplane Algorithm

We can now embed the calculation of separating hyperplanes in an algorithm. Algorithm `HYP-SOS` takes as input a mixed-integer polynomial optimization problem (MIPOP), an order $r \in \mathbb{N}$, a tolerance $\varepsilon > 0$, and initializes the iteration counter t.

First, an initial relaxed set Ω_0 is defined, and the corresponding relaxed solution x_0^{relax} is computed. In our numerical tests, the initial relaxed set is the feasible set of the mixed-integer linear relaxation (MILP) by only taking the linear constraints from (MIPOP).

The algorithm then enters a loop. Using the relaxed solution of (MILP), ($\text{SOS}_r^{\text{HYP}}$) is solved to obtain a polynomial $h_t \in \mathbb{R}[x]_1$ in each iteration. If the optimal value of ($\text{SOS}_r^{\text{HYP}}$) is less than the tolerance ε, h_t cannot efficiently cut off the relaxed solution and the algorithm is terminated. The current relaxed solution is returned as final output. By evaluating the objective, a lower bound for (MIPOP) can be derived.

If the algorithm does not terminate, the valid linear inequality $-h_t(x) \geq 0$ is added to the description of the relaxed feasible set Ω_t, creating a new relaxed set Ω_{t+1} that excludes the relaxed solution x_t^{relax}. This refined relaxation is solved to obtain a new relaxed solution x_{t+1}^{relax} and the iteration counter is incremented. This process repeats until the current relaxed solution x_t^{relax} cannot be effectively separated from the feasible set of (MIPOP) by h_t anymore.

Algorithm `HYP-SOS` (Separating hyperplane algorithm for MIPOPs)

Require: MIPOP, $r \in \mathbb{N}$, tolerance $\varepsilon > 0$.

```
t ← 0.
Define initial relaxed set Ω_0 and obtain initial relaxed solution x_0^relax.
while True do
    Solve (SOS_r^HYP) with x_t^relax and obtain h_t ∈ ℝ[x]_1.
    if h_t(x_t^relax) < ε then
        return x_t^relax and terminate the algorithm.
    end if
    Add linear inequality h_t(x) ≤ 0 to Ω_t to define new relaxed set Ω_{t+1}.
    Solve relaxation with Ω_{t+1} and obtain new relaxed solution x_{t+1}^relax.
    t ← t + 1.
end while
```

Before we prove some convergence results, we want to illustrate our method by considering Example 1 from [LK22].

Example 5.1. Let us consider the following optimization problem:

$$\begin{aligned} \min_{(x,y)\in\mathbb{R}\times\mathbb{Z}} \quad & f(x,y) \coloneqq 4x - 15y \\ \text{s.t.} \quad & l_1(x,y) \coloneqq -x - 10y \leq -6,\ l_2(x,y) \coloneqq x - 10y \leq 4, \\ & g_1(x,y) \coloneqq 8.8x - x^2 + 7y + xy - y^2 \leq 23.5, \\ & g_2(x,y) \coloneqq -10x + x^2 - 10y + 2y^2 \leq -25.1, \\ & 2 \leq x \leq 8,\ 0 \leq y \leq 2. \end{aligned}$$

The problem involves two bounded variables x and y, where x is continuous and y integer. Furthermore, there are two linear constraints l_1 and l_2 and two non-linear constraints g_1 and g_2. The constraint g_1 is nonconvex and g_2 is convex. In Figure 1, the integer-relaxed feasible set is illustrated as a banana-shaped black area, whereas the dotted horizontal lines represent the possible integer values of y. The dark gray area covers the feasible set of the linear relaxation, and the black dot is the solution x_t^{relax} of the current mixed-integer linear relaxation obtained by an MILP solver. The black lines are the hyperplanes obtained by solving ($\text{SOS}_r^{\text{HYP}}$) for $r = 2$, and the corresponding valid linear inequalities are iteratively added to strengthen the linear relaxation. After adding three inequalities, the algorithm terminates successfully. The algorithm even gives the correct solution, which cannot be expected in general.

In contrast to Example 5.1, an optimal solution cannot be found by our algorithm in general. But convergence of the lower bound to the optimal value can be guaranteed under some assumptions. To prove this convergence, we will closely follow the structure in [KLW16]. The main difference from [KLW16] is that our problems are polynomial and hence can be nonconvex. We first state the necessary assumptions that shall hold for the remaining section. Then, some auxiliary results are proven.

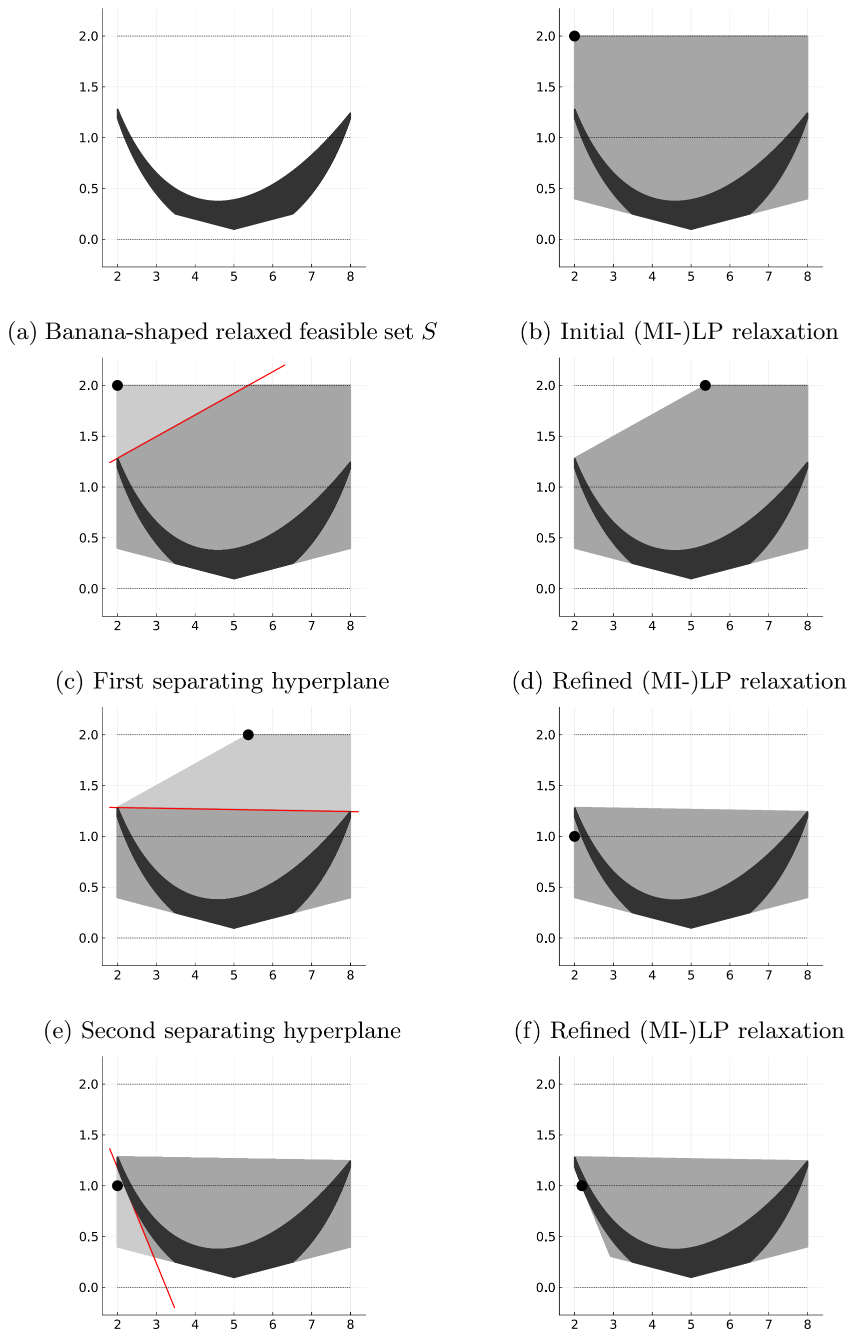
Assumption 5.2.

1. All integer constraints are expressed by (low degree) polynomials, which means $M = S$. (cf. beginning of Section 3).
2. The order $r \in \mathbb{N}$ can be chosen, such that all problems ($\text{SOS}_r^{\text{HYP}}$) are solved to optimality, that means the solution coincides with a solution of (HYP).
3. Relaxations with relaxed set Ω_t are also solved to optimality.
4. The tolerance ε is set to 0.

Lemma 5.3. *Let $t \in \mathbb{N}_0$. For the current relaxed solution x_t^{relax} in Algorithm* ***HYP-SOS*** *holds*

$$x_t^{\text{relax}} \notin \text{conv}(S) \iff h_t(x_t^{\text{relax}}) > 0,$$

where $h_t \in \mathbb{R}[x]_1$ is the polynomial calculated in iteration t.



(a) Banana-shaped relaxed feasible set S

(b) Initial (MI-)LP relaxation

(c) First separating hyperplane

(d) Refined (MI-)LP relaxation

(e) Second separating hyperplane

(f) Refined (MI-)LP relaxation

(g) Third separating hyperplane

(h) Optimality reached!

Figure 1: Visualization of Algorithm `HYP-SOS` for Example 5.1.

Proof. First, let $x_t^{\text{relax}} \notin \text{conv}(S)$. By Remark 3.1, the optimal value of (HYP) is positive. Since the solutions of ($\text{SOS}_r^{\text{HYP}}$) and (HYP) coincide by Assumption 5.2, $h_t(x_t^{\text{relax}}) > 0$ holds. Secondly, let $h_t(x_t^{\text{relax}}) > 0$. By definition, we have $h_t \leq 0$ on $\text{conv}(S)$, which yields $x_t^{\text{relax}} \notin \text{conv}(S)$. □

Lemma 5.4. *Let $T \in \mathbb{N}_0$. Then Algorithm `HYP-SOS` stops after T iterations, if and only if $x_T^{\text{relax}} \in \text{conv}(S)$. If $x_T^{\text{relax}} \in \text{conv}(S)$, then $f(x_T^{\text{relax}}) = f_{\min}$.*

Proof. By Lemma 5.3, the termination condition $h_t(x_t^{\text{relax}}) = 0$ is not satisfied if and only if $x_t^{\text{relax}} \notin \text{conv}(S)$. Negation of both statements yields the first claim. Further, let $x_T^{\text{relax}} \in \text{conv}(S)$, then we have $f_{\min} \leq f(x_T^{\text{relax}})$ by Proposition 1.1. Since x_T^{relax} is the solution of a relaxation with same objective function f, we have $f(x_T^{\text{relax}}) \leq f_{\min}$. Thus, $f(x_T^{\text{relax}}) = f_{\min}$. □

Lemma 5.5. *If Algorithm `HYP-SOS` does not stop in a finite number of iterations, it generates a convergent subsequence $\{x_{t_i}^{\text{relax}}\}_{i=0}^{\infty}$.*

Proof. If Algorithm `HYP-SOS` do not stop in a finite number of iterations, then all the iterates are not in $\text{conv}(S)$ by Lemma 5.4. By Lemma 5.3, the iterates are all distinct. Since all variables are bounded, the sequence of iterates is bounded as well. Thus, there exists a convergent subsequence by the Bolzano-Weierstrass theorem. □

Lemma 5.6. *Let $\{x_t^{\text{relax}}\}_{t=0}^{\infty}$ be a convergent (sub-)sequence generated by Algorithm `HYP-SOS`. Then its limit $\tilde{x}$ belongs to $\text{conv}(S)$ and $f(\tilde{x}) = f_{\min}$.*

Proof. We first show, that $h_t(\tilde{x}) \leq 0$ holds for all $t \in \mathbb{N}_0$. Assume $h_{t'}(\tilde{x}) > 0$ holds for some $t' \in \mathbb{N}_0$. Then $\tilde{x}$ is excluded from the relaxed set $\Omega_{t'+1}$, which is a closed set, but $x_t \in \Omega_{t'+1}$ for all $t \geq t'+1$. Thus, $\tilde{x}$ cannot be the limit of the sequence $\{x_t^{\text{relax}}\}_{t=0}^{\infty}$.

Assume now that $\tilde{x} \notin \text{conv}(S)$. There exists $T \in \mathbb{N}_0$ such that

$$\|x_T - \tilde{x}\| < \text{dist}_1(x_T, \text{conv}(S)).$$

Indeed, choose $T_1 \in \mathbb{N}_0$ and $\varepsilon > 0$ with $\varepsilon < \text{dist}_1(\tilde{x}, \text{conv}(S))$, such that $\|x_t - \tilde{x}\|_1 < \text{dist}_1(\tilde{x}, \text{conv}(S)) - \varepsilon$ for all $t \geq T_1$. Next, since $\text{dist}_1(\cdot, \text{conv}(S))$ is a continuous function, we have $\text{dist}_1(x_t, \text{conv}(S)) \to \text{dist}_1(\tilde{x}, \text{conv}(S))$ for $t \to \infty$. Choose $T_2 \in \mathbb{N}_0$, such that $|\text{dist}_1(\tilde{x}, \text{conv}(S)) - \text{dist}_1(x_t, \text{conv}(S))| < \varepsilon$ for all $t \geq T_2$. Define $T := \max(T_1, T_2)$. Then,

$$\begin{aligned}
\|x_T - \tilde{x}\|_1 &< \text{dist}_1(\tilde{x}, \text{conv}(S)) - \varepsilon \\
&< \text{dist}_1(\tilde{x}, \text{conv}(S)) - |\text{dist}_1(\tilde{x}, \text{conv}(S)) - \text{dist}_1(x_T, \text{conv}(S))| \\
&\leq \text{dist}_1(\tilde{x}, \text{conv}(S)) - \text{dist}_1(\tilde{x}, \text{conv}(S)) + \text{dist}_1(x_T, \text{conv}(S)) \\
&= \text{dist}_1(x_T, \text{conv}(S)).
\end{aligned}$$

By Assumption 5.2 and Proposition 3.2, $h_T(x_T) = \text{dist}_1(x_T, \text{conv}(S))$ and

$$|\text{dist}_1(x_T, \text{conv}(S)) - h_T(\tilde{x})| = |h_T(x_T) - h_T(\tilde{x})|$$

$$\leq \|x_T - \tilde{x}\|_1 < \mathrm{dist}_1(x_T, \mathrm{conv}(S))$$

hold (cf. Proof of Lemma 2.7). Thus, we must have $h_T(\tilde{x}) > 0$, which is a contradiction. Thus, $\tilde{x} \in \mathrm{conv}(S)$ must hold.

Since $\tilde{x} \in \mathrm{conv}(S)$, we have $f_{\min} \leq f(\tilde{x})$ by Proposition 1.1. Since all x_t^{relax} are the solutions of a relaxation with same objective function f, we have $f(x_t^{\mathrm{relax}}) \leq f_{\min}$ for all $t \in \mathbb{N}_0$. By continuity of f, we also have $f(\tilde{x}) \leq f_{\min}$ Thus, $f(\tilde{x}) = f_{\min}$. □

We can now show that Algorithm HYP-SOS converges under Assumption 5.2. If the tolerance is not zero, then the algorithm stops after finitely many steps with arbitrary precision. By our assumptions, Algorithm HYP-SOS reduces to a general cutting plane procedure and therefore similar results can be found in the literature (e.g., [BCM20, Section 2]).

Theorem 5.7. *Algorithm HYP-SOS either stops after finitely many iterations T for some $T \in \mathbb{N}$ with $f(x_T^{\mathrm{relax}}) = f_{\min}$ or generates a convergent (sub-)sequence $\{x_t^{\mathrm{relax}}\}_{t=0}^{\infty}$ with limit $\tilde{x}$ satisfying $f(\tilde{x}) = f_{\min}$.*

Proof. The first part follows from Lemma 5.4 and the second part from Lemmas 5.5 and 5.6. □

Theorem 5.8. *For tolerance $\varepsilon > 0$, the Algorithm HYP-SOS terminates after finitely many iterations T for some $T \in \mathbb{N}$ satisfying*

$$\mathrm{dist}_1(x_T^{\mathrm{relax}}, \mathrm{conv}(S)) \leq \varepsilon.$$

Proof. Assume that Algorithm HYP-SOS does not terminate after finitely many iterations. By Lemma 5.6, the algorithm generates a convergent sequence $\{x_t^{\mathrm{relax}}\}_{t=0}^{\infty}$ with $h_t(x_t^{\mathrm{relax}}) \to 0$ for $t \to \infty$. Thus, there exists a $T \in \mathbb{N}$ with $h_T(x_T^{\mathrm{relax}}) \leq \varepsilon$ and the termination criterion is fulfilled, which is a contradiction. By Assumption 5.2 and Proposition 3.2,

$$\mathrm{dist}_1(x_T^{\mathrm{relax}}, \mathrm{conv}(S)) = h_T(x_T^{\mathrm{relax}}) \leq \varepsilon$$

which gives the claim. □

6 A Heuristic Using Sparsity

In theory, Algorithm HYP-SOS is able to approximate the optimal value of (MIPOP), but in practice scalability heavily relies on how fast the problems ($\mathrm{SOS}_r^{\mathrm{HYP}}$) can be solved. Usually, solving ($\mathrm{SOS}_r^{\mathrm{HYP}}$) is very expensive, even for very small orders, since the typical size of the largest involved semidefinite matrices is $\binom{n+r}{r}$, where r is the order and n the number of variables. One way to reduce running time is to reduce the number of variables of the problems ($\mathrm{SOS}_r^{\mathrm{HYP}}$). If (MIPOP) has some sparse structure, that roughly means only a small number of different variables appear in each constraint, this structure can be used for a straightforward heuristic. But this comes at the price of losing the convergence to the convex hull of S.

Instead of finding a hyperplane h with $h \leq 0$ on the whole feasible set S, which is described by all constraints with all variables, we could focus on supersets of S, that are described by only one single constraint, or by a subset of constraints sharing many fewer variables than the original problem. Therefore, let $J \subseteq \{1, \ldots, k\}$ represent the indices of a subset of constraints and $I^J \subseteq \{1, \ldots, n\}$ the indices of variables which appear in the constraints with index in J. Then we can define a reduced problem

$$\begin{aligned} \rho_r^J = \sup_{h \in \mathbb{R}[x_{I^J}]_1} \quad & h(\bar{x}_{I^J}) \\ \text{s.t.} \quad & -h \in Q_r(g_J, \varphi_{I^J}), \\ & \|w_h\|_\infty \leq 1, \end{aligned} \tag{$\mathrm{SOS}_r^{\mathrm{HYP,J}}$}$$

where $x_{I^J} = (x_i)_{i \in I^J}$, $g_J = (g_j)_{j \in J}$, $\varphi_{I^J} = (\varphi_i)_{i \in I^J}$. Now, ($\mathrm{SOS}_r^{\mathrm{HYP}}$) can be replaced by ($\mathrm{SOS}_r^{\mathrm{HYP,J}}$) with appropriate index set J in each iteration in Algorithm `HYP-SOS`. Note that the hyperplane h obtained by ($\mathrm{SOS}_r^{\mathrm{HYP,J}}$) is a polynomial only in the variables x_{I^J}. Since we consider only a subset of constraints, ($\mathrm{SOS}_r^{\mathrm{HYP,J}}$) fails to produce a valid separating hyperplane more often. Therefore, different subsets can be tested in each iteration until a separating hyperplane is found or the algorithm stops. Before the choice of subsets is discussed, we present the modified algorithm.

Algorithm `Sparse HYP-SOS` (Sparse separating hyperplane algorithm)

Require: MIPOP, $r \in \mathbb{N}$, tolerance $\varepsilon > 0$.

$t \leftarrow 0$.
Define initial relaxed set Ω_0 and obtain initial relaxed solution x_0^{relax}.
while True **do**
  Define sets $J_1^t, \ldots, J_{d_t}^t \subseteq \{1, \ldots, k\}$.
  for $l \in \{1, \ldots, d_t\}$ **do**
    Solve ($\mathrm{SOS}_r^{\mathrm{HYP,J}}$) with x_t^{relax} and $J = J_l^t$ and obtain $h_{t,l} \in \mathbb{R}[x_{I^J}]_1$.
    if $h_{t,l}(x_t^{\mathrm{relax}}) \geq \varepsilon$ **then**
      $h_t \leftarrow h_{t,l}$.
      break
    end if
    if $l = d_t$ **then**
      return x_t^{relax} and terminate the algorithm.
    end if
  end for
  Add linear inequality $h_t(x) \leq 0$ to Ω_t to define new relaxed set Ω_{t+1}.
  Solve relaxation with Ω_{t+1} and obtain new relaxed solution x_{t+1}^{relax}.
  $t \leftarrow t + 1$.
end while

Compared to Algorithm `HYP-SOS`, only the step in which the separating hyperplane is calculated is replaced in each iteration t. Instead, $d_t \in \mathbb{N}$ subsets of constraints are defined at first, represented by their indices in the sets J_l^t. Then, problem ($\text{SOS}_r^{\text{HYP,J}}$) is solved for each subset. As soon as a valid separating hyperplane is found, the hyperplane is added to the description of the relaxed feasible set and the algorithm continues as in `HYP-SOS`. If no valid separating hyperplane is found after iterating over all defined subsets of constraints, the algorithm stops and returns the solution of the current relaxation. In the case $d_t = 1$ and $J_1 = \{1, \ldots, k\}$ the algorithm coincides with `HYP-SOS`.

But how shall the sets J_l^t be chosen in each iteration t? In the following, we discuss two approaches.

Single: Each J_l^t is defined as a singleton, which corresponds to one (nonlinear) constraint violated by the current relaxed solution. Then the sets are ordered such that J_1^t contains the index of the most violated constraint, J_2^t the second most violated constraint, and so on, expecting that a separating hyperplane can be found more quickly if the constraint is more violated. This approach usually leads to separating hyperplanes of poor quality, since the interplay with the other constraints is not considered. Thus, considering only one constraint might be too restrictive. However, when more than one constraint is added, the number of variables should not increase too much. This inspires the second approach.

Cliques: To find out which constraints share the same variables, we can rely on so-called correlative sparsity, which is a preliminary step to exploit sparsity in polynomial optimization first introduced in [Wak+06]. Roughly, the variables are decomposed in a set of cliques according to whether variables appear in the same constraints or the same terms in the objective function. Then all constraints are assigned to the cliques, which contain all variables that appear in the respective constraint. For our numerical experiments, we use the clique decomposition provided by the `Julia` package `TSSOS.jl` [MW21]. For details on how these cliques are calculated, the reader is referred to [MW23, Chapter 3]. As for the single constraint, the algorithm starts calculating a hyperplane for the subset of constraints that contains the most violated constraint and continues with the subsets containing fewer violated constraints.

We will show the impact of these suggestions for the small continuous example `ex3_1_1` from the MINLPLib [MIN25].

Example 6.1. Let us consider the following example with bilinear inequality constraints:

$$\begin{aligned}
\min_{x\in\mathbb{R}^8} \quad & x_1 + x_2 + x_3 \\
\text{s.t.} \quad & 0.0025(x_4 + x_6) - 1 \leq 0, \\
& 0.0025(-x_4 + x_5 + x_7) - 1 \leq 0, \\
& 0.01(-x_5 + x_8) - 1 \leq 0, \\
& 100x_1 - x_1x_6 + 833.33252x_4 - 83333.333 \leq 0, \\
& x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \leq 0, \\
& x_3x_5 - x_3x_8 - 2500x_5 + 1250000 \leq 0,
\end{aligned}$$

$$100 \leq x_1 \leq 10000,$$
$$1000 \leq x_2, x_3 \leq 10000,$$
$$10 \leq x_4, x_5, x_6, x_7, x_8 \leq 1000.$$

In Figure 2, the two approaches *single* and *cliques* are compared with the performance of the original algorithm `HYP-SOS` (*all*) for two different orders $r = 2$ and $r = 3$ and tolerance $\varepsilon = 10^{-6}$. The performance is measured by how much of the gap between the optimal value of the initial mixed-integer linear relaxation (MILP) and the optimal value of the original problem (MIPOP) is closed after iteration t (in %):

$$\text{gap}_{\text{cl},t} = 100 \cdot \frac{f(x_t^{\text{relax}}) - f(x_0^{\text{relax}})}{f_{\min} - f(x_0^{\text{relax}})}, \tag{4}$$

As expected, the *single* approach (in red) produces the poorest performance. The gap can be closed only up to 11 %, and there cannot be seen any improvement by raising the order from $r = 2$ to $r = 3$. In contrast, the *cliques* approach (in blue) manages to close the gap up to 89 %. Compared to order $r = 2$, the number of iterations required at order $r = 3$ is slightly lower, and the final relaxation value is marginally better. However, when all constraints are considered (in green), the results are the best as one would expect. At order $r = 2$, the algorithm performs the most iterations until no further separating hyperplane can be found and the gap can be nearly closed. If the order is raised to $r = 3$, then optimality is reached.

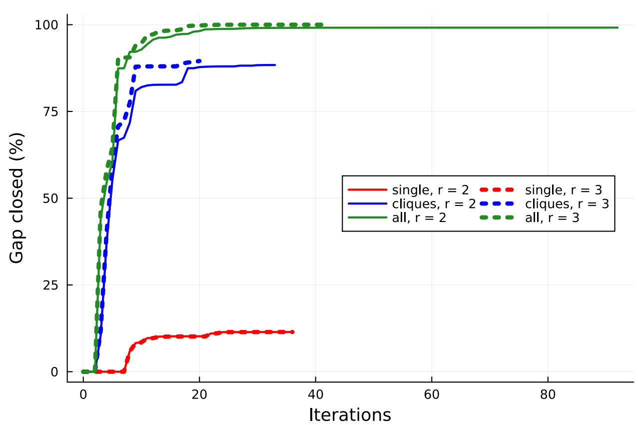


Figure 2: Performance of `Sparse HYP-SOS` for Example 6.1 using different subsets of constraints and orders r.

7 Numerical Study

In the following numerical study the separating hyperplane algorithm `Sparse HYP-SOS` is tested on examples from the MINLPLib [MIN25]. To show how the algorithm can be integrated in standard MINLP software, a modified version of the software tool `TSSOS` [MW21] is integrated into the `SHOT` solver [LKW22]. `TSSOS` is used to solve $\mathrm{SOS}_r^{\mathrm{HYP,J}}$ by exploiting the sparsity of the polynomial optimization problems. Originally, `SHOT` was intended for convex mixed-integer nonlinear problems, but its functionality also extends to nonconvex problems, but guaranteed global optimality is not provided in general [LK22]. Now, with the proposed separating hyperplane algorithm `Sparse HYP-SOS` this restriction may be lifted, at least for nonconvex mixed-integer polynomial optimization problems.

Because the `SHOT` solver mainly relies on hyperplanes, the theoretical integration comes quite naturally. Practically, since `SHOT` is written in `C++` and `TSSOS` in `Julia`, a custom interface between `C++` and `Julia` was created, utilizing `SHOT`'s newly developed callback functionality. The callback to the `Julia` code replaces `SHOT`'s built-in hyperplane generation functionality. The solution to each MILP relaxation is passed on through the callback to the `Julia` code, which in turn returns a separating hyperplane that is used by `SHOT` to cut away the previous relaxed solution. The `Julia` code is compiled the first time the code is executed, resulting in a significant execution time overhead. While a complete precompiled `C++` implementation would likely be more efficient, our current focus was on testing and validating the proposed methods. A full `C++` implementation may, however, be considered in future work.

All tests were carried out on a laptop with `Ubuntu 24.04.3 LTS`, `Intel Core i7-13700HX` with 24 cores and `64 GB RAM`. A prerelease version of `SHOT` is implemented. The programming language `Julia` is used with version `1.12.4` [Bez+17]. For solving the MILP relaxations in `SHOT`, the MILP solver `Cbc 2.10.11` is used [FL05]. For solving the SDPs, `Mosek 11.0.1` [MOS25] is used through `Julia`. Finally, `SHOT` requires an NLP solver for calculating feasible points and terminating the solution process, for which `IPOPT 3.14.20` is used [WB06].

For our numerical study, we compare the two approaches *single* and *cliques* from Section 6 for order 2. This enables us to apply the algorithm to problems with more than 1000 variables and constraints. The tolerance is set to $\varepsilon = 10^{-6}$, the time limit of the SDP solver for solving each ($\mathrm{SOS}_r^{\mathrm{HYP,J}}$) is set to 200 s, and the total time limit of the solver is set to 1000 s; note that this total time includes `Julia` compilation time and SDP solver runtime. Besides the termination criterion from Algorithm `Sparse HYP-SOS`, the solution process stops as soon as one of the internal termination criteria of `SHOT` are fulfilled.

There is a limit of 25 variables allowed to appear in the subset of constraints. If this limit is reached in the *single* approach, that means the current nonlinear constraint violating the current relaxed solution contains more than 25 variables, then this constraint is skipped, and the next constraint is chosen. In the *cliques* approach, if the clique of the current violated constraint has more than 25 variables, then there is a fallback into the *single* approach before skipping.

After reading the problem file, there are some presolving steps carried out, first by SHOT, then by the Julia script. In SHOT, the problem is reformulated. In particular, nonlinear objectives are rewritten using an epigraph reformulation. Further, a basic bound tightening procedure is done. Then, this reformulated problem is read by the Julia script, and the variables and constraints are scaled whenever possible. This is needed because sometimes the moment-/SOS-hierarchy can cause numerical issues if high degree monomials are involved [Wak+06, Section 5.6]. To avoid these numerical issues, all bounded variables are scaled before calculating the hyperplanes, such that they are in the interval $[-1, 1]$. Further, each constraint is divided through the absolute value of its coefficient by the maximal absolute value. The scaling of (MIPOP) is done once when the algorithm is started and then reused in each iteration. In each iteration only the current relaxed solution has to be scaled as input for $(\mathrm{SOS}_r^{\mathrm{HYP,J}})$, and the obtained hyperplane has to be rescaled to be used as a valid inequality in the relaxation. The SDP solver may still run into numerical problems, and no reliable hyperplane can be extracted, in particular, when unbounded variables are involved. In this case the current subset of constraints is skipped as well.

For our test set all (mixed-integer) polynomial optimization problems from the MINLPLib are considered. Then, problems with a very large number of variables or constraints, insufficient sparsity structure, very high-degree polynomials, or an unbounded initial MILP relaxation are excluded. Furthermore, we excluded problems for which significant numerical or memory issues occurred during the solution process. For a few instances of the `squfl` problem type, memory issues are avoided by setting the iteration limit to 999. In total, our test set consists of 352 polynomial instances, 133 of them are (mixed-)integer problems.

Unlike our assumption, some of the problems are formulated as maximization problems. Then, the objective value of the relaxations yield upper bounds for the optimal value instead of lower bounds. To avoid confusion, we call the bounds generated by our algorithm dual bounds, which means upper bounds for maximization problems and lower bounds for minimization problems. To measure the performance of the algorithm, we use the initial gap closure (4) from Section 6. This definition can be used for maximization problems as well. Further, there are some instances in the MINLPLib which are not solved to global optimality (i.e., less than three different solvers claim global optimality). In this case, we use the dual bound reported in the MINLPLib instead.

The detailed results of the computation can be found in Appendix A in Table A.1. The time to initialize `Julia`, including the scaling of the variables, is on average about 25 seconds, but varies from 18 seconds to over 400 seconds for large instances like `gabriel05`. There are 24 problems where the initial MILP relaxation yields the optimal value or at least the dual bound reported in the MINLPLib. Using the *single* (*cliques*) approach, Algorithm `Sparse HYP-SOS` can improve the dual bound in 212 (214) out of the remaining 328 instances, that corresponds to roughly two-thirds of the instances. The algorithm terminates for 283 (259) instances within the time limit. For 29 (50) instances, there are no improvements of the initial dual bound within the time limit.

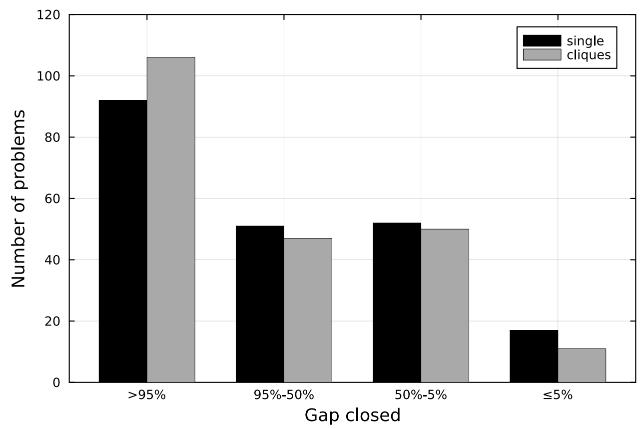


Figure 3: Performance of `Sparse HYP-SOS` for the *single* and *cliques* approach on instances from the MINLPLib

Figure 3 summarizes the results for the 212 (214) instances, for which the initial dual bound can be improved. In most of the instances, the initial gap can be closed significantly, and in about half of the instances, the initial gap can be closed almost completely. Only a few instances can be closed up to 5%. As expected, the *cliques* approach has a better performance than the *single* approach, but only slightly, which is probably due to longer runtime in each iteration and thus less separating hyperplanes that are added to the relaxation.

Our numerical study only concentrates on order $r = 2$ and Example 6.1 might indicate that raising the order yields only marginal better results. Nevertheless, raising the order can have a tremendous effect. As an illustration, Table 1 compares the gap closure for all instances of the `sssd` problem type for order $r = 2$ and $r = 3$. These instances are nonconvex mixed-binary quadratically constrained problems with up to 256 variables, most of which are binary, and up to 89 constraints. As reported in the MINLPLib, even some well-known mixed-integer solvers struggle with calculating good dual bounds.

For most of these instances Algorithm `Sparse HYP-SOS` can improve the dual bound only slightly, if the order is set to $r = 2$. For four instances the gap can be closed significantly, but the dual bound is still far away from optimal. If instead the order is set to $r = 3$, then all instances can be solved to optimality within the specified tolerances. For the instances `sssd18-08persp` and `sssd20-08persp` even the dual bound reported in the MINLPLib can be improved as shown in Table 2. Although merely illustrative, these results show a promising direction for future research.

name	single		cliques	
	2	3	2	3
sssd08-04persp	0.1	99.9	10.4	99.9
sssd12-05persp	0.1	99.9	0.2	99.9
sssd15-04persp	52.9	99.9	54.5	99.9
sssd15-06persp	7.1	99.9	15.0	99.9
sssd15-08persp	4.8	99.9	7.1	99.9
sssd16-07persp	52.8	99.9	52.7	99.9
sssd18-06persp	3.1	99.9	7.4	99.9
sssd18-08persp	9.2	**100.1**	12.1	**100.1**
sssd20-04persp	50.6	99.9	56.2	99.9
sssd20-08persp	0.6	**100.0**	56.4	**100.0**
sssd22-08persp	0.4	99.9	7.8	99.9
sssd25-04persp	49.6	99.9	50.7	99.9
sssd25-08persp	0.1	99.9	5.2	99.9

Table 1: Gap closure for `sssd` problems for orders $r = 2$ and $r = 3$. In **bold**: Improved dual bounds compared to the bounds from the MINLPLib.

8 Conclusion and Outlook

In this article, a separating hyperplane algorithm for mixed-integer polynomial optimization problems is proposed. Starting from an initial relaxation, in our case a mixed-integer linear relaxation, linear inequalities are iteratively added to the relaxation that cut off infeasible solutions of the previous relaxation. These separating hyperplanes are calculated by applying the moment-/SOS-hierarchy from polynomial optimization. With this approach the moment-/SOS-hierarchy can be made accessible to common MINLP solvers, which is illustrated by an integration into the `SHOT` solver, extending its ability to nonconvex mixed-integer polynomial optimization problems. Testing our approach on a large set of examples from the MINLPLib, we find a significant improvement of the initial relaxation in about two-thirds of the instances, in more than one quarter of the problems the gap between the objective value of the initial relaxation and the optimal value can be closed almost completely. The algorithm shows strong performance on some challenging problems, even outperforming well-known mixed-integer solvers. These results appear very promising and inspires further research directions.

Because the `SHOT` solver mainly relies on hyperplanes, the integration is quite natural. However, the calculated separating hyperplanes are valid global cuts for mixed-integer polynomial optimization problems, and they can be used in other kinds of MINLP solvers, for example, those that rely on Branch and Bound algorithms. Here, the hyperplanes can be used to tighten the relaxation in the root node. Further, an improvement can be reached by using linear inequalities that are introduced by the underlying MILP solver when infeasible integer points are cut off. If sufficiently sparse, these cutting planes can be

name	single		cliques		best bound
	s	bound	s	bound	
sssd08-04persp	33.5	182021.366	33.1	182022.253	182022.570
sssd12-05persp	38.3	281392.903	38.8	281386.467	281408.465
sssd15-04persp	35.7	205050.242	33.1	205046.343	205054.353
sssd15-06persp	46.5	539588.665	66.5	539606.089	539635.249
sssd15-08persp	170.3	562588.463	196.1	562575.772	562617.855
sssd16-07persp	79.6	417159.936	75.9	417159.817	417188.774
sssd18-06persp	46.9	397965.464	52.3	397958.933	397992.136
sssd18-08persp	237.4	**832749.246**	305.7	**832741.918**	831802.891
sssd20-04persp	36.1	347671.370	35.7	347688.509	347691.293
sssd20-08persp	248.7	**469596.569**	265.1	**469572.891**	469378.794
sssd22-08persp	373.8	508669.238	229.4	508673.475	508713.688
sssd25-04persp	37.1	300147.019	37.0	300165.359	300176.551
sssd25-08persp	173.9	472060.649	198.8	472047.332	472092.831

Table 2: Dual bounds and times for `sssd` problems for orders $r = 2$ and $r = 3$. In **bold**: Improved dual bounds compared to the bounds from the MINLPLib (last column).

added to the subsets of constraints and used for calculating better separating hyperplanes. In general, a combination with further methods for MINLPs might be a promising next step.

Acknowledgments: The first author received partial funding from the German Federal Environmental Foundation. The first and the last author were supported by the City of Constance's Smart Green City program. It is part of the "Model Projects Smart Cities" funding program of the German Federal Ministry for Housing, Urban Development and Building. The third author acknowledges financial support from Högskolestiftelsen i Österbotten. Further, we would like to express our gratitude to our colleagues Moritz Link and Markus Schweighofer for very fruitful discussions.

References

[BS20] Sönke Behrends and Anita Schöbel. "Generating Valid Linear Inequalities for Nonlinear Programs via Sums of Squares". In: *Journal of Optimization Theory and Applications* 186.3 (2020), pp. 911–935.

[Bel+13] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. "Mixed-integer nonlinear optimization". In: *Acta Numer.* 22 (2013), pp. 1–131.

[Ber08] Dimitri P. Bertsekas. *Nonlinear Programming*. 2nd ed. Athena Scientific Optimization and Computation Series. Athena Scientific, Belmont, MA, 2008.

[Bez+17] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. "Julia: A fresh approach to numerical computing". In: *SIAM review* 59.1 (2017), pp. 65–98.

[BCM20] Daniel Bienstock, Chen Chen, and Gonzalo Muñoz. "Outer-product-free sets for polynomial optimization and oracle-based cuts". In: *Mathematical Programming* 183.1 (2020), pp. 105–148.

[DG86] Marco A. Duran and Ignacio E. Grossmann. "An outer-approximation algorithm for a class of mixed-integer nonlinear programs". In: *Mathematical Programming* 36.3 (1986), pp. 307–339.

[ESV25] Carl Eggen, Oliver Stein, and Stefan Volkwein. "Granularity for Mixed-Integer Polynomial Optimization Problems". In: *Journal of Optimization Theory and Applications* 205.2 (2025), p. 22.

[FL05] John Forrest and Robin Lougee-Heimer. *CBC User Guide*. 2005. URL: `https://www.coin-or.org/Cbc/`.

[GVA16] Bissan Ghaddar, Juan C. Vera, and Miguel F. Anjos. "A dynamic inequality generation scheme for polynomial programming". In: *Mathematical Programming* 156.1 (2016), pp. 21–57.

[HN10] J. William Helton and Jiawang Nie. "Semidefinite representation of convex sets". In: *Mathematical Programming* 122.1 (2010), pp. 21–64.

[Jer73] Robert G. Jeroslow. "There cannot be any algorithm for integer programming with quadratic constraints". In: *Operations Res.* 21 (1973). Mathematical programming and its applications, pp. 221–224.

[KM78] Ravindran Kannan and Clyde L. Monma. "On the Computational Complexity of Integer Programming Problems". In: *Optimization and Operations Research*. Ed. by Rudolf Henn, Bernhard Korte, and Werner Oettli. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 161–172.

[KBG25] Jan Kronqvist, David E. Bernal Neira, and Ignacio E. Grossmann. "50 years of mixed-integer nonlinear and disjunctive programming". In: *European Journal of Operational Research* (2025).

[KLW16] Jan Kronqvist, Andreas Lundell, and Tapio Westerlund. "The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming". In: *Journal of Global Optimization* 64.2 (2016), pp. 249–272.

[Las01] Jean B. Lasserre. "Global optimization with polynomials and the problem of moments". In: *SIAM J. Optim.* 11.3 (2001), pp. 796–817.

[Las09] Jean B. Lasserre. *Moments, Positive Polynomials and Their Applications*. Vol. 1. Series on Optimization and Its Applications. Imperial College Press, London, 2009.

[Las15] Jean B. Lasserre. *An Introduction to Polynomial and Semi-Algebraic Optimization*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2015.

[LN07] Jean B. Lasserre and Tim Netzer. “SOS approximations of nonnegative polynomials via simple high degree perturbations”. In: *Mathematische Zeitschrift* 256.1 (2007), pp. 99–112.

[Lau09] Monique Laurent. “Sums of squares, moment matrices and optimization over polynomials”. In: *Emerging applications of algebraic geometry*. Vol. 149. IMA Vol. Math. Appl. Springer, New York, 2009, pp. 157–270.

[LK22] Andreas Lundell and Jan Kronqvist. “Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in SHOT”. In: *Journal of Global Optimization* 82.4 (2022), pp. 863–896.

[LKW22] Andreas Lundell, Jan Kronqvist, and Tapio Westerlund. “The supporting hyperplane optimization toolkit for convex MINLP”. In: *Journal of Global Optimization* 84.1 (2022), pp. 1–41.

[MW21] Victor Magron and Jie Wang. *TSSOS: a Julia library to exploit sparsity for large-scale polynomial optimization*. 2021. arXiv: `2103.00915 [math.OC]`.

[MW23] Victor Magron and Jie Wang. *Sparse polynomial optimization—theory and practice*. Vol. 5. Series on Optimization and its Applications. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2023.

[McC76] Garth P. McCormick. “Computability of global solutions to factorable nonconvex programs: Part I —Convex underestimating problems”. In: *Mathematical Programming* 10.1 (1976), pp. 147–175.

[MIN25] MINLPLib. *A library of mixed-integer and continuous nonlinear programming instances*. 2025. URL: `https://www.minlplib.org`.

[MOS25] MOSEK ApS. *MOSEK Optimizer API for Julia 11.1.3*. 2025. URL: `https://docs.mosek.com/latest/juliaapi/index.html`.

[Nie23] Jiawang Nie. *Moment and polynomial optimization*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2023.

[Put93] Mihai Putinar. “Positive Polynomials on Compact Semi-algebraic Sets”. In: *Indiana University Mathematics Journal* 42.3 (1993), pp. 969–984.

[Rud73] Walter Rudin. *Functional analysis.* McGraw-Hill Series in Higher Mathematics. McGraw-Hill Book Co., New York, 1973.

[Sch14] Rolf Schneider. *Convex bodies: the Brunn-Minkowski theory.* 2nd ed. Vol. 151. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 2014.

[Ste24] Oliver Stein. *Basic concepts of nonlinear optimization.* 2nd ed. Vol. 8. Mathematics Study Resources. Springer, Berlin, 2024.

[WB06] Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Math. Program.* 106.1 (2006), pp. 25–57.

[Wak+06] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. "Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity". In: *SIAM J. Optim.* 17.1 (2006), pp. 218–242.

[Wan+22] Jie Wang, Victor Magron, J. B. Lasserre, and Ngoc Hoang Anh Mai. "CS-TSSOS: correlative and term sparsity for large-scale polynomial optimization". In: *ACM Trans. Math. Software* 48.4 (2022), Art. 42, 26.

[WML22] Jie Wang, Victor Magron, and Jean B. Lasserre. "Certifying Global Optimality of AC-OPF Solutions via sparse polynomial optimization". In: *Electrical Power Systems Research* (2022).

[WML21a] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. "Chordal-TSSOS: a moment-SOS hierarchy that exploits term sparsity with chordal extension". In: *SIAM J. Optim.* 31.1 (2021), pp. 114–141.

[WML21b] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. "TSSOS: a moment-SOS hierarchy that exploits term sparsity". In: *SIAM J. Optim.* 31.1 (2021), pp. 30–58.

[WP95] Tapio Westerlund and Frank Pettersson. "An extended cutting plane method for solving convex MINLP problems". In: *Computers and Chemical Engineering* 19 (1995), pp. 131–136.

A Appendix

In Table A.1, the detailed results for the two approaches *single* and *cliques* from Section 7 are shown. In total, there are 352 instances from the MINLPLib. The columns of the table read as follows.

- *name*: name of instance in the MINLPLib,
- %: gap closed in per cent,
- *s*: total solving time in seconds,
- #*hyp*: number of hyperplanes calculated,
- *bound*: dual bound after termination of the algorithm (if the objective is minimized then it is a lower bound otherwise an upper bound),
- *init. bound*: dual bound of the initial MILP relaxation,
- *best bound*: dual bound reported in the MINLPLib.

Table A.1: Numerical results for `Sparse HYP-SOS` on instances from the MINLPLib

name	single				cliques				init bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
alan	99.9	33.6	18	2.925	99.9	34.6	18	2.925	-1.667	2.925
alkyl	95.8	36.4	91	-2.510	95.8	37.9	92	-2.501	-19.623	-1.765
bayes2_20	-	59.1	49	0.000	-	72.9	91	0.000	0.000	6.866e-14
bayes2_30	-	58.3	85	0.000	-	164.6	411	0.000	0.000	5.054e-11
bayes2_50	0.0	445.4	1304	0.000	0.0	1003.7	1353	0.000	0.000	0.520
blend029	1.1	45.2	27	20.268	39.2	136.7	131	17.604	20.348	13.359
blend146	0.0	1047.1	35	48.382	0.0	1019.8	35	48.382	48.382	45.297
blend480	0.0	832.0	44	11.641	0.0	829.0	44	11.641	11.642	9.227
blend531	33.3	1007.0	120	21.521	33.3	1015.9	102	21.521	22.262	20.039
blend718	7.0	1013.8	82	21.034	5.8	1021.0	90	21.215	22.067	7.394
blend721	0.0	1088.8	69	14.767	0.0	1047.7	43	14.767	14.767	13.527
blend852	0.0	1003.2	65	55.260	41.7	1012.0	70	54.719	55.260	53.963
camshape100	46.7	28.9	9	-5.305	47.2	29.0	9	-5.297	-6.034	-4.475
camshape200	56.1	27.7	11	-5.368	56.5	28.9	14	-5.363	-6.054	-4.832
camshape400	62.9	28.4	9	-5.403	63.0	28.4	10	-5.401	-6.065	-5.013
camshape800	70.3	28.2	13	-5.419	70.4	28.9	14	-5.418	-6.070	-5.144
chp_shorttermplan1a	99.5	1000.2	333	214.735	99.9	1000.0	372	214.824	193.577	214.839
clay0203m	99.9	32.8	56	41564.111	99.9	36.5	73	41538.447	0.000	41573.262
clay0204m	0.0	31.6	37	0.000	99.9	35.9	60	6545.000	0.000	6545.000
clay0205m	99.9	52.1	74	8092.500	99.9	63.3	87	8085.000	0.000	8092.500
clay0303m	99.9	35.1	67	26655.377	99.9	36.2	60	26651.035	0.000	26669.110
clay0304m	99.9	52.9	95	40258.297	99.9	71.3	117	40261.148	0.000	40262.387
clay0305m	99.9	51.8	66	8085.000	99.9	68.2	69	8085.000	0.000	8092.500
crudeoil_li01	25.7	1000.0	518	5229.578	12.7	1000.0	348	5246.004	5262.000	5136.306
crudeoil_li03	56.7	1000.1	183	3553.181	57.4	1000.0	132	3552.722	3586.677	3527.602
crudeoil_li05	17.6	1000.0	141	3487.000	7.2	1001.1	133	3516.877	3537.527	3250.980
crudeoil_li06	99.9	779.7	38	3355.000	100.0	1001.7	88	3355.000	3375.000	3355.000
cvxnonsep_normcon20r	99.9	28.1	62	-21.764	99.9	27.9	62	-21.764	-41.350	-21.749
cvxnonsep_normcon30r	99.9	28.2	101	-34.265	99.9	28.2	101	-34.265	-82.300	-34.244
cvxnonsep_normcon40r	99.9	29.1	134	-32.661	99.9	29.4	134	-32.661	-89.400	-32.630
dispatch	99.9	27.9	29	3155.287	99.9	29.3	46	3155.288	2978.167	3155.288
elf	0.0	32.1	28	0.000	0.0	30.9	6	0.000	0.000	0.192
ex14_1_1	94.6	27.8	22	-40.745	99.9	27.7	10	-5.549e-09	-758.000	-7.994e-15
ex14_1_2	0.0	27.8	13	-1.000	99.9	28.7	23	-1.469e-07	-1.000	0.000
ex14_1_5	0.0	27.9	1	-31.000	100.0	27.9	3	2.546e-08	-31.000	-8.000e-11
ex14_1_6	67.0	33.6	70	-0.321	100.0	41.8	116	9.479e-10	-0.974	0.000
ex2_1_1	99.1	27.8	15	-18.900	99.8	29.7	78	-17.458	-250.000	-17.000
ex2_1_2	99.9	28.6	19	-213.000	99.9	28.3	15	-213.000	-228.000	-213.000

name	single				cliques				init. bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
ex2_1_3	100.0	175.3	62	-15.000	100.0	218.7	67	-15.000	-35.000	-15.000
ex2_1_4	99.9	29.2	6	-11.000	99.9	28.2	16	-11.000	-19.500	-11.000
ex2_1_5	98.3	40.4	23	-269.453	100.0	57.2	49	-268.015	-355.000	-268.015
ex2_1_6	98.8	37.7	25	-44.400	99.9	74.7	87	-39.006	-500.000	-39.000
ex2_1_7	56.0	1024.9	15	-76410.768	25.9	1050.9	13	-125734.370	-168420.000	-4150.410
ex2_1_8	0.0	226.3	1	-63482.000	0.0	529.7	3	-63482.000	-63482.000	15639.000
ex2_1_9	91.5	58.9	58	-2.200	99.9	85.0	112	-0.375	-22.000	-0.375
ex3_1_1	17.1	28.0	34	2946.588	92.8	29.3	38	6695.138	2100.000	7049.248
ex3_1_2	98.7	28.7	25	-30685.400	99.9	27.9	25	-30665.543	-32217.431	-30665.539
ex3_1_3	86.2	29.0	16	-440.000	100.0	29.0	23	-310.000	-1256.000	-310.000
ex3_1_4	0.0	27.8	1	-6.000	70.1	28.4	24	-4.598	-6.000	-4.000
ex4_1_1	0.0	28.1	1	-335474.930	0.0	27.2	1	-335474.930	-335474.930	-7.487
ex4_1_3	0.0	27.2	1	-16997.183	0.0	27.2	1	-16997.183	-16997.183	-443.672
ex4_1_4	98.8	27.3	13	-5.619	99.9	28.2	20	-0.001	-500.000	-1.000e-09
ex4_1_6	0.0	27.8	1	-9125.000	0.0	27.4	1	-9125.000	-9125.000	7.000
ex4_1_8	99.9	28.0	12	-16.739	99.9	27.4	13	-16.739	-26.000	-16.739
ex4_1_9	22.3	27.8	2	-6.667	22.3	27.3	2	-6.667	-7.000	-5.508
ex5_2_2_case1	0.1	27.4	12	-2097.301	0.2	28.4	15	-2095.911	-2100.000	-400.000
ex5_2_2_case2	0.0	27.3	5	-2699.398	0.0	27.6	9	-2698.408	-2700.000	-600.000
ex5_2_2_case3	0.1	28.0	12	-2097.301	0.3	27.5	16	-2095.911	-2100.000	-750.000
ex5_2_4	44.1	30.9	46	-2933.333	99.9	31.7	45	-450.011	-4900.000	-450.000
ex5_3_2	0.0	29.0	65	0.998	0.0	28.8	31	0.998	0.998	1.864
ex5_3_3	51.5	1000.2	1979	1.346	59.2	1000.2	2857	1.524	0.146	2.473
ex5_4_2	20.5	27.9	37	3214.844	97.9	28.8	34	7399.897	2100.000	7512.230
ex7_3_1	0.0	27.4	1	0.000	0.0	28.7	1	0.000	0.000	0.342
ex7_3_2	0.0	28.4	1	0.000	0.0	28.1	1	3.326e-08	0.000	1.090
ex7_3_3	0.0	27.8	2	0.000	0.0	27.5	1	0.000	0.000	0.818
ex7_3_4	71.4	27.2	1	4.485	71.4	27.6	1	4.485	0.000	6.275
ex7_3_5	0.0	27.5	2	0.000	0.0	28.2	4	0.000	0.000	1.207
ex8_3_2	0.0	63.1	51	-10.000	0.0	75.0	50	-10.000	-10.000	-0.580
ex8_3_3	0.0	65.3	51	-10.000	0.0	74.8	50	-10.000	-10.000	-0.580
ex8_3_4	0.0	155.5	104	-10.000	0.0	167.9	101	-10.000	-10.000	-5.800
ex8_3_5	0.0	69.0	50	-10.000	0.0	76.3	47	-10.000	-10.000	-1.000
ex8_3_8	0.0	205.3	139	-10.000	0.0	224.5	148	-10.000	-10.000	-5.700
ex8_3_9	0.0	246.1	168	-10.000	0.0	334.5	202	-10.000	-10.000	-1.000
ex8_4_1	99.7	42.4	252	0.298	99.8	693.8	3472	0.448	-145.650	0.619
ex8_4_2	99.6	47.9	554	-0.000	99.6	104.4	429	-0.000	-145.650	0.449
ex9_1_2	0.0	27.5	1	-17.000	0.0	28.4	1	-17.000	-17.000	-16.000
ex9_1_4	0.0	27.2	1	-63.000	0.0	27.3	1	-63.000	-63.000	-37.000
ex9_1_5	-	27.9	1	-1.000	-	27.5	1	-1.000	-1.000	-1.000
ex9_1_8	-	18.2	0	-3.250	-	18.2	0	-3.250	-3.250	-3.250

name	single				cliques				init. bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
ex9_2_2	77.7	27.5	8	55.556	79.9	28.1	10	59.832	-100.000	100.000
ex9_2_3	0.0	27.7	1	-30.000	0.0	28.0	1	-30.000	-30.000	0.000
ex9_2_4	99.9	27.7	19	-0.016	99.9	28.3	19	-0.002	-796.000	0.500
ex9_2_5	90.3	28.0	19	-3.856e-05	90.3	28.2	26	-4.844e-05	-47.000	5.000
ex9_2_6	99.8	27.8	21	-1.492	50.6	30.4	10	-200.756	-405.500	-1.000
ex9_2_7	53.1	27.5	4	2.000	53.1	27.4	5	2.000	-15.000	17.000
fuel	99.8	30.1	16	8558.896	99.9	31.0	23	8562.775	4818.750	8566.119
gabriel01	0.0	1019.2	70	48.382	0.0	1000.5	69	48.382	48.382	45.244
gabriel02	0.0	1028.0	81	48.382	0.0	1000.2	80	48.382	48.382	39.610
gabriel04	18.2	1001.9	107	11.200	18.2	1001.0	107	11.200	11.642	9.227
gabriel05	12.0	1006.4	35	51.777	12.0	1001.3	35	51.777	51.910	50.808
gasprod_sarawak01	0.0	33.8	38	-61684.694	0.0	37.0	38	-61685.211	-61696.133	-32445.405
gasprod_sarawak16	99.7	1000.0	1015	-32410.766	99.7	1000.0	719	-32411.453	-61482.390	-32349.506
genpooling_lee1	27.3	108.9	334	-5894.868	27.5	735.5	1455	-5891.614	-6366.480	-4640.082
genpooling_lee2	19.6	67.1	97	-5872.748	19.6	69.1	100	-5872.962	-6366.480	-3849.265
genpooling_meyer04	2.2	1007.6	68	399376.350	2.2	1011.1	68	399376.350	383835.877	1.086e+06
haverly	0.0	28.2	4	-2100.000	0.0	27.8	3	-2100.000	-2100.000	-400.000
himmel11	98.7	29.1	39	-30685.378	98.7	28.5	31	-30685.372	-32217.431	-30665.539
hybriddynamic_fixed	99.9	30.8	13	1.473	99.9	69.2	33	1.474	-1.222	1.474
hydro	99.9	28.0	83	4.367e+06	99.9	101.6	104	4.367e+06	3.814e+06	4.367e+06
hydroenergy1	99.1	68.9	224	214353.278	99.5	286.5	324	212522.552	780781.922	209728.760
hydroenergy2	99.5	196.4	371	379491.629	99.6	1000.0	365	378393.147	1.238e+06	375370.080
hydroenergy3	99.1	1000.0	364	770434.949	91.7	1000.2	233	868479.904	2.085e+06	758703.371
kall_circles_c6a	0.0	42.7	404	0.000	0.0	1001.7	318	0.000	0.000	2.112
kall_circles_c6b	0.0	37.5	287	0.000	0.0	1010.2	308	0.000	0.000	1.974
kall_circles_c6c	0.0	76.5	1156	0.000	0.0	1002.9	115	0.000	0.000	1.770
kall_circles_c7a	0.0	33.6	153	0.000	0.0	1003.9	105	0.000	0.000	2.663
kall_circles_c8a	0.0	33.8	173	0.000	0.0	1014.2	44	0.000	0.000	2.541
kall_circlespolygons_c1p11	99.9	33.7	142	0.200	99.9	67.3	312	0.200	0.000	0.200
kall_circlespolygons_c1p12	0.0	31.4	109	0.000	0.0	55.3	276	0.000	0.000	0.340
kall_circlespolygons_c1p13	0.0	30.4	96	0.000	0.0	47.6	221	0.000	0.000	0.340
kall_circlesrectangles_c1r11	99.9	37.6	145	0.200	99.9	47.2	343	0.200	0.000	0.200
kall_circlesrectangles_c1r12	0.0	38.6	157	0.000	0.0	46.5	258	0.000	0.000	0.340
kall_circlesrectangles_c1r13	0.0	31.3	109	0.000	0.0	41.3	242	0.000	0.000	0.215
kall_circlesrectangles_c6r1	0.0	524.1	1567	0.000	0.0	1000.6	7842	0.000	0.000	4.388
kall_circlesrectangles_c6r29	-	1000.0	9106	0.000	-	669.5	9999	0.000	0.000	0.000
kall_circlesrectangles_c6r39	-	1000.1	9190	0.000	-	667.0	9999	0.000	0.000	0.000
kall_congruentcircles_c31	0.0	29.6	121	0.000	0.0	38.9	279	0.000	0.000	0.644
kall_congruentcircles_c32	0.0	27.6	15	0.000	6.4	30.5	72	0.089	0.000	1.376
kall_congruentcircles_c41	0.0	27.7	4	1.293e-09	81.0	29.4	49	0.695	0.000	0.858
kall_congruentcircles_c42	0.0	28.3	35	0.000	0.0	46.0	155	0.000	0.000	0.858

name	single				cliques				init. bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
kall_congruentcircles_c51	0.0	35.2	334	0.000	0.0	272.8	580	0.000	0.000	1.073
kall_congruentcircles_c52	0.0	28.7	18	0.000	0.0	106.7	203	0.000	0.000	1.537
kall_congruentcircles_c61	0.0	40.4	366	0.000	0.0	477.6	241	0.000	0.000	1.288
kall_congruentcircles_c62	0.0	28.9	27	0.000	0.0	558.3	432	0.000	0.000	1.288
kall_congruentcircles_c63	0.0	28.7	26	0.000	0.0	566.1	428	0.000	0.000	1.288
kall_congruentcircles_c71	0.0	45.5	441	0.000	0.0	1000.6	299	0.000	0.000	1.502
kall_congruentcircles_c72	0.0	32.1	114	0.000	0.0	1002.2	318	0.000	0.000	1.966
kall_diffcircles_10	0.0	42.4	235	2.730e-08	0.0	1002.9	26	0.000	0.000	11.935
kall_diffcircles_5a	39.9	29.7	49	2.042	40.7	268.9	422	2.085	0.000	5.116
kall_diffcircles_5b	0.0	34.9	294	0.000	4.3	471.7	1106	0.221	0.000	5.116
kall_diffcircles_6	0.0	33.9	135	0.000	48.9	1000.4	784	3.813	0.000	7.788
kall_diffcircles_8	0.0	50.1	279	1.553e-08	0.0	1000.5	162	0.000	0.000	14.481
kall_diffcircles_9	0.0	46.9	321	1.773e-08	0.0	1015.3	60	0.000	0.000	13.350
knp3-12	0.0	39.0	124	12.000	0.0	40.1	135	12.000	12.000	2.281
knp4-24	0.0	486.3	504	16.000	0.0	847.8	518	16.000	16.000	3.676
knp5-40	0.0	1075.9	653	20.000	0.0	1140.5	695	20.000	20.000	4.000
knp5-41	0.0	1018.4	667	20.000	0.0	1026.7	735	20.000	20.000	4.000
knp5-42	0.0	1089.8	692	20.000	0.0	1151.6	711	20.000	20.000	4.000
knp5-43	0.0	1042.3	698	20.000	0.0	1098.3	750	20.000	20.000	4.000
knp5-44	0.0	1020.6	713	20.000	0.0	1046.9	790	20.000	20.000	4.000
nous1	91.1	1000.4	1794	-0.728	90.8	1002.3	1139	-0.792	-24.216	1.567
nous2	93.9	1000.0	1740	-0.873	91.0	1000.2	1054	-1.605	-24.216	0.626
nvs02	0.3	30.4	74	5.921	100.0	31.3	100	5.964	5.921	5.964
nvs03	71.9	27.8	3	-28.000	100.0	27.7	5	16.000	-140.667	16.000
nvs07	0.0	19.9	1	3.000	0.0	19.9	1	3.000	3.000	4.000
nvs10	99.9	27.9	30	-329.812	99.9	27.9	36	-316.621	-23080.000	-310.800
nvs11	99.9	28.1	43	-503.269	100.0	28.3	72	-431.000	-274400.000	-431.000
nvs12	99.9	29.2	110	-481.200	84.2	27.8	9	-45082.385	-283400.000	-481.200
nvs13	100.0	32.1	133	-585.200	100.0	32.7	128	-585.200	-760480.000	-585.200
nvs14	8.0	29.7	85	-40757.280	100.0	29.9	87	-40358.118	-40792.141	-40358.155
nvs17	37.8	28.7	17	-987419.384	99.9	104.2	377	-1100.800	-1.588e+06	-1100.400
nvs18	99.9	41.4	204	-778.996	99.8	30.2	35	-2516.328	-1.168e+06	-778.400
nvs19	99.9	102.4	384	-1099.400	99.9	158.9	433	-1099.400	-2.070e+06	-1098.400
nvs21	0.0	27.5	1	-2.573e+07	0.0	27.4	1	-2.573e+07	-2.573e+07	-5.685
nvs23	99.9	155.1	444	-1128.200	99.9	308.1	493	-1129.200	-2.638e+06	-1125.200
nvs24	89.5	38.6	34	-367024.741	99.9	690.1	706	-1036.793	-3.514e+06	-1033.200
orth_d3m6	0.0	40.2	128	0.000	0.0	1002.6	419	0.000	0.000	0.707
p_ball_10b_5p_2d_m	99.9	61.8	79	18.707	99.9	415.5	248	18.703	0.000	18.719
p_ball_10b_5p_3d_m	99.9	214.6	148	43.962	84.2	1001.0	246	37.094	0.000	44.004
p_ball_10b_5p_4d_m	99.8	1000.0	221	71.298	0.0	1004.7	129	0.000	0.000	71.372
p_ball_10b_7p_3d_m	91.7	1000.0	118	100.728	0.0	1000.8	282	0.000	0.000	109.803

name	single				cliques				init. bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
p_ball_15b_5p_2d_m	99.9	117.4	132	6.598	0.0	1005.9	106	0.000	0.000	6.600
p_ball_20b_5p_2d_m	99.9	133.1	81	2.435	0.0	1116.1	14	0.000	0.000	2.437
p_ball_20b_5p_3d_m	95.6	1000.1	139	18.871	95.6	1000.0	139	18.871	0.000	19.736
p_ball_30b_10p_2d_m	0.0	1000.2	118	0.000	0.0	1000.2	118	0.000	0.000	19.512
p_ball_30b_5p_2d_m	98.9	1000.0	131	0.289	98.9	1000.0	131	0.289	0.000	0.292
p_ball_30b_5p_3d_m	95.8	1000.0	144	7.873	93.6	1000.0	187	7.699	0.000	8.218
p_ball_30b_7p_2d_m	24.4	1000.9	123	3.412	23.5	1001.0	123	3.277	0.000	13.934
p_ball_40b_5p_3d_m	71.8	1000.2	110	7.027	71.8	1000.1	110	7.027	0.000	9.777
p_ball_40b_5p_4d_m	74.8	1000.1	173	22.559	74.8	1000.0	173	22.559	0.000	30.133
pointpack02	0.0	19.7	1	4.000	0.0	19.8	1	4.000	4.000	2.000
pointpack04	90.0	27.9	27	1.250	0.0	32.1	9	3.500	3.500	1.000
pointpack06	71.6	28.4	36	1.250	0.0	112.6	14	3.500	3.500	0.361
pointpack08	69.6	29.5	50	1.250	0.0	1064.6	16	3.500	3.500	0.268
pointpack10	68.5	33.1	68	1.250	0.0	1020.0	13	3.500	3.500	0.216
pointpack14	68.6	34.7	82	1.250	68.6	32.9	82	1.250	3.500	0.224
pooling_adhya1pq	5.1	28.3	35	-840.316	5.1	28.9	34	-840.489	-856.251	-549.803
pooling_adhya1stp	5.1	31.4	78	-840.316	5.1	48.1	106	-840.489	-856.251	-549.803
pooling_adhya1tp	0.0	29.2	49	-856.251	0.0	33.3	41	-856.251	-856.251	-549.803
pooling_adhya2pq	0.0	27.9	21	-574.783	0.0	28.3	25	-574.783	-574.783	-549.803
pooling_adhya2stp	0.0	29.4	50	-574.783	0.0	37.2	67	-574.783	-574.783	-549.803
pooling_adhya2tp	0.0	28.1	29	-574.783	0.0	39.3	40	-574.783	-574.783	-549.803
pooling_adhya3pq	0.0	27.7	21	-574.783	0.0	28.3	25	-574.783	-574.783	-561.045
pooling_adhya3stp	0.0	29.2	50	-574.783	0.0	35.6	62	-574.783	-574.783	-561.045
pooling_adhya3tp	0.0	28.0	29	-574.783	0.0	39.7	36	-574.783	-574.783	-561.045
pooling_adhya4pq	14.4	29.1	50	-962.127	13.8	32.0	81	-962.734	-976.439	-877.646
pooling_adhya4stp	14.4	31.9	95	-962.127	13.8	129.8	368	-962.734	-976.439	-877.646
pooling_adhya4tp	0.0	28.3	41	-976.439	0.0	36.3	122	-976.439	-976.439	-877.646
pooling_bental4pq	24.1	27.6	23	-525.822	24.2	29.2	43	-525.728	-550.000	-450.000
pooling_bental4stp	24.1	28.5	36	-525.822	24.9	29.2	42	-525.000	-550.000	-450.000
pooling_bental4tp	14.4	27.4	12	-535.566	24.9	27.6	12	-525.000	-550.000	-450.000
pooling_bental5pq	-	32.3	152	-3500.000	-	40.9	192	-3500.000	-3500.000	-3500.000
pooling_bental5stp	-	95.0	716	-3500.000	-	916.8	908	-3500.000	-3500.000	-3500.000
pooling_bental5tp	-	28.8	119	-3500.000	-	992.9	326	-3500.000	-3500.000	-3500.000
pooling_foulds2pq	-	27.8	17	-1100.000	-	28.5	9	-1100.000	-1100.000	-1100.000
pooling_foulds2stp	-	28.3	34	-1100.000	-	30.7	33	-1100.000	-1100.000	-1100.000
pooling_foulds2tp	-	27.5	17	-1100.000	-	28.9	17	-1100.000	-1100.000	-1100.000
pooling_foulds3pq	-	596.0	2866	-8.000	-	775.2	3185	-8.000	-8.000	-8.000
pooling_foulds3stp	-	1000.0	5014	-8.000	-	1000.0	4221	-8.000	-8.000	-8.000
pooling_foulds3tp	-	723.4	6085	-8.000	-	1000.9	447	-8.000	-8.000	-8.000
pooling_foulds4pq	-	457.7	2176	-8.000	-	1000.0	3968	-8.000	-8.000	-8.000
pooling_foulds4stp	-	1000.2	5131	-8.000	-	1000.1	3688	-8.000	-8.000	-8.000

name	single				cliques				init. bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
pooling_foulds4tp	-	820.9	6555	-8.000	-	1002.7	318	-8.000	-8.000	-8.000
pooling_foulds5pq	-	1001.3	3425	-8.000	-	1001.1	1939	-8.000	-8.000	-8.000
pooling_foulds5stp	-	1000.2	4233	-8.000	-	1000.9	2362	-8.000	-8.000	-8.000
pooling_foulds5tp	-	1000.0	6746	-8.000	-	1000.0	6633	-8.000	-8.000	-8.000
pooling_haverly1pq	0.0	28.2	8	-500.000	0.0	28.1	7	-500.000	-500.000	-400.000
pooling_haverly1stp	0.0	28.2	20	-500.000	0.0	27.6	16	-500.000	-500.000	-400.000
pooling_haverly1tp	0.0	27.8	13	-500.000	0.0	27.4	10	-500.000	-500.000	-400.000
pooling_haverly2pq	0.0	27.9	8	-1000.000	0.0	27.5	7	-1000.000	-1000.000	-600.000
pooling_haverly2stp	0.0	27.4	17	-1000.000	0.0	27.8	15	-1000.000	-1000.000	-600.000
pooling_haverly2tp	0.0	27.8	10	-1000.000	0.0	27.5	9	-1000.000	-1000.000	-600.000
pooling_haverly3pq	59.8	28.0	19	-800.172	59.9	27.7	15	-800.037	-875.000	-750.000
pooling_haverly3stp	59.8	28.4	45	-800.172	59.9	28.2	22	-800.036	-875.000	-750.000
pooling_haverly3tp	0.0	27.8	13	-875.000	21.5	28.1	17	-848.011	-875.000	-750.000
pooling_rt2pq	0.0	28.3	44	-6034.871	0.0	28.0	30	-6034.871	-6034.871	-4391.826
pooling_rt2stp	29.9	34.3	135	-5542.652	28.5	34.1	147	-5566.553	-6034.871	-4391.826
pooling_rt2tp	29.9	30.0	75	-5542.654	28.5	30.5	111	-5566.554	-6034.871	-4391.826
powerflow0009r	0.0	45.6	165	1188.750	0.0	73.8	176	1188.750	1188.750	5296.683
powerflow0014r	0.0	39.5	69	0.000	0.0	84.1	118	0.000	0.000	8080.331
powerflow0030r	0.0	1000.4	1764	0.000	0.0	1000.3	819	0.000	0.000	575.216
powerflow0039r	0.0	479.3	948	2.000	0.0	1000.9	846	2.000	2.000	41265.707
powerflow0057r	0.0	321.8	345	0.000	0.0	795.4	432	0.000	0.000	41871.902
powerflow0118r	0.0	1002.4	576	0.000	0.0	1007.2	205	0.000	0.000	71946.606
prob02	100.0	28.3	6	112235.000	99.9	27.8	11	112234.998	48000.000	112235.000
prob03	78.4	27.2	3	8.920	78.4	27.7	3	8.920	5.000	10.000
prob06	36.3	27.5	11	1.064	100.0	27.9	5	1.177	1.000	1.177
ringpack_10_1	-	130.9	270	-20.858	-	264.8	555	-20.858	-20.858	-20.858
ringpack_10_2	-	128.5	488	-20.858	-	281.7	894	-20.858	-20.858	-20.858
sep1	2.4	30.2	8	-718.307	14.6	30.8	19	-692.137	-723.500	-510.081
sfacloc1_2_95	0.0	46.4	166	0.000	0.0	37.3	98	0.000	0.000	15.773
sfacloc1_3_90	0.0	1000.0	3856	0.000	0.0	1000.0	4293	0.000	0.000	4.235
sfacloc1_3_95	0.0	1000.0	2730	0.000	0.0	1003.1	3000	0.000	0.000	4.998
sfacloc1_4_80	0.0	1000.0	2052	0.000	0.0	1000.0	2059	0.000	0.000	0.235
sfacloc1_4_90	0.0	1000.0	4172	0.000	0.0	1000.0	4017	0.000	0.000	1.182
sfacloc1_4_95	0.0	1000.0	2012	0.000	0.0	1000.3	1805	0.000	0.000	1.337
sfacloc2_2_90	98.6	113.9	145	18.349	98.6	109.6	128	18.348	0.000	18.594
sfacloc2_2_95	99.9	44.0	58	19.558	99.9	49.0	68	19.565	0.000	19.578
sfacloc2_3_90	99.0	539.7	195	14.958	99.0	801.1	245	14.958	0.000	15.095
sfacloc2_3_95	99.9	173.4	80	16.136	99.9	276.1	107	16.139	0.000	16.151
sfacloc2_4_90	98.8	1000.0	210	13.255	81.1	1000.1	282	10.882	0.000	13.412
sfacloc2_4_95	99.9	273.4	101	14.291	99.9	509.3	156	14.288	0.000	14.299
slay04h	99.9	35.3	48	9850.552	99.9	35.2	48	9850.552	-651095.000	9859.660

name	single				cliques				init. bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
slay04m	99.9	32.6	52	9851.607	99.9	32.7	52	9851.607	-695525.000	9859.660
slay05h	99.9	41.1	63	22646.113	99.9	40.9	63	22646.113	-850097.421	22664.679
slay05m	99.9	33.9	62	22645.772	99.9	34.1	62	22645.772	-921350.000	22664.679
slay06h	99.9	48.9	73	32726.823	99.9	48.7	73	32726.823	-945504.901	32757.020
slay06m	99.9	35.9	72	32731.629	99.9	36.2	72	32731.629	-1.026e+06	32757.020
slay07h	99.9	69.7	81	64688.238	99.9	69.4	81	64688.238	-1.879e+06	64748.825
slay07m	99.9	42.6	78	64691.874	99.9	42.8	78	64691.874	-1.982e+06	64748.825
slay08h	99.9	79.8	89	84878.433	99.9	79.5	89	84878.433	-2.560e+06	84960.212
slay08m	99.9	56.2	92	84887.469	99.9	56.2	92	84887.469	-2.729e+06	84960.212
slay09h	99.9	131.5	97	107703.493	99.9	130.3	97	107703.493	-3.049e+06	107805.753
slay09m	99.9	77.1	96	107699.402	99.9	76.5	96	107699.402	-3.267e+06	107805.753
slay10h	99.9	494.7	127	129461.177	99.9	494.8	127	129461.177	-3.349e+06	129579.884
slay10m	99.9	172.1	125	129456.698	99.9	174.9	125	129456.698	-3.585e+06	129579.884
squfl010-025persp	99.8	275.6	706	213.898	99.8	281.1	710	213.898	2.000	214.111
squfl010-040persp	99.8	357.1	1455	240.358	99.8	355.2	1455	240.361	3.000	240.599
squfl010-080persp	95.1	1000.0	2046	485.629	48.9	591.5	999	255.163	11.000	509.706
squfl015-060persp	72.1	1000.0	1470	264.651	47.4	765.2	999	174.470	1.000	366.622
squfl015-080persp	36.5	930.4	999	155.911	43.3	1000.0	1272	182.555	14.000	402.489
squfl020-040persp	71.7	517.6	999	151.777	71.7	520.5	999	151.929	6.000	209.255
squfl020-050persp	46.3	751.5	999	110.000	57.0	1000.0	1287	133.842	6.000	230.202
squfl025-025persp	99.8	983.8	1260	168.639	99.8	983.8	1266	168.639	7.000	168.807
squfl025-030persp	91.5	1000.0	1138	188.303	93.3	1000.0	1163	192.069	2.000	205.502
squfl025-040persp	57.3	1000.1	1122	115.681	59.8	1000.0	1168	120.544	6.000	197.334
sssd08-04persp	0.1	30.9	13	1579.844	10.4	31.0	12	20147.506	1316.512	182022.570
sssd12-05persp	0.1	31.3	14	2537.764	0.2	31.4	18	2823.800	2234.320	281408.465
sssd15-04persp	52.9	210.9	53	109898.521	54.6	32.2	12	113304.998	2940.792	205054.353
sssd15-06persp	7.1	33.8	14	41221.089	15.0	38.6	26	83614.774	3073.900	539635.249
sssd15-08persp	4.8	39.5	22	29852.913	7.1	40.1	32	42543.573	2687.317	562617.855
sssd16-07persp	52.8	39.6	23	221347.945	52.7	38.1	22	221047.437	1891.191	417188.774
sssd18-06persp	3.1	36.0	21	15356.594	7.4	35.7	24	32545.317	3087.230	397992.136
sssd18-08persp	9.2	44.6	19	80041.081	12.1	44.6	21	104383.593	3591.231	831802.891
sssd20-04persp	50.6	1011.8	56	177888.787	56.2	1006.5	204	197156.109	3801.495	347691.293
sssd20-08persp	0.6	52.8	23	6726.419	56.4	55.0	37	266609.377	3650.118	469378.794
sssd22-08persp	0.4	59.8	24	5926.539	7.8	59.3	30	43235.328	3437.371	508713.688
sssd25-04persp	49.6	34.8	14	151593.931	50.7	35.3	14	154948.867	5084.906	300176.551
sssd25-08persp	0.1	72.6	26	3689.573	5.2	73.9	41	27816.592	3163.631	472092.831
st_bpaf1a	98.4	61.1	69	-46.084	99.9	94.1	116	-45.380	-90.417	-45.380
st_bpaf1b	99.4	63.6	71	-43.236	99.9	105.0	132	-42.976	-92.417	-42.963
st_e01	99.9	28.0	1	-6.667	99.9	32.1	1	-6.667	-10.000	-6.667
st_e02	71.2	27.5	16	186.470	71.2	27.6	11	186.470	150.000	201.159
st_e03	98.1	31.6	72	-1494.878	93.6	30.1	35	-2334.183	-19617.865	-1161.337

name	single				cliques				init. bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
st_e05	95.6	27.7	20	6853.242	96.4	27.9	20	6889.413	2500.000	7049.249
st_e07	32.3	28.0	13	-1550.000	52.9	27.8	14	-1200.000	-2100.000	-400.000
st_e08	95.3	27.3	10	0.707	100.0	27.5	4	0.742	0.000	0.742
st_e09	55.5	27.2	7	-1.167	99.9	27.2	10	-0.500	-2.000	-0.500
st_e13	99.9	30.4	2	2.000	99.9	30.3	2	2.000	0.000	2.000
st_e18	99.9	27.2	15	-2.828	99.9	27.6	14	-2.828	-4.000	-2.828
st_e22	55.8	27.2	3	-104.000	99.9	27.3	7	-85.000	-128.000	-85.000
st_e23	77.3	27.5	7	-2.375	99.9	27.8	11	-1.084	-6.792	-1.083
st_e24	88.8	27.8	5	-5.944e-06	99.9	27.3	6	3.000	-24.000	3.000
st_e25	81.4	27.7	8	0.729	100.0	27.6	2	0.890	0.020	0.890
st_e26	88.9	27.1	3	-198.422	99.9	27.2	5	-185.779	-300.259	-185.779
st_e27	0.0	30.2	6	-36.444	99.1	30.5	8	1.677	-36.444	2.000
st_e28	98.7	28.5	39	-30685.378	98.7	28.3	31	-30685.372	-32217.431	-30665.539
st_e30	0.0	27.8	15	-3.000	0.0	27.9	11	-3.000	-3.000	-1.581
st_e31	0.0	31.5	41	-3.000	0.0	32.7	69	-3.000	-3.000	-2.000
st_e33	37.3	28.1	18	-1465.524	52.9	27.8	16	-1200.008	-2100.000	-400.000
st_e34	99.9	27.8	2	0.016	99.9	27.6	2	0.016	0.000	0.016
st_e42	100.0	28.0	2	18.784	100.0	29.9	13	18.784	0.000	18.784
st_glmp_fp1	73.0	27.4	4	-147.787	73.0	27.3	4	-147.787	-575.000	10.000
st_glmp_fp2	32.2	27.6	3	-28.096	32.2	27.6	3	-28.096	-45.000	7.345
st_glmp_fp3	54.5	27.6	2	-106.572	54.5	27.5	2	-106.572	-220.000	-12.000
st_ph1	94.6	27.6	6	-243.811	99.9	29.1	21	-230.117	-486.646	-230.117
st_ph10	18.3	27.5	4	-15.399	87.2	27.7	5	-11.266	-16.500	-10.500
st_ph11	96.3	27.5	8	-11.750	99.9	27.7	11	-11.287	-24.000	-11.281
st_ph12	96.5	27.5	9	-23.500	99.9	28.0	12	-22.626	-48.000	-22.625
st_ph13	96.3	27.4	8	-11.750	99.9	28.1	12	-11.291	-24.000	-11.281
st_ph14	93.0	27.5	6	-231.000	99.8	27.6	12	-229.758	-248.000	-229.722
st_ph15	91.8	27.3	5	-574.153	99.9	27.9	15	-392.822	-2628.125	-392.704
st_ph2	96.2	28.2	6	-1064.496	99.9	29.2	17	-1028.118	-2004.885	-1028.117
st_ph3	93.7	27.6	5	-447.849	93.7	27.6	5	-447.849	-862.879	-420.235
supplychain	99.9	31.4	29	2259.833	99.9	31.1	29	2259.833	947.600	2260.257
tloss	-	45.4	19	16.300	-	58.5	32	16.300	16.300	16.300
tltr	54.7	1006.0	93	26.337	54.3	1000.0	94	26.133	-0.000	48.067
wastewater02m1	51.1	31.9	74	66.874	53.3	34.2	95	69.680	0.000	130.703
wastewater02m2	0.0	27.5	5	78.000	0.0	27.7	3	78.000	78.000	130.703
wastewater04m1	37.9	37.4	140	34.108	35.2	54.4	256	31.681	0.000	89.836
wastewater04m2	0.0	27.5	15	69.239	0.0	30.1	19	69.239	69.239	89.836
wastewater05m1	9.1	90.1	202	21.050	0.0	1000.0	1053	0.011	0.000	229.701
wastewater05m2	0.0	28.6	46	99.898	0.0	74.3	259	99.898	99.898	229.701
wastewater11m1	0.0	1028.4	21	0.000	0.0	1045.9	21	0.000	0.000	2127.115
wastewater11m2	0.0	29.6	94	1024.800	0.0	36.6	49	1024.800	1024.800	1492.570

name	single				cliques				init. bound	best bound
	%	s	#hyp	bound	%	s	#hyp	bound		
wastewater12m1	0.0	20.2	1	0.000	0.0	20.0	1	0.000	0.000	1201.038
wastewater12m2	0.0	29.0	85	648.000	0.0	96.4	74	648.000	648.000	887.522
wastewater13m1	0.0	20.0	1	0.000	0.0	20.0	1	0.000	0.000	1564.958
wastewater13m2	0.0	31.2	155	1017.200	0.0	573.7	183	1017.200	1017.200	1192.022
wastewater14m1	16.2	1002.5	380	83.480	0.0	1000.6	167	0.000	0.000	512.950
wastewater14m2	0.0	28.3	14	337.654	0.0	28.3	9	337.654	337.654	403.912
wastewater15m1	26.2	660.8	1042	643.062	5.7	1000.7	1110	141.703	0.000	2446.429
wastewater15m2	0.0	29.1	35	1212.707	0.0	93.3	224	1212.707	1212.707	2446.429
waterno2_01	33.2	33.8	110	6.469	36.7	35.6	128	7.149	0.000	19.457
waterno2_02	33.4	68.9	302	13.233	37.0	109.7	398	14.654	0.000	39.571
waterno2_03	26.6	402.0	526	30.704	29.5	664.3	637	33.998	0.000	115.002
waterno2_04	34.7	1000.0	780	50.597	30.5	1000.0	802	44.405	0.000	145.439
waterno2_06	20.8	1000.3	742	22.653	20.8	1000.1	817	22.649	0.000	108.404
waterno2_09	20.2	1000.7	843	27.135	9.6	1002.0	948	12.889	0.000	134.224
waterno2_12	0.6	1000.0	911	1.937	0.0	1000.0	1081	0.000	0.000	280.554
waterund01	14.6	257.7	583	21.258	75.1	765.5	1332	67.673	10.000	86.789
waterund08	17.1	1002.3	338	28.141	18.6	1001.0	278	30.729	0.000	164.490
waterund11	17.8	1002.5	538	55.675	8.3	1000.6	586	50.006	45.000	104.886
waterund14	9.3	1000.1	395	39.880	4.2	1001.1	237	23.625	10.000	329.474
waterund17	9.5	1002.7	474	14.782	6.3	1001.5	487	9.927	0.000	155.536
waterund18	44.0	1000.5	850	121.335	63.3	857.2	692	161.269	30.000	237.143
waterund22	0.0	1006.6	90	10.000	0.0	1013.5	52	10.000	10.000	323.421
waterund25	0.0	1012.2	70	10.000	0.0	1009.7	42	10.000	10.000	395.144
waterund27	0.0	28.6	80	0.000	0.0	29.1	80	0.000	0.000	513.731
waterund28	0.0	33.5	348	0.000	0.0	37.1	348	0.000	0.000	1719.489
waterund32	0.0	30.5	190	0.000	0.0	32.3	190	0.000	0.000	330.269
waterund36	0.0	1020.6	26	35.000	0.0	1011.3	26	35.000	35.000	607.776