

SEMIDEFINITE RELAXATIONS FOR SEMI-INFINITE POLYNOMIAL PROGRAMMING

LI WANG AND FENG GUO

ABSTRACT. This paper studies how to solve semi-infinite polynomial programming (SIPP) problems by semidefinite relaxation method. We first introduce two SDP relaxation methods for solving polynomial optimization problems with finitely many constraints. Then we propose an exchange algorithm with SDP relaxations to solve SIPP problems with compact index set. At last, we extend the proposed method to SIPP problems with noncompact index set via homogenization. Numerical results show that the algorithm is efficient in practice.

1. INTRODUCTION

Consider the *semi-infinite polynomial programming* (SIPP) problem:

$$(P) : \begin{cases} f^* := \min_{x \in X} f(x) \\ \text{s.t. } g(x, u) \geq 0, \forall u \in U, \end{cases}$$

where

$$\begin{aligned} X &= \{x \in \mathbb{R}^n \mid \theta_1(x) \geq 0, \dots, \theta_{m_2}(x) \geq 0\}, \\ U &= \{u \in \mathbb{R}^p \mid h_1(u) \geq 0, \dots, h_{m_1}(u) \geq 0\}. \end{aligned}$$

Here $f(x), \theta_i(x)$ are polynomials in $x \in \mathbb{R}^n$, $h_j(u)$ are polynomials in $u \in \mathbb{R}^p$ and $g(x, u)$ is a polynomial in $(x, u) \in \mathbb{R}^n \times \mathbb{R}^p$. Throughout this paper, we assume that X is compact and U is an infinite index set, i.e., there are infinitely many constraints in (P) . The SIPP problem is a special subclass of the *semi-infinite programming* (SIP) which has many applications, e.g., Chebyshev approximation, maneuverability problems, some mathematical physics problems and so on [10, 16].

There are various algorithms for SIP problems based on discretization schemes of U , such as central cutting plane method [3], Newton's method [24], SQP methods [26] and the like. Most of algorithms for SIP problems, however, are only locally convergent or globally convergent under some strong assumptions, like convexity or linearity, and to the authors best knowledge, few of them are specially designed for SIPP problems exploiting features of polynomial optimization problems. Parpas and Rustem [22] proposed a discretization like method to solve min-max polynomial optimization problems, which can be reformulated as SIPP problems. Using a polynomial approximation and an appropriate hierarchy of semidefinite relaxations, Lasserre presented an algorithm to solve the *generalized* SIPP problems in [15].

Before introducing the contribution of this paper, we first review some of the considerable progress recently made in solving polynomial optimization problems with *finite* constraints via sums of squares relaxations, which are typically based on the Positivstellensatz [23]. We define a so-called quadratic module which is a set

of polynomials generated by the finitely many constraints, to which any polynomials positive over the feasible set belong. The classic Lasserre's hierarchy [13] is to compute the maximal real number, minus which the objective lies in the quadratic module. By increasing the order of the quadratic module, Lasserre's hierarchy results in a sequence of lower bounds of the global optimum and the asymptotical convergence is established under the Archimedean Condition. Interestingly, finite convergence of Lasserre's hierarchy is generic [19]. To guarantee the finite convergence of Lasserre's hierarchy, Nie [20] proposed a refined SDP relaxation by some "Jacobian-type" technique which represents optimality conditions of the considered polynomial optimization problem. More importantly, these SDP relaxation methods are global and the minimizers can be extracted if the flat extension condition [2] or more general, flat truncation condition [21] holds. The aim of this paper is to apply these SDP relaxation methods to solve SIPP problems.

An efficient method based on discretization scheme for solving SIP is the exchange method which approaches the optimum in an iterative manner. Generally speaking, given a finite subset $U_k \subseteq U$ in an iteration, we obtain at least one global minimizer x^k of $f(x)$ under the associated finitely many constraints and then compute the global minimum g^k and minimizers u_1, \dots, u_t of $g(x^k, u)$ over U . If $g^k \geq 0$, stop; otherwise, update $U_{k+1} = U_k \cup \{u_1, \dots, u_t\}$ and proceed to the next iteration. Therefore, to guarantee the success of the exchange method, the subproblems in each iteration need to be globally solved and at least one minimizer of each subproblem can be extracted. The compactness of the index set U is commonly assumed in many algorithms for SIP problems, which ensure the existence of global minimizers for constraint subproblem. However, when the constraint subproblem is nonconvex, globally solving it and extracting global minimizers are very challenging.

Specializing the exchange method in SIPP problem (P), the subproblems are polynomial optimization problems with finitely many constraints, which can be solved exactly by SDP relaxations. Assuming the index set U is compact, an exchange type method with SDP relaxations is given in this paper. Numerical experiments show that this algorithm is efficient in practice. We also apply this approach to optimization problems with polynomial matrix inequality and get good numerical performance. If U is noncompact, the exchange method might fail, see Example 4.1. Another novelty of this paper is that we extend the proposed algorithm to solve SIPP problems with noncompact U . By a technique of homogenization, we first reformulate the original SIPP problem as a new one with a compact index set, to which we then apply the proposed semidefinite relaxation algorithm. We prove that these two problems are equivalent under some generic conditions.

The paper is organized as follows. In Section 2, we introduce two SDP relaxation methods for solving polynomial optimization problems with finitely many constraints. In Section 3, we propose a semidefinite relaxation algorithm to solve SIPP problem (P) with compact index set U . In Section 4, we consider how to apply the proposed algorithm to solve SIPP problems with noncompact index set U by homogenization.

Notation. The symbol \mathbb{N} (resp., \mathbb{R} , \mathbb{C}) denotes the set of nonnegative integers (resp., real numbers, complex numbers). For any $t \in \mathbb{R}$, $\lceil t \rceil$ denotes the smallest integer that is not smaller than t . For integer $n > 0$, $[n]$ denotes the set $\{1, \dots, n\}$. For $x \in \mathbb{R}^n$, x_i denotes the i -th component of x . For $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{N}^n$, x^α denotes $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. For a finite set T , $|T|$ denotes its cardinality. $\mathbb{R}[x] =$

$\mathbb{R}[x_1, \dots, x_n]$ denotes the ring of polynomials in (x_1, \dots, x_n) with real coefficients. For a symmetric matrix W , $W \succeq 0$ (> 0) means that W is positive semidefinite (definite). For any vector $u \in \mathbb{R}^p$, $\|u\|$ denotes the standard Euclidean 2-norm.

2. SDP RELAXATIONS FOR POLYNOMIAL OPTIMIZATION

In this section, we study how to solve the following polynomial optimization problem with finitely many constraints:

$$(2.1) \quad \begin{cases} f_{\min} := \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } h_1(x) = \dots = h_{m_1}(x) = 0, \\ g_1(x) \geq 0, \dots, g_{m_2}(x) \geq 0, \end{cases}$$

where $f(x), h_i(x), g_j(x) \in \mathbb{R}[x]$. Based on the Positivstellensatz, considerable works have recently been done on solving (2.1) by means of SDP relaxation. Generally speaking, these methods relax (2.1) as a sequence of SDPs whose optima are lower bounds of f_{\min} and converge to f_{\min} under some assumptions. We first introduce the classic Lasserre's SDP relaxation [13] and then Nie's Jacobian SDP relaxation [20] with property of finite convergence.

2.1. Lasserre's SDP relaxation. Denote K as the feasible set of (2.1). Let $\mathcal{F} := \{h_1, \dots, h_{m_1}, g_0, g_1, \dots, g_{m_2}\}$ and $g_0 = 1$. We say a polynomial is SOS if it is a sum of squares of other polynomials. The k -th truncated *quadratic module* generated by \mathcal{F} is defined as

$$Q_k(\mathcal{F}) := \left\{ \sum_{j=1}^{m_1} \phi_j h_j + \sum_{i=0}^{m_2} \sigma_i g_i \mid \begin{array}{l} \sigma_i \text{ are SOS, } \phi_j \in \mathbb{R}[x], \forall i, j \\ \deg(\sigma_i g_i) \leq 2k, \deg(\phi_j h_j) \leq 2k \end{array} \right\}.$$

The k -th Lasserre's SDP relaxation [13] for solving (2.1) (k is also called the relaxation order) is

$$(2.2) \quad f_k := \max \gamma \quad \text{s.t. } f(x) - \gamma \in Q_k(\mathcal{F}).$$

The relaxation (2.2) is equivalent to a semidefinite program and could be solved efficiently by numerical methods like interior-point algorithms. Clearly, $f_k \leq f_{\min}$ for every k and the sequence $\{f_k\}$ is monotonically increasing. The quadratic module generated by \mathcal{F} is

$$Q(\mathcal{F}) := \bigcup_{k=1}^{\infty} Q_k(\mathcal{F}).$$

Definition 2.1. The set $Q(\mathcal{F})$ satisfies the *Archimedean Condition* if there exists $\psi \in Q(\mathcal{F})$ such that inequality $\psi(x) \geq 0$ defines a compact set in $x \in \mathbb{R}^n$.

Note that the Archimedean Condition implies the feasible set K is compact but the inverse is not necessarily true. However, for any compact K we can always "force" the associated quadratic module to satisfy the Archimedean Condition by adding a "redundant" constraint, e.g., $\rho - \|x\|^2 \geq 0$ for sufficiently large ρ .

The convergence for Lasserre's hierarchy (2.2), i.e., $\lim_{k \rightarrow \infty} f_k = f_{\min}$, is implied by Putinar's Positivstellensatz:

Theorem 2.2. ([22]) *If a polynomial p is positive on K and the Archimedean Condition holds, then $p \in Q(\mathcal{F})$.*

We next consider the dual optimization problem of (2.2). Let y be a *truncated moment sequence (tms)* of degree $2k$, i.e., $y = (y_\alpha)$ be a sequence of real numbers which are indexed by $\alpha := (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ with $|\alpha| := \alpha_1 + \dots + \alpha_n \leq 2k$. The associated k -th *moment matrix* is denoted as $M_k(y)$ which is indexed by \mathbb{N}_k^n , with (α, β) -th entry $y_{\alpha+\beta}$. Given polynomial $p(x) = \sum_{\alpha} p_{\alpha} x^{\alpha}$ where $x^{\alpha} := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, denote $d_p = \lceil \deg(p)/2 \rceil$. For $k \geq d_p$, the $(k - d_p)$ -th *localizing moment matrix* $L_p^{(k-d_p)}(y)$ is defined as the moment matrix of the *shifted vector* $((py)_{\alpha})_{\alpha \in \mathbb{N}_{2(k-d_p)}^n}$ with $(py)_{\alpha} = \sum_{\beta} p_{\beta} y_{\alpha+\beta}$. Denote by \mathcal{M}_{2k} the space of all tms whose degrees are $2k$. Let $\mathbb{R}[x]_{2k}$ be the space of real polynomials in x with degree at most $2k$. For any $y \in \mathcal{M}_{2k}$, a Riesz functional \mathcal{L}_y on $\mathbb{R}[x]_{2k}$ is defined as

$$\mathcal{L}_y \left(\sum_{\alpha} q_{\alpha} x_1^{\alpha_1} \cdots x_n^{\alpha_n} \right) = \sum_{\alpha} q_{\alpha} y_{\alpha}, \quad \forall q(x) \in \mathbb{R}[x]_{2k}.$$

For convenience, we hereafter still use q to denote the coefficient vector of $q(x)$ in the graded lexicographical ordering and denote $\langle q, y \rangle = \mathcal{L}_y(q)$. From the definition of the localizing moment matrix $L_p^{(k-d_p)}(y)$, it is easy to check that

$$q^T L_p^{(k-d_p)}(y) q = \mathcal{L}_y(p(x)q(x)^2), \quad \forall q(x) \in \mathbb{R}[x]_{k-d_p}.$$

The dual optimization problem of (2.2) is ([13, 14])

$$(2.3) \quad \begin{cases} f_k^* := \min_{y \in \mathcal{M}_{2k}} \langle f, y \rangle \\ \text{s.t. } L_{h_j}^{(k-d_{h_j})}(y) = 0, \quad j \in [m_1], \quad L_{g_i}^{(k-d_{g_i})}(y) \succeq 0, \quad i \in [m_2], \\ M_k(y) \succeq 0, \quad \langle 1, y \rangle = 1. \end{cases}$$

Let

$$d = \max\{1, d_{g_i}, d_{h_j} \mid i \in [m_1], j \in [m_2]\}.$$

Lasserre [13] shows that $f_k \leq f_k^* \leq f_{\min}$ for every $k \geq \max\{d_f, d\}$ and both $\{f_k\}$ and $\{f_k^*\}$ converge to f_{\min} if the Archimedean Condition holds.

We say Lasserre's hierarchy (2.2) and (2.3) has *finite convergence* if

$$(2.4) \quad f_{k_1} = f_{k_1}^* = f_{\min} \quad \text{for some order } k_1 < \infty.$$

Interestingly, Nie proved that under the Archimedean Condition, Lasserre's SDP relaxation has finite convergence *generically* (cf. [19, Theorem 1.1]). Since f_{\min} is usually unknown, a practical issue is how to certify the finite convergence if it happens. Moreover, if it is certified, how do we get minimizers?

Let y^* be an optimizer of (2.3). By [2, Theorem 1.1], $f_k^* = f_{\min}$ for some k if the *flat extension condition* (FEC) [2] holds, i.e.,

$$(2.5) \quad \text{rank } M_{k-d}(y^*) = \text{rank } M_k(y^*).$$

By solving some SVD and eigenvalue problems ([7]), we can get $r := \text{rank } M_k(y^*)$ global optimizers for (2.1). However, (2.5) is not a generally necessary condition for checking finite convergence of Lasserre's hierarchy (cf. [21, Example 1.1]). To certify the finite convergence of (2.2) and get minimizers of (2.1) from (2.3), a weaker condition was proposed in [21]. We say a minimizer y^* of (2.3) satisfies *flat truncation condition* (FTC) if there exists an integer $t \in [\max\{d_f, d\}, k]$ such that

$$(2.6) \quad \text{rank } M_{t-d}(y^*) = \text{rank } M_t(y^*).$$

If an optimizer of (2.3) has a flat truncation, by [2, Theorem 1.1] again, we still have $f_k^* = f_{\min}$.

Moreover, if there is no duality gap between (2.2) and (2.3), we obtain $f_k = f_{\min}$. More importantly, [21, Theorem 2.2] shows that the flat truncation is also necessary for Lasserre’s hierarchy (2.2) under some *generic* assumptions.

Algorithm 2.3. Lasserre’s SDP relaxation

Input: Objective function $f(x)$, constraint functions $h_i(x), g_j(x)$ and maximal relaxation order k_{\max} .

Output: Global minimum and minimizers of problem (2.1).

- I Set $d := \max\{1, d_f, d_{h_i}, d_{g_j}\}$ and initial relaxation order $k = d$.
- II Solve primal and dual SDP problems (2.2) and (2.3) by standard SDP solver (e.g., SeDuMi [25], SDPT3 [27], SDPNAL [28]).
- III For $t \in [d, k]$, check condition (2.6).
 - 1 If (2.6) holds for some t , get minimizers by Extraction Algorithm [7] and stop;
 - 2 Otherwise, go to Step IV.
- IV If $k > k_{\max}$, stop; otherwise, set $k = k + 1$ and go to Step II.

2.2. Jacobian SDP relaxation. The convergence of Lasserre’s SDP relaxations (2.2) and (2.3) might be *asymptotic* for some instances, i.e., only lower bounds are found for each order k . To overcome this hurdle, Nie [20] proposed a refined reformulation of (2.1) by some “Jacobian-type” technique whose SDP relaxation has finite convergence.

Roughly speaking, Jacobian SDP relaxation is to add auxiliary constraints to (2.1) which represent optimality conditions under the assumption that the optimum f_{\min} is achievable. The basic idea is that at each optimizer, the Jacobian matrix of the objective function, the equality constraints and the active inequality constraints must be singular, i.e., all its maximal minors vanish. For convenience, denote

$$h := (h_1, \dots, h_{m_1}) \quad \text{and} \quad g := (g_1, \dots, g_{m_2}).$$

For a subset $J = \{j_1, \dots, j_k\} \subseteq [m_2]$, denote $g_J := (g_{j_1}, \dots, g_{j_k})$. Symbols ∇h and ∇g_J represent the gradient vectors of the polynomials in h and g_J , respectively. Denote the determinantal variety of (f, h, g_J) ’s Jacobian being singular by

$$G_J = \{x \in \mathbb{C}^n \mid \text{rank } B^J(x) \leq m_1 + |J|\},$$

where

$$B^J(x) = [\nabla f(x) \quad \nabla h(x) \quad \nabla g_J(x)].$$

Instead of using all maximal minors to define G_J , [20, Section 2.1] discusses how to get the smallest number of defining equations. Let $\eta_1^J, \dots, \eta_{\text{len}(J)}^J$ be the set of defining polynomials for G_J where $\text{len}(J)$ is the number of these polynomials. For each $i = 1, \dots, \text{len}(J)$, define

$$(2.7) \quad \varphi_i^J(x) = \eta_i^J \cdot \prod_{j \in J^c} g_j(x), \quad \text{where } J^c = [m_2] \setminus J.$$

For simplicity, we list all possible φ_i^J in (2.7) sequentially as

$$\varphi_1, \varphi_2, \dots, \varphi_r, \quad \text{where } r = \sum_{J \subseteq [m_2], |J| \leq m - m_1} \text{len}(J).$$

Consider the following optimization by adding all φ_l 's to (2.1):

$$(2.8) \quad \begin{cases} s^* := \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } h_i(x) = 0, \quad i \in [m_1], \quad \varphi_l(x) = 0, \quad l \in [r], \\ g_j(x) \geq 0, \quad j \in [m_2]. \end{cases}$$

As shown in [20, Lemma 3.1] and [6, Lemma 3.5], by adding auxiliary constraints $\varphi_l(x) = 0$, the feasible set of (2.8) is restricted to the KKT points and singular points of the feasible set of (2.1). Therefore, (2.1) and (2.8) are equivalent if the minimum f_{\min} of (2.1) is achievable.

Lemma 2.4. ([6, Lemma 3.6]) *Assume $m_1 \leq n$ and at most $n - m_1$ of $g_1(x), \dots, g_{m_2}(x)$ vanish for any feasible point x . If the minimum f_{\min} of (2.1) is achievable, then $s^* = f_{\min}$.*

Remark 2.5. Since s^* is the minimal value of $f(x)$ achieved among the KKT points and singular points of the feasible set of (2.1), it is possible that $s^* > f_{\min}$ (cf. [20, Section 2.2]) if f_{\min} is not achievable.

If the Archimedean Condition holds for the feasible set K , then K is compact and f_{\min} is achievable. By Lemma 2.4, we always have $s^* = f_{\min}$. Applying Lasserre's SDP relaxations (2.2) and (2.3) to (2.8), the resulting SDP relaxations for (2.8) have finite convergence under some *generic* conditions (cf. [20, Theorem 4.2], [6, Theorem 3.9]).

Algorithm 2.6. Nie's Jacobian SDP relaxation

Input: Objective function $f(x)$, constraints functions $h_i(x), g_j(x)$, maximal relaxation order k_{\max} .

Output: Global minimum and minimizers of problem (2.1).

- I Construct the auxiliary polynomials $\varphi_l(x)$'s.
- II Set $d := \max\{1, d_f, d_{h_i}, d_{g_j}, d_{\varphi_l}\}$ and initial relaxation order $k = d$.
- III Solve (2.8) by Algorithm 2.3.
- IV For $t \in [d, k]$, check condition (2.6).
 - 1 If (2.6) holds for some t , get minimizers by Extraction Algorithm [7] and stop;
 - 2 Otherwise, go to Step V.
- V If $k > k_{\max}$, stop; otherwise, set $k = k + 1$ and go to Step III.

In contrast to Lasserre's SDP relaxation, Jacobian SDP relaxation is more complicated due to the auxiliary polynomials $\varphi_l(x)$'s. We refer to [20, Section 4] for some simplified versions of Jacobian SDP relaxation method.

3. SIPP WITH COMPACT SET U

The two SDP relaxation algorithms shown in Section 2 provide strong tools to globally solve polynomial optimization problems with finitely many constraints. In this section, we will discuss how to use them to solve SIPP problems globally.

3.1. A semidefinite relaxation algorithm. One main difficulty in solving a SIP problem is that there are infinite number of constraints. How to deal with the infinite index set U is the key difference among various SIP algorithms. Exchange method is commonly used in SIP computation, and is regarded as the most efficient method on solving SIP problems [10, 16]. The general steps of exchange method

are determined algorithmically as follows [10]. Given a subset $U_k \subseteq U$ in iteration k with $|U_k| < \infty$, compute at least one global solution x^k of

$$(3.1) \quad \min_{x \in X} f(x) \quad \text{s.t. } g(x, u) \geq 0, \quad \forall u \in U_k,$$

and solutions u_1, \dots, u_t of the subproblem

$$(3.2) \quad g^k := \min_{u \in U} g(x^k, u).$$

If $g^k \geq 0$, stop; otherwise, set $U_{k+1} = U_k \cup \{u_1, \dots, u_t\}$ and go to next iteration. Therefore, to successfully apply exchange method to solve SIPP problems, we need to globally solve subproblems (3.1)-(3.2) and extract global minimizers in each iteration. As we have discussed in Section 2, the SDP relaxation methods are proper means for this propose. The specific description of exchange method with SDP relaxations for SIPP problems is shown in the following.

Algorithm 3.1. Semidefinite relaxations for SIPP

Input: Objective function $f(x)$, constraint function $g(x, u)$, semi-algebraic sets X , U , tolerance ϵ and maximum iteration number k_{\max} .

Output: Global optimum f^* and set X^* of minimizers of problem (P) .

Step 1 Choose random $u_0 \in U$ and let $U_0 = \{u_0\}$. Set $X^* = \emptyset$ and $k = 0$.

Step 2 Use Algorithm 2.3 to solve

$$(3.3) \quad (P_k) : \begin{cases} f_k^{\min} := \min_{x \in X} f(x) \\ \text{s.t. } g(x, u) \geq 0, \quad \forall u \in U_k. \end{cases}$$

Let $S_k = \{x_1^k, \dots, x_{r_k}^k\}$ be the set of the global minimizers of problem (P_k) .

Step 3 Set $U_{k+1} = U_k$. For $i = 1, \dots, r_k$,

(a) Use Algorithm 2.6 to solve

$$(3.4) \quad (Q_i^k) : \quad g_i^k := \min_{u \in U} g(x_i^k, u).$$

Let $T_i^k = \{u_{i,j}^k, j = 1, \dots, t_i^k\}$ be the set of global minimizers of (Q_i^k) .

(b) Update $U_{k+1} = U_{k+1} \cup T_i^k$.

(c) If $g_i^k \geq -\epsilon$, then update $X^* = X^* \cup \{x_i^k\}$.

Step 4 If $X^* \neq \emptyset$ or $k > k_{\max}$, stop;

otherwise, set $k = k + 1$ and go back to Step 2.

Remark 3.2. Subproblems (P_k) and (Q_i^k) in Algorithm 3.1 can be solved by both Algorithm 2.3 and 2.6. Finite convergence can be guaranteed by Algorithm 2.6 which, however, produces SDPs of size exponentially depending on the number of the constraints. Since U_k enlarges as k increases, subproblem (P_k) consequently becomes hard to be solved by Algorithm 2.6. Therefore, we solve (P_k) by Algorithm 2.3 which is also proved to have finite convergence generically [19]. Because the index set U is fixed and compact, Algorithm 2.6 is a better choice for solving (Q_i^k) .

Proposition 3.3 (Monotonic Property). *For optimal values of (P_k) in (3.3), we have*

$$(3.5) \quad f_1^{\min} \leq \dots \leq f_k^{\min} \leq f_{k+1}^{\min} \leq \dots \leq f^*.$$

Proof. Because

$$U_1 \subseteq \dots \subseteq U_k \subseteq U_{k+1} \subseteq \dots \subseteq U.$$

So the feasible sets of (P_k) and (P) satisfy

$$K \subseteq \cdots \subseteq K_{k+1} \subseteq K_k \subseteq \cdots \subseteq K_1,$$

we obtain the conclusion. \square

We have the following convergence analysis of Algorithm 3.1:

Theorem 3.4. *Suppose that X is compact. If at each step k ,*

- (a) *subproblems (P_k) and each (Q_i^k) are globally solved,*
- (b) *intermediate results S_k and at least one T_i^k are nonempty,*

then either Algorithm 3.1 stops with solutions to (P) in a finite number of iterations or for any sequence $\{x^k\}$ with $x^k \in S_k$, there exists at least one limit point as k increases and each of them solves (P) .

Proof. At each step, if (a) holds, then global optima f_k^{\min} and g_i^k are obtained and monotonic property (3.5) is true. Additionally, if (b) is satisfied, then Algorithm 3.1 either stops in a finite number of iterations or proceeds without interrupt as k increases.

If Algorithm 3.1 stops at k -th iteration with $k < k_{\max}$, then $g_i^k \geq 0$ for some i , which implies that the associated x_i^k is feasible for (P) . Moreover, x_i^k is a global minimizer of (P) by (3.5). Now we assume $g_i^k < 0$ for each k and i which implies $T_i^k \not\subseteq U_k$ and $U_k \subset U_{k+1}$ for all k . The following argument is based on the proof of [10, Theorem 7.2]. For any $x \in X$, define

$$v(x) := \min\{g(x, u), u \in U\}.$$

Obviously, $v(x)$ is continuous. Fix a sequence $\{x^k\}$ with $x^k \in S_k$, then a limit point $\bar{x} \in X$ always exists since X is compact. Without loss of generality, assume $x^k \rightarrow \bar{x}$. By (3.5), it suffices to prove that \bar{x} is feasible for (P) . Let $v(x^k) = g(x^k, u^k)$ and X_k be the feasible set of (P_k) . Since $U_k \subset U_{k+1}$, we have $\bar{x} \in \bigcap_{k=1}^{\infty} X_k$ and therefore $g(\bar{x}, u^k) \geq 0$. Then

$$\begin{aligned} v(\bar{x}) &= v(x^k) + [v(\bar{x}) - v(x^k)] \\ &= g(x^k, u^k) + [v(\bar{x}) - v(x^k)] \\ &\geq [g(x^k, u^k) - g(\bar{x}, u^k)] + [v(\bar{x}) - v(x^k)]. \end{aligned}$$

By the continuity of v and g , we have $v(\bar{x}) \geq 0$, i.e., \bar{x} is feasible for (P) . \square

If X and U are compact, then the optima of (P_k) and (Q_i^k) are achievable. By applying SDP relaxations Algorithm 2.3 and Algorithm 2.6 to (P_k) and (Q_i^k) , as we have mentioned in Section 2, (a) and (b) are *generically* satisfied no matter what initial U_0 we choose. In section 4, we will consider the case when U is noncompact for which the convergence of Algorithm 3.1 might fail if we choose an arbitrary initial U_0 (Example 4.1). We will deal with this issue by the technique of homogenization.

3.2. Numerical experiments. This subsection presents some numerical examples to illustrate the efficiency of Algorithm 3.1. The computation is implemented with Matlab 7.12 on a Dell 64-bit Linux Desktop running CentOS (5.6) with 8GB memory and Intel(R) Core(TM) i7 CPU 860 2.8GHz. Algorithm 3.1 is implemented with software Gloptipoly [9]. SeDuMi [25] is used as a standard SDP solver. Throughout the computational experiments, we set parameters $k_{\max} = 15$, $\epsilon = 10^{-4}$ in Algorithm 3.1. After Algorithm 3.1 terminates, let X^* be the output set of global

minimizers of (P) , f^* be the value of the objective function f over X^* and Iter be the number of iterations Algorithm 3.1 has proceeded. Let

$$\text{Obj}_2 := \min_{x^* \in X^*} \min_{u \in U} g(x^*, u).$$

By the discussion in Subsection 3.1, the global minimizers in X^* can be certified by inequality $\text{Obj}_2 \geq -\epsilon$.

3.2.1. *Examples of small SIPP problems.* We test some small examples taken from [1, Appendix A]. For nonpolynomial functions, e.g., sine, cosine or exponential function, we use their Taylor polynomial approximations, see Appendix A. Let $X = [-100, 100]^n$. Test results are reported in Table 1. The Iter column in Table 1 indicates that Algorithm 3.1 takes a very few steps to find the global minimizer which are certified by the Obj_2 column.

TABLE 1. Computational results for small SIPP problems.

No.	x^*	Iter	f^*	Obj_2
Example A.1	(-0.0008, 0.4999)	2	-0.2504	6.4744e-7
Example A.2	(-0.7500, -0.6180)	3	0.1945	3.5305e-7
Example A.3	(-0.1514, -1.7484, 2.5725)	2	9.6973	7.8870e-5
Example A.4	(-1,0,0)	2	1	6.2320e-5
Example A.5	(0,0)	2	0	-1.1578e-12
Example A.6	(0,0,0)	2	4	-4.7070e-12
Example A.7	(0,0)	2	0	1.9285e-12

3.2.2. *Examples of random SIPP problems.* We test the performance of Algorithm 3.1 on some random SIPP problems which are generated as follows.

Let $x = (x_1, \dots, x_n)$ and $u = (u_1, \dots, u_p)$. Given $d \in \mathbb{N}$, let $[x]_d$ and $[u]_d$ be the vectors of monomials with degree up to d in $\mathbb{R}[x]$ and $\mathbb{R}[u]$, respectively. Denote $\langle [x]_d, [u]_d \rangle$ as the vector obtained by stacking $[x]_d$ and $[u]_d$. Let $f(x) = \eta^T [x]_{2d_1}$ be the objective function where η is a Gaussian random vector of matching dimension. Let $g(x, u) = \tau - \langle [x]_{d_2}, [u]_{d_2} \rangle^T M \langle [x]_{d_2}, [u]_{d_2} \rangle$, where τ is a random number in $[1, 10]$ and M is a random positive semidefinite matrix of matching dimension. Let $X = B_n(0, 1)$ be the unit ball in \mathbb{R}^n and U varies among $U_1 = B_p(0, 1)$, $U_2 = [-1, 1]^p$ and $U_3 = \Delta_p$ where Δ_p is the p dimensional simplex.

The results using Algorithm 3.1 are shown in Table 2 where the Inst column denotes the number of randomly generated instances, the consumed computer time is in the format hr:mn:sc with hr (resp. mn, sc) standing for the consumed hours (resp. minutes, seconds). The column Obj_2 shows that Algorithm 3.1 successfully solves all the random problems.

3.3. **Application to PMI problems.** In this subsection, we apply Algorithm 3.1 to the following optimization problem with *polynomial matrix inequality* (PMI):

$$(3.6) \quad f^{\min} := \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad G(x) \succeq 0,$$

where $f(x) \in \mathbb{R}[x]$ and $G(x)$ is an $m \times m$ symmetric matrix with entries $G_{ij}(x) \in \mathbb{R}[x]$. PMI is a special SIPP problem and has been widely arising in control system design, e.g., static output feedback design problems [8]. PMI is also interesting in optimization theory, e.g., SDP representation of a convex semialgebra set [17].

TABLE 2. Computational results for random SIPP problems

No.	n	p	d_1	d_2	Inst	U	time (min, max)		Obj ₂ (min, max)	
1	5	3	3	2	10	U_1	0:00:17	0:00:28	1.3479	2.0779
2	5	3	2	2	10	U_3	0:00:06	0:00:12	-9.5236e-9	0.6343
3	6	2	2	2	10	U_1	0:00:19	0:00:22	1.7144	2.1185
4	6	3	2	2	10	U_1	0:00:19	0:00:24	1.0450	1.7220
5	7	3	3	2	10	U_3	0:00:26	0:00:59	3.7797e-8	0.3198
6	8	3	2	2	10	U_1	0:04:52	0:05:18	1.3213	1.8438
7	9	2	2	2	5	U_1	0:45:26	0:49:28	1.5850	2.2807
8	9	2	2	2	5	U_3	0:44:40	0:52:49	1.7521e-8	2.9119e-7
9	5	2	2	2	5	U_2	0:57:17	1:04:02	1.3116e-6	1.6986e-5

Some traditional methods for globally solving (3.6) are based on branch-and-bound schemes and alike [5] which, as pointed in [8], are computationally expensive. Recently, some global methods based on SOS relaxations are proposed in [11, 12] as well as in [5] in a dual view.

Define

$$X := \{x \in \mathbb{R}^n \mid G(x) \succeq 0\} \quad \text{and} \quad U := \{u \in \mathbb{R}^m \mid \|u\|^2 = 1\}.$$

Then problem (3.6) is equivalent to the following SIPP problem

$$(3.7) \quad \begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } g(x, u) = u^T G(x) u \geq 0, \forall u \in U. \end{cases}$$

Assume the feasible set X is compact, then we can apply Algorithm 3.1 to solve SIPP problem (3.7). The following examples show that Algorithm 3.1 is efficient to solve PMI problems.

Example 3.5. Consider the following PMI problem:

$$(3.8) \quad \begin{cases} \min_{x \in \mathbb{R}^2} f(x) = x_1 + x_2 \\ \text{s.t. } G(x) = \begin{bmatrix} 4 - x_1^2 - x_2^2 & x_1 & x_2 \\ x_1 & x_2^2 - x_1 & x_1 x_2 \\ x_2 & x_1 x_2 & x_1^2 - x_2 \end{bmatrix} \succeq 0. \end{cases}$$

The characteristic polynomial of matrix $G(x)$ is:

$$p(t, x) = \det(tI_3 - G(x)) = t^3 - g_1(x)t^2 + g_2(x)t - g_3(x)$$

where

$$\begin{aligned} g_1(x) &= 4 - x_1 - x_2, \\ g_2(x) &= x_1^2 x_2 - 4x_2 - x_1^4 + x_1 x_2 - x_2^4 - 2x_1^2 x_2^2 + x_1 x_2^2 - 4x_1 + 3x_1^2 + 3x_2^2, \\ g_3(x) &= x_1^2 x_2 + 4x_1 x_2 + 2x_1^2 x_2^2 + x_1 x_2^2 - x_1^3 x_2 - 4x_1^3 + x_2^2 x_1^3 - x_2^3 x_1 - 4x_2^3 \\ &\quad - x_1^4 - x_2^4 + x_1^5 + x_2^5 + x_1^2 x_2^3. \end{aligned}$$

According to Descartes' rule of signs [8], the feasible set of (3.8) is

$$\{x \in \mathbb{R}^2 \mid g_1(x) \geq 0, g_2(x) \geq 0, g_3(x) \geq 0\}$$

which is shown shaded in Figure 1. We first reformulate (3.8) as a SIPP problem (3.7), then apply Algorithm 3.1 to it. After 5 iterations, we get a global minimizer $x^* \approx (-1.2853, -1.2763)$ which is certified by $\text{Obj}_2 = -1.4523 \times 10^{-4}$. The accuracy of this result can be seen from Figure 1. \square

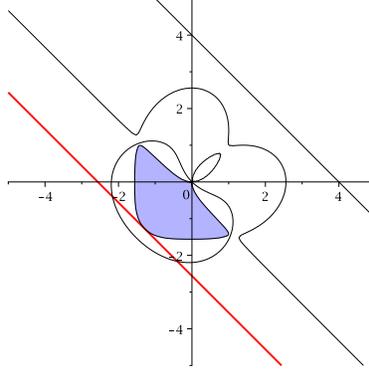


FIGURE 1. Feasible region of PMI problem (3.8) in Example 3.5.

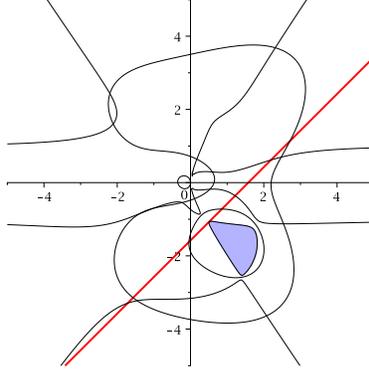


FIGURE 2. Feasible region of PMI problem (3.9) in Example 3.6.

Example 3.6. Consider the following PMI problem:

$$(3.9) \quad \begin{cases} \min_{x \in \mathbb{R}^2} f(x) = x_1 - x_2 \\ \text{s.t. } G(x) = \begin{bmatrix} 10 - x_1^2 - x_2^2 & x_1 & -x_1^2 + x_2 & x_2 + 3 \\ x_1 & x_2^2 & x_1 - x_2^2 & x_1 \\ -x_1^2 + x_2 & x_1 - x_2^2 & x_1 + 2x_2^2 & x_2 \\ x_2 + 3 & x_1 & x_2 & x_2^2 \end{bmatrix} \succeq 0. \end{cases}$$

Similar to Example 3.5, we obtain the feasible set of (3.9) by Descartes' rule of signs [8] and show it shaded in Figure 2. Applying Algorithm 3.1 to the reformulation (3.7) of problem (3.9), we get global minimizer $x^* \approx (0.5093, -1.0678)$ and minimum $f(x^*) \approx 1.5771$ which are certified by $\text{Obj}_2 = -9.4692 \times 10^{-5}$. From Figure 2, we can see this result is accurate. \square

We end this subsection by pointing out a trick hidden in the reformulation (3.7) of (3.6). PMI optimization problem (3.6) can be regarded as a SIPP problem with noncompact index set $\tilde{U} = \mathbb{R}^m$. Since the constraint function $g(x, u)$ is homogenous in u , we can restrict \tilde{U} to the unit sphere U . By Theorem 3.4, to guarantee the convergence of Algorithm 3.1, the optimum of (Q_i^k) needs to be achievable for each

k which might fail if U is noncompact. The reformulation (3.7) of (3.6) gives us a clue for dealing with SIPP with noncompact U by the technique of homogenization. We will go into detail about this technique in next section.

4. SIPP WITH NONCOMPACT SET U

At some k -th iteration of Algorithm 3.1, if the global minima g_i^k of (Q_i^k) are not achievable for all $x_i^k \in S_k$, then by Remark 2.5, either

- case 1. $T_i^k = \emptyset$, then U_{k+1} can not be updated and consequently S_{k+1} remains the same as S_k , or
case 2. U_{k+1} is updated by T_i^k which consists of KKT points or singular points of the feasible set of (Q_i^k) rather than global minimizers.

As we have discussed in Subsection 3.1, the convergence property of Algorithm 3.1 might fail or wrong global minimizers might be outputted if the above cases happen. For example,

Example 4.1. Consider the following problem:

$$(4.1) \quad \begin{cases} f^* := \min_{x_1, x_2 \in \mathbb{R}} -x_1 - x_2 \\ \text{s.t. } x_1(u_1^2 - 1) + (x_2 - u_1 u_2)^2 \geq 0, \forall u_1, u_2 \in \mathbb{R}, \\ x_1^2 + x_2^2 = 2. \end{cases}$$

We choose u_1, u_2 such that $x_2 - u_1 u_2 = 0$. By letting u_1 tend to infinity and 0 respectively, we obtain that $x_1 = 0$ for any feasible point x . Therefore, there are only two feasible points $(0, \pm\sqrt{2})$ and the global minimum is $-\sqrt{2}$ with minimizer $(0, \sqrt{2})$.

We claim that Algorithm 3.1 *fails* to solve (4.1) if we set initial $U_0 = \{(u_1^0, u_2^0)\}$ such that

$$(u_1^0, u_2^0) \notin \mathcal{U} := \{u \in \mathbb{R}^2 \mid u_1 u_2 = \sqrt{2}, u_1^2 < 2\sqrt{2} - 2\}.$$

We prove it in the following. First, we show that for any $(u_1, u_2) \in \mathbb{R}^2$ there always exists (\bar{x}_1, \bar{x}_2) with $\bar{x}_1 > 0, \bar{x}_2 > 0$ such that

$$g(\bar{x}, u) := \bar{x}_1(u_1^2 - 1) + (\bar{x}_2 - u_1 u_2)^2 \geq 0, \quad \bar{x}_1^2 + \bar{x}_2^2 = 2.$$

This is true if $g((0, \sqrt{2}), u) > 0$ or $g((\sqrt{2}, 0), u) > 0$ by the continuity of $g(x, u)$. Now we assume

$$g((0, \sqrt{2}), u) \leq 0 \quad \text{and} \quad g((\sqrt{2}, 0), u) \leq 0.$$

From the first inequality, we get $u_1 u_2 = \sqrt{2}$. Then by the second inequality, we have $u_1^2 \leq 1 - \sqrt{2}$ which is a contradiction. Therefore, the following subproblem

$$(P_0) : \begin{cases} f_0^{\min} := \min_{x \in \mathbb{R}^2} -x_1 - x_2 \\ \text{s.t. } x_1^2 + x_2^2 = 2, g(x, u) \geq 0, \forall u_1, u_2 \in \mathbb{R}, \end{cases}$$

has global minimizer $S_0 = \{(\tilde{x}_1, \tilde{x}_2)\}$ with $\tilde{x}_1 > 0, \tilde{x}_2 > 0$. Then we solve subproblem

$$(4.2) \quad (Q^0) : \quad g^0 := \min_{u \in \mathbb{R}^2} g(\tilde{x}, u) = \tilde{x}_1(u_1^2 - 1) + (\tilde{x}_2 - u_1 u_2)^2.$$

Obviously, $g^0 = -\tilde{x}_1$ is not achievable. Applying Jacobian SDP relaxation Algorithm 2.6, we obtain $T^0 = \{(0, 0)\}$ which consists of the only critical point $(0, 0)$

of map $g(x^0, u)$ with critical value $\tilde{x}_2^2 - \tilde{x}_1$. If $\tilde{x}_2^2 - \tilde{x}_1 \geq 0$, then Algorithm 3.1 terminates and outputs $X^* = \{(\tilde{x}_1, \tilde{x}_2)\}$ which is a wrong solution. Now we assume $\tilde{x}_2^2 - \tilde{x}_1 < 0$ and continue. By Algorithm 3.1, $U_1 = \{(\bar{u}_1, \bar{u}_2), (0, 0)\}$. Then we go to the next iteration and solve

$$(P_1) : \begin{cases} f_1^{\min} := \min_{x \in \mathbb{R}^2} -x_1 - x_2 \\ \text{s.t. } x_2^2 - x_1 \geq 0, g(x, \bar{u}) \geq 0, \\ x_1^2 + x_2^2 = 2. \end{cases}$$

Let K_1 be the feasible set of (P_1) , then

- case 1. There exists no $(\bar{x}_1, \bar{x}_2) \in K_1$ with $\bar{x}_1 > 0, \bar{x}_2 > 0$. The global minimizer of (P_1) is $S_1 = \{(0, \sqrt{2})\}$ and $g^1 := \min_{u \in \mathbb{R}^2} g((0, \sqrt{2}), u) \geq 0$. Therefore, the correct global solution of (4.1) is outputted. In this case, by the continuity of $g(x, u)$, we have $g((0, \sqrt{2}), \bar{u}) \leq 0$ and $g((1, 1), \bar{u}) < 0$. From these two inequalities, we get $(\bar{u}_1, \bar{u}_2) \in \mathcal{U}$.
- case 2. There exists $(\hat{x}_1, \hat{x}_2) \in K_1$ with $\hat{x}_1 > 0, \hat{x}_2 > 0$. Then the global minimizer of (P_1) is $S_1 = \{(\hat{x}_1, \hat{x}_2)\}$ with $\hat{x}_1 > 0, \hat{x}_2 > 0$. Similar to g^0 , g^1 is not achievable and $U_1 = \{(\bar{u}_1, \bar{u}_2), (0, 0)\}$ can not be updated. Consequently, the same process will be repeated in the following iterations.

Now we have proved the claim. Since the set \mathcal{U} is a subset of a Zariski closed set of \mathbb{R}^2 , Algorithm 3.1 fails if we choose a generic initial $U_0 = \{(u_1, u_2)\}$. \square

Hence, Algorithm 3.1 might fail to solve SIPP problem (P) if the optima of subproblems (Q_i^k) can not be reached for all $x_i^k \in S_k$ which might happen when U is noncompact. As we have mentioned at the end of Section 3, the reformulation (3.7) of (3.6) sheds light on this issue by the technique of homogenization. In the following, we apply this technique to general SIPP problem (P) with noncompact index set U .

For given polynomial $q(u) \in \mathbb{R}[u] := \mathbb{R}[u_1, \dots, u_p]$ with degree $d = \deg(q)$, let $\tilde{q}(\tilde{u}) = u_0^d q(u/u_0)$ be the homogenization of $q(u)$ where $\tilde{u} = (u_0, u) \in \mathbb{R}^{p+1}$. Define

$$\tilde{g}(x, \tilde{u}) = u_0^{d_g} g(x, u/u_0) \quad \text{where } d_g = \deg_u g(x, u)$$

and

$$U = \{u \in \mathbb{R}^p \mid h_1(u) \geq 0, \dots, h_{m_1}(u) \geq 0\},$$

$$U_0 = \{\tilde{u} \in \mathbb{R}^{p+1} \mid \tilde{h}_1(\tilde{u}) \geq 0, \dots, \tilde{h}_{m_1}(\tilde{u}) \geq 0, u_0 > 0, \|\tilde{u}\|^2 = 1\},$$

$$\tilde{U} = \{\tilde{u} \in \mathbb{R}^{p+1} \mid \tilde{h}_1(\tilde{u}) \geq 0, \dots, \tilde{h}_{m_1}(\tilde{u}) \geq 0, u_0 \geq 0, \|\tilde{u}\|^2 = 1\}.$$

Proposition 4.2. $q(u) \geq 0$ on U if and only if $\tilde{q}(\tilde{u}) \geq 0$ on $\text{closure}(U_0)$.

Proof. “If” direction. Suppose there exists $v \in U$ such that $q(v) < 0$. For $i \in [m_1]$, we have $h_i(v) \geq 0$. Let $\tilde{v} = (\frac{1}{\sqrt{1+\|v\|^2}}, \frac{v}{\sqrt{1+\|v\|^2}})$, then

$$\tilde{h}_i(\tilde{v}) = (1 + \|v\|^2)^{-\frac{\deg(h_i)}{2}} h_i(v) \geq 0, \quad i \in [m_1],$$

which implies $\tilde{v} \in U_0$ and

$$\tilde{q}(\tilde{v}) = (1 + \|v\|^2)^{-\frac{d}{2}} q(v) < 0.$$

It contradicts the assumption that $\tilde{q}(\tilde{v}) \geq 0$ on $\text{closure}(U_0)$.

“Only if” direction. Let $\tilde{v} = (v_0, v) \in \text{closure}(U_0)$, then there exists a sequence $\tilde{v}^k = (v_0^k, v^k) \in U_0$ such that $\lim_{k \rightarrow \infty} (v_0^k, v^k) = (v_0, v)$ with $v_0^k > 0$ for all k . We have

$$h_i(v^k/v_0^k) = (v_0^k)^{-\deg(h_i)} \tilde{h}_i(\tilde{v}^k) \geq 0, \quad i \in [m_1], \quad \text{for all } k.$$

Therefore, the sequence $\{v^k/v_0^k\} \in U$ and $q(v^k/v_0^k) \geq 0$. Since q is continuous,

$$\tilde{q}(\tilde{v}) = \lim_{k \rightarrow \infty} \tilde{q}(\tilde{v}^k) = \lim_{k \rightarrow \infty} (v_0^k)^d q(v^k/v_0^k) \geq 0,$$

which shows $\tilde{q}(\tilde{v}) \geq 0$ on $\text{closure}(U_0)$. The proof is completed. \square

Corollary 4.3. *A polynomial $q(u) \geq 0$ on \mathbb{R}^p if and only if $\tilde{q}(\tilde{u}) \geq 0$ on $\{\tilde{u} \in \mathbb{R}^{p+1} \mid \|\tilde{u}\|^2 = 1\}$.*

Proof. From the proof of Proposition 4.2, we can see the inequality $u_0 > 0$ can be removed from U_0 such that $q(u) \geq 0$ on \mathbb{R}^p if and only if $\tilde{q}(\tilde{u}) \geq 0$ on

$$\text{closure}(\{\tilde{u} \in \mathbb{R}^{p+1} \mid \|\tilde{u}\|^2 = 1\}) = \{\tilde{u} \in \mathbb{R}^{p+1} \mid \|\tilde{u}\|^2 = 1\}.$$

\square

By Proposition 4.2, we have the following equivalent reformulation of problem (P):

$$(P_0) : \begin{cases} f^* := \min_{x \in X} f(x) \\ \text{s.t. } \tilde{g}(x, \tilde{u}) \geq 0, \quad \forall \tilde{u} \in \text{closure}(U_0). \end{cases}$$

Some natural questions arise: how to get the explicit expression of semi-algebraic set $\text{closure}(U_0)$? Is it true that $\text{closure}(U_0) = \tilde{U}$? Clearly, we have

$$(4.3) \quad \text{closure}(U_0) \subseteq \tilde{U}.$$

Unfortunately, the equality does not always hold even if set U is compact (cf. [18, Example 5.2]).

Definition 4.4. ([18]) *U is closed at ∞ if $\text{closure}(U_0) = \tilde{U}$.*

Since it might be hard to express $\text{closure}(U_0)$ for a given particular SIPP problem, we consider to solve the following problem in general:

$$(\tilde{P}) : \begin{cases} \tilde{f}^* := \min_{x \in X} f(x) \\ \text{s.t. } \tilde{g}(x, \tilde{u}) \geq 0, \quad \forall \tilde{u} \in \tilde{U}. \end{cases}$$

As set \tilde{U} is compact, the semidefinite relaxation Algorithm 3.1 in Section 3 can successfully solve this problem with any arbitrary initial U_0 . Next we investigate the relation between problem (P) and problem (\tilde{P}) .

We define

$$\begin{aligned} M &= \{x \in \mathbb{R}^n \mid g(x, u) \geq 0, \quad \forall u \in U\}. \\ \tilde{M} &= \{x \in \mathbb{R}^n \mid \tilde{g}(x, \tilde{u}) \geq 0, \quad \forall \tilde{u} \in \tilde{U}\}. \end{aligned}$$

Proposition 4.5. *We have $\tilde{M} \subseteq M$ and the equality holds if U is closed at ∞ .*

Proof. By Proposition 4.2, we have

$$M = \{x \in \mathbb{R}^n \mid \tilde{g}(x, \tilde{u}) \geq 0, \quad \forall \tilde{u} \in \text{closure}(U_0)\}.$$

Then the conclusion follows due to the relationship (4.3). \square

Consequently, we have

Theorem 4.6. $\tilde{f}^* \geq f^*$ and the equality holds if U is closed at ∞ .

Corollary 4.3 shows that $U = \mathbb{R}^p$ is closed at ∞ and therefore,

Corollary 4.7. *The following two problems are equivalent:*

$$\begin{cases} \min_{x \in X} f(x) \\ \text{s.t. } g(x, u) \geq 0, \forall u \in \mathbb{R}^p, \end{cases} \quad \begin{cases} \min_{x \in X} f(x) \\ \text{s.t. } \tilde{g}(x, \tilde{u}) \geq 0, \forall \tilde{u} \in \tilde{U}, \end{cases}$$

where $\tilde{U} = \{\tilde{u} \in \mathbb{R}^{p+1} \mid \|\tilde{u}\|^2 = 1\}$.

Example 4.1 (Continued). We reformulate the problem (4.1) as

$$(4.4) \quad \begin{cases} \tilde{f}^* := \min_{x_1, x_2 \in \mathbb{R}} -x_1 - x_2 \\ \text{s.t. } x_1(u_1^2 - u_0^2) + (x_2 u_0^2 - u_1 u_2)^2 \geq 0, \forall \tilde{u} \in \tilde{U}, \\ x_1^2 + x_2^2 = 2, \end{cases}$$

where $\tilde{U} = \{(u_0, u_1, u_2) \in \mathbb{R}^3 \mid u_0^2 + u_1^2 + u_2^2 = 1\}$. By choosing $u_0 = 1$, we know $\tilde{M} \supseteq \{(0, \pm\sqrt{2})\}$ which, obviously, are feasible to (4.4). Therefore, $\tilde{f}^* = f^* = -\sqrt{2}$ with minimizer $(0, \sqrt{2})$. Choosing $U_0 = \{(1, 0, 0)\}$ in Algorithm 3.1, Figure 3 shows the feasible regions of subproblems (P_k) for iterations $k = 0, 1, \dots, 5$. Let $h(x) = x_1^2 + x_2^2 - 2$. At i -th iteration, the feasible region is defined by

$$K_i := \{x \in \mathbb{R}^2 \mid h(x) = 0, g_0(x) \geq 0, \dots, g_i(x) \geq 0\}$$

where

$$\begin{aligned} g_0 &= -x_1 + x_2^2, \\ g_1 &\approx 0.026046 - 0.31963x_1 - 0.19679x_2 + 0.37171x_2^2, \\ g_2 &\approx 0.054893 - 0.11577x_1 - 0.18811x_2 + 0.16116x_2^2, \\ g_3 &\approx 0.06865 - 0.049084x_1 - 0.14992x_2 + 0.081854x_2^2, \\ g_4 &\approx 0.072498 - 0.025711x_1 - 0.12039x_2 + 0.049977x_2^2, \\ g_5 &\approx 0.073368 - 0.018151x_1 - 0.10683x_2 + 0.038891x_2^2. \end{aligned}$$

For each i , the feasible region K_i is the intersection of the left parts of the circle $x_1^2 + x_2^2 = 2$ divided by hyperbolas $g_i(x) = 0, i = 0, \dots, 5$. From Figure 3, we can see the minimizers of subproblems (P_k) converge to $(0, \sqrt{2})$ which is the minimizer of problem (4.1). \square

We would like to point out that if U is not closed at ∞ , we might have $\tilde{f}^* > f^*$. For example,

Example 4.8. Consider the following SIPP problem:

$$(4.5) \quad \begin{cases} f^* := \min_{x \in \mathbb{R}} x^2 \\ \text{s.t. } x(u_1 - u_2 + 1) \geq 0, \forall u \in U, \\ x \in [1, 2], \end{cases}$$

where

$$U = \{u \in \mathbb{R}^2 : u_1^2(u_1 - u_2) - 1 = 0\}.$$

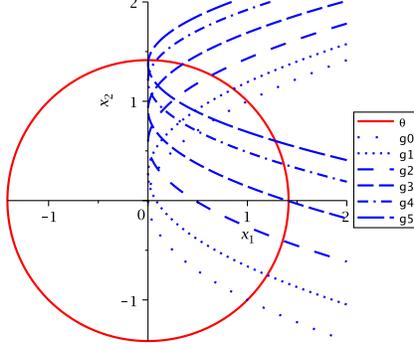


FIGURE 3. Feasible region of Example 4.1 at each iteration.

Since for all $u \in U$,

$$g(1, u) = u_1 - u_2 + 1 = \frac{1}{u_1^2} + 1 > 0,$$

$x^* = 1$ is feasible and furthermore the minimizer of problem (4.5). Hence, $f^* = 1$. By definition,

$$\tilde{U} = \{\tilde{u} \in \mathbb{R}^3 : u_1^2(u_1 - u_2) - u_0^3 = 0, u_0 \geq 0, u_0^2 + u_1^2 + u_2^2 = 1\}.$$

As is shown in [6, 18], U is not closed at ∞ because there exists a point $(0, 0, 1) \in \tilde{U}$ but $(0, 0, 1) \notin \text{closure}(U_0)$. Since for any $x \in [1, 2]$,

$$\tilde{g}(x, (0, 0, 1)) = -x < 0,$$

we have $\tilde{M} = \emptyset$. Therefore, $\tilde{f}^* = \infty > f^*$. \square

Example 4.8 shows that the problem (\tilde{P}) might not be equivalent to (P) when set U is not closed at ∞ . In the following, however, we show that U is closed at ∞ in general. In other words, U is closed at ∞ if it is defined by generic polynomials.

Suppose that U is not closed at ∞ , then by definition there exists $(0, \bar{u}) \in \tilde{U} \setminus \text{closure}(U_0)$ with $0 \neq \bar{u} \in \mathbb{R}^p$. Let \hat{h}_i denote the homogeneous part of highest degree of h_i for $i \in [m_1]$ and

$$\{j_1, \dots, j_\ell\} := \{j \in [m_1] \mid \tilde{h}_j(0, \bar{u}) = \hat{h}_j(\bar{u}) = 0\}.$$

Then \bar{u} is a solution to the polynomial system

$$(4.6) \quad \hat{h}_{j_1}(\bar{u}) = \dots = \hat{h}_{j_\ell}(\bar{u}) = \|\bar{u}\|^2 - 1 = 0.$$

The Jacobian matrix of the system (4.6) at \bar{u} is

$$A(u) := \begin{bmatrix} \frac{\partial \hat{h}_{j_1}}{\partial u_1}(\bar{u}) & \dots & \frac{\partial \hat{h}_{j_1}}{\partial u_p}(\bar{u}) \\ \vdots & \vdots & \vdots \\ \frac{\partial \hat{h}_{j_\ell}}{\partial u_1}(\bar{u}) & \dots & \frac{\partial \hat{h}_{j_\ell}}{\partial u_p}(\bar{u}) \\ 2\bar{u}_1 & \dots & 2\bar{u}_p \end{bmatrix}$$

Lemma 4.9. ([6, Lemma 2.10]) *Suppose U is not closed at ∞ and $\ell < p$, then $\text{rank } A(u) < \ell + 1$.*

Let $\hat{h}_{m_1+1} := \|\tilde{u}\|^2 - 1$ and $J(\bar{u}) = \{j_1, \dots, j_\ell, m_1 + 1\}$. We review some background about *resultants* and *discriminants* in Appendix B. By Proposition B.1 and Proposition B.3, we have

Theorem 4.10. *If U is not closed at ∞ , then*

(a) *If $|J(\bar{u})| > p$, then for every subset $\{j_1, \dots, j_{p+1}\} \subseteq J(\bar{u})$,*

$$\text{Res}(\hat{h}_{j_1}, \dots, \hat{h}_{j_{p+1}}) = 0.$$

(b) *If $|J(\bar{u})| \leq p$, then $\Delta(\hat{h}_{j_1}, \dots, \hat{h}_{j_\ell}, \hat{h}_{m_1+1}) = 0$.*

The above theorem shows that if U is defined by some generic polynomials, then it is closed at ∞ . Hence, the assumption that U is closed at ∞ is a generic condition. Therefore, SIPP problems (P) and (\tilde{P}) are *equivalent* in general.

Example 4.11. Consider the following problem

$$(4.7) \quad \begin{cases} \min_{x \in X} f(x) = x_1^2 + x_2^2 \\ \text{s.t. } g(x, u) = x_1 u_1 + u_2 + x_2 \geq 0, \quad \forall u \in U, \end{cases}$$

where

$$X := \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 4\} \quad \text{and} \quad U := \{(u_1, u_2) \in \mathbb{R}^2 \mid u_1^3 + u_2^3 - 3u_1 u_2 \geq 0\}.$$

The set U is shown shaded in Figure 4. Since $u_1 + u_2 + 1 = 0$ is the asymptote of the curve $u_1^3 + u_2^3 - 3u_1 u_2 = 0$, the inequality $g(x, u) \geq 0$ for all $u \in U$ requires $x_1 = 1$ and $x_2 \geq 1$. Therefore, the feasible set of (4.7) is $\{x \in \mathbb{R}^2 \mid x_1 = 1, 1 \leq x_2 \leq \sqrt{3}\}$ and the global minimizer is $x^* = (1, 1)$. It is easy to see that for a given $(\bar{x}_1, \bar{x}_2) \in X$, the global minimum of $g(\bar{x}, u)$ over U is either $-\infty$ or finite but not achievable. Therefore, by the discussion at the beginning of this section, Algorithm 3.1 might fail to solve (4.7). For example, if we set $U_0 = \{(1, -1)\}$, then we get minimizer $X^* = \{(0.5000, 0.4999)\}$; if $U_0 = \{(1, 0)\}$, then $X^* = \{(0.0262, 0.3086) \times 10^{-5}\}$.

Now we use the homogenization technique to reformulate (4.7). First, we show that U is closed at ∞ . Let

$$U_0 = \{(u_0, u_1, u_2) \in \mathbb{R}^3 \mid u_1^3 + u_2^3 - 3u_1 u_2 u_0 \geq 0, \quad u_0^2 + u_1^2 + u_2^2 = 1, \quad u_0 > 0\},$$

$$\tilde{U} = \{(u_0, u_1, u_2) \in \mathbb{R}^3 \mid u_1^3 + u_2^3 - 3u_1 u_2 u_0 \geq 0, \quad u_0^2 + u_1^2 + u_2^2 = 1, \quad u_0 \geq 0\}.$$

By definition, if U is not closed at ∞ , then there exists $(0, \bar{u}_1, \bar{u}_2) \in \tilde{U} \setminus \text{closure}(U_0)$ which implies

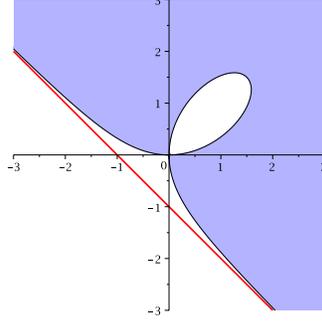
$$\bar{u}_1^3 + \bar{u}_2^3 = 0, \quad \bar{u}_1^2 + \bar{u}_2^2 = 1.$$

Therefore

$$(\bar{u}_1, \bar{u}_2) \in \left\{ \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right), \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right) \right\}.$$

Let

$$\tilde{u}_k := \left(\sqrt{2\varepsilon_k}, -\sqrt{\frac{1}{2} - \varepsilon_k}, \sqrt{\frac{1}{2} - \varepsilon_k} \right), \quad \hat{u}_k := \left(\sqrt{2\varepsilon_k}, \sqrt{\frac{1}{2} - \varepsilon_k}, -\sqrt{\frac{1}{2} - \varepsilon_k} \right).$$

FIGURE 4. The feasible region U in Example 4.11.

Let $\varepsilon_k \rightarrow 0$, then $\tilde{u}_k, \hat{u}_k \in U_0$ for all k large enough and

$$\lim_{k \rightarrow \infty} \tilde{u}_k = \left(0, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right), \quad \lim_{k \rightarrow \infty} \hat{u}_k = \left(0, \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right).$$

This shows U is closed at ∞ . Therefore, by homogenization, we reformulate (4.7) as the following equivalent problem

$$\begin{cases} \min_{x \in X} x_1^2 + x_2^2 \\ \text{s.t. } \tilde{g}(x, \tilde{u}) = x_1 u_1 + u_2 + x_2 u_0 \geq 0, \tilde{u} \in \tilde{U}. \end{cases}$$

By Algorithm 3.1, we find a global minimizer

$$x^* \approx (0.9999, 0.9998) \quad \text{with} \quad \text{Obj}_2 = -9.8148 \times 10^{-7},$$

after several iterations. \square

In this section, by homogenization technique, we reformulate the SIPP problem (P) with noncompact index set U as the problem (\tilde{P}) with compact index set \tilde{U} which can be globally solved by Algorithm 3.1. Under the assumption that set U is closed at ∞ which is a generic condition, we show the two problems are equivalent.

APPENDIX A. SMALL SIPP EXAMPLES

Example A.1. Let $U = [0, 2]$ and

$$f(x) = \frac{1}{3}x_1^2 + \frac{1}{2}x_1 + x_2^2 - x_2, \quad g(x, u) = -x_1^2 - 2x_1x_2u^2 + \sin(u).$$

Replace the function $\sin(u)$ by $u - \frac{u^3}{6}$.

Example A.2. Let $U = [0, 1]$ and

$$f(x) = \frac{1}{3}x_1^2 + x_2^2 + \frac{1}{2}x_1, \quad g(x, u) = -(1 - x_1^2u^2)^2 + x_1u^2 + x_2^2 - x_2.$$

Example A.3. Let $U = [0, 1]$ and

$$f(x) = x_1^2 + x_2^2 + x_3^2, \quad g(x, u) = -x_1 - x_2e^{x_3u} - e^{2u} + 2\sin(4u).$$

Replace function e^{x_3u} by $1 + x_3u + \frac{1}{2}x_3^2u^2 + \frac{1}{6}x_3^3u^3 + \frac{1}{24}x_3^4u^4$, function e^{2u} by $1 + 2u + 2u^2 + \frac{4}{3}u^3 + \frac{2}{3}u^4$, and function $\sin(4u)$ by $4u - \frac{32}{3}u^3$.

Example A.4. Let $U = [0, 1]^2$ and

$$f(x) = x_1^2 + x_2^2 + x_3^2, \quad g(x, u) = -x_1(u_1 + u_2^2 + 1) - x_2(u_1 u_2 - u_2^2) - x_3(u_1 u_2 + u_2^2 + u_2) - 1.$$

Example A.5. Let $U = [0, \pi]$ and

$$f(x) = x_2^2 - 4x_2, \quad g(x, u) = -x_1 \cos(u) - x_2 \sin(u) + 1.$$

Replace function $\sin(u)$ by $u - \frac{1}{6}u^3$ and $\cos(u)$ by $1 - \frac{1}{2}u^2 + \frac{1}{24}u^4$.

Example A.6. Let $U = [0, \pi]$ and

$$\begin{aligned} f(x) &= (x_1 + x_2 - 2)^2 + (x_1 - x_2)^2 + 30 \min(0, (x_1 - x_2))^2, \\ g(x, u) &= -x_1 \cos(u) - x_2 \sin(u) + 1. \end{aligned}$$

Like in [15], let $x_3 = \min(0, (x_1 - x_2))$, then $f(x) = (x_1 + x_2 - 2)^2 + (x_1 - x_2)^2 + 30x_3^2$. We add new constraints $x_3^2 = (x_1 - x_2)^2$ and $x_3 \geq 0$ in X . Replace function $\sin(u)$ by $u - \frac{u^3}{6} + \frac{u^5}{5!}$.

Example A.7. Let $U = [-1, 1]$ and

$$f(x) = x_2, \quad g(x, u) = -2x_1^2 u^2 + u^4 - x_1^2 + x_2.$$

APPENDIX B. RESULTANTS AND DISCRIMINANTS

We review some background about *resultants* and *discriminants*. More details can be found in [4, 18, 20].

Let f_1, \dots, f_n be homogeneous polynomials in $x = (x_1, \dots, x_n)$. The resultant $\text{Res}(f_1, \dots, f_n)$ is a polynomial in the coefficients of f_1, \dots, f_n satisfying

$$\text{Res}(f_1, \dots, f_n) = 0 \Leftrightarrow \exists 0 \neq u \in \mathbb{C}^n, f_1(u) = \dots = f_n(u) = 0.$$

Let f_1, \dots, f_m be homogenous polynomials with $m < n$. The discriminant for f_1, \dots, f_m is denoted by $\Delta(f_1, \dots, f_m)$, which is a polynomial in the coefficients of f_1, \dots, f_m such that

$$\Delta(f_1, \dots, f_m) = 0$$

if and only if the polynomial system

$$f_1(x) = \dots = f_m(x) = 0$$

has a solution $0 \neq u \in \mathbb{C}^n$ such that the Jacobian matrix of f_1, \dots, f_m does not have full rank.

Given inhomogeneous polynomial $h(x) \in \mathbb{R}[x]$, let \tilde{h} denote the homogenization of h , i.e., $\tilde{h} = \tilde{h}(\tilde{x}) = x_0^{\deg(h)} h(x/x_0)$. For inhomogeneous polynomials $f_0, f_1, \dots, f_n \in \mathbb{R}[x]$, the resultant $\text{Res}(f_0, f_1, \dots, f_n)$ is defined to be

$$\text{Res}(\tilde{f}_0, \tilde{f}_1, \dots, \tilde{f}_n).$$

For inhomogeneous polynomials $f_1, \dots, f_m \in \mathbb{R}[x]$ with $m \leq n$, the discriminant $\Delta(f_1, \dots, f_m)$ is defined as

$$\Delta(\tilde{f}_1, \dots, \tilde{f}_m).$$

We have

Proposition B.1. *Let $f_0, f_1, \dots, f_n \in \mathbb{R}[x]$ be inhomogeneous polynomials. Suppose the polynomial system*

$$f_0(x) = f_1(x) = \dots = f_n(x) = 0$$

has a solution in \mathbb{C}^n , then

$$\text{Res}(f_0, f_1, \dots, f_n) = 0.$$

Proof. If the polynomial system

$$f_0(x) = f_1(x) = \dots = f_n(x) = 0$$

has a solution $u \in \mathbb{C}^n$, then the polynomial system

$$\tilde{f}_0(\tilde{x}) = \tilde{f}_1(\tilde{x}) = \dots = \tilde{f}_n(\tilde{x}) = 0$$

has a nonzero solution $(1, u) \in \mathbb{C}^{n+1}$. The conclusion follows by the properties of resultant for homogeneous polynomials. \square

Proposition B.2. *Let $m \leq n$. The polynomial system*

$$f_1(x) = \dots = f_m(x) = 0$$

has a solution $u \in \mathbb{C}^n$ such that the Jacobian matrix of f_1, \dots, f_m is rank deficient at u if and only if the polynomial system

$$\tilde{f}_1(\tilde{x}) = \dots = \tilde{f}_m(\tilde{x}) = 0$$

has a solution $(1, u) \in \mathbb{C}^{n+1}$ such that the Jacobian matrix of $\tilde{f}_1, \dots, \tilde{f}_m$ is rank deficient at $(1, u)$.

Proof. Let $d_i = \deg_x(f_i)$, $f_{i,j}$ denote the homogenous part of degree j of polynomial f_i and $\tilde{f}_{i,j} = x_0^{d_i-j} f_{i,j}$ for $i = 1, \dots, m$ and $j = 0, \dots, d_i$. Denote

$$\nabla_x := \left\{ \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n} \right\} \quad \text{and} \quad \nabla_{\tilde{x}} := \left\{ \frac{\partial}{\partial x_0}, \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n} \right\}.$$

The “if” direction is implied by

$$(B.1) \quad \frac{\partial \tilde{f}_i}{\partial x_j}(1, u) = \frac{\partial f_i}{\partial x_j}(u), \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Next we prove the “only if” direction. By assumption, there exists a set of n scalars c_1, \dots, c_n , not all zero, such that

$$\sum_{i=1}^m c_i (\nabla_x f_i)(u) = 0$$

which means

$$\sum_{i=1}^m c_i \left(\sum_{j=1}^{d_i} \frac{\partial f_{i,j}}{\partial x_k}(u) \right) = 0, \quad k = 1, \dots, n.$$

Then by Euler's Homogeneous Function Theorem, we have

$$\begin{aligned}
0 &= \sum_{k=1}^n \sum_{i=1}^m c_i \left(\sum_{j=1}^{d_i} \frac{\partial f_{i,j}}{\partial x_k}(u) u_k \right) \\
&= \sum_{i=1}^m c_i \left(\sum_{j=1}^{d_i} \sum_{k=1}^n \frac{\partial f_{i,j}}{\partial x_k}(u) u_k \right) \\
&= \sum_{i=1}^m c_i \left(\sum_{j=1}^{d_i} j f_{i,j}(u) \right) \\
&= \sum_{i=1}^m c_i \left(\sum_{j=1}^{d_i} j f_{i,j}(u) + \sum_{j=0}^{d_i} (d_i - j) f_{i,j}(u) - \sum_{j=0}^{d_i} (d_i - j) f_{i,j}(u) \right) \\
&= \sum_{i=1}^m c_i \left(d_i \sum_{j=0}^{d_i} f_{i,j}(u) - \sum_{j=0}^{d_i} \frac{\partial \tilde{f}_{i,j}}{\partial x_0}(1, u) \right) \\
&= \sum_{i=1}^m c_i \left(d_i f_i(u) - \frac{\partial \tilde{f}_i}{\partial x_0}(1, u) \right) \\
&= - \sum_{i=1}^m c_i \frac{\partial \tilde{f}_i}{\partial x_0}(1, u).
\end{aligned}$$

By combining (B.1), we obtain

$$\sum_{i=1}^m c_i (\nabla_{\tilde{x}} \tilde{f}_i)(1, u) = 0$$

which concludes the proof. \square

By Proposition B.2 and the properties of discriminant for homogeneous polynomials, we have

Proposition B.3. *Let $m \leq n$ and $f_1, \dots, f_m \in \mathbb{R}[x]$ be inhomogeneous polynomials. Suppose that the polynomial system*

$$f_1(x) = \dots = f_m(x) = 0$$

has a solution in \mathbb{C}^n at which the Jacobian matrix of f_1, \dots, f_m is rank deficient, then

$$\Delta(f_1, \dots, f_m) = 0.$$

Note that the reverses of Proposition B.1 and Proposition B.3 are not necessarily true.

References

- [1] Binita Bhattacharjee, William H. Green, and Paul I. Barton. Interval methods for semi-infinite programs. *Computational Optimization and Applications*, 30(1):63–93, 2005.
- [2] Raúl E. Curto and Lawrence A. Fialkow. Truncated K -moment problems in several variables. *Journal of Operator Theory*, 54(1):189–226, 2005.
- [3] Jack Elzinga and Thomas G. Moore. A central cutting plane algorithm for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975.

- [4] Israel M. Gelfand, Mikhail Kapranov, and Andrei Zelevinsky. *Discriminants, Resultants, and Multidimensional Determinants*. Mathematics: Theory & Applications. Birkhäuser, 1994.
- [5] Keat-Choon Goh, Michael G. Safonov, and George P. Papavassilopoulos. Global optimization for the biaffine matrix inequality problem. *Journal of Global Optimization*, 7(4):365–380, 1995.
- [6] Feng Guo, Li Wang, and Guangming Zhou. Minimizing rational polynomial by exact Jacobian SDP relaxation applicable to finite singularities. *Journal of Global Optimization*, to appear. Available at <http://arxiv.org/abs/1205.6442>.
- [7] Didier Henrion and Jean B. Lasserre. *Detecting global optimality and extracting solutions in GloptiPoly*, volume 312. Springer, Berlin, 2005.
- [8] Didier Henrion and Jean B. Lasserre. Convergent relaxations of polynomial matrix inequalities and static output feedback. *IEEE Transactions on Automatic Control*, 51(2):192–202, 2006.
- [9] Didier Henrion, Jean B. Lasserre, and Johan Löfberg. GloptiPoly 3: moments, optimization and semidefinite programming. *Optimization Methods and Software*, 24(4-5):761–779, 2009.
- [10] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.
- [11] Camile W. J. Hol and Carsten W. Scherer. Sum of squares relaxations for polynomial semidefinite programming. In *International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, July 2004.
- [12] Masakazu Kojima. Sums of squares relaxations of polynomial semidefinite programs. Technical Report B-397, Department of Mathematical and Computing Sciences Tokyo Institute of Technology, Tokyo, Japan, 2003.
- [13] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.
- [14] Jean B. Lasserre. *Moments, Positive Polynomials and Their Applications*. Imperial College Press, London, UK, 2009.
- [15] Jean B. Lasserre. An algorithm for semi-infinite polynomial optimization. *TOP*, 20(1):119–129, 2012.
- [16] Marco López and Georg Still. Semi-infinite programming. *European Journal of Operational Research*, 180(2):491–518, 2007.
- [17] Jiawang Nie. Polynomial matrix inequality and semidefinite representation. *Mathematics of operations research*, 36(3):398–415, 2011.
- [18] Jiawang Nie. Discriminants and nonnegative polynomials. *Journal of Symbolic Computation*, 47(2):167–191, 2012.
- [19] Jiawang Nie. Optimality conditions and finite convergence of Lasserre’s hierarchy. *preprint*, 2012.
- [20] Jiawang Nie. An exact Jacobian SDP relaxation for polynomial optimization. *Mathematical Programming, Ser. A*, 137:225–255, 2013.
- [21] Jiawang Nie. Certifying convergence of Lasserre’s hierarchy via flat truncation. *Mathematical Programming, Ser. A*, to appear.
- [22] Panos Parpas and Berç Rustem. An algorithm for the global optimization of a class of continuous minimax problems. *Journal of Optimization Theory and Applications*, 141(2):461–473, 2009.
- [23] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42:969–984, 1993.
- [24] Georg Still. Generalized semi-infinite programming: numerical aspects. *Optimization*, 49(3):223–242, 2001.
- [25] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11/12:625–653, 1999.
- [26] Yoshihiro Tanaka, Masao Fukushima, and Toshihide Ibaraki. A globally convergent SQP method for semi-infinite nonlinear optimization. *Journal of Computational and Applied Mathematics*, 23:141–153, 1988.
- [27] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3 - a MATLAB software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1998.
- [28] Xinyuan Zhao, Defeng Sun, and Kim chuan Toh. SDPNAL version 0.1 – a MATLAB software for semidefinite programming based on a semi-smooth Newton-CG augmented Lagrangian method. <http://www.math.nus.edu.sg/~mattohkc/SDPNAL.html>.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, 9500 GILMAN DRIVE, LA JOLLA,
CA 92093.

E-mail address: `liw022@ucsd.edu`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, 9500 GILMAN DRIVE, LA JOLLA,
CA 92093.

E-mail address: `f1guo@math.ucsd.edu`.