SIAM J. OPTIM.
Vol. 28, No. 1, pp. 1–29

A MULTIGRID APPROACH TO SDP RELAXATIONS OF SPARSE POLYNOMIAL OPTIMIZATION PROBLEMS*

JUAN S. CAMPOS† AND PANOS PARPAS†

Abstract. We propose a multigrid approach for the global optimization of polynomial optimization problems with sparse support. The problems we consider arise from the discretization of infinite dimensional optimization problems, such as PDE optimization problems, boundary value problems, and some global optimization applications. In many of these applications, the level of discretization can be used to obtain a hierarchy of optimization models that capture the underlying infinite dimensional problem at different degrees of fidelity. This approach, inspired by multigrid methods, has been successfully used for decades to solve large systems of linear equations. However, multigrid methods are difficult to apply to semidefinite programming (SDP) relaxations of polynomial optimization problems. The main difficulty is that the information between grids is lost when the original problem is approximated via an SDP relaxation. Despite the loss of information, we develop a multigrid approach and propose prolongation operators to relate the primal and dual variables of the SDP relaxation between lower and higher levels in the hierarchy of discretizations. We develop sufficient conditions for the operators to be useful in practice. Our conditions are easy to verify, and we discuss how they can be used to reduce the complexity of infeasible interior point methods. Our preliminary results highlight two promising advantages of following a multigrid approach compared to a pure interior point method: the percentage of problems that can be solved to a high accuracy is much greater, and the time necessary to find a solution can be reduced significantly, especially for large scale problems.

Key words. multigrid, semidefinite programming, sparse polynomial optimization, differential equations

AMS subject classifications. 35G30, 65H10, 65N06, 65N22, 65N55, 90C22, 90C26, 90C51, 90C59

DOI. 10.1137/16M1109060

1. Introduction.

Exploiting sparsity with specialized semidefinite programming (SDP) relaxations had a huge impact on the application of SDP relaxations to realistic polynomial optimization problems. Indeed, when using the classical Lasserre hierarchy, it is only possible to solve problems with a few dimensions, but by exploiting the sparsity present in many applications, it is possible to solve problems with several hundred variables [18, 36]. In this paper, we argue that many applications have additional structure that can be exploited to a similar effect. In particular, many large scale polynomial optimization problems have their origins in the discretization of an infinite dimensional model. The resulting finite dimensional model is sparse but has a large number of degrees of freedom. Optimization models that fit this class are boundary value problems [28], optimization with PDEs [12, 4], optimal control [3, 8], and Markov decision processes [13], among others. Despite the progress made in the last decade, it is still not possible to solve realistic instances of the models arising in these applications using sparse SDP relaxations. The main contribution of this paper is to show how to take advantage of both sparse and hierarchical structures present in many applications. Our theoretical results suggest that under appropriate conditions

*Received by the editors December 21, 2016; accepted for publication (in revised form) September 14, 2017; published electronically January 5, 2018.
http://www.siam.org/journals/siopt/28-1/M110906.html
Funding: The work of the second author was partially supported by EPSRC grant EP/M028240.
†Department of Computing, Imperial College London, London SW7 2AZ, UK (jsc12@ic.ac.uk, panos.parpas@ic.ac.uk).

we should expect significant improvements in computational complexity. Our numerical results further support this claim, and we show that a multigrid approach can improve the robustness and reduce the time required to solve large scale polynomial optimization problems.

Our approach is inspired by multigrid methods. When solving a system of linear equations, and in some optimization problems, it is widely accepted that if a multigrid method is applicable, then it is often the best numerical method to use [5, 33]. For examples of the multigrid approach to various optimization problems we refer the interested reader to [5, 10, 26, 39] for PDE optimization, [15] for convex optimization problems in image processing, and [13] for Markov decision processes. The core principle of multigrid methods is to construct a coarse model of the original (high resolution/fine) model and use the information obtained from solving the coarse model to improve the current solution. This approach works extremely well when coarse and fine models share a common structure. Additionally, based on the intuition that the coarse model is a global approximation to the fine model (as opposed to only using local information to construct a search direction), the hope is that multigrid methods can potentially be applied to global optimization problems too.

Motivated by the potential numerical improvements and the fact that the coarse model retains global information about the model, we develop the multigrid principle for SDP relaxations of polynomial optimization problems (POP). In particular, we propose a multigrid framework for the SDP relaxation of the following POP:

$$\min_{\mathbf{x}\in\mathbb{R}^n} F_n(\mathbf{x}) \triangleq \sum_{k=1}^{n_1-1} f_k(\mathbf{x}_k) + \sum_{k=n_1}^{n_2} f_0(\mathbf{x}_k) + \sum_{k=n_2+1}^{n-p+1} f_k(\mathbf{x}_k), \tag{1}$$

where $f_k : \mathbb{R}^p \mapsto \mathbb{R}$ are p-dimensional polynomial functions of degree d_k, $k = 0, 1, \ldots, n_1 - 1, n_2 + 1, \ldots, n - p + 1$, $\mathbf{x}_k = (x_k, x_{k+1}, \ldots, x_{k+p-1})$, and n_1, n_2, n are positive integers such that $n_1+p+1 \leq n_2 \leq n-p+1$. Note that the problem is sparse in the sense that every variable only appears together with $p-1$ of its neighbors. In our numerical experiments we typically have $n_1 = 2$ and $n_2 = n - p$. The principal technical difficulty of applying multigrid to a (sparse or otherwise) SDP relaxation of (1) is that the information among the variables is lost through the relaxation process. In this paper, we take the first steps towards addressing this issue. We show that despite the loss of information, it is still possible to obtain useful information from the coarse SDP relaxation and to construct a good approximation to the solution of the fine SDP relaxation. We do not propose a new hierarchy for polynomial problems. Instead, we take the most popular and widely used hierarchy for sparse problems [36] and show how the resulting semidefinite programs can be solved more efficiently by making use of additional structures present in many applications. In particular, our main contributions are as follows:

1. We present the construction of operators that relate the primal and dual solutions of the coarse SDP relaxation to the original SDP relaxation. Borrowing terminology from the multigrid literature we call these operators *prolongation operators*.

2. We give derivations of sufficient and easily verifiable conditions for these operators to be useful in practice.

3. We show that if our conditions are satisfied, then it is possible to improve the worst case complexity of infeasible interior point methods.

4. We report the results from our numerical experiments that show that our conditions are indeed satisfied in many practical problems and that our multigrid framework can be used to improve the numerical performance of infeasible interior

point methods.

A multigrid approach in the context of sparse SDP relaxations was used in [21, 22] to solve finite difference approximations to optimal control problems and PDE problems. However, the authors used the standard multigrid operators to interpolate between the variables in the original space, whereas we work directly with the primal/dual variables of the SDP relaxation. The SDP variables contain much more information than just the solution to (1). The additional information can be put to good use in the next level of the hierarchy. This advantage is reflected in our numerical and theoretical results. In particular, we can solve bigger problems with our approach rather than using SDP relaxations as a black box. Another related approach is the application of multigrid methods to SDP relaxations of combinatorial optimization problems [20]. Their approach is specific to the particular SDP relaxations appearing in graph problems and are not applicable to the general SDP relaxations of (1) that we consider in this paper.

The rest of the paper is structured as follows: section 2 defines the notation used, and in section 3 we review sparse relaxations for unconstrained problems developed in [36]. In section 4 we study the characteristics of the relaxation when applied to problem (1), and section 5 defines the hierarchy at different levels and the prolongation operators. Section 5 also establishes improvements in worst case complexity of an infeasible interior point method when our assumptions are satisfied. Finally, in section 6 we report results from our numerical experiments.

2. Notation and preliminaries. Given a real-valued polynomial function $f : \mathbb{R}^n \to \mathbb{R}$ of degree d, we denote the monomial $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ by $\mathbf{x}^{\boldsymbol{\alpha}}$ and denote its coefficient by $b_{\boldsymbol{\alpha}}$, where $\boldsymbol{\alpha} \in \mathbb{N}^n$ and $\mathbb{N}$ is the set of nonnegative integers. Letting $\Gamma_d^n = \{\boldsymbol{\alpha} \in \mathbb{N}^n : \sum_i \alpha_i \leq d\}$, any polynomial of degree at most d can be written as $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \Gamma_d^n} b_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$. The support of f is defined by $\operatorname{supp}(f) = \{\boldsymbol{\alpha} \in \Gamma_d^n : b_{\boldsymbol{\alpha}} \neq 0\}$. For any set $\Phi \subseteq \{1, 2, \dots, n\}$, let $\mathbb{A}_d^\Phi = \{\boldsymbol{\alpha} \in \mathbb{N}^n : \sum_i \alpha_i \leq d, \alpha_i = 0 \text{ if } i \notin \Phi\}$, and let $u\left(\mathbf{x}, \mathbb{A}_d^\Phi\right)$ be a column vector with the monomials $\mathbf{x}^{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathbb{A}_d^\Phi$. For example, if $\Phi = \{2, 4\}$ and $d = 2$, then $u\left(\mathbf{x}, \mathbb{A}_d^\Phi\right) = [1, x_2, x_4, x_2^2, x_2 x_4, x_4^2]^\top$. The size of the vector $u\left(\mathbf{x}, \mathbb{A}_d^\Phi\right)$ is equal to $\binom{|\Phi|+d}{d} = \frac{(|\Phi|+d)!}{|\Phi|!d!}$ and will be denoted by $g(|\Phi|, d)$, where $|\Phi|$ corresponds to the number of elements in the set Φ. For any matrix $Q \in \mathbb{R}^{r_1 \times r_2}$, $[Q]_{i,j}$ will correspond to the element in position (i, j) (if $r_1 = 1$ or $r_2 = 1$, we will write $[Q]_i$ for the ith element). If $Q \in \mathbb{R}^{r \times r}$ is a symmetric matrix, then $\lambda_i(Q)$ will represent the ith eigenvalue of Q, where $\lambda_1(Q) \leq \lambda_2(Q) \leq \dots \leq \lambda_r(Q)$. Likewise, if $Q_1, Q_2 \in \mathbb{R}^{r \times r}$ are two matrices, we will use the usual inner product $\langle Q_1, Q_2 \rangle = \operatorname{Tr}(Q_1^\top Q_2)$ and its induced norm $\|Q\|^2 = \langle Q, Q \rangle$ (where $\operatorname{Tr}(Q)$ is the trace of the matrix Q). For any symmetric matrix $Q \in \mathbb{R}^{r \times r}$, $Q \succeq (\succ)\ 0$ means that Q is positive semidefinite (resp., definite). The matrix $I \in \mathbb{R}^{r \times r}$ will represent the identity matrix, and its size will be understood from the context. The notation so far is standard, and we refer the interested reader to [36] for more details and examples. Below we introduce two definitions that are specific to this paper.

DEFINITION 1. *For any* $1 \leq i \leq n$, *let* $B_i \triangleq \{\boldsymbol{\alpha} \in \Gamma_d^n : \alpha_i > 0, \alpha_j = 0 \text{ for } j < i\}$.

Note that $\Gamma_d^n \setminus \{\mathbf{0}\} = \cup_{i=1}^n B_i$, and $B_i \cap B_j = \emptyset$ for any $i \neq j$.

Example 2. If $n = d = 2$, then

$$\Gamma_d^n = \{[0,0]^\top, [1,0]^\top, [0,1]^\top, [2,0]^\top, [1,1]^\top, [0,2]^\top\},$$
$$B_1 = \{[1,0]^\top, [1,1]^\top, [2,0]^\top\},$$
$$B_2 = \{[0,1]^\top, [0,2]^\top\}.$$

DEFINITION 3. *If* $\boldsymbol{\alpha} \in \mathbb{R}^n$ *is equal to* $[\alpha_1, \alpha_2, \ldots, \alpha_n]^\top$, *then* $\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^- \in \mathbb{R}^n$ *are defined as* $\boldsymbol{\alpha}^+ \triangleq [0, \alpha_1, \alpha_2, \ldots, \alpha_{n-1}]^\top$ *and* $\boldsymbol{\alpha}^- \triangleq [\alpha_2, \alpha_3, \ldots, \alpha_n, 0]^\top$. *Likewise, if* $t \in \mathbb{N}$ *and* $t \geq 2$, *then* $\boldsymbol{\alpha}^{-t} \in \mathbb{R}^n$ *is defined as* $\boldsymbol{\alpha}^{-t} \triangleq [\alpha_{t+1}, \alpha_{t+2}, \ldots, \alpha_n, 0, \ldots, 0]^\top$.

Example 4. Let $\boldsymbol{\alpha} = [1,4,6,0]^\top$. Then $\boldsymbol{\alpha}^- = [4,6,0,0]^\top$, $\boldsymbol{\alpha}^+ = [0,1,4,6]^\top$, and $\boldsymbol{\alpha}^{-2} = [6,0,0,0]^\top$.

3. Sparse POP relaxations. In our work we will use the relaxations formulated in [36] to find an approximate solution for problem (1). In this section, we briefly describe the so-called sparse relaxations for unconstrained problems.

Consider the unconstrained POP for the function $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \Gamma_d^n} b_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$ with even degree d, given by

$$f^\star \triangleq \min_{\mathbf{x} \in \mathbb{R}^n} \sum_{\boldsymbol{\alpha} \in \Gamma_d^n} b_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}. \tag{2}$$

In [16] Lasserre developed a hierarchy of SDP relaxations for polynomial minimization and proved that under certain assumptions, it is possible to extract an approximate solution for problems like (2) by solving these relaxations (see [11] for details). This hierarchy can be obtained by viewing (2) as a moment problem or trying to compute a sum-of-squares decomposition of $f(x) - f^\star$ (see, for example, [29, 30] for more details). However, if the number of variables (or constraints in the general case) is large, the resulting semidefinite program can be too large to be solved in practice. For the case when the number of elements in the support of the objective function is small, Waki et al. [36] developed a sparse relaxation that reduces the number of variables and constraints in the SDP relaxation. Although this methodology was originally heuristic, Lasserre proved in [17] that the optimal value of this new hierarchy converges to the optimal value of the polynomial problem under some additional technical assumptions.

In order to define the sparse hierarchy, Waki et al. [36] used the structure of the so-called correlative sparsity pattern (CSP) matrix. If R is the CSP matrix, then $[R]_{i,j}$ is nonzero if there exists a monomial with variables x_i and x_j which has a nonzero coefficient in the objective function, and $[R]_{i,j}$ is zero otherwise. If R is sparse, then problem (2) is called correlatively sparse. Associated to the CSP matrix is the CSP graph $G(V,E)$. The node set is $V = \{1, 2, \ldots, n\}$ and $E = \{\{i,j\} : i, j \in V, [R]_{i,j} = \star, i < j\}$, where $\star$ is any nonzero real number. The idea is to generate a set of support sets for the polynomial function using the maximal cliques of this graph. However, finding the maximal cliques of a graph is in general NP-hard (see, for example, [2]), and for this reason the sparse relaxations are defined using the maximal cliques of a chordal extension of the CSP graph.[1] We refer the reader to [1, 9] for more information about chordal extensions and algorithms to find maximal cliques in chordal graphs.

[1] $G(V,E')$ is a chordal extension of $G(V,E)$ if $E \subseteq E'$ and $G(V,E')$ is chordal.

Let $\{\Phi_l\}_{l=1}^m$ be the maximal cliques of a chordal extension of $G(V,E)$, and note that $zz^\top \succeq 0$ for any real vector z. Then, adding the constraints $u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right) u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)^\top \succeq 0$ $(l = 1, 2, \ldots, m)$ to problem (2), we obtain the following equivalent problem:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{\boldsymbol{\alpha}\in\Gamma_d^n} b_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \\ &\text{s.t. } u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right) u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)^\top \succeq 0,\ l = 1, 2, \ldots, m, \end{aligned} \tag{3}$$

where $w \geq d/2$ is a degree that denotes the relaxation order. Note that the left-hand side of constraint l is a square matrix containing monomials $\mathbf{x}^{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in \mathbb{A}_{2w}^{\Phi_l}$. Let $M_w^{\Phi_l}(\mathbf{x}^{\boldsymbol{\alpha}}) = u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right) u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)^\top$, and let $y = \{y_{\boldsymbol{\alpha}}\}$. If the monomial $\mathbf{x}^{\boldsymbol{\alpha}}$ is replaced with the real variable $y_{\boldsymbol{\alpha}}$, the wth sparse SDP relaxation is given by

$$\begin{aligned} &\min_{y} \sum_{\boldsymbol{\alpha}\in\mathcal{F}} b_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}} \\ &\text{s.t. } M_w^{\Phi_l}(y) \succeq 0,\ l = 1, 2, \ldots, m, \end{aligned} \tag{4}$$

where $\mathcal{F} = \cup_{l=1}^m \mathbb{A}_{2w}^{\Phi_l} \setminus \{\mathbf{0}\}$. The matrix $M_w^{\Phi_l}(y)$ is called the wth moment matrix for variables indexed by the set $\mathbb{A}_w^{\Phi_l}$. Note that this SDP relaxation admits a strict interior point (see Theorem 3.1 in [27]).

4. Sparse POP relaxations for problem (1). In this section we analyze the sparse relaxation for problem (1) and derive connections between the variables and constraints that will be useful in the multigrid setting.

Let $R \in \mathbb{R}^{n\times n}$ be the CSP matrix for (1), and let $G(V,E)$ be the associated CSP graph. The CSP matrix is a band symmetric matrix with bandwidth equal to $p-1$. This follows from the fact that the polynomials f_k in (1) are functions of $\{x_l, x_{l+1}, \ldots, x_{l+p-1}\}$ for $l = 1, 2, \ldots, n-p+1$, and therefore for any $\boldsymbol{\alpha} \in \operatorname{supp}(F_n)$, if $\alpha_i > 0$ and $\alpha_j > 0$, then $|i-j| \leq p-1$. Given that the graph G is not necessarily chordal, we will consider as a chordal extension of G the graph $G(V,E')$, where $E \subseteq E' = \{\{i,j\} : i,j \in V, |i-j| \leq p-1\}$. The CSP matrix for this chordal extension is

$$[R]_{i,j} = \begin{cases} \star \text{ if } i = j, \\ \star \text{ if } |i-j| \ \leq p-1, \\ 0 \text{ otherwise.} \end{cases} \tag{5}$$

The next lemma establishes that $G(V,E')$ is indeed a chordal graph and identifies its maximal cliques.

LEMMA 5. *If the CSP matrix R is given by* (5)*, then the associated CSP graph $G(V,E')$ is chordal, and the maximal cliques are given by $\Phi_l = \{l, l+1, \ldots, l+p-1\}$ for $l = 1, 2, \ldots, n-p+1$.*

Proof. The graph $G(V,E')$ is chordal because it is an interval graph with intervals $I_k = [k, k+p-1]$ for $k = 1, 2, \ldots, n$ (see section 3.2 in [35] for a definition of interval graphs). Using the definition of maximal cliques it is straightforward to show that the sets Φ_l for $l = 1, 2, \ldots, n-p+1$ are indeed the maximal cliques of $G(V,E')$. ∎

Let d (even) be the degree of $F_n(\mathbf{x})$ in (1), and write $F_n(\mathbf{x})$ as $\sum_{\boldsymbol{\alpha}\in\Gamma_d^n} b_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$ for appropriate $b_{\boldsymbol{\alpha}}$. If $w \geq d/2$ denotes the order of the relaxation, and Φ_l $(l = 1, 2, \ldots, n-p+1)$ are the maximal cliques defined in Lemma 5 of the chordal extension

$G(V, E')$, then the wth SDP sparse relaxation for problem (1) (after deleting any constant terms in the objective function) is given by

$$\begin{aligned} &\min_{y} \sum_{\boldsymbol{\alpha}\in\mathcal{F}} b_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}} \\ &\text{s.t. } M_w^{\Phi_l}(y) \succeq 0,\ l = 1, 2, \ldots, n-p+1, \end{aligned} \tag{6}$$

where $\mathcal{F} = \cup_{l=1}^{n-p+1} \mathbb{A}_{2w}^{\Phi_l} \setminus \{\mathbf{0}\}$. Letting $S = (S_1, S_2, \ldots, S_{n-p+1})$, we can write (6) as

$$\begin{aligned} &\min_{y,S} \sum_{\boldsymbol{\alpha}\in\mathcal{F}} b_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}} \\ &\text{s.t. } \sum_{\boldsymbol{\alpha}\in\mathcal{F}} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}} + C = S_l, \\ &\qquad S_l \succeq 0,\ l = 1, 2, \ldots, n-p+1, \end{aligned} \tag{7}$$

where $A_{(l,\boldsymbol{\alpha})}$ can be deduced from the definition of the moment matrix $M_w^{\Phi_l}(y)$ ($l = 1, 2, \ldots, n-p+1$), and C is a matrix with one in position $(1,1)$ and zeros everywhere else corresponding to the monomial of degree zero. Due to Lemma 5 all the moment matrices in (6) and S_l have the same dimension $g(p,w) \times g(p,w)$. Letting $X = (X_1, X_2, \ldots, X_{n-p+1})$, the dual of problem (7) is

$$\begin{aligned} &\max_{X} - \sum_{l=1}^{n-p+1} \langle C, X_l \rangle \\ &\text{s.t.} \quad \sum_{l=1}^{n-p+1} \left\langle A_{(l,\boldsymbol{\alpha})}, X_l \right\rangle = b_{\boldsymbol{\alpha}} \text{ for } \boldsymbol{\alpha} \in \mathcal{F}, \\ &\qquad X_l \succeq 0,\ l = 1, 2, \ldots, n-p+1. \end{aligned} \tag{8}$$

The matrices in the constraints of the relaxations satisfy important properties that will be used in the proofs. We illustrate two of these properties with an example and summarize them along with other properties in Lemma 7.

Example 6. Suppose that $\Phi_l = \{l, l+1\}$ (i.e., $p = 2$) and $w = 1$; then

$$\begin{aligned} u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_i}\right) u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_i}\right)^\top &= \begin{bmatrix} 1 & x_i & x_{i+1} \\ x_i & x_i^2 & x_i x_{i+1} \\ x_{i+1} & x_i x_{i+1} & x_{i+1}^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + x_i \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + x_{i+1} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ &\quad + x_i^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + x_i x_{i+1} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} + x_{i+1}^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

The six matrices in the above equation are independent of the clique Φ_i. This means that the matrices multiplying the monomials $x_i, x_{i+1}, x_i^2, x_i x_{i+1}, x_{i+1}^2$ in this equation will be the same matrices multiplying the monomials x_{i+1}, x_{i+2}, x_{i+1}^2, $x_{i+1}x_{i+2}$, x_{i+2}^2 in the equation for $i+1$, respectively (i.e., $u(\mathbf{x}, \mathbb{A}_w^{\Phi_{i+1}}) u(\mathbf{x}, \mathbb{A}_w^{\Phi_{i+1}})^\top$). Also, monomials $\mathbf{x}^{\boldsymbol{\alpha}}$ with $\alpha_k > 0$ for $k \leq i-1$ or $k \geq i+2$ do not belong to the equation which means they have a zero matrix coefficient.

LEMMA 7. *Let $A_{(l,\boldsymbol{\alpha})}$ for $l = 1, 2, \ldots, n-p+1$ and $\boldsymbol{\alpha} \in \mathcal{F}$ be the matrices in problems* (7) *and* (8)*.*

(a) *$A_{(l,\boldsymbol{\alpha})} = A_{(l+1,\boldsymbol{\alpha}^+)}$ for any $1 \leq l \leq n-p$. Equivalently $A_{(l,\boldsymbol{\alpha})} = A_{(l-1,\boldsymbol{\alpha}^-)}$ for any $2 \leq l \leq n-p+1$.*

(b) *If $\alpha_k > 0$ and $k \leq l-1$ or $k \geq l+p$, then $A_{(l,\boldsymbol{\alpha})} = \mathbf{0}$ for $1 \leq k \leq n$.*

(c) *For all $\mathcal{H} \subseteq \mathcal{F}$, if $z_{\boldsymbol{\alpha}} \in \mathbb{R}$ is such that $|z_{\boldsymbol{\alpha}}| \leq \zeta$ for all $\boldsymbol{\alpha} \in \mathcal{F}$, then $\left\|\sum_{\boldsymbol{\alpha}\in\mathcal{H}} A_{(l,\boldsymbol{\alpha})} z_{\boldsymbol{\alpha}}\right\| \leq g(p,w)\zeta$ for any $l = 1, 2, \ldots, n-p+1$.*

(d) *$\sum_{\boldsymbol{\alpha}\in B_{l-1}} A_{(l-1,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\alpha}\in B_l} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}^-}$ for $l = 2, 3, \ldots, n-p+1$.*

(e) *$\sum_{\boldsymbol{\alpha}\in\mathcal{F}} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}$ for $l = 1, 2, \ldots, n-p+1$.*

Proof. By definition, $M_w^{\Phi_l}(y) = \sum_{\boldsymbol{\alpha}\in\mathcal{F}} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}} + C$ ($l = 1, 2, \ldots, n-p+1$), and therefore,

$$M_w^{\Phi_l}(\mathbf{x}^{\boldsymbol{\alpha}}) = u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right) u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)^\top = \sum_{\boldsymbol{\alpha}\in\mathcal{F}} A_{(l,\boldsymbol{\alpha})} \mathbf{x}^{\boldsymbol{\alpha}} + C.$$

From the equation above we can deduce that for any $1 \leq i, j \leq g(p,w)$ ($u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)$ is a vector with $g(|\Phi_l|, w) = g(p,w)$ elements),

$$\left[A_{(l,\boldsymbol{\alpha})}\right]_{i,j} = \begin{cases} 1 \text{ if } \mathbf{x}^{\boldsymbol{\alpha}} = \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_i \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_j, \\ 0 \text{ otherwise.} \end{cases} \tag{9}$$

(a) We will prove the first part of the statement; the second part can be shown following the same reasoning. Note that if the vectors u are ordered according to some polynomial ordering (see [7] for more on orderings) we have that if $\mathbf{x}^{\boldsymbol{\gamma}_i} = \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_i$, then $\mathbf{x}^{\boldsymbol{\gamma}_i^+} = [u(\mathbf{x}, \mathbb{A}_w^{\Phi_{l+1}})]_i$ (i.e., if the monomial $\mathbf{x}^{\boldsymbol{\gamma}_i}$ is the ith element of $u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)$, then $\mathbf{x}^{\boldsymbol{\gamma}_i{}^+}$ is the ith element of $u(\mathbf{x}, \mathbb{A}_w^{\Phi_{l+1}})$). Therefore, (9) implies that to prove (a) we need to show that the following two conditions are true: If $\mathbf{x}^{\boldsymbol{\alpha}} = \mathbf{x}^{\boldsymbol{\gamma}_i}\mathbf{x}^{\boldsymbol{\gamma}_j}$, then $\mathbf{x}^{\boldsymbol{\alpha}^+} = \mathbf{x}^{\boldsymbol{\gamma}_i^+}\mathbf{x}^{\boldsymbol{\gamma}_j^+}$ (i.e., $\left[A_{(l,\boldsymbol{\alpha})}\right]_{i,j} = \left[A_{(l+1,\boldsymbol{\alpha}^+)}\right]_{i,j} = 1$), and if $\mathbf{x}^{\boldsymbol{\alpha}} \neq \mathbf{x}^{\boldsymbol{\gamma}_i}\mathbf{x}^{\boldsymbol{\gamma}_j}$, then $\mathbf{x}^{\boldsymbol{\alpha}^+} \neq \mathbf{x}^{\boldsymbol{\gamma}_i^+}\mathbf{x}^{\boldsymbol{\gamma}_j^+}$ (i.e., $\left[A_{(l,\boldsymbol{\alpha})}\right]_{i,j} = \left[A_{(l+1,\boldsymbol{\alpha}^+)}\right]_{i,j} = 0$). Since $\left(\boldsymbol{\gamma}_i + \boldsymbol{\gamma}_j\right)^+ = \boldsymbol{\gamma}_i^+ + \boldsymbol{\gamma}_j^+$, we can deduce that if $\boldsymbol{\alpha} = \boldsymbol{\gamma}_i + \boldsymbol{\gamma}_j$, then $\mathbf{x}^{\boldsymbol{\alpha}^+} = \mathbf{x}^{\left(\boldsymbol{\gamma}_i+\boldsymbol{\gamma}_j\right)^+} = \mathbf{x}^{\boldsymbol{\gamma}_i^+ + \boldsymbol{\gamma}_j^+}$, from which it follows that both conditions are true.

(b) If $k \leq l-1$ or $k \geq l+p$, it follows from the definition of Φ_l that the variable x_k has a zero exponent in every monomial in $u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)$. Thus, if $\alpha_k > 0$, $\mathbf{x}^{\boldsymbol{\alpha}} \neq \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_i \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_j$ for any i, j. We can therefore conclude that $\left[A_{(l,\boldsymbol{\alpha})}\right]_{i,j} = 0$ for all i, j.

(c) For any $\boldsymbol{\alpha} \neq \boldsymbol{\gamma}$, if $\mathbf{x}^{\boldsymbol{\alpha}} = \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_i \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_j$, it follows that $\mathbf{x}^{\boldsymbol{\gamma}} \neq \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_i \left[u\left(\mathbf{x}, \mathbb{A}_w^{\Phi_l}\right)\right]_j$. Then according to (9), if $A_{(l,\boldsymbol{\alpha})}$ has a nonzero element in position (i,j), $A_{(l,\boldsymbol{\gamma})}$ must have a zero in position (i,j) for any $\boldsymbol{\alpha} \neq \boldsymbol{\gamma}$. This implies that $\sum_{\boldsymbol{\alpha}\in\mathcal{H}} A_{(l,\boldsymbol{\alpha})}$ is a $g(p,w) \times g(p,w)$ matrix of ones and zeros, and also that $\left\langle A_{(l,\boldsymbol{\alpha})}, A_{(l,\boldsymbol{\gamma})}\right\rangle = 0$ if $\boldsymbol{\alpha} \neq \boldsymbol{\gamma}$, from which we can conclude that $\sum_{\boldsymbol{\alpha}\in\mathcal{H}} \left\|A_{(l,\boldsymbol{\alpha})}\right\|^2 = \left\|\sum_{\boldsymbol{\alpha}\in\mathcal{H}} A_{(l,\boldsymbol{\alpha})}\right\|^2 \leq g(p,w)^2$. If $|z_{\boldsymbol{\alpha}}| \leq \zeta$, by using $\left\langle A_{(l,\boldsymbol{\alpha})}, A_{(l,\boldsymbol{\gamma})}\right\rangle = 0$ and the previous inequality it is not difficult to see that $\left\|\sum_{\boldsymbol{\alpha}\in\mathcal{H}} A_{(l,\boldsymbol{\alpha})} z_{\boldsymbol{\alpha}}\right\|^2 = \sum_{\boldsymbol{\alpha}\in\mathcal{H}} z_{\boldsymbol{\alpha}}^2 \left\|A_{(l,\boldsymbol{\alpha})}\right\|^2 \leq \zeta^2 \sum_{\boldsymbol{\alpha}\in\mathcal{H}} \left\|A_{(l,\boldsymbol{\alpha})}\right\|^2 \leq g(p,w)^2\zeta^2$.

(d) Note that for any $2 \leq l \leq n$, $B_{l-1} = \{\boldsymbol{\alpha}^- : \boldsymbol{\alpha} \in B_l\}$; then (d) follows by noticing that $\sum_{\boldsymbol{\alpha}\in B_{l-1}} A_{(l-1,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}$ is equal to $\sum_{\boldsymbol{\alpha}\in B_l} A_{(l-1,\boldsymbol{\alpha}^-)} y_{\boldsymbol{\alpha}^-}$, and then using (a).

(e) This equality is obtained by replacing $\mathcal{F}$ by $\cup_{k=1}^n B_k$ and then eliminating the zero matrices according to (b). ∎

5. Lower dimensional SDP relaxations. In this section, we will define fine and coarse level problems and relate their corresponding hierarchies of sparse SDP relaxations. The coarse level model has the same structure as the fine level model, but it has fewer dimensions. We will define prolongation operators for the primal and dual variables of the SDP relaxation. The aim of the prolongation operators is to transfer information from the lower dimensional coarse model to the high dimensional fine model. In the multigrid literature this operation is called prolongation, and we adopt the same terminology here. We will study the properties of these operators and establish theoretical results. In particular we will derive conditions that will guarantee that the prolongation solution is within ϵ of the true solution (where $\epsilon > 0$ is a user-specified parameter). The conditions only include information from the coarse model and thus are easy to compute in practice (Corollary 14). We show that when these conditions are satisfied for a low tolerance ϵ, the complexity of infeasible interior point methods is expected to improve (Theorem 15).

Consider the following problem for $0 \leq t \leq n_2 - n_1 - p$:

(10)
$$POP_t : \min_{\mathbf{x}^{n-t} \in \mathbb{R}^{n-t}} F_{n-t}(\mathbf{x}^{n-t}) \triangleq \sum_{k=1}^{n_1-1} f_k(\mathbf{x}_k^{n-t}) + \sum_{k=n_1}^{n_2-t} f_0(\mathbf{x}_k^{n-t}) + \sum_{k=n_2-t+1}^{n-t-p+1} f_{k+t}(\mathbf{x}_k^{n-t}).$$

Note that $t = 0$ corresponds to the original problem (which we call the *fine problem* or the *problem at the fine level*); models for $t \geq 1$ are lower dimensional problems. We will refer to lower dimensional models as *coarse problems* or *problems at the coarse level*. Let $y^{n-t} = \{y_{\boldsymbol{\alpha}}^{n-t}\}_{\boldsymbol{\alpha} \in \mathcal{F}^{n-t}}$, $S^{n-t} = \left(S_1^{n-t}, S_2^{n-t}, \ldots, S_{n-t-p+1}^{n-t}\right)$, and $X^{n-t} = \left(X_1^{n-t}, X_2^{n-t}, \ldots, X_{n-t-p+1}^{n-t}\right)$ for $0 \leq t \leq n_2 - n_1 - p$ be variables in the coarse levels ($t \geq 1$) and fine level ($t = 0$) spaces. If the order of the relaxation w is fixed, a sparse SDP relaxation using different values of t can be constructed,

(11)
$$SDP_t : \begin{cases} \min\limits_{y^{n-t}, S^{n-t}} \sum\limits_{\boldsymbol{\alpha} \in \mathcal{F}^{n-t}} b_{\boldsymbol{\alpha}}^{n-t} y_{\boldsymbol{\alpha}}^{n-t} \\ \text{s.t.} \sum\limits_{\boldsymbol{\alpha} \in \mathcal{F}^{n-t}} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-t} + C = S_l^{n-t}, \\ \quad S_l^{n-t} \succeq 0,\ l = 1, 2, \ldots, n-t-p+1, \end{cases}$$

where $\mathcal{F}^{n-t} = \cup_{l=1}^{n-t+p-1} \mathbb{A}_{2w}^{\Phi_l} \setminus \{\mathbf{0}\}$ and $0 \leq t \leq n_2 - n_1 - p$. Likewise the dual SDP is

(12)
$$SDP_t^* : \begin{cases} \max\limits_{X^{n-t}} - \sum\limits_{l=1}^{n-t-p+1} \left\langle C, X_l^{n-t} \right\rangle \\ \text{s.t.} \sum\limits_{l=1}^{n-t-p+1} \left\langle A_{(l,\boldsymbol{\alpha})}, X_k^{n-t} \right\rangle = b_{\boldsymbol{\alpha}}^{n-t} \text{ for } \boldsymbol{\alpha} \in \mathcal{F}^{n-t}, \\ \quad X_l^{n-t} \succeq 0,\ l = 1, 2, \ldots, n-t-p+1. \end{cases}$$

Note that all the properties in Lemma 7 are still valid for any fixed t (the underlying POP has the same structure as (1)), and the sets Φ_l are the same as in Lemma 5. The next example illustrates other important properties relating the coefficients of the monomials for the different relaxations at different levels (coarse and fine). Lemma 9 formalizes these properties.

Example 8. Let $F_{n-t}(\mathbf{x}) = \sum_{k=1}^{n-t-1} (x_k^{n-t} - x_{k+1}^{n-t})^2$ (in this case $n_1 = 1$, $n_2 = 4$). If $n = 5$, consider the functions at levels $t = 0$ and $t = 1$ (where the superscript $n - t$

was dropped for simplicity):

$$t=0:\sum_{k=1}^{4}(x_k-x_{k+1})^2=(x_1^2-2x_1x_2)+\left(\sum_{k=2}^{4}2x_k^2-2x_kx_{k+1}\right)+(x_5^2),$$
$$t=1:\sum_{k=1}^{3}(x_k-x_{k+1})^2=(x_1^2-2x_1x_2)+\left(\sum_{k=2}^{3}2x_k^2-2x_kx_{k+1}\right)+(x_4^2).$$

Note that the monomials $x_i^{\alpha_i}x_{i+1}^{\alpha_{i+1}}$ have the same coefficients in both levels for $i=1,2,3$, and the coefficient of the monomial x_4^2 at level $t=1$ is the same as the coefficient of x_5^2 for the level $t=0$.

LEMMA 9. *For any $\bar{t}$ such that $0\le\bar{t}\le n_2-n_1-p$, the SDP models in* (11) *and* (12) *satisfy the following:*

(a) *For any $i\in\{1,2,\dots,n_2-\bar{t}\}$, if $\alpha_i>0$, then $b^n_{\boldsymbol{\alpha}}=b^{n-\bar{t}}_{\boldsymbol{\alpha}}$.*

(b) *For any $i\in\{n_2,n_2+1,\dots,n-\bar{t}-p+1\}$, if $\alpha_i>0$, then $b^{n-\bar{t}}_{\boldsymbol{\alpha}}=b^{n-(\bar{t}+1)}_{\boldsymbol{\alpha}^-}$.*

(c) *For any $i\in\{n_1+p,n_1+p+1,\dots,n_2-\bar{t}-1\}$, if $\alpha_i>0$, then $b^{n-\bar{t}}_{\boldsymbol{\alpha}}=b^{n-\bar{t}}_{\boldsymbol{\alpha}^-}$.*

Proof. (a) Let $\alpha_i>0$ for some $i\in\{1,2,\dots,n_2-\bar{t}\}$. Then if $l\ge n_2-\bar{t}+1$, $f_k(\mathbf{x}_l)$ does not contain the monomial $\mathbf{x}^{\boldsymbol{\alpha}}$ for any k. Using (10) with $t=0$ and $t=\bar{t}$, we have that the coefficients $b^n_{\boldsymbol{\alpha}}$ and $b^{n-\bar{t}}_{\boldsymbol{\alpha}}$ are determined by the functions $\sum_{k=1}^{n_1-1}f_k(\mathbf{x}_k^n)+\sum_{k=n_1}^{n_2-\bar{t}}f_0(\mathbf{x}_k^n)$ and $\sum_{k=1}^{n_1-1}f_k(\mathbf{x}_k^{n-\bar{t}})+\sum_{k=n_1}^{n_2-\bar{t}}f_0(\mathbf{x}_k^{n-\bar{t}})$, respectively, from where the result follows.

(b) Let $\alpha_i>0$ for some $i\in\{n_2,n_2+1,\dots,n-\bar{t}-p+1\}$. Then if $l\le n_2-p$, $f_k(\mathbf{x}_l)$ does not contain the monomial $\mathbf{x}^{\boldsymbol{\alpha}}$ for any k. Setting $t=\bar{t}$ in (10), we observe that the coefficient $b^{n-\bar{t}}_{\boldsymbol{\alpha}}$ is determined by the function $\sum_{k=n_2-p+1}^{n_2-\bar{t}}f_0(\mathbf{x}_k^{n-\bar{t}})+\sum_{k=n_2-\bar{t}+1}^{n-\bar{t}-p+1}f_{k+\bar{t}}(\mathbf{x}_k^{n-\bar{t}})$. Likewise, $\boldsymbol{\alpha}^-$ is positive in position $i-1$ because $\alpha_i>0$; therefore if $l\le n_2-p-1$, $f_k(\mathbf{x}_l)$ does not contain the monomial $\mathbf{x}^{\boldsymbol{\alpha}}$ for any k. Then, setting $t=\bar{t}+1$ in (10), it can be deduced that the coefficient $b^{n-(\bar{t}+1)}_{\boldsymbol{\alpha}^-}$ is determined by the function $\sum_{k=n_2-p+1}^{n_2-\bar{t}}f_0(\mathbf{x}_{k-1}^{n-\bar{t}-1})+\sum_{k=n_2-\bar{t}+1}^{n-\bar{t}-p+1}f_{k+\bar{t}}(\mathbf{x}_{k-1}^{n-\bar{t}-1})$, from where the statement follows.

(c) Similarly to (b), the statement in (c) follows by noticing that $b^{n-\bar{t}}_{\boldsymbol{\alpha}}$ and $b^{n-\bar{t}}_{\boldsymbol{\alpha}^-}$ are determined by the functions $\sum_{k=n_1+1}^{n_2-\bar{t}-1}f_0(\mathbf{x}_k^{n-\bar{t}})$ and $\sum_{k=n_1+1}^{n_2-\bar{t}-1}f_0(\mathbf{x}_{k-1}^{n-\bar{t}})$, respectively. ∎

5.1. Analysis for models F_n and F_{n-1}. In this subsection we will analyze the relation between levels n and $n-1$ ($t=0$ and $t=1$). The goal is to define prolongation operators that allow the transformation of any point in the coarse level semidefinite program into the fine level semidefinite program. It is not computationally advantageous to consider a coarse model with $n-1$ components since the dimensionality reduction is too small. In section 6 we will show how to use the operators obtained from this simple case in a recursive manner to obtain much coarser models with, for example, $\lceil n/2\rceil$ components.

Action of the primal prolongation operators on the coarse primal variables. We denote the prolongation operators from the coarse level by P_y and P_S for the variables y^{n-1} and S^{n-1}, respectively. If $y^n=P_y\left(y^{n-1}\right)$ and $S^n=P_S\left(S^{n-1}\right)$,

then these operators prolongate the variables to the fine level following (13) and (14):

$$y_{\boldsymbol{\alpha}}^n = \begin{cases} y_{\boldsymbol{\alpha}}^{n-1} & \text{if } \boldsymbol{\alpha} \in \cup_{j=1}^{i_0-1} B_j, \\ 0.5\left(y_{\boldsymbol{\alpha}^-}^{n-1} + y_{\boldsymbol{\alpha}}^{n-1}\right) & \text{if } \boldsymbol{\alpha} \in B_{i_0}, \\ y_{\boldsymbol{\alpha}^-}^{n-1} & \text{if } \boldsymbol{\alpha} \in \cup_{j=i_0+1}^{n} B_j, \end{cases} \tag{13}$$

$$S_l^n = \begin{cases} S_l^{n-1} & \text{if } l \in \{1, 2, \ldots, i_0 - 1\}, \\ 0.5\left(S_{l-1}^{n-1} + S_l^{n-1}\right) & \text{if } l = i_0, \\ S_{l-1}^{n-1} & \text{if } l \in \{i_0 + 1, i_0 + 2, \ldots, n - p + 1\}, \end{cases} \tag{14}$$

where $2 \leq i_0 \leq n - p - 1$. These operators are linear and depend on an integer $i_0 \in \{2, 3, \ldots, n - p + 1\}$. This means that we can construct different operators by selecting different values of i_0. To gain intuition on how the primal operators work, note that the fine primal semidefinite program has one more matrix variable than the coarse primal semidefinite program. Suppose that the additional matrix at the fine level is the i_0th matrix. Then P_S computes the i_0th matrix as an average of the i_0th and $(i_0 - 1)$th coarse matrices. Likewise, the operator P_y assumes that the additional variables $y_{\boldsymbol{\alpha}}^n$ for the fine primal relaxation correspond to the monomials $\mathbf{x}^{\boldsymbol{\alpha}}$ with $\boldsymbol{\alpha} \in B_{i_0}$. Obviously, the operators depend on the choice of i_0, and we discuss how to choose i_0 in practice in subsection 6.1.

For any feasible set for the coarse problems, the following theorem characterizes the feasibility of the prolongated primal variables at the fine level. For $\left(y^i, S^i, X^i\right)$ we define the primal (R_l^i) and dual residuals $(r_{\boldsymbol{\alpha}}^i)$ as

$$R_l^i \triangleq \sum_{\boldsymbol{\alpha} \in \mathcal{F}^i} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^i + C - S_l^i, \quad r_{\boldsymbol{\alpha}}^i \triangleq \sum_{k=1}^{i-p+1} \left\langle A_{(k,\boldsymbol{\alpha})}, X_k^i \right\rangle - b_{\boldsymbol{\alpha}}^i \tag{15}$$

for $l = 1, 2, \ldots, i - p + 1$ and $\boldsymbol{\alpha} \in \mathcal{F}^i$.

THEOREM 10. *Let $\left(y^{n-1}, S^{n-1}\right)$ be feasible points for the coarse primal problem (11) for $t = 1$. If $y^n = P_y(y^{n-1})$, $S^n = P_S\left(S^{n-1}\right)$ are defined according to (13) and (14), respectively, for some $2 \leq i_0 \leq n - p - 1$, then*

(a) *$S_l^n \succeq 0$ for $l = 1, 2, \ldots, n - p + 1$.*

(b) *Let $\epsilon_1 = \max\left\{\left|y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1}\right|\right\}_{\boldsymbol{\alpha} \in \cup_{k=i_0}^{i_0+p-1} B_k}$, then*

$$\|R_l^n\| \leq \begin{cases} 0 & \text{if } l \leq i_0 - p \text{ or } l \geq i_0 + 1, \\ g(p, w)\epsilon_1 & \text{if } i_0 - p + 1 \leq l \leq i_0, \end{cases}$$

where R_l^n is the residual matrix defined in (15).

Proof. (a) S_l^n is positive semidefinite for $l = 1, 2, \ldots, n - p + 1$ because the point S^{n-1} is feasible in the coarse relaxation, and hence the coarse matrices are positive semidefinite.

(b) To calculate the feasibility of the primal fine problem, we use Lemma 7(e) to write the feasibility constraints of the $n - t$ relaxation as

$$\sum_{\boldsymbol{\alpha} \in \mathcal{F}^{n-t}} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-t} + C - S_l^{n-t} = \sum_{\boldsymbol{\alpha} \in \cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-t} + C - S_l^{n-t}, \tag{16}$$

where $1 \leq l \leq n - t - p + 1$.

Also, using Lemma 7(d) we have

$$\sum_{\boldsymbol{\alpha}\in B_{k-1}} A_{(k-1,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-t} = \sum_{\boldsymbol{\alpha}\in B_k} A_{(k,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}^-}^{n-t} \quad \text{for } 0 \le t \le n_2 - n_1 - p. \tag{17}$$

We can now evaluate five different cases for the residual constraints R_l^n. We will only show the details of two of these cases which illustrate how to use the properties of the problem; the remaining three cases can be proved using a similar argument. Given that y^{n-1} and S^{n-1} are feasible points for the coarse primal relaxation, when the variable S_l^{n-1} appears it will be replaced according to the constraints in (11) for $t = 1$, and the variables $y_{\boldsymbol{\alpha}}^n$ and S_l^n will be replaced by the operators (13) and (14), respectively.

Case 1. $l = 1, 2, \ldots, i_0 - p$:

$$\begin{aligned} R_l^n &= \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^n + C - S_l^n \\ &= \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-1} + C - S_l^{n-1} \\ &= \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-1} + C - \left(\sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-1} + C \right) \\ &= 0. \end{aligned}$$

Case 2. $l = i_0 + 1, i_0 + 2, \ldots, n - p + 1$:

$$\begin{aligned} R_l^n &= \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^n + C - S_l^n \\ &= \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}^-}^{n-1} + C - S_{l-1}^{n-1} \\ &= \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}^-}^{n-1} + C - \left(\sum_{\boldsymbol{\alpha}\in\cup_{k=l-1}^{l+p-2} B_k} A_{(l-1,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-1} + C \right) \\ &= \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}^-}^{n-1} - \sum_{\boldsymbol{\alpha}\in\cup_{k=l}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}^-}^{n-1} \\ &= 0, \end{aligned}$$

where (17) was used to replace $\sum_{\boldsymbol{\alpha}\in\cup_{k=l-1}^{l+p-2} B_k} A_{(l-1,\boldsymbol{\alpha})} y_{\boldsymbol{\alpha}}^{n-1}$ to go from the third to the fourth equality.

Case 3. $l = i_0$:

$$R_l^n = \sum_{\boldsymbol{\alpha}\in\cup_{k=i_0+1}^{i_0+p-1} B_k} A_{(i_0,\boldsymbol{\alpha})} 0.5 \left(y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1} \right).$$

Case 4. $l = i_0 - p + 2, \ldots, i_0 - 1$:

$$R_l^n = \sum_{\boldsymbol{\alpha}\in B_{i_0}} A_{(l,\boldsymbol{\alpha})} 0.5 \left(y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1} \right) + \sum_{\boldsymbol{\alpha}\in\cup_{k=i_0+1}^{l+p-1} B_k} A_{(l,\boldsymbol{\alpha})} \left(y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1} \right).$$

Case 5. $l = i_0 - p + 1$:

$$R_l^n = \sum_{\boldsymbol{\alpha} \in B_{i_0}} A_{(l,\boldsymbol{\alpha})} 0.5 \left(y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1} \right).$$

Then, if $|y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1}| \leq \epsilon_1$ for $\boldsymbol{\alpha} \in \cup_{k=i_0}^{i_0+p-1} B_k$, using Lemma 7(c) we can conclude from the previous five cases that the norm of the constraints of the fine primal relaxation are either zero or smaller than $g(p,w)\epsilon_1$. ∎

We note that if $\epsilon_1 \to 0$, then $\|R_l^n\| \to 0$. We also note that ϵ_1 is easy to calculate from coarse information only.

Action of the dual prolongation operators on the coarse dual variables. We now turn our attention to the relationship between the coarse and fine dual variables. As the fine dual relaxation contains one more matrix variable than the coarse relaxation, the operator works in the same fashion as P_S for the primal case (i.e., the additional matrix is calculated as an average of the coarse matrices). We perform an analysis similar to that in the primal case.

Let P_X be the prolongation operator for the variable X^{n-1}. As in the case of the primal operators, the dual prolongation will depend on an integer j_0, which allows us to define different operators. If $X^n = P_X\left(X^{n-1}\right)$, then

$$X_l^n = \begin{cases} X_l^{n-1} & \text{if } l \in \{1, 2, \ldots, j_0 - 1\}, \\ 0.5\left(X_{l-1}^{n-1} + X_l^{n-1}\right) & \text{if } l = j_0, \\ X_{l-1}^{n-1} & \text{if } l \in \{j_0 + 1, j_0 + 2, \ldots, n - p + 1\}, \end{cases} \tag{18}$$

where $n_1 + p + 1 \leq j_0 \leq n_2 - 2$. The feasibility of the dual prolongation is proven below.

THEOREM 11. *Let X^{n-1} be a feasible point for the coarse dual problem* (12) *for $t = 1$. If $X^n = P_X\left(X^{n-1}\right)$ is defined according to* (18) *for some $n_1 + p + 1 \leq j_0 \leq n_2 - 2$, then*

(a) *$X_l^n \succeq 0$ for $l = 1, 2, \ldots, n - p + 1$.*

(b) *Let $\epsilon_2 = \max\left\{\left\|X_{l-1}^{n-1} - X_l^{n-1}\right\|\right\}_{l=j_0}^{j_0+p-1}$, then for any $\boldsymbol{\alpha} \in \mathcal{F}^n$,*

$$|r_{\boldsymbol{\alpha}}^n| \leq \begin{cases} 0 & \text{if } \boldsymbol{\alpha} \in B_l, \text{ and } l \leq j_0 - 1 \text{ or } l \geq j_0 + p, \\ g(p,w)p\epsilon_2 & \text{if } \boldsymbol{\alpha} \in B_l, \text{ and } j_0 \leq l \leq j_0 + p - 1, \end{cases}$$

where $r_{\boldsymbol{\alpha}}^n$ is the residual defined in (15).

Proof. (a) As in the primal case, the matrices in $P_X\left(X^{n-1}\right)$ are positive semidefinite because they are a linear combination of positive semidefinite matrices (X^{n-1} is feasible for the coarse dual relaxation).

(b) Once more we will divide the proof into different cases to calculate the dual residual for $r_{\boldsymbol{\alpha}}^n$ ($\boldsymbol{\alpha} \in \mathcal{F}^n$), and we will make use of the following fact. If $\boldsymbol{\alpha} \in B_l$, then a consequence of Lemma 7(b) is

$$\sum_{k=1}^{n-t-p+1} \left\langle A_{(k,\boldsymbol{\alpha})}, X_k^{n-t} \right\rangle = \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_k^{n-t} \right\rangle. \tag{19}$$

Given that X^{n-1} is feasible for the coarse dual relaxation, when the variable $b_{\boldsymbol{\alpha}}^{n-1}$ appears it will be replaced according to the constraints in (12) for $t = 1$, and the

variable X_l^n will be replaced by the operator (18). We will show the details for two of the five cases to illustrate how to use the properties of the problem; the remaining cases can be deduced following a similar procedure.

Case 1. $\boldsymbol{\alpha} \in B_l$ and $l = 1, 2, \ldots, j_0 - 1$:

$$\begin{aligned} r_{\boldsymbol{\alpha}}^n &= \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_k^n \right\rangle - b_{\boldsymbol{\alpha}}^n \\ &= \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_k^{n-1} \right\rangle - b_{\boldsymbol{\alpha}}^{n-1} \\ &= \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_k^{n-1} \right\rangle - \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_k^{n-1} \right\rangle \\ &= 0, \end{aligned}$$

where $b_{\boldsymbol{\alpha}}^n$ was replaced by $b_{\boldsymbol{\alpha}}^{n-1}$ using Lemma 9(a) with $\bar{t} = 0$ (note that we can use Lemma 9 because $\alpha_l > 0$ by the definition of B_l).

Case 2. $\boldsymbol{\alpha} \in B_l$ and $l = n_2, n_2 + 1, \ldots, n$:

$$\begin{aligned} r_{\boldsymbol{\alpha}}^n &= \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_k^n \right\rangle - b_{\boldsymbol{\alpha}}^n \\ &= \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_{k-1}^{n-1} \right\rangle - b_{\boldsymbol{\alpha}^-}^{n-1} \\ &= \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_{k-1}^{n-1} \right\rangle - \sum_{k=l-p}^{l-1} \left\langle A_{(k,\boldsymbol{\alpha}^-)}, X_k^{n-1} \right\rangle \\ &= \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_{k-1}^{n-1} \right\rangle - \sum_{k=l-p+1}^{l} \left\langle A_{(k-1,\boldsymbol{\alpha}^-)}, X_{k-1}^{n-1} \right\rangle \\ &= \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_{k-1}^{n-1} \right\rangle - \sum_{k=l-p+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_{k-1}^{n-1} \right\rangle \\ &= 0, \end{aligned}$$

where we used Lemma 9(b) with $\bar{t} = 0$ to substitute $b_{\boldsymbol{\alpha}}^n$ for $b_{\boldsymbol{\alpha}^-}^{n-1}$, and used Lemma 7(a) to substitute $A_{(k-1,\boldsymbol{\alpha}^-)}$ for $A_{(k,\boldsymbol{\alpha})}$ in the last equation.

Case 3. $\boldsymbol{\alpha} \in B_l$ and $l = \min\{j_0 + p - 1, n_2 - 1\} + 1, j_0 + p + 1, \ldots, n_2 - 1$:

$$r_{\boldsymbol{\alpha}}^n = 0.$$

Case 4. $\boldsymbol{\alpha} \in B_l$ and $l = j_0 + 1, \ldots, \min\{j_0 + p - 1, n_2 - 1\}$:

$$r_{\boldsymbol{\alpha}}^n = \left\langle A_{(j_0,\boldsymbol{\alpha})}, 0.5\left(X_{j_0-1}^{n-1} - X_{j_0}^{n-1}\right)\right\rangle + \sum_{k=j_0+1}^{l} \left\langle A_{(k,\boldsymbol{\alpha})}, X_{k-1}^{n-1} - X_k^{n-1} \right\rangle.$$

Case 5. $\boldsymbol{\alpha} \in B_l$ for $l = j_0$:

$$r_{\boldsymbol{\alpha}}^n = \left\langle A_{(j_0,\boldsymbol{\alpha})}, 0.5\left(X_{j_0-1}^{n-1} - X_{j_0}^{n-1}\right)\right\rangle.$$

Note that Case 3 might not be needed for all problems, for example, if $n = 5$, $n_1 = 1$, $n_2 = 4$, $p = 2$, and $j_0 = 3$.

To find the upper bound of the constraints that are not equal to zero (Cases 4 and 5), let $Z_k \in \mathbb{R}^{g(p,w)\times g(p,w)}$ be such that $\|Z_k\| \leq \epsilon$. Note that $A_{(k,\boldsymbol{\alpha})}$ is a matrix of zeros and ones with at most $g(p,w)^2 - 1$ elements different from zero. Then, using the triangle and Cauchy–Bunyakovskii–Schwarz inequalities, we observe that for any $m_1, m_2 \in \mathbb{N}$ with $1 \leq m_1 \leq m_2 \leq n - p + 1$,

$$\left| \sum_{k=m_1}^{m_2} \left\langle A_{(k,\boldsymbol{\alpha})}, Z_k \right\rangle \right| \leq (m_2 - m_1 + 1) g(p,w) \epsilon.$$

Note that for the cases where the constraints are not zero, the number of terms in the summation does not exceed p (in Case 4 if $l = \min\{j_0 + p - 1, n_2 - 1\} = j_0 + p - 1$, we have $m_1 = j_0$ and $m_2 = j_0 + p - 1$). Then, if $\left\|X_{k-1}^{n-1} - X_k^{n-1}\right\| \leq \epsilon_2$ for $l = j_0, j_0 + 1, \ldots, j_0 + p - 1$, and $\boldsymbol{\alpha} \in \mathcal{F}^n$, using the previous inequality with $\epsilon = \epsilon_1$ and $Z_k = X_{k-1}^{n-1} - X_k^{n-1}$, it is easy to see that the constraints that are not zero are less than $g(p,w)p\epsilon_2$. ∎

As in the primal case, we note that if $\epsilon_2 \to 0$, then $|r_{\boldsymbol{\alpha}}^n| \to 0$, and ϵ_2 is easy to calculate from coarse information only.

Note that the primal prolongations were defined for some i_0 (with $2 \leq i_0 \leq n - p - 1$), and the dual prolongation were defined for some j_0 (with $n_1 + p + 1 \leq j_0 \leq n_2 - 2$), but these numbers do not need to be the same. The constant j_0 for the dual prolongation has to be selected from a bounded set that depends on the variables n_1 and n_2, but i_0 does not. This is due to the fact that the constraints of the primal relaxation depend only on the sets Φ_l (which do not depend on any particular value of n_1 or n_2), while the constraints of dual relaxation depend directly on the coefficients of F_n, which, given the structure of problem (1), are a function of n_1 and n_2. The case $i_0 = j_0$ characterized the duality gap, leading to the following theorem (Theorem 12).

THEOREM 12. *Let* $\left(y^{n-1}, S^{n-1}, X^{n-1}\right)$ *be feasible points for the coarse primal and dual problems* (11) *and* (12) *for* $t = 1$. *If* $y^n = P_y\left(y^{n-1}\right)$, $S^n = P_S\left(S^{n-1}\right)$, $X^n = P_X\left(X^{n-1}\right)$ *are defined as in* (13), (14), *and* (18) *for some* $n_1 + p + 1 \leq i_0 \leq n_2 - 2$ *and* $j_0 = i_0$, *then* $\left\|S_{i_0-1}^{n-1} - S_{i_0}^{n-1}\right\| \leq g(p,w)\epsilon_1$ *and*

$$\sum_{k=1}^{n-p+1} \langle X_k^n, S_k^n \rangle \leq \sum_{k=1}^{n-p} \left\langle X_k^{n-1}, S_k^{n-1} \right\rangle + 0.5\left(\left\langle X_{i_0}^{n-1}, S_{i_0}^{n-1} \right\rangle + \left\langle X_{i_0-1}^{n-1}, S_{i_0-1}^{n-1} \right\rangle\right) + g(p,w)\epsilon_1\epsilon_2, \tag{20}$$

where ϵ_1, ϵ_2 *are defined in Theorem* 10 *and Theorem* 11, *respectively.*

Proof. Using the constraints in (11) for $t = 1$, and properties (e) (with $l = i_0$) and (d) of Lemma 7, we can deduce that

$$S_{i_0-1}^{n-1} - S_{i_0}^{n-1} = \sum_{\boldsymbol{\alpha} \in \cup_{k=i_0}^{i_0+p-1} B_k} A_{(i_0,\boldsymbol{\alpha})}\left(y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1}\right).$$

Using the definition of ϵ_1 in Theorem 10 and Lemma 7(c) with $\mathcal{H} = \cup_{k=i_0}^{i_0+p-1} B_k$, $z_{\boldsymbol{\alpha}} = y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1}$, and $\zeta = \epsilon_1$, we conclude that $\left\|S_{i_0-1}^{n-1} - S_{i_0}^{n-1}\right\| \leq g(p,w)\epsilon_1$.

Let $E_{i_0}^S = S_{i_0-1}^{n-1} - S_{i_0}^{n-1}$ and $E_{i_0}^X = X_{i_0-1}^{n-1} - X_{i_0}^{n-1}$. Replacing X^n by $P_X\left(X^{n-1}\right)$ and S^n by $P_S\left(S^{n-1}\right)$, and using the upper bound for $\|S_{i_0-1}^{n-1} - S_{i_0}^{n-1}\|$ and the conditions of part (b) of Theorem 11 ($\|X_{i_0-1}^{n-1} - X_{i_0}^{n-1}\| \le \epsilon_2$), we obtain

$$\begin{aligned}
\sum_{k=1}^{n-p+1} \langle X_k^n, S_k^n\rangle &= \sum_{k=1}^{i_0-1} \left\langle X_k^{n-1}, S_k^{n-1}\right\rangle + \sum_{k=i_0+1}^{n-p+1} \left\langle X_{k-1}^{n-1}, S_{k-1}^{n-1}\right\rangle \\
&\quad + \sum_{k=i_0}^{i_0} \left\langle 0.5\left(X_{k-1}^{n-1} + X_k^{n-1}\right), 0.5\left(S_{k-1}^{n-1} + S_k^{n-1}\right)\right\rangle \\
&= \sum_{k=1}^{n-p} \left\langle X_k^{n-1}, S_k^{n-1}\right\rangle + 0.5\left(\left\langle X_{i_0}^{n-1}, S_{i_0}^{n-1}\right\rangle + \left\langle X_{i_0-1}^{n-1}, S_{i_0-1}^{n-1}\right\rangle\right) \\
&\quad - 0.25\left\langle E_{i_0}^X, E_{i_0}^S\right\rangle \\
&\le \sum_{k=1}^{n-p} \left\langle X_k^{n-1}, S_k^{n-1}\right\rangle + 0.5\left(\left\langle X_{i_0}^{n-1}, S_{i_0}^{n-1}\right\rangle + \left\langle X_{i_0-1}^{n-1}, S_{i_0-1}^{n-1}\right\rangle\right) \\
&\quad + \left|\left\langle E_{i_0}^X, E_{i_0}^S\right\rangle\right| \\
&\le \sum_{k=1}^{n-p} \left\langle X_k^{n-1}, S_k^{n-1}\right\rangle + 0.5\left(\left\langle X_{i_0}^{n-1}, S_{i_0}^{n-1}\right\rangle + \left\langle X_{i_0-1}^{n-1}, S_{i_0-1}^{n-1}\right\rangle\right) \\
&\quad + g(p,w)\epsilon_1\epsilon_2.
\end{aligned}$$

To obtain the last inequality, we used the inequality $\left|\left\langle E_{i_0}^X, E_{i_0}^S\right\rangle\right| \le \left\|E_{i_0}^X\right\| \left\|E_{i_0}^S\right\|$ (Cauchy–Bunyakovskii–Schwarz inequality) to bound $\left|\left\langle E_{i_0}^X, E_{i_0}^S\right\rangle\right|$ by $g(p,w)\epsilon_1\epsilon_2$. ∎

COROLLARY 13. *Under the assumptions of Theorem* 12*, if* $\left(y^{n-1}, S^{n-1}, X^{n-1}\right)$ *is feasible and* $\mu = \sum_{k=1}^{n-p} \left\langle X_k^{n-1}, S_k^{n-1}\right\rangle / ((n-p)g(p,w))$*, then*

$$\sum_{k=1}^{n-p+1} \langle X_k^n, S_k^n\rangle \le 2g(p,w)(n-p)\mu + g(p,w)\epsilon_1\epsilon_2. \tag{21}$$

All the bounds calculated so far in this section depend on the terms ϵ_1 and ϵ_2 defined in Theorem 10 and Theorem 11. It is straightforward to see that if the goal is to obtain a feasible point (y^n, S^n, X^n) for the fine problem, it would be enough to have a feasible coarse point such that ϵ_1 and ϵ_2 are zero. In the next corollary, we formalize this idea by giving conditions for a coarse point $(y^{n-1}, S^{n-1}, X^{n-1})$ to provide a prolongated point that is ϵ-optimal.

COROLLARY 14. *Let* $(y^{n-1}, S^{n-1}, X^{n-1})$ *be feasible points for the coarse primal and dual problems* (11) *and* (12) *(*$t = 1$*), and let* $\epsilon \in \mathbb{R}$ *be a nonnegative scalar.*

(a) *If there exist* i_0*,* $j_0 \in \mathbb{N}$ *with* $2 \le i_0 \le n-p-1$ *and* $n_1 + p + 1 \le j_0 \le n_2 - 2$ *such that*

$$\begin{aligned}
\epsilon_1 &= \max\left\{\left|y_{\boldsymbol{\alpha}^-}^{n-1} - y_{\boldsymbol{\alpha}}^{n-1}\right|\right\}_{\boldsymbol{\alpha}\in\cup_{k=i_0}^{i_0+p-1} B_k} \le \frac{\epsilon}{g(p,w)}, \\
\epsilon_2 &= \max\left\{\left\|X_{l-1}^{n-1} - X_l^{n-1}\right\|\right\}_{l=j_0}^{j_0+p-1} \le \frac{\epsilon}{g(p,w)p},
\end{aligned} \tag{22}$$

then it is possible to prolongate the coarse variables using (13)*,* (14)*, and* (18) *to obtain* (y^n, S^n, X^n) *such that* $|r_{\boldsymbol{\alpha}}^n| \le \epsilon$ *(*$\boldsymbol{\alpha} \in \mathcal{F}^n$*) and* $\|R_l^n\| \le \epsilon$ *(*$l = 1, 2, \ldots, n-p+1$*).*

(b) *If in addition* $i_0 = j_0$ *and* $(y^{n-1}, S^{n-1}, X^{n-1})$ *are optimal points with zero duality gap for the coarse problem, then* $\sum_{k=1}^{n-p+1} \langle X_k^n, S_k^n \rangle \leq \epsilon^2/(g(p,w)p)$.

5.2. Exploiting multigrid structure in infeasible interior point methods (IPM). The complexity (in terms of number of iterations) of infeasible IPMs depends on the feasibility of the initial points and the associated duality gap. In light of the results of Theorem 10, Theorem 11, and Theorem 12, it is reasonable to expect that if a solution of the coarse level is prolongated and used as an initial point to solve the fine level model using an infeasible IPM, then the complexity will depend again on the variables ϵ_1 and ϵ_2 defined in Theorem 10, Theorem 11, and the duality gap of the coarse solution. In this subsection we will use the results of the algorithm proposed in [31] to show that its complexity can be improved as long as the values of ϵ_1, ϵ_2 and the coarse duality gap are small. Thus the proposed approach is reminiscent of one-way multigrid methods; i.e., we start at the bottom with the coarsest model, and then use the solution of the coarse model to initialize the solution of the model one level up. Our results in the next section will show that this approach can yield significant benefits.

If the infeasible IPM proposed in [31] is used to solve the SDP relaxation at level $t = 0$ with feasible or near feasible starting points $\left(y^{n,0}, S^{n,0}, X^{n,0}\right) \in \mathcal{N}(\gamma, \tau^0)$, then it will terminate in at most $\mathcal{O}(\sqrt{N}\ln(\epsilon_0/\epsilon))$ iterations (see Theorem 3.7 in [31]), where ϵ is the user-specified solution accuracy, $N = g(p,w)(n-p+1)$, and

$$\epsilon_0 \triangleq \max\left\{ \sum_{k=1}^{n-p+1} \left\langle X_k^{n,0}, S_k^{n,0} \right\rangle, \left(\sum_{l=1}^{n-p+1} \left\| R_l^{n,0} \right\|^2 \right)^{1/2}, \left\{ \left| r_{\boldsymbol{\alpha}}^{n,0} \right| \right\}_{\boldsymbol{\alpha} \in \mathcal{F}^n} \right\}, \tag{23}$$

$$\mathcal{N}(\gamma, \tau^0) \triangleq \left\{ (y^n, S^n, X^n) : S^n \succ 0, X^n \succ 0, \rho(X^n, S^n, \tau^0)^{1/2} \leq \gamma \tau^0 \right\}, \tag{24}$$

$$\rho(X^n, S^n, \tau^0) \triangleq \sum_{k=1}^{n-p+1} \sum_{i=1}^{g(p,w)} \left(\lambda_i \left(X_k^n S_k^n \right) - \tau^0 \right)^2, \tag{25}$$

with $0 < \gamma < 1$, $\tau^0 = \frac{1}{N} \sum_{k=1}^{n-p+1} \langle X_k^{n,0}, S_k^{n,0} \rangle$.

Let $\left(y^{n-1}, S^{n-1}, X^{n-1}\right)$ be a feasible point for the coarse problem, and use (13), (14), and (18) with $i_0 = j_0$ to calculate the fine level point $\left(y^{n,0}, S^{n,0}, X^{n,0}\right)$ as $y^{n,0} = P_y(y^{n-1})$, $S^{n,0} = P_S(S^{n-1})$, and $X^{n,0} = P_X(X^{n-1})$. Then using Theorem 10, Theorem 11, and Corollary 13, and setting $\mu = \sum_{k=1}^{n-p} \left\langle X_k^{n-1}, S_k^{n-1} \right\rangle / ((n-p)g(p,w))$, it is not difficult to see that

$$\epsilon_0 \leq \max\left\{ 2(N - g(p,w))\mu + g(p,w)\epsilon_1\epsilon_2, g(p,w)p^{1/2}\epsilon_1, g(p,w)p\epsilon_2 \right\}. \tag{26}$$

If $\left(y^{n,0}, S^{n,0}, X^{n,0}\right) \in \mathcal{N}(\gamma, \tau^0)$, the previous inequality shows how smaller values of μ, ϵ_1, and ϵ_2 can reduce the maximum number of iterations needed to achieve a solution with tolerance equal to ϵ (smaller values of μ will be expected if the initial coarse point is close to the coarse solution). Although it is not possible to guarantee that any prolongated solution of the coarse level will belong to $\mathcal{N}(\gamma, \tau^0)$ for some $\gamma \in (0,1)$, the next result shows that if ϵ_1 and ϵ_2 are small enough, and the coarse point is close to the infeasible central path of the coarse relaxation (i.e., close to the set of points $\{(X_k^{n-1}, S_k^{n-1})\}_{k=1}^{n-p}$ such that $X_k^{n-1} S_k^{n-1} = \mu I$), then $\left(y^{n,0}, S^{n,0}, X^{n,0}\right) \in \mathcal{N}(\gamma, \tau^0)$.

THEOREM 15. *Under the assumptions of Theorem* 12, *if* $X_k^{n-1} \succ 0$ *and* $S_k^{n-1} \succ 0$ *(*$k = 1, 2, \ldots, n-p$*) and* $\mu > 0$, *then* $\tau^0 > 0$ *and* $\rho(X^{n,0}, S^{n,0}, \tau^0)^{1/2} \leq \overline{w_1}\epsilon_1\epsilon_2 + \overline{w_2}\epsilon_\mu$,

where $\epsilon_\mu = \max\{\|X_k^{n-1}S_k^{n-1} - \mu I\|\}_{k=1}^{n-p}$, *and* $\overline{w_1}, \overline{w_2} < \infty$ *are constants that depend only on the parameters* n, p *and the order of the relaxation* w.

Proof. Replacing $S^{n,0}$ and $X^{n,0}$ by the prolongated coarse solutions and using $\sum_{k=1}^{n-p} \langle X_k^{n-1}, S_k^{n-1}\rangle = (N - g(p,w))\mu$, we obtain

$$\tau^0 = \frac{1}{N}\left((N - g(p,w))\mu + \left\langle X_{i_0}^{n,0}, S_{i_0}^{n,0}\right\rangle\right). \tag{27}$$

Note that $\lambda_i(X_{i_0}^{n,0}S_{i_0}^{n,0}) > 0$ for all i because $X_{i_0}^{n,0}$ and $S_{i_0}^{n,0}$ are positive definite (see Corollary 7.6.2 in [14]), and therefore $\langle X_{i_0}^{n,0}, S_{i_0}^{n,0}\rangle = \mathrm{Tr}(X_{i_0}^{n,0}S_{i_0}^{n,0}) = \sum_i \lambda_i(X_{i_0}^{n,0}S_{i_0}^{n,0}) > 0$. From this it follows that $\tau^0 > \frac{1}{N}(N - g(p,w))\mu > 0$.

To prove the bound for $\rho(X^{n,0}, S^{n,0}, \tau^0)$, first note that if $Q_k = X_k^{n,0}S_k^{n,0} - \mu I$, then $\|Q_k\| \le \epsilon_\mu$ for any $k \ne i_0$. If $k = i_0$, by replacing $X_{i_0}^{n,0}$ and $S_{i_0}^{n,0}$ by the prolongated coarse points we can write $Q_{i_0} = 0.5(Q_{i_0+1} + Q_{i_0-1}) - 0.25(X_{i_0-1}^{n-1} - X_{i_0}^{n-1})(S_{i_0-1}^{n-1} - S_{i_0}^{n-1})$ and therefore $\|Q_{i_0}\| \le \epsilon_\mu + g(p,w)\epsilon_1\epsilon_2$ (here we used the fact that under the assumptions of Theorem 12, $\|S_{i_0-1}^{n-1} - S_{i_0}^{n-1}\| \le g(p,w)\epsilon_1$ and $\|X_{i_0-1}^{n-1} - X_{i_0}^{n-1}\| \le \epsilon_2$). Also, using the Bauer–Fike theorem (see Theorem 6.3.2 in [14]) we can deduce that

$$|\lambda_i(X_k^{n,0}S_k^{n,0}) - \tau^0| = |\lambda_i(\mu I + Q_k) - \mu + (\mu - \tau^0)| \le g(p,w)\|Q_k\| + |\mu - \tau^0|. \tag{28}$$

Let $g = g(p,w)$. Using (28) and the bounds for $\|Q_k\|$, we have

$$\begin{aligned}
\rho(X^{n,0}, S^{n,0}, \tau^0) &= \sum_{k=1}^{n-p+1}\sum_{i=1}^{g}\left(\lambda_i\left(X_k^{n,0}S_k^{n,0}\right) - \tau^0\right)^2 \\
&\le \sum_{k=1}^{n-p+1}\sum_{i=1}^{g}\left(g\|Q_k\| + |\mu - \tau^0|\right)^2 \\
&\le \sum_{k=1}^{n-p}\sum_{i=1}^{g}\left(g\epsilon_\mu + |\mu - \tau^0|\right)^2 + \sum_{k=i_0}^{i_0}\sum_{i=1}^{g}\left(g(\epsilon_\mu + g\epsilon_1\epsilon_2) + |\mu - \tau^0|\right)^2 \\
&= (n-p)g\left(g\epsilon_\mu + |\mu - \tau^0|\right)^2 + g\left(g(\epsilon_\mu + g\epsilon_1\epsilon_2) + |\mu - \tau^0|\right)^2 \\
&\le (n-p+1)g\left(g(\epsilon_\mu + g\epsilon_1\epsilon_2) + |\mu - \tau^0|\right)^2 \\
&= N\left(g(\epsilon_\mu + g\epsilon_1\epsilon_2) + \frac{1}{N}\left|\mathrm{Tr}(Q_{i_0})\right|\right)^2 \\
&\le N\left(g(\epsilon_\mu + g\epsilon_1\epsilon_2) + \frac{1}{N}(g(\epsilon_\mu + g\epsilon_1\epsilon_2))\right)^2 \\
&= \left(\overline{w_1}\epsilon_1\epsilon_2 + \overline{w_2}\epsilon_\mu\right)^2,
\end{aligned}$$

where $\overline{w_1} = \frac{N+1}{N^{1/2}}g^2$, $\overline{w_2} = \frac{N+1}{N^{1/2}}g$, and (27) was used to replace $\mu - \tau^0$ by $\mathrm{Tr}(Q_{i_0})$. ∎

6. Numerical experiments. In this section we present numerical experiments to evaluate the performance of the prolongation operators. We implement two types of tests. The first set of experiments aims to illustrate how close a prolongated solution is to optimality. The second applies the operators in a one-way multigrid fashion along with an infeasible IPM to solve the resulting problems. The goal of this second test is to compare a basic multigrid method with a pure interior point algorithm.

To generate the problems, we use the package SparsePOP version 3.0 [37], which is an implementation of the algorithm in [36], and to solve the SDP relaxation, we use the infeasible IPM implemented in SDPT3 version 4.0 [34]. In order to obtain problems with a unique solution we perturb every polynomial by adding a small linear term (see [36]). If d is the degree of F_n, then the order of each relaxation is taken as $w = \lceil d/2 \rceil$. We also note that for the problems considered in this section, the maximal cliques created using SparsePOP are identical to those described in section 4.

The infeasibility and gaps for (y^i, S^i, X^i) at level $n = i$ are defined as follows:

- Primal feasibility: $p_{feas} \triangleq \frac{\left\|(\mathbf{A}^i)^\top y^i + \mathbf{C}^i - \mathbf{S}^i\right\|}{(1+\|\mathbf{C}^i\|)}$.
- Dual feasibility: $d_{feas} \triangleq \frac{\left\|\mathbf{A}^i(\mathbf{X}^i) - b^i\right\|}{(1+\|\mathbf{b}^i\|)}$.
- Gap: $gap \triangleq \frac{\left|\langle \mathbf{X}^i, \mathbf{S}^i \rangle\right|}{(1+|(\mathbf{b}^i)^\top y^i| + |\langle \mathbf{C}^i, \mathbf{X}^i \rangle|)}$.

$\mathbf{X}^i, \mathbf{S}^i, \mathbf{C}^i \in \mathbb{R}^{N \times N}$ are a block diagonal matrix of dimension $N = (i - p + 1)g(p, w)$. In the main diagonal the matrices $\mathbf{X}^i$ and $\mathbf{S}^i$ contain the matrices $X_1^i, X_2^i, \ldots, X_{i-p+1}^i$ and $S_1^i, S_2^i, \ldots, S_{i-p+1}^i$, respectively, while $\mathbf{C}^i$ contains $i - p + 1$ times the matrix C. $\mathbf{b}^i = \{b_{\boldsymbol{\alpha}}^i\}_{\boldsymbol{\alpha} \in \mathcal{F}^i}$, and $\mathbf{A}^i : \mathbb{R}^{N \times N} \to \mathbb{R}^{|\mathcal{F}^i|}$ is the linear operator such that

$$\mathbf{A}^i(\mathbf{X}^i) \triangleq \left(\sum_{l=1}^{i-p+1} \left\langle A_{(l,\boldsymbol{\alpha})}, X_l^i \right\rangle \right)_{\boldsymbol{\alpha} \in \mathcal{F}^i},$$

with adjoint $(\mathbf{A}^i)^\top$. If ϵ is the given tolerance level, then SDPT3 will stop when p_{feas}, d_{feas}, and gap are less than ϵ.

Our set of problems includes the following two classical test functions for global optimization problems:

- Broyden tridiagonal [24]: $F_n(\mathbf{x}) = \sum_{k=1}^n ((3 - 2x_k)x_k - x_{k-1} - 2x_{k+1} + 1)^2, x_0 = x_{n+1} = 0$.
- Generalized Rosenbrock [25]: $F_n(\mathbf{x}) = \sum_{k=2}^n \left(100(x_k - x_{k-1}^2)^2 + (1 - x_k)^2\right)$.

The second set of test problems follows the approach in [27] to solve problems that arise when a finite difference method is used to solve nonlinear boundary value problems. The boundary value problem of finding a function $x(t)$ such that $f\left(t, x(t), x'(t), x''(t)\right) = 0,\ x(a) = x_a,\ x(b) = x_b,\ a \le t \le b$, can be solved numerically by uniformly discretizing the domain, using central differences to approximate the derivatives of $x(t)$ and then solving the following system of polynomial equations:

$$f_k\left(t_k, x_{k-1}, x_k, x_{k+1}\right) = f\left(t_k, x_k, \frac{x_{k+1} - x_{k-1}}{2h}, \frac{x_{k-1} - 2x_k + x_{k+1}}{h^2}\right) = 0,$$

where $x_0 = x_a,\ x_{n+1} = x_b,\ t_k = a + hk$, and $h = (b - a)/(n + 1)$ $(k = 1, 2, \ldots, n)$. This system can be solved by minimizing the sum of the squares of the functions f_k. For example, consider the following boundary value problem:

- $x''(t) - 2x(t)^3 = 0,\ x(0) = \frac{1}{2},\ x(1) = \frac{1}{3}$ [27]. The associated polynomial is

$$F_n(\mathbf{x}) = \sum_{k=1}^n \left(x_{k-1} - 2x_k + x_{k+1} - 2h^2 x_k^3\right)^2, x_0 = \frac{1}{2},\ x_{n+1} = \frac{1}{3}.$$

We use this approach to solve nine nonlinear boundary value problems for the case where all problem data are given by polynomials. To formulate the constrained problems as in (1), the constraints for the first and last variables are eliminated

by introducing them into the objective function. Note that for the boundary value problems the function F_n does not follow exactly the structure of (1). However, when the number of points in the grid is large (i.e., h is small) the difference between the functions goes to zero and the formulation would follow the same structure. The nine nonlinear differential equation problems we considered are given in Table 2.

6.1. Prolongated variables results. In section 5 we discussed how the prolongation operators can be defined for a single level, i.e., $t = 1$. If $t > 1$ in the relaxation of the coarse problem (10), we can prolongate points in the coarse level space $n - t$ into level n by using t times the one level operators. In order to allow for $t > 1$ we start by selecting $i_0 \in \{n_1 + p + 1, n_1 + p + 2, \ldots, n_2 - t - 1\}$ and use it to define the prolongation operators applied to the points (y^{n-t}, S^{n-t}), and select $j_0 \in \{2, 3, \ldots, n - t - p\}$ for the prolongation of X^{n-t} to obtain the points one level up: $(y^{n-t+1}, S^{n-t+1}, X^{n-t+1})$. Then the process is repeated for the points at level $n-t+1$ by selecting $i_0 \in \{n_1+p+1, n_1+p+2, \ldots, n_2-t\}$, $j_0 \in \{2, 3, \ldots, n-t-p+1\}$ for the prolongation of (y^{n-t+1}, S^{n-t+1}) and X^{n-t}, respectively, and so on until level n is reached.

Using the process described above we now specify prolongation operators for more than one level. In the case of the dual operator (18), Theorem 11 suggests that the selection of the integer j_0 should be done such that ϵ_2 is small. We found that for our problems, selecting $j_0 = \lfloor (n-t-p+1)/2 \rfloor$ (i.e., selecting the middle) returned small values for ϵ_2. From this observation, our strategy consists of prolongating X^{n-t} using (18) t consecutive times, setting $j_0 = \lfloor (n - t - p + 1)/2 \rfloor$ for all the prolongations. If $X^n = (X_1^n, X_2^n, \ldots, X_{n+p-1}^n)$ is obtained following the previous description, we can write it as

$$X_l^n = \begin{cases} X_l^{n-t} \text{ if } l \in \{1, 2, \ldots, j_0 - 1\}, \\ \frac{2^t-1}{2^t} X_{j_0-1}^{n-t} + \frac{1}{2^t} X_{j_0}^{n-t} \text{ if } l \in \{j_0, j_0 + 1, \ldots, j_0 + t - 1\}, \\ X_{l-t}^{n-t} \text{ if } l \in \{j_0 + t, \ldots, n - p + 1\}. \end{cases} \tag{29}$$

It is possible to take two different approaches for the primal variables. The first is a linear operator similar to the one used for the dual variables. The prolongation for S^{n-t} follows the same criterion as the one used for the dual variable using j_0. The prolongated primal variable y^n is defined as

$$y_{\boldsymbol{\alpha}}^n = \begin{cases} y_{\boldsymbol{\alpha}}^{n-t} \text{ if } \boldsymbol{\alpha} \in B_l \text{ and } l \in \{1, 2, \ldots, i_0 - 1\}, \\ \frac{2^t-1}{2^t} y_{\boldsymbol{\alpha}^-}^{n-t} + \frac{1}{2^t} y_{\boldsymbol{\alpha}}^{n-t} \text{ if } \boldsymbol{\alpha} \in B_l \text{ and } l \in \{i_0, i_0 + 1, \ldots, i_0 + t - 1\}, \\ y_{\boldsymbol{\alpha}^{-t}}^{n-t} \text{ if } \boldsymbol{\alpha} \in B_l \text{ and } l \in \{i_0 + t, i_0 + t + 1, \ldots, n\}, \end{cases} \tag{30}$$

where $i_0 = j_0 = \lfloor (n-t-p+1)/2 \rfloor$. We found this approach effective for the Broyden tridiagonal and generalized Rosenbrock problems. Consider the solution for the generalized Rosenbrock function; in this case the optimal polynomial variables extracted after solving the SDP relaxation are $x_1 \approx -1$ and $x_j \approx 1$ for $j \geq 2$ independent of the number n in F_n, and therefore $x_i^{\alpha_i} x_{i+1}^{\alpha_{i+1}} \ldots x_{i+p-1}^{\alpha_{i+p-1}} - x_{i+1}^{\alpha_i} x_{i+2}^{\alpha_{i+1}} \ldots x_{i+p}^{\alpha_{i+p-1}} \approx 0$ for any $\boldsymbol{\alpha} \in \text{supp}(F_n)$ for $i \geq 2$ (note that the primal variables $y_{\boldsymbol{\alpha}}$ replace the monomials $\mathbf{x}^{\boldsymbol{\alpha}}$ to obtain the SDP relaxation; then we expect that $|\mathbf{x}^{\boldsymbol{\alpha}^-} - \mathbf{x}^{\boldsymbol{\alpha}}|$ should be approximately equal to $|y_{\boldsymbol{\alpha}^-} - y_{\boldsymbol{\alpha}}|$, which determines ϵ_1 in Theorem 10). The Broyden tridiagonal test problem also has a constant solution for most of its variables, and therefore the prolongation works well for this problem too. In general we expect this kind of operator to work well when the difference between the variables in the original formulation is small or constant.

For the boundary value problems, the use of the linear operators for the primal coarse variables gave prolongated points at fine levels with large values for p_{feas} for many of the problems. This is due to the fact that the difference between consecutive variables (see Theorem 10(b)) should be small when the corresponding points in the grid are close enough for the linear operator to be useful. Although the difference between the variables of the solution in the polynomial space should decrease as the number of points in the grid increases (assuming convergence of the method), in our experiments the value of n used was not large enough to make the linear operator useful. To address this issue, we defined a second prolongation for the primal variable that directly follows the ideas of multigrid theory (see, for example, [6]). Let y^{n_c} be a point for the coarse level n_c of the primal relaxation of a nonlinear differential equation with boundary conditions x_0 and x_{n_c+1}. If $e_j \in \mathbb{N}^n$ is a unit vector with one in position j, then the nonlinear prolongation $y^n = P_y^{nl}(y^{n_c})$ from level n_c to level n is calculated as

$$y^n_{\boldsymbol{\alpha}} = \left(y^n_{e_1}\right)^{\alpha_1}\left(y^n_{e_2}\right)^{\alpha_2}\cdots\left(y^n_{e_n}\right)^{\alpha_n}, \quad \boldsymbol{\alpha}\in\mathcal{F}^n, \tag{31}$$

where

$$y^n_{e_i} = y^{n_c}_{e_j} + \left(y^{n_c}_{e_{(j+1)}} - y^{n_c}_{e_j}\right)\left(i\left(\frac{n_c-1}{n-1}\right) - j\right) \quad \text{if } i\left(\frac{n_c-1}{n-1}\right) \in [j, j+1],$$

with $y^{n_c}_{e_0} = x_0$ and $y^{n_c}_{e_{n_c+1}} = x_{n_c+1}$. The operator calculates the first moment fine variables $(y^n_{e_i})$ by using a linear interpolation of the first moment coarse variables $(y^{n_c}_{e_i})$, and then calculates the rest of the moments in a nonlinear way. It is not difficult to see that this operator gives feasible points for any y^{n_c} in the sense that $\sum_{\boldsymbol{\alpha}\in\mathcal{F}^n} A_{(l,\boldsymbol{\alpha})}[P_y^{nl}(y^{n_c})]_{\boldsymbol{\alpha}} + C \succeq 0$ for $l = 1, 2, \ldots, n-p+1$. Therefore, there is no need to define an operator for the variable S^{n_c}, and hence we take $S^n_l = \sum_{\boldsymbol{\alpha}\in\mathcal{F}^n} A_{(l,\boldsymbol{\alpha})}[P_y^{nl}(y^{n_c})]_{\boldsymbol{\alpha}} + C$. Note that in practice it may not be known which of the two prolongations to use. Since both prolongations are computationally inexpensive, the best approach is to calculate both and use the one that provides the least error.

In our first set of experiments we evaluate the performance of the prolongated solutions in terms of feasibility and gaps in the fine level. For $n = 1000$ we prolongate the solutions of the coarse model $n/2$ and calculate feasibility and optimality measures of the new points (the SDPT3 tolerance was set to 10^{-7}). Table 1 shows the results for the Broyden tridiagonal and the generalized Rosenbrock functions using the linear operators (29) and (30). The first three columns of each row contain the results when the prolongated variables are used in the fine level n, and the last three columns contain the information $\{p_{feas}, d_{feas}, gap\}$ for the coarse model ($n_c = n/2$). In both problems the prolongated solutions give points with infeasibility and duality gap no greater than 10^{-4}.

TABLE 1
Feasibility and gaps of projected variables for Broyden tridiagonal and generalized Rosenbrock functions ($n = 1000$).

Problem	p_{feas}	d_{feas}	gap	$p^{n_c}_{feas}$	$d^{n_c}_{feas}$	gap^{n_c}
Broyden tridiagonal	4e-06	7e-05	8e-08	1e-09	2e-12	8e-08
Generalized Rosenbrock	9e-08	3e-07	3e-08	8e-11	8e-12	3e-08

The same exercise was done for the boundary value problems but using the nonlinear operator (31) and size $n + 1 = 500$ (see Table 2). In this case the coarse model

TABLE 2
Feasibility and gaps of projected variables for the nonlinear differential equations ($n+1=500$).

Problem	d_{feas}	gap	$p^{n_c}_{feas}$	$d^{n_c}_{feas}$	gap^{n_c}
1. $x''-2x^3=0,\ x(0)=\frac{1}{2},\ x(1)=\frac{1}{3}$	6e-05	2e-07	3e-08	3e-10	3e-07
2. $x''+\frac{1}{2}(x+t)^3=0,\ x(0)=0,\ x(1)=0$	4e-05	3e-05	4e-09	6e-11	1e-04
3. $x''-2x^3+100\sin(t)=0,\ x(0)=\frac{1}{2},\ x(1)=\frac{1}{3}$	3e-05	2e-05	2e-08	7e-11	6e-05
4. $x''+(\frac{1}{7}x')^2+1=0,\ x(0)=0,\ x(1)=0$	2e-07	5e-05	2e-08	2e-09	1e-04
5. $x''-\frac{3}{2}x^2=0,\ x(0)=4,\ x(1)=1$	8e-06	1e-06	5e-09	2e-09	3e-06
6. $x''+x'x-x^3=0,\ x(1)=\frac{1}{2},\ x(2)=\frac{1}{3}$	8e-04	7e-06	3e-09	3e-11	7e-05
7. $x''-\frac{1}{8}\left(32+2t^3-x'x\right)=0,\ x(1)=17,\ x(3)=\frac{43}{3}$	2e-03	2e-06	5e-09	8e-11	4e-06
8. $x''t^2-2=0,\ x(1)=0,\ x(2)=0$	6e-05	1e-05	2e-08	7e-10	2e-05
9. $2x''x+(x')^2=0,\ x(1)=0,\ x(100)=2$	4e-06	6e-03	4e-08	1e-08	1e-03

has $n_c=(n-1)/2$ variables. The results show how the prolongated solutions can provide good initial points to use, for example, as initial guesses for an algorithm to solve the fine level. This idea will be explored in the next subsection. Note also that SDPT3 cannot solve some of the coarse problems for the differential equations to the desired accuracy (10^{-7}). We will also address this point in the next subsection.

6.2. A multigrid approach to solve the fine problems. In our final set of experiments we use a one-way multigrid approach to solve large scale boundary value problems. In the previous subsection we observed that even when the size of the problem is moderate, SDPT3 could not solve many of the relaxations for nonlinear differential equations to the required tolerance. We overcome this problem for a large number of cases using a multigrid approach in conjunction with an infeasible IPM (in our experiments we use SDPT3). We describe the method in Algorithm 1. Let $IPM(A,b,C,y^0,S^0,X^0,\epsilon)$ be a function that uses an infeasible IPM to solve the SDP problem with parameters A, b, C, using the initial point (y^0,S^0,X^0). This function then returns a solution (y,S,X) that satisfies the required error tolerance ϵ. If no initial point is given, we will write $IPM(A,b,C,[\,],[\,],[\,],\epsilon)$ (in which case we use the default initialization procedure of the algorithm). Also, let P_{X,j_0} be the prolongation operator defined by (29) for some $n_1+p+1\le j_0\le n_2-2$, and let P^{nl}_y be the nonlinear operator defined by (31). The parameter L indicates the total number of levels, including the fine level, that are going to be used. If the goal is to solve a problem with n variables, the method uses $L-1$ coarse levels with $n_i=\lfloor n_{i+1}/2\rfloor$ variables at level i ($n_L=n$). Then the first coarse level (n_1 variables) is solved to the accuracy $[tol]_1$ with no initial point provided, and the prolongated solution is used as a starting point to solve the second coarse level (n_2 variables) to an accuracy of $[tol]_2$. The process is repeated until the level L corresponding to the fine level is solved to an accuracy of $[tol]_L$.

There is a trade-off between solving the coarse levels to a high accuracy and the CPU time used to achieve it. The coarse solution must provide meaningful information to the fine level. However, computing a highly accurate coarse solution may not be the most efficient use of CPU time; even an exact solution to the coarse model will be at best an approximation for the fine model. We found that starting with an accuracy of 10^{-4} for the coarse level with the fewest variables, and slowly increasing the accuracy as the number of variables increases, is a good rule of thumb. We use Algorithm 1 in conjunction with an infeasible IPM (SDPT3) to solve the SDP relaxation for the nine nonlinear differential equations to an accuracy of 10^{-7} (i.e.,

$[tol]_L = 10^{-7}$ in Algorithm 1). The sizes of the problems are $n = 20, 30, 40, \ldots, 1000$ (99 relaxations for each of the boundary value problems). The first experiment consists of setting $L = 2$ and $tol = [10^{-4}, 10^{-7}]$ (we will refer to this settings as $Multi_2$). In the second experiment we allow more than two levels, depending on the size of the problem ($Multi_{L\geq 2}$). In particular if $n \leq 100$, then $L = 2$, $tol = [10^{-4}, 10^{-7}]$; if $101 \leq n \leq 200$, then $L = 3$, $tol = [10^{-4}, 10^{-5}, 10^{-7}]$; if $201 \leq n \leq 500$, then $L = 4$, $tol = [10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}]$; and if $n \geq 501$, then $L = 5$, $tol = [10^{-4}, 10^{-5}, 10^{-6}, 5 * 10^{-7}, 10^{-7}]$.

Algorithm 1. Multigrid method L levels ($Multi_L$) to solve SDP relaxation of (1) with n variables.

Input: $L, \{n_i\}_{i=1}^{L} \in \mathbb{N}$ such that $n_i = \lfloor n_{i+1}/2 \rfloor$ and $n_L = n$, $tol \in \mathbb{R}^L$ such that $[tol]_i > 0$, and $\{\mathbf{A}^{n_i}, \mathbf{b}^{n_i}, \mathbf{C}^{n_i}\}_{i=1}^{L}$ as defined at the beginning of section 6.

Procedure:

for $i = 1$ to $i = L$ **do**
  if $i = 1$ **then**
    $(y^{n_i}, S^{n_i}, X^{n_i}) \leftarrow IPM\left(\mathbf{A}^{n_i}, \mathbf{b}^{n_i}, \mathbf{C}^{n_i}, [\,], [\,], [\,], [tol]_i\right)$
  else
    $(y^{n_i}, S^{n_i}, X^{n_i}) \leftarrow IPM\left(\mathbf{A}^{n_i}, \mathbf{b}^{n_i}, \mathbf{C}^{n_i}, y^{n_i,0}, S^{n_i,0}, X^{n_i,0}, [tol]_i\right)$
  end if
  if $i < L$ **then**
    $j_0 \leftarrow \lfloor (n_i - p + 1)/2 \rfloor$
    $y^{n_{i+1},0} \leftarrow P_y^{nl}(y^{n_i})$
    $X^{n_{i+1},0} \leftarrow P_{X,j_0}(X^{n_i})$
    $S_l^{n_{i+1},0} \leftarrow \sum_{\boldsymbol{\alpha}\in\mathcal{F}^{n_{i+1}}} A_{(l,\boldsymbol{\alpha})} y^{n_{i+1},0} + C,$ for $l = 1, 2, \ldots, n_{i+1} - p + 1$
  end if
end for

The prolongated matrices $P_X\left(X^{n_c}\right)$ should, in theory, be positive definite as they are calculated using SDPT3. However, numerical errors make their eigenvalues nonpositive in some cases. Also, the primal variables $P_S\left(S^{n_c}\right)$ are positive semidefinite when the nonlinear operator is used. For this reason, before the initial guesses are given to SDPT3 to solve the fine problem, we perturbed the matrices such that it is possible to calculate a Cholesky factorization. When using Algorithm 1, the early stops of SDPT3 given by the parameter OPTIONS.stoplevel were set to zero, and we increased the tolerance of the early stop criteria for the infeasibility given in line 721 of the code sqlpmain.m (we substituted 10^{-4} tolerance for 10^{-12}). These changes in the code were done after observing that the initial length steps calculated by SDPT3 after giving the initial prolongated solution were very small, which, combined with the small duality gap and/or infeasibilities, made SDPT3 end prematurely.

In our experiments we also compare our results with *SeDuMi* version 1.3 and SDPA. For both solvers we set the tolerance again to 10^{-7}. SDPA could only solve 20 of the 891 test problems to an accuracy of at least 10^{-4}, and therefore we do not report these results. It is important to note that the stopping criteria of *SeDuMi* is not the same as the one used by SDPT3. Therefore, if *SeDuMi* reports that the solution found has the required accuracy, it might not satisfy the accuracy criteria of SDPT3. Table 3 shows, for each of the nine models based on nonlinear differential equations, how many of the 99 fine relaxations were solved to the desired accuracy by SDPT3, *SeDuMi,* and the multigrid approach. A relaxation is considered solved if the solver reports that the solution satisfies the required accuracy.

SDPT3 solved a total of 105 out of the 891 relaxations, with no more than 19 out

TABLE 3
Comparison between relaxations solved by SDPT3, SeDuMi, and the multigrid approach for the nonlinear differential equations.

Differential equation*	1	2	3	4	5	6	7	8	9
# relaxations solved by SDPT3	18	14	15	0	10	17	9	19	3
# relaxations solved by *SeDuMi*	41	13	37	21	99	37	99	9	0
# relaxations solved by $Multi_2$	99	97	87	10	76	99	99	99	16
# relaxations solved by $Multi_{L\geq 2}$	99	97	91	97	97	99	93	99	27

* The total number of relaxations per differential equation is 99.

of the 99 relaxations solved for each differential equation. *SeDuMi* improves over these results by solving 356 relaxations. Using the multigrid approach $Multi_2$ we are able to solve 682 relaxations, with problems 4, 5, and 9 having the worst results with only 10, 76, and 16 relaxations solved. Using more than two levels, we can match or increase the number of relaxations solved by $Multi_2$, with a total of 799 relaxations solved (except in problem 7, where we saw a small difference of six additional relaxations solved by $Multi_2$). In particular, for problems 4 and 5 $Multi_{L\geq 2}$ can solve 97 of the 99 relaxations. We attribute the improvement to the fact that with more levels the algorithm can find a better solution to prolongate because smaller problems are easier to solve to a higher tolerance. We also tried two level experiments with 10^{-7} accuracy for the coarse level, but the results did not improve with respect to $Multi_2$ due to the inability of SDPT3 to solve the relaxation to that accuracy.

$Multi_{L\geq 2}$ performed well, except for problem 9 where it was only able to solve 27 relaxations. Upon further investigation we found that for problem 9 the finite difference scheme used in this paper is not a consistent approximation for the underlying differential equation. However, the resulting POP shares the same polynomial structure as the other problems considered in this paper. We therefore included this problem in our numerical experiments to see if the inconsistency of the discretization scheme has a large effect. Nevertheless, it is important to remark that $Multi_{L\geq 2}$ was able to solve 55 and 94 relaxations to an accuracy of 10^{-6} and 10^{-4}, respectively, while SDPT3 solved 6 and 12 relaxations to an accuracy of 10^{-6} and 10^{-4}, respectively, and *SeDuMi* reports 9 and 99 cases to an accuracy of 10^{-6} 10^{-4}, respectively.

When comparing how many models are solved as a function of the size (see Figure 1), we observe that as n becomes larger, both SDPT3 and *SeDuMi* have difficulties solving the sparse relaxations. In contrast, the multigrid approach with more than two levels is able to solve almost all the problems (except problem 9) independent of the size.

To investigate the reasons for the increase in performance of the multigrid approach, we report the condition number of the Schur-complement matrix for the last iteration performed by the SDPT3 and $Multi_{L\geq 2}$ (see [34] for more on the Schur-complement matrix). In [32], it was shown how exploiting sparsity for SDP problems using chordal completion and maximal clique decomposition approaches (like those used by [36] and this paper) may lead to SDP problems that are primal degenerate. As a consequence, the numerical experiments in this paper show an increase in the condition number of the Schur-complement matrix when IPMs were used to solve the sparse problem, compared with the condition number of the Schur-complement matrix of the IPM of the nonsparse semidefinite program. Table 4 shows, for each problem, in how many of the 99 relaxations SDPT3 had a larger condition number for the last iteration than $Multi_{L\geq 2}$ ("# $CN_{SDPT3} > CN_{Multi_{L\geq 2}}$"), the average of the ratio between the condition number of the last iteration using SDPT3 and the

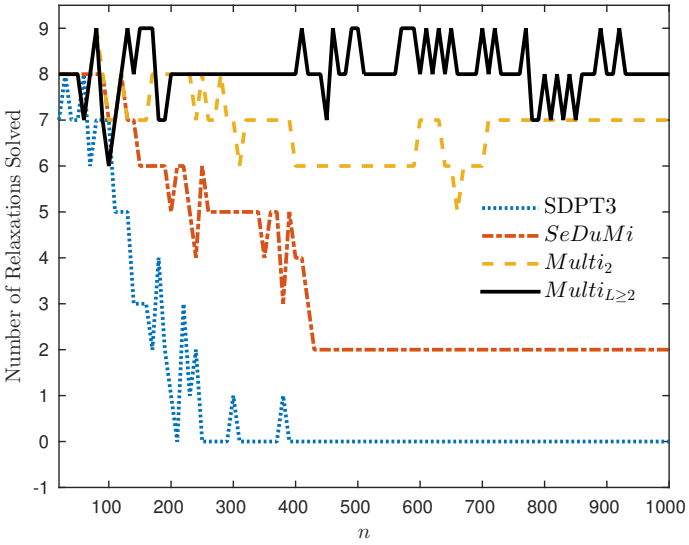

FIG. 1. *Comparison of relaxations solved by SDPT3, SeDuMi, and the multigrid algorithm as a function of the size of the problem (n) for the nonlinear differential equations.*

TABLE 4
Condition number of the Schur-complement matrix for the last iteration at the fine level using SDPT3 and $Multi_{L\geq 2}$ for the nonlinear differential equations.

Differential equation	1	2	3	4	5	6	7	8	9
$\#\ CN_{SDPT3} > CN_{Multi_{L\geq 2}}$	95	99	89	97	94	97	76	85	42
$\text{mean}_{CN_{SDPT3}/CN_{Multi_{L\geq 2}}}$	5e+13	7e+13	6e+07	4e+14	7e+06	3e+12	7e+15	3e+02	7e+01
$\text{min}_{CN_{SDPT3}/CN_{Multi_{L\geq 2}}}$	8e-02	4e+00	1e-06	5e-02	5e-02	9e-02	5e-37	6e-02	6e-04
$\text{max}_{CN_{SDPT3}/CN_{Multi_{L\geq 2}}}$	4e+15	5e+15	5e+09	4e+16	1e+08	9e+13	7e+17	1e+04	6e+03

condition number of the last iteration using $Multi_{L\geq 2}$ ("$\text{mean}_{CN_{SDPT3}/CN_{Multi_{L\geq 2}}}$"), and the minimum and maximum ratios. With the exception of problem 9, for most of the problems the condition number is larger for the pure IPM method compared with $Multi_{L\geq 2}$, which could explain the results of the multigrid approach. However, in some cases the condition number was larger for the multigrid approach, but SDPT3 could not solve the problem and $Multi_{L\geq 2}$ could. More research is needed to determine if this is the only reason that explains the superior performance of the multigrid method and exactly how the approach helps in the case of SDP problems with degenerate solutions.

In the final set of experiments, we use Algorithm 1 with the same settings as in $Multi_2$ and $Multi_{L\geq 2}$, but we change the 10^{-7} tolerance at the fine level. In particular, we set $[tol]_L$ in Algorithm 1 equal to the maximum between the feasibility and duality gap measures for the fine model when it is solved by SDPT3. We repeat this exercise using the maximum between the feasibility and duality gap measures for the fine model obtained by *SeDuMi* (since *SeDuMi* uses a different feasibility measures than SDPT3, we calculate p_{feas}, d_{feas}, and gap using the solutions reported by the solver so they match the SDPT3 criteria). For these experiments we only considered the cases where SDPT3 and *SeDuMi* achieved at least a 10^{-4} tolerance. We show the results in three tables based on the number of variables n: small size (Table 5, where

TABLE 5
CPU time comparison for the nonlinear differential equations problems solved to at least a 10^{-4} *accuracy. Small size:* $n = 20, 30, \ldots, 100$ *(nine relaxations per differential equation).*

SDPT3									
Differential equations	1	2	3	4	5	6	7	8	9
# Solved SDPT3	9	9	9	9	9	9	9	9	9
# Solved $Multi_2$	8	9	3	9	7	9	9	9	8
# Solved $Multi_{L\geq 2}$	8	9	3	9	7	9	9	9	8
% Faster $Multi_2$	88%	100%	0%	67%	0%	89%	11%	0%	25%
% Faster $Multi_{L>2}$	88%	100%	0%	67%	0%	89%	0%	0%	0%
t_{SPDT3}/t_{Multi_2}	1.29	1.30	0.56	1.10	0.77	1.19	0.89	0.70	0.86
$t_{SPDT3}/t_{Multi_{L\geq 2}}$	1.16	1.19	0.48	1.13	0.71	1.08	0.80	0.62	0.79
SeDuMi									
Differential equations	1	2	3	4	5	6	7	8	9
# Solved *SeDuMi*	9	9	2	9	8	9	0	9	9
# Solved $Multi_2$	9	9	2	9	8	9	-	9	9
# Solved $Multi_{L\geq 2}$	9	9	2	7	8	9	-	9	9
% Faster $Multi_2$	100%	100%	100%	100%	100%	89%	-	100%	100%
% Faster $Multi_{L\geq 2}$	89%	100%	100%	100%	100%	78%	-	100%	100%
t_{SeDuMi}/t_{Multi_2}	1.25	1.69	1.33	1.56	4.10	1.12	-	3.34	1.96
$t_{SeDuMi}/t_{Multi_{L\geq 2}}$	1.15	1.53	1.15	1.61	3.75	1.01	-	2.91	1.76

TABLE 6
CPU time comparison for the nonlinear differential equations problems solved to at least a 10^{-4} *accuracy. Medium size:* $n = 110, 120, \ldots, 500$ *(40 relaxations per differential equation).*

SDPT3									
Differential equations	1	2	3	4	5	6	7	8	9
# solved SDPT3	35	32	36	12	40	25	40	25	3
# solved $Multi_2$	35	32	36	12	40	25	40	25	3
# solved $Multi_{L\geq 2}$	35	32	35	12	39	25	40	25	3
% faster $Multi_2$	100%	100%	97%	92%	90%	100%	100%	80%	67%
% faster $Multi_{L\geq 2}$	100%	100%	57%	100%	21%	100%	82%	44%	67%
t_{SPDT3}/t_{Multi_2}	1.79	1.78	1.31	1.71	1.15	1.55	1.24	1.15	1.33
$t_{SPDT3}/t_{Multi_{L\geq 2}}$	2.49	2.72	1.07	1.52	0.90	2.03	1.20	0.99	1.07
SeDuMi									
Differential equations	1	2	3	4	5	6	7	8	9
# solved *SeDuMi*	40	40	40	40	37	40	0	40	40
# solved $Multi_2$	40	40	40	2	37	40	-	39	19
# solved $Multi_{L\geq 2}$	40	40	40	40	37	40	-	40	30
% faster $Multi_2$	100%	100%	95%	100%	100%	100%	-	100%	100%
% faster $Multi_{L\geq 2}$	100%	100%	62%	100%	100%	100%	-	100%	90%
t_{SeDuMi}/t_{Multi_2}	2.14	2.76	1.66	2.06	8.10	2.14	-	7.37	2.14
$t_{SeDuMi}/t_{Multi_{L\geq 2}}$	3.05	4.30	1.38	4.36	7.25	2.86	-	6.53	1.60

$n = 20, 30, \ldots, 100$), medium size (Table 6, where $n = 110, 120, \ldots, 500$), and large size (Table 7, where $n = 510, 520, \ldots, 1000$). Each table is divided into two parts: the first part shows the comparison with SDPT3, and the second part shows comparison with *SeDuMi*. We described the first part; the second part with the *SeDuMi* results has the same structure. For each of the nine differential equations, the first part shows the number of relaxations solved by SDPT3 to at least 10^{-4} accuracy ("# solved SDPT3"), how many of those models solved were also solved by $Multi_2$ and $Multi_{L\geq 2}$ to the same accuracy obtained by SDPT3 ("# solved $Multi_2$," "# solved $Multi_{L\geq 2}$"), the percentage of the relaxations solved to the same accuracy where the time spent by SDPT3 was larger than the multigrid time ("% faster $Multi_2$," "% faster $Multi_{L\geq 2}$"), and, in the last two rows, the average of the ratio between the CPU time required by SDPT3 and the CPU time required by the multigrid approach

TABLE 7
CPU time comparison for the nonlinear differential equations problems solved to at least a 10^{-4} *accuracy. Large size:* $n = 510, 520, \ldots, 1000$ *(*50 *relaxations per differential equation).*

SDPT3									
Differential equations	1	2	3	4	5	6	7	8	9
# solved SDPT3	12	12	6	0	50	16	50	0	0
# solved $Multi_2$	12	12	6	-	50	16	50	-	-
# solved $Multi_{L>2}$	12	12	6	-	50	16	50	-	-
% faster $Multi_2$	100%	100%	100%	-	100%	100%	100%	-	-
% faster $Multi_{L>2}$	100%	100%	100%	-	12%	100%	74%	-	-
t_{SPDT3}/t_{Multi_2}	2.30	1.95	1.49	-	1.09	1.53	1.17	-	-
$t_{SPDT3}/t_{Multi_{L>2}}$	4.36	3.89	1.91	-	0.89	2.68	1.14	-	-
SeDuMi									
Differential equations	1	2	3	4	5	6	7	8	9
# solved *SeDuMi*	50	50	50	50	8	50	0	50	50
# solved $Multi_2$	50	50	50	23	8	50	-	50	7
# solved $Multi_{L>2}$	50	50	50	50	8	50	-	50	32
% faster $Multi_2$	100%	100%	100%	100%	100%	100%	-	100%	100%
% faster $Multi_{L>2}$	100%	100%	100%	100%	100%	100%	-	100%	75%
t_{SeDuMi}/t_{Multi_2}	1.96	1.84	1.73	1.69	8.93	1.72	-	7.44	1.34
$t_{SeDuMi}/t_{Multi_{L>2}}$	4.04	4.16	1.72	3.99	8.74	3.10	-	8.29	1.26

("t_{SDPT3}/t_{Multi_2}," "$t_{SDPT3}/t_{Multi_{L\geq 2}}$"). We note that if tol_{SDPT3} is the maximum error among p_{feas}, d_{feas}, and gap when using SDPT3, then we only compare times if $tol_{SDPT3} \leq 10^{-4}$ and if the multigrid approach achieved a solution with tol_{SDPT3} accuracy (the same applies when using *SeDuMi*). The reported CPU time for the multigrid approach includes the time spent creating the coarse model and prolongating the variables.

In general, the multigrid algorithms can solve the relaxations to the same accuracy as *SeDuMi,* with the exception of problem 9, where the multigrid approach cannot achieve the same accuracy as *SeDuMi* for many relaxations in the medium and large ranges (the same was observed for problem 5 but only compared with the two level multigrid method). When compared with SDPT3, the multigrid method can achieve the same accuracy for medium and large problems, but for small sizes there are many relaxations where the accuracy obtained by SDPT3 cannot be reached by the multigrid algorithm. In terms of times, the multigrid method is faster than *SeDuMi* on average, while $Multi_2$ is faster for medium and large sizes than SDPT3. For small sizes, SDPT3 performs better for five of the nine problems than the two versions of the multigrid method. It is important to note that *SeDuMi* achieved better accuracy in general than SDPT3, and it is not surprising that it takes longer than SDPT3. Our results suggest that using more than two levels in Algorithm 1 is beneficial for solving medium to large scale problems with high accuracy. However, if the problem is small and/or a low accuracy is required, then the overhead of creating many levels and prolongating the solutions may be significant.

7. Conclusion. We showed how to take advantage of both sparsity and hierarchical structure present in many large scale polynomial optimization problems. The hierarchical structure of many polynomial optimization problems indicates that it is often possible to define fine and coarse models that capture the underlying application at different levels of fidelity. Our main contribution was to show how to take advantage of the information from the SDP relaxations of coarse models. Our the-

oretical results suggest that if some easy-to-check conditions are satisfied, then we should expect a significant reduction in complexity by integrating multigrid ideas with infeasible IPMs. Our numerical results back up our theoretical analysis. In particular, we showed how the multigrid approach can improve the robustness of IPMs and reduce solution times, especially for medium to large problems and when high accuracy solutions are required. This work suggests some interesting directions for future research. Using a basic one-way multigrid approach, we observed substantial improvements in practical applications. Therefore, it would be interesting to implement a full multigrid algorithm that may include v and w-cycles. Another obvious extension is to study the constrained cases such as in [21, 23], where we would expect similar patterns such as those described in this work. We concentrated on SDP relaxations arising from global optimization problems, particularly for the solution of boundary value problems; however, similar ideas are applicable in other settings such as moment relaxations of optimal control problems [19]. Finally, in our work we used the sparse polynomial relaxation developed in [36]. Recently a sparse relaxation was proposed in [38] with very promising theoretical and numerical results. The application of multigrid and the development of prolongation operators for this new hierarchy are also likely to lead to improvements.

Acknowledgments. The authors wish to acknowledge the associate editor and the four anonymous referees for their helpful comments that led to substantial improvements in both the theoretical and numerical results of the paper.

REFERENCES

[1] J. R. S. Blair and B. Peyton, *An introduction to chordal graphs and clique trees*, in Graph Theory and Sparse Matrix Computation, IMA Vol. Math. Appl. 56, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer-Verlag, 1993, pp. 1–29.

[2] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, *The maximum clique problem*, in Handbook of Combinatorial Optimization, Springer, 1999, pp. 1–74.

[3] A. Borzì, K. Kunisch, and D. Y. Kwak, *Accuracy and convergence properties of the finite difference multigrid solution of an optimal control optimality system*, SIAM J. Control Optim., 41 (2003), pp. 1477–1497, https://doi.org/10.1137/S0363012901393432.

[4] A. Borzì and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, Comput. Sci. Eng. 8, SIAM, 2011, https://doi.org/10.1137/1.9781611972054.

[5] A. Borzì and V. Schulz, *Multigrid methods for PDE optimization*, SIAM Rev., 51 (2009), pp. 361–395, https://doi.org/10.1137/060671590.

[6] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000, https://doi.org/10.1137/1.9780898719505.

[7] D. Cox, J. Little, and D. O'shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer-Verlag, 1992.

[8] A. Dontchev and W. Hager, *The Euler approximation in state constrained optimal control*, Math. Comp., 70 (2001), pp. 173–203.

[9] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, 2nd ed., Ann. Discrete Math. 57, Elsevier, 2004.

[10] S. Gratton, A. Sartenaer, and P. L. Toint, *Recursive trust-region methods for multiscale nonlinear optimization*, SIAM J. Optim., 19 (2008), pp. 414–444, https://doi.org/10.1137/050623012.

[11] D. Henrion and J.-B. Lasserre, *Detecting global optimality and extracting solutions in gloptipoly*, in Positive Polynomials in Control, Springer, 2005, pp. 293–310.

[12] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Math. Model. Theory Appl. 23, Springer Science+Business Media, 2008.

[13] C. P. Ho and P. Parpas, *Singularly perturbed Markov decision processes: A multiresolution algorithm*, SIAM J. Control Optim., 52 (2014), pp. 3854–3886, https://doi.org/10.1137/130944254.

[14] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2012.
[15] V. Hovhannisyan, P. Parpas, and S. Zafeiriou, *Magma: Multi-level accelerated gradient mirror descent algorithm for large-scale convex composite minimization*, SIAM J. Imaging Sci., 9 (2016), pp. 1829–1857, https://doi.org/10.1137/130944254.
[16] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817, https://doi.org/10.1137/S1052623400366802.
[17] J. B. Lasserre, *Convergent SDP-relaxations in polynomial optimization with sparsity*, SIAM J. Optim., 17 (2006), pp. 822–843, https://doi.org/10.1137/05064504X.
[18] J. B. Lasserre, *An Introduction to Polynomial and Semi-algebraic Optimization*, Cambridge Texts Appl. Math. 52, Cambridge University Press, 2015.
[19] J. B. Lasserre, D. Henrion, C. Prieur, and E. Trélat, *Nonlinear optimal control via occupation measures and lmi-relaxations*, SIAM J. Control Optim., 47 (2008), pp. 1643–1666, https://doi.org/10.1137/070685051.
[20] F. Lin, Z. Di, and S. Leyffer, *A multiscale approach to a class of semidefinite programs*, to in Proceedings of the 2016 American Control Conference, IEEE, pp. 7153–7158.
[21] M. Mevissen, M. Kojima, J. Nie, and N. Takayama, *Solving partial differential equations via sparse sdp relaxations*, P. J. Optim., 4 (2008), pp. 213–241.
[22] M. Mevissen, J. B. Lasserre, and D. Henrion, *Moment and SDP relaxation techniques for smooth approximations of problems involving nonlinear differential equations*, in Proceedings of the 18th IFAC World Congress on Automatic Control, Elsevier, 2011, pp. 10887–10892.
[23] M. Mevissen, K. Yokoyama, and N. Takayama, *Solutions of polynomial systems derived from the steady cavity flow problem*, in Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation, ACM, 2009, pp. 252–262.
[24] J. J. More, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.
[25] S. G. Nash, *Newton-type minimization via Lanczos method*, SIAM J. Numer. Anal., 21 (1984), pp. 770–788, https://doi.org/10.1137/0721052.
[26] S. G. Nash, *A multigrid approach to discretized optimization problems*, Optim. Methods Softw., 14 (2000), pp. 99–116.
[27] J. Nie and J. Demmel, *Sparse SOS relaxations for minimizing functions that are summations of small polynomials*, SIAM J. Optim., 19 (2008), pp. 1534–1558, https://doi.org/10.1137/060668791.
[28] I. Papamichail and C. S. Adjiman, *A rigorous global optimization algorithm for problems with ordinary differential equations*, J. Global Optim., 24 (2002), pp. 1–33.
[29] P. A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Programming, 96 (2003), pp. 293–320.
[30] P. A. Parrilo and B. Sturmfels, *Minimizing polynomial functions*, in Algorithmic and Quantitative Real Algebraic Geometry, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 60, S. Basu and L. Gonzalez-Vega, eds., American Mathematical Society, 2003, pp. 83–100.
[31] F. A. Potra and R. Sheng, *A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 1007–1028, https://doi.org/10.1137/S1052623495294955.
[32] A. U. Raghunathan and A. V. Knyazev, *Degeneracy in maximal clique decomposition for semidefinite programs*, in Proceedings of the 2016 American Control Conference (ACC), IEEE, 2016, pp. 5605–5611.
[33] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003, https://doi.org/10.1137/1.9780898718003.
[34] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, *On the implementation and usage of sdpt3–a MATLAB software package for semidefinite-quadratic-linear programming, version* 4.0, in Handbook on Semidefinite, Conic and Polynomial Optimization, Springer, 2012, pp. 715–754.
[35] L. Vandenberghe and M. S. Andersen, *Chordal Graphs and Semidefinite Optimization*, Now Publishers Incorporated, 2015.
[36] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, *Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity*, SIAM J. Optim., 17 (2006), pp. 218–242, https://doi.org/10.1137/050623802.
[37] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto, *Algorithm* 883: *Sparsepop—a sparse semidefinite programming relaxation of polynomial optimization prob-*

lems, ACM Trans. Math. Software (TOMS), 35 (2008), 15.

[38] T. Weisser, J. B. Lasserre, and K.-C. Toh, *Sparse-bsos: A bounded degree SOS hierarchy for large scale polynomial optimization with sparsity*, Math. Prog. Comput., 2017, pp. 1–32, https://doi.org/10.1007/s12532-017-0121-6.

[39] Z. Wen and D. Goldfarb, *A line search multigrid method for large-scale nonlinear optimization*, SIAM J. Optim., 20 (2009), pp. 1478–1503, https://doi.org/10.1137/08071524X.