Mathematical Programming Computation (2019) 11:503–586
https://doi.org/10.1007/s12532-019-00164-4

FULL LENGTH PAPER

Sieve-SDP: a simple facial reduction algorithm to preprocess semidefinite programs

Yuzixuan Zhu[1] · **Gábor Pataki**[1] · **Quoc Tran-Dinh**[1]

Received: 20 September 2017 / Accepted: 7 February 2019 / Published online: 1 July 2019

Abstract
We introduce Sieve-SDP, a simple facial reduction algorithm to preprocess semidefinite programs (SDPs). Sieve-SDP inspects the constraints of the problem to detect lack of strict feasibility, deletes redundant rows and columns, and reduces the size of the variable matrix. It often detects infeasibility. It does not rely on any optimization solver: the only subroutine it needs is Cholesky factorization, hence it can be implemented in a few lines of code in machine precision. We present extensive computational results on several problem collections from the literature, with many SDPs coming from polynomial optimization.

Keywords Semidefinite programming · Preprocessing · Strict feasibility · Strong duality · Facial reduction · Polynomial optimization

Mathematics Subject Classification 90-08 · 90C22 · 90C25 · 90C06

1 Introduction and the preprocessing algorithm

Consider a semidefinite programming problem (SDP) in the form

$$\begin{aligned} \inf_{X} \quad & C \bullet X \\ \text{s.t.} \quad & A_i \bullet X = b_i \;\; (i = 1, \ldots, m), \\ & X \succeq 0, \end{aligned} \tag{P}$$

✉ Gábor Pataki
gabor@unc.edu

Yuzixuan Zhu
zyzx@live.unc.edu

Quoc Tran-Dinh
quoctd@email.unc.edu

[1] Department of Statistics and Operations Research, The University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3260, USA

where the A_i and C are $n \times n$ symmetric matrices, the b_i are scalars, $X \succeq 0$ means that X is in $\mathcal{S}^n_+$, the set of symmetric, positive semidefinite (psd) matrices, and the $\bullet$ inner product of symmetric matrices is the trace of their regular product.

SDPs are some of the most versatile, useful, and widespread optimization problems of the last three decades. They find applications in control theory, integer programming, and combinatorial optimization, to name just a few areas. Several good solvers are available to solve SDPs (see for example [1,6,7,16,17,21,39,44,50]); among these, Mosek [1] is commercially available.

SDPs—as all optimization problems—often have redundant variables and/or constraints. The redundancy we address is lack of *strict feasibility*, i.e., when there is no feasible positive definite X in (P). When (P) is not strictly feasible, the optimal value of (P) and of its dual may differ, and the latter may not be attained.[1] Hence, when attempting to solve such an SDP, solvers often struggle, or fail.

It is, of course, useful to detect lack of strict feasibility in a preprocessing stage. This paper describes a very simple preprocessing algorithm for SDPs, called Sieve-SDP, which belongs to the class of facial reduction algorithms [4,12,13,22,30,31,34,43,47]. Sieve-SDP can detect lack of strict feasibility, reduce the size of the problem, and can be implemented in a few lines of code in machine precision.

To motivate our algorithm, let us consider an example:

Example 1 The SDP instance (with an arbitrary objective function)

$$\begin{aligned} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X &= 0 \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \bullet X &= -1 \\ X &\succeq 0, \end{aligned} \tag{1}$$

is infeasible. Indeed, suppose $X = (x_{ij})_{i,j=1}^3$ is feasible in (1). Then $x_{11} = 0$, hence the first row and column of X are zero by positive semidefiniteness, so the second constraint implies $x_{22} = -1$, which is a contradiction.

Note that if we replace -1 in the second constraint of (1) by a positive number, then (1) can be restated over the set of psd matrices with first row and column equal to zero. Thus, even if we do not detect infeasibility, such preprocessing is still useful.

Our algorithm Sieve-SDP repeats the Basic Step shown in Fig. 1. Hereafter $D \succ 0$ means that a symmetric matrix D is positive definite.

Example 2 (Example 1 continued) When we first execute the Basic Step on (1), we find the first constraint, delete it, and also delete the first row and column from the second constraint matrix. Next, we find the constraint

[1] More precisely, when (P) *is* strictly feasible, *strong duality* holds between (P) and its dual, i.e., their values agree and the latter is attained.

Basic Step

1. Find $i \in \{1, \ldots, m\}$ (if any) such that the ith constraint of (P), after permuting rows and columns, and possibly multiplying both sides by -1, is of the form

$$\begin{pmatrix} D_i & 0 \\ 0 & 0 \end{pmatrix} \bullet X = b_i, \tag{2}$$

where $D_i \succ 0$ and $b_i \leq 0$. If there is no such i, STOP; (P) cannot be preprocessed further.
2. If $b_i < 0$, then STOP; (P) is infeasible.
3. If $b_i = 0$, then delete this constraint. Also delete all rows and columns in the other constraints that correspond to rows and columns of D_i.

Fig. 1 The Basic Step of Sieve-SDP

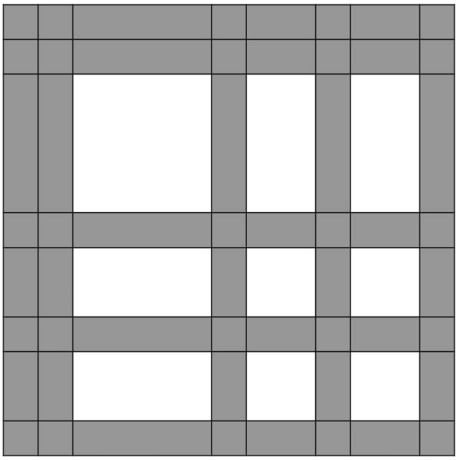

Fig. 2 The sieve structure

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \bullet X = -1,$$

and declare that (1) is infeasible.

We call our algorithm Sieve-SDP, since by shading the deleted rows and columns in the variable matrix X (and the A_i) we obtain a sieve-like structure: see Fig. 2.

Sieve-SDP is easy to implement and fast: it only needs an incomplete Cholesky factorization subroutine to check positive definiteness, and we can delete rows and columns using fast matrix operations. Even the worst case complexity of Sieve-SDP is reasonable: an easy calculation shows that it can fully preprocess (P) using $O(\min\{m, n\}n^3m)$ arithmetic operations.

Sieve-SDP is a heuristic: it does not always detect infeasibility, or lack of strict feasibility. For example, it will not work on problem (1), if we apply a similarity transformation $T^\top(\cdot)T$ to all A_i, where T is a random invertible matrix.

Given its simplicity, and how easily it is "fooled", it is natural to ask whether our algorithm works in practice. So the main research question we address, and answer in the affirmative, is:

– Can Sieve-SDP help us compute more accurate solutions and reduce the computing time on a broad range of SDPs?

Related work Sieve-SDP belongs to the family of *facial reduction algorithms*, which we now describe. When (P) is not strictly feasible, one can replace the constraint $X \in \mathcal{S}^n_+$ by

$$X \in F,$$

where F is a proper face of $\mathcal{S}^n_+$.[2] Since any such face can be written as (see e.g. [29])

$$F = V\mathcal{S}^r_+ V^\top, \tag{3}$$

where $r < n$ and V is an $n \times r$ matrix, the reduced problem can be restated over a smaller semidefinite cone. Facial reduction algorithms—for more general conic programs—originated in the papers [3,4]. Later simplified, more easily implementable variants were given in [30,31,47], and in [43] for the SDP case. A recent, very concise version with a short proof of convergence is in [24].

Facial reduction algorithms, when applied to (P), find the face F by solving a sequence of SDP subproblems, which may be as hard to solve as (P) itself. Thus one is led to seek simpler alternatives.

Simplified and implementable versions of facial reduction are described in [34]. The algorithms in [34] reduce the feasible set of (P) (or of an SDP in a different shape) by solving linear programs instead of SDPs. Thus they do not find *all* reductions, but still simplify the SDPs in many cases. They are available as public domain codes, and we will compare them with Sieve-SDP in Sect. 2. A facial reduction algorithm embedded in an interior point method was implemented in [35].

We next review facial reduction algorithms that work by simply inspecting constraints. For example, [15] notes that if

$$A \bullet X = 0$$

is a constraint in (P) with $A \succeq 0$, then we can restrict X to belong to a face of the form (3), where V spans the nullspace of A. A similar idea was used in [22] to reduce Euclidean Distance Matrix completion problems. For a rigorous derivation of the algorithm in [22] see [13], which used an intermediate step of analyzing the semidefinite completion problem. For followup work, see [12] on the noisy version of the same problem, and [41] for a more theoretical study.

We finally mention two very accurate SDP solvers, which do not rely on facial reduction. The first is SDPA-GMP [16], which uses the GMP library and computes solutions of (P) and of its dual using several hundred digits of accuracy. We will use SDPA-GMP in later sections to check the accuracy of the solutions computed by Sieve-SDP and Mosek. The SPECTRA solver [20] computes a feasible solution

[2] That is, $F \neq \mathcal{S}^n_+$, F is convex, and $X, Y \in \mathcal{S}^n_+$, $\frac{1}{2}(X + Y) \in F$ implies that X and Y are in F.

of (P) (if one exists) in exact arithmetic. Although these solvers cannot handle large SDPs, they can solve small ones very accurately.

Sieve-SDP differs in several aspects from previously proposed facial reduction algorithms:

- It needs only Cholesky factorization as a subroutine and, unlike the algorithms in [34], it does not rely on any optimization solver.
- It detects very simple redundancies, which are easy to explain even to a user not trained in optimization, and can help him/her to better formulate other problems.
- As soon as Sieve-SDP finds a reducing constraint, it deletes this constraint, and it also deletes redundant rows and columns from the other constraint matrices. Hence errors do not accumulate. Thus Sieve-SDP is as accurate as Cholesky factorization, which works in machine precision [42, Theorem 23.2].
- Sieve-SDP can also detect infeasibility.
- It is easy to run in a *safe mode* (explained in the next section) to even better safeguard against numerical errors.
- Finally, we present extensive computational results on general SDPs, which, as far as we know, are not yet available for such a simple algorithm.

The rest of the paper is organized as follows. In Sect. 2 we describe how we implemented Sieve-SDP, the computational setup, and the criteria for comparison with competing codes. In this section we also give a small SDP with a positive duality gap (in Example 3), and show how to construct a pair of primal–dual solutions with arbitrarily small constraint violation and arbitrarily small duality gap. This example shows that a solution with a *smaller* DIMACS error (see [27]) may be actually *less accurate*. We also show that such a less accurate solution is actually computed by Mosek, one of the leading SDP solvers.

In Sect. 3 we comment in detail on the results on some of the problems, and on the strengths and weaknesses of the preprocessors. For example, we examine whether they help to find the correct solution of numerically difficult SDPs; and how fast they are on large scale problems.

In Sect. 4 we summarize the preprocessing results, and conclude the paper.

We have four appendices. In Appendix A we present very detailed computational results on all the problems. In Appendix B we give the core Matlab code of Sieve-SDP, containing only about 65 lines. In Appendix C we provide the definition of the DIMACS errors for completeness. In Appendix D we discuss the issue of recovering an optimal solution of the dual of (P) from the optimal solution of the dual of the reduced problem.

2 Implementation, setup for computational testing, codes used for comparison, and the issue of positive duality gaps

2.1 Implementation and computing environment

We implemented our algorithm in Matlab R2015a, using the standard Cholesky factorization (subroutine `chol`) to check positive definiteness.

We ran both Sieve-SDP and the competing preprocessors (which we describe in Sect. 2.3) on a MacBook Pro with processor Intel Core i5 running at 2.7 GHz, and 8 GB of RAM.

2.2 Safe mode

To safeguard against numerical errors we use a *safe mode*. We set

$$\epsilon := 2^{-52} \approx 2.2204 \cdot 10^{-16} = \text{the machine precision in Matlab.}$$

In the Basic Step in Fig. 1, if we find a constraint of type (2), then, instead of checking $b_i < 0$ we check whether

$$b_i < -\sqrt{\epsilon}\max\{\|b\|_\infty, 1\}\text{ holds.}$$

If this test fails, then instead of checking $b_i = 0$ we check whether

$$b_i > -\epsilon\max\{\|b\|_\infty, 1\}\text{ holds.}$$

Note that this step is correct, because in the Basic Step we already ensured $b_i \leq 0$.

2.3 Preprocessors used for comparison

We compare Sieve-SDP with the algorithms proposed by Permenter and Parrilo in [34]. Their algorithms solve linear programming subproblems to reduce the size of an SDP. They can work either on the problem (P), which we call the *primal*; or on its dual:

$$\begin{aligned} \sup_y \ & \sum_{i=1}^m y_i b_i \\ \text{s.t.} \ & \sum_{i=1}^m y_i A_i \preceq C. \end{aligned} \tag{D}$$

They can use either diagonal, or diagonally dominant reductions (for details, see [34]).

Thus, there are four algorithms from [34] that we tested: pd1, pd2, dd1, and dd2. Here pd1 stands for primal diagonal; pd2 for primal diagonally dominant; dd1 for dual diagonal; and dd2 for dual diagonally dominant.

Remark 1 In the theoretical description of the algorithms in [34] the SDP which is called the primal is actually our dual (D). However, in their implementation and their code posted on the github website, their primal is the same as our primal (P).

2.4 The datasets

We tested Sieve-SDP and competing methods on five datasets, which contain 771 problems overall.

- The first is the dataset from [34], which we call the Permenter–Parrilo or PP dataset. This dataset contains 68 problems, whose original sources are [2,5,8–10,14,33,36,37,45,46,48]. Although a few problems in this dataset are randomly generated, most come from applications.

The PP dataset contains SDPs that are notoriously difficult for solvers, and some are known to be not strictly feasible. Hence we added the following four datasets to make our testing more comprehensive:

- A dataset we obtained from Hans Mittelmann's website, which we call the Mittelmann dataset. This dataset contains 31 problems.
- A collection of SDP relaxations of polynomial optimization problems based on the paper of Dressler, Illiman, and de Wolff [11], which we call the Dressler–Illiman–de Wolff dataset, or DIW dataset for short. This dataset has 155 problems.
- A problem set kindly provided to us by Didier Henrion and Kim–Chuan Toh, which we call the Henrion–Toh dataset. This dataset contains 98 problems.
- A problem set kindly provided to us by Kim–Chuan Toh, whose description is in [40] and [49]. We call this dataset the Toh–Sun–Yang dataset, and it has 419 problems.

From the PP dataset we excluded only two problems: copos_5 and cprank_3, since they were too large to be solved by Mosek on our computer.

Our datasets contain many different types of SDPs and, not surprisingly, the performance of the preprocessors on them varies widely. Many of our SDPs may be strictly feasible, and such SDPs could not be reduced by even more sophisticated preprocessors. For example, in the Toh–Sun–Yang dataset no problems were reduced by the preprocessors. Although this is a bit disappointing, Sieve-SDP and pd1 delivered the "no reduction found" result very quickly, so it did not hurt to preprocess.

Yet, even in the datasets other than the PP dataset many SDPs *were* reduced by some preprocessor. In the Henrion–Toh dataset, pd1, pd2, and Sieve-SDP all reduced 18 problems, whereas dd1 and dd2 reduced none. In the Mittelmann dataset, pd1, pd2, and Sieve-SDP reduced 8 problems; dd1 and dd2 reduced none.

Strikingly, in the DIW dataset Sieve-SDP proved infeasibility of 59 problems out of 155, and reduced total solving time by a factor of more than a hundred! Pd1 did only slightly worse.

We illustrate this point with Fig. 3, which shows the size and sparsity structure of the problem "ex4.2_order20"[3] before (on the left) and after (on the right) applying Sieve-SDP. Each row in the displayed matrices corresponds to an A_i matrix stretched out as a vector. Red dots correspond to positive entries, blue dots correspond to negative entries, and white areas correspond to zero entries.

[3] This SDP is from the DIW dataset.

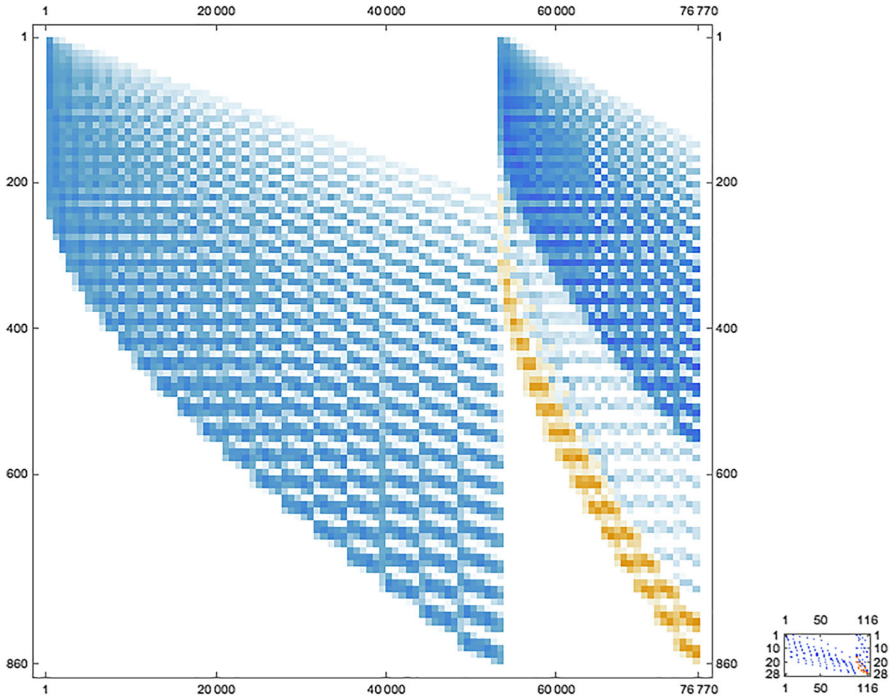

Fig. 3 Problem "ex4.2_order20": size and sparsity before and after Sieve-SDP

2.5 Internal format and input/output format

Internally we store the A_i matrices as an $n \times (nm)$ sparse matrix of the form

$$\left(A_1, A_2, \ldots, A_m\right)$$

(i.e., the A_i are stored side-by-side), and C as an $n \times n$ sparse matrix. The input- and the output format of the preprocessors is the widely used Mosekopt format.

2.6 The choice of the SDP solver and LP solver

For all preprocessors we use Mosek 8.1.0.27 (from now on, simply "Mosek") as SDP solver: we solve the SDPs with Mosek before and after preprocessing. We also solve the linear programming (LP) subproblems in the algorithms of [34] by Mosek. We believe that Mosek is the best choice, since it is a reliable commercial SDP and LP solver, and it is being actively developed and improved.

Our settings are different from the ones used in [34], where Sedumi [39] format is used as input format, Mosek as LP solver, and Sedumi as SDP solver. With our settings the algorithms of [34] work faster, because Mosek is much faster than Sedumi. Although we must convert the data from Mosekopt format to Sedumi format (to do the preprocessing), and then back (to solve the preprocessed problem with Mosek), the total conversion time is negligible: for each of pd1, pd2, dd1 and dd2 it is less than 100 seconds on *all* 771 SDPs. To be fair, in the detailed comparison tables of Appendix A we list conversion time, and preprocessing time separately.

2.7 Criteria for comparison

Let us recall the main question that we address in this paper:

– Can Sieve-SDP help us compute more accurate solutions and reduce the computing time on a broad range of SDPs?

Thus, our three criteria for comparing the preprocessors are as follows, in order of priority:

1. Do they help detect infeasibility? If not, do they help to find a correct optimal solution?
   Precisely, suppose that Mosek reports an incorrect optimal value of an SDP before preprocessing. Does Mosek find a correct optimal value after preprocessing? (We assume that the optimal value of the SDP is known mathematically.)
2. Does preprocessing reduce computing time?
   This criterion is secondary, since preprocessing is often essential to compute any accurate solution: see Sects. 3.1 through 3.3. Thus, we believe that we should always preprocess SDPs, as long as we can do this with very high precision, even if preprocessing *increases* the solution time.
3. Does preprocessing improve numerical accuracy measured by the six DIMACS errors [27]?[4] Let

   $$\text{DIMACS}_{\text{before}} \text{ and } \text{DIMACS}_{\text{after}}$$

   be the largest absolute value of the DIMACS errors before and after preprocessing, respectively. We say that a method *improves* the DIMACS error if it does not detect infeasibility and

   $$\text{DIMACS}_{\text{before}} > 10^{-6} \quad \text{and} \quad \frac{\text{DIMACS}_{\text{after}}}{\text{DIMACS}_{\text{before}}} < \frac{1}{10}.$$

   This last criterion must be taken with a grain of salt. While the DIMACS errors are very natural (they measure constraint violation and duality gap), Example 3 below shows that they do not always measure accurately how good a solution is. In fact, a *larger* DIMACS error may correspond to a *better* solution!

[4] The description of the DIMACS errors is given in Appendix C.

Example 3 Consider the SDP

$$
\begin{aligned}
\inf_X \; & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X \\
\text{s.t.} \; & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X = 0 \\
& \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \bullet X = 1 \\
& X = (x_{ij}) \succeq 0,
\end{aligned} \tag{4}
$$

and its dual

$$
\begin{aligned}
\sup_y \; & y_2 \\
\text{s.t.} \; & y_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + y_2 \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \preceq \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.
\end{aligned} \tag{5}
$$

We claim that the duality gap between them is 1. Indeed, let X be a feasible solution of (4). Since $x_{11} = 0,$ the first row and column of X must be zero, hence

$$
X = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}
$$

is an optimal solution with objective value 1. In turn, in (5) we have $y_2 = 0$ for all feasible y, so its optimal value is 0.

Next, let $\epsilon > 0$ and define $M_\epsilon > 0$ so that

$$
X_\epsilon := \begin{pmatrix} \epsilon & 0 & (1-\epsilon)/2 \\ 0 & \epsilon & 0 \\ (1-\epsilon)/2 & 0 & M_\epsilon \end{pmatrix}
$$

is positive semidefinite. Then X_ϵ is an approximate solution of (4), which violates only the first constraint (by ϵ) and has objective value 2ϵ.

Do such "fake" solutions arise in practice? At first look it seems that they do not. If we feed the pair (4)–(5) to Mosek, it returns a solution with DIMACS errors

$$
(0.5000, 0, 0.7071, 0, -5.5673 \cdot 10^{-9}, 5.9077 \cdot 10^{-17}).
$$

Since the first and third errors are large, we cannot conclude that the problem has been "solved".

However, let us apply a similarity transformation $T^\top(\cdot)T$ to all matrices in (4) with

$$T = \begin{pmatrix} 3 & 5 & -2 \\ 4 & 1 & 1 \\ -4 & -4 & 5 \end{pmatrix}.$$

Then the resulting primal–dual pair still has a duality gap of 1. Yet, Mosek now returns a solution with DIMACS errors

$$(1.6093 \cdot 10^{-6},\ 0,\ 5.2111 \cdot 10^{-9},\ 3.287 \cdot 10^{-12},\ -8.1484 \cdot 10^{-5},\ 3.0511 \cdot 10^{-5}),$$

which may seem "essentially all zero" to a user.

We argue that in any SDP pair with positive duality gap such "fake" solutions can arise. Indeed, suppose

$$\text{val}\,(D) < \text{val}\,(P),$$

where val$(\cdot)$ denotes the optimal value of an optimization problem. Then by the theory of asymptotic duality (see e.g., Section 3 in [38]) there is a sequence $\{X_\epsilon \succeq 0 \mid \epsilon > 0\}$ such that X_ϵ violates each primal constraint by at most ϵ, and

$$C \bullet X_\epsilon \to \text{val}\,(D), \text{ as } \epsilon \searrow 0.$$

As Example 3 shows, such "fake" or approximate solutions are sometimes indeed found by SDP solvers.

We note that [9] also presented computational results on SDPs with positive duality gaps, and noted that Sedumi often gave an incorrect solution on such problems. However, [9] did not report the DIMACS errors.

3 Detailed comments on some of the preprocessing results

We now report in detail how the preprocessors perform on some of the problems. We thus examine them from several angles: for example, can they help to find known optimal solutions of difficult SDPs? How do they perform on large-scale SDPs? How fast are they when they do not reduce an SDP by much, or at all?

We first look at how the preprocessors perform on the "Compact", "unbound" and "Example" problems, for which the exact optimal values are known, but are hard to compute. (These problems are from the PP dataset). The question we address is whether preprocessing helps to find these optimal values.

First we note that Sieve-SDP does not change the optimal value of (P), since it deletes rows and columns from the variable matrix X that are always zero anyway. However, it deletes rows and columns in the constraint matrices, so after applying it, in the dual (D) we require only a principal minor of $C - \sum_{i=1}^m y_i A_i$ to be psd. Thus applying Sieve-SDP may increase the optimal value of (D).

To quantify this argument, let (P_{pre}) and (D_{pre}) be the primal and dual problems after preprocessing by Sieve-SDP, respectively. Then

$$\mathrm{val}\,(D) \le \mathrm{val}(D_{\mathrm{pre}}) \le \mathrm{val}(P_{\mathrm{pre}}) = \mathrm{val}\,(P). \tag{6}$$

First assume $\mathrm{val}\,(D) < \mathrm{val}\,(P)$. Then we can show by examples that any inequality in (6) may be strict. For example, in Example 3 Sieve-SDP deletes the first row and first column in all constraint matrices, and it is easy to check that the corresponding optimal values are $0 < 1 = 1 = 1,$ respectively. In detail, for this example (D_{pre}) is

$$\begin{aligned} &\sup_{y_2}\ y_2 \\ &\text{s.t. } y_2 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \preceq \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \end{aligned} \tag{7}$$

whose optimal value is 1.

On the other hand, suppose $\mathrm{val}\,(P) = \mathrm{val}\,(D)$. Then (6) implies that Sieve-SDP changes neither the primal, nor the dual optimal values.

Which optimal values are changed or kept the same by the other preprocessors? Pd1 and pd2 also reduce the primal (P), so when we apply them, the primal optimal value (but maybe not that of the dual) will remain the same. On the other hand, dd1 and dd2 reduce the dual problem (D), so they keep its optimal value the same. However, they may change the optimal value of the primal (P).

In all tables in this section we use the following convention: the first reported objective value is the primal and the second is the dual.

3.1 "Compact" problems: 10 problems from [46]

These instances are *weakly infeasible*, i.e., the affine subspace

$$H = \{ X \mid A_i \bullet X = b_i \ (i = 1, \cdots, m)\}$$

does not intersect $\mathcal{S}_+^n$, but the distance of H to $\mathcal{S}_+^n$ is zero. Weakly infeasible SDPs are particularly challenging to SDP solvers. However, a recent algorithm in [20] can detect (in)feasibility of small SDPs in exact arithmetic, and [25] presented an algorithm that is tailored to detect weak infeasibility.

On these problems pd1 and pd2 produced the same results, while dd1 and dd2 reduced none of them. Pd1 and pd2 combined with Mosek correctly detected primal infeasibility of all problems, while Sieve-SDP correctly proved primal infeasibility without Mosek. (Since it found the primal infeasible, we did not compute a dual solution).

The results are in Table 1.

We mention here another set of infeasible, and weakly infeasible SDPs. They were presented in [24], and are available from the webpage of Gábor Pataki. Some of these SDPs are classified as "clean" and some as "messy". In the "clean" instances the structure that proves infeasibility is apparent, while in the "messy" instances that

Table 1 Results on the "Compact" problems

Problem	Correct obj (P, D)	Obj before	After pd1/pd2	After dd1/dd2	After Sieve-SDP
CompactDim2R1	Infeas, $+\infty$	3.79e+06, 4.20e+06	Infeas, 1	3.79e+06, 4.20e+06	Infeas, –
CompactDim2R2	Infeas, $+\infty$	6.41e−10, 6.81e−10	Infeas, 2	6.41e−10, 6.81e−10	Infeas, –
CompactDim2R3	Infeas, $+\infty$	1.5, 1.5	Infeas, 2	1.5, 1.5	Infeas, –
CompactDim2R4	Infeas, $+\infty$	1.5, 1.5	Infeas, 2	1.5, 1.5	Infeas, –
CompactDim2R5	Infeas, $+\infty$	1.5, 1.5	Infeas, 2	1.5, 1.5	Infeas, –
CompactDim2R6	Infeas, $+\infty$	1.5, 1.5	Infeas, 2	1.5, 1.5	Infeas, –
CompactDim2R7	Infeas, $+\infty$	1.5, 1.5	Infeas, 2	1.5, 1.5	Infeas, –
CompactDim2R8	Infeas, $+\infty$	1.5, 1.5	Infeas, 2	1.5, 1.5	Infeas, –
CompactDim2R9	Infeas, $+\infty$	1.5, 1.5	Infeas, 2	1.5, 1.5	Infeas, –
CompactDim2R10	Infeas, $+\infty$	1.5, 1.5	Infeas, 2	1.5, 1.5	Infeas, –
Correctness	100%, 100%	0%, 0%	100%, 0%	0%, 0%	100%, –

structure was obscured by two kinds of operations: random elementary row operations on the constraints and a random similarity transformation.

Indeed, in our testing all clean instances were found infeasible by Sieve-SDP, pd1, and pd2. In contrast, no messy instances were reduced by any of the preprocessors. Since the clean instances are evidently easy for Sieve-SDP, and the messy ones are hard for all preprocessors, we did not include the SDPs from [24] in our test set, since we felt that this would not be fair.

3.2 "Unbound" problems: 10 problems from [48]

The mathematically correct optimal values of both the primal and the dual are 0 in this problem collection. However, before preprocessing Mosek returned wrong optimal values for 6 out of 10 problems. Although Mosek found solutions with almost correct optimal value in problems 2, 3 and 4, these solutions are inaccurate, as the DIMACS errors are of the order 10^{-1} (this is marked by "*" symbols in Table 2).

In summary, 9 out of 10 problems in this dataset need preprocessing to obtain a reasonable solution.

Sieve-SDP, pd1 and pd2 corrected all objective values, as Table 2 shows.

It is interesting that the authors in [48] computed the correct optimal solution of these instances using SDPA-GMP [16], a high-precision SDP solver that carries several hundred significant digits. Doing so is, of course, more time consuming, than running Sieve-SDP and Mosek.

3.3 "Example" problems: 8 problems from [9]

The mathematically correct objective values are reported in [9] in table 12.1. (Note that in [9] our primal is considered the dual, and vice versa, so that table must be read accordingly.)

Table 3 shows the objective values before and after preprocessing. We consider an objective value correct if it is less than 10^{-6} away from the true optimal value.

We excluded "Example5" of [9] from this table, since in Table 12.1 in [9] its optimal value is not reported. For all other problems, except for "Example9size20" and "Example9size100", we manually verified the correctness of the optimal values in exact arithmetic.

Note that the comparison in Table 3 is somewhat unfair to Sieve-SDP: if it found a problem infeasible, it did not compute a dual solution.

3.4 "Finance" problems: 4 problems from [5]

The PP dataset contains four "finance" problems: "leverage_limit", "long_only", "sector_neutral" and "unconstrained". We report on these problems in detail, since these are the largest in the PP dataset. For example, "long_only" has 100 semidefinite variable blocks of order 91 and another 100 of order 30.

Table 2 Results on the "unbound" problems

Problem	Correct obj (P, D)	Obj before	After pd1/pd2	After dd1/dd2	After Sieve-SDP
unboundDim1R1	0, 0	1.33e−09, − 7.05e−10	1.33e−09, − 7.05e−10	1.33e−09, − 7.05e−10	0, 0
unboundDim1R2	0, 0	− 8.19e−15*, − 8.01e−15*	0, 0	− 8.19e−15*, − 8.01e−15*	0, 0
unboundDim1R3	0, 0	− 2.04e−11*, − 2.02e−11*	0, 0	− 2.04e−11*, − 2.02e−11*	0, 0
unboundDim1R4	0, 0	− 2.34e−10*, − 2.32e−10*	0, 0	− 2.34e−10*, − 2.32e−10*	0, 0
unboundDim1R5	0, 0	− 1, − 1	0, 0	− 1, − 1	0, 0
unboundDim1R6	0, 0	− 1, − 1	0, 0	− 1, − 1	0, 0
unboundDim1R7	0, 0	− 1, − 1	0, 0	− 1, − 1	0, 0
unboundDim1R8	0, 0	− 1, − 1	0, 0	− 1, − 1	0, 0
unboundDim1R9	0, 0	− 1, − 1	0, 0	− 1, − 1	0, 0
unboundDim1R10	0, 0	− 1, − 1	0, 0	− 1, − 1	0, 0
Correctness	100%, 100%	10%, 10%	100%, 100%	10%, 10%	100%, 100%

Table 3 Results on the "Example" problems

Problem	Correct obj (P, D)	Obj before	After pd1/pd2	After dd1/dd2	After Sieve-SDP
Example1	0, 0	0, 0	0, 0	0, 0	0, 0
Example2	1, 0	3.33e−01, 3.33e−01	1, 1	4.73e−15, 1.82e−14	1, 1
Example3	0, 0	3.33e−01, 3.33e−01	1.17e−07, 1.69e−07	4.73e−15, 1.82e−14	1.17e−07, 1.69e−07
Example4	Infeas, 0	Infeas, 3.74e−07	Infeas, 1	0, 0	Infeas, –
Example6	1, 1	1, 1	1, 1	1, 1	1, 1
Example7	0, 0	0, 0	0, 0	0, 0	0, 0
Example9size20	Infeas, 0	Infeas, 3.39e−01	Infeas, 1	0, 0	Infeas, –
Example9size100	Infeas, 0	Infeas, 3.43e−01	Infeas, 1	0, 0	Infeas, –
Correctness	100%, 100%	75%, 50%	100%, 50%	50%, 100%	100%, 50%

Table 4 Results on the "finance" problems

Method	n_{sdp}	n_{nonneg}	n_{free}	m	nnz
None	60,400	51,100	0	251,777	2,895,756
After pd1	60,400	51,100	0	251,777	2,895,756
After pd2	60,280	51,100	0	249,797	2,880,876
After dd1	27,429	51,100	2,286,000	251,777	2,844,756
After dd2	36,400	51,100	2,521,005	251,777	2,605,807
After Sieve-SDP	56,766	50,873	0	215,210	2,466,573

Table 4 shows how much the preprocessors reduced these SDPs: here n_{sdp} is the total size of the semidefinite blocks; n_{nonneg} is the total number of nonnegative variables; n_{free} is the total number of free variables; m is the total number of constraints; and nnz is the total number of nonzeros.

While dd1 and dd2 significantly reduced the size of the SDP blocks, they added many free variables. Sieve-SDP reduced the size of the SDP blocks, without adding free variables, and it eliminated the most constraints. We mention that after preprocessing with dd2 Mosek detected that problem "leverage_limit" is "dual infeasible". This may be because of numerical instability, and does not contradict the result we get after preprocessing with Sieve-SDP.

We remark that preprocessing actually *increased* the solution time on these problems, though not by much. For example, the total time spent on preprocessing with Sieve-SDP plus solving with Mosek is about 21% higher than the solving time with Mosek without preprocessing. Still, since the primary goal of preprocessing is to improve solution accuracy, we believe that we should do it whenever we can.

Furthermore, on these instances Sieve-SDP performed a large number of iterations, and deleted only a small submatrix in each one. Thus, we could easily reduce the time spent by Sieve-SDP by limiting the maximum number of iterations it is allowed to perform. We do not report results with such a setting, since we do not want to "overtune" our code.

3.5 Dressler–Illiman–de Wolff (DIW) dataset (155 problems)

Consider the optimization problem

$$\begin{aligned} &\min_x \ f(x) \\ &\text{s.t.} \ \ g_i(x) \geq 0 \ \ (i = 1, \ldots, m), \end{aligned} \tag{8}$$

where f and the g_i are multivariate polynomials.

As shown in the seminal work of Lasserre [23], the optimal value of (8) can be lower bounded by solving SDPs. Under suitable conditions the lower bounds converge to the optimal value of (8), as the so-called Lasserre relaxation order increases. However, no useful lower bound is obtained when the SDPs are infeasible. See Parrilo [28] for a related scheme to construct SDP relaxations of (8).

Table 5 Relaxation orders for examples in [11]

ex	3.3	4.1	4.2	4.3	4.4	5.4	5.5	5.6	5.7
Relaxation orders	$6\cdots 20$	$3\cdots 20$	$6\cdots 20$	$2\cdots 20$	$3\cdots 20$	$5\cdots 20$	$4\cdots 20$	$4\cdots 20$	$5\cdots 20$

Table 6 Results for the DIW dataset

Method	# Reduced	n	m	Preprocessing (s)	Solving (s)	# Infeas
None	–	53,523	186,225	–	139,493.56	–
pd1	155	1,450	3,278	1,632.43	128.46	56
pd2	155	1,450	3,278	10,831.32	124.44	56
dd1	0	53,523	186,225	65.18	139,493.56	0
dd2	0	53,523	186,225	22,152.57	139,493.56	0
Sieve-SDP	155	1,385	3,204	1,232.27	87.53	59

Since solving the Lasserre SDPs can be challenging, Dressler, Illiman and de Wolff [11] proposed an alternative relaxation, based on so-called nonnegative circuit polynomials, and they compared their approach with the SDP-based one.

We constructed the SDPs in the "DIW" dataset by taking the polynomial optimization problems from [11] and using Gloptipoly 3 [19] to generate their SDP relaxations.

We describe our SDPs in Table 5 with their Lasserre relaxation order, which ranges from the lowest possible (half the degree of the highest degree monomial in the polynomials) to 20. For example, the SDP named "ex3.3_order4" is obtained by applying the Lasserre relaxation of order 4 to Example 3.3 in [11].

Table 6 shows the results: "n" is the sum of the orders of all psd and nonnegative blocks, and "m" is the sum of the number of constraints in all problems.

The results are quite striking. Sieve-SDP, pd1, and pd2 ran fast, reduced all problems, detected infeasibility of more than a third, and reduced overall computing time by a factor of more than a hundred! Sieve-SDP was the best in all aspects, with pd1 a close second.

Note that without preprocessing Mosek failed to detect infeasibility of any of these SDPs.

These results are somewhat surprising since [11] solved some of these SDPs to near optimality, and managed to extract approximate optimal solutions of the original polynomial optimization problems. See [18] for similar results on similar SDPs. In fact, [18] took the view that numerical inaccuracy of the SDP solvers actually *helps* find near-optimal solutions of the polynomial optimization problems. See [26] for a more recent and thorough study of the same issue.

We remark that these SDPs are likely to be weakly infeasible.

We were thus motivated to double check that Sieve-SDP indeed reduced these SDPs correctly. Precisely, we verified that in the Basic Step (in Fig. 1) it only eliminated constraints in one of the following forms: either of the form

$$\begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \bullet X = 0,$$

where D is positive definite diagonal, of order 1 or 2, and the smallest diagonal element is 1 or 0.5 or $1/3 = 0.3333\ldots$; or of the form

$$O \bullet X = 0,$$

where O is the zero matrix. Furthermore, Sieve-SDP always detected infeasibility by finding a constraint

$$\begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \bullet X = \beta,$$

where D is as above, and $\beta = -3$ or -8.

The zeroes in all these constraints are zeroes in absolute machine precision, i.e., in the sparse SDPs returned by Gloptipoly 3 these entries do not appear at all. Thus Sieve-SDP performed all reductions correctly.

3.6 Henrion–Toh dataset (98 problems)

This dataset was kindly provided to us by Didier Henrion and Kim–Chuan Toh. The problems come mostly from polynomial optimization.

Among these problems 18 were reduced by pd1, pd2, or Sieve-SDP and none by dd1 or dd2. Table 7 shows the time details in seconds. The last column "Pre versus solve" shows the time spent on preprocessing as a percentage of time spent on solving. It is

$$\frac{\text{preprocessing time}}{\text{solving time without preprocessing}} \times 100\,\%. \tag{9}$$

This is a dataset on which the preprocessors are less successful: pd1, pd2, and Sieve-SDP detected infeasibility of only one problem (of "sedumi-l4") and they reduced solving time only a little. However, the preprocessing times are small, or even negligible: for example, Sieve-SDP spent only about 0.3% of the time that it took for Mosek to solve the problems.

In Fig. 4 we illustrate how Sieve-SDP works on the instance "sedumi-fp32": we show the sparsity structure of the constraints of the original problem (on the left), and

Table 7 Time results on the Henrion–Toh dataset

Method	Preprocessing (s)	Solving (s)	Pre. versus solve
None	–	1420.02	–
pd1	10.27	1373.70	0.72%
pd2	49.84	1374.31	3.51%
dd1	3.93	1420.02	0.28%
dd2	29.24	1420.02	2.06%
Sieve-SDP	4.58	1376.27	0.32%

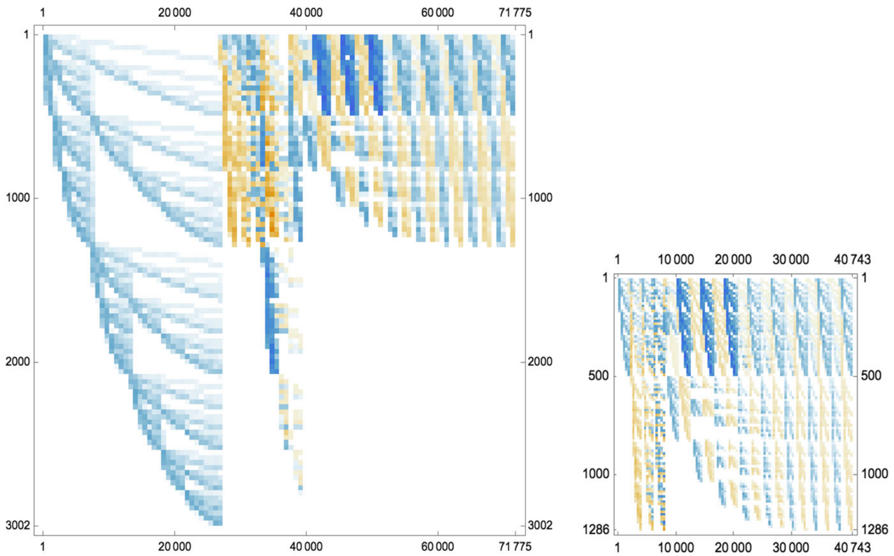

Fig. 4 Instance "sedumi-fp32": size and sparsity before (left) and after (right) preprocessing

after Sieve-SDP (on the right). Just like in Fig. 3, each row corresponds to an A_i matrix stretched out as a vector. Red dots correspond to positive entries, blue dots correspond to negative entries, and white areas to zero entries.

Here we also discuss problem "sedumi-fp33" on which preprocessing by Sieve-SDP makes the DIMACS error *worse*. Since this is the only such instance, we looked at it in more detail. The worst DIMACS error (of a solution computed by Mosek) before Sieve-SDP is 3.36×10^{-7}, which is acceptable. After Sieve-SDP the worst error is about $0.0928,$ which is unacceptable.

We also solved this instance using the high accuracy SDP solver SDPA-GMP [16]. The DIMACS errors were

$$2.3497 \cdot 10^2, 0.0000, 1.8552 \cdot 10^1, 0.0000, -9.9999 \cdot 10^{-1}, 8.5173 \cdot 10^{-2}$$

before Sieve-SDP, and

$$3.4075 \cdot 10^2, 0.0000, 1.9636 \cdot 10^1, 0.0000, -9.9999 \cdot 10^{-1}, 6.1901 \cdot 10^{-1}$$

after Sieve-SDP. In both cases the largest error is more than $200,$ which is unacceptably large.

Given the high accuracy of SDPA-GMP, it seems that this SDP cannot be accurately solved by current fast solvers, and the *worse* DIMACS error returned by Mosek after Sieve-SDP alerts the user to this fact: this problem may actually have a positive duality gap (cf. Example 3).

Table 8 Timing on the Toh–Sun–Yang dataset

Method	Preprocessing (s)	Solving (s)	Pre. versus solve
pd1	220.18	27,635.46	0.80%
pd2	4, 029.61	27,635.46	14.58%
dd1	134.64	27,635.46	0.49%
dd2	2, 428.82	27,635.46	8.79%
Sieve-SDP	152.14	27,635.46	0.55%

3.7 Toh–Sun–Yang dataset (419 problems) from [40,49]

Although none of the five methods reduced the SDPs in this collection, we still comment on them in detail, since it is interesting that pd1, dd1 and Sieve-SDP spent only a negligible amount of time on preprocessing. Thus using these three methods it does not hurt to preprocess: see Table 8. The last column "Pre versus Solve" shows the time spent on preprocessing as a percentage of time spent on solving; see Eq. (9). Pd2 and dd2, on the other hand, spent considerably more time on preprocessing.

4 Summary

We now compare all preprocessors on all instances in Tables 9, 10 and 11.

In Table 9 the second column shows how many problems were reduced. The third column shows how many problems were found to be infeasible. The fourth column shows on how many instances the preprocessing improved the DIMACS errors, as we discussed in Sect. 2.7.

The last column "Memory" shows how many times a method ran out of memory, or crashed: this happened with pd2 six times and with dd2 four times. To ensure fair reporting we reran these methods on the same instances on a machine with 24 GB RAM, and the results were the same.

Table 10 shows the preprocessing and solving times in seconds. The first column shows the preprocessing time and the second shows the solving time by Mosek after preprocessing. Column "Pre versus solve" shows the relative speed of the preprocessors: see Eq. (9). The last column, "Time reduction", shows by how much preprocessing decreased the solving time. It is

Table 9 Infeasibility detection and error reduction on all 771 problems

Method	# Reduced	# Infeas detected	# DIMACS error improved	Memory
pd1	209	67	74	0
pd2	230	67	78	6
dd1	14	0	2	0
dd2	21	0	4	4
Sieve-SDP	216	73	74	0

Table 10 Preprocessing and solving times on all 771 problems

Method	Preprocessing (s)	Solving (s)	Pre versus solve	Time reduction
None	–	272,427.23	–	–
pd1	2,486.51	132,356.63	0.91%	50.50%
pd2	23,323.07	131,636.47	8.56%	43.12%
dd1	587.93	272,244.62	0.22%	− 0.15%
dd2	35,984.45	272,031.04	13.21%	− 13.16%
Sieve-SDP	2,170.13	131,837.25	0.80%	51.81%

$$\frac{\text{solving time w.o. preprocessing} - (\text{preprocessing time} + \text{solving time after preprocessing})}{\text{solving time w.o. preprocessing}} \times 100\,\%.$$

Of course, the higher this percentage, the more a preprocessor reduces solution time. A negative percentage means that preprocessing actually *increased* the total time.

Finally, Table 11 shows by "how much" the problems were reduced. As in Table 9, the second column shows the number of problems reduced by each method.

To explain the other columns, let us fix an SDP in the primal form (P) with potentially several semidefinite block variables (some of which may be of order 1, i.e., they may be just nonnegative variables).

Let n_{before} and n_{after} be the total size of the semidefinite blocks before and after reduction. We define the reduction rate on n as

$$\frac{\sum n_{\text{before}} - \sum n_{\text{after}}}{\sum n_{\text{before}}},$$

where the sum is over all 771 problems.

Similarly, let m_{before} and m_{after} be the number of constraints in a problem before and after reduction. We define the reduction rate on m as

$$\frac{\sum m_{\text{before}} - \sum m_{\text{after}}}{\sum m_{\text{before}}},$$

where the sum is again taken over all 771 problems.

Methods dd1 and dd2 added free variables, and the fifth column in Table 11 shows how many.

The sixth column "nnz" shows the total number of nonzeros in the constraint matrices.

Given these tables we now summarize our findings. In all aspects Sieve-SDP is competitive with the other preprocessing methods. In detail:

- It is competitive considering the number of problems reduced.
- It is competitive in computing known optimal solutions; see Tables 1, 2 and 3.
- The time spent on preprocessing with Sieve-SDP versus solving is negligible. It is also negligible for pd1 and dd1, but less so for pd2 and dd2. See Table 10.

Table 11 Size reduction on all 771 problems

Method	# Reduced	Red. on n	Red. on m	Extra free vars	nnz
None	–	–	–	–	300,989,332
pd1	209	15.47%	17.79%	0	211,299,702
pd2	230	15.59%	18.23%	0	211,257,726
dd1	14	6.74%	0.00%	2,293,495	300,936,120
dd2	21	9.28%	0.00%	2,315,849	299,272,012
Sieve-SDP	216	16.55%	20.66%	0	206,061,059

In several aspects Sieve-SDP is the best.

- It is best in detecting infeasibility: see Table 9. It is important that Sieve-SDP detects infeasibility without using any optimization solver, whereas the other methods rely on Mosek.
- It reduced solution time the most, with pd1 a close second. See Table 10.
- It reduced the size of the instances the most: see Table 11.
- It needs very little additional memory, precisely $O(nm)$. For details, and the Matlab code, see Appendix B.
- It is very accurate and stable: it is as accurate as Cholesky factorization, which works in machine precision. Sieve-SDP is also easily implemented in a *safe mode*: see Sect. 2.2.
- It is the simplest: the core Matlab code consists of only 65 lines.

The code is available from

https://github.com/unc-optimization/SieveSDP

Acknowledgements We thank the Technical Editor and the referees for their helpful comments. The second author, Gábor Pataki, is supported by the National Science Foundation, Award DMS-1817272. The third author, Quoc Tran-Dinh, is supported in part by the National Science Foundation, Award DMS-1619884. We are very grateful to Erling Andersen at Mosek for running several SDPs, and explaining the results; to Joachim Dahl at Mosek for helpful discussions on converting SDPs, and for providing his conversion code; to Didier Henrion and Kim-Chuan Toh for providing us with some of the datasets; to Frank Permenter and Johan Löfberg for helpful comments; to Oktay Günlük for helping us to invent the name "Sieve-SDP"; and to Hans Mittelmann for helping us with some of the large-scale SDPs.

A Very detailed results

We now give very detailed computational results on all problems, separately for the five datasets. We only report on problems that were reduced by at least one of the five preprocessors.

In all tables the first column gives the number of the SDP, the second gives the name, and the third gives the names of the preprocessing methods.

The next two columns describe the size of the problem. The entry "f; l; s" describes the size of the variables of the problem, where

- the number "f" is the number of *free* variables;
- the number "l" is the number of *linear nonnegative* variables;
- the number, or numbers "s" describe the size of the *semidefinite* variable blocks, possibly with multiplicity.

For example, 3; 5; 6 means that a problem has 3 free variables; 5 linear nonnegative variables; and a semidefinite matrix variable block of order 6. The tuple 3; 5; 6, 5_3 means that a problem has 3 free variables; 5 linear nonnegative variables; and four semidefinite matrix variable blocks, which are of order 6, 5, 5, 5, respectively. The number m is the number of constraints.

In the next three columns we put information about the preprocessors. In the column "red." we put 1, if a preprocessor reduced a problem, and 0 if it did not. In this column under Sieve-SDP we put the same entries, except when Sieve-SDP actually proved infeasibility. In that case we entered "inf" there. The number t_{prep} is the time spent on preprocessing; the number t_{conv} is the time spent on converting from Mosek format to Sedumi format and back (for the methods pd1, pd2, dd1, dd2).

In the next four columns we show how Mosek performed. In the column "inf" we have a 1 if Mosek detected infeasibility, and 0 if it did not. The column obj (P, D) shows the objective values (primal and dual, respectively). The column DIMACS contains the *largest* absolute value of the DIMACS errors.

In the last column we show *help codes*, which show whether a preprocessor helped or hurt to solve an SDP. Although the help codes can be deduced from the previous columns, they still help to quickly evaluate the preprocessors. A positive help code means that a preprocessor helped, and a negative one means that it hurt.

In detail, let us recall from Sect. 2.7 that $\text{DIMACS}_{\text{before}}$ [$\text{DIMACS}_{\text{after}}$] is the absolute value of the DIMACS error that is largest in absolute value before [after] preprocessing. We let $\text{obj}_{\text{before}}$ and $\text{obj}_{\text{after}}$ be the primal objective values before and after preprocessing, respectively.

Given this notation,

- the help code is 1, if
  - Sieve-SDP detects infeasibility, or
  - Mosek does not detect infeasibility *before* preprocessing but it does detect infeasibility *after* preprocessing;
- the help code is -1, if
  - Mosek detects infeasibility *before* preprocessing but does not detect infeasibility *after* preprocessing;
- the help code is 2, if
  - it is not ± 1 and preprocessing improved the DIMACS error, i.e.,

$$\text{DIMACS}_{\text{before}} > 10^{-6} \quad \text{and} \quad \frac{\text{DIMACS}_{\text{after}}}{\text{DIMACS}_{\text{before}}} < \frac{1}{10};$$

- the help code is -2, if
  - it is not ± 1 and preprocessing made the DIMACS error worse, i.e.,

$$\text{DIMACS}_{\text{after}} > 10^{-6} \quad \text{and} \quad \frac{\text{DIMACS}_{\text{after}}}{\text{DIMACS}_{\text{before}}} > 10;$$

- the help code is 3, if preprocessing shifted the objective value, i.e.,
  - if help codes ± 1 and -2 do not apply, and

$$\frac{|\text{obj}_{\text{before}} - \text{obj}_{\text{after}}|}{1 + |\text{obj}_{\text{before}}|} > 10^{-6};$$

- the help code is MM if a code ran out of memory or crashed.

A.1 Detailed results on the Permenter–Parrilo (PP) dataset

This dataset has 68 problems. From these 59 problems were reduced by at least one of the five methods.

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
1	CompactDim2R1	None	0; 3; 3	5				0	3.79e+06, 4.20e+06	2.22e+01	3.02	
		pd1	0; 3; 1	3	1	0.05	0.00	1	0.00e+00, 1.00e+00	7.07e−01	0.64	1
		pd2	0; 3; 1	3	1	0.04	0.00	1	0.00e+00, 1.00e+00	7.07e−01	0.69	1
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			inf	0.01					0.00	1
2	CompactDim2R2	None	$0; 0; 6, 3_3$	14				0	6.41e−10, 6.81e−10	7.07e−01	3.16	
		pd1	$0; 0; 1_3$	2	1	0.11	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.13	1
		pd2	$0; 0; 1_3$	2	1	0.09	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.04	1
		dd1			0	0.03	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			inf	0.01					0.00	1
3	CompactDim2R3	None	$0; 0; 10, 6_3$	27				0	1.50e+00, 1.50e+00	1.15e−07	2.03	
		pd1	$0; 0; 1_3$	2	1	0.14	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.07	1
		pd2	$0; 0; 1_3$	2	1	0.13	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.09	1
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			inf	0.01					0.00	1
4	CompactDim2R4	None	$0; 0; 15, 10_3$	44				0	1.50e+00, 1.50e+00	1.13e−07	2.07	
		pd1	$0; 0; 1_3$	2	1	0.20	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.09	1
		pd2	$0; 0; 1_3$	2	1	0.17	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.09	1
		dd1			0	0.03	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			inf	0.02					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
5	CompactDim2R5	None	0; 0; 21, 15_3	65				0	1.50e+00, 1.50e+00	1.83e−07	2.05	
		pd1	0; 0; 1_3	2	1	0.25	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.06	1
		pd2	0; 0; 1_3	2	1	0.27	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.07	1
		dd1			0	0.02	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP			inf	0.03					0.00	1
6	CompactDim2R6	None	0; 0; 28, 21_3	90				0	1.50e+00, 1.50e+00	2.70e−07	2.06	
		pd1	0; 0; 1_3	2	1	0.32	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.10	1
		pd2	0; 0; 1_3	2	1	0.38	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.03	1
		dd1			0	0.05	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP			inf	0.04					0.00	1
7	CompactDim2R7	None	0; 0; 36, 28_3	119				0	1.50e+00, 1.50e+00	3.66e−07	2.13	
		pd1	0; 0; 1_3	2	1	0.41	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.08	1
		pd2	0; 0; 1_3	2	1	0.59	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.06	1
		dd1			0	0.03	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP			inf	0.06					0.00	1
8	CompactDim2R8	None	0; 0; 45, 36_3	152				0	1.50e+00, 1.50e+00	5.61e−07	2.07	
		pd1	0; 0; 1_3	2	1	0.56	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.02	1
		pd2	0; 0; 1_3	2	1	0.86	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.05	1
		dd1			0	0.03	0.00					
		dd2			0	0.09	0.00					
		Sieve-SDP			inf	0.08					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
9	CompactDim2R9	None	0; 0; 55, 45_3	189				0	1.50e+00, 1.50e+00	6.27e−07	2.11	
		pd1	0; 0; 1_3	2	1	0.71	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.08	1
		pd2	0; 0; 1_3	2	1	1.28	0.00	1	1.00e+00, 2.00e+00	7.07e−01	1.05	1
		dd1			0	0.03	0.00					
		dd2			0	0.14	0.00					
		Sieve-SDP			inf	0.11					0.00	1
10	CompactDim2R10	None	0; 0; 66, 55_3	230				0	1.50e+00, 1.50e+00	5.17e−07	2.28	
		pd1	0; 0; 1_3	2	1	0.86	0.01	1	1.00e+00, 2.00e+00	7.07e−01	1.17	1
		pd2	0; 0; 1_3	2	1	1.90	0.01	1	1.00e+00, 2.00e+00	7.07e−01	1.09	1
		dd1			0	0.04	0.01					
		dd2			0	0.18	0.01					
		Sieve-SDP			inf	0.15					0.00	1
11	Example1	None	0; 0; 3	2				0	0.00e+00, 0.00e+00	0.00e+00	1.72	
		pd1	0; 0; 2	1	1	0.06	0.01	0	0.00e+00, 0.00e+00	0.00e+00	1.05	
		pd2	0; 0; 2	1	1	0.05	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.96	
		dd1	5; 0; 1	2	1	0.07	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.96	
		dd2	5; 0; 1	2	1	0.06	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.96	
		Sieve-SDP	0; 0; 2	1	1	0.03		0	0.00e+00, 0.00e+00	0.00e+00	1.63	
12	Example2	None	0; 0; 3	2				0	3.33e−01, 3.33e−01	5.05e−02	1.73	
		pd1	0; 0; 2	1	1	0.05	0.01	0	1.00e+00, 1.00e+00	0.00e+00	0.94	2, 3
		pd2	0; 0; 2	1	1	0.05	0.00	0	1.00e+00, 1.00e+00	0.00e+00	0.97	2, 3
		dd1	3; 0; 2	2	1	0.05	0.00	0	4.73e−15, 1.82e−14	2.75e−14	1.01	2, 3
		dd2	3; 0; 2	2	1	0.02	0.00	0	4.73e−15, 1.82e−14	2.75e−14	1.01	2, 3
		Sieve-SDP	0; 0; 2	1	1	0.01		0	1.00e+00, 1.00e+00	0.00e+00	2.61	2, 3
13	Example3	None	0; 0; 3	4				0	3.33e−01, 3.33e−01	6.90e−02	1.79	
		pd1	0; 0; 2	1	1	0.03	0.00	0	1.17e−07, 1.69e−07	5.14e−08	1.44	2, 3
		pd2	0; 0; 2	1	1	0.03	0.00	0	1.17e−07, 1.69e−07	5.14e−08	1.48	2, 3
		dd1	3; 0; 2	4	1	0.02	0.01	0	4.73e−15, 1.82e−14	2.75e−14	1.00	2, 3
		dd2	3; 0; 2	4	1	0.03	0.00	0	4.73e−15, 1.82e−14	2.75e−14	0.99	2, 3
		Sieve-SDP	0; 0; 2	1	1	0.01		0	1.17e−07, 1.69e−07	5.14e−08	1.59	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
14	Example4	None	0; 0; 3	3				1	0.00e+00, 3.74e−07	5.00e−01	1.43	
		pd1	0; 0; 1	1	1	0.03	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.64	
		pd2	0; 0; 1	1	1	0.03	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.63	
		dd1	5; 0; 1	3	1	0.03	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.96	− 1
		dd2	5; 0; 1	3	1	0.04	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.99	− 1
		Sieve-SDP			inf	0.00					0.00	1
15	Example6	None	0; 0; 8	8				0	1.00e+00, 1.00e+00	1.95e−08	0.66	
		pd1	0; 0; 5	4	1	0.04	0.00	0	1.00e+00, 1.00e+00	0.00e+00	0.99	
		pd2	0; 0; 5	4	1	0.04	0.00	0	1.00e+00, 1.00e+00	0.00e+00	0.98	
		dd1	26; 0; 4	8	1	0.02	0.00	0	1.00e+00, 1.00e+00	9.75e−09	1.02	
		dd2	26; 0; 4	8	1	0.02	0.00	0	1.00e+00, 1.00e+00	9.75e−09	1.19	
		Sieve-SDP	0; 0; 5	4	1	0.01		0	1.00e+00, 1.00e+00	0.00e+00	0.56	
16	Example7	None	0; 0; 5	3				0	0.00e+00, 0.00e+00	0.00e+00	0.60	
		pd1	0; 0; 4	2	1	0.02	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.96	
		pd2	0; 0; 4	2	1	0.03	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.97	
		dd1	14; 0; 1	3	1	0.03	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.98	
		dd2	14; 0; 1	3	1	0.03	0.00	0	0.00e+00, 0.00e+00	0.00e+00	1.00	
		Sieve-SDP	0; 0; 4	2	1	0.00		0	0.00e+00, 0.00e+00	0.00e+00	0.54	
17	Example9size20	None	0; 0; 20	20				1	0.00e+00, 3.39e−01	5.00e−01	2.58	
		pd1	0; 0; 1	1	1	0.06	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.63	
		pd2	0; 0; 1	1	1	0.04	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.62	
		dd1	209; 0; 1	20	1	0.19	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.95	− 1
		dd2	209; 0; 1	20	1	0.24	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.97	− 1
		Sieve-SDP			inf	0.00					0.00	1
18	Example9size100	None	0; 0; 100	100				1	0.00e+00, 3.43e−01	5.00e−01	0.83	
		pd1	0; 0; 1	1	1	0.04	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.64	
		pd2	0; 0; 1	1	1	0.19	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.63	
		dd1	5049; 0; 1	100	1	1.33	0.00	0	0.00e+00, 0.00e+00	0.00e+00	1.01	− 1
		dd2	5049; 0; 1	100	1	3.50	0.00	0	0.00e+00, 0.00e+00	0.00e+00	1.00	− 1
		Sieve-SDP			inf	0.00					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
19	RandGen6	None	0; 0; 320	140				0	3.95e−06, 3.24e−06	2.29e−05	24.07	
		pd1			0	3.64	1.00					
		pd2			0	16.39	1.00					
		dd1			0	0.75	1.00					
		dd2	19985; 0; 250	140	1	37.13	2.14	0	1.68e−07, 1.26e−11	8.00e−07	5.88	2, 3
		Sieve-SDP	0; 0; 120	70	1	2.10		0	3.73e−06, 3.04e−06	9.17e−06	2.32	
20	RandGen7	None	0; 0; 40	27				0	9.42e−07, 4.22e−07	4.69e−06	0.67	
		pd1			0	0.03	0.01					
		pd2	0; 0; 28	14	1	0.10	0.02	0	9.85e−07, 4.53e−07	3.27e−06	1.04	
		dd1			0	0.02	0.01					
		dd2	649; 0; 18	27	1	0.11	0.01	0	2.65e−11, 4.69e−16	7.21e−11	1.08	2
		Sieve-SDP	0; 0; 28	14	1	0.02		0	9.85e−07, 4.53e−07	3.27e−06	0.72	
21	RandGen8	None	0; 0; 60	40				0	5.41e−09, 2.44e−09	9.31e−08	0.83	
		pd1			0	0.04	0.01					
		pd2			0	0.22	0.01					
		dd1			0	0.02	0.01					
		dd2	1269; 0; 33	40	1	0.33	0.02	0	2.15e−15, 2.78e−19	6.90e−14	1.05	
		Sieve-SDP	0; 0; 30	20	1	0.03		0	1.52e−09, 6.33e−10	9.04e−09	0.69	
22	copos_1	None	0; 0; 35	210				0	0.00e+00, 1.11e−08	4.40e−07	0.66	
		pd1			0	0.02	0.00					
		pd2	0; 0; 25	160	1	0.06	0.02	0	0.00e+00, − 3.86e−10	2.12e−08	1.01	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP			0	0.02						
23	copos_2	None	0; 0; 120	1716				0	0.00e+00, 5.76e−11	1.69e−08	1.83	
		pd1			0	0.03	0.00					
		pd2	0; 0; 96	1524	1	0.57	0.11	0	0.00e+00, − 2.31e−13	6.38e−11	1.72	
		dd1			0	0.02	0.00					
		dd2			0	0.13	0.00					
		Sieve-SDP			0	0.09						

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
24	copos_3	None	0; 0; 286	8008				0	0.00e+00, − 4.93e−10	1.59e−07	44.68	
		pd1			0	0.10	0.01					
		pd2	0; 0; 242	7524	1	37.41	0.57	0	0.00e+00, − 4.51e−11	1.26e−08	30.28	
		dd1			0	0.06	0.01					
		dd2			0	0.85	0.01					
		Sieve-SDP			0	0.46						
25	copos_4	None	0; 0; 560	27132				0	0.00e+00, − 9.00e−11	7.21e−08	1526.50	
		pd1			0	0.46	0.06					
		pd2	0; 0; 490	26152	1	26.16	1.98	0	0.00e+00, − 1.70e−10	6.56e−08	1139.18	
		dd1			0	0.36	0.06					
		dd2			0	5.09	0.06					
		Sieve-SDP			0	1.80						
26	cprank_1	None	9; 0; 1_9, 10, 9	46				0	− 3.00e+00, − 3.00e+00	3.50e−08	1.32	
		pd1			0	0.08	0.00					
		pd2			0	0.03	0.00					
		dd1	30; 0; 1_7, 8, 9	46	1	0.07	0.01	0	− 3.00e+00, − 3.00e+00	4.62e−08	1.16	
		dd2	30; 0; 1_7, 8, 9	46	1	0.06	0.01	0	− 3.00e+00, − 3.00e+00	3.88e−08	1.17	
		Sieve-SDP			0	0.01						
27	cprank_2	None	1296; 0; 1_{81}, 82, 81	3322				0	− 9.00e+00, − 9.00e+00	6.62e−08	15.40	
		pd1			0	0.08	0.00					
		pd2			0	0.18	0.00					
		dd1	3456; 0; 1_{49}, 50, 81	3322	1	0.14	0.31	0	− 9.00e+00, − 9.00e+00	6.64e−09	10.50	
		dd2	3456; 0; 1_{49}, 50, 81	3322	1	0.50	0.32	0	− 9.00e+00, − 9.00e+00	1.51e−09	9.75	
		Sieve-SDP			0	0.06						
28	hinf12	None	0; 0; 6_2, 12	43				0	− 1.45e−13, − 1.17e−13	1.80e+00	1.38	
		pd1			0	0.03	0.00					
		pd2	0; 0; 6, 2, 6	22	1	0.04	0.00	0	− 2.64e−15, − 1.77e−15	1.79e+00	1.69	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP			0	0.01						

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
29	horn2	None	0; 0; 4	7				0	0.00e+00, 6.69e−13	9.06e−13	2.03	
		pd1			0	0.02	0.00					
		pd2	0; 0; 2	3	1	0.06	0.00	0	0.00e+00, 0.00e+00	1.57e−16	0.99	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			0	0.01						
30	horn3	None	0; 0; 10	28				0	0.00e+00, 1.46e−07	8.62e−07	2.00	
		pd1			0	0.02	0.00					
		pd2	0; 0; 6	16	1	0.05	0.00	0	0.00e+00, 3.53e−09	2.65e−08	0.99	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			0	0.00						
31	horn4	None	0; 0; 20	84				0	0.00e+00, 1.13e−07	1.90e−06	2.14	
		pd1			0	0.02	0.00					
		pd2	0; 0; 14	60	1	0.07	0.01	0	0.00e+00, 7.11e−09	7.44e−08	1.07	2
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			0	0.01						
32	horn5	None	0; 0; 35	210				0	0.00e+00, 1.07e−08	2.69e−07	2.05	
		pd1			0	0.02	0.00					
		pd2	0; 0; 25	160	1	0.08	0.01	0	0.00e+00, − 2.28e−09	2.35e−07	0.99	
		dd1			0	0.03	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			0	0.03						
33	hornD2	None	0; 0; 4	3				0	− 5.25e−08, 0.00e+00	5.25e−08	2.04	
		pd1			0	0.02	0.00					
		pd2			0	0.03	0.00					
		dd1			0	0.03	0.00					
		dd2	7; 0; 2	3	1	0.05	0.00	0	− 1.88e−16, 0.00e+00	1.78e−15	1.00	
		Sieve-SDP			0	0.00						

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
34	hornD3	None	0; 0; 10	27				0	−5.58e−08, 0.00e+00	8.62e−07	2.01	
		pd1			0	0.03	0.00					
		pd2			0	0.02	0.00					
		dd1			0	0.03	0.00					
		dd2	34; 0; 6	27	1	0.05	0.00	0	−8.68e−10, 0.00e+00	1.88e−08	1.13	
		Sieve-SDP			0	0.01						
35	hornD4	None	0; 0; 20	126				0	1.77e−07, 0.00e+00	1.02e−06	2.04	
		pd1			0	0.02	0.00					
		pd2			0	0.04	0.00					
		dd1			0	0.02	0.00					
		dd2	105; 0; 14	126	1	0.06	0.01	0	7.49e−08, 0.00e+00	2.38e−07	1.12	
		Sieve-SDP			0	0.02						
36	hornD5	None	0; 0; 35	420				0	2.32e−08, 0.00e+00	1.83e−07	2.06	
		pd1			0	0.03	0.00					
		pd2			0	0.04	0.00					
		dd1			0	0.03	0.00					
		dd2	305; 0; 25	420	1	0.09	0.03	0	5.58e−10, 0.00e+00	2.00e−09	1.26	
		Sieve-SDP			0	0.04						
37	hybridLyap	None	860; 0; 6, 108, 11_{10}	3093				0	0.00e+00, 7.29e−07	2.11e−04	7.85	
		pd1	860; 0; 6, 56, 11, 1_2, 11, 1_2, 11_2	1607	1	0.16	0.09	0	0.00e+00, 3.48e−07	6.61e−05	1.49	
		pd2	860; 0; 6, 34, 8, 1_2, 8, 1_2, 9, 7	1173	1	1.02	0.05	0	0.00e+00, 4.24e−09	4.86e−07	1.23	2
		dd1			0	0.05	0.00					
		dd2			0	0.14	0.00					
		Sieve-SDP			0	0.05						

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
38	leverage_limit	None	0; 18100; $151_{100}, 30_{100}$	68195				0	−8.75e+01, −8.75e+01	1.53e−05	278.60	
		pd1			0	2.10	0.17					
		pd2	0; 18100; $151_{99}, 121, 30_{100}$	67700	1	120.98	7.87	0	−8.75e+01, −8.75e+01	5.63e−06	150.78	3
		dd1	958500; 18100; $61_{100}, 30_{100}$	68195	1	3.87	7.20	0	−8.75e+01, −8.75e+01	2.45e−05	250.27	
		dd2	1193505; 18100; $1_{99}, 31$	68195	1	291.58	1.39	−1	−3.35e+00, 0.00e+00	1.03e+01	1.97	−2
		Sieve-SDP	0; 18100; $143_{97}, 141_3, 26_{98}, 25_2$	56196	1	253.43		0	−8.74e+01, −8.74e+01	1.73e−05	179.26	3
39	long_only	None	0; 9000; $91_{100}, 30_{100}$	59095				0	−4.13e+01, −4.13e+01	5.23e−06	373.38	
		pd1			0	1.18	0.17					
		pd2	0; 9000; $91_{99}, 61, 30_{100}$	58600	1	24.33	6.91	0	−4.13e+01, −4.13e+01	4.64e−07	205.50	2, 3
		dd1	229500; 9000; $61_{100}, 30_{100}$	59095	1	1.77	6.96	0	−4.13e+01, −4.13e+01	6.47e−06	246.03	3
		dd2	229500; 9000; $61_{100}, 30_{100}$	59095	1	531.60	6.94	0	−4.13e+01, −4.13e+01	2.80e−06	315.18	3
		Sieve-SDP	0; 8573; $83_{97}, 81_3, 26_{98}, 25_2$	46670	1	190.92		0	−4.13e+01, −4.13e+01	1.64e−06	94.12	3
40	sector_neutral	None	0; 12000; $121_{100}, 30_{100}$	62392				0	−1.21e+02, −1.21e+02	8.35e−05	152.27	
		pd1			0	1.84	0.26					
		pd2	0; 12000; $121_{99}, 91, 30_{100}$	61897	1	183.96	7.17	0	−1.21e+02, −1.21e+02	2.79e−04	150.19	3
		dd1	549000; 12000; $61_{100}, 30_{100}$	62392	1	2.79	7.05	0	−1.21e+02, −1.21e+02	8.23e−05	154.78	3
		dd2	549000; 12000; $61_{100}, 30_{100}$	62392	1	217.62	7.13	0	−1.21e+02, −1.21e+02	1.24e−04	140.83	3
		Sieve-SDP	0; 12000; $121_{99}, 111, 30_{100}$	62247	1	52.25		0	−1.21e+02, −1.21e+02	1.76e−04	151.52	3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
41	unconstrained	None	0; 12000; $121_{100}, 30_{100}$	62095				0	− 1.33e+02, − 1.33e+02	7.89e−05	279.82	
		pd1			0	1.52	0.15					
		pd2	0; 12000; $121_{99}, 91, 30_{100}$	61600	1	38.70	6.95	0	− 1.33e+02, − 1.33e+02	1.42e−05	282.87	3
		dd1	549000; 12000; $61_{100}, 30_{100}$	62095	1	2.63	6.74	0	− 1.33e+02, − 1.33e+02	3.34e−05	258.89	3
		dd2	549000; 12000; $61_{100}, 30_{100}$	62095	1	505.61	6.70	0	− 1.33e+02, − 1.33e+02	3.64e−05	260.11	3
		Sieve-SDP	0; 12000; $113_{97}, 111_3, 26_{98}, 25_2$	50097	1	213.04		0	− 1.28e+02, − 1.28e+02	1.64e−05	185.98	3
42	unboundDim 1R1	None	0; 2; 2	2				0	1.33e−09, − 7.05e−10	4.38e−09	2.89	
		pd1			0	0.05	0.01					
		pd2			0	0.03	0.01					
		dd1			0	0.03	0.01					
		dd2			0	0.02	0.01					
		Sieve-SDP	0; 1; 1	1	1	0.01		0	0.00e+00, 0.00e+00	0.00e+00	2.25	
43	unboundDim1R2	None	0; 0; 3, 2_2	4				0	− 8.91e−15, − 8.01e−15	7.07e−01	4.52	
		pd1	0; 0; 1_2	1	1	0.11	0.01	0	0.00e+00, 0.00e+00	0.00e+00	0.43	2
		pd2	0; 0; 1_2	1	1	0.10	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.47	2
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.00		0	0.00e+00, 0.00e+00	0.00e+00	2.39	2
44	unboundDim1R3	None	0; 0; 4, 3_2	6				0	− 2.04e−11, − 2.02e−11	7.07e−01	4.17	
		pd1	0; 0; 1_2	1	1	0.12	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.50	2
		pd2	0; 0; 1_2	1	1	0.11	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.47	2
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.01		0	0.00e+00, 0.00e+00	0.00e+00	2.31	2

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
45	unboundDim1R4	None	0; 0; 5, 4_2	8				0	− 2.34e−10, − 2.32e−10	7.07e−01	3.79	
		pd1	0; 0; 1_2	1	1	0.14	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.43	2
		pd2	0; 0; 1_2	1	1	0.14	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.43	2
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.01		0	0.00e+00, 0.00e+00	0.00e+00	2.41	2
46	unboundDim1R5	None	0; 0; 6, 5_2	10				0	− 1.00e+00, − 1.00e+00	9.88e−08	2.74	
		pd1	0; 0; 1_2	1	1	0.16	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.43	3
		pd2	0; 0; 1_2	1	1	0.20	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.42	3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.01		0	0.00e+00, 0.00e+00	0.00e+00	2.32	3
47	unboundDim1R6	None	0; 0; 7, 6_2	12				0	− 1.00e+00, − 1.00e+00	2.15e−07	2.78	
		pd1	0; 0; 1_2	1	1	0.20	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.50	3
		pd2	0; 0; 1_2	1	1	0.22	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.49	3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.01		0	0.00e+00, 0.00e+00	0.00e+00	2.24	3
48	unboundDim1R7	None	0; 0; 8, 7_2	14				0	− 1.00e+00, − 1.00e+00	5.11e−08	2.82	
		pd1	0; 0; 1_2	1	1	0.21	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.45	3
		pd2	0; 0; 1_2	1	1	0.23	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.44	3
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.02		0	0.00e+00, 0.00e+00	0.00e+00	2.32	3
49	unboundDim1R8	None	0; 0; 9, 8_2	16				0	− 1.00e+00, − 1.00e+00	5.43e−08	2.29	
		pd1	0; 0; 1_2	1	1	0.52	0.02	0	0.00e+00, 0.00e+00	0.00e+00	0.44	3
		pd2	0; 0; 1_2	1	1	0.53	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.45	3
		dd1			0	0.07	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.13		0	0.00e+00, 0.00e+00	0.00e+00	1.74	3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
50	unboundDim1R9	None	0; 0; 10, 9_2	18				0	− 1.00e+00, − 1.00e+00	6.50e−08	2.09	
		pd1	0; 0; 1_2	1	1	0.31	0.01	0	0.00e+00, 0.00e+00	0.00e+00	0.43	3
		pd2	0; 0; 1_2	1	1	0.30	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.64	3
		dd1			0	0.04	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.05		0	0.00e+00, 0.00e+00	0.00e+00	1.60	3
51	unboundDim1R10	None	0; 0; 11, 10_2	20				0	− 1.00e+00, − 1.00e+00	1.41e−07	2.76	
		pd1	0; 0; 1_2	1	1	0.28	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.44	3
		pd2	0; 0; 1_2	1	1	0.32	0.00	0	0.00e+00, 0.00e+00	0.00e+00	0.45	3
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 1_2	1	1	0.04		0	0.00e+00, 0.00e+00	0.00e+00	2.33	3
52	vamos_5_34	None	0; 0; 52	721				0	0.00e+00, − 4.18e−09	5.21e−08	2.10	
		pd1			0	0.07	0.00					
		pd2				MM						MM
		dd1			0	0.05	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP			0	0.06						
53	wei_wagner_F7_minus_4	None	0; 0; 8	31				0	0.00e+00, − 9.60e−13	1.11e−11	1.87	
		pd1			0	0.02	0.00					
		pd2	0; 0; 5	14	1	0.08	0.01	0	0.00e+00, − 5.80e−11	2.12e−10	0.99	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			0	0.01						
54	wei_wagner_P7	None	0; 0; 8	32				0	0.00e+00, − 1.46e−08	9.09e−08	1.99	
		pd1			0	0.02	0.00					
		pd2	0; 0; 4	10	1	0.05	0.00	0	0.00e+00, − 3.02e−10	1.31e−09	1.04	
		dd1			0	0.03	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			0	0.01						

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
55	wei_wagner_W3Plus	None	0; 0; 8	31				0	0.00e+00, −6.06e−09	5.47e−08	1.95	
		pd1			0	0.02	0.00					
		pd2	0; 0; 3	6	1	0.05	0.00	0	0.00e+00, −4.77e−09	1.11e−08	1.01	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP			0	0.00						
56	wei_wagner_W3_PlusE	None	0; 0; 9	38				0	0.00e+00, −9.18e−09	5.53e−08	1.98	
		pd1			0	0.02	0.00					
		pd2	0; 0; 5	15	1	0.06	0.00	0	0.00e+00, −7.21e−09	3.21e−08	1.02	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			0	0.00						
57	wei_wagner_nP_minus_1_24	None	0; 0; 12	64				0	0.00e+00, −5.50e−09	8.80e−08	2.03	
		pd1			0	0.02	0.00					
		pd2	0; 0; 6	21	1	0.07	0.00	0	0.00e+00, −1.08e−11	5.60e−11	1.02	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			0	0.01						
58	wei_wagner_nP_minus_9_12	None	0; 0; 12	64				0	0.00e+00, −3.92e−09	4.87e−08	1.98	
		pd1			0	0.02	0.00					
		pd2	0; 0; 5	15	1	0.05	0.00	0	0.00e+00, −4.11e−15	2.34e−14	1.02	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP			0	0.01						
59	wei_wagner_vamos_12	None	0; 0; 16	103				0	0.00e+00, −1.59e−08	1.38e−07	2.10	
		pd1			0	0.02	0.00					
		pd2	0; 0; 13	74	1	0.06	0.01	0	0.00e+00, −2.54e−10	1.62e−09	1.03	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP			0	0.01						

A.2 Detailed results on the Mittelmann dataset

This dataset has 31 problems. From these 8 problems were reduced by at least one of the five methods. There were 5 problems on which pd2 or dd2 ran out of memory or crashed.

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
1	diamond_patch	None	0; 0; 5477	5478				0	1.63e+01, 1.63e+01	3.56e−04	10854.97	
		pd1			0	31.05	0.06					
		pd2				MM						MM
		dd1			0	27.94	0.06					
		dd2			0	3008.86	0.06					
		Sieve-SDP			0	1.12						
2	e_moment_stable_17_0.5_2_2	None	0; 342; 171, 18_{17}	5984				0	− 1.98e−01, − 1.98e−01	1.14e−05	38.53	
		pd1	0; 342; 18_{18}	1139	1	0.71	0.16	0	− 1.98e−01, − 1.98e−01	8.44e−06	1.64	
		pd2	0; 342; 18_{18}	1139	1	0.86	0.13	0	− 1.98e−01, − 1.98e−01	8.44e−06	1.66	
		dd1			0	0.08	0.01					
		dd2			0	0.34	0.01					
		Sieve-SDP	0; 342; 18_{18}	1139	1	0.52		0	− 1.98e−01, − 1.98e−01	8.44e−06	1.63	
3	ice_2.0	None	0; 0; 8113	8113				0	6.81e+03, 6.81e+03	4.58e−07	17680.82	
		pd1			0	65.58	0.01					
		pd2				MM						MM
		dd1			0	64.03	0.01					
		dd2				MM						MM
		Sieve-SDP			0	0.80						
4	G60_mb	None	0; 0; 7000	7001				0	1.93e+03, 1.93e+03	6.64e−05	29138.79	
		pd1			0	107.66	10.87					
		pd2				MM						MM
		dd1			0	72.76	10.87					
		dd2				MM						MM
		Sieve-SDP			0	22.42						

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
5	maxG60	None	0; 0; 7000	7000				0	− 1.52e+04, − 1.52e+04	6.73e−07	5217.88	
		pd1			0	47.47	0.01					
		pd2				MM						MM
		dd1			0	45.65	0.01					
		dd2				MM						MM
		Sieve-SDP			0	0.47						
6	neu3	None	0; 2; 418	7364				0	7.10e−08, 1.12e−08	2.01e−06	153.03	
		pd1	0; 2; 87	1152	1	0.94	0.11	0	4.69e−08, 3.50e−08	1.94e−07	3.01	2
		pd2	0; 2; 87	1152	1	5.41	0.10	0	4.69e−08, 3.50e−08	1.94e−07	2.97	2
		dd1			0	0.16	0.02					
		dd2			0	2.29	0.02					
		Sieve-SDP	0; 2; 87	1152	1	2.34		0	4.69e−08, 3.50e−08	1.94e−07	2.99	2
7	neu3g	None	0; 0; 462	8007				0	4.58e−08, − 2.89e−09	8.67e−07	151.22	
		pd1	0; 0; 87	1151	1	1.32	0.11	0	8.91e−08, 5.65e−08	2.91e−07	3.00	
		pd2	0; 0; 87	1151	1	10.68	0.11	0	8.91e−08, 5.65e−08	2.91e−07	3.09	
		dd1			0	0.19	0.03					
		dd2			0	2.66	0.03					
		Sieve-SDP	0; 0; 87	1151	1	2.26		0	8.91e−08, 5.65e−08	2.91e−07	3.03	
8	p_auss2_3.0	None	0; 0; 9115	9115				0	8.62e+03, 8.62e+03	2.36e−07	25651.19	
		pd1			0	93.91	0.02					
		pd2				MM						MM
		dd1			0	97.11	0.02					
		dd2				MM						MM
		Sieve-SDP			0	0.76						
9	rose13	None	0; 0; 105	2379				0	1.20e+01, 1.20e+01	1.65e−06	7.63	
		pd1	0; 0; 92	1911	1	0.11	0.14	0	1.20e+01, 1.20e+01	4.86e−07	5.26	
		pd2	0; 0; 80	1523	1	0.51	0.11	0	1.20e+01, 1.20e+01	1.98e−07	2.94	
		dd1			0	0.05	0.01					
		dd2			0	0.12	0.01					
		Sieve-SDP	0; 0; 92	1911	1	0.39		0	1.20e+01, 1.20e+01	4.86e−07	5.28	

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
10	rose15	None	0; 2; 135	3860				0	− 3.11e−06, − 2.94e−06	1.83e−05	19.47	
		pd1	0; 2; 121	3181	1	0.08	0.24	0	− 3.52e−07, − 1.52e−07	5.07e−05	11.73	3
		pd2	0; 2; 107	2593	1	0.66	0.19	0	− 1.59e−09, − 1.57e−09	1.10e−08	5.74	2, 3
		dd1			0	0.07	0.00					
		dd2			0	0.18	0.00					
		Sieve-SDP	0; 2; 121	3181	1	0.52		0	− 3.52e−07, − 1.52e−07	5.07e−05	11.71	3
11	taha1a	None	0; 0; 252, 56_3, 126_{10}	3002				0	− 1.00e+00, − 1.00e+00	9.39e−07	37.54	
		pd1	0; 0; 126, 56_3, 126_{10}	2001	1	10.57	0.72	0	− 1.00e+00, − 1.00e+00	1.20e−07	21.55	
		pd2	0; 0; 126, 56_3, 126_{10}	2001	1	18.98	0.75	0	− 1.00e+00, − 1.00e+00	1.20e−07	21.50	
		dd1			0	0.21	0.06					
		dd2			0	21.47	0.06					
		Sieve-SDP	0; 0; 126, 56_3, 126_{10}	2001	1	1.75		0	− 1.00e+00, − 1.00e+00	1.20e−07	21.70	
12	taha1b	None	0; 3; 286, 66_{20}	8007				0	− 7.73e−01, − 7.73e−01	1.59e−07	148.99	
		pd1	0; 3; 66_{21}	3002	1	13.97	0.87	0	− 7.73e−01, − 7.73e−01	1.32e−07	34.29	
		pd2	0; 3; 66_{21}	3002	1	18.37	0.85	0	− 7.73e−01, − 7.73e−01	1.32e−07	33.03	
		dd1			0	0.16	0.04					
		dd2			0	1.82	0.04					
		Sieve-SDP	0; 3; 66_{21}	3002	1	1.97		0	− 7.73e−01, − 7.73e−01	1.32e−07	32.97	
13	taha1c	None	0; 0; 462, 126_3, 252_{10}	6187				0	− 1.00e+00, − 1.00e+00	3.12e−07	314.61	
		pd1	0; 0; 252, 126_3, 252_{10}	4367	1	148.36	2.11	0	− 1.00e+00, − 1.00e+00	4.37e−07	178.22	
		pd2	0; 0; 252, 126_3, 252_{10}	4367	1	187.99	2.01	0	− 1.00e+00, − 1.00e+00	4.37e−07	177.80	
		dd1			0	0.75	0.25					
		dd2			0	156.72	0.25					
		Sieve-SDP	0; 0; 252, 126_3, 252_{10}	4367	1	10.85		0	− 1.00e+00, − 1.00e+00	4.37e−07	182.86	

A.3 Detailed results on the Dressler-Illiman-de Wolff (DIW) dataset

This is a collection of 155 SDP relaxations from polynomial optimization generated by Gloptipoly 3 based on the paper [11].

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
1	ex3.3_order4	None	0; 1; 15	44				0	3.54e−10, 3.56e−10	6.87e−01	1.69	
		pd1	0; 1; 2	3	1	0.10	0.00	1	0.00e+00, 5.00e−01	5.27e−01	0.63	1
		pd2	0; 1; 2	3	1	0.35	0.00	1	0.00e+00, 5.00e−01	5.27e−01	0.73	1
		dd1			0	0.04	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			inf	0.03					0.00	1
2	ex3.3_order5	None	0; 0; 21, 3	65				0	6.16e−02, 6.16e−02	1.18e−06	1.25	
		pd1	0; 0; 2, 1	3	1	0.15	0.00	1	0.00e+00, 5.00e−01	5.27e−01	0.64	1
		pd2	0; 0; 2, 1	3	1	0.15	0.00	1	0.00e+00, 5.00e−01	5.27e−01	0.75	1
		dd1			0	0.04	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			inf	0.03					0.00	1
3	ex3.3_order6	None	0; 0; 28, 6	90				0	6.16e−02, 6.16e−02	1.60e−06	1.00	
		pd1	0; 0; 8, 2	22	1	0.10	0.00	0	6.16e−02, 6.16e−02	4.23e−08	1.08	2
		pd2	0; 0; 8, 2	22	1	0.16	0.00	0	6.16e−02, 6.16e−02	4.23e−08	0.69	2
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 8, 2	22	1	0.03		0	6.16e−02, 6.16e−02	4.23e−08	0.69	2
4	ex3.3_order7	None	0; 0; 36, 10	119				0	6.16e−02, 6.16e−02	3.07e−06	0.65	
		pd1	0; 0; 10, 4	31	1	0.13	0.00	0	6.16e−02, 6.16e−02	6.03e−08	0.59	2, 3
		pd2	0; 0; 10, 4	31	1	0.24	0.00	0	6.16e−02, 6.16e−02	6.03e−08	0.65	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 10, 4	31	1	0.05		0	6.16e−02, 6.16e−02	6.03e−08	0.59	2, 3
5	ex3.3_order8	None	0; 0; 45, 15	152				0	6.16e−02, 6.16e−02	2.45e−06	1.46	
		pd1	0; 0; 12, 6	38	1	0.25	0.01	0	6.16e−02, 6.16e−02	5.23e−08	0.65	2, 3
		pd2	0; 0; 12, 6	38	1	0.37	0.01	0	6.16e−02, 6.16e−02	5.23e−08	0.67	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP	0; 0; 12, 6	38	1	0.07		0	6.16e−02, 6.16e−02	5.23e−08	0.58	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
6	ex3.3_order9	None	0; 0; 55, 21	189				0	6.16e−02, 6.16e−02	4.63e−06	0.94	
		pd1	0; 0; 13, 7	41	1	0.18	0.01	0	6.16e−02, 6.16e−02	5.54e−08	0.60	2, 3
		pd2	0; 0; 13, 7	41	1	0.36	0.01	0	6.16e−02, 6.16e−02	5.54e−08	0.61	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP	0; 0; 13, 7	41	1	0.09		0	6.16e−02, 6.16e−02	5.54e−08	0.69	2, 3
7	ex3.3_order10	None	0; 0; 66, 28	230				0	6.16e−02, 6.16e−02	4.33e−06	1.29	
		pd1	0; 0; 14, 8	44	1	0.33	0.03	0	6.16e−02, 6.16e−02	6.73e−08	1.06	2, 3
		pd2	0; 0; 14, 8	44	1	0.56	0.01	0	6.16e−02, 6.16e−02	6.73e−08	1.00	2, 3
		dd1			0	0.06	0.00					
		dd2			0	0.08	0.00					
		Sieve-SDP	0; 0; 14, 8	44	1	0.15		0	6.16e−02, 6.16e−02	6.73e−08	1.24	2, 3
8	ex3.3_order11	None	0; 0; 78, 36	275				0	6.16e−02, 6.16e−02	1.12e−05	1.38	
		pd1	0; 0; 16, 10	53	1	0.31	0.01	0	6.16e−02, 6.16e−02	9.05e−08	1.06	2, 3
		pd2	0; 0; 16, 10	53	1	0.70	0.01	0	6.16e−02, 6.16e−02	9.05e−08	1.06	2, 3
		dd1			0	0.04	0.00					
		dd2			0	0.10	0.00					
		Sieve-SDP	0; 0; 16, 10	53	1	0.18		0	6.16e−02, 6.16e−02	9.05e−08	1.22	2, 3
9	ex3.3_order12	None	0; 0; 91, 45	324				0	6.16e−02, 6.16e−02	2.34e−05	1.50	
		pd1	0; 0; 18, 12	60	1	0.38	0.01	0	6.16e−02, 6.16e−02	1.12e−07	1.38	2, 3
		pd2	0; 0; 18, 12	60	1	1.17	0.01	0	6.16e−02, 6.16e−02	1.12e−07	0.63	2, 3
		dd1			0	0.04	0.00					
		dd2			0	0.11	0.00					
		Sieve-SDP	0; 0; 18, 12	60	1	0.22		0	6.16e−02, 6.16e−02	1.12e−07	0.68	2, 3
10	ex3.3_order13	None	0; 0; 105, 55	377				0	6.15e−02, 6.15e−02	3.47e−05	1.38	
		pd1	0; 0; 19, 13	63	1	0.45	0.01	0	6.16e−02, 6.16e−02	1.21e−07	0.61	2, 3
		pd2	0; 0; 19, 13	63	1	1.50	0.01	0	6.16e−02, 6.16e−02	1.21e−07	0.59	2, 3
		dd1			0	0.03	0.01					
		dd2			0	0.15	0.01					
		Sieve-SDP	0; 0; 19, 13	63	1	0.28		0	6.16e−02, 6.16e−02	1.21e−07	0.57	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
11	ex3.3_order14	None	0; 0; 120, 66	434				0	6.16e−02, 6.16e−02	1.41e−05	1.50	
		pd1	0; 0; 20, 14	66	1	0.58	0.01	0	6.16e−02, 6.16e−02	1.31e−07	0.57	2, 3
		pd2	0; 0; 20, 14	66	1	2.07	0.01	0	6.16e−02, 6.16e−02	1.31e−07	1.39	2, 3
		dd1			0	0.05	0.00					
		dd2			0	0.22	0.00					
		Sieve-SDP	0; 0; 20, 14	66	1	0.38		0	6.16e−02, 6.16e−02	1.31e−07	1.27	2, 3
12	ex3.3_order15	None	0; 0; 136, 78	495				0	6.16e−02, 6.16e−02	1.06e−05	2.72	
		pd1	0; 0; 22, 16	75	1	0.73	0.01	0	6.16e−02, 6.16e−02	1.47e−07	1.06	2, 3
		pd2	0; 0; 22, 16	75	1	2.89	0.02	0	6.16e−02, 6.16e−02	1.47e−07	1.06	2, 3
		dd1			0	0.05	0.00					
		dd2			0	0.24	0.00					
		Sieve-SDP	0; 0; 22, 16	75	1	0.47		0	6.16e−02, 6.16e−02	1.47e−07	1.10	2, 3
13	ex3.3_order16	None	0; 0; 153, 91	560				0	6.16e−02, 6.16e−02	1.18e−05	3.64	
		pd1	0; 0; 24, 18	82	1	0.92	0.01	0	6.16e−02, 6.16e−02	1.79e−07	0.95	2, 3
		pd2	0; 0; 24, 18	82	1	3.88	0.01	0	6.16e−02, 6.16e−02	1.79e−07	0.92	2, 3
		dd1			0	0.05	0.00					
		dd2			0	0.29	0.00					
		Sieve-SDP	0; 0; 24, 18	82	1	0.58		0	6.16e−02, 6.16e−02	1.79e−07	0.93	2, 3
14	ex3.3_order17	None	0; 0; 171, 105	629				0	6.15e−02, 6.15e−02	2.84e−05	4.62	
		pd1	0; 0; 25, 19	85	1	1.16	0.02	0	6.16e−02, 6.16e−02	1.89e−07	0.94	2, 3
		pd2	0; 0; 25, 19	85	1	5.37	0.02	0	6.16e−02, 6.16e−02	1.89e−07	0.94	2, 3
		dd1			0	0.06	0.01					
		dd2			0	0.41	0.01					
		Sieve-SDP	0; 0; 25, 19	85	1	0.73		0	6.16e−02, 6.16e−02	1.89e−07	0.92	2, 3
15	ex3.3_order18	None	0; 0; 190, 120	702				0	6.15e−02, 6.15e−02	6.51e−05	6.29	
		pd1	0; 0; 26, 20	88	1	1.55	0.02	0	6.16e−02, 6.16e−02	1.98e−07	0.93	2, 3
		pd2	0; 0; 26, 20	88	1	7.78	0.02	0	6.16e−02, 6.16e−02	1.98e−07	0.58	2, 3
		dd1			0	0.05	0.01					
		dd2			0	0.50	0.01					
		Sieve-SDP	0; 0; 26, 20	88	1	0.94		0	6.16e−02, 6.16e−02	1.98e−07	0.61	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
16	ex3.3_order19	None	0; 0; 210, 136	779				0	6.15e−02, 6.15e−02	2.92e−04	10.01	
		pd1	0; 0; 28, 22	97	1	2.18	0.02	0	6.16e−02, 6.16e−02	2.10e−07	1.22	2, 3
		pd2	0; 0; 28, 22	97	1	10.34	0.03	0	6.16e−02, 6.16e−02	2.10e−07	0.69	2, 3
		dd1			0	0.06	0.01					
		dd2			0	0.60	0.01					
		Sieve-SDP	0; 0; 28, 22	97	1	1.31		0	6.16e−02, 6.16e−02	2.10e−07	0.67	2, 3
17	ex3.3_order20	None	0; 0; 231, 153	860				0	6.15e−02, 6.15e−02	2.95e−04	21.36	
		pd1	0; 0; 30, 24	104	1	2.68	0.03	0	6.16e−02, 6.16e−02	2.16e−07	1.10	2, 3
		pd2	0; 0; 30, 24	104	1	13.41	0.03	0	6.16e−02, 6.16e−02	2.16e−07	1.09	2, 3
		dd1			0	0.09	0.02					
		dd2			0	0.86	0.02					
		Sieve-SDP	0; 0; 30, 24	104	1	1.35		0	6.16e−02, 6.16e−02	2.16e−07	1.31	2, 3
18	ex4.1_order3	None	0; 1; 10	27				0	1.24e−09, 1.30e−09	8.29e−01	1.20	
		pd1	0; 1; 1	2	1	0.07	0.00	1	1.18e−12, 4.80e+00	8.28e−01	0.79	1
		pd2	0; 1; 1	2	1	0.06	0.00	1	1.18e−12, 4.80e+00	8.28e−01	0.62	1
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP			inf	0.01					0.00	1
19	ex4.1_order4	None	0; 0; 15, 3	44				0	2.55e−09, 2.60e−09	8.29e−01	0.84	
		pd1	0; 0; 1	1	1	0.10	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.35	1
		pd2	0; 0; 1	1	1	0.08	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.66	1
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			inf	0.02					0.00	1
20	ex4.1_order5	None	0; 0; 21, 6	65				0	3.08e−09, 3.12e−09	8.29e−01	0.94	
		pd1	0; 0; 1	1	1	0.10	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.39	1
		pd2	0; 0; 1	1	1	0.11	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.34	1
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			inf	0.02					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
21	ex4.1_order6	None	0; 0; 28, 10	90				0	1.00e+00, 1.00e+00	6.31e−07	0.85	
		pd1	0; 0; 1	1	1	0.14	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.39	1
		pd2	0; 0; 1	1	1	0.15	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.36	1
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			inf	0.02					0.00	1
22	ex4.1_order7	None	0; 0; 36, 15	119				0	1.00e+00, 1.00e+00	1.00e−06	0.61	
		pd1	0; 0; 1	1	1	0.17	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.34	1
		pd2	0; 0; 1	1	1	0.21	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.61	1
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			inf	0.04					0.00	1
23	ex4.1_order8	None	0; 0; 45, 21	152				0	1.00e+00, 1.00e+00	1.42e−06	0.68	
		pd1	0; 0; 1	1	1	0.19	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.39	1
		pd2	0; 0; 1	1	1	0.27	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.35	1
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			inf	0.05					0.00	1
24	ex4.1_order9	None	0; 0; 55, 28	189				0	1.00e+00, 1.00e+00	1.10e−06	0.87	
		pd1	0; 0; 1	1	1	0.26	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.39	1
		pd2	0; 0; 1	1	1	0.45	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.39	1
		dd1			0	0.02	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP			inf	0.07					0.00	1
25	ex4.1_order10	None	0; 0; 66, 36	230				0	1.00e+00, 1.00e+00	9.94e−07	0.69	
		pd1	0; 0; 1	1	1	0.32	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.39	1
		pd2	0; 0; 1	1	1	0.50	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.33	1
		dd1			0	0.02	0.00					
		dd2			0	0.06	0.00					
		Sieve-SDP			inf	0.09					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
26	ex4.1_order11	None	0; 0; 78, 45	275				0	1.00e+00, 1.00e+00	2.60e−06	0.93	
		pd1	0; 0; 1	1	1	0.33	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.38	1
		pd2	0; 0; 1	1	1	0.71	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.34	1
		dd1			0	0.02	0.00					
		dd2			0	0.09	0.00					
		Sieve-SDP			inf	0.13					0.00	1
27	ex4.1_order12	None	0; 0; 91, 55	324				0	1.00e+00, 1.00e+00	2.29e−06	1.06	
		pd1	0; 0; 1	1	1	0.50	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.34	1
		pd2	0; 0; 1	1	1	1.02	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.41	1
		dd1			0	0.02	0.00					
		dd2			0	0.11	0.00					
		Sieve-SDP			inf	0.16					0.00	1
28	ex4.1_order13	None	0; 0; 105, 66	377				0	1.00e+00, 1.00e+00	6.83e−06	1.00	
		pd1	0; 0; 1	1	1	0.52	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.77	1
		pd2	0; 0; 1	1	1	1.55	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.37	1
		dd1			0	0.03	0.00					
		dd2			0	0.14	0.00					
		Sieve-SDP			inf	0.19					0.00	1
29	ex4.1_order14	None	0; 0; 120, 78	434				0	1.00e+00, 1.00e+00	2.19e−06	1.59	
		pd1	0; 0; 1	1	1	0.81	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.43	1
		pd2	0; 0; 1	1	1	2.20	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.38	1
		dd1			0	0.03	0.00					
		dd2			0	0.20	0.00					
		Sieve-SDP			inf	0.25					0.00	1
30	ex4.1_order15	None	0; 0; 136, 91	495				0	1.00e+00, 1.00e+00	4.32e−06	1.66	
		pd1	0; 0; 1	1	1	0.87	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.36	1
		pd2	0; 0; 1	1	1	2.79	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.35	1
		dd1			0	0.04	0.00					
		dd2			0	0.24	0.00					
		Sieve-SDP			inf	0.29					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
31	ex4.1_order16	None	0; 0; 153, 105	560				0	1.00e+00, 1.00e+00	7.99e−07	2.41	
		pd1	0; 0; 1	1	1	1.10	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.36	1
		pd2	0; 0; 1	1	1	4.06	0.00	1	0.00e+00, 3.00e+00	7.50e−01	0.53	1
		dd1			0	0.04	0.00					
		dd2			0	0.31	0.00					
		Sieve-SDP			inf	0.37					0.00	1
32	ex4.1_order17	None	0; 0; 171, 120	629				0	1.00e+00, 1.00e+00	1.45e−06	3.23	
		pd1	0; 0; 1	1	1	1.27	0.01	1	0.00e+00, 3.00e+00	7.50e−01	0.34	1
		pd2	0; 0; 1	1	1	5.29	0.01	1	0.00e+00, 3.00e+00	7.50e−01	0.37	1
		dd1			0	0.05	0.01					
		dd2			0	0.44	0.01					
		Sieve-SDP			inf	0.47					0.00	1
33	ex4.1_order18	None	0; 0; 190, 136	702				0	1.00e+00, 1.00e+00	2.43e−06	4.52	
		pd1	0; 0; 1	1	1	1.72	0.01	1	0.00e+00, 3.00e+00	7.50e−01	0.37	1
		pd2	0; 0; 1	1	1	6.99	0.01	1	0.00e+00, 3.00e+00	7.50e−01	0.46	1
		dd1			0	0.08	0.01					
		dd2			0	0.52	0.01					
		Sieve-SDP			inf	0.67					0.00	1
34	ex4.1_order19	None	0; 0; 210, 153	779				0	1.00e+00, 1.00e+00	2.76e−06	6.04	
		pd1	0; 0; 1	1	1	2.01	0.01	1	0.00e+00, 3.00e+00	7.50e−01	0.37	1
		pd2	0; 0; 1	1	1	9.39	0.01	1	0.00e+00, 3.00e+00	7.50e−01	0.37	1
		dd1			0	0.06	0.01					
		dd2			0	0.69	0.01					
		Sieve-SDP			inf	0.87					0.00	1
35	ex4.1_order20	None	0; 0; 231, 171	860				0	1.00e+00, 1.00e+00	5.65e−06	8.86	
		pd1	0; 0; 1	1	1	2.68	0.02	1	0.00e+00, 3.00e+00	7.50e−01	0.35	1
		pd2	0; 0; 1	1	1	12.71	0.02	1	0.00e+00, 3.00e+00	7.50e−01	0.34	1
		dd1			0	0.07	0.02					
		dd2			0	0.87	0.02					
		Sieve-SDP			inf	0.88					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
36	ex4.2_order4	None	0; 1; 15	44				0	1.00e−09, 1.01e−09	7.07e−01	0.81	
		pd1	0; 1; 1	2	1	0.08	0.00	0	1.00e+00, 1.00e+00	5.00e−01	0.57	3
		pd2	0; 1; 1	2	1	0.09	0.00	0	1.00e+00, 1.00e+00	5.00e−01	0.61	3
		dd1			0	0.04	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP			inf	0.03					0.00	1
37	ex4.2_order5	None	0; 0; 21, 3	65				0	5.53e−01, 5.53e−01	9.19e−08	1.85	
		pd1	0; 0; 1	1	1	0.15	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.65	1
		pd2	0; 0; 1	1	1	0.14	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.44	1
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			inf	0.03					0.00	1
38	ex4.2_order6	None	0; 0; 28, 6	90				0	5.53e−01, 5.53e−01	3.24e−07	0.65	
		pd1	0; 0; 8, 2	22	1	0.09	0.00	0	5.53e−01, 5.53e−01	7.64e−09	0.63	
		pd2	0; 0; 8, 2	22	1	0.14	0.00	0	5.53e−01, 5.53e−01	7.64e−09	0.62	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 8, 2	22	1	0.03		0	5.53e−01, 5.53e−01	7.64e−09	0.74	
39	ex4.2_order7	None	0; 0; 36, 10	119				0	5.53e−01, 5.53e−01	2.87e−07	0.65	
		pd1	0; 0; 9, 3	25	1	0.13	0.00	0	5.53e−01, 5.53e−01	4.89e−09	0.57	
		pd2	0; 0; 9, 3	25	1	0.16	0.00	0	5.53e−01, 5.53e−01	4.89e−09	0.55	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 9, 3	25	1	0.04		0	5.53e−01, 5.53e−01	4.89e−09	0.68	
40	ex4.2_order8	None	0; 0; 45, 15	152				0	5.53e−01, 5.53e−01	1.08e−06	0.64	
		pd1	0; 0; 10, 4	28	1	0.17	0.00	0	5.53e−01, 5.53e−01	3.90e−08	0.58	2
		pd2	0; 0; 10, 4	28	1	0.23	0.00	0	5.53e−01, 5.53e−01	3.90e−08	0.57	2
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP	0; 0; 10, 4	28	1	0.06		0	5.53e−01, 5.53e−01	3.90e−08	0.70	2

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
41	ex4.2_order9	None	0; 0; 55, 21	189				0	5.53e−01, 5.53e−01	1.10e−06	0.67	
		pd1	0; 0; 10, 4	28	1	0.22	0.00	0	5.53e−01, 5.53e−01	3.90e−08	0.58	2
		pd2	0; 0; 10, 4	28	1	0.33	0.00	0	5.53e−01, 5.53e−01	3.90e−08	0.57	2
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP	0; 0; 10, 4	28	1	0.08		0	5.53e−01, 5.53e−01	3.90e−08	0.71	2
42	ex4.2_order10	None	0; 0; 66, 28	230				0	5.53e−01, 5.53e−01	1.42e−06	0.74	
		pd1	0; 0; 10, 4	28	1	0.39	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.71	2, 3
		pd2	0; 0; 10, 4	28	1	0.56	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.60	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.06	0.00					
		Sieve-SDP	0; 0; 10, 4	28	1	0.13		0	5.53e−01, 5.53e−01	3.90e−08	0.76	2, 3
43	ex4.2_order11	None	0; 0; 78, 36	275				0	5.53e−01, 5.53e−01	1.61e−06	1.66	
		pd1	0; 0; 10, 4	28	1	0.49	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.80	2, 3
		pd2	0; 0; 10, 4	28	1	0.82	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.73	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.10	0.00					
		Sieve-SDP	0; 0; 10, 4	28	1	0.16		0	5.53e−01, 5.53e−01	3.90e−08	0.70	2, 3
44	ex4.2_order12	None	0; 0; 91, 45	324				0	5.53e−01, 5.53e−01	2.77e−06	1.23	
		pd1	0; 0; 10, 4	28	1	0.51	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.68	2, 3
		pd2	0; 0; 10, 4	28	1	1.06	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.57	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.12	0.00					
		Sieve-SDP	0; 0; 10, 4	28	1	0.19		0	5.53e−01, 5.53e−01	3.90e−08	0.70	2, 3
45	ex4.2_order13	None	0; 0; 105, 55	377				0	5.53e−01, 5.53e−01	4.58e−06	1.53	
		pd1	0; 0; 10, 4	28	1	0.51	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.68	2, 3
		pd2	0; 0; 10, 4	28	1	1.54	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.62	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.15	0.00					
		Sieve-SDP	0; 0; 10, 4	28	1	0.25		0	5.53e−01, 5.53e−01	3.90e−08	0.73	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
46	ex4.2_order14	None	0; 0; 120, 66	434				0	5.53e−01, 5.53e−01	1.61e−06	2.24	
		pd1	0; 0; 10, 4	28	1	0.82	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.73	2, 3
		pd2	0; 0; 10, 4	28	1	2.36	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.76	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.26	0.00					
		Sieve-SDP	0; 0; 10, 4	28	1	0.32		0	5.53e−01, 5.53e−01	3.90e−08	0.61	2, 3
47	ex4.2_order15	None	0; 0; 136, 78	495				0	5.53e−01, 5.53e−01	3.52e−06	2.90	
		pd1	0; 0; 10, 4	28	1	1.08	0.01	0	5.53e−01, 5.53e−01	3.90e−08	1.50	2, 3
		pd2	0; 0; 10, 4	28	1	3.81	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.86	2, 3
		dd1			0	0.04	0.01					
		dd2			0	0.34	0.01					
		Sieve-SDP	0; 0; 10, 4	28	1	0.59		0	5.53e−01, 5.53e−01	3.90e−08	0.57	2, 3
48	ex4.2_order16	None	0; 0; 153, 91	560				0	5.53e−01, 5.53e−01	2.54e−06	4.55	
		pd1	0; 0; 10, 4	28	1	1.50	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.67	2, 3
		pd2	0; 0; 10, 4	28	1	4.37	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.60	2, 3
		dd1			0	0.04	0.01					
		dd2			0	0.37	0.01					
		Sieve-SDP	0; 0; 10, 4	28	1	0.50		0	5.53e−01, 5.53e−01	3.90e−08	0.62	2, 3
49	ex4.2_order17	None	0; 0; 171, 105	629				0	5.52e−01, 5.52e−01	1.06e−04	4.96	
		pd1	0; 0; 10, 4	28	1	1.52	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.57	2, 3
		pd2	0; 0; 10, 4	28	1	5.71	0.01	0	5.53e−01, 5.53e−01	3.90e−08	0.63	2, 3
		dd1			0	0.05	0.01					
		dd2			0	0.50	0.01					
		Sieve-SDP	0; 0; 10, 4	28	1	0.59		0	5.53e−01, 5.53e−01	3.90e−08	0.62	2, 3
50	ex4.2_order18	None	0; 0; 190, 120	702				0	5.52e−01, 5.52e−01	1.70e−04	9.65	
		pd1	0; 0; 10, 4	28	1	2.01	0.02	0	5.53e−01, 5.53e−01	3.90e−08	0.62	2, 3
		pd2	0; 0; 10, 4	28	1	8.06	0.02	0	5.53e−01, 5.53e−01	3.90e−08	0.65	2, 3
		dd1			0	0.06	0.01					
		dd2			0	0.65	0.01					
		Sieve-SDP	0; 0; 10, 4	28	1	0.78		0	5.53e−01, 5.53e−01	3.90e−08	0.62	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
51	ex4.2_order19	None	0; 0; 210, 136	779				0	5.52e−01, 5.52e−01	9.26e−04	11.00	
		pd1	0; 0; 10, 4	28	1	2.40	0.02	0	5.53e−01, 5.53e−01	3.90e−08	0.59	2, 3
		pd2	0; 0; 10, 4	28	1	9.99	0.02	0	5.53e−01, 5.53e−01	3.90e−08	0.60	2, 3
		dd1			0	0.06	0.02					
		dd2			0	0.80	0.02					
		Sieve-SDP	0; 0; 10, 4	28	1	0.93		0	5.53e−01, 5.53e−01	3.90e−08	0.75	2, 3
52	ex4.2_order20	None	0; 0; 231, 153	860				0	5.49e−01, 5.49e−01	4.36e−03	17.65	
		pd1	0; 0; 10, 4	28	1	2.93	0.03	0	5.53e−01, 5.53e−01	3.90e−08	0.58	2, 3
		pd2	0; 0; 10, 4	28	1	12.77	0.03	0	5.53e−01, 5.53e−01	3.90e−08	0.57	2, 3
		dd1			0	0.07	0.02					
		dd2			0	0.94	0.02					
		Sieve-SDP	0; 0; 10, 4	28	1	1.11		0	5.53e−01, 5.53e−01	3.90e−08	0.68	2, 3
53	ex4.3_order2	None	0; 1; 10	34				0	1.73e−07, 1.83e−07	9.09e−01	4.39	
		pd1	0; 1; 4	10	1	0.14	0.03	1	0.00e+00, 8.00e+00	9.09e−01	1.24	1
		pd2	0; 1; 4	10	1	0.34	0.00	1	0.00e+00, 8.00e+00	9.09e−01	1.30	1
		dd1			0	0.08	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			inf	0.52					0.00	1
54	ex4.3_order3	None	0; 0; 20, 4	83				0	6.22e−09, 6.48e−09	9.09e−01	3.78	
		pd1	0; 0; 1	1	1	0.19	0.01	1	0.00e+00, 8.00e+00	8.89e−01	1.29	1
		pd2	0; 0; 1	1	1	0.13	0.00	1	0.00e+00, 8.00e+00	8.89e−01	1.28	1
		dd1			0	0.05	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP			inf	0.05					0.00	1
55	ex4.3_order4	None	0; 0; 35, 10	164				0	2.50e−09, 2.53e−09	9.09e−01	3.88	
		pd1	0; 0; 1	1	1	0.11	0.00	1	0.00e+00, 8.00e+00	8.89e−01	1.31	1
		pd2	0; 0; 1	1	1	0.13	0.00	1	0.00e+00, 8.00e+00	8.89e−01	1.29	1
		dd1			0	0.03	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP			inf	0.04					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
56	ex4.3_order5	None	0; 0; 56, 20	285				0	5.18e−09, 5.22e−09	9.09e−01	3.30	
		pd1	0; 0; 1	1	1	0.15	0.00	1	0.00e+00, 8.00e+00	8.89e−01	1.24	1
		pd2	0; 0; 1	1	1	0.20	0.00	1	0.00e+00, 8.00e+00	8.89e−01	1.22	1
		dd1			0	0.04	0.00					
		dd2			0	0.08	0.00					
		Sieve-SDP			inf	0.06					0.00	1
57	ex4.3_order6	None	0; 0; 84, 35	454				0	1.60e+01, 1.60e+01	3.34e−06	3.20	
		pd1	0; 0; 1	1	1	0.21	0.00	1	0.00e+00, 8.00e+00	8.89e−01	1.16	1
		pd2	0; 0; 1	1	1	0.41	0.00	1	0.00e+00, 8.00e+00	8.89e−01	1.20	1
		dd1			0	0.05	0.00					
		dd2			0	0.13	0.00					
		Sieve-SDP			inf	0.10					0.00	1
58	ex4.3_order7	None	0; 0; 120, 56	679				0	1.60e+01, 1.60e+01	5.26e−06	4.45	
		pd1	0; 0; 1	1	1	0.30	0.01	1	0.00e+00, 8.00e+00	8.89e−01	1.23	1
		pd2	0; 0; 1	1	1	0.83	0.01	1	0.00e+00, 8.00e+00	8.89e−01	1.19	1
		dd1			0	0.06	0.01					
		dd2			0	0.23	0.01					
		Sieve-SDP			inf	0.17					0.00	1
59	ex4.3_order8	None	0; 0; 165, 84	968				0	1.60e+01, 1.60e+01	5.17e−06	9.40	
		pd1	0; 0; 1	1	1	0.48	0.01	1	0.00e+00, 8.00e+00	8.89e−01	0.65	1
		pd2	0; 0; 1	1	1	2.05	0.01	1	0.00e+00, 8.00e+00	8.89e−01	0.71	1
		dd1			0	0.05	0.01					
		dd2			0	0.45	0.01					
		Sieve-SDP			inf	0.27					0.00	1
60	ex4.3_order9	None	0; 0; 220, 120	1329				0	1.60e+01, 1.60e+01	5.88e−06	17.62	
		pd1	0; 0; 1	1	1	0.85	0.02	1	0.00e+00, 8.00e+00	8.89e−01	0.56	1
		pd2	0; 0; 1	1	1	4.46	0.02	1	0.00e+00, 8.00e+00	8.89e−01	0.59	1
		dd1			0	0.07	0.02					
		dd2			0	0.88	0.02					
		Sieve-SDP			inf	0.53					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
61	ex4.3_order10	None	0; 0; 286, 165	1770				0	1.60e+01, 1.60e+01	3.92e−05	42.57	
		pd1	0; 0; 1	1	1	1.66	0.04	1	0.00e+00, 8.00e+00	8.89e−01	0.35	1
		pd2	0; 0; 1	1	1	9.01	0.04	1	0.00e+00, 8.00e+00	8.89e−01	0.34	1
		dd1			0	0.09	0.04					
		dd2			0	1.79	0.04					
		Sieve-SDP			inf	0.87					0.00	1
62	ex4.3_order11	None	0; 0; 364, 220	2299				0	7.85e−06, 7.85e−06	9.09e−01	116.27	
		pd1	0; 0; 1	1	1	3.12	0.07	1	0.00e+00, 8.00e+00	8.89e−01	0.34	1
		pd2	0; 0; 1	1	1	17.61	0.07	1	0.00e+00, 8.00e+00	8.89e−01	0.34	1
		dd1			0	0.16	0.07					
		dd2			0	18.38	0.07					
		Sieve-SDP			inf	1.51					0.00	1
63	ex4.3_order12	None	0; 0; 455, 286	2924				0	2.62e−06, 2.62e−06	9.09e−01	330.94	
		pd1	0; 0; 1	1	1	5.71	0.11	1	0.00e+00, 8.00e+00	8.89e−01	0.52	1
		pd2	0; 0; 1	1	1	34.17	0.11	1	0.00e+00, 8.00e+00	8.89e−01	0.36	1
		dd1			0	0.25	0.11					
		dd2			0	39.81	0.11					
		Sieve-SDP			inf	2.90					0.00	1
64	ex4.3_order13	None	0; 0; 560, 364	3653				0	4.85e−07, 4.85e−07	9.09e−01	814.60	
		pd1	0; 0; 1	1	1	10.54	0.18	1	0.00e+00, 8.00e+00	8.89e−01	0.34	1
		pd2	0; 0; 1	1	1	61.06	0.18	1	0.00e+00, 8.00e+00	8.89e−01	0.35	1
		dd1			0	0.40	0.18					
		dd2			0	74.87	0.18					
		Sieve-SDP			inf	5.29					0.00	1
65	ex4.3_order14	None	0; 0; 680, 455	4494				0	1.01e+01, 1.01e+01	9.40e−02	1178.45	
		pd1	0; 0; 1	1	1	17.38	0.27	1	0.00e+00, 8.00e+00	8.89e−01	0.34	1
		pd2	0; 0; 1	1	1	109.31	0.27	1	0.00e+00, 8.00e+00	8.89e−01	0.47	1
		dd1			0	0.63	0.27					
		dd2			0	146.45	0.27					
		Sieve-SDP			inf	9.73					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
66	ex4.3_order15	None	0; 0; 816, 560	5455				0	8.94e+00, 8.94e+00	1.76e−01	2010.36	
		pd1	0; 0; 1	1	1	33.32	0.41	1	0.00e+00, 8.00e+00	8.89e−01	0.50	1
		pd2	0; 0; 1	1	1	192.22	0.41	1	0.00e+00, 8.00e+00	8.89e−01	0.70	1
		dd1			0	0.99	0.41					
		dd2			0	303.20	0.41					
		Sieve-SDP			inf	16.20					0.00	1
67	ex4.3_order16	None	0; 0; 969, 680	6544				0	7.95e+00, 7.95e+00	2.23e−01	3158.88	
		pd1	0; 0; 1	1	1	46.39	0.55	1	0.00e+00, 8.00e+00	8.89e−01	1.43	1
		pd2	0; 0; 1	1	1	295.70	0.55	1	0.00e+00, 8.00e+00	8.89e−01	1.33	1
		dd1			0	1.49	0.55					
		dd2			0	485.30	0.55					
		Sieve-SDP			inf	29.21					0.00	1
68	ex4.3_order17	None	0; 0; 1140, 816	7769				0	7.45e+00, 7.45e+00	2.00e−01	5618.65	
		pd1	0; 0; 1	1	1	71.92	0.81	1	0.00e+00, 8.00e+00	8.89e−01	1.34	1
		pd2	0; 0; 1	1	1	472.15	0.81	1	0.00e+00, 8.00e+00	8.89e−01	1.33	1
		dd1			0	2.13	0.81					
		dd2			0	949.49	0.81					
		Sieve-SDP			inf	49.63					0.00	1
69	ex4.3_order18	None	0; 0; 1330, 969	9138				0	7.16e+00, 7.16e+00	2.14e−01	11769.31	
		pd1	0; 0; 1	1	1	112.46	1.13	1	0.00e+00, 8.00e+00	8.89e−01	1.40	1
		pd2	0; 0; 1	1	1	753.21	1.13	1	0.00e+00, 8.00e+00	8.89e−01	1.47	1
		dd1			0	3.05	1.13					
		dd2			0	1624.34	1.13					
		Sieve-SDP			inf	81.60					0.00	1
70	ex4.3_order19	None	0; 0; 1540, 1140	10659				0	6.82e+00, 6.82e+00	2.63e−01	22830.51	
		pd1	0; 0; 1	1	1	171.19	1.62	1	0.00e+00, 8.00e+00	8.89e−01	1.26	1
		pd2	0; 0; 1	1	1	1177.71	1.60	1	0.00e+00, 8.00e+00	8.89e−01	1.23	1
		dd1			0	4.53	1.60					
		dd2			0	2852.27	1.60					
		Sieve-SDP			inf	134.13					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
71	ex4.3_order20	None	0; 0; 1771, 1330	12340				0	6.52e+00, 6.52e+00	3.66e−01	38786.88	
		pd1	0; 0; 1	1	1	375.53	2.81	1	0.00e+00, 8.00e+00	8.89e−01	1.09	1
		pd2	0; 0; 1	1	1	2479.13	2.79	1	0.00e+00, 8.00e+00	8.89e−01	0.64	1
		dd1			0	6.68	2.79					
		dd2			0	6408.87	2.79					
		Sieve-SDP			inf	260.68					0.00	1
72	ex4.4_order3	None	0; 0; 20, 10	83				1	9.88e−02, 1.18e−01	8.66e−01	1.72	
		pd1	0; 0; 1	1	1	0.09	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.60	
		pd2	0; 0; 1	1	1	0.10	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.53	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP			inf	0.01					0.00	1
73	ex4.4_order4	None	0; 0; 35, 20	164				1	1.73e−05, 1.84e−05	8.66e−01	2.17	
		pd1	0; 0; 1	1	1	0.12	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.54	
		pd2	0; 0; 1	1	1	0.14	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.60	
		dd1			0	0.02	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP			inf	0.03					0.00	1
74	ex4.4_order5	None	0; 0; 56, 35	285				0	5.66e−08, 5.85e−08	8.66e−01	2.11	
		pd1	0; 0; 1	1	1	0.14	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.60	1
		pd2	0; 0; 1	1	1	0.25	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.59	1
		dd1			0	0.03	0.00					
		dd2			0	0.14	0.00					
		Sieve-SDP			inf	0.04					0.00	1
75	ex4.4_order6	None	0; 0; 84, 56	454				0	1.49e−08, 1.50e−08	8.66e−01	3.11	
		pd1	0; 0; 1	1	1	0.20	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.53	1
		pd2	0; 0; 1	1	1	0.47	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.55	1
		dd1			0	0.03	0.00					
		dd2			0	0.43	0.00					
		Sieve-SDP			inf	0.08					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
76	ex4.4_order7	None	0; 0; 120, 84	679				0	6.64e−09, 6.68e−09	8.66e−01	4.44	
		pd1	0; 0; 1	1	1	0.30	0.01	1	0.00e+00, 1.00e+00	5.00e−01	0.58	1
		pd2	0; 0; 1	1	1	0.96	0.01	1	0.00e+00, 1.00e+00	5.00e−01	0.57	1
		dd1			0	0.04	0.01					
		dd2			0	1.12	0.01					
		Sieve-SDP			inf	0.17					0.00	1
77	ex4.4_order8	None	0; 0; 165, 120	968				0	1.80e−09, 1.80e−09	8.66e−01	12.00	
		pd1	0; 0; 1	1	1	0.61	0.01	1	0.00e+00, 1.00e+00	5.00e−01	0.61	1
		pd2	0; 0; 1	1	1	2.09	0.01	1	0.00e+00, 1.00e+00	5.00e−01	0.55	1
		dd1			0	0.05	0.01					
		dd2			0	3.86	0.01					
		Sieve-SDP			inf	0.32					0.00	1
78	ex4.4_order9	None	0; 0; 220, 165	1329				0	4.13e−10, 4.14e−10	8.66e−01	31.08	
		pd1	0; 0; 1	1	1	1.18	0.03	1	0.00e+00, 1.00e+00	5.00e−01	0.80	1
		pd2	0; 0; 1	1	1	5.85	0.03	1	0.00e+00, 1.00e+00	5.00e−01	0.76	1
		dd1			0	0.12	0.03					
		dd2			0	8.18	0.03					
		Sieve-SDP			inf	0.53					0.00	1
79	ex4.4_order10	None	0; 0; 286, 220	1770				0	2.65e−10, 2.65e−10	8.66e−01	71.26	
		pd1	0; 0; 1	1	1	2.24	0.07	1	0.00e+00, 1.00e+00	5.00e−01	0.69	1
		pd2	0; 0; 1	1	1	10.10	0.06	1	0.00e+00, 1.00e+00	5.00e−01	0.56	1
		dd1			0	0.12	0.06					
		dd2			0	13.87	0.06					
		Sieve-SDP			inf	0.89					0.00	1
80	ex4.4_order11	None	0; 0; 364, 286	2299				0	1.60e−05, 1.61e−05	4.44e−05	64.15	
		pd1	0; 0; 1	1	1	4.21	0.10	1	0.00e+00, 1.00e+00	5.00e−01	0.62	1
		pd2	0; 0; 1	1	1	22.13	0.08	1	0.00e+00, 1.00e+00	5.00e−01	0.63	1
		dd1			0	0.20	0.08					
		dd2			0	30.60	0.08					
		Sieve-SDP			inf	1.77					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
81	ex4.4_order12	None	0; 0; 455, 364	2924				0	1.93e−07, 1.78e−07	2.52e−06	115.81	
		pd1	0; 0; 1	1	1	5.82	0.13	1	0.00e+00, 1.00e+00	5.00e−01	1.05	1
		pd2	0; 0; 1	1	1	32.88	0.13	1	0.00e+00, 1.00e+00	5.00e−01	1.07	1
		dd1			0	0.31	0.12					
		dd2			0	53.13	0.12					
		Sieve-SDP			inf	2.99					0.00	1
82	ex4.4_order13	None	0; 0; 560, 455	3653				0	1.45e−08, 2.31e−09	1.10e−06	238.42	
		pd1	0; 0; 1	1	1	9.81	0.19	1	0.00e+00, 1.00e+00	5.00e−01	1.18	1
		pd2	0; 0; 1	1	1	61.06	0.19	1	0.00e+00, 1.00e+00	5.00e−01	1.02	1
		dd1			0	0.49	0.19					
		dd2			0	100.91	0.19					
		Sieve-SDP			inf	5.35					0.00	1
83	ex4.4_order14	None	0; 0; 680, 560	4494				0	− 1.06e−08, − 3.67e−08	1.87e−06	455.22	
		pd1	0; 0; 1	1	1	16.86	0.30	1	0.00e+00, 1.00e+00	5.00e−01	0.46	1
		pd2	0; 0; 1	1	1	107.61	0.30	1	0.00e+00, 1.00e+00	5.00e−01	0.45	1
		dd1			0	0.68	0.30					
		dd2			0	169.13	0.30					
		Sieve-SDP			inf	9.68					0.00	1
84	ex4.4_order15	None	0; 0; 816, 680	5455				0	− 1.65e−08, − 4.13e−08	1.79e−06	923.64	
		pd1	0; 0; 1	1	1	28.95	0.45	1	0.00e+00, 1.00e+00	5.00e−01	0.50	1
		pd2	0; 0; 1	1	1	182.62	0.45	1	0.00e+00, 1.00e+00	5.00e−01	0.44	1
		dd1			0	1.13	0.45					
		dd2			0	284.93	0.45					
		Sieve-SDP			inf	17.79					0.00	1
85	ex4.4_order16	None	0; 0; 969, 816	6544				0	− 2.21e−08, − 4.65e−08	1.75e−06	1906.07	
		pd1	0; 0; 1	1	1	45.76	0.63	1	0.00e+00, 1.00e+00	5.00e−01	0.53	1
		pd2	0; 0; 1	1	1	301.76	0.63	1	0.00e+00, 1.00e+00	5.00e−01	0.46	1
		dd1			0	1.70	0.63					
		dd2			0	476.46	0.63					
		Sieve-SDP			inf	30.87					0.00	1

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
86	ex4.4_order17	None	0; 0; 1140, 969	7769				0	− 7.69e−09, − 1.50e−08	5.48e−07	3654.29	
		pd1	0; 0; 1	1	1	69.43	0.88	1	0.00e+00, 1.00e+00	5.00e−01	1.18	1
		pd2	0; 0; 1	1	1	474.62	0.89	1	0.00e+00, 1.00e+00	5.00e−01	1.22	1
		dd1			0	2.52	0.88					
		dd2			0	823.92	0.88					
		Sieve-SDP			inf	51.79					0.00	1
87	ex4.4_order18	None	0; 0; 1330, 1140	9138				0	− 9.27e−09, − 1.60e−08	4.84e−07	7063.56	
		pd1	0; 0; 1	1	1	103.53	1.24	1	0.00e+00, 1.00e+00	5.00e−01	1.55	1
		pd2	0; 0; 1	1	1	723.31	1.23	1	0.00e+00, 1.00e+00	5.00e−01	1.69	1
		dd1			0	3.71	1.23					
		dd2			0	1328.75	1.23					
		Sieve-SDP			inf	86.64					0.00	1
88	ex4.4_order19	None	0; 0; 1540, 1330	10659				0	− 2.12e−08, − 3.68e−08	1.22e−06	12500.09	
		pd1	0; 0; 1	1	1	150.44	1.70	1	0.00e+00, 1.00e+00	5.00e−01	1.47	1
		pd2	0; 0; 1	1	1	1128.87	1.70	1	0.00e+00, 1.00e+00	5.00e−01	1.64	1
		dd1			0	5.13	1.69					
		dd2			0	2112.63	1.69					
		Sieve-SDP			inf	140.48					0.00	1
89	ex4.4_order20	None	0; 0; 1771, 1540	12340				0	− 3.07e−08, − 5.42e−08	2.04e−06	25422.27	
		pd1	0; 0; 1	1	1	249.05	2.70	1	0.00e+00, 1.00e+00	5.00e−01	0.61	1
		pd2	0; 0; 1	1	1	1889.89	2.69	1	0.00e+00, 1.00e+00	5.00e−01	0.63	1
		dd1			0	7.15	2.69					
		dd2			0	3796.27	2.69					
		Sieve-SDP			inf	256.35					0.00	1
90	ex5.4_order5	None	0; 0; 21	65				0	2.28e+00, 2.28e+00	5.89e−07	0.90	
		pd1	0; 0; 10	31	1	0.07	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.86	
		pd2	0; 0; 10	31	1	0.09	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.78	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.02		0	2.28e+00, 2.28e+00	6.97e−08	0.77	

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
91	ex5.4_order6	None	0; 0; 28	90				0	2.28e+00, 2.28e+00	1.89e−06	0.95	
		pd1	0; 0; 10	31	1	0.09	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.78	2, 3
		pd2	0; 0; 10	31	1	0.11	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.77	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.03		0	2.28e+00, 2.28e+00	6.97e−08	0.84	2, 3
92	ex5.4_order7	None	0; 0; 36	119				0	2.28e+00, 2.28e+00	2.55e−06	0.98	
		pd1	0; 0; 10	31	1	0.10	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.78	2, 3
		pd2	0; 0; 10	31	1	0.13	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.81	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.04		0	2.28e+00, 2.28e+00	6.97e−08	0.82	2, 3
93	ex5.4_order8	None	0; 0; 45	152				0	2.28e+00, 2.28e+00	2.86e−06	0.96	
		pd1	0; 0; 10	31	1	0.11	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.77	2, 3
		pd2	0; 0; 10	31	1	0.17	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.78	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.04		0	2.28e+00, 2.28e+00	6.97e−08	0.85	2, 3
94	ex5.4_order9	None	0; 0; 55	189				0	2.28e+00, 2.28e+00	4.47e−06	0.95	
		pd1	0; 0; 10	31	1	0.13	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.78	2, 3
		pd2	0; 0; 10	31	1	0.23	0.00	0	2.28e+00, 2.28e+00	6.97e−08	0.85	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.05		0	2.28e+00, 2.28e+00	6.97e−08	0.77	2, 3
95	ex5.4_order10	None	0; 0; 66	230				0	2.28e+00, 2.28e+00	2.08e−06	1.98	
		pd1	0; 0; 10	31	1	0.21	0.01	0	2.28e+00, 2.28e+00	6.97e−08	1.59	2, 3
		pd2	0; 0; 10	31	1	0.33	0.00	0	2.28e+00, 2.28e+00	6.97e−08	1.62	2, 3
		dd1			0	0.05	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.10		0	2.28e+00, 2.28e+00	6.97e−08	1.62	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
96	ex5.4_order11	None	0; 0; 78	275				0	2.28e+00, 2.28e+00	9.31e−06	3.15	
		pd1	0; 0; 10	31	1	0.23	0.00	0	2.28e+00, 2.28e+00	6.97e−08	3.16	2, 3
		pd2	0; 0; 10	31	1	0.43	0.00	0	2.28e+00, 2.28e+00	6.97e−08	2.25	2, 3
		dd1			0	0.04	0.00					
		dd2			0	0.09	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.12		0	2.28e+00, 2.28e+00	6.97e−08	1.91	2, 3
97	ex5.4_order12	None	0; 0; 91	324				0	2.28e+00, 2.28e+00	1.09e−05	2.52	
		pd1	0; 0; 10	31	1	0.29	0.01	0	2.28e+00, 2.28e+00	6.97e−08	1.72	2, 3
		pd2	0; 0; 10	31	1	0.54	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.82	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.16	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.20		0	2.28e+00, 2.28e+00	6.97e−08	1.20	2, 3
98	ex5.4_order13	None	0; 0; 105	377				0	2.28e+00, 2.28e+00	4.58e−06	2.22	
		pd1	0; 0; 10	31	1	0.42	0.00	0	2.28e+00, 2.28e+00	6.97e−08	1.13	2, 3
		pd2	0; 0; 10	31	1	0.76	0.00	0	2.28e+00, 2.28e+00	6.97e−08	1.22	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.10	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.13		0	2.28e+00, 2.28e+00	6.97e−08	1.23	2, 3
99	ex5.4_order14	None	0; 0; 120	434				0	2.28e+00, 2.28e+00	1.14e−05	1.78	
		pd1	0; 0; 10	31	1	0.31	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.87	2, 3
		pd2	0; 0; 10	31	1	0.98	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.84	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.12	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.15		0	2.28e+00, 2.28e+00	6.97e−08	0.77	2, 3
100	ex5.4_order15	None	0; 0; 136	495				0	2.28e+00, 2.28e+00	6.12e−06	2.11	
		pd1	0; 0; 10	31	1	0.43	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.79	2, 3
		pd2	0; 0; 10	31	1	1.34	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.75	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.16	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.18		0	2.28e+00, 2.28e+00	6.97e−08	0.74	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
101	ex5.4_order16	None	0; 0; 153	560				0	2.28e+00, 2.28e+00	6.71e−06	3.38	
		pd1	0; 0; 10	31	1	0.53	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.83	2, 3
		pd2	0; 0; 10	31	1	1.84	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.85	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.20	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.23		0	2.28e+00, 2.28e+00	6.97e−08	0.80	2, 3
102	ex5.4_order17	None	0; 0; 171	629				0	2.28e+00, 2.28e+00	4.57e−06	4.32	
		pd1	0; 0; 10	31	1	0.62	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.77	2, 3
		pd2	0; 0; 10	31	1	2.50	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.95	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.26	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.27		0	2.28e+00, 2.28e+00	6.97e−08	1.28	2, 3
103	ex5.4_order18	None	0; 0; 190	702				0	2.27e+00, 2.27e+00	3.25e−03	7.00	
		pd1	0; 0; 10	31	1	0.93	0.01	0	2.28e+00, 2.28e+00	6.97e−08	1.14	2, 3
		pd2	0; 0; 10	31	1	4.02	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.81	2, 3
		dd1			0	0.08	0.01					
		dd2			0	0.33	0.01					
		Sieve-SDP	0; 0; 10	31	1	0.33		0	2.28e+00, 2.28e+00	6.97e−08	0.79	2, 3
104	ex5.4_order19	None	0; 0; 210	779				0	2.23e+00, 2.23e+00	2.09e−02	8.09	
		pd1	0; 0; 10	31	1	0.98	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.77	2, 3
		pd2	0; 0; 10	31	1	4.38	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.77	2, 3
		dd1			0	0.04	0.00					
		dd2			0	0.61	0.00					
		Sieve-SDP	0; 0; 10	31	1	0.67		0	2.28e+00, 2.28e+00	6.97e−08	1.66	2, 3
105	ex5.4_order20	None	0; 0; 231	860				0	2.14e+00, 2.15e+00	5.08e−02	10.64	
		pd1	0; 0; 10	31	1	1.19	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.75	2, 3
		pd2	0; 0; 10	31	1	6.78	0.01	0	2.28e+00, 2.28e+00	6.97e−08	0.99	2, 3
		dd1			0	0.08	0.01					
		dd2			0	0.65	0.01					
		Sieve-SDP	0; 0; 10	31	1	0.48		0	2.28e+00, 2.28e+00	6.97e−08	0.75	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
106	ex5.5_order4	None	0; 0; 15	44				0	1.62e−01, 1.62e−01	2.66e−06	0.88	
		pd1	0; 0; 7	20	1	0.05	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.81	2, 3
		pd2	0; 0; 7	20	1	0.07	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.02		0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
107	ex5.5_order5	None	0; 0; 21	65				0	1.62e−01, 1.62e−01	1.73e−06	0.83	
		pd1	0; 0; 7	20	1	0.07	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
		pd2	0; 0; 7	20	1	0.08	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.02		0	3.07e−01, 3.07e−01	1.08e−07	0.92	2, 3
108	ex5.5_order6	None	0; 0; 28	90				0	1.62e−01, 1.62e−01	2.59e−06	0.88	
		pd1	0; 0; 7	20	1	0.08	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
		pd2	0; 0; 7	20	1	0.10	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.02		0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
109	ex5.5_order7	None	0; 0; 36	119				0	1.62e−01, 1.62e−01	4.17e−06	0.90	
		pd1	0; 0; 7	20	1	0.09	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
		pd2	0; 0; 7	20	1	0.13	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.03		0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
110	ex5.5_order8	None	0; 0; 45	152				0	1.62e−01, 1.62e−01	1.97e−06	0.96	
		pd1	0; 0; 7	20	1	0.12	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
		pd2	0; 0; 7	20	1	0.18	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.77	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.04		0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
111	ex5.5_order9	None	0; 0; 55	189				0	1.62e−01, 1.62e−01	1.33e−06	0.92	
		pd1	0; 0; 7	20	1	0.14	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.77	2, 3
		pd2	0; 0; 7	20	1	0.22	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.05		0	3.07e−01, 3.07e−01	1.08e−07	0.74	2, 3
112	ex5.5_order10	None	0; 0; 66	230				0	1.62e−01, 1.62e−01	1.21e−05	0.94	
		pd1	0; 0; 7	20	1	0.17	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.86	2, 3
		pd2	0; 0; 7	20	1	0.29	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.06		0	3.07e−01, 3.07e−01	1.08e−07	0.74	2, 3
113	ex5.5_order11	None	0; 0; 78	275				0	1.62e−01, 1.62e−01	5.86e−06	1.06	
		pd1	0; 0; 7	20	1	0.19	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.92	2, 3
		pd2	0; 0; 7	20	1	0.39	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.81	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.06	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.08		0	3.07e−01, 3.07e−01	1.08e−07	1.13	2, 3
114	ex5.5_order12	None	0; 0; 91	324				0	1.62e−01, 1.62e−01	5.58e−06	1.49	
		pd1	0; 0; 7	20	1	0.35	0.00	0	3.07e−01, 3.07e−01	1.08e−07	1.03	2, 3
		pd2	0; 0; 7	20	1	0.55	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.82	2, 3
		dd1			0	0.02	0.00					
		dd2			0	0.08	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.09		0	3.07e−01, 3.07e−01	1.08e−07	0.85	2, 3
115	ex5.5_order13	None	0; 0; 105	377				0	1.62e−01, 1.62e−01	1.01e−05	1.24	
		pd1	0; 0; 7	20	1	0.29	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.88	2, 3
		pd2	0; 0; 7	20	1	0.76	0.00	0	3.07e−01, 3.07e−01	1.08e−07	1.26	2, 3
		dd1			0	0.04	0.00					
		dd2			0	0.17	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.15		0	3.07e−01, 3.07e−01	1.08e−07	0.77	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
116	ex5.5_order14	None	0; 0; 120	434				0	1.62e−01, 1.62e−01	2.12e−05	1.54	
		pd1	0; 0; 7	20	1	0.34	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.83	2, 3
		pd2	0; 0; 7	20	1	1.15	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.12	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.15		0	3.07e−01, 3.07e−01	1.08e−07	0.96	2, 3
117	ex5.5_order15	None	0; 0; 136	495				0	1.62e−01, 1.62e−01	4.22e−05	1.85	
		pd1	0; 0; 7	20	1	0.44	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.77	2, 3
		pd2	0; 0; 7	20	1	1.39	0.00	0	3.07e−01, 3.07e−01	1.08e−07	0.79	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.17	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.19		0	3.07e−01, 3.07e−01	1.08e−07	0.88	2, 3
118	ex5.5_order16	None	0; 0; 153	560				0	1.62e−01, 1.62e−01	4.38e−05	2.43	
		pd1	0; 0; 7	20	1	0.52	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
		pd2	0; 0; 7	20	1	1.93	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
		dd1			0	0.03	0.01					
		dd2			0	0.22	0.01					
		Sieve-SDP	0; 0; 7	20	1	0.22		0	3.07e−01, 3.07e−01	1.08e−07	0.77	2, 3
119	ex5.5_order17	None	0; 0; 171	629				0	1.62e−01, 1.62e−01	6.83e−06	3.24	
		pd1	0; 0; 7	20	1	0.63	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.74	2, 3
		pd2	0; 0; 7	20	1	2.52	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.75	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.26	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.28		0	3.07e−01, 3.07e−01	1.08e−07	0.79	2, 3
120	ex5.5_order18	None	0; 0; 190	702				0	1.62e−01, 1.62e−01	4.05e−06	4.51	
		pd1	0; 0; 7	20	1	0.92	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.89	2, 3
		pd2	0; 0; 7	20	1	3.45	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.89	2, 3
		dd1			0	0.04	0.00					
		dd2			0	0.34	0.00					
		Sieve-SDP	0; 0; 7	20	1	0.33		0	3.07e−01, 3.07e−01	1.08e−07	0.80	2, 3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
121	ex5.5_order19	None	0; 0; 210	779				0	1.62e−01, 1.62e−01	4.64e−05	7.12	
		pd1	0; 0; 7	20	1	1.41	0.01	0	3.07e−01, 3.07e−01	1.08e−07	1.23	2, 3
		pd2	0; 0; 7	20	1	5.53	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.82	2, 3
		dd1			0	0.04	0.01					
		dd2			0	0.43	0.01					
		Sieve-SDP	0; 0; 7	20	1	0.39		0	3.07e−01, 3.07e−01	1.08e−07	0.80	2, 3
122	ex5.5_order20	None	0; 0; 231	860				0	1.62e−01, 1.62e−01	6.34e−05	8.90	
		pd1	0; 0; 7	20	1	1.22	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
		pd2	0; 0; 7	20	1	5.88	0.01	0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
		dd1			0	0.05	0.01					
		dd2			0	0.52	0.01					
		Sieve-SDP	0; 0; 7	20	1	0.46		0	3.07e−01, 3.07e−01	1.08e−07	0.76	2, 3
123	ex5.6_order4	None	0; 0; 15	44				0	3.04e−01, 3.04e−01	1.15e−07	1.01	
		pd1	0; 0; 8	21	1	0.07	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.94	
		pd2	0; 0; 8	21	1	0.08	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.84	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.02		0	3.04e−01, 3.04e−01	1.01e−08	1.02	
124	ex5.6_order5	None	0; 0; 21	65				0	3.04e−01, 3.04e−01	1.74e−07	1.10	
		pd1	0; 0; 8	21	1	0.07	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.84	
		pd2	0; 0; 8	21	1	0.07	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.90	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.02		0	3.04e−01, 3.04e−01	1.01e−08	0.92	
125	ex5.6_order6	None	0; 0; 28	90				0	3.04e−01, 3.04e−01	3.45e−07	0.89	
		pd1	0; 0; 8	21	1	0.07	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.78	
		pd2	0; 0; 8	21	1	0.09	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.86	
		dd1			0	0.03	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.05		0	3.04e−01, 3.04e−01	1.01e−08	2.02	

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
126	ex5.6_order7	None	0; 0; 36	119				0	3.04e−01, 3.04e−01	3.21e−07	1.17	
		pd1	0; 0; 8	21	1	0.12	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.96	
		pd2	0; 0; 8	21	1	0.17	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.85	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.03		0	3.04e−01, 3.04e−01	1.01e−08	1.39	
127	ex5.6_order8	None	0; 0; 45	152				0	3.04e−01, 3.04e−01	4.51e−07	0.91	
		pd1	0; 0; 8	21	1	0.12	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.79	
		pd2	0; 0; 8	21	1	0.16	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.94	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.04		0	3.04e−01, 3.04e−01	1.01e−08	0.74	
128	ex5.6_order9	None	0; 0; 55	189				0	3.04e−01, 3.04e−01	7.42e−07	0.93	
		pd1	0; 0; 8	21	1	0.14	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.79	
		pd2	0; 0; 8	21	1	0.24	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.82	
		dd1			0	0.03	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.06		0	3.04e−01, 3.04e−01	1.01e−08	0.82	
129	ex5.6_order10	None	0; 0; 66	230				0	3.04e−01, 3.04e−01	7.57e−07	0.87	
		pd1	0; 0; 8	21	1	0.16	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.77	
		pd2	0; 0; 8	21	1	0.28	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.74	
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.06		0	3.04e−01, 3.04e−01	1.01e−08	0.75	
130	ex5.6_order11	None	0; 0; 78	275				0	3.04e−01, 3.04e−01	6.11e−07	0.90	
		pd1	0; 0; 8	21	1	0.18	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.74	
		pd2	0; 0; 8	21	1	0.40	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.76	
		dd1			0	0.02	0.00					
		dd2			0	0.06	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.08		0	3.04e−01, 3.04e−01	1.01e−08	0.76	

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
131	ex5.6_order12	None	0; 0; 91	324				0	3.04e−01, 3.04e−01	9.01e−07	1.02	
		pd1	0; 0; 8	21	1	0.23	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.75	
		pd2	0; 0; 8	21	1	0.54	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.75	
		dd1			0	0.02	0.00					
		dd2			0	0.08	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.10		0	3.04e−01, 3.04e−01	1.01e−08	0.74	
132	ex5.6_order13	None	0; 0; 105	377				0	3.04e−01, 3.04e−01	7.94e−07	1.20	
		pd1	0; 0; 8	21	1	0.28	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.77	
		pd2	0; 0; 8	21	1	0.71	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.84	
		dd1			0	0.03	0.00					
		dd2			0	0.11	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.13		0	3.04e−01, 3.04e−01	1.01e−08	0.79	
133	ex5.6_order14	None	0; 0; 120	434				0	3.04e−01, 3.04e−01	1.19e−06	1.31	
		pd1	0; 0; 8	21	1	0.34	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.82	2
		pd2	0; 0; 8	21	1	1.01	0.00	0	3.04e−01, 3.04e−01	1.01e−08	0.74	2
		dd1			0	0.02	0.00					
		dd2			0	0.13	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.15		0	3.04e−01, 3.04e−01	1.01e−08	0.79	2
134	ex5.6_order15	None	0; 0; 136	495				0	3.04e−01, 3.04e−01	8.76e−07	1.62	
		pd1	0; 0; 8	21	1	0.43	0.00	0	3.04e−01, 3.04e−01	1.01e−08	1.27	
		pd2	0; 0; 8	21	1	1.45	0.01	0	3.04e−01, 3.04e−01	1.01e−08	1.21	
		dd1			0	0.04	0.00					
		dd2			0	0.16	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.19		0	3.04e−01, 3.04e−01	1.01e−08	0.76	
135	ex5.6_order16	None	0; 0; 153	560				0	3.04e−01, 3.04e−01	1.79e−06	1.81	
		pd1	0; 0; 8	21	1	0.52	0.01	0	3.04e−01, 3.04e−01	1.01e−08	0.75	2
		pd2	0; 0; 8	21	1	1.91	0.01	0	3.04e−01, 3.04e−01	1.01e−08	0.74	2
		dd1			0	0.03	0.00					
		dd2			0	0.21	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.22		0	3.04e−01, 3.04e−01	1.01e−08	0.76	2

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
136	ex5.6_order17	None	0; 0; 171	629				0	3.04e−01, 3.04e−01	1.78e−06	2.29	
		pd1	0; 0; 8	21	1	0.60	0.01	0	3.04e−01, 3.04e−01	1.01e−08	0.83	2
		pd2	0; 0; 8	21	1	2.47	0.01	0	3.04e−01, 3.04e−01	1.01e−08	0.84	2
		dd1			0	0.03	0.00					
		dd2			0	0.26	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.28		0	3.04e−01, 3.04e−01	1.01e−08	0.74	2
137	ex5.6_order18	None	0; 0; 190	702				0	3.04e−01, 3.04e−01	2.53e−06	2.68	
		pd1	0; 0; 8	21	1	0.75	0.01	0	3.04e−01, 3.04e−01	1.01e−08	0.76	2
		pd2	0; 0; 8	21	1	3.24	0.01	0	3.04e−01, 3.04e−01	1.01e−08	0.83	2
		dd1			0	0.04	0.00					
		dd2			0	0.31	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.33		0	3.04e−01, 3.04e−01	1.01e−08	0.74	2
138	ex5.6_order19	None	0; 0; 210	779				0	3.04e−01, 3.04e−01	2.54e−06	3.34	
		pd1	0; 0; 8	21	1	1.06	0.01	0	3.04e−01, 3.04e−01	1.01e−08	1.52	2
		pd2	0; 0; 8	21	1	4.87	0.01	0	3.04e−01, 3.04e−01	1.01e−08	0.89	2
		dd1			0	0.05	0.00					
		dd2			0	0.45	0.00					
		Sieve-SDP	0; 0; 8	21	1	0.50		0	3.04e−01, 3.04e−01	1.01e−08	0.92	2
139	ex5.6_order20	None	0; 0; 231	860				0	3.04e−01, 3.04e−01	3.35e−06	5.55	
		pd1	0; 0; 8	21	1	1.48	0.01	0	3.04e−01, 3.04e−01	1.01e−08	1.02	2
		pd2	0; 0; 8	21	1	6.57	0.01	0	3.04e−01, 3.04e−01	1.01e−08	0.91	2
		dd1			0	0.07	0.01					
		dd2			0	0.58	0.01					
		Sieve-SDP	0; 0; 8	21	1	0.53		0	3.04e−01, 3.04e−01	1.01e−08	1.02	2
140	ex5.7_order5	None	0; 1; 21	65				0	8.52e−09, − 5.99e−09	7.10e−08	0.93	
		pd1	0; 1; 8	22	1	0.08	0.00	0	1.81e−08, − 2.87e−09	5.26e−08	1.20	
		pd2	0; 1; 8	22	1	0.14	0.00	0	1.81e−08, − 2.87e−09	5.26e−08	1.12	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP	0; 1; 8	22	1	0.03		0	1.81e−08, − 2.87e−09	5.26e−08	1.17	

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
141	ex5.7_order6	None	0; 0; 28, 3	90				0	2.28e−08, − 2.32e−09	7.80e−08	0.97	
		pd1	0; 0; 10, 2	28	1	0.09	0.00	0	7.23e−09, − 4.48e−09	2.43e−08	1.07	
		pd2	0; 0; 10, 2	28	1	0.18	0.00	0	7.23e−09, − 4.48e−09	2.43e−08	0.93	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 10, 2	28	1	0.03		0	7.23e−09, − 4.48e−09	2.43e−08	1.02	
142	ex5.7_order7	None	0; 0; 36, 6	119				0	2.01e−08, − 9.32e−09	1.19e−07	0.89	
		pd1	0; 0; 11, 3	33	1	0.14	0.00	0	7.70e−09, − 4.95e−09	2.93e−08	0.89	
		pd2	0; 0; 11, 3	33	1	0.18	0.00	0	7.70e−09, − 4.95e−09	2.93e−08	0.91	
		dd1			0	0.02	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 11, 3	33	1	0.04		0	7.70e−09, − 4.95e−09	2.93e−08	0.95	
143	ex5.7_order8	None	0; 0; 45, 10	152				0	1.43e−08, − 1.11e−08	1.17e−07	0.93	
		pd1	0; 0; 12, 4	36	1	0.16	0.00	0	2.24e−08, − 2.72e−09	7.97e−08	0.91	
		pd2	0; 0; 12, 4	36	1	0.26	0.00	0	2.24e−08, − 2.72e−09	7.97e−08	1.02	
		dd1			0	0.02	0.00					
		dd2			0	0.04	0.00					
		Sieve-SDP	0; 0; 12, 4	36	1	0.07		0	2.24e−08, − 2.72e−09	7.97e−08	1.00	
144	ex5.7_order9	None	0; 0; 55, 15	189				0	6.00e−09, − 4.83e−09	5.61e−08	1.24	
		pd1	0; 0; 13, 5	41	1	0.24	0.01	0	2.15e−08, − 3.69e−09	8.53e−08	1.19	
		pd2	0; 0; 13, 5	41	1	0.39	0.01	0	2.15e−08, − 3.69e−09	8.53e−08	0.89	
		dd1			0	0.02	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP	0; 0; 13, 5	41	1	0.10		0	2.15e−08, − 3.69e−09	8.53e−08	0.88	
145	ex5.7_order10	None	0; 0; 66, 21	230				0	1.30e−08, − 1.43e−08	1.69e−07	0.91	
		pd1	0; 0; 13, 5	41	1	0.31	0.01	0	2.15e−08, − 3.69e−09	8.53e−08	0.95	
		pd2	0; 0; 13, 5	41	1	0.53	0.01	0	2.15e−08, − 3.69e−09	8.53e−08	1.58	
		dd1			0	0.03	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP	0; 0; 13, 5	41	1	0.15		0	2.15e−08, − 3.69e−09	8.53e−08	0.94	

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
146	ex5.7_order11	None	0; 0; 78, 28	275				0	1.71e−08, − 1.50e−08	2.11e−07	1.11	
		pd1	0; 0; 14, 6	45	1	0.36	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.91	
		pd2	0; 0; 14, 6	45	1	0.75	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.93	
		dd1			0	0.02	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP	0; 0; 14, 6	45	1	0.17		0	1.73e−08, − 3.62e−09	8.09e−08	0.95	
147	ex5.7_order12	None	0; 0; 91, 36	324				0	5.85e−09, − 6.21e−09	7.87e−08	1.04	
		pd1	0; 0; 14, 6	45	1	0.44	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.91	
		pd2	0; 0; 14, 6	45	1	1.04	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	1.03	
		dd1			0	0.03	0.00					
		dd2			0	0.12	0.00					
		Sieve-SDP	0; 0; 14, 6	45	1	0.24		0	1.73e−08, − 3.62e−09	8.09e−08	1.50	
148	ex5.7_order13	None	0; 0; 105, 45	377				0	2.13e−08, − 2.19e−08	3.23e−07	1.15	
		pd1	0; 0; 14, 6	45	1	0.49	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.90	
		pd2	0; 0; 14, 6	45	1	1.56	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.95	
		dd1			0	0.03	0.00					
		dd2			0	0.14	0.00					
		Sieve-SDP	0; 0; 14, 6	45	1	0.31		0	1.73e−08, − 3.62e−09	8.09e−08	1.05	
149	ex5.7_order14	None	0; 0; 120, 55	434				0	2.24e−08, − 2.28e−08	3.72e−07	1.69	
		pd1	0; 0; 14, 6	45	1	0.77	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	1.07	
		pd2	0; 0; 14, 6	45	1	2.39	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.94	
		dd1			0	0.03	0.01					
		dd2			0	0.22	0.01					
		Sieve-SDP	0; 0; 14, 6	45	1	0.37		0	1.73e−08, − 3.62e−09	8.09e−08	0.88	
150	ex5.7_order15	None	0; 0; 136, 66	495				0	2.09e−08, − 2.22e−08	3.87e−07	1.56	
		pd1	0; 0; 14, 6	45	1	0.82	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.84	
		pd2	0; 0; 14, 6	45	1	2.80	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.86	
		dd1			0	0.03	0.01					
		dd2			0	0.26	0.01					
		Sieve-SDP	0; 0; 14, 6	45	1	0.45		0	1.73e−08, − 3.62e−09	8.09e−08	1.38	

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
151	ex5.7_order16	None	0; 0; 153, 78	560				0	1.87e−08, − 1.91e−08	3.70e−07	2.13	
		pd1	0; 0; 14, 6	45	1	1.13	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	0.97	
		pd2	0; 0; 14, 6	45	1	3.91	0.01	0	1.73e−08, − 3.62e−09	8.09e−08	1.41	
		dd1			0	0.04	0.01					
		dd2			0	0.34	0.01					
		Sieve-SDP	0; 0; 14, 6	45	1	0.60		0	1.73e−08, − 3.62e−09	8.09e−08	1.43	
152	ex5.7_order17	None	0; 0; 171, 91	629				0	1.84e−08, − 1.83e−08	3.89e−07	2.56	
		pd1	0; 0; 14, 6	45	1	1.47	0.02	0	1.73e−08, − 3.62e−09	8.09e−08	0.92	
		pd2	0; 0; 14, 6	45	1	5.58	0.02	0	1.73e−08, − 3.62e−09	8.09e−08	1.13	
		dd1			0	0.05	0.01					
		dd2			0	0.46	0.01					
		Sieve-SDP	0; 0; 14, 6	45	1	0.95		0	1.73e−08, − 3.62e−09	8.09e−08	1.47	
153	ex5.7_order18	None	0; 0; 190, 105	702				0	1.76e−08, − 1.75e−08	4.12e−07	3.01	
		pd1	0; 0; 14, 6	45	1	1.76	0.02	0	1.73e−08, − 3.62e−09	8.09e−08	0.84	
		pd2	0; 0; 14, 6	45	1	7.46	0.02	0	1.73e−08, − 3.62e−09	8.09e−08	1.15	
		dd1			0	0.07	0.01					
		dd2			0	0.58	0.01					
		Sieve-SDP	0; 0; 14, 6	45	1	0.94		0	1.73e−08, − 3.62e−09	8.09e−08	1.36	
154	ex5.7_order19	None	0; 0; 210, 120	779				0	1.68e−08, − 1.67e−08	4.25e−07	4.46	
		pd1	0; 0; 14, 6	45	1	2.49	0.02	0	1.73e−08, − 3.62e−09	8.09e−08	1.09	
		pd2	0; 0; 14, 6	45	1	10.47	0.02	0	1.73e−08, − 3.62e−09	8.09e−08	0.89	
		dd1			0	0.05	0.02					
		dd2			0	0.84	0.02					
		Sieve-SDP	0; 0; 14, 6	45	1	1.11		0	1.73e−08, − 3.62e−09	8.09e−08	1.01	
155	ex5.7_order20	None	0; 0; 231, 136	860				0	1.54e−08, − 1.69e−08	4.40e−07	5.84	
		pd1	0; 0; 14, 6	45	1	3.28	0.03	0	1.73e−08, − 3.62e−09	8.09e−08	1.02	
		pd2	0; 0; 14, 6	45	1	13.67	0.03	0	1.73e−08, − 3.62e−09	8.09e−08	1.17	
		dd1			0	0.08	0.02					
		dd2			0	1.04	0.02					
		Sieve-SDP	0; 0; 14, 6	45	1	1.71		0	1.73e−08, − 3.62e−09	8.09e−08	0.86	

A.4 Detailed results on the Henrion-Toh dataset

This dataset has 98 problems. From these 18 problems were reduced by at least one of the five methods.

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
1	sedumi-brown	None	925; 0; 56	461				0	− 7.34e−09, 0.00e+00	3.75e−07	0.93	
		pd1	925; 0; 21	251	1	0.15	0.05	0	− 9.33e−11, 0.00e+00	6.25e−09	0.80	
		pd2	925; 0; 21	251	1	0.23	0.02	0	− 9.33e−11, 0.00e+00	6.25e−09	0.78	
		dd1			0	0.03	0.00					
		dd2			0	0.06	0.00					
		Sieve-SDP	925; 0; 21	251	1	0.04		0	− 9.33e−11, 0.00e+00	6.25e−09	0.79	
2	sedumi-conform3	None	630; 0; 56	285				0	2.05e−08, 0.00e+00	4.54e−07	0.66	
		pd1	630; 0; 53	273	1	0.04	0.02	0	2.51e−08, 0.00e+00	4.90e−07	0.71	
		pd2	630; 0; 53	273	1	0.11	0.03	0	2.51e−08, 0.00e+00	4.90e−07	0.73	
		dd1			0	0.03	0.00					
		dd2			0	0.06	0.00					
		Sieve-SDP	630; 0; 53	273	1	0.02		0	2.51e−08, 0.00e+00	4.90e−07	0.68	
3	sedumi-conform4	None	1890; 0; 84	454				0	− 2.51e−08, 0.00e+00	5.57e−06	0.84	
		pd1	1890; 0; 81	442	1	0.07	0.04	0	− 6.47e−09, 0.00e+00	1.74e−06	0.81	
		pd2	1890; 0; 81	442	1	0.26	0.04	0	− 6.47e−09, 0.00e+00	1.74e−06	0.88	
		dd1			0	0.02	0.00					
		dd2			0	0.08	0.00					
		Sieve-SDP	1890; 0; 81	442	1	0.02		0	− 6.47e−09, 0.00e+00	1.74e−06	0.77	
4	sedumi-fp23	None	0; 0; 28, 7_{13}	209				0	2.13e+02, 2.13e+02	3.97e−06	1.50	
		pd1	0; 0; 7_{14}	83	1	0.09	0.02	0	2.13e+02, 2.13e+02	9.96e−07	1.60	3
		pd2	0; 0; 7_{14}	83	1	0.13	0.02	0	2.13e+02, 2.13e+02	9.96e−07	1.44	3
		dd1			0	0.05	0.00					
		dd2			0	0.06	0.00					
		Sieve-SDP	0; 0; 7_{14}	83	1	0.04		0	2.13e+02, 2.13e+02	9.96e−07	1.33	3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
5	sedumi-fp24	None	$0; 0; 105, 14_{35}$	2379				0	1.95e+02, 1.95e+02	9.68e−08	6.36	
		pd1	$0; 0; 14_{36}$	559	1	0.32	0.21	0	1.95e+02, 1.95e+02	1.74e−10	1.77	
		pd2	$0; 0; 14_{36}$	559	1	0.82	0.20	0	1.95e+02, 1.95e+02	1.74e−10	1.90	
		dd1			0	0.06	0.00					
		dd2			0	0.25	0.00					
		Sieve-SDP	$0; 0; 14_{36}$	559	1	0.23		0	1.95e+02, 1.95e+02	1.74e−10	1.78	
6	sedumi-fp25	None	$0; 0; 28, 7_{15}$	209				0	1.10e+01, 1.10e+01	6.63e−06	1.39	
		pd1	$0; 0; 7_{16}$	83	1	0.12	0.03	0	1.10e+01, 1.10e+01	1.39e−07	1.46	2, 3
		pd2	$0; 0; 7_{16}$	83	1	0.17	0.05	0	1.10e+01, 1.10e+01	1.39e−07	1.28	2, 3
		dd1			0	0.05	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP	$0; 0; 7_{16}$	83	1	0.03		0	1.10e+01, 1.10e+01	1.39e−07	1.30	2, 3
7	sedumi-fp26	None	$0; 0; 66, 11_{31}$	1000				0	2.68e+02, 2.68e+02	3.74e−08	2.11	
		pd1	$0; 0; 11_{32}$	285	1	0.17	0.16	0	2.68e+02, 2.68e+02	1.18e−07	1.46	
		pd2	$0; 0; 11_{32}$	285	1	0.53	0.15	0	2.68e+02, 2.68e+02	1.18e−07	1.48	
		dd1			0	0.08	0.00					
		dd2			0	0.46	0.00					
		Sieve-SDP	$0; 0; 11_{32}$	285	1	0.11		0	2.68e+02, 2.68e+02	1.18e−07	1.53	
8	sedumi-fp27	None	$0; 0; 66, 11_{25}$	1000				0	3.90e+01, 3.90e+01	1.96e−10	2.50	
		pd1	$0; 0; 11_{26}$	285	1	0.16	0.10	0	3.90e+01, 3.90e+01	3.98e−09	1.50	
		pd2	$0; 0; 11_{26}$	285	1	0.39	0.11	0	3.90e+01, 3.90e+01	3.98e−09	1.45	
		dd1			0	0.05	0.01					
		dd2			0	0.30	0.01					
		Sieve-SDP	$0; 0; 11_{26}$	285	1	0.11		0	3.90e+01, 3.90e+01	3.98e−09	1.46	
9	sedumi-fp32	None	$0; 0; 165, 45_{22}$	3002				0	− 7.05e+00, − 7.05e+00	2.79e−07	47.43	
		pd1	$0; 0; 45_4, 9_3, 45_{16}$	1286	1	2.17	0.56	0	− 7.05e+00, − 7.05e+00	2.50e−06	8.58	3
		pd2	$0; 0; 45_4, 9_3, 45_{16}$	1286	1	4.55	0.57	0	− 7.05e+00, − 7.05e+00	2.50e−06	9.26	3
		dd1			0	0.11	0.02					
		dd2			0	9.78	0.02					
		Sieve-SDP	$0; 0; 45_4, 9_3, 45_{16}$	1286	1	1.21		0	− 7.05e+00, − 7.05e+00	2.50e−06	12.14	3

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
10	sedumi-fp33	None	0; 0; 21, 6_{16}	125				0	− 1.01e+04, − 1.01e+04	3.36e−07	0.75	
		pd1	0; 0; 13, 6_{16}	105	1	0.10	0.04	0	− 1.01e+04, − 1.01e+04	3.01e−07	0.94	
		pd2	0; 0; 13, 6_{16}	105	1	0.20	0.03	0	− 1.01e+04, − 1.01e+04	3.01e−07	1.32	
		dd1			0	0.04	0.00					
		dd2			0	0.12	0.00					
		Sieve-SDP	0; 0; 14, 6_{16}	111	1	0.03		0	− 1.18e+04, − 1.18e+04	9.28e−02	1.05	− 2
11	sedumi-fp34	None	0; 0; 28, 7_{16}	209				0	1.72e+02, 1.72e+02	8.10e−07	0.94	
		pd1	0; 0; 7, 1_2, 7_{14}	83	1	0.14	0.04	0	1.72e+02, 1.72e+02	3.11e−07	0.88	
		pd2	0; 0; 7, 1_2, 7_{14}	83	1	0.11	0.03	0	1.72e+02, 1.72e+02	3.11e−07	0.73	
		dd1			0	0.03	0.00					
		dd2			0	0.07	0.00					
		Sieve-SDP	0; 0; 7, 1_2, 7_{14}	83	1	0.02		0	1.72e+02, 1.72e+02	3.11e−07	0.78	
12	sedumi-fp35	None	0; 0; 35, 20_8	164				0	4.00e+00, 4.00e+00	5.76e−06	0.86	
		pd1	0; 0; 20_8, 10	119	1	0.16	0.04	0	4.00e+00, 4.00e+00	5.66e−07	0.80	2, 3
		pd2	0; 0; 20_8, 10	119	1	0.36	0.05	0	4.00e+00, 4.00e+00	5.66e−07	0.85	2, 3
		dd1			0	0.03	0.00					
		dd2			0	0.28	0.00					
		Sieve-SDP	0; 0; 20_8, 10	119	1	0.05		0	4.00e+00, 4.00e+00	5.66e−07	0.89	2, 3
13	sedumi-fp44	None	0; 0; 4, 3_2	6				0	4.44e+02, 4.44e+02	4.67e−08	1.02	
		pd1	0; 0; 3_3	5	1	0.04	0.00	0	4.44e+02, 4.44e+02	1.55e−08	0.87	
		pd2	0; 0; 3_3	5	1	0.04	0.00	0	4.44e+02, 4.44e+02	1.55e−08	0.82	
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 3_3	5	1	0.01		0	4.44e+02, 4.44e+02	1.55e−08	0.77	
14	sedumi-fp46	None	0; 0; 10, 6_2	27				0	6.70e−08, − 2.54e−07	4.78e−07	0.89	
		pd1	0; 0; 5, 3, 2	11	1	0.13	0.00	0	1.54e−07, − 2.20e−08	2.22e−07	0.76	
		pd2	0; 0; 5, 3, 2	11	1	0.14	0.00	0	1.54e−07, − 2.20e−08	2.22e−07	0.69	
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	0; 0; 5, 3, 2	11	1	0.01		0	1.54e−07, − 2.20e−08	2.22e−07	0.76	

No.	Name	Method	f; l; s	m	red.	t_{prep}	t_{conv}	inf	obj (P, D)	DIMACS	t_{sol}	Help
15	sedumi-fp49	None	1; 0; 6, 3_4	14				0	1.67e+01, 1.67e+01	5.98e−08	1.04	
		pd1	1; 0; 4, 3_4	10	1	0.05	0.00	0	1.67e+01, 1.67e+01	1.40e−08	0.62	
		pd2	1; 0; 4, 3_4	10	1	0.06	0.00	0	1.67e+01, 1.67e+01	1.40e−08	0.58	
		dd1			0	0.02	0.00					
		dd2			0	0.02	0.00					
		Sieve-SDP	1; 0; 4, 3_4	10	1	0.01		0	1.67e+01, 1.67e+01	1.40e−08	0.60	
16	sedumi-fp210	None	66; 0; 66, 11_{10}	1000				0	3.75e−01, 3.75e−01	2.15e−07	1.59	
		pd1	66; 0; 11_{11}	285	1	0.12	0.05	0	3.75e−01, 3.75e−01	2.55e−08	1.35	
		pd2	66; 0; 11_{11}	285	1	0.17	0.03	0	3.75e−01, 3.75e−01	2.55e−08	1.28	
		dd1			0	0.05	0.00					
		dd2			0	0.09	0.00					
		Sieve-SDP	66; 0; 11_{11}	285	1	0.05		0	3.75e−01, 3.75e−01	2.55e−08	1.31	
17	sedumi-fp410	None	1; 0; 6, 3_4	14				0	1.67e+01, 1.67e+01	5.98e−08	0.87	
		pd1	1; 0; 4, 3_4	10	1	0.08	0.01	0	1.67e+01, 1.67e+01	1.40e−08	0.81	
		pd2	1; 0; 4, 3_4	10	1	0.08	0.00	0	1.67e+01, 1.67e+01	1.40e−08	0.90	
		dd1			0	0.03	0.00					
		dd2			0	0.03	0.00					
		Sieve-SDP	1; 0; 4, 3_4	10	1	0.02		0	1.67e+01, 1.67e+01	1.40e−08	0.81	
18	sedumi-l4	None	0; 0; 45	152				0	3.70e−02, 3.70e−02	7.10e−08	0.83	
		pd1	0; 0; 1	1	1	0.15	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.44	1
		pd2	0; 0; 1	1	1	0.19	0.00	1	0.00e+00, 1.00e+00	5.00e−01	0.42	1
		dd1			0	0.02	0.00					
		dd2			0	0.05	0.00					
		Sieve-SDP			inf	0.04					0.00	1

B Core Matlab code

In this section we provide our core Matlab code of Sieve-SDP (not including input, output, and dual solution recovery) with comments. In our code we physically delete rows and columns of the A_i and of C only at the very end. During the execution of the algorithm we only *mark* such rows, columns and constraints as deleted.

We use two arrays to keep track of what has been marked deleted:

1. The m-vector `undeleted`, whose ith entry is 1 if constraint i *has not* been deleted, and 0 if it *has* been deleted.
2. The sparse array `I` $\in \{0, 1\}^{n\times(m+1)}$ with entries defined as follows.
   (a) For all i and for $1 \le j \le m$

$$I(i, j) = \begin{cases} 0, & \text{if in } A_j \text{ the } i \text{ th row and column are all zero or have been deleted;} \\ 1, & \text{otherwise.} \end{cases}$$

   (b) For all i

$$I(i, m+1) = \begin{cases} 0, & \text{if in all } A_j \text{ the } i \text{ th row and column have been deleted;} \\ 1, & \text{otherwise.} \end{cases}$$

```
function[Ar, br, Cr, info] = SieveSDP(A, b, C, EPS)

    % Inputs:
    %  A: n-by-n*m sparse matrix,
    %     which is m symmetric n-by-n matrices side by side
    %  b: the vector of rhs in R m, and b <= 0;
    %  C: the objective coefficient n-by-n matrix;
    %  EPS: accuracy for safe mode, with default value eps
    % Outputs:
    %   Ar, br, cr: Reduced data after preprocessing
    %   info: A structure containing preprocessing info

    if nargin < 4, EPS = eps; end
    sqrtEPS = sqrt(EPS);

    Ar = []; br = []; Cr = [];
    n = size(C, 1); m = length(b);
    I = true(n, m + 1); % initial nonzero indices
    for i = 1:m,
        I(:, i) = any(A(:, (n*(i - 1) + 1):(n*i)), 2);
    end

    not_done     = 1;            % 1 means preprocessing not done
    undeleted    = ones(m, 1); % keep track of deleted constraints
    constr_ind   = (1:m);       % indices or undeleted constraints
    mr           = m;           % reduced number of constraints
    info.infeas = 0;            % infeasibility detected?
    info.red     = 0;            % any reduction?

    bn = -sqrtEPS*max(1, norm(b, inf));
```

```
                    % b < 0 if b < -sqrt(epsilon)*max{1, ||b||}
bz = bn*sqrtEPS; % b = 0 if -epsilon*max{1, ||b||} < b <= 0

% Preprocessing
while not_done
    not_done = 0;
    for ii = 1:mr
        i  = constr_ind(ii);
        Ii = I(:, i); % indicates undeleted vars in matrix i
        Ai = A(Ii, n*(i - 1) + find(Ii)); % nonzero submatrix
        Iaux = any(Ai, 2);
        if find(Iaux == false, 1),
            I(Ii, i) = Iaux; Ii = I(:, i); Ai = Ai(Iaux, Iaux);
        end
        if isempty(Ai)
            if b(i) < bn, info.infeas = 1; return; end
                % Ai=0 and bi<0 => infeasible
            if b(i) > bz, undeleted(i) = 0; continue; end
                % Ai=0 and bi=0 => reduce
        end
        if b(i) < bn
            [~, pd_check] = chol(Ai);
            if pd_check == 0, info.infeas = 1; return; end
                % Ai pd and bi<0 => infeasible
        else
            if b(i) > bz
                [~, pd_check] = chol(Ai);
                if pd_check == 0
                    % Ai pd and bi=0 => reduce
                    I(Ii, :) = false; undeleted(i) = 0;
                    not_done = 1;
                else
                    [~, nd_check] = chol(-Ai);
                    if nd_check == 0
                        % Ai nd and bi=0 => reduce
                        I(Ii, :) = false; undeleted(i) = 0;
                        not_done = 1;
                    end
                end
            end
        end
    end
    constr_ind = find(undeleted); mr = length(constr_ind);
end

% Undeleted rows/columns are marked in I(:, m + 1)
% Now do physical deletion
if mr == m
    Ar = A; br = b; Cr = C; info.red = 0; return;
end
info.red = 1;
I_nonzero = I(:, m + 1); nr = nnz(I_nonzero);
Ar         = sparse(nr, nr*mr);
for ii = 1:mr
    i = constr_ind(ii);
    Ar(:, (nr*(ii - 1) + 1):(nr*ii)) ...
        = A(I_nonzero, n*(i - 1) + find(I_nonzero));
```

```
    end
    br = b(constr_ind);
    Cr = C(I_nonzero, I_nonzero);

end
```

C The DIMACS errors

For the sake of completeness in this section we describe the DIMACS errors, which are commonly used to measure the accuracy of approximate solutions X of (P) and y of (D).

Define the operator $\mathcal{A} : \mathbb{R}^m \to \mathcal{S}^n$ and its adjoint as

$$\begin{aligned} \mathcal{A}(X) &= (A_1 \bullet X, \ldots, A_m \bullet X), \\ \mathcal{A}^*(y) &= \sum_{i=1}^{m} y_i A_i. \end{aligned}$$

Suppose we are given an approximate solution X of (P) and an approximate solution y of (D). For brevity, define $Z = C - \mathcal{A}^*(y)$.

Then the DIMACS error measures are defined as follows:

$$\begin{aligned} \text{err}_1 &= \frac{\|\mathcal{A}(X) - b\|_2}{1+ \|b\|_\infty}, \\ \text{err}_2 &= \max\left\{0, \frac{-\lambda_{\min}(X)}{1+ \|b\|_\infty}\right\}, \\ \text{err}_3 &= \frac{\|\mathcal{A}^*(y) - C - Z\|_F}{1+ \|C\|_\infty}, \\ \text{err}_4 &= \max\left\{0, \frac{-\lambda_{\min}(Z)}{1+ \|C\|_\infty}\right\}, \\ \text{err}_5 &= \frac{b^\top y - C \bullet X}{1 + |C \bullet X| + |b^\top y|}, \\ \text{err}_6 &= \frac{Z \bullet X}{1 + |C \bullet X| + |b^\top y|}. \end{aligned}$$

In the above equations we use the following notation. If $M = (m_{ij}) \in \mathcal{S}^n$, then we write $\|M\|_F$ for the Frobenius norm of M and $\|M\|_\infty$ for the infinity norm of M, i.e.,

$$\begin{aligned} \|M\|_F &= \sqrt{\sum_{i,j} m_{ij}^2} \\ \|M\|_\infty &= \max_{i,j} |m_{ij}|. \end{aligned}$$

We also write $\lambda_{\min}(M)$ for the smallest eigenvalue of M.

D Dual solution recovery

In this section we address the following question: suppose we preprocessed the problem (P) by Sieve-SDP, then computed an optimal solution of the preprocessed SDP, (P_{pre}), and of its dual, (D_{pre}). Can we compute an optimal solution of the original primal (P) and of its dual (D)? The answer to the first question (primal solution recovery) is easy, while the issue of dual solution recovery is much more subtle.

First let us look at primal solution recovery. Since Sieve-SDP deletes rows and columns from the variable matrix X that are always zero anyway, if X^{pre} is an optimal solution of (P_{pre}), then by simply padding X^{pre} with zeroes we obtain an optimal solution of (P).

Next we discuss dual solution recovery. For simplicity we first assume that Sieve-SDP performed just one iteration. Further, we also assume that in the Basic Step (in Fig. 1) it eliminated the constraint $A_1 \bullet X = 0$, where

$$A_1 = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix},$$

with $D \succ 0$ and we let r be the order of D.

Next, let us write out (D_{pre}) :

$$\begin{aligned} &\sup_y \sum_{i=2}^m b_i y_i \\ &\text{s.t. } C - \sum_{i=2}^m y_i A_i \in \begin{pmatrix} \times & \times \\ \times & \oplus \end{pmatrix}, \end{aligned} \tag{D_{pre}}$$

where the notation means that the lower right $(n - r) \times (n - r)$ principal block of $C - \sum_{i=2}^m y_i A_i$ is positive semidefinite, and the rest is arbitrary. Thus clearly

$$\text{val}\,(D) \le \text{val}(D_{\text{pre}}), \tag{10}$$

since (D_{pre}) has a feasible region which is at least as large as that of (D) (and usually it is larger). Assume that $y^{\text{pre}} = (y_2^{\text{pre}}, \ldots, y_m^{\text{pre}})$ is an optimal solution of (D_{pre}). Our recovery procedure, which we call Basic-Recovery, fixes y^{pre} and seeks y_1 such that (y_1, y^{pre}) is feasible in (D), i.e.,

$$y_1 A_1 + \sum_{i=2}^m y_i^{\text{pre}} A_i \preceq C. \tag{11}$$

We do this by a very basic linesearch: we first try the values $y_1 = 0, -1,$ and -2. If these all fail, then we try $y_1 = -100$. If we fail with $y_1 = -100$, we stop; otherwise we test $y_1 = -3, -4, \ldots$ and find the largest y_1 such that (11) holds.

To test whether (11) holds, again Cholesky factorization comes into play: using it, we test whether

$$C - \left(y_1 A_1 + \sum_{i=2}^{m} y_i^{\mathrm{pre}} A_i \right) + 10^{-6} I \succ 0 \, \text{holds.}$$

Basic-Recovery is inspired by the dual solution recovery procedure in [34], which builds on the ideas in [32], and it assumes that the dual problem (D) is reduced.[5]

The procedure Basic-Recovery may fail. To see why, first assume it succeeds, i.e., it computes a feasible solution of (D). Since y_1 has zero objective coefficient in (D), this solution has objective value $\mathrm{val}(D_{\mathrm{pre}})$, hence by inequality (10) it is optimal in (D), thus $\mathrm{val}\,(D) = \mathrm{val}(D_{\mathrm{pre}})$. Conversely, if $\mathrm{val}\,(D) < \mathrm{val}(D_{\mathrm{pre}})$, then Basic-Recovery *must* fail.

Example 4 (Example 3 continued) When we apply Sieve-SDP to the SDP (4), it deletes the first row and first column in all matrices and it also deletes the first constraint.

Let us write out (D_{pre}) again for this problem (i.e., repeat the SDP (7)):

$$\begin{aligned} &\sup_{y_2} \; y_2 \\ &\text{s.t.} \; y_2 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \preceq \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \end{aligned} \tag{12}$$

whose optimal solution is $y_2^{\mathrm{pre}} = 1$.

Thus, Basic-Recovery seeks y_1 such that

$$y_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \preceq \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and clearly there is no such y_1.

One can construct more sophisticated examples in which $\mathrm{val}(D_{\mathrm{pre}}) = \mathrm{val}\,(D)$, but Basic-Recovery still fails.

We next look at dual solution recovery when Sieve-SDP deleted several constraints: then we run Basic-Recovery to find the corresponding y_i sequentially. For simplicity assume that Sieve-SDP deleted constraints $1, 2, \ldots, k$ and we found an optimal primal and dual solution of the resulting SDP (by Mosek). We then attempt to find an optimal dual solution of the SDP obtained by deleting only constraints $1, \ldots, k-1$; then of the SDP obtained by deleting only constraints $1, \ldots, k-2$; and so on.

To conclude this section we make the point that dual solution recovery is much more difficult in SDP than in linear programming. We thus implemented an "ideal" recovery procedure, which we call Ideal-Recovery. It works as follows. Suppose $y^{\mathrm{pre}} = (y_{k+1}^{\mathrm{pre}}, \ldots, y_m^{\mathrm{pre}})$ is an optimal dual solution of the SDP obtained by deleting constraints $1, \ldots, k$. Ideal-Recovery fixes y^{pre}, then calls Mosek to find a feasible solution $(y_1, \ldots, y_k)$ of

[5] See Remark 1 about how the primal and dual are defined in [34].

Table 12 Dual solution recovery by four methods

Method	# Reduced feasible	# Success	# Failure	Success rate	Time (s)
pd1	137	23	114	16.8%	154.75
pd2	158	39	119	24.7%	172.13
Sieve-SDP + Basic-Recovery	143	25	118	17.5%	12.62
Sieve-SDP + Ideal-Recovery	143	103	40	72.0%	1313.57

Table 13 Dual solution recovery assuming the tightest standard for "success"

Method	# Reduced feasible	# Success	# Failure	Success rate	Time (s)
pd1	137	19	118	13.9%	154.75
pd2	158	34	124	21.5%	172.13
Sieve-SDP + Basic-Recovery	143	25	118	17.5%	12.62
Sieve-SDP + Ideal-Recovery	143	17	126	11.9%	1313.57

$$\sum_{i=1}^{k} y_i A_i + \sum_{i=k+1}^{m} y_i^{\text{pre}} A_i \preceq C. \tag{13}$$

Table 12 shows on how many instances pd1, pd2, Sieve-SDP+Basic-Recovery and Sieve-SDP+Ideal-Recovery succeeded. (Note that they succeeded on overlapping, but different problem sets, as a preprocessor may reduce an SDP, while another preprocessor may not reduce the same SDP. We do not report results with dd1 and dd2, since they only reduced a very small percentage of the instances.)

What do we mean by "success"? For pd1 and pd2 it means that their dual solution recovery code reported success. For Sieve-SDP+Basic-Recovery it means that it succeeded in every iteration: it computed the y_i for every deleted constraint. For Sieve-SDP+Ideal-Recovery it means that Mosek did *not* report that (13) is infeasible.

Next we made the criterion of "success" more rigorous: we redefined "success" as returning a pair of primal-dual optimal solutions whose largest DIMACS error in absolute value is at most 10^{-6}. Table 13 shows the results: now Sieve-SDP+Basic-Recovery is the winner, as it beats the supposedly perfect Sieve-SDP+Ideal-Recovery procedure.

Nevertheless, none of the methods do very well, and Table 13 shows that dual solution recovery in facial reduction remains a challenge, and an interesting area for further research.

References

1. MOSEK ApS. Mosek optimization toolbox for MATLAB 8.0.0.94 (2017)
2. Baston, V.: Extreme copositive quadratic forms. Acta Arith. **15**(3), 319–327 (1969)
3. Borwein, J.M., Wolkowicz, H.: Facial reduction for a cone–convex programming problem. J. Aust. Math. Soc. **30**, 369–380 (1981)

4. Borwein, J.M., Wolkowicz, H.: Regularizing the abstract convex program. J. Math. Anal. Appl. **83**, 495–530 (1981)
5. Boyd, S., Mueller, M.T., O'Donoghue, B., Wang, Y.: Performance bounds and suboptimal policies for multi-period investment. Found. Trends® Optim. **1**(1), 1–72 (2014)
6. Burer, S., Monteiro, R.D.C.: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Math. Program. **95**(2), 329–357 (2003)
7. Burer, S., Monteiro, R.D.C., Zhang, Y.: Solving a class of semidefinite programs via nonlinear programming. Math. Program. **93**(1), 97–122 (2002)
8. Burton, S., Vinzant, C., Youm, Y.: A real stable extension of the Vamos matroid polynomial. arXiv preprint arXiv:1411.2038 (2014)
9. Cheung, V., Wolkowicz, H., Schurr, S.: Preprocessing and regularization for degenerate semidefinite programs. In: Bailey, D., Bauschke, H.H., Garvan, F., Théra, M., Vanderwerff, J.D., Wolkowicz, H. (eds.) Proceedings of Jonfest: A Conference in Honour of the 60th Birthday of J. Borwein. Springer, Berlin (2013)
10. Diananda, P.H.: On non-negative forms in real variables some or all of which are non-negative. Proc. Camb. Philos. Soc. **58**, 17–25 (1962)
11. Dressler, M., Iliman, S., de Wolff, T.: An approach to constrained polynomial optimization via nonnegative circuit polynomials and geometric programming. J. Symb. Comput. **91**, 149–172 (2019)
12. Drusviyatskiy, D., Krislock, N., Voronin, Y.L., Wolkowicz, H.: Noisy Euclidean distance realization: robust facial reduction and the Pareto frontier. SIAM J. Optim. **27**(4), 2301–2331 (2017)
13. Drusviyatsky, D., Pataki, G., Wolkowicz, H.: Coordinate shadows of semi-definite and Euclidean distance matrices. SIAM J. Optim. **25**(2), 1160–1178 (2015)
14. Fawzi, H., Parrilo, P.A.: Self-scaled bounds for atomic cone ranks: applications to nonnegative rank and CP-rank. Math. Program. **158**(1–2), 417–465 (2016)
15. Friberg, H.: Facial reduction heuristics and the motivational example of mixed integer conic optimization. Technical report, Optimization Online (2016)
16. Fujisawa, K., Fukuda, M., Kobayashi, K., Kojima, M., Nakata, K., Nakata, M., Yamashita, M.: SDPA (semidefinite programming algorithm) and SDPA-GMP User's Manual—Version 7.1.0. Department of Mathematical and Computing Sciences, Institute of Technology, Tokyo. Research Reports on Mathematical and Computing Sciences Series B-448 (2008)
17. Fujisawa, K., Kojima, M., Nakata, K., Yamashita, M.: SDPA (semidefinite programming algorithm) User's Manual Version 6.2. 0. Department of Mathematical and Computing Sciences, Institute of Technology, Tokyo. Research Reports on Mathematical and Computing Sciences Series B: Operations Research (2002)
18. Henrion, D., Lasserre, J.-B.: Detecting global optimality and extracting solutions in GloptiPoly. In: Positive Polynomials in Control, pp. 293–310. Springer, Berlin (2005)
19. Henrion, D., Lasserre, J.-B., Löfberg, J.: GloptiPoly 3: moments, optimization and semidefinite programming. Optim. Methods Softw. **24**(4–5), 761–779 (2009)
20. Henrion, D., Naldi, S., ed Din, M.S.: Exact algorithms for linear matrix inequalities. SIAM J. Optim. **26**(4), 2512–2539 (2016)
21. Kočvara, M., Stingl, M.: Pennon: a code for convex nonlinear and semidefinite programming. Optim. Methods Softw. **18**(3), 317–333 (2003)
22. Krislock, N., Wolkowicz, H.: Explicit sensor network localization using semidefinite representations and facial reductions. SIAM J. Optim. **20**, 2679–2708 (2010)
23. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**(3), 796–817 (2001)
24. Liu, M., Pataki, G.: Exact duals and short certificates of infeasibility and weak infeasibility in conic linear programming. Math. Program. **167**(2), 435–480 (2018)
25. Liu, Y., Ryu, E.K., Yin, W.: A new use of Douglas–Rachford splitting for identifying infeasible, unbounded, and pathological conic programs. Math. Program. 1–29 (2018)
26. Lasserre, J.B., Magron, V.: In SDP relaxations, inaccurate solvers do robust optimization. arXiv preprint arXiv:1811.02879 (2018)
27. Mittelmann, H.D.: An independent benchmarking of SDP and SOCP solvers. Math. Program. **95**(2), 407–430 (2003)
28. Parrilo, P.: Semidefinite programming relaxations for semialgebraic problems. Math. Program. **96**(2), 293–320 (2003)

29. Pataki, G.: The geometry of semidefinite programming. In: Saigal, R., Vandenberghe, L., Wolkowicz, H. (eds.) Handbook of Semidefinite Programming. Kluwer Academic Publishers, Canterbury (2000)
30. Pataki, G.: A simple derivation of a facial reduction algorithm and extended dual systems. Technical Report, Columbia University (2000)
31. Pataki, G.: Strong duality in conic linear programming: facial reduction and extended duals. In: Bailey, D., Bauschke, H.H., Garvan, F., Théra, M., Vanderwerff, J.D., Wolkowicz, H. (eds.) Proceedings of Jonfest: A Conference in Honor of the 60th Birthday of Jon Borwein. Springer, Berlin (2013)
32. Pataki, G.: Bad semidefinite programs: they all look the same. SIAM J. Optim. **27**(1), 146–172 (2017)
33. Pataki, G., Schmieta, S.: The DIMACS library of mixed semidefinite-quadratic-linear programs. http://dimacs.rutgers.edu/Challenges/Seventh/Instances/
34. Permenter, F., Parrilo, P.: Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. Math. Program. **171**(1–2), 1–54 (2018)
35. Permenter, F., Friberg, H.A., Andersen, E.D.: Solving conic optimization problems via self-dual embedding and facial reduction: a unified approach. SIAM J. Optim. **27**(3), 1257–1282 (2017)
36. Posa, M., Tobenkin, M., Tedrake, R.: Lyapunov analysis of rigid body systems with impacts and friction via sums-of-squares. In: Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control, pp. 63–72. ACM, New York (2013)
37. Quist, A.J., de Klerk, E., Roos, C., Terlaky, T.: Copositive relaxation for general quadratic programming. Optim. Methods Softw. **9**(1–3), 185–208 (1998)
38. Renegar, J.: A Mathematical View of Interior-Point Methods in Convex Optimization, vol. 3. SIAM, Philadelphia (2001)
39. Sturm, J.: Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. Optim. Methods Softw. **11**(1–4), 625–653 (1999)
40. Sun, D., Toh, K.-C., Yang, L.: A convergent 3-block semi-proximal alternating direction method of multipliers for conic programming with 4-type constraints. SIAM J. Optim. **25**(2), 882–915 (2015)
41. Tanigawa, S.: Singularity degree of the positive semidefinite matrix completion problem. SIAM J. Optim. **27**(2), 986–1009 (2017)
42. Trefethen, L.N., Bau III, D.: Numerical Linear Algebra, vol. 50. SIAM, Philadelphia (1997)
43. Tunçel, L.: Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization. Fields Institute Monographs. American Mathematical Society, Providence (2011)
44. Tutuncu, R.H., Toh, K.-C., Todd, M.J.: Solving semidefinite-quadratic-linear programming using SDPT3. Math. Program. **95**, 189–217 (2003)
45. Wagner, D.G., Wei, Y.: A criterion for the half-plane property. Discrete Math. **309**(6), 1385–1390 (2009)
46. Waki, H.: How to generate weakly infeasible semidefinite programs via Lasserre's relaxations for polynomial optimization. Optim. Lett. **6**(8), 1883–1896 (2012)
47. Waki, H., Muramatsu, M.: Facial reduction algorithms for conic optimization problems. J. Optim. Theory Appl. **158**(1), 188–215 (2013)
48. Waki, H., Nakata, M., Muramatsu, M.: Strange behaviors of interior-point methods for solving semidefinite programming problems in polynomial optimization. Comput. Optim. Appl. **53**(3), 823–844 (2012)
49. Yang, L., Sun, D., Toh, K.-C.: SDPNAL+: a majorized semi-smooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. Math. Program. Comput. **7**(3), 331–366 (2015)
50. Zhao, X.Y., Sun, D., Toh, K.-C.: A Newton-CG augmented Lagrangian method for semidefinite programming. SIAM J. Optim. **20**(4), 1737–1765 (2010)