Journal of the Operations Research Society of Japan
Vol. 54, No. 4, December 2011, pp. 161–190

AN EXTENSION OF THE ELIMINATION METHOD FOR A SPARSE SOS POLYNOMIAL

Hayato Waki Masakazu Muramatsu
The University of Electro-Communications

(Received June 27, 2011; Revised September 22, 2011)

Abstract We propose a method to reduce the sizes of SDP relaxation problems for a given polynomial optimization problem (POP). This method is an extension of the elimination method for a sparse SOS polynomial in [8] and exploits sparsity of polynomials involved in a given POP. In addition, we show that this method is a partial application of a facial reduction algorithm, which generates a smaller SDP problem with an interior feasible solution. In general, SDP relaxation problems for POPs often become highly degenerate because of a lack of interior feasible solutions. As a result, the resulting SDP relaxation problems obtained by this method may have an interior feasible solution, and one may be able to solve the SDP relaxation problems effectively. Numerical results in this paper show that the resulting SDP relaxation problems obtained by this method can be solved fast and accurately.

Keywords: Nonlinear programming, semidefinite programming, semidefinite programming relaxation, polynomial optimization, facial reduction algorithm

1. Introduction

The problem of detecting whether a given polynomial is globally nonnegative or not, appears in various fields in science and engineering. For such problems, Parrilo [11] proposed an approach via semidefinite programming (SDP) and sums of squares (SOS) of polynomials. Indeed, he replaced the problem by another problem of detecting whether a given polynomial can be represented as an SOS polynomial or not. If the answer is yes, then the global nonnegativity of the polynomial is guaranteed. It is known in Powers and Wörmann [13] that the latter problem can be converted as an SDP problem equivalently. Therefore, one can solve the latter problem by using existing SDP solvers, such as SeDuMi [18], SDPA [5], SDPT3 [19] and CSDP [1]. However, in the case where the given polynomial is large-scale, *i.e.*, the polynomial has a lot of variables and/or higher degree, the resulting SDP problem becomes too large-scale to be handled by the state of the arts computing technology, practically.

To recover this difficulty, for a sparse polynomial, *i.e.*, a polynomial which has few monomials, Kojima *et al.* in [8] proposed an effective method for reducing the size of the resulting SDP problem by exploiting the sparsity of the given polynomial. Following [23], we call the method the *elimination method for a sparse SOS polynomial* (EMSSOSP).

In this paper, we deal with the problem to detect whether a given polynomial is nonnegative over a given semialgebraic set or not. More precisely, for given polynomials $f, f_1, \ldots, f_m$, the problem is to detect whether f is nonnegative over the set $D := \{x \in \mathbb{R}^n \mid f_1(x) \geq 0, \ldots, f_m(x) \geq 0\}$ or not. For this problem, we apply a similar argument in

Parrilo [11]. Then we obtain the following problem:

$$\begin{cases} \text{Find} & \sigma_j \text{ for } j = 1, \dots, m, \\ \text{subject to} & f(x) = \sum_{j=1}^{m} f_j(x)\sigma_j(x) \ (\forall x \in \mathbb{R}^n), \\ & \sigma_j \text{ is an SOS polynomial for } j = 1, \dots, m. \end{cases} \tag{1.1}$$

If we find all SOS polynomials σ_j in (1.1), then the given polynomial f is nonnegative over the set D. However, (1.1) is still intractable because the degree of each σ_j is unknown in advance. Therefore, we put a bound of the degree of σ_j. Then, we can convert (1.1) to an SDP problem by applying Lemma 2.1 in [8] or Theorem 1 in [13] to the problem. In this case, we have the same computational difficulty as the problem of finding an SOS representation of a given polynomial, namely the resulting SDP problem becomes large-scale if $f, f_1, \dots, f_m$ has a lot of variables and/or higher degree.

The contribution of this paper is to propose a method for reducing the size of the resulting SDP problem if $f, f_1, \dots, f_m$ are sparse. To this end, we extend EMSSOSP, so that the proposed method removes unnecessary monomials in any representation of f with $f_1, \dots, f_m$ from σ_j ($j = 1, \dots, m$). If $f, f_1, \dots, f_m$ are sparse, then we can expect that the resulting SDP problem obtained by our proposed method becomes smaller than the SDP problem obtained from (1.1). In this paper, we call our proposed method *EEM*, which stands for the Extension of Elimination Method.

Another contribution of this paper is to show that EEM is a partial application of a facial reduction algorithm (FRA). FRA was proposed by Borwein and Wolkowicz [2, 3]. Ramana *et al.* [17] showed that FRA for SDP with nonempty feasible region generates an equivalent SDP with an interior feasible solution. In addition, Pataki [12] simplified FRA of Borwein and Wolkowicz. Waki and Muramatsu [22] proposed FRA for conic optimization problems. It is pointed out in [23] that EMSSOSP is a partial application of FRA. In general, SDP relaxation problems for polynomial optimization problems (POPs) become highly degenerate because of a lack of interior feasible solutions. As a consequence, the resulting SDP problems obtained by EEM may have an interior feasible solution, and thus we can expect an improvement on the computational efficiency of primal-dual interior-point methods.

The organization of this paper is as follows: We discuss our problem and usage of the sparsity of given polynomials $f, f_1, \dots, f_m$ in Section 2 and propose EEM in Section 3. SDP relaxations [9, 20] for POPs are applications of EEM. We apply EEM to some test problems in GLOBAL Library [6] and randomly generated problems, and solve the resulting SDP relaxation problems by SeDuMi [18]. In Section 4, we present the numerical results. When we execute EEM to remove unnecessary monomials in SOS representations, there is a flexibility in EEM. We focus on the flexibility and show facts on SDP relaxation for POPs and SOS representations of a given polynomial in Section 5. A relationship between our method and FRA is provided in Section 6. Section 7 is devoted to concluding remarks. We give some discussion and proofs on EEM in Appendix.

1.1. Notation and symbols

$\mathbb{S}^n$ and $\mathbb{S}^n_+$ denote the sets of $n \times n$ symmetric matrices and $n \times n$ symmetric positive semidefinite matrices, respectively. For $X, Y \in \mathbb{S}^n$, we define $X \bullet Y = \sum_{i,j=1}^{n} X_{ij}Y_{ij}$. For every finite set A, $\#(A)$ denotes the number of elements in A. We define $A+B = \{a+b \mid a \in A, b \in B\}$ for $A, B \subseteq \mathbb{R}^n$. We remark that $A+B = \emptyset$ when either A or B is empty. For $A \subseteq \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, αA denotes the set $\{\alpha a \mid a \in A\}$. Let $\mathbb{N}^n$ be the set of n-dimensional nonnegative integer vectors. We define $\mathbb{N}^n_r := \{\alpha \in \mathbb{N}^n \mid \sum_{i=1}^n \alpha_i \leq r\}$. Let $\mathbb{R}[x]$ be the set of polynomials with n-dimensional real vector x. For every $\alpha \in \mathbb{N}^n$, x^α denotes the monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. For $f \in \mathbb{R}[x]$, let F be the set of exponents α of monomials x^α whose coefficients f_α are nonzero. Then we can write $f(x) = \sum_{\alpha \in F} f_\alpha x^\alpha$. We call F the *support* of f. The degree $\deg(f)$ of f is the maximum value of $|\alpha| := \sum_{i=1}^n \alpha_i$ over the support F. For $G \subseteq \mathbb{N}^n$, $\mathbb{R}_G[x]$ denotes the set of polynomials whose supports are contained in G. In particular, $\mathbb{R}_r[x]$ is the set of polynomials with the degree up to r.

2. On Our Problem and Exploiting the Sparsity of Given Polynomials $f, f_1, \ldots, f_m$

2.1. Our problem

In this subsection, we discuss how to convert (1.1) into an SDP problem. For a finite set $G \subseteq \mathbb{N}^n$, let Σ_G be the set of SOS polynomials whose supports are contained in G. In particular, we denote Σ_r if $G = \mathbb{N}^n_r$. Let $d = \lceil \deg(f)/2 \rceil$, $d_j = \lceil \deg(f_j)/2 \rceil$ for all $j = 1, \ldots, m$ and $\bar{r} = \max\{d, d_1, \ldots, d_m\}$. We fix a positive integer $r \geq \bar{r}$ and define $r_j = r - d_j$ for all $j = 1, \ldots, m$. Then we obtain the following SOS problem from (1.1):

$$\left\{ \begin{array}{ll} \text{Find} & \sigma_j \in \Sigma_{r_j} \text{ for all } j = 1, \ldots, m, \\ \text{subject to} & f(x) = \displaystyle\sum_{j=1}^m f_j(x)\sigma_j(x) \ (\forall x \in \mathbb{R}^n). \end{array} \right. \tag{2.1}$$

We say that f has *an SOS representation with* $f_1, \ldots, f_m$ if (2.2) has a solution $(\sigma_1, \ldots, \sigma_m)$. In this case, (1.1) also has a solution, and thus f is nonnegative over the set D.

We assume that $\sigma_j \in \Sigma_{G_j}$ in any SOS representations of f with $f_1, \ldots, f_m$. Then we can obtain the following SOS problem from (2.1):

$$\left\{ \begin{array}{ll} \text{Find} & \sigma_j \in \Sigma_{G_j} \text{ for all } j = 1, \ldots, m, \\ \text{subject to} & f(x) = \displaystyle\sum_{j=1}^m f_j(x)\sigma_j(x) \ (\forall x \in \mathbb{R}^n). \end{array} \right. \tag{2.2}$$

Note that SOS problem (2.2) is equivalent to SOS problem (2.1) by the assumption.

Let F and F_j be the support of f and f_j, respectively. Without loss of generality, we assume $F \subseteq \bigcup_{j=1}^m (F_j + G_j + G_j)$. In fact, if $F \not\subseteq \bigcup_{j=1}^m (F_j + G_j + G_j)$, then (2.2) does not have any solutions.

To reformulate (2.2) into an SDP problem, we use the following lemma:

Lemma 2.1 (Lemma 2.1 in [8]) *Let G be a finite subset of $\mathbb{N}^n$ and $u(x, G) = (x^\alpha : \alpha \in G)$. Then, f is in Σ_G if and only if there exists a positive semidefinite matrix $V \in \mathbb{S}_+^{\#(G)}$ such that $f(x) = u(x, G)^T V u(x, G)$ for all $x \in \mathbb{R}^n$.*

We apply Lemma 2.1 to σ_j in (2.2). Then we can reformulate (2.2) into the following problem:

$$\begin{cases} \text{Find} & V_j \in \mathbb{S}_+^{\#(G_j)} \text{ for all } j = 1, \dots, m, \\ \text{subject to} & f(x) = \sum_{j=1}^{m} u(x, G_j)^T V_j u(x, G_j) f_j(x) \ (\forall x \in \mathbb{R}^n), \end{cases} \tag{2.3}$$

where $u(x, G_j)$ be the column vector consisting of all monomials x^α ($\alpha \in G_j$). We define the matrices $E_{j,\alpha} \in \mathbb{S}^{\#(G_j)}$ for all $\alpha \in \bigcup_{j=1}^m (F_j + G_j + G_j)$ and for all $j = 1, \dots, m$ as follows:

$$(E_{j,\alpha})_{\beta,\gamma} = \begin{cases} (f_j)_\delta & \text{if } \alpha = \beta + \gamma + \delta \text{ and } \delta \in F_j, \\ 0 & \text{o.w.} \end{cases} \quad (\text{for all } \beta, \gamma \in G_j).$$

Comparing the coefficients in both sides of the identity in (2.3), we obtain the following SDP:

$$\begin{cases} \text{Find} & V_j \in \mathbb{S}_+^{\#(G_j)} \text{ for all } j = 1, \dots, m, \\ \text{subject to} & f_\alpha = \sum_{j=1}^{m} E_{j,\alpha} \bullet V_j, \ (\alpha \in \bigcup_{j=1}^{m} (F_j + G_j + G_j)). \end{cases} \tag{2.4}$$

We observe from (2.4) that the resulting SDP (2.4) may become small enough to handle it if G_j is much smaller than $\mathbb{N}_{r_j}^n$ for all $j = 1, \dots, m$.

Remark 2.2 Some sets G_j may be empty. For instance, if $G_1 = \emptyset$, then the problem (2.1) is equivalent to the problem of finding an SOS representation of f with $f_2, \dots, f_m$ under the condition $\sigma_j \in \Sigma_{r_j}$.

2.2. Exploiting the sparsity of given polynomials $f, f_1, \dots, f_m$

We present a lemma which plays an essential role in our proposed method EEM. EEM applies this lemma to (2.1) repeatedly, so that we obtain (2.2). This lemma is an extension of Corollary 3.2 in [8] and Proposition 3.7 in [4].

Lemma 2.3 *Let $G_j \subseteq \mathbb{N}^n$ be a finite set for all $j = 1, \dots, m$. f and f_j denote polynomials with support F and F_j, respectively. For $\delta \in \bigcup_{j=1}^m (F_j + G_j + G_j)$, we define $J(\delta) \subseteq \{1, \dots, m\}, B_j(\delta) \subseteq G_j$ and $T_j \subseteq F_j + G_j + G_j$ as follows:*

$$\begin{aligned} J(\delta) &:= \{j \in \{1, \dots, m\} \mid \delta \in F_j + 2G_j\}, \\ B_j(\delta) &:= \{\alpha \in G_j \mid \delta - 2\alpha \in F_j\}, \\ T_j &:= \{\gamma + \alpha + \beta \mid \gamma \in F_j, \alpha, \beta \in G_j, \alpha \neq \beta\}. \end{aligned}$$

Assume that f has an SOS representation with $f_1, \dots, f_m$ and $G_1, \dots, G_m$ as follows:

$$f(x) = \sum_{j=1}^{m} f_j(x) \sum_{i=1}^{k_j} \left(\sum_{\alpha \in G_j} (g_{j,i})_\alpha x^\alpha \right)^2, \tag{2.5}$$

where $(g_{j,i})_\alpha$ is the coefficient of the polynomial $g_{j,i}$. In addition, we assume that for a fixed $\delta \in \bigcup_{j=1}^m (F_j + G_j + G_j) \setminus (F \cup \bigcup_{j=1}^m T_j)$, $J(\delta)$ and $(B_1(\delta), \dots, B_m(\delta))$ satisfy

$$\begin{cases} (f_j)_{\delta - 2\alpha} > 0 \text{ for all } j \in J(\delta) \text{ and } \alpha \in B_j(\delta) \text{ or} \\ (f_j)_{\delta - 2\alpha} < 0 \text{ for all } j \in J(\delta) \text{ and } \alpha \in B_j(\delta). \end{cases} \tag{2.6}$$

Then, f has an SOS representation with $f_1, \ldots, f_m$ and $G_1 \setminus B_1(\delta), \ldots, G_m \setminus B_m(\delta)$, i.e., $(g_{j,i})_\alpha = 0$ for all $j \in J(\delta), \alpha \in B_j(\delta)$ and $i = 1, \ldots, k_j$ and

$$f(x) = \sum_{j=1}^{m} f_j(x) \sum_{i=1}^{k_j} \left(\sum_{\alpha \in G_j \setminus B_j(\delta)} (g_{j,i})_\alpha x^\alpha \right)^2. \tag{2.7}$$

We postpone a proof of Lemma 2.3 until Appendix A.

Remark 2.4 We have the following remarks on Lemma 2.3.

1. If f is not sparse, *i.e.*, f has a lot of monomials with nonzero coefficient, then the set $\bigcup_{j=1}^m (F_j + G_j + G_j) \setminus (F \cup \bigcup_{j=1}^m T_j)$ may be empty. In this case, we do not have any candidates δ for (2.6). In addition, if $f_1, \ldots, f_m$ are sparse, then coefficients to be checked in (2.6) may be few in number, and thus we can expect that there exists δ such that $J(\delta)$ and $B_j(\delta)$ satisfy (2.6).
2. Lemma 2.3 is an extension to Corollary 3.2 in Kojima *et al.* [8] and (2) of Proposition 3.7 in Choi *et al.* [4]. In fact, the authors deal with the case where $m = 1$ and $f_1 = 1$ in these papers. In that case, we have $F_1 = \{0\}$ and the coefficient $(f_1)_0$ of a constant term in f is 1.

We give an example of notation $J(\delta)$, $B_j(\delta)$ and T_j.

Example 2.5 Let $f = x, f_1 = 1, f_2 = x$ and $f_3 = x^2 - 1$. Then we have $F = \{1\}, F_1 = \{0\}, F_2 = \{1\}$ and $F_3 = \{0, 2\}$.

We consider the case where we have $G_1 = \{0, 1, 2\}, G_2 = G_3 = \{0, 1\}$. Then we obtain

$$\left(\bigcup_{j=1}^{3} (F_j + G_j + G_j) \right) \setminus F = \{0, 2, 3, 4\}, T_1 = \{1, 2, 3\}, T_2 = \{2\} \text{ and } T_3 = \{1, 3\}.$$

In this case, we can choose $\delta \in \{0, 4\}$. If we choose $\delta = 4$, then $J(\delta) = \{1, 3\}$ and we obtain $B_1(\delta) = \{2\}, B_2(\delta) = \emptyset$ and $B_3(\delta) = \{1\}$. Moreover, $J(\delta)$ and $(B_1(\delta), B_2(\delta), B_3(\delta))$ satisfy (2.6). Lemma 2.3 implies that if f has an SOS representation with f_j and G_j, then f also has an SOS representation with f_j, $G_1 \setminus B_1(\delta) = \{0, 1\}$, $G_2 \setminus B_2(\delta) = \{0, 1\}$ and $G_3 \setminus B_3(\delta) = \{0\}$.

3. An Extension of EMSSOSP

For given polynomials $f, f_1, \ldots, f_m$ and a positive integer $r \geq \bar{r}$, we set $r_j = r - \lceil \deg(f_j)/2 \rceil$ for all j. We assume that f can be represented as follows:

$$f(x) = \sum_{j=1}^{m} f_j(x) \sigma_j(x) \tag{3.1}$$

for some $\sigma_j \in \Sigma_{r_j}$. We remark that the support of σ_j is contained in $\mathbb{N}^n_{2r_j}$. By applying Lemma 2.3 repeatedly, our method may remove unnecessary monomials of σ_j in (3.1) for all SOS representations of f with $f_1, \ldots, f_m$ and $G_1, \ldots, G_m$ before deciding all coefficients of σ_j. We give the detail of our method in Algorithm 3.1.

Algorithm 3.1 (The elimination method for a sparse SOS representation with $f_1, \ldots, f_m$)

Input *polynomials* $f, f_1, \dots, f_m$.
Output $G^* := (G_1^*, \dots, G_m^*)$.
Step 1 *Set* $i = 0$ *and* $G^0 := (\mathbb{N}^n_{r_1}, \dots, \mathbb{N}^n_{r_m})$.
Step 2 *For* $G^i = (G_1^i, \dots, G_m^i)$, *if there does not exist any* $\delta \in \bigcup_{j=1}^m (F_j + G_j^i + G_j^i) \setminus (F \cup \bigcup_{j=1}^m T_j^i)$ *such that* $B(\delta) = (B_1(\delta), \dots, B_m(\delta))$ *and* $J(\delta)$ *satisfy (2.6) and* $B_j(\delta) \neq \emptyset$ *for some* $j = 1, \dots, m$, *then stop and return* G^i.
Step 3 *Otherwise set* $G_j^{i+1} = G_j^i \setminus B_j(\delta)$ *for all* $j = 1, \dots, m$, *and* $i = i + 1$, *and go back to Step 2.*

We call Algorithm 3.1 *EEM* in this paper. In this section, we show that EEM always returns the smallest set $(G_1^*, \dots, G_m^*)$ in a set family. To this end, we give some notation and definitions, and use some results in Appendix B.

For $G = (G_1, \dots, G_m), H = (H_1, \dots, H_m) \subseteq X := \mathbb{N}^n_{r_1} \times \cdots \times \mathbb{N}^n_{r_m}$, we define $G \cap H := (G_1 \cap H_1, \dots, G_m \cap H_m)$, $G \setminus H := (G_1 \setminus H_1, \dots, G_m \setminus H_m)$ and $G \subseteq H$ if $G_j \subseteq H_j$ for all $j = 1, \dots, m$.

Definition 3.2 *We define a function* $P : 2^X \times 2^X \to \{\mathit{true}, \mathit{false}\}$ *as follows:*

$$P(G, B) = \begin{cases} \mathit{true} & \text{if all } B_j \text{ are empty or there exists } \delta \in \bigcup_{j=1}^m (F_j + G_j + G_j) \setminus (F \cup \\ & \bigcup_{j=1}^m T_j) \text{ such that } B = B(\delta) \text{ and, } B \text{ and } J(\delta) \text{ satisfy } (2.6), \\ \mathit{false} & \text{otherwise} \end{cases}$$

for all $G = (G_1, \dots, G_m), B := (B_1, \dots, B_m) \subseteq X$. *Moreover, let* $G^0 := (\mathbb{N}^n_{r_1}, \dots, \mathbb{N}^n_{r_m})$. *We define a set family* $\Gamma(G^0, P) \subseteq 2^X$ *as follows:* $G \in \Gamma(G^0, P)$ *if and only if* $G = G^0$ *or there exists* $G' \in \Gamma(G^0, P)$ *such that* $G \subsetneq G'$ *and* $P(G', G' \setminus G) = \mathit{true}$.

The following theorem guarantees that EEM always returns the smallest set $(G_1^*, \dots, G_m^*) \in \Gamma(G^0, P)$.

Theorem 3.3 *Let* P *and* $G(G^0, P)$ *be as in Definition 3.2. Assume that* f *has an SOS representation with* $f_1, \dots, f_m$ *and* G^0. *Then,* $\Gamma(G^0, P)$ *has the smallest set* G^* *in the sense that* $G^* \subseteq G$ *for all* $G \in \Gamma(G^0, P)$. *In addition, EEM described in Algorithm 3.1 always returns the smallest set* G^* *in the set family* $\Gamma(G^0, P)$.

For this proof, we use some results in Appendix B. We postpone the proof till Appendix C.

We give an example to see a behavior of EEM.

Example 3.4 (Continuation of Example 2.5) Let $f = x, f_1 = 1, f_2 = x$ and $f_3 = x^2 - 1$. Clearly, f is nonnegative over the set $D = \{x \in \mathbb{R} \mid f_1, f_2, f_3 \geq 0\} = [1, +\infty)$. Let $r = 2$. Then we have $r_1 = 2, r_2 = r_3 = 1$. We consider the following SOS problem:

$$\begin{cases} \text{Find} & \sigma_j \in \Sigma_{r_j} \text{for } j = 1, 2, 3, \\ \text{subject to} & f(x) = \sigma_1(x) f_1(x) + \sigma_2(x) f_2(x) + \sigma_3(x) f_3(x) \ (\forall x \in \mathbb{R}). \end{cases} \tag{3.2}$$

The initial $G^0 = (\mathbb{N}_2, \mathbb{N}_1, \mathbb{N}_1)$. From Example 2.5, we have already known $\delta^0 = 4$, $G_1^1 = G_1^0 \setminus B_1^0(\delta^0) = \{0, 1\}$ and $G^1 = (\{0, 1\}, \{0, 1\}, \{0\})$.

Table 1 shows $\delta, J(\delta), B_j(\delta)$ and T_j in Example 3.4 in the ith iteration of EEM for the identity of (3.2).

For $G^1 \in \Gamma(G^0, P)$, we choose $\delta^1 = 3 \in \bigcup_{j=1}^3 (F_j + G_j^1 + G_j^1) \setminus (F \cup \bigcup_{j=1}^3 T_j^1) = \{0, 3\}$. Then we obtain $J(\delta^1)$ and $B_j^1(\delta^1)$ in the third row of Table 1 and $G^2 = (\{0, 1\}, \{0\}, \{0\})$.

For $G^2 \in \Gamma(G^0, P)$, we choose $\delta^2 = 2 \in \bigcup_{j=1}^3 (F_j + G_j^2 + G_j^2) \setminus (F \cup \bigcup_{j=1}^3 T_j^2) = \{0, 2\}$. Then we obtain $J(\delta^2)$ and $B_j^2(\delta^2)$ in the fourth row of Table 1 and $G^3 = (\{0\}, \{0\}, \emptyset)$. This implies that f_3 is redundant for all SOS representations of f with f_1, f_2, f_3 and G^3.

For $G^3 \in \Gamma(G^0, P)$, we choose $\delta^3 = 0 \in \bigcup_{j=1}^3 (F_j + G_j^3 + G_j^3) \setminus (F \cup \bigcup_{j=1}^3 T_j^3) = \{0\}$. Then we obtain $J(\delta^3)$ and $B_j^3(\delta^3)$ in the fifth row of Table 1 and $G^4 = (\emptyset, \{0\}, \emptyset)$. This implies that f_1 is redundant for all SOS representations of f with f_1, f_2, f_3 and G^4.

For $G^4 \in \Gamma(G^0, P)$, because the set $\bigcup_{j=1}^3 (F_j + G_j^4 + G_j^4) \setminus (F \cup \bigcup_{j=1}^3 T_j^4)$ is empty, EEM stops and returns $G^* = (\emptyset, \{0\}, \emptyset)$. Then by using G^*, from SOS problem (3.2), we obtain the following Linear Programming (LP):

$$\begin{cases} \text{Find} & \lambda_2 \geq 0 \\ \text{subjecto to} & \lambda_2 = 1. \end{cases}$$

Because this LP has the solution $\lambda_2 = 1$, SOS problem (3.2) has a solution $(\sigma_1, \sigma_2, \sigma_3) = (0, 1, 0)$. This implies that $f = 0 \cdot f_1 + 1 \cdot f_2 + 0 \cdot f_3$.

Table 1: $\delta, J(\delta), B_j(\delta)$ and T_j in Example 3.4

	δ^i	$J(\delta^i)$	$B_1^i(\delta^i)$	$B_2^i(\delta^i)$	$B_3^i(\delta^i)$	T_1^i	T_2^i	T_3^i
$i=0$	4	$\{1,3\}$	$\{2\}$	$\emptyset$	$\{1\}$	$\{1,2,3\}$	$\{2\}$	$\{1,3\}$
$i=1$	3	$\{2\}$	$\emptyset$	$\{1\}$	$\emptyset$	$\{1\}$	$\{2\}$	$\emptyset$
$i=2$	2	$\{1,3\}$	$\{1\}$	$\emptyset$	$\{0\}$	$\{1\}$	$\emptyset$	$\emptyset$
$i=3$	0	$\{1\}$	$\{0\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$i=4$	$-$	$-$	$-$	$-$	$-$	$\emptyset$	$\emptyset$	$\emptyset$

4. Numerical Results for Some POPs

In this section, we present the numerical performance of EEM for Lasserre's and sparse SDP relaxations for Polynomial Optimization Problems (POPs) in [6] and randomly generated POPs with a sparse structure. To this end, we explain how to apply EEM to SDP relaxations for POPs.

For given polynomials f, f_1, $\ldots$, f_m, POP is formulated as follows:

$$\inf_{x \in \mathbb{R}^n} \{f(x) \mid f_j(x) \geq 0 \ (j = 1, \ldots, m)\}. \tag{4.1}$$

We can reformulate POP (4.1) into the following problem:

$$\sup_{\rho \in \mathbb{R}} \{\rho \mid f(x) - \rho \geq 0 \ (\forall x \in D)\}. \tag{4.2}$$

Here D is the feasible region of POP (4.1). We choose an integer r with $r \geq \bar{r}$. For (4.2) and r, we consider the following SOS problem:

$$\rho_r^* := \sup_{\rho \in \mathbb{R}, \sigma_j \in \Sigma_{r_j}} \left\{ \rho \,\middle|\, f(x) - \rho = \sigma_0 + \sum_{j=1}^m \sigma_j(x) f_j(x) \ (\forall x \in \mathbb{R}^n) \right\} \tag{4.3}$$

where we define $r_j = r - \lceil \deg(f_j)/2 \rceil$ for all $j = 1, \ldots, m$ and $r_0 = r$. If we find a feasible solution $(\rho, \sigma_0, \sigma_1, \ldots, \sigma_m)$ of (4.3), then ρ is a lower bound of optimal value of (4.1), clearly. It follows that $\rho^*_{r+1} \geq \rho^*_r$ for all $r \geq \bar{r}$ because we have $\Sigma_k \subseteq \Sigma_{k+1}$ for all $k \in \mathbb{N}$.

We apply EEM to the identity in (4.3). Then, we can regard $f(x) - \rho$ in (4.3) as a polynomial with variable x. It should be noted that the support F of $f(x) - \rho$ always contains 0 because $-\rho$ is regarded as a constant term in the identity in (4.3). Moreover, let $f_0(x) = 1$ for all $x \in \mathbb{R}^n$. We replace $\sigma_0(x)$ by $\sigma_0(x)f_0(x)$ in the identity in (4.3). Then we can apply EEM directly, so that we obtain finite sets $G^*_0, G^*_1, \ldots, G^*_m$. We can construct an SDP problem by using $G^*_0, G^*_1, \ldots, G^*_m$ and applying a similar argument described in subsection 2.1.

It was proposed in [20] to construct a smaller SDP relaxation problem than Lasserre's SDP relaxation when a given POP has a special sparse structure called correlative sparsity. We call their method the WKKM sparse SDP relaxation for POPs in this paper. In the numerical experiments, we have tested EEM applied to both Lasserre's and the WKKM sparse SDP relaxations.

We use a computer with Intel (R) Xeon (R) 2.40 GHz cpus and 24GB memory, and Matlab R2009b and SeDuMi 1.21 with the default parameters to solve the resulting SDP relaxation problems. In particular, the default tolerance for stopping criterion of SeDuMi is 1.0e-9. We use SparsePOP [21] to make SDP relaxation problems. To see the quality of the approximate solution obtained by SeDuMi, we check DIMACS errors. If the six errors are sufficiently small, then the solution is regarded as an optimal solution. See [10] for the definitions.

To check whether the optimal value of an SDP relaxation problem is the exact optimal value of a given POP or not, we use the following two criteria $\epsilon_{\rm obj}$ and $\epsilon_{\rm feas}$: Let $\hat{x}$ be a candidate of an optimal solution of the POP obtained by Lasserre's or the WKKM sparse SDP relaxation. See [20] for the way to obtain $\hat{x}$. We define:

$$
\begin{aligned}
\epsilon_{\rm obj} &:= \frac{|\text{the optimal value of the SDP relaxation} - f(\hat{x})|}{\max\{1, f(\hat{x})\}}, \\
\epsilon_{\rm feas} &:= \min\{f_k(\hat{x}) \ (k = 1, \ldots, m)\}.
\end{aligned}
$$

If $\epsilon_{\rm feas} = 0$, then $\hat{x}$ is feasible for the POP. In addition, if $\epsilon_{\rm obj} = 0$, then $\hat{x}$ is an optimal solution of the POP and $f(\hat{x})$ is the optimal value of the POP.

Some POPs in [6] are so badly scaled that the resulting SDP relaxation problems suffer severe numerical difficulty. We may obtain inaccurate values and solutions for such POPs. To avoid this difficulty, we apply a linear transformation to the variables in POP with finite lower and upper bounds on variables x_i $(i = 1, \ldots, n)$. See [24] for the effect of such transformations.

Although EMSSOSP is designed for an unconstrained POP, we can apply it to POP (4.1) in such a way that it removes unnecessary monomials in σ_0 in (4.3). It is presented in subsection 6.3 of [20] that such application of EMSSOSP is effective for a large-scale POP. In this section, we also compare EEM with EMSSOSP.

Table 2 shows notation used in the description of the numerical results in subsections 4.1 and 4.2.

Table 2: Notation

Method	a method for reducing the size of the SDP relaxation problems. "Orig." means that we do not apply EMSSOSP and EEM to POPs. "EMSSOSP" and "EEM" mean that we apply EMSSOSP and EEM to POPs, respectively.
sizeA	the size of coefficient matrix A in the SeDuMi input format
nnzA	the number of nonzero elements in coefficient matrix A in the SeDuMi input format
#LP	the number of linear constraints in the SeDuMi input format
#SDP	the number of positive semidefinite constraints in the SeDuMi input format
a.SDP	the average of the sizes of positive semidefinite constraints in the SeDuMi input format
m.SDP	the maximum of the sizes of positive semidefinite constraints in the SeDuMi input format
SDPobj	the objective value obtained by SeDuMi for the resulting SDP relaxation problem
POPobj	the value of f at a solution $\hat{x}$ retrieved by SparesPOP
eTime	cpu time consumed by SeDuMi or SDPA-GMP in seconds
n.e.	n.e. = 1 if SeDuMi cannot find an accurate solution due to numerical difficulty. Otherwise, n.e. = 0
p.v.	phase value returned by SDPA-GMP. If it is "pdOPT", then SDPA-GMP terminates normally.
iter.	the number of iterations in SeDuMi

4.1. Numerical results for GLOBAL Library

In this numerical experiment, we solve some POPs in GLOBAL Library [6]. This library contains POPs which have polynomial equalities $h(x) = 0$. In this case, we divide $h(x) = 0$ into two polynomial inequalities $h(x) \geq 0$ and $-h(x) \geq 0$ and replace them by their polynomial inequalities in the original POP. We remark that in our tables, a POP whose initial is "B" is obtained by adding some lower and upper bounds. In addition, we do not apply the WKKM sparse SDP relaxation to POPs "Bex3_1_4", "st_e01", "st_e09" and "st_e34" because their POPs do not have the sparse structure.

Tables 3 and 4 show the numerical results for Lasserre's SDP relaxation [9] for some POPs in [6]. Tables 5 and 6 show the numerical results for the WKKM sparse SDP relaxation [20] for some POPs in [6].

We observe the following.

- From Tables 4 and 6, the sizes of SDP relaxation problems obtained by EEM are the smallest of the three for all POPs. As a result, EEM spends the least cpu time to solve the resulting SDP problems except for Bst_07 on Table 3 and Bex_5_2_2_case2 on Table 5. Table 4 tells us that EEM needs one more iteration than that by EMSSOSP for Bst_07. Looking at the behavior of SeDuMi in this case carefully, we noticed that SeDuMi consumes much more CPU time in the last iteration for computing the search direction than in the other iterations. Also in the case of Bex_5_2_2_case2, EEM needs three more iterations than that by EMSSOSP. Exact reasons of them should be investigated in further research.
- For all POPs except for Bex5_2_2_case1, 2, 3, the optimal values of the SDP relaxation problems are the same. For the WKKM sparse SDP relaxation of Bex5_2_2_case1, 2, 3, all three methods cannot obtain accurate solutions; their DIMACS errors are not sufficiently small. Consequently, these computed optimal values are considered to be inaccurate.
- From Tables 4 and 6, for almost all POPs, DIMACS errors for SDP relaxation problems obtained by EEM are smaller than the other methods. This means that SeDuMi returns more accurate solutions for the resulting SDP relaxation problems by EEM.
- We cannot obtain optimal values and solutions for some POPs, *e.g.*, Bex5_2_2_case1, 2, 3. In contrast, the optimal values and solutions for them are obtained in [20]. At a glance, it seems that the numerical result for some POPs may be worse than [20]. The reason is as follows: In [20], the authors add some valid polynomial inequalities to POPs and apply Lasserre's or the WKKM sparse SDP relaxation, so that they obtain tighter lower bounds or the exact values. See Section 5.5 in [20] for the details. In the experiments of this section, however, we do not add such valid polynomial inequalities in order to observe the efficiency of EMSSOSP and EEM.

Table 3: Numerical results of Lasserre's SDP relaxation problems

Problem	r	Method	SDPobj	POPobj	ϵ_{obj}	ϵ_{feas}	ϵ'_{feas}	eTime
Bex3_1_1	3	Orig.	7.049248e+03	7.049248e+03	2.5e-11	-3.662e-04	-1.812e-10	157.40
Bex3_1_1	3	EMSSOSP	7.049248e+03	7.049248e+03	2.7e-11	-4.041e-04	-1.999e-10	44.35
Bex3_1_1	3	EEM	7.049248e+03	7.049248e+03	6.2e-12	-1.024e-04	-5.064e-11	20.22
Bex3_1_4	4	Orig.	-4.000002e+00	-4.000002e+00	5.4e-10	-1.069e-03	-2.741e-05	0.56
Bex3_1_4	4	EMSSOSP	-4.000000e+00	-4.000000e+00	3.7e-10	-2.207e+00	-9.194e-02	0.55
Bex3_1_4	4	EEM	-4.000000e+00	-4.000000e+00	4.4e-09	-2.193e+00	-9.139e-02	0.42
Bex5_2_2_case1	2	Orig.	-4.000001e+02	-4.000002e+02	3.8e-07	-4.901e-02	-4.791e-04	3.47
Bex5_2_2_case1	2	EMSSOSP	-4.000001e+02	-4.000002e+02	2.2e-07	-3.743e-02	-3.663e-04	0.42
Bex5_2_2_case1	2	EEM	-4.000001e+02	-4.000002e+02	2.2e-07	-3.743e-02	-3.663e-04	0.42
Bex5_2_2_case2	2	Orig.	-6.000000e+02	-6.000001e+02	1.3e-07	-2.405e-02	-1.919e-04	3.35
Bex5_2_2_case2	2	EMSSOSP	-6.000001e+02	-6.000002e+02	2.6e-07	-4.543e-02	-3.790e-04	0.44
Bex5_2_2_case2	2	EEM	-6.000001e+02	-6.000002e+02	2.6e-07	-4.543e-02	-3.790e-04	0.44
Bex5_2_2_case3	2	Orig.	-7.500000e+02	-7.500000e+02	9.8e-09	-4.246e-03	-1.415e-05	3.03
Bex5_2_2_case3	2	EMSSOSP	-7.500000e+02	-7.500000e+02	1.2e-08	-2.391e-03	-7.968e-06	0.37
Bex5_2_2_case3	2	EEM	-7.500000e+02	-7.500000e+02	1.2e-08	-2.391e-03	-7.968e-06	0.36
Bst_e07	2	Orig.	-1.809184e+03	-1.809184e+03	8.4e-12	-1.125e-05	-1.837e-08	4.91
Bst_e07	2	EMSSOSP	-1.809184e+03	-1.809184e+03	1.5e-11	-2.012e-05	-3.286e-08	0.64
Bst_e07	2	EEM	-1.809184e+03	-1.809184e+03	3.5e-12	-4.666e-06	-7.621e-09	0.75
Bst_e33	2	Orig.	-4.000000e+02	-4.000000e+02	8.6e-11	-3.178e-09	-1.254e-09	2.36
Bst_e33	2	EMSSOSP	-4.000000e+02	-4.000000e+02	1.6e-10	-7.255e-09	-2.812e-09	0.29
Bst_e33	2	EEM	-4.000000e+02	-4.000000e+02	1.6e-10	-7.255e-09	-2.812e-09	0.29
st_e01	3	Orig.	-6.666667e+00	-6.666667e+00	1.6e-11	-3.183e-09	-7.957e-10	0.06
st_e01	3	EMSSOSP	-6.666667e+00	-6.666667e+00	4.5e-12	-2.173e-10	-5.433e-11	0.05
st_e01	3	EEM	-6.666667e+00	-6.666667e+00	5.3e-10	-8.596e-09	-2.149e-09	0.04
st_e09	3	Orig.	-5.000000e-01	-5.000000e-01	2.8e-10	0.000e+00	0.000e+00	0.05
st_e09	3	EMSSOSP	-5.000000e-01	-5.000000e-01	8.8e-10	0.000e+00	0.000e+00	0.05
st_e09	3	EEM	-5.000000e-01	-5.000000e-01	9.3e-10	0.000e+00	0.000e+00	0.04
st_e34	2	Orig.	1.561953e-02	1.561953e-02	2.3e-10	0.000e+00	0.000e+00	0.32
st_e34	2	EMSSOSP	1.561953e-02	1.561953e-02	6.6e-11	0.000e+00	0.000e+00	0.32
st_e34	2	EEM	1.561952e-02	1.561952e-02	6.3e-12	-1.968e-06	-3.960e-07	0.16

Table 4: Numerical errors, iterations, DIMACS errors and sizes of Lasserre's SDP relaxation problems. We omit err2 and err3 from this table because they are zero for all POPs and methods.

Problem	r	Method	n.e.	iter.	err1	err4	err5	err6	sizeA	nnzA	#LP	#SDP	a.SDP	m.SDP
Bex3_1_1	3	Orig.	0	20	2.4e-09	6.0e-12	-1.2e-11	5.8e-09	3002 × 71775	126436	0	23	50.22	165
Bex3_1_1	3	EMSSOSP	0	20	2.6e-09	6.4e-12	-1.3e-11	5.1e-09	2171 × 51439	104840	0	23	46.65	83
Bex3_1_1	3	EEM	0	21	6.5e-10	1.6e-12	-2.9e-12	7.0e-10	1286 × 40743	72076	0	23	40.30	45
Bex3_1_4	4	Orig.	0	18	1.5e-10	1.8e-11	-1.8e-10	1.4e-10	164 × 4825	11618	0	10	21.50	35
Bex3_1_4	4	EMSSOSP	0	18	2.0e-10	1.7e-11	-1.2e-10	3.3e-10	164 × 4825	11618	0	10	21.50	35
Bex3_1_4	4	EEM	0	18	6.7e-11	6.1e-11	-1.5e-09	-1.4e-09	119 × 3700	7793	0	10	19.00	20
Bex5_2_2_case1	2	Orig.	0	21	2.7e-10	2.5e-11	-2.7e-08	-1.8e-08	714 × 5245	7085	0	21	12.14	55
Bex5_2_2_case1	2	EMSSOSP	0	19	6.7e-10	2.0e-11	-1.6e-08	-1.5e-08	300 × 2364	4204	0	21	10.10	12
Bex5_2_2_case1	2	EEM	0	19	6.7e-10	2.0e-11	-1.6e-08	-1.5e-08	300 × 2364	4204	0	21	10.10	12
Bex5_2_2_case2	2	Orig.	0	21	4.3e-11	4.7e-12	-8.2e-09	-6.4e-09	714 × 5245	7085	0	21	12.14	55
Bex5_2_2_case2	2	EMSSOSP	0	20	2.4e-11	8.8e-12	-1.7e-08	-1.7e-08	300 × 2364	4204	0	21	10.10	12
Bex5_2_2_case2	2	EEM	0	20	2.4e-11	8.8e-12	-1.7e-08	-1.7e-08	300 × 2364	4204	0	21	10.10	12
Bex5_2_2_case3	2	Orig.	0	18	1.6e-10	1.4e-11	-1.4e-09	5.8e-10	714 × 5245	7085	0	21	12.14	55
Bex5_2_2_case3	2	EMSSOSP	0	16	6.1e-11	8.2e-12	-1.7e-09	-1.6e-09	300 × 2364	4204	0	21	10.10	12
Bex5_2_2_case3	2	EEM	0	16	6.1e-11	8.2e-12	-1.7e-09	-1.6e-09	300 × 2364	4204	0	21	10.10	12
Bst_e07	2	Orig.	0	17	1.1e-10	2.4e-13	-1.8e-12	4.1e-10	1000 × 7348	10131	0	23	13.39	66
Bst_e07	2	EMSSOSP	0	16	2.4e-10	7.6e-13	-3.3e-12	5.4e-10	430 × 3188	5971	0	23	11.13	14
Bst_e07	2	EEM	0	17	5.5e-11	1.7e-13	-7.4e-13	1.1e-10	385 × 3044	5476	0	23	10.70	13
Bst_e33	2	Orig.	0	14	2.2e-10	6.5e-13	-6.1e-12	1.0e-09	714 × 5245	7395	0	21	12.14	55
Bst_e33	2	EMSSOSP	0	12	4.5e-10	1.4e-12	-1.1e-11	1.2e-09	300 × 2364	4514	0	21	10.10	12
Bst_e33	2	EEM	0	12	4.5e-10	1.4e-12	-1.1e-11	1.2e-09	300 × 2364	4514	0	21	10.10	12
st_e01	3	Orig.	0	11	8.6e-11	2.4e-12	-5.5e-12	1.3e-10	27 × 280	384	0	6	6.67	10
st_e01	3	EMSSOSP	0	11	3.3e-11	5.0e-13	-1.6e-12	3.8e-11	25 × 244	348	0	6	6.33	8
st_e01	3	EEM	0	9	2.6e-10	8.0e-12	1.8e-10	3.8e-10	20 × 189	266	0	6	5.50	6
st_e09	3	Orig.	0	11	2.2e-10	5.9e-12	1.6e-10	3.5e-10	27 × 280	456	0	6	6.67	10
st_e09	3	EMSSOSP	0	10	3.0e-10	2.0e-11	4.1e-10	6.0e-10	25 × 244	420	0	6	6.33	8
st_e09	3	EEM	0	9	3.0e-10	1.9e-11	4.0e-10	4.8e-10	20 × 189	284	0	6	5.50	6
st_e34	2	Orig.	0	22	9.6e-11	5.9e-14	5.4e-11	1.1e-10	209 × 1568	3272	0	17	8.24	28
st_e34	2	EMSSOSP	0	27	3.7e-12	1.0e-13	1.5e-11	2.8e-11	158 × 953	2639	0	17	7.35	13
st_e34	2	EEM	0	22	2.3e-14	2.8e-14	1.4e-12	1.5e-12	83 × 641	953	4	13	7.00	7

Table 5: Numerical results of the WKKM sparse SDP relaxation problems

Problem	r	Method	SDPobj	POPobj	ϵ_{obj}	ϵ_{feas}	ϵ'_{feas}	eTime
Bex3_1_1	3	Orig.	7.049248e+03	7.049248e+03	1.8e-11	-3.137e-04	-1.552e-10	0.87
Bex3_1_1	3	EMSSOSP	7.049248e+03	7.049248e+03	2.1e-11	-3.819e-04	-1.889e-10	0.59
Bex3_1_1	3	EEM	7.049248e+03	7.049248e+03	2.4e-11	-4.335e-04	-2.144e-10	0.30
Bex5_2_2_case1	2	Orig.	-4.778484e+02	-4.779311e+02	1.7e-04	-5.678e+02	-9.157e-01	0.71
Bex5_2_2_case1	2	EMSSOSP	-4.778414e+02	-4.778925e+02	1.1e-04	-5.678e+02	-9.156e-01	0.37
Bex5_2_2_case1	2	EEM	-4.778472e+02	-4.779112e+02	1.3e-04	-5.676e+02	-9.156e-01	0.35
Bex5_2_2_case1	3	Orig.	-4.086439e+02	-4.100943e+02	3.5e-03	-3.737e+02	-8.124e-01	5.41
Bex5_2_2_case1	3	EMSSOSP	-4.002579e+02	-4.002719e+02	3.5e-05	-3.535e+01	-2.465e-01	3.11
Bex5_2_2_case1	3	EEM	-4.000032e+02	-4.000075e+02	1.1e-05	-6.845e-01	-6.340e-03	2.02
Bex5_2_2_case2	2	Orig.	-8.499660e+02	-8.499680e+02	2.3e-06	-8.569e+02	-7.360e-01	1.05
Bex5_2_2_case2	2	EMSSOSP	-8.498378e+02	-8.498394e+02	1.9e-06	-8.564e+02	-7.359e-01	0.37
Bex5_2_2_case2	2	EEM	-8.498379e+02	-8.498395e+02	1.9e-06	-8.564e+02	-7.359e-01	0.36
Bex5_2_2_case2	3	Orig.	-6.456600e+02	-6.841458e+02	6.0e-02	-2.015e+03	-8.918e-01	5.67
Bex5_2_2_case2	3	EMSSOSP	-6.016427e+02	-6.021145e+02	7.8e-04	-4.235e+02	-7.415e-01	3.21
Bex5_2_2_case2	3	EEM	-6.001027e+02	-6.001363e+02	5.6e-05	-2.073e+01	-5.488e-01	3.40
Bex5_2_2_case3	2	Orig.	-7.695283e+02	-7.695349e+02	8.6e-06	-6.731e+01	-2.204e-01	0.75
Bex5_2_2_case3	2	EMSSOSP	-7.695063e+02	-7.695068e+02	6.8e-07	-6.696e+01	-2.194e-01	0.45
Bex5_2_2_case3	2	EEM	-7.695067e+02	-7.695072e+02	6.9e-07	-6.688e+01	-2.192e-01	0.40
Bex5_2_2_case3	3	Orig.	-7.503488e+02	-7.504064e+02	7.7e-05	-1.715e+01	-5.430e-02	6.24
Bex5_2_2_case3	3	EMSSOSP	-7.500063e+02	-7.500092e+02	3.8e-06	-1.670e+00	-5.539e-03	3.04
Bex5_2_2_case3	3	EEM	-7.500001e+02	-7.500001e+02	3.7e-09	-1.204e-02	-4.013e-05	2.20
Bst_e07	2	Orig.	-1.809184e+03	-1.809184e+03	9.6e-12	-1.409e-05	-2.302e-08	0.50
Bst_e07	2	EMSSOSP	-1.809184e+03	-1.809184e+03	5.6e-10	-7.653e-06	-1.250e-08	0.19
Bst_e07	2	EEM	-1.809184e+03	-1.809184e+03	5.8e-10	-1.079e-05	-1.762e-08	0.17
Bst_e33	2	Orig.	-4.000000e+02	-4.000000e+02	1.1e-09	-4.381e-09	-2.190e-09	0.30
Bst_e33	2	EMSSOSP	-4.000000e+02	-4.000000e+02	5.7e-10	-1.026e-09	-5.132e-10	0.13
Bst_e33	2	EEM	-4.000000e+02	-4.000000e+02	8.6e-10	-1.257e-09	-6.285e-10	0.11

Table 6: Numerical errors, iterations, DIMACS errors and sizes of the WKKM sparse SDP relaxation problems. We omit err2 and err3 from this table because they are zero for all POPs and methods.

Problem	r	Method	n.e.	iter.	err1	err4	err5	err6	sizeA	nnzA	#LP	#SDP	a.SDP	m.SDP
Bex3_1_1	3	Orig.	0	16	8.3e-10	2.8e-12	-8.6e-12	1.5e-09	363 × 4975	8784	0	25	13.00	35
Bex3_1_1	3	EMSSOSP	0	16	1.0e-09	3.4e-12	-1.0e-11	1.5e-09	295 × 3828	7589	0	25	12.00	22
Bex3_1_1	3	EEM	0	16	1.2e-09	3.8e-12	-1.1e-11	1.0e-09	225 × 3007	5344	0	25	10.52	15
Bex5_2_2_case1	2	Orig.	0	40	4.2e-11	4.4e-11	-1.4e-05	-1.3e-05	235 × 1507	2028	0	23	6.74	21
Bex5_2_2_case1	2	EMSSOSP	0	38	3.9e-11	2.9e-11	-8.9e-06	-8.8e-06	137 × 752	1264	0	23	5.39	8
Bex5_2_2_case1	2	EEM	0	38	3.6e-11	3.6e-11	-1.1e-05	-1.1e-05	123 × 720	1168	0	23	5.22	8
Bex5_2_2_case1	3	Orig.	1	27	3.4e-07	3.0e-08	-2.6e-04	7.0e-04	825 × 11342	15723	0	23	19.57	56
Bex5_2_2_case1	3	EMSSOSP	1	33	7.1e-09	1.8e-09	-2.5e-06	2.2e-05	603 × 7378	11727	0	23	17.04	30
Bex5_2_2_case1	3	EEM	0	34	4.0e-11	3.8e-11	-7.7e-07	-7.5e-07	543 × 6912	10617	0	23	16.26	30
Bex5_2_2_case2	2	Orig.	0	39	2.4e-10	4.6e-11	-2.1e-07	9.4e-06	235 × 1507	2028	0	23	6.74	21
Bex5_2_2_case2	2	EMSSOSP	0	37	5.2e-11	2.5e-11	-1.7e-07	5.1e-07	137 × 752	1264	0	23	5.39	8
Bex5_2_2_case2	2	EEM	0	37	5.1e-11	2.5e-11	-1.7e-07	-1.7e-07	123 × 720	1168	0	23	5.22	8
Bex5_2_2_case2	3	Orig.	1	30	2.6e-07	2.9e-08	-4.1e-03	-3.2e-03	825 × 11342	15723	0	23	19.57	56
Bex5_2_2_case2	3	EMSSOSP	1	36	1.8e-08	1.9e-09	-5.1e-05	3.2e-05	603 × 7378	11727	0	23	17.04	30
Bex5_2_2_case2	3	EEM	1	39	9.2e-10	9.4e-11	-3.6e-06	-2.7e-06	543 × 6912	10617	0	23	16.26	30
Bex5_2_2_case3	2	Orig.	0	41	1.3e-10	4.2e-11	-1.2e-06	1.3e-06	235 × 1507	2028	0	23	6.74	21
Bex5_2_2_case3	2	EMSSOSP	0	38	9.8e-11	9.4e-12	-9.6e-08	-1.7e-08	137 × 752	1264	0	23	5.39	8
Bex5_2_2_case3	2	EEM	0	36	9.6e-11	1.0e-11	-9.7e-08	-9.7e-08	123 × 720	1168	0	23	5.22	8
Bex5_2_2_case3	3	Orig.	1	30	5.2e-08	8.3e-09	-1.1e-05	2.6e-05	825 × 11342	15723	0	23	19.57	56
Bex5_2_2_case3	3	EMSSOSP	1	34	2.0e-09	4.6e-10	-5.3e-07	4.0e-07	603 × 7378	11727	0	23	17.04	30
Bex5_2_2_case3	3	EEM	0	35	2.9e-11	3.3e-12	-5.1e-10	7.4e-10	543 × 6912	10617	0	23	16.26	30
Bst_e07	2	Orig.	0	20	1.5e-10	2.9e-13	-2.1e-12	5.2e-10	296 × 2209	2888	0	28	7.61	21
Bst_e07	2	EMSSOSP	0	16	3.5e-11	4.0e-13	1.2e-10	1.8e-10	171 × 1037	1707	0	28	5.75	9
Bst_e07	2	EEM	0	16	4.6e-11	5.2e-13	1.3e-10	1.8e-10	154 × 954	1504	1	27	5.59	8
Bst_e33	2	Orig.	0	21	7.7e-11	3.2e-13	7.7e-11	3.9e-10	235 × 1507	2120	0	23	6.74	21
Bst_e33	2	EMSSOSP	0	15	2.7e-11	7.4e-14	4.1e-11	9.4e-11	137 × 752	1356	0	23	5.39	8
Bst_e33	2	EEM	0	15	1.5e-11	1.0e-13	6.1e-11	8.8e-11	123 × 720	1228	0	23	5.22	8

Table 7: Numerical results of the WKKM sparse SDP relaxation problems with relaxation order $r = 2$ for POP (4.4)

$2n$	Method	SDPobj	POPobj	ϵ_{obj}	ϵ_{feas}	ϵ'_{feas}	eTime
24	Orig.	-6.539703e-01	-1.9839457e-01	4.6e-01	-4.540e-01	-2.270e-01	14.67
24	EMSSOSP	-6.539851e-01	-1.9842222e-01	4.6e-01	-4.540e-01	-2.270e-01	6.91
24	EEM	-6.540020e-01	-1.9846984e-01	4.6e-01	-4.539e-01	-2.270e-01	1.13
44	Orig.	-4.809124e-01	-4.0113645e-01	8.0e-02	-2.124e-01	-1.062e-01	35.85
44	EMSSOSP	-4.809140e-01	-4.0113753e-01	8.0e-02	-2.124e-01	-1.062e-01	8.63
44	EEM	-4.809147e-01	-4.0113958e-01	8.0e-02	-2.124e-01	-1.062e-01	1.34
64	Orig.	-7.670512e-02	-1.6406578e-02	6.0e-02	-4.161e-01	-2.080e-01	51.36
64	EMSSOSP	-7.670623e-02	-1.6412691e-02	6.0e-02	-4.092e-01	-2.046e-01	18.43
64	EEM	-7.670639e-02	-1.6427020e-02	6.0e-02	-3.884e-01	-1.942e-01	1.76
84	Orig.	-1.923065e-01	-9.2640748e-02	1.0e-01	-6.817e-01	-3.409e-01	103.33
84	EMSSOSP	-1.923129e-01	-9.2667806e-02	1.0e-01	-6.824e-01	-3.412e-01	27.46
84	EEM	-1.923141e-01	-9.2675663e-02	1.0e-01	-6.827e-01	-3.414e-01	2.30
104	Orig.	-3.090968e-01	-4.7102083e-02	2.6e-01	-2.992e-01	-1.496e-01	126.64
104	EMSSOSP	-3.090987e-01	-4.7103057e-02	2.6e-01	-3.550e-01	-1.759e-01	8.76
104	EEM	-3.090987e-01	-4.7103023e-02	2.6e-01	-3.979e-01	-1.971e-01	4.95

4.2. Numerical results for randomly generated POP with a special structure

Let $C_i = \{i, i+1, i+2\}$ for all $i = 1, \ldots, n-2$. x_{C_i} and y_{C_i} denote the subvectors (x_i, x_{i+1}, x_{i+2}) and (y_i, y_{i+1}, y_{i+2}) of $x, y \in \mathbb{R}^n$, respectively. We consider the following POP:

$$\left\{ \begin{array}{ll} \inf_{x,y\in\mathbb{R}^n} & \displaystyle\sum_{i=1}^{n-2} (x_{C_i}, y_{C_i})^T P_i \begin{pmatrix} x_{C_i} \\ y_{C_i} \end{pmatrix} \\ \text{subject to} & x_{C_i}^T Q_i y_{C_i} + c_i^T x_{C_i} + d_i^T y_{C_i} + \gamma_i \geq 0 \ (i = 1, \ldots, n-2), \\ & 0 \leq x_i, y_i \leq 1 \ (i = 1, \ldots, n), \end{array} \right. \tag{4.4}$$

where $c_i, d_i \in \mathbb{R}^3$, $Q_i \in \mathbb{R}^{3\times 3}$, and $P_i \in \mathbb{S}^{6\times 6}$ is a symmetric positive semidefinite matrix. This POP has $2n$ variables and $5n - 2$ polynomial inequalities.

In this subsection, we generate POP (4.4) randomly and apply the WKKM sparse SDP relaxation with relaxation order $r = 2, 3$. The SDP relaxation problems obtained by Lasserre's SDP relaxation are too large-scale to be handled for these problems.

Tables 7 and 8 show the numerical results of the WKKM sparse SDP relaxation with relaxation order $r = 2$. To obtain more accurate values and solutions, we use SDPA-GMP [5]. Tables 11 and 12 show the numerical result by SDPA-GMP with tolerance ϵ = 1.0e-15 and precision 256. With this precision, SDPA-GMP calculate floating point numbers with approximately 77 significant digits. Tables 9 and 10 show the numerical results of the WKKM sparse SDP relaxation with relaxation order $r = 3$. Tables 13 and 14 show the numerical result by SDPA-GMP with the same tolerance and precision as above. In this case, we solve only $2n = 24$ and 44 because otherwise the resulting SDP problems become too large-scale to be solved by SDPA-GMP.

Table 8: Numerical errors, iterations, DIMACS errors and sizes of the WKKM sparse SDP relaxation problems with relaxation order $r = 2$ for POP (4.4). We omit err2 and err3 from this table because they are zero for all POPs and methods.

$2n$	Method	n.e.	iter.	err1	err4	err5	err6	sizeA	nnzA	#LP	#SDP	a.SDP	m.SDP
24	Orig.	1	49	2.1e-09	0.0e+00	-1.9e-08	1.6e-05	2015 × 16192	26728	0	69	11.88	36
24	EMSSOSP	1	51	2.4e-09	0.0e+00	-1.0e-08	6.9e-06	1101 × 6572	15050	0	69	9.16	19
24	EEM	0	20	1.4e-10	6.6e-13	5.6e-10	1.2e-09	593 × 3531	5067	10	59	7.71	8
44	Orig.	0	59	2.7e-10	0.0e+00	-8.8e-08	1.4e-06	4055 × 32352	53728	0	129	12.25	36
44	EMSSOSP	0	55	1.7e-10	0.0e+00	-3.5e-10	3.7e-07	2201 × 12662	29745	0	129	9.32	19
44	EEM	0	21	9.3e-11	7.8e-13	7.9e-10	1.4e-09	1233 × 6741	9667	20	109	7.84	8
64	Orig.	1	58	5.5e-09	0.0e+00	-3.8e-08	1.6e-06	6095 × 48512	80728	0	189	12.38	36
64	EMSSOSP	1	53	2.7e-09	0.0e+00	-9.0e-11	1.9e-07	3301 × 18752	44457	0	189	9.38	19
64	EEM	0	27	9.1e-12	6.1e-15	4.2e-11	9.9e-11	1873 × 9951	14267	30	159	7.89	8
84	Orig.	1	64	8.1e-09	0.0e+00	-2.8e-09	9.5e-06	7940 × 62821	105067	0	249	12.27	36
84	EMSSOSP	1	63	6.9e-10	0.0e+00	-1.5e-07	8.6e-07	4332 × 24428	58239	0	249	9.31	19
84	EEM	0	20	3.7e-11	5.7e-14	1.0e-10	2.5e-10	2465 × 12846	18417	40	209	7.82	8
104	Orig.	0	88	3.7e-10	0.0e+00	-1.7e-07	1.6e-06	9916 × 78394	131225	0	309	12.30	36
104	EMSSOSP	0	25	6.1e-11	9.8e-14	5.6e-10	1.2e-09	5410 × 30410	72723	0	309	9.33	19
104	EEM	0	22	1.8e-10	2.2e-15	1.3e-10	9.7e-10	3090 × 15981	22912	50	259	7.83	8

Table 9: Numerical results of the WKKM sparse SDP relaxation problems with relaxation order $r = 3$ for POP (4.4)

$2n$	Method	SDPobj	POPobj	ϵ_{obj}	ϵ_{feas}	ϵ'_{feas}	eTime
24	Orig.	-5.160104e-01	-5.1400753e-01	2.0e-03	0.000e+00	0.000e+00	475.04
24	EMSSOSP	-5.160105e-01	-5.1400132e-01	2.0e-03	0.000e+00	0.000e+00	190.01
24	EEM	-5.160105e-01	-5.1401615e-01	2.0e-03	0.000e+00	0.000e+00	35.51
44	Orig.	-4.409558e-01	-4.2159001e-01	1.9e-02	-5.408e-02	-2.704e-02	1650.50
44	EMSSOSP	-4.409530e-01	-4.2158583e-01	1.9e-02	-5.704e-02	-2.852e-02	338.54
44	EEM	-4.409561e-01	-4.2158866e-01	1.9e-02	-5.894e-02	-2.947e-02	98.19
64	Orig.	-6.235235e-02	-6.2352342e-02	6.3e-09	-1.208e-01	-6.039e-02	1316.66
64	EMSSOSP	-6.235225e-02	-6.2352050e-02	2.0e-07	-1.265e-01	-6.327e-02	551.48
64	EEM	-6.235235e-02	-6.2352349e-02	1.4e-09	-1.219e-01	-6.095e-02	114.98
84	Orig.	-1.344148e-01	-1.0153127e-01	3.3e-02	-8.336e-02	-4.168e-02	2321.02
84	EMSSOSP	-1.344157e-01	-1.0148873e-01	3.3e-02	-8.324e-02	-4.162e-02	962.95
84	EEM	-1.344151e-01	-1.0153038e-01	3.3e-02	-8.340e-02	-4.170e-02	256.85
104	Orig.	-1.579301e-01	-8.8999964e-02	6.9e-02	-2.032e-01	-1.016e-01	2956.81
104	EMSSOSP	-1.578784e-01	-8.8947707e-02	6.9e-02	-2.101e-01	-1.050e-01	744.44
104	EEM	-1.579325e-01	-8.8995970e-02	6.9e-02	-2.171e-01	-1.086e-01	279.25

Table 10: Numerical errors, iterations, DIMACS errors and sizes of the sparse SDP relaxation problems with relaxation order $r = 3$ for POP (4.4). We omit err2 and err3 from this table because they are zero for all POPs and methods.

$2n$	Method	n.e.	iter.	err1	err4	err5	err6	sizeA	nnzA	#LP	#SDP	a.SDP	m.SDP
24	Orig.	1	39	2.7e-09	3.3e-12	-3.3e-11	1.0e-07	11995 × 203344	409859	0	69	45.97	120
24	EMSSOSP	1	33	8.1e-09	1.1e-11	-6.6e-11	1.6e-07	9283 × 131628	325901	0	69	40.52	86
24	EEM	1	31	1.5e-09	2.0e-12	-1.1e-11	1.7e-08	4770 × 68370	104995	0	69	29.94	36
44	Orig.	1	54	1.3e-08	7.0e-12	-8.1e-11	4.8e-07	24535 × 412144	838939	0	129	47.84	120
44	EMSSOSP	1	29	3.5e-07	2.0e-10	-1.4e-09	6.6e-06	18940 × 262038	662594	0	129	41.91	86
44	EEM	1	35	6.4e-09	3.5e-12	-2.1e-11	6.8e-08	10170 × 133810	205915	0	129	30.59	36
64	Orig.	1	36	9.5e-10	1.2e-13	-1.4e-12	1.4e-08	37075 × 620944	1268019	0	189	48.53	120
64	EMSSOSP	1	31	3.8e-08	7.1e-12	-5.1e-11	4.2e-07	28594 × 392448	999618	0	189	42.41	86
64	EEM	0	33	2.5e-10	4.8e-14	-3.1e-13	2.0e-09	15570 × 199250	306835	0	189	30.83	36
84	Orig.	1	46	3.8e-08	5.0e-12	-5.8e-09	1.1e-06	47964 × 796960	1636667	0	249	47.78	120
84	EMSSOSP	1	42	7.1e-07	1.0e-10	1.1e-06	8.9e-06	37121 × 504711	1293315	0	249	41.80	86
84	EEM	1	38	1.6e-09	2.8e-12	-4.0e-12	1.2e-08	20327 × 253893	391675	0	249	30.26	36
104	Orig.	1	44	1.2e-07	6.3e-12	-2.9e-10	4.5e-06	59964 × 995856	2047139	0	309	47.96	120
104	EMSSOSP	1	26	1.6e-05	8.0e-10	-1.7e-08	2.5e-04	46408 × 630096	1617625	0	309	41.94	86
104	EEM	1	36	1.8e-09	8.7e-14	-1.7e-12	1.6e-08	25523 × 316758	488771	0	309	30.34	36

Table 11: Numerical results by SDPA-GMP 7.1.2 with ϵ=1.0e-15 for the WKKM sparse SDP relaxation problems with relaxation order $r = 2$ in POP (4.4)

$2n$	Method	p.v.	iter.	eTime	SDPobj by SDPA-GMP	SDPobj by SeDuMi
24	Orig.	pdOPT	53	719.260	-6.54001981e-01	-6.539703e-01
24	EMSSOSP	pdOPT	49	115.920	-6.54001981e-01	-6.539851e-01
24	EEM	pdOPT	44	17.370	-6.54001981e-01	-6.540020e-01
44	Orig.	pdOPT	57	1655.640	-4.80914669e-01	-4.809124e-01
44	EMSSOSP	pdOPT	50	249.480	-4.80914669e-01	-4.809140e-01
44	EEM	pdOPT	43	31.350	-4.80914669e-01	-4.809147e-01
64	Orig.	pdOPT	58	2599.260	-7.67063944e-02	-7.670512e-02
64	EMSSOSP	pdOPT	52	385.260	-7.67063944e-02	-7.670623e-02
64	EEM	pdOPT	48	53.090	-7.67063944e-02	-7.670639e-02
84	Orig.	pdOPT	53	3066.100	-1.92314133e-01	-1.923065e-01
84	EMSSOSP	pdOPT	48	453.890	-1.92314133e-01	-1.923129e-01
84	EEM	pdOPT	41	60.660	-1.92314133e-01	-1.923141e-01
104	Orig.	pdOPT	59	4341.780	-3.09098714e-01	-3.090968e-01
104	EMSSOSP	pdOPT	50	625.520	-3.09098714e-01	-3.090987e-01
104	EEM	pdOPT	46	80.070	-3.09098714e-01	-3.090987e-01

Table 12: DIMACS errors by SDPA-GMP 7.1.2 with ϵ=1.0e-15 for the sparse SDP relaxation problems with relaxation order $r = 2$ in POP (4.4). We omit err2 and err4 from this table because they are zero for all POPs and methods.

$2n$	Method	err1	err3	err5	err6
24	Orig.	2.987e-32	7.456e-74	2.595e-16	3.272e-16
24	EMSSOSP	2.013e-33	3.365e-74	3.473e-16	3.729e-16
24	EEM	1.682e-68	3.950e-77	3.877e-16	3.877e-16
44	Orig.	1.852e-31	1.059e-73	3.274e-16	4.486e-16
44	EMSSOSP	1.549e-32	7.730e-74	3.861e-16	4.448e-16
44	EEM	2.012e-68	2.908e-77	7.564e-17	7.564e-17
64	Orig.	4.578e-31	8.935e-74	6.357e-16	8.141e-16
64	EMSSOSP	7.600e-33	6.223e-74	6.360e-16	6.889e-16
64	EEM	2.402e-76	4.331e-77	3.539e-16	3.539e-16
84	Orig.	6.983e-32	2.500e-73	6.106e-16	9.097e-16
84	EMSSOSP	1.859e-34	9.731e-61	2.978e-16	3.202e-16
84	EEM	9.632e-69	4.153e-77	4.269e-16	4.269e-16
104	Orig.	3.615e-31	1.695e-73	4.399e-16	7.407e-16
104	EMSSOSP	8.722e-34	1.041e-73	1.602e-16	1.698e-16
104	EEM	1.798e-68	2.114e-77	1.703e-16	1.703e-16

We observe the following.

- The sizes of the resulting SDP relaxation problems by EEM is again the smallest in the three methods. In particular, when we apply EEM and the WKKM sparse SDP relaxation with relaxation order $r = 2$, positive semidefinite constraints corresponding to the quadratic constraints in (4.2) are replaced by linear constraints in SDP relaxation problems. EEM removes all monomials except for the constant term in $\sigma_j \in \Sigma_1$ because those monomials are redundant for all SOS representations of f. Then $\sigma_j \in \Sigma_1$ for all $j = 1, \ldots, n$ can be replaced by $\sigma_j \in \Sigma_0$ for all $j = 1, \ldots, n$. This is equivalent to $\sigma_j \geq 0$ for all $j = 1, \ldots, n$. Therefore, we obtain n linear constraints in the resulting SDP relaxation problems.

Table 13: Numerical results by SDPA-GMP 7.1.2 with ϵ=1.0e-15 for the WKKM sparse SDP relaxation problems with relaxation order $r = 3$ in POP (4.4)

$2n$	Method	p.v.	iter.	eTime	SDPobj by SDPA-GMP	SDPobj by SeDuMi
24	Orig.	pdOPT	63	108657.790	-5.16010521e-01	-5.160104e-01
24	EMSSOSP	pdOPT	62	50558.410	-5.16010521e-01	-5.160105e-01
24	EEM	pdOPT	60	5815.020	-5.16010521e-01	-5.160105e-01
44	Orig.	pdOPT	71	268508.010	-4.40956190e-01	-4.409558e-01
44	EMSSOSP	pdOPT	70	125076.640	-4.40956190e-01	-4.409530e-01
44	EEM	pdOPT	67	13884.150	-4.40956190e-01	-4.409561e-01

Table 14: DIMACS errors by SDPA-GMP 7.1.2 with ϵ=1.0e-15 for the sparse SDP relaxation problems with relaxation order $r = 3$ in POP (4.4). We omit err2 and err4 from this table because they are zero for all POPs and methods.

$2n$	Method	err1	err3	err5	err6
24	Orig.	3.769e-30	1.713e-73	4.862e-16	5.763e-16
24	EMSSOSP	3.433e-30	2.059e-58	2.822e-16	3.196e-16
24	EEM	1.229e-65	3.354e-76	2.458e-16	2.458e-16
44	Orig.	6.368e-30	2.474e-73	3.905e-16	4.838e-16
44	EMSSOSP	5.010e-30	2.579e-73	4.820e-16	5.563e-16
44	EEM	4.523e-66	5.239e-76	3.112e-16	3.112e-16

- From Tables 11 and 12, SDPA-GMP with precision 256 can solve all SDP relaxation problems accurately. In particular, SDPA-GMP solves SDP relaxation problems obtained by EEM more than 8 and 50 times faster than EMSSOSP and Orig., respectively.
- From Tables 7, 8 and 11, SeDuMi returns the optimal values of SDP relaxation problems obtained by EEM almost exactly as accurately as SDPA-GMP and more than 15 times faster than SDPA-GMP, while SeDuMi terminates before we obtain accurate solutions of SDP relaxation problems obtained by the other methods.
- From Table 10, in SDP relaxation problems with relaxation order $r = 3$, DIMACS errors for SDP relaxation problems by EEM are the smallest in all methods. Moreover, from Tables 13 and 14, the optimal values of SDP relaxation problems by EEM coincide the optimal values found by SDPA-GMP for $2n = 24$ and 44. However, SeDuMi cannot obtain accurate solutions because these values are larger than the tolerance 1.0e-9 of SeDuMi.

5. An Application of EEM to Specific POPs

As we have seen in Section 3, we have a flexibility in choosing δ although EEM always returns the smallest set $G^* \in \Gamma(G^0, P)$. We focus on this flexibility and we prove the following two facts in this section: (i) if POP (4.1) satisfies a specific assumption, each optimal value of the SDP relaxation problem with relaxation order $r > \bar{r}$ is equal to that of the relaxation order $\bar{r}$. To prove this fact, we choose δ to be the largest element in $\bigcup_{j=0}^{m}(F_j + G_j + G_j) \setminus (F \cup \bigcup_{j=0}^{m} T_j)$ with the graded lexicographic order*. (ii) We give an extension of Proposition 4 in [7], where we choose δ to be the smallest element.

*We define the graded lexicographic order $\alpha \succeq \beta$ for $\alpha, \beta \in \mathbb{N}^n$ as follows: $|\alpha| > |\beta|$ or , $|\alpha| = |\beta|$ and $\alpha_i > \beta_i$ for the smallest index i with $\alpha_i \neq \beta_i$.

First of all, we state (i) exactly. Let γ_j be the largest element with the graded lexicographic order in F_j. For $\gamma \in \mathbb{N}^n$, we define $\mathcal{I}(\gamma) = \{k \in \{0, 1, \ldots, m\} \mid \gamma_k \equiv \gamma \pmod 2\}$. We impose the following assumption on polynomials in POP (4.1).

Assumption 5.1 *For any fixed $j \in \{0, 1, \ldots, m\}$, for each $k \in \mathcal{I}(\gamma_j)$, the largest monomial x^{γ_k} in f_k has the same sign as $(f_j)_{\gamma_j}$.*

The following theorem guarantees that we do not need to increase a relaxation order for POP which satisfies Assumption 5.1 in order to obtain a tighter lower bound.

Theorem 5.2 *Assume $r > \bar{r}$. Then under Assumption 5.1, the optimal value of the SDP relaxation problem with relaxation order r is the same as that of relaxation order $\bar{r}$.*

We postpone a proof of Theorem 5.2 till Appendix D.

We give two examples for Theorem 5.2.

Example 5.3 Let $f = x, f_1 = 1, f_2 = x, f_3 = x^2 - 1$. We consider the following POP:

$$\inf\{x \mid x \geq 0, x^2 - 1 \geq 0\}. \tag{5.1}$$

Then clearly, we have $\gamma_1 = 0, \gamma_2 = 1, \gamma_3 = 2$ and $\mathcal{I}(\gamma_1) = \mathcal{I}(\gamma_3) = \{1, 3\}$, $\mathcal{I}(\gamma_2) = \{2\}$, and this POP satisfies Assumption 5.1. Therefore, it follows from Theorem 5.2 that the optimal value of each SDP relaxation problem with relaxation order $r \geq 1$ is equal to the optimal value of the SDP relaxation problem with relaxation order 1. We give the SOS problem with relaxation order 1:

$$\sup_{\rho \in \mathbb{R}, \sigma_1 \in \Sigma_1, \sigma_2, \sigma_3 \geq 0} \{\rho \mid x - \rho = \sigma_1(x) + x\sigma_2 + (x^2 - 1)\sigma_3 \ (\forall x \in \mathbb{R})\}.$$

Furthermore, we can apply EEM to the identity to reduce the size of the SOS problem above. Then the obtained SOS problem is equivalent to LP as follows:

$$\sup_{\rho \in \mathbb{R}, \sigma_1, \sigma_2 \geq 0} \{\rho \mid x - \rho = \sigma_1 + x\sigma_2 \ (\forall x \in \mathbb{R})\} = \sup_{\rho \in \mathbb{R}, \sigma_1, \sigma_2 \geq 0} \{\rho \mid \sigma_1 = -\rho, \sigma_2 = 1\}.$$

Clearly, the optimal value of this LP is 0, and thus the optimal value of the SDP relaxation problem with arbitrary relaxation order is 0.

This POP is dealt with in [25] and it is shown by using positive semidefiniteness in SDP relaxation problems that the optimal values of all SDP relaxation problems are 0. In [22], it is shown that the approach is FRA and this fact is a motivation to show a relationship between EEM and FRA. We give the details in Section 6.

Example 5.4 Let $f = -x, f_1 = 1, f_2 = 2 - x, f_3 = x^2 - 1$. Then clearly, we have the same γ_j and $\mathcal{I}(\gamma_j)$ as in Example 5.3. This POP also satisfies Assumption 5.1. We solve SDP relaxation problem with relaxation order $r = 1$. Then we obtain the following SOS problem with relaxation order $r = 1$:

$$\sup_{\rho \in \mathbb{R}, \sigma_1 \in \Sigma_1, \sigma_2, \sigma_3 \geq 0} \{\rho \mid -x - \rho = \sigma_1(x) + (2 - x)\sigma_2 + (x^2 - 1)\sigma_3 \ (\forall x \in \mathbb{R})\}.$$

Applying EEM to the identity, we obtain the following LP problem:

$$\begin{aligned} &\sup_{\rho, \sigma_1, \sigma_2 \geq 0} \{\rho \,|{-x} - \rho = \sigma_1 + (2 - x)\sigma_2 \ (\forall x \in \mathbb{R})\} \\ = &\sup_{\rho, \sigma_1, \sigma_2 \geq 0} \{\rho \,|{-\rho} = \sigma_1 + 2\sigma_2, 1 = \sigma_2\} = -2. \end{aligned}$$

From this result, the optimal value of the SDP relaxation problem with arbitrary relaxation order is -2, which is equal to the optimal value of the POP.

Next, we show (ii). Consider an SOS representation of f with $f_0, f_1, \ldots, f_m$, *i.e.*, $f = f_0\sigma_0 + f_1\sigma_1 + \cdots + f_m\sigma_m$, where $\sigma_j \in \Sigma_{r_j}$ for $j = 0, 1, \ldots, m$. In particular, we have $r_0 = r$ because $f_0 = 1$. Let ϵ_j be the smallest element in the graded lexicographic order in F_j for $j = 0, 1, \ldots, m$. For $f, f_0, f_1, \ldots, f_m$, we impose the following condition:

Assumption 5.5 *1. f is a homogeneous polynomial with degree $2r$,*

2. for any fixed $j = 0, 1, \ldots, m$, for each $k \in \mathcal{I}(\epsilon_j)$, the smalleset monomial x^{ϵ_k} has the same sign as $(f_j)_{\epsilon_j}$,

3. $|\epsilon_j| < \deg(f_j)$ for all $j = 1, \ldots, m$.

We remark that $f_0 = 1$ is not contained in 3 of Assumption 5.5.

Theorem 5.6 *We assume that $f \in \sum_{j=0}^{m} f_j\Sigma_{r_j}$. Then under Assumption 5.5, $f \in f_0\Sigma_{r_0} = \Sigma_r$.*

We give a proof in Appendix D.

Theorem 5.6 is an extension of Proposition 4 in [7]. Indeed, in [7], the authors show that for a homogeneous polynomial f with degree $2r$, $f \in \Sigma_r + f_1\Sigma_{r-1}$ if and only if $f \in \Sigma$, where $f_1 = 1 - \sum_{i=1}^{n} x_i^2$. Clearly, f, f_0 and f_1 satisfy Assumption 5.5.

6. A Relationship between EEM and FRA

In this section, we establish a relationship between EEM and a facial reduction algorithm (FRA) proposed in [22]. In [22], the authors extended FRAs proposed in [2, 12, 16, 17] into conic optimization problems and derived a more practical FRA for SDP (6.1). It is called *FRA-SDP*. In [23], the authors mentioned that in the case where $m = 1$ and $f_1 = 1$, EMSSOSP can be interpreted as a partial application of FRA-SDP. In this section, we show that in more general case, EEM can be interpreted as a partial application of FRA-SDP. This implies that EEM may generate an SDP problem which has an interior feasible solution.

FRA-SDP works for the following SDP (6.1). By using FRA-SDP, we can generate another SDP which is equivalent to the original SDP and has an interior feasible point in the feasible region:

$$\inf_{X \in \mathbb{S}^n_+} \{C \bullet X \,|A_k \bullet X = b_k \ (k = 1, \ldots, p)\} \tag{6.1}$$

where $C, A_k \in \mathbb{S}^n$ and $b \in \mathbb{R}^p$.

We give the algorithm of FRA-SDP for SDP (6.1). See [22] for more details of this algorithm:

Algorithm 6.1 (FRA-SDP)

Step 1 *Set $i := 0$ and $\mathcal{F}_0 := \mathbb{S}^n_+$.*

Step 2 *Find a nonzero $(y, W) \in \mathbb{R}^p \times \mathcal{F}_i$ of the homogenized dual system (HDS)*

$$b^T y \geq 0, W = -\sum_{k=1}^{p} A_k y_k, W \in \mathbb{S}^n_+. \tag{6.2}$$

Step 3 *If there exists no such (y, W), then stop and return $\mathcal{F}_i$.*

Step 4 *If $b^T y > 0$, then stop; the problem is infeasible. Otherwise, go to Step 4-1.*

Step 4-1 *Decompose* $W = RR^T$ *and find an* $n \times n$ *nonsingular matrix* Z *such that* $Z = (L, R)$ *for a matrix* L.

Step 4-2 *Set* $n := n - \mathrm{rank}(W)$, $\mathcal{F}_{i+1} := \mathbb{S}^n_+$, $\tilde{C} := Z^{-1}CZ^{-T}$ *and* $\tilde{A}_k := Z^{-1}A_kZ^{-T}$ $(k = 1, \ldots, p)$.

Step 4-3 *Make the following smaller SDP:*

$$\inf_{X\in\mathbb{S}^n_+} \left\{ \tilde{C}^1 \bullet X \,\middle|\, \tilde{A}^1_k \bullet X = b_k \ (k = 1, \ldots, p) \right\}, \tag{6.3}$$

where

$$\tilde{C} = \begin{pmatrix} \tilde{C}^1 & \tilde{C}^2 \\ \tilde{C}^{2T} & \tilde{C}^3 \end{pmatrix} \text{ and } \tilde{A}_j = \begin{pmatrix} \tilde{A}^1_k & \tilde{A}^2_k \\ \tilde{A}^{2T}_k & \tilde{A}^3_k \end{pmatrix}.$$

Go to Step 5.

Step 5 *Set* $i := i + 1$, *and go back to Step 2.*

It is shown in [22] that (i) FRA-SDP terminates in finitely many iterations, (ii) the resulting SDP (6.3) has an interior feasible solution if the original SDP (6.1) is feasible, and (iii) any solution (y, W) in (6.2) satisfies $b^Ty = 0$ if SDP (6.1) is feasible. From (ii), by solving the resulting SDP instead of SDP (6.1), we can expect that the computational stability and efficiency of primal-dual interior-point methods for SDP (6.3) are improved.

We consider SDP (2.4) obtained from the problem (2.3). In this section, we add the zero objective function in SDP (2.4) and regard SDP (2.4) as the minimization problem. The SDP problem is as follows:

$$\inf_{V_j\in\mathbb{S}^{\#(G_j)}_+ \ (j=1,\ldots,m)} \left\{ 0 \,\middle|\, \sum_{j=1}^m E_{j,\alpha} \bullet V_j = f_\alpha \ (\alpha \in \bigcup_{j=1}^m (F_j + G_j + G_j)) \right\}, \tag{6.4}$$

where $E_{j,\alpha} \in \mathbb{S}^{\#(G_j)}$. When we apply Lemma 2.3 to (2.3) once, it can construct $H := G \setminus B(\delta)$ from G if there exists δ which satisfies (2.6) in Lemma 2.3 is found. The SDP obtained from SOS problem (2.3) with $H = (H_1, \ldots, H_m)$ is

$$\inf_{V_j\in\mathbb{S}^{\#(H_j)}_+ \ (j=1,\ldots,m)} \left\{ 0 \,\middle|\, \sum_{j=1}^m (E_{j,\alpha})_{H_j,H_j} \bullet V_j = f_\alpha \ (\alpha \in \bigcup_{j=1}^m (F_j + H_j + H_j)) \right\}, \tag{6.5}$$

where $(E_{j,\alpha})_{H_j,H_j}$ is the leading principal submatrix of $E_{j,\alpha}$ indexed by H_j for $j = 1, \ldots, m$.

The following theorem shows that FRA-SDP can generate SDP (6.5) from SDP (6.4). This implies that EEM is a partial application of FRA.

Theorem 6.2 *We assume that* f *has an SOS representation with* $f_1, \ldots, f_m$ *and* $G_1, \ldots, G_m$. *Let* δ, $J(\delta)$ *and* $B(\delta)$ *be as in Lemma 2.3. We define*

$$\begin{aligned}
y_\delta &= \begin{cases} 1 & \text{if } (f_j)_{\delta-2\alpha} < 0 \text{ for all } j \in J(\delta) \text{ and } \alpha \in B_j(\delta), \\ -1 & \text{if } (f_j)_{\delta-2\alpha} > 0 \text{ for all } j \in J(\delta) \text{ and } \alpha \in B_j(\delta), \end{cases} \\
y_\alpha &= 0 \text{ for all } \alpha \in \left(\bigcup_{j=1}^m (F_j + G_j + G_j) \right) \setminus \{\delta\} \text{ and} \\
(W_j)_{\beta,\gamma} &= \begin{cases} -(f_j)_{\delta-2\beta} y_\delta & \text{if } \beta = \gamma \in B_j(\delta), \\ 0 & o.w. \end{cases} \quad \text{for all } (\beta, \gamma) \in G_j \text{ and } j = 1, \ldots, m.
\end{aligned}$$

Then $(y, (W_1, \ldots, W_m))$ is a solution of (6.2) which is constructed from SDP (6.4). Moreover, SDP (6.5) is the same as SDP (6.3) obtained by $W = (W_1, \ldots, W_m)$.

Proof: We prove that (y, W) satisfies (6.2) obtained from SDP (2.4). We have $f^T y = f_\delta y_\delta = 0$ because of $y_\alpha = 0$ $(\alpha \in \bigcup_{j=1}^m (F_j + G_j + G_j) \setminus \{\delta\})$ and $\delta \notin F$. In addition, $W_j \in \mathbb{S}_+^{\#(G_j)}$ because W_j is diagonal and $-(f_j)_{\delta-2\alpha} y_\delta > 0$. We show the equality $W_j = -\sum_{\alpha \in F_j+G_j+G_j} E_{j,\alpha} y_\alpha$. $S_j(y)$ denotes $-\sum_{\alpha \in F_j+G_j+G_j} E_{j,\alpha} y_\alpha$ for simplicity. If $j \notin J(\delta)$, then it is clear that $S_j(y) = O = W_j$ because $\delta \notin F_j + G_j + G_j$. We consider the case where $j \in J(\delta)$. From definitions of $E_{j,\alpha} \in \mathbb{S}^{\#(G_j)}$ and y, we have

$$(S_j(y))_{\beta_1,\beta_2} = (-E_{j,\delta})_{\beta_1,\beta_2} y_\delta = \begin{cases} -(f_j)_{\delta-\beta_1-\beta_2} y_\delta & \text{if } \delta - \beta_1 - \beta_2 \in F_j, \\ 0 & \text{o.w.} \end{cases}$$

for $\beta_1, \beta_2 \in G_j, j = 1, \ldots, m$. In the case where $\beta_1 \neq \beta_2$, $(S(y))_{\beta_1,\beta_2} = 0 = (W_j)_{\beta_1,\beta_2}$ because $\delta \notin T_j$. In the case where $\beta_1 = \beta_2$, it follows that

$$(S_j(y))_{\beta_1,\beta_1} = -(E_{j,\delta})_{\beta_1,\beta_1} y_\delta = \begin{cases} -(f_j)_{\delta-2\beta_1} y_\delta & \text{if } \beta_1 \in B_j(\delta), \\ 0 & \text{o.w.} \end{cases}$$

for $\beta_1 \in G_j$. This shows that $(S_j(y))_{\beta,\beta} = (W_j)_{\beta,\beta}$ for all $\beta \in G_j$. Therefore, we have $-\sum_{\alpha \in F_j+G_j+G_j} E_{j,\alpha} y_\alpha = S_j(y) = W_j$ for $j = 1, \ldots, m$.

We show the second statement. Let $H_j := G_j \setminus B_j(\delta)$ for $j = 1, \ldots, m$. From the definition of W_j, we define a nonsingular block diagonal matrix $Z = \text{diag}(Z_j; j = 1, \ldots, m)$ as follows:

$$Z_j = (L_j, R_j), \ R_j = \left(\sqrt{-(f_j)_{\delta-2\alpha} y_\delta} e_\alpha\right)_{\alpha \in B_j(\delta)} \text{ and } L_j = (e_\alpha)_{\alpha \in H_j},$$

where $e_\alpha \in \mathbb{R}^{\#(G_j)}$ is the α-th standard column vector. Then we have $W_j = R_j R_j^T$ and Z_j is nonsingular. In fact, we can give an explicit form of the inverse of Z_j as follows:

$$Z_j^{-1} = \begin{pmatrix} L_j^T \\ R_j'^T \end{pmatrix}, \ R_j' = \left(\frac{1}{\sqrt{-(f_j)_{\delta-2\alpha} y_\delta}} e_\alpha\right)_{\alpha \in B_j(\delta)}.$$

It is easy to verify the following:

$$\begin{aligned}
(E_{j,\alpha})_{H_j,H_j} &= L_j^T E_{j,\alpha} L_j \text{ for } \alpha \in \bigcup_{j=1}^m (F_j + H_j + H_j) \text{ and } j = 1, \ldots, m, \\
(E_{j,\alpha})_{H_j,H_j} &= O \text{ for } \alpha \in \bigcup_{j=1}^m (F_j + G_j + G_j) \setminus \left(\bigcup_{j=1}^m (F_j + H_j + H_j)\right) \text{ and } j = 1, \ldots, m.
\end{aligned}$$

Consequently, we obtain the following smaller SDP problem:

$$\inf_{V_j \in \mathbb{S}_+^{\#(H_j)} (j=1,\ldots,m)} \left\{ 0 \;\middle|\; \begin{array}{l} \sum_{j=1}^m (E_{j,\alpha})_{H_j,H_j} \bullet V_j = f_\alpha \ (\alpha \in \bigcup_{j=1}^m (F_j + G_j + G_j)), \\ \sum_{j=1}^m O \bullet V_j = f_\alpha \ (\alpha \in (\bigcup_{j=1}^m (F_j + G_j + G_j)) \setminus (\bigcup_{j=1}^m (F_j + H_j + H_j)) \end{array} \right\}. \tag{6.6}$$

This SDP is corresponding to SDP (6.3) in FRA-SDP. Here we use the following claim:

Claim 1 *We have* $F \subseteq \bigcup_{j=1}^m (F_j + G_j \setminus B_j(\delta) + G_j \setminus B_j(\delta)) \subseteq \bigcup_{j=1}^m (F_j + G_j + G_j)$.

Proof of Claim 1: We obtain the desired result from SOS representations (2.5) and (2.7). $\square$

It follows from Claim 1 that $f_\alpha = 0$ for all $\alpha \in \bigcup_{j=1}^m (F_j+G_j+G_j)\setminus(\bigcup_{j=1}^m (F_j+H_j+H_j))$ because such α is not contained in F. Therefore we can remove linear equalities on V_j for $\alpha \in \bigcup_{j=1}^m (F_j + G_j + G_j) \setminus (\bigcup_{j=1}^m (F_j + H_j + H_j))$ from SDP (6.6). Then the obtained SDP is equivalent to SDP (6.5). $\square$

We remark that in some cases, FRA can reduce the size of SDP (2.4) more than EEM. In the case where $m = 1$ and $f_1 = 1$, such an example is presented in [23].

7. Concluding Remarks

SDP relaxation problems obtained from POP often become large-scale and highly degenerate. To overcome these difficulties, in this paper, we extend EMSSOSP by Kojima *et al.* [8] into constrained POPs. EEM can reduce the sizes of the resulting SDP relaxation problems by using sparsity of $f, f_1, \ldots, f_m$. Moreover, EEM is a partial application of FRA and we can expect that the resulting SDP relaxation problems have an interior feasible solution and that computational efficiency of primal-dual interior-point methods is improved. We apply EEM to POPs in subsections 4.1 and 4.2 and observe that EEM is effective for those POPs. For SDP relaxation problems with relaxation order $r = 3$ for POPs in subsection 4.2, all DIMACS errors by EEM are smaller than the other methods although SeDuMi terminates before returning an accurate value and solution.

We cannot know whether SDP relaxation problems obtained by EEM have an interior feasible solution or not in advance although EEM is a partial application of FRA. If not, one can obtain such an SDP by applying FRA-SDP. However, we may encounter a numerical difficulty in FRA-SDP because FRA-SDP is comparable to solving the original SDP. We need to develop an algorithm for avoiding such a difficulty. This is one of our future works for SDP relaxation in POPs.

A. A Proof of Lemma 2.3

From (2.5), we obtain

$$f(x) = \sum_{j\in J(\delta)} f_j(x) \sum_{i=1}^{k_j} \left(\sum_{\alpha\in G_j} (g_{j,i}) x^\alpha \right)^2 + \sum_{j\in\{1,\ldots,m\}\setminus J(\delta)} f_j(x) \sum_{i=1}^{k_j} \left(\sum_{\alpha\in G_j} (g_{j,i}) x^\alpha \right)^2 .$$

From the definition of $J(\delta)$ and (2.6), the monomial x^δ does not appear in the second term. Indeed, if $j \notin J(\delta)$, then $\delta \notin F_j + G_j + G_j$ because $\delta \notin T_j$. Thus, we focus on the first term:

$$\begin{aligned} & \sum_{j\in J(\delta)} f_j(x) \sum_{i=1}^{k_j} \left(\sum_{\alpha\in G_j} (g_{j,i}) x^\alpha \right)^2 \\ = & \sum_{j\in J(\delta)} \sum_{i=1}^{k_j} \sum_{\epsilon\in F_j+G_j+G_j} \left(\sum_{\alpha\in F_j, \beta,\gamma\in G_j, \epsilon=\alpha+\beta+\gamma} (f_j)_\alpha (g_{j,i})_\beta (g_{j,i})_\gamma \right) x^\epsilon . \end{aligned}$$

Because the monomial x^δ does not appear in the polynomial f, we obtain the following equation:

$$0 = \sum_{\alpha \in F_j, \beta, \gamma \in G_j, \alpha+\beta+\gamma=\delta} (f_j)_\alpha (g_{j,i})_\beta (g_{j,i})_\gamma \quad \text{for all } i = 1, \ldots, k_j \text{ and } j \in J(\delta).$$

Moreover, it follows from $\delta \notin T_j$ and the definition of $B_j(\delta)$ that this equation is equivalent to the following equation:

$$0 = \sum_{\alpha \in F_j, \beta \in B_j(\delta), \alpha+2\beta=\delta} (f_j)_\alpha (g_{j,i})_\beta^2 \quad \text{for all } i = 1, \ldots, k_j, \text{ and } j \in J(\delta).$$

From (2.6), we obtain $(g_{j,i})_\alpha = 0$ for all $j \in J(\delta), i = 1, \ldots, k_j$ and $\alpha \in B_j(\delta)$. Because $B_j(\delta) = \emptyset$ for all $j \notin J(\delta)$, we obtain the desired result.

B. Some Results on the Set Theory

To show Theorem 3.3, we deal with a more general case of a function P and a set family $\Gamma(G^0, P)$ defined in Definition 3.2. We will prove Theorem 3.3 in Appendix C by using results obtained in this appendix.

Let X be a finite set. We consider a function $P : 2^X \times 2^X \to \{\text{true}, \text{false}\}$ and assume that $P(C, \emptyset) = \text{true}$ for all $C \subseteq X$.

Definition B.1 *For a given set $G^0 \in 2^X$ and the function P, we define a set family $\Gamma(G^0, P) \subseteq 2^X$ as follows: $G \in \Gamma(G^0, P)$ if and only if $G = G^0$ or there exists $G' \in \Gamma(G^0, P)$ such that $G \subsetneq G'$ and $P(G', G' \setminus G) = $ true.*

Remark B.2 1. If $G \in \Gamma(G^0, P)$ and there exists $A \subseteq G$ such that $P(G, A) = \text{true}$, then $G \setminus A \in \Gamma(G^0, P)$. Indeed, if A is empty, then clearly $G \setminus A \in \Gamma(G^0, P)$. Otherwise, H' and H denote G and $G \setminus A$, respectively. Then we have $H' \in \Gamma(G^0, P)$, $H \subsetneq H'$ and $P(H', H' \setminus H) = P(G, A) = \text{true}$. Therefore, $H = G \setminus A \in \Gamma(G^0, P)$.

2. It follows from Definition B.1 that for all $G \in \Gamma(G^0, P)$ except for G^0, there exist two sequences $\{G^p\}_{p=0}^q$ and $\{A^p\}_{p=0}^{q-1}$ satisfying

$$\begin{cases} G^p \in \Gamma(G^0, P), G^q = G, A^p \subseteq G^p, \\ G^{p+1} = G^p \setminus A^p \text{ and } P(G^p, A^p) = \text{true for all } p = 0, \ldots, q-1. \end{cases} \tag{B.1}$$

For the family $\Gamma(G^0, P)$ and the function P, we assume the following in Appendix B:

Assumption B.3 *If $G \in \Gamma(G^0, P)$, $A \subseteq G$ and $P(G, A) = $ true, then $P(G \cap H, A \cap H) = $ true for any $H \in \Gamma(G^0, P)$.*

Under Assumption B.3, the following lemma ensures the existence of the smallest set G^* in $\Gamma(G^0, P)$ in the sense that $G^* \subseteq G$ for any $G \in \Gamma(G^0, P)$.

Lemma B.4 *Let $G, H \in \Gamma(G^0, P)$. Then $G \cap H \in \Gamma(G^0, P)$.*

Proof: For G, we have the sequences $\{G^p\}_{p=0}^q$ and $\{A^p\}_{p=0}^{q-1}$ satisfying (B.1). We prove by induction on p that $G^p \cap H \in \Gamma(G^0, P)$ for all p. This implies $G \cap H = G^q \cap H \in \Gamma(G^0, P)$. Because $G \subseteq G^0$ for any $G \in \Gamma(G^0, P)$, it follows that $G^0 \cap H = H \in \Gamma(G^0, P)$. Next, we assume that $G^p \cap H \in \Gamma(G^0, P)$ for some p. Then it follows from 1 of Remark B.2 and

Assumption B.3 that $(G^p \cap H) \setminus (A^p \cap H) \in \Gamma(G^0, P)$. We have $G^{p+1} \cap H = (G^p \setminus A^p) \cap H = (G^p \cap H) \setminus A^p = (G^p \cap H) \setminus (A^p \cap H)$ and this completes the proof. $\square$

It follows from Lemma B.4 that $G^* := \bigcap_{G \in \Gamma(G^0,P)} G$ is the smallest set in $\Gamma(G^0, P)$ in the sense that $G^* \subseteq G$ for all $G \in \Gamma(G^0, P)$.

We propose an algorithm for finding G^*.

Algorithm B.5 (The elimination method)

Step 1 *Set* $i = 0$.

Step 2 *If there does not exist any nonempty subsets* A *of* G^i *such that* $P(G^i, A) = true$, *then stop and return* G^i.

Step 3 *Otherwise set* $G^{i+1} = G^i \setminus A$ *and* $i = i + 1$, *and go back to Step 2.*

At Step 2 of Algorithm B.5, we have a flexibility in choosing a nonempty subset A. The next theorem ensures that Algorithm B.5 always returns the smallest set G^*.

Theorem B.6 *The set returned by Algorithm B.5 is* $G^* \in \Gamma(G^0, P)$.

Proof: Let $\hat{G}$ be the set returned by Algorithm B.5. From Algorithm B.5, for any nonempty set $C \subseteq \hat{G}$, $P(\hat{G}, C) = \text{false}$. We assume $G^* \subsetneq \hat{G}$. Then there exists nonempty set D such that $\hat{G} = G^* \cup D$ and $G^* \cap D = \emptyset$. We have the sequences $\{G^p\}_{p=0}^{q}$ and $\{A^p\}_{p=0}^{q-1}$ satisfying (B.1) and $G^q = G^*$. Because $G^* \cap D = \emptyset$, there exists $p \in \{0, \dots, q-1\}$ such that $D \subseteq G^p$ and $D \not\subseteq G^{p+1} = G^p \setminus A^p$. This implies that $d \in A^p$ for some $d \in D$, and thus $A^p \cap \hat{G}$ is nonempty. Then $P(G^p \cap \hat{G}, A^p \cap \hat{G}) = P(\hat{G}, A^p \cap \hat{G}) = \text{true}$. For $\hat{G}$, by choosing $A = A^p \cap \hat{G}$ at Step 2, we can get a smaller set than $\hat{G}$. This implies that Algorithm B.5 returns a smaller set than $\hat{G}$, and thus contradicts the property of $\hat{G}$. Therefore $G^* = \hat{G}$. $\square$

C. A Proof of Theorem 3.3

In this appendix, we give a proof of Theorem 3.3. To this end, we use the results in Appendix B. In Definition B.1, we set $X = \mathbb{N}^n_{r_1} \times \cdots \times \mathbb{N}^n_{r_m}$ and P to be as in Definition 3.2. Then Definition B.1 is equivalent to the definition of $\Gamma(G^0, P)$ in Definition 3.2. Therefore, if the set family $\Gamma(G^0, P)$ defined in Definition 3.2 satisfies Assumption B.3, it follows from Theorem B.4 that $G \cap H \in \Gamma(G^0, P)$ if $G, H \in \Gamma(G^0, P)$. This ensures the existence of the smallest element G^* in $\Gamma(G^0, P)$. Moreover, it follows from Theorem B.6 that EEM described in Algorithm 3.1 always returns G^*. Therefore, it is sufficient to show that the set family $\Gamma(G^0, P)$ satisfies Assumption B.3.

The following lemma guarantees that the set family $\Gamma(G^0, P)$ satisfies Assumption B.3.

Lemma C.1 *Let* $G, H \in \Gamma(G^0, P)$. *If* $P(G, B(\delta)) = true$ *for some* $\delta \in \bigcup_{j=1}^m (F_j + G_j + G_j) \setminus (F \cup \bigcup_{j=1}^m T_j)$, *then* $P(G \cap H, B(\delta) \cap H) = true$.

Proof: For $G \cap H$, sets $J'(\delta), B'_j(\delta)$ and T'_j which correspond to $J(\delta), B_j(\delta)$ and T_j are as follows:

$$\begin{aligned}
J'(\delta) &= \{j \in \{1, \dots, m\} \mid \delta \in F_j + 2(G_j \cap H_j)\}, \\
B'_j(\delta) &= \{\alpha \in G_j \cap H_j \mid \delta - 2\alpha \in F_j\}, \\
T'_j &= \{\gamma + \alpha + \beta \mid \gamma \in F_j, \alpha, \beta \in G_j \cap H_j, \alpha \neq \beta\}.
\end{aligned}$$

Clearly, we have $B'_j(\delta) = B_j(\delta) \cap H_j$. Let $B'(\delta) := (B'_1(\delta), \dots, B'_m(\delta))$. If all $B'_j(\delta)$ are empty, then it follows from definition of P that $P(G \cap H, B(\delta) \cap H) =$ true. We assume that there exists $j \in \{1, \dots, m\}$ such that $B'_j(\delta) \neq \emptyset$. We will prove $P(G \cap H, B'(\delta)) =$ true under this assumption. Clearly, $\delta \notin F \cup \bigcup_{j=1}^m T'_j$ because of $T'_j \subseteq T_j$. If $\delta \notin \bigcup_{j=1}^m (F_j + (G_j \cap H_j) + (G_j \cap H_j))$, then $B'_j(\delta) = \emptyset$ for all $j = 1, \dots, m$. We consider the case where $\delta \in \bigcup_{j=1}^m (F_j + (G_j \cap H_j) + (G_j \cap H_j))$. $B'_j(\delta)$ and $J'(\delta)$ satisfy (2.6) because of $B'_j(\delta) = B_j(\delta) \cap H_j$ and $J'(\delta) \subseteq J(\delta)$. Therefore, $P(G \cap H, B(\delta) \cap H) =$ true. □

D. Proofs of Theorem 5.2 and Theorem 5.6

First of all, we give a proof of Theorem 5.2. To this end, we use the following lemma.

Lemma D.1 *If $r > \bar{r}$, then $2r - 1 > \deg(f)$.*

Proof: We have $r \geq \bar{r} + 1 \geq \lceil \deg(f)/2 \rceil + 1$ because of $r > \bar{r}$. Then we have $2r - 1 \geq 2\lceil \frac{\deg(f)}{2} \rceil + 1 > 2\lceil \frac{\deg(f)}{2} \rceil \geq \deg(f)$. □

Before we introduce Lemma D.2, we give some notation and symbols. Given $S_j \subseteq \mathbb{N}^n_{r_j} \setminus \mathbb{N}^n_{r_j - 1}$, let s_j be the largest element with the graded lexicographic order in the set S_j. For $S := (S_0, S_1, \dots, S_m)$, the set $A(S)$ denotes $\{\gamma_j + 2s_j \mid j = 0, \dots, m\}$. Note that $A(S)$ is empty if all sets $S_0, S_1, \dots, S_m$ are empty.

The following lemma ensures that the largest element in $A(S)$ satisfies (2.6) under Assumption 5.1.

Lemma D.2 *Assume $r > \bar{r}$. We define $G_j := \mathbb{N}^n_{r_j - 1} \cup S_j$ for all $j = 0, 1, \dots, m$. $\delta \in A(S)$ denotes the largest element with the graded lexicographic order in the set $A(S)$. Then $\delta \in \bigcup_{j=0}^m (F_j + G_j + G_j) \setminus (F \cup \bigcup_{j=0}^m T_j)$, and $J(\delta)$ and $B(\delta)$ satisfy (2.6), where $B(\delta) := (B_0(\delta), \dots, B_m(\delta))$.*

Proof: It follows from Lemma D.1 that $\delta \notin F$ because of $|\delta| = \deg(f_j) + 2r_j \geq 2r - 1$, and thus $\delta \in \bigcup_{j=0}^m (F_j + G_j + G_j) \setminus F$. For $\delta \in A(S)$, we consider $J(\delta) = \{j \in \{0, 1, \dots, m\} \mid \delta = \gamma_j + 2s_j\}$. Note that we have $\alpha_1 + \alpha_3 \succeq \alpha_2 + \alpha_4$ if given elements $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{N}^n$ satisfy $\alpha_1 \succeq \alpha_2$ and $\alpha_3 \succeq \alpha_4$. From this fact on the graded lexicographic order and the fact that δ, γ_j, s_j are the largest elements in $A(S), F_j, G_j$, respectively, it follows that $\gamma_j + 2s_j$ is the largest element in $F_j + G_j + G_j$, and thus δ is the largest element in $\bigcup_{j=0}^m (F_j + G_j + G_j)$. As a consequence, we have

$$B_j(\delta) = \begin{cases} \{s_j\} & \text{if } j \in J(\delta), \\ \emptyset & \text{o.w.} \end{cases}$$

Moreover, we can prove $\delta \notin T_j$ for all $j \in \{0, 1, \dots, m\}$. Indeed, if $\delta \in T_j$ for some j, then we have $\delta = \alpha + \beta_1 + \beta_2$ for some $\alpha \in F_j$ and $\beta_1 \neq \beta_2 \in G_j$. We assume $\beta_1 \succeq \beta_2$. Then $\alpha + 2\beta_1 \in F_j + G_j + G_j$ and $\alpha + 2\beta_1 \succeq \delta$, and thus this contradicts the fact that δ is the largest element in $\bigcup_{j=0}^m (F_j + G_j + G_j)$. Therefore $\delta \in \bigcup_{j=0}^m (F_j + G_j + G_j) \setminus (F \cup \bigcup_{j=0}^m T_j)$.

We need to check the sign of $(f_j)_{\gamma_j}$ for all $j \in J(\delta)$. We denote $J(\delta) = \{j_1, \dots, j_p\}$. Then $J(\delta) \subseteq \mathcal{I}(\gamma_{j_k})$ for $k = 1, \dots, p$ because we have $\gamma_{j_k} = \delta - 2s_{j_k} \equiv \delta \bmod 2$ for all $k = 1, \dots, p$. It follows from Assumption 5.1 that all the signs of $(f_j)_{\gamma_j}$ for all $j \in J(\delta)$ are the same sign. Therefore $J(\delta)$ and $B(\delta)$ satisfy (2.6). □

Proof of Theorem 5.2: We define $S_j = \mathbb{N}^n_{r_j} \setminus \mathbb{N}^n_{r_j - 1}$ and $G_j = \mathbb{N}^n_{r_j - 1} \cup S_j$ for all $j = 0, 1, \dots, m$. By applying Lemma D.2, then we can remove s_j from G_j and S_j for $j \in$

$J(\delta)$. Next, we construct the set $A(S)$ from the resulting sets $S_0, S_1, \ldots, S_m$ and apply Lemma D.2 again. Lemma D.2 ensures that one can remove the largest element s_j in some sets S_j as long as the set $A(S)$ is not empty. By repeating this procedure, all sets $S_0, S_1, \ldots, S_m$ become empty. This implies that the resulting SOS problem is equivalent to SOS problem with relaxation order $r-1$. Therefore, by induction on r, the SDP relaxation problem with relaxation order r is equivalent to the SDP relaxation problem with relaxation order $\bar{r}$. $\square$

Next, we prove Theorem 5.6. Let $G_j \subseteq \mathbb{N}^n_{r_j}$ and s_j be the smallest element in G_j with the graded lexicographic order for $j = 0, 1, \ldots, m$. We define the set $C(G) = \{\epsilon_j + 2s_j \mid j = 0, 1, \ldots, m\}$ for $G := (G_0, \ldots, G_m)$.

Lemma D.3 *$\delta \in C(G)$ denotes the smallest element with the graded lexicographic order in the set $C(G)$. Then $\delta \in \bigcup_{j=0}^m (F_j + G_j + G_j) \setminus (F \cup \bigcup_{j=0}^m T_j)$, and $J(\delta)$ and $B(\delta)$ satisfy (2.6), where $B(\delta) := (B_0(\delta), \ldots, B_m(\delta))$.*

Proof: It follows from 1 and 3 of Assumption 5.5 that $\alpha \notin F$ for all $\alpha \in C(G)$. By applying a similar discussion in Lemma D.2 and 2 of Assumption 5.5, we can prove this lemma. $\square$

Proof of Theorem 5.6: Let $G_j = \mathbb{N}^n_{r_j}$ for all $j = 0, 1, \ldots, m$. Applying Lemma D.3, we can remove s_j from G_j for $j \in J(\delta)$. Next, we construct the set $C(G)$ from the resulting sets $G_0, G_1, \ldots, G_m$ and apply Lemma D.3 again. Lemma D.3 ensures that one can remove the smallest element s_j in some sets G_j as long as the set $C(G)$ is not empty. Note that we have $|\epsilon_j + 2s_j| < 2r$ for $j = 1, \ldots, m$ because of 3 of Assumption 5.5. Therefore, by applying this procedure repeatedly, all sets $G_1, \ldots, G_m$ become empty and $G_0 = \mathbb{N}^n_r \setminus \mathbb{N}^n_{r-1}$. This implies $f \in f_0 \Sigma_{r_0} = \Sigma_r$. $\square$

Acknowledgements

We would like to thank Dr. Yoshio Okamoto for very helpful comments on Appendix B. The first author was supported by Grant-in-Aid for JSPS Fellow 20003236 and Grant-in-Aid for Young Scientists (B) 22740056. The second author was partially supported by Grant-in-Aid for Scientific Research (C) 19560063.

References

[1] B. Borchers: CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, **11** (1999), 613–623.

[2] J.M. Borwein and H. Wolkowicz: Facial reduction for a cone-convex programming problem. *Journal of the Australian Mathematical Society*, **30** (1981), 369–380.

[3] J.M. Borwein and H. Wolkowicz: Regularizing the abstract convex program. *Journal of Mathematical Analysis and Applications*, **83** (1981), 495–530.

[4] M.D. Choi, T.Y. Lam, and B. Reznick, Sums of squares of real polynomials. In B. Jacob and A. Rosenberg (eds.): *Proceeding of Symposia Pure Mathematics*, **58** (American Mathematical Society, 1995), 103–126.

[5] K. Fujisawa, M. Fukuda, K. Kobayashi, M. Kojima, K. Nakata, M. Nakata, and M. Yamashita: SDPA (SemiDefinite Programming Algorithm) user's manual — version 7.0.5. Department of Mathematical and Computer Sciences, Tokyo Institute of Technology (2008), available at http://sdpa.indsys.chuo-u.ac.jp/sdpa/

[6] GLOBAL Library, http://www.gamsworld.org/global/globallib.htm

[7] E. de Klerk, M. Laurent, and P.A. Parrilo: On the equivalence of algebraic approaches to the minimization of forms on the simplex. In D. Henrion and A. Garulli (eds.): *Positive Polynomials in Control* (Springer-Verlag, Berlin, 2005), 121–132.

[8] M. Kojima, S. Kim, and H. Waki: Sparsity in sums of squares of polynomials. *Mathematical Programming*, **103** (2005), 45–62.

[9] J.B. Lasserre: Global optimization with polynomials and the problems of moments. *SIAM Journal on Optimization*, **11** (2001), 796–817.

[10] H.D. Mittelmann: An independent benchmarking of SDP and SOCP solvers. *Mathematical Programming*, **95** (2003), 407–430.

[11] P.A. Parrilo: Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, **96** (2003), 293–320.

[12] G. Pataki: A simple derivation of a facial reduction algorithm and extended dual systems. Department of Statistics and OR, University of North Carolina at Chapel Hill (2000).

[13] V. Powers and T. Wörmann: An algorithm for sums of squares of real polynomials. *Journal of Pure and Applied Algebra*, **127** (1998), 99–104.

[14] A. Prestel and C. N. Delzell: *Positive Polynomials* (Springer-Verlag, Berlin, 2001).

[15] M. Putinar: Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, **42** (1993), 969–984.

[16] M.V. Ramana: An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, **77** (1997), 129–162.

[17] M.V. Ramana, L. Tunçel, and H. Wolkowicz: Strong duality for semidefinite programming. *SIAM Journal on Optimization*, **7** (1997), 641–662.

[18] J.F. Strum: SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, **11 & 12** (1999), 625–653.

[19] R.H. Tütüncü, K.C. Toh, and M.J. Todd: Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming Series B*, **95** (2003), 189–217.

[20] H. Waki, S. Kim, M. Kojima, and M. Muramatsu: Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, **17** (2006), 218–242.

[21] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto: Algorithm 883: SparsePOP : a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Transactions on Mathematical Software*, **15** (2008), 15:1–15:13, available at http://sourceforge.net/projects/sparsepop/

[22] H. Waki and M. Muramatsu: Facial reduction algorithms for conic optimization problems. Department of Computer Science, The University of Electro-Communications (2009), available at http://www.optimization-online.org/DB_HTML/2009/07/2355.html

[23] H. Waki and M. Muramatsu: A facial reduction algorithm for finding sparse SOS representations. *Operations Research Letters*, **38** (2010), 361–365.

[24] H. Waki, M. Muramatsu, and M. Kojima: Invariance under affine transformation in semidefinite programming relaxation for polynomial optimization problems. *Pacific Journal of Optimization*, **5** (2009), 297–312.

[25] H. Waki, M. Nakata, and M. Muramatsu: Strange behaviors of interior-point methods for solving semidefinite programming problems in polynomial optimization. to appear in *Computational Optimization and Applications* (DOI: 10.1007/s10589-011-9437-8).

Hayato Waki
The University of Electro-Communications
Chofu-gaoka 1-5-1, Chofu
Tokyo 182-8585, Japan
E-mail: `waki@cs.uec.ac.jp`