A COMPARISON OF THE PERFORMANCE OF SDP SOLVERS

by

Celia Flicker

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Mathematics
with Specialization in Operations Research and Statistics

NEW MEXICO TECH
SCIENCE • ENGINEERING • RESEARCH UNIVERSITY

New Mexico Institute of Mining and Technology
Socorro, New Mexico
January, 2021

ABSTRACT

This thesis compares the performance of several first-order positive semidefinite programming solvers on two sets of benchmarking problems. The performance is evaluated by robustness, accuracy, and speed. The four solvers tested are the boundary point method and its generalization, the alternating direction augmented Lagrangian method, and the splitting conic solver. Over all, the splitting conic solver was generally fastest and the alternating direction augmented Lagrangian method was most robust, while the boundary point method was generally least robust and was usually outperformed in speed and robustness by its alternating-direction-augmented-Lagrangian-like generalization.

Keywords: Optimization; Benchmarking; Semidefinite Programming; First-Order Methods; Large SDPs

ACKNOWLEDGMENTS

I'd like to thank my advisor, Brian Borchers, for all of his support and guidance. I would also like to thank Oleg Makhnin and Anwar Hossain for serving on my thesis committee.

This thesis was typeset with LaTeX[1] by the author.

[1] The LaTeX document preparation system was developed by Leslie Lamport as a special version of Donald Knuth's TeX program for computer typesetting. TeX is a trademark of the American Mathematical Society. The LaTeX macro package for the New Mexico Institute of Mining and Technology thesis format was written by John W. Shipman.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

This thesis is accepted on behalf of the faculty of the Institute by the following committee:

Brian Borchers

Academic and Research Advisor

Anwar Hossain

Oleg Makhnin

I release this document to the New Mexico Institute of Mining and Technology.

Celia Flicker November 18, 2020

.

CHAPTER 1

INTRODUCTION

Semidefinite Programming problems, or SDPs, are convex optimization problems in which the trace of the product of an $n \times n$ symmetric matrix and a constant objective $n \times n$ symmetric matrix is minimized, subject to m equality constraints on the trace of the same matrix and constant constraint $n \times n$ symmetric matrices, and the constraint that the matrix is in the cone of $n \times n$ symmetric positive definite matrices, $\mathbb{S}^n_+$:

$$\begin{array}{lrcl} \min & \operatorname{tr}(CX) & & \\ (P) & \operatorname{tr}(A_i X) & = & b_i, i = 1, \dots, m \\ & X & \in & \mathbb{S}^n_+ \end{array} \tag{1.1}$$

with the dual of the problem being to maximize the scalar product of a vector of length m and the constraint vector, subject to the constraint that the difference between the objective matrix and the linear combination of the equality constraint matrices when scaled by the elements of the vector is symmetric and positive semidefinite.

$$\begin{array}{lrcl} \max & b^T y & & \\ (D) & \sum_{i=1}^m y_i A_i + Z & = & C \\ & Z & \in & \mathbb{S}^n_+ \end{array} \tag{1.2}$$

There are well-known primal-dual interior point methods to solve these problems quickly and accurately, but they do not scale well to very large problems, necessitating the use of first-order methods [18]. In this thesis, we benchmark and compare the performance of first-order solvers on large and small SDPs.

Generally, an interior point method can solve SDPs in polynomial time. However, if the number of $n \times n$ constraint matrices m is of order $O\left(n^2\right)$, then the time and memory complexity required for each iteration becomes unreasonably large, $O\left(n^6\right)$ and $O\left(m^2\right)$ respectively, to find a (usually dense) $m \times m$ positive definite Schur complement matrix and compute its Cholesky factorization. For particularly large problems, the matrices that need to be calculated in each iteration may not even fit into a computer's memory [19].

In contrast, most first-order methods are based on the augmented Lagrangian method, and involve $O\left(m\right)$ and $O\left(n^2\right)$ storage. Usually they involve solving an $m \times m$ system of equations which could *theoretically* require up to $O\left(m^2\right)$ storage, but this system is typically very sparse, making storage and calculation more efficient, and allowing it to fit in memory and be solved quickly.

SDPs were significantly used starting in the early 1990s, for some combinatorial optimization and graph theory problems. Later, they were also used for problems in structural optimization, signal processing, circuit design, algebraic geometry, communications and information theory, quantum computing, and finance [4, 18]. SDPs also arise in analysis of moments of statistical distributions, and in relaxations to simpler convex optimization problems to make them more robust to uncertainty in inputs, and in many other fields [6]. Primal-dual interior-point methods, such as the predictor-corrector infeasible interior-point method SDPA started in 1995 and fully implemented in 1998 [20], self-dual minimization using a centering-predictor-corrector method SeDuMi created in 1998 [15], infeasible predictor-corrector barrier method CSDP created in 1999 [2], and infeasible interior-point path following method SDPT3 created in 2001 [17], were developed, and generally work well on small and medium sized SDPs [10]. Starting in the mid 2000s, first-order methods began to be developed to solve larger problems, including the general first-order methods investigated in this thesis: the boundary point method BPM in 2006 [14], its generalization MPRW in 2009 [9], the alternating-direction augmented Lagrangian method SDPAD in 2009 [19], and the self-dual alternating-direction method of multipliers, or splitting conic solver, SCS in 2016 [12].

CHAPTER 2

POSITIVE SEMIDEFINITE PROGRAMMING DUALITY AND ALGORITHMS

We standardize the form of the SDP problem with C, $A_1, \ldots, A_m \in \mathbb{S}^n$ ($n \times n$ symmetric matrices) and $b \in \mathbb{R}^m$ to:

$$\begin{array}{rrcl} & \min \quad \langle C, X\rangle & & \\ (P) & A\left(X\right) & = & b \\ & X & \succeq & 0 \end{array} \tag{2.1}$$

where $\langle\, , \,\rangle$ is the Frobenius inner product of two matrices of the same size, which defines the Frobenius norm $\|\cdot\|_F^2$:

$$\begin{gathered} \langle P, Q\rangle = \sum_i \sum_j P_{ij} Q_{ij} = \operatorname{tr}\left(P^T Q\right) \\ \|P\|_F^2 = \langle P, P\rangle \end{gathered} \tag{2.2}$$

(note that, because the matrices in question for these problems are all square and symmetric, the transpose is omitted and the product is written as $\operatorname{tr}\left(PQ\right)$ in this thesis, which is equivalent), the function $A : \mathbb{S}^n \rightarrow \mathbb{R}^m$ is

$$A(X) := \begin{bmatrix} \langle A_1, X\rangle \\ \vdots \\ \langle A_m, X\rangle \end{bmatrix}, \tag{2.3}$$

and $\succeq$ is the conic inequality defined in relation to $\mathbb{S}^n_+$, the cone of $n \times n$ symmetric positive semidefinite matrices, so $P \succeq 0 \Leftrightarrow P \in \mathbb{S}^n_+$ and $P \succeq Q \Leftrightarrow P - Q \in \mathbb{S}^n_+$. (Similarly, $\mathbb{S}^n_{++}$ is the cone of $n \times n$ symmetric positive definite matrices, and $P \succ 0 \Leftrightarrow P \in \mathbb{S}^n_{++}$.)

The dual of this problem over $y \in \mathbb{R}^m$ is:

$$\begin{array}{cccc} \max & b^T y & & \\ (D) & A^T(y) & +Z = & C \\ & Z & \succeq & 0 \end{array} \tag{2.4}$$

where $A^T(y) = \sum_{i=1}^m y_i A_i$, derived below.

A primal SDP problem is of the form (2.1). We associate a Lagrange multiplier vector $y \in \mathbb{R}^m$ with the equality constraints, associating element y_i with constraint $b_i - \langle A_i, X \rangle = 0$ for $i = 1, \ldots, m$, and a Lagrange multiplier symmetric matrix $Z \in \mathbb{S}^n$ with the conic inequality constraint $-X \preceq 0$. Then, the Lagrangian of the problem is

$$\mathcal{L}(X, y, Z) = \langle C, X \rangle + \sum_{i=1}^m y_i (b_i - \langle A_i, X \rangle) + \langle Z, -X \rangle \tag{2.5}$$

so the Lagrangian dual function is

$$\begin{aligned} g(y, Z) &= \inf_{X \in \mathbb{S}^n} (\mathcal{L}(X, Z, y)) \\ &= \inf_{X \in \mathbb{S}^n} \left(\langle C, X \rangle + \textstyle\sum_{i=1}^m y_i (b_i - \langle A_i, X \rangle) + \langle Z, -X \rangle \right) \\ &= b^T y + \inf_{X \in \mathbb{S}^n} \left(\langle C, X \rangle - \textstyle\sum_{i=1}^m \langle y_i A_i, X \rangle + \langle -Z, X \rangle \right) \\ &= b^T y + \inf_{X \in \mathbb{S}^n} \left(\langle (C - \textstyle\sum_{i=1}^m y_i A_i - Z), X \rangle \right) \\ &= b^T y + \inf_{X \in \mathbb{S}^n} \left(\left\langle \left(C - A^T(y) - Z \right), X \right\rangle \right). \end{aligned} \tag{2.6}$$

Note that $\left\langle \left(C - A^T(y) - Z \right), X \right\rangle$ is linear in X, and so is not bounded below except when $\left(C - A^T(y) - Z \right)$ is identically 0, so the Lagrangian dual function is

$$g(y, Z) = \begin{cases} b^T y, & C - A^T(y) - Z = 0 \in \mathbb{S}^n \\ -\infty, & \text{otherwise.} \end{cases} \tag{2.7}$$

Then, the Lagrangian dual problem is

$$\begin{aligned} \max \quad & g\left(y, Z\right) \\ (D) \quad & Z \succeq 0 \end{aligned} \tag{2.8}$$

(where Z must be in the dual cone of $\mathbb{S}_+^n$ to have the lower bound $g\left(Z, y\right) \leq \langle C, X\rangle$ for all feasible X, but the positive semidefinite cone is self-dual, so this is simply $Z \in \mathbb{S}_+^n$) and can be written as

$$\begin{aligned} \max \quad & b^T y \\ (D) \quad & A^T\left(y\right) + Z = C \\ & Z \succeq 0 \end{aligned} \tag{2.9}$$

which is the dual problem in its standard form for this thesis: (2.4) [6, p264-265].

Weak duality, from the construction of the Lagrangian dual problem, gives that $\langle C, X\rangle \geq b^T y$ for all feasible X and y. Strong duality, where there exist feasible and optimal $X^\star$, $y^\star$, and $Z^\star$ such that $\langle C, X^\star\rangle = b^T y^\star$, does not always hold, but holds when Slater's condition, that both the primal and dual problems are feasible and either of them is strictly feasible holds, i.e. $\exists X \succ 0 \ni A\left(X\right) = b$ or $\exists y \ni A^T\left(y\right) \succ C$ (i.e. $Z = C - A^T\left(y\right) \succ 0$). When strong duality holds, an optimal solution will satisfy the complementary slackness condition: $\langle X^\star, Z^\star\rangle = 0$ [6, p265-266]. Most solvers assume that Slater's condition and thus strong duality hold, which is the case for many but not all possible SDP problems.

For convenience, we define vectorization of the matrices as an invertible function vec : $\mathbb{S}^n \to \mathbb{R}^{\hat{n}}$, which must maintain

$$\operatorname{vec}\left(P\right)^T \operatorname{vec}\left(Q\right) = \langle P, Q\rangle, \forall P, Q \in \mathbb{S}^n. \tag{2.10}$$

and let $\operatorname{vec}\left(P\right) \in \operatorname{vec}\left(\mathbb{S}_+^n\right)$ denote $P \in \mathbb{S}_+^n$. This allows us to define a single constraint matrix

$$\boldsymbol{A} = \begin{bmatrix} \operatorname{vec}\left(A_1\right)^T \\ \vdots \\ \operatorname{vec}\left(A_m\right)^T \end{bmatrix} \in \mathbb{R}^{m \times \hat{n}}, \tag{2.11}$$

so for $x = \operatorname{vec}(X)$, $\boldsymbol{A}x = A(X)$, and $\boldsymbol{A}^T y = A^T(y)$, for *any* valid vectorization function.

Most SDP solvers use a vectorization $\operatorname{vec}_{\mathrm{C}} : \mathbb{S}^n \to \mathbb{R}^{\hat{n}}$, $\hat{n} = n^2$ of symmetric matrices, by vertically concatenating all of a matrix's columns, in order. This vectorization maintains $\operatorname{vec}_{\mathrm{C}}(P)^T \operatorname{vec}_{\mathrm{C}}(Q) = \langle P, Q\rangle, \forall P, Q \in \mathbb{S}^n$ because $\langle P, Q\rangle = \Sigma_{i=1}^n \Sigma_{j=1}^n P_{ij}Q_{ij}$ is exactly the same as element-wise multiplication and addition of each element of each column of P and Q, which is the same as taking the vector dot product of their $\operatorname{vec}_{\mathrm{C}}$ vectorizations. Therefore, it is a valid vectorization. To illustrate this with an example:

$$\begin{gathered}\operatorname{vec}_{\mathrm{C}}\left(\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}\right)^T \operatorname{vec}_{\mathrm{C}}\left(\begin{bmatrix} 7 & 8 & 9 \\ 8 & 10 & 11 \\ 9 & 11 & 12 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ 4 \\ 5 \\ 3 \\ 5 \\ 6 \end{bmatrix}^T \begin{bmatrix} 7 \\ 8 \\ 9 \\ 8 \\ 10 \\ 11 \\ 9 \\ 11 \\ 12 \end{bmatrix} = 315 \\ \left\langle \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}, \begin{bmatrix} 7 & 8 & 9 \\ 8 & 10 & 11 \\ 9 & 11 & 12 \end{bmatrix} \right\rangle = \operatorname{tr}\left(\begin{bmatrix} 50 & 61 & 67 \\ 91 & 111 & 122 \\ 115 & 140 & 154 \end{bmatrix}\right) = 315 \end{gathered} \tag{2.12}$$

SCS uses a vectorization $\operatorname{vec}_{\mathrm{S}} : \mathbb{S}^n \to \mathbb{R}^{\hat{n}}$, $\hat{n} = \frac{n(n+1)}{2}$ of symmetric matrices, scaling all elements of a matrix but those on the diagonal by $\sqrt{2}$, and vertically concatenating the lower-triangular parts of the columns in order. This vectorization maintains $\operatorname{vec}_{\mathrm{S}}(P)^T \operatorname{vec}_{\mathrm{S}}(Q) = \langle P, Q\rangle, \forall P, Q \in \mathbb{S}^n$, because $\langle P, Q\rangle = \Sigma_{i=1}^n \Sigma_{j=1}^n P_{ij}Q_{ij}$ can be separated out into strictly lower-triangular, diagonal, and strictly upper triangular parts:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} P_{ij}Q_{ij} = \sum_{i=1}^{n}\sum_{j=i+1}^{n} P_{ij}Q_{ij} + \sum_{i=1}^{n} P_{ii}Q_{ii} + \sum_{i=1}^{n}\sum_{j=1}^{i-1} P_{ij}Q_{ij}. \tag{2.13}$$

Then, because the matrices are symmetric, the sum of the products of the strictly lower triangular parts of the matrices is equal to the sum of the products of the strictly upper triangular parts of the matrices:

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n} P_{ij}Q_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{i-1} P_{ij}Q_{ij} = 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} P_{ij}Q_{ij} = \sum_{i=1}^{n}\sum_{j=i+1}^{n} \left(\sqrt{2}P_{ij}\right)\left(\sqrt{2}Q_{ij}\right). \tag{2.14}$$

The entire trace can be simplified to

$$\mathrm{vec}_{\mathrm{S}}\left(P\right)^{T}\mathrm{vec}_{\mathrm{S}}\left(Q\right) = \sum_{i=1}^{n}\sum_{j=i+1}^{n} \left(\sqrt{2}P_{ij}\right)\left(\sqrt{2}Q_{ij}\right) + \sum_{i=1}^{n}\sum_{j=i+1}^{n} P_{ij}Q_{ij} = \langle P, Q\rangle, \tag{2.15}$$

so it is a valid vectorization. To illustrate this with an example:

$$\begin{gathered}
\mathrm{vec}_{\mathrm{S}}\left(\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}\right)^{T} \mathrm{vec}_{\mathrm{S}}\left(\begin{bmatrix} 7 & 8 & 9 \\ 8 & 10 & 11 \\ 9 & 11 & 12 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 2\sqrt{2} \\ 3\sqrt{2} \\ 4 \\ 5\sqrt{2} \\ 6 \end{bmatrix}^{T} \begin{bmatrix} 7 \\ 8\sqrt{2} \\ 9\sqrt{2} \\ 10 \\ 11\sqrt{2} \\ 12 \end{bmatrix} = 315 \\
\left\langle \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}, \begin{bmatrix} 7 & 8 & 9 \\ 8 & 10 & 11 \\ 9 & 11 & 12 \end{bmatrix} \right\rangle = \mathrm{tr}\left(\begin{bmatrix} 50 & 61 & 67 \\ 91 & 111 & 122 \\ 115 & 140 & 154 \end{bmatrix}\right) = 315
\end{gathered} \tag{2.16}$$

Let functions $(\cdot)_{+} : \mathbb{S}^n \to \mathbb{S}^n_{+}$ and $(\cdot)_{-} : \mathbb{S}^n \to \mathbb{S}^n_{-}$ denote the projection of symmetric matrices onto the positive semidefinite and negative semidefinite cones, found by performing a spectral decomposition and replacing the negative or positive eigenvalues with zeros, respectively. For example, given a spectral decomposition

$$P = \sum_{i} \lambda_i v_i v_i^T \tag{2.17}$$

of matrix $P \in \mathbb{S}^n$, so eigenvalues are real and eigenvectors v_i are chosen real and pairwise orthonormal. Then, the projections onto the positive definite and negative definite cones are respectively:

$$\begin{aligned} P_+ &= \sum_{\lambda_i>0} \lambda_i v_i v_i^T, \\ P_- &= \sum_{\lambda_i<0} \lambda_i v_i v_i^T \end{aligned} \tag{2.18}$$

and maintain

$$P = P_+ + P_-. \tag{2.19}$$

By the Eckart-Young-Mirsky Theorem, P_+ is the projection of P onto the positive semidefinite cone, for both the Frobenius norm and the Euclidean norm [7, 6, p399]:

$$\begin{array}{c} \min \quad \|P-Q\|_F \\ Q \succeq 0 \end{array} = \begin{array}{c} \min \quad \|P-Q\|_2 \\ Q \succeq 0 \end{array} = P_+. \tag{2.20}$$

Most solvers also assume that the constraint matrices $A_1, \ldots, A_m$ are linearly independent; if this is not true, then some can be eliminated as redundant constraints until it is. Because of this, for any vectorization vec : $\mathbb{S}^n \to \mathbb{R}^{\hat{n}}$, the resulting constraint matrix $\boldsymbol{A} \in \mathbb{R}^{m\times\hat{n}}$ has full row rank, rank $(\boldsymbol{A}) = m$. Therefore, the matrix $\boldsymbol{A}\boldsymbol{A}^T \in \mathbb{S}^m_{++}$ also has full rank, rank $\left(\boldsymbol{A}\boldsymbol{A}^T\right) = m$, so it is invertible, i.e. $\exists \left(\boldsymbol{A}\boldsymbol{A}^T\right)^{-1} \in \mathbb{S}^m_{++}$. Then, the nonvectorized composition $\left(AA^T\right) = A \circ A^T : \mathbb{R}^m \to \mathbb{R}^m$ is invertible, and

$$\exists \left(AA^T\right)^{-1} : \mathbb{R}^m \to \mathbb{R}^m. \tag{2.21}$$

If C, A_1, ..., A_m share a block diagonal structure, meaning that they are each diagonal concatenations of the same number and sizes of smaller symmetric matrices, then X and Z will share that block diagonal structure, and can be stored more efficiently. Suppose there are n_s diagonal blocks in each of the matrices, possibly of varying sizes. Then, when C_k, X_k, Z_k , and A_{ik} are the kth blocks of C, X, Z, and A_i respectively, the primal and dual problems can be rewritten as

$$\begin{array}{llcll} \min & \sum_{k=1}^{n_s} \langle C_k, X_k \rangle & & & \\ (P) & \sum_{k=1}^{n_s} \langle A_{ik}, X_k \rangle & = & b_i & , \quad i = 1, \ldots, m \\ & \quad X_k & \succeq & 0 & , \quad k = 1, \ldots, n_s \end{array} \tag{2.22}$$

$$\begin{array}{llrcl}
\max & b^T y & & & \\
(D) & \sum_{i=1}^{m} y_i A_{ik} & +Z_k = & C_k & ,\quad k = 1, \ldots, n_s \\
& Z_k & \succeq & 0 & ,\quad k = 1, \ldots, n_s
\end{array} \tag{2.23}$$

for more efficient storage and computation.

Linear Programming problems, of the form

$$\begin{array}{llcl}
\min & c^T x & & \\
(P) & Ax & = & b \\
& x & \geq & 0
\end{array}
,\quad
\begin{array}{llcl}
\max & b^T y & & \\
(D) & A^T y & +z = & c \\
& z & \geq & 0
\end{array} \tag{2.24}$$

with $x, z \in \mathbb{R}_+^{|x|}$, $c \in \mathbb{R}^{|x|}$, can be incorporated as constraints and variables in an SDP, by treating each element of Linear Programming variables x and z as a 1×1 SDP block [18]. SDP solvers therefore can also solve Linear Programming problems, although they often have a separate format to input Linear Programming constraints on SDPs, to perform calculations more efficiently than by converting them to SDP form.

2.1 The Solvers

2.1.1 Boundary Point Method (BPM)

In 2006, Povh, Rendl, and Wiegele [14] developed a boundary point method (BPM) which follows the boundary of the cones of positive semidefinite matrices for the primal and dual problem until it finds a feasible solution, using an augmented Lagrangian function.

BPM uses the problem format:

$$\begin{array}{llcl}
\max & \langle C, X \rangle & & \\
(P) & A(X) & = & b \\
& X & \succeq & 0
\end{array}
,\quad
\begin{array}{llcl}
\min & b^T y & & \\
(D) & A^T(y) & -C = & Z \\
& Z & \succeq & 0
\end{array} \tag{2.25}$$

To convert the BPM variables to the standard variable definitions, substitute $C_{BPM} = -C$ and $y_{BPM} = -y$, to get

$$\begin{array}{llllllll} \max & \langle -C, X\rangle & & & \min & b^T(-y) & & \\ (P) & A(X) & = & b\ , & (D) & A^T(-y) & +C = & Z \\ & X & \succeq & 0 & & Z & \succeq & 0 \end{array} \tag{2.26}$$

which is equivalent to problems (2.1) and (2.4), the standard format of the primal and dual problems.

Applying an augmented Lagrangian method to problem (2.4) is approaching the problem with a boundary point method, which maintains $X \succeq 0$, $Z \succeq 0$, and $XZ = 0$ in each iteration, following the boundary of the cone of positive semidefinite matrices until it reaches the affine subspace from the equality constraints on (2.1) and (2.4), satisfying

$$\begin{aligned} A(X) &= b \\ A^T(y) + Z &= C, \end{aligned} \tag{2.27}$$

i.e. maintaining the complementary slackness condition $\langle X, Z\rangle = 0$ and maintaining positive semidefinite X and Z, until it reaches a point that is feasible for the equality constraints. Then, when it reaches a point satisfying equation (2.27), the complementary slackness condition $\langle X, Z\rangle = 0$ and the positive semidefinite feasibility conditions $X \succeq 0$ and $Z \succeq 0$ will be true because they are true in every iteration, and the equality feasibility conditions (2.27) will then be satisfied, so the solution will be feasible and satisfy the complementary slackness condition, and therefore must be optimal.

The augmented Lagrangian method to solve the dual problem (2.4) uses X as the Lagrange multiplier for constraint $A^T(y) + Z - C = 0$ and introduces a fixed $\sigma > 0$ as a scaling factor on the squared norm of $A^T(y) + Z - C$, which is 0 for feasible solutions:

$$\mathcal{L}_\sigma(y, Z, X) = -b^T y + \left\langle X, A^T(y) + Z - C\right\rangle + \frac{\sigma}{2}\left\|A^T(y) + Z - C\right\|_F^2. \tag{2.28}$$

Defining

$$W(y) = -A^T(y) + C - \frac{1}{\sigma}X \tag{2.29}$$

the augmented Lagrangian can be written as

$$\begin{aligned}\mathcal{L}_\sigma\left(y,Z,X\right) &= -b^Ty+\left\langle X, Z-W\left(y\right)-\tfrac{1}{\sigma}X\right\rangle+\tfrac{\sigma}{2}\left\langle Z-W\left(y\right)-\tfrac{1}{\sigma}X, Z-W\left(y\right)-\tfrac{1}{\sigma}X\right\rangle \\ &= -b^Ty+\tfrac{\sigma}{2}\left\|Z-W\left(y\right)\right\|_F^2-\tfrac{1}{2\sigma}\left\|X\right\|_F^2.\end{aligned} \tag{2.30}$$

Then, removing the terms that depend only on X, the augmented Lagrangian objective function is

$$f\left(y,Z\right)=-b^Ty+\frac{\sigma}{2}\left\|Z-W\left(y\right)\right\|_F^2 \tag{2.31}$$

so each iteration of the augmented Lagrangian method to solve the dual problem (2.4) consists of minimizing $f\left(y,Z\right)$ with some fixed $X\succeq 0$ over $Z\succeq 0$ to get y and Z, then updating X as $X\leftarrow X+\sigma\left(A^T\left(y\right)+Z-C\right)$. The process iterates until convergence, following the standard method of multipliers or augmented Lagrangian update [5].

The inner minimization of $f\left(y,Z\right)$ over $Z\succeq 0$ is the main computation of each iteration. It can be written

$$\begin{aligned} \min \quad & -b^Ty+\tfrac{\sigma}{2}\left\|Z-W\left(y\right)\right\|_F^2 \\ & y\in\mathbb{R}^m \\ & Z\succeq 0 \end{aligned} \tag{2.32}$$

which is a convex quadratic SDP. This problem is too hard to solve in every iteration, so instead it is iteratively approximately solved. Introducing the Lagrange multiplier $V\succeq 0$ for the constraint $-Z\preceq 0$, this problem's Lagrangian is

$$\mathcal{L}\left(y,Z,V\right)=f\left(y,Z\right)-\left\langle V,Z\right\rangle \tag{2.33}$$

whose Karush-Kuhn-Tucker (KKT) necessary and sufficient conditions for optimality [6, p244] are

$$\begin{aligned}\nabla_y\mathcal{L}\left(y,Z,V\right)&=-b+\sigma A\left(A^T\left(y\right)+Z-C+\tfrac{1}{\sigma}X\right)=0\\ \nabla_Z\mathcal{L}\left(y,Z,V\right)&=\sigma\left(A^T\left(y\right)+Z-C+\tfrac{1}{\sigma}X\right)-V=0\\ Z\succeq 0,\ V&\succeq 0,\ VZ=0.\end{aligned} \tag{2.34}$$

Expanding the gradiant conditions shows that y, Z is optimal if and only if there exists V such that

$$\begin{aligned}\left(AA^T\right)(y) &= \tfrac{1}{\sigma}b - A\left(Z - C + \tfrac{1}{\sigma}X\right)\\ V &= \sigma Z - \sigma W(y)\\ Z \succeq 0,\ V &\succeq 0,\ VZ = 0.\end{aligned} \tag{2.35}$$

For fixed y, the problem $\min_{Z\succeq 0} f(y, Z)$ is a projection onto the cone of positive semidefinite matrices. Therefore, Z must also satisfy the projection condition

$$Z = W(y)_+ \tag{2.36}$$

defined earlier. Then, the necessary and sufficient conditions can be reformulated as follows: The triple (y, Z, V) satisfies (2.35) if and only if

$$\begin{aligned}\left(AA^T\right)(y) &= \tfrac{1}{\sigma}b - A\left(Z - C + \tfrac{1}{\sigma}X\right)\\ Z &= W(y)_+\\ V &= -\sigma W(y)_- .\end{aligned}$$

Thus, while keeping Z constant, y can be obtained from the linear system (2.35), and while keeping y constant, Z can be obtained from (2.36), so the inner minimization can be calculated by alternately solving (2.35) for y and (2.36) for Z.

Finally, after the inner minimization is done, the update on X is given by the standard method of multipliers update

$$X \leftarrow X + \sigma\left(A^T(y) + Z - C\right) = \sigma Z - \sigma W(y) = -\sigma W(y)_- = V \succeq 0. \tag{2.37}$$

The method uses a sequence of tolerances $\left\{\varepsilon^{(k)}\right\} \to 0$, $\sum_k \varepsilon^{(k)} < \infty$ for the inner optimization, and a single tolerance ε for the outer optimization. It starts with $k = 0$ and initializes $X^{(0)} = Z^{(0)} = 0 \succeq 0$. The inner optimization solves (2.35) for $y^{(k)}$, calculates W from (2.29), $Z^{(k)}$ from (2.36), and $V^{(k)} = -\sigma W(y)_-$, repeating until the inner optimization's infeasibility is within tolerance for the iteration $\left\|A\left(V^{(k)}\right) - b\right\|_F \leq \sigma\varepsilon^{(k)}$, then updating X as $X^{(k+1)} = V^{(k)}$, incrementing k, and calculating whether the outer problem's infeasibility is within tolerance $\left\|A^T\left(y^{(k)}\right)+Z^{(k)}-C\right\|_F \leq \varepsilon$, and starting the next iteration's inner optimization if not.

The stopping condition of the inner optimization k, with $\left(y^{(k)}, Z^{(k)}\right)$ corresponding to a fixed $X^{(k)}$ and σ, is the scaled primal infeasibility of $V^{(k)}$ as a new value of $X^{(k+1)}$:

$$\left\|A\left(V^{(k)}\right)-b\right\|_F \leq \sigma\varepsilon^{(k)} \tag{2.38}$$

and using

$$\begin{aligned} Z^{(k)} &= W\left(y^{(k)}\right)_+ \\ V^{(k)} &= \sigma\left(A^T\left(y^{(k)}\right)+Z^{(k)}-C\right)+X^{(k)} \end{aligned} \tag{2.39}$$

the condition can be rewritten as

$$\left\|A\left(A^T\left(y^{(k)}\right)+Z^{(k)}-C\right)-\frac{1}{\sigma}\left(A\left(X^{(k)}\right)-b\right)\right\|_F \leq \varepsilon^{(k)} \tag{2.40}$$

to show the primal and dual linear components of the error.

The stopping condition of the outer optimization

$$\left\|A^T\left(y^{(k)}\right)+Z^{(k)}-C\right\|_F \leq \varepsilon \tag{2.41}$$

is simply checking whether the dual linear infeasibility is sufficiently small.

Although the BPM paper shows that the method eventually converges for any choice of $\sigma > 0$ [14], choice of σ affects the speed at which primal and dual feasibility are reached, with large σ causing the inner problem to become increasingly difficult while the outer problem of dual feasibility is reached more easily, and small σ causing primal feasibility in the inner problem to be reached easily while the outer problem of dual feasibility is difficult and slow.

For the computations, BPM uses the vectorization $\mathrm{vec}_C : \mathbb{S}^n \to \mathbb{R}^{\hat{n}}$, $\hat{n} = n^2$ defined earlier. To speed up computation of $y^{(k)}$ from (2.35)

$$\boldsymbol{A}\boldsymbol{A}^T y^{(k)} = \frac{1}{\sigma}b - \boldsymbol{A}\left(z^{(k)} - c + \frac{1}{\sigma}x^{(k)}\right), \tag{2.42}$$

the sparse Cholesky factorization of $\boldsymbol{A}\boldsymbol{A}^T = LL^T$ (where L is lower-triangular with all positive diagonal entries, which exists because $\boldsymbol{A}\boldsymbol{A}^T \in \mathbb{S}^m_{++}$ and every symmetric positive definite matrices has a Cholesky factorization [6, p118]) is precomputed once at the beginning, and then in each iteration the system is solved for $y^{(k)}$ with simple vector and scalar arithmetic, matrix-vector multiplication, and back substitution. Additionally, a full spectral decomposition of $W\left(y^{(k)}\right)$ is performed in each inner iteration, which is generally the most computationally expensive step.

Algorithm 2.1 Boundary Point Method (From Table 2 of [14])

```
Select $\sigma > 0$, $\left\{\varepsilon^{(k)}\right\} \to 0$, $\varepsilon > 0$.
Set $k = 0$, $x^{(k)} = 0 \in \mathrm{vec}_C\left(\mathbb{S}^n\right)$, $Z^{(k)} = 0$.
Precompute sparse Cholesky factorization of $\boldsymbol{AA}^T$.
repeat until $\delta_{outer} < \varepsilon$:
(Outer iteration for $k = 0, 1, \dots$)

    repeat until $\delta_{inner} < \sigma\varepsilon^{(k)}$:
    (Inner iteration with $X^{(k)}$ and $\sigma$ held constant)
        Solve for $y^{(k)}$ using Cholesky factorization:
            $AA^T\left(y^{(k)}\right) = \frac{1}{\sigma}b - A\left(Z^{(k)} - C + \frac{1}{\sigma}X^{(k)}\right)$
        $W(y) = -A^T\left(y^{(k)}\right) + C - \frac{1}{\sigma}X^{(k)}$
        $Z^{(k)} = W_+$
        $V^{(k)} = -\sigma W_-$
        $\delta_{inner} = \left\|A\left(V^{(k)}\right) - b\right\|_2$
    end
    $X^{(k+1)} = V^{(k)}$
    $k = k + 1$
    $\delta_{outer} = \left\|A^T\left(y^{(k)}\right) + Z^{(k)} - C\right\|_F$
end
```

2.1.2 Boundary Point Method Generalization (MPRW)

In 2009, Malick, Povh, Rendl, and Wiegele [9] developed a generalization of BPM, an algorithm which uses two equivalent quadratic regularizations, one for the primal problem and one for the dual problem, to generate a sequence of primal and dual solutions that converge to a feasible approximate solution for a given tolerance. For convenience, this algorithm is sometimes refered to by the authors' initials, as MPRW. The regularizations improve the numerical stability of the solutions against perturbations in the problem. For the primal problem, they apply a Moreau-Yosida regularization to replace the primal problem (2.1) with the equivalent problem

$$\begin{gathered} \min \langle C, X \rangle + \tfrac{1}{2\sigma} \| X - Y \|_2^2 \\ A(X) = b \\ X \succeq 0 \\ Y \in \mathbb{S}^n \end{gathered} \tag{2.43}$$

introducing a parameter $\sigma > 0$. Then, they separate this into two steps,

$$F_\sigma(Y) = \begin{gathered} \min \langle C, X \rangle + \tfrac{1}{2\sigma} \| X - Y \|_2^2 \\ A(X) = b \\ X \succeq 0 \end{gathered} \tag{2.44}$$

as an intermediate step, and the problem solved at

$$\min_{Y \in \mathbb{S}^n} F_\sigma(Y) = \begin{gathered} \min \langle C, X \rangle + \tfrac{1}{2\sigma} \| X - Y \|_2^2 \\ A(X) = b \\ X \succeq 0 \\ Y \in \mathbb{S}^n \end{gathered} = \begin{gathered} \min \langle C, X \rangle \\ A(X) = b \\ X \succeq 0. \end{gathered} \tag{2.45}$$

The X that minimizes $F_\sigma(Y)$ is called the proximal point of Y with parameter σ, denoted

$$P_\sigma(Y) = \begin{gathered} \text{argmin}_X \langle C, X \rangle + \tfrac{1}{2\sigma} \| X - Y \|_2^2 \\ X \succeq 0 \\ A(X) = b. \end{gathered} \tag{2.46}$$

Similarly, for the dual problem (2.4), as in the BPM, they take the augmented Lagrangian of the dual problem

$$\mathcal{L}_\sigma(y, Z; Y) = b^T y - \left\langle Y, A^T(y) + Z - C \right\rangle - \frac{\sigma}{2}\left\| A^T(y) + Z - C \right\|_2^2 \tag{2.47}$$

with parameter $\sigma > 0$, which is the usual Lagrangian for the equivalent problem

$$\begin{aligned} \max b^T y - \tfrac{\sigma}{2}\left\| A^T(y) + Z - C \right\|_2^2 \\ C - A^T(y) = Z \\ Z \succeq 0. \end{aligned} \tag{2.48}$$

Then, the dual of this is

$$\Theta_\sigma(Y) = \begin{array}{c} \max \mathcal{L}_\sigma(y, Z; Y) \\ Z \succeq 0. \end{array} \tag{2.49}$$

(Note that solving the augmented Lagrangian of the dual problem is exactly the approach used in the boundary point method.)

The two quadratic regularizations are equivalent processes on either the primal or dual problem, and using the same scaling factor σ, $\Theta_\sigma(Y) = F_\sigma(Y)$. Then, a solution to the problem can be found by minimizing F_σ or Θ_σ, where $F_\sigma(Y)$ or $\Theta_\sigma(Y)$ is *also* an optimization problem. If strong duality holds, then an optimal solution $(X^\star, y^\star, Z^\star)$ to $F_\sigma(Y)$ and $\Theta_\sigma(Y)$ also satisfies

$$\begin{aligned} X^\star &= \sigma\left(\tfrac{1}{\sigma}Y + A^T(y^\star) - C\right)_+ \\ Z^\star &= -\left(\tfrac{1}{\sigma}Y + A^T(y^\star) - C\right)_- \\ \left(AA^T\right)(y^\star) + A(Z^\star - C) &= \tfrac{1}{\sigma}(b - A(Y)) \end{aligned} \tag{2.50}$$

[9, 14]. (Note that the theoretical derivation in the paper of the algorithm would have led to an algorithm much like the BPM, with $X^{(k)}$ fixed while alternately minimizing over $y^{(k+1)}$ and $Z^{(k+1)}$ until the inner minimization was solved within tolerance, followed by a single update to $X^{(k)}$, but with the new inclusion of numerical adjustments, scaling to normalize the objective matrix and vector, and

dynamic reduction of the scaling factor if the relative dual error becomes larger than the relative primal error. However, in their experiments they determined that the overall performance was better if they performed exactly one iteration of the inner optimization, meaning that each of the dual variables was updated once before the primal variable was updated, rather than approximately solving the inner problem more closely first, giving an overall approach identical to that of SDPAD.)

Algorithm 2.2 Boundary Point Method Generalization (From Algorithm 5.1 of [9])

```
    Scale problem so $\|C\|_2 = \|b\|_2 = 1$ if possible.
    Precompute sparse Cholesky factorization of $\boldsymbol{AA}^T$.
    Choose $\sigma \in [0.1, 10]$, $Y \in \mathbb{S}^n$, and $\varepsilon > 0$.
    Set $Z = 0$.
    repeat until $\delta < \varepsilon$:


        Solve for and update $y$ using Cholesky factorization:
            $AA^T(y) = A(C - Z) + \frac{1}{\sigma}(b - A(Y))$
        $Z \leftarrow -\left(\frac{1}{\sigma}Y + A^T(y) - C\right)_-$
        $X \leftarrow \sigma\left(\frac{1}{\sigma}Y + A^T(y) - C\right)_+$
        $Y \leftarrow X$
        $\delta_p \leftarrow \frac{\|A(X)-b\|_2}{1+\|b\|_\infty}$, $\delta_d \leftarrow \frac{\|C-Z-A^T(y)\|_2}{1+\|C\|_\infty}$, $\delta \leftarrow \max\{\delta_p, \delta_d\}$
        Every 10 iterations, if $\delta_p \leq \delta_d$:
            update $\sigma \leftarrow 0.9\sigma$

    end
    Unscale solution.
```

2.1.3 Alternating Direction SDP Solver (SDPAD)

In 2009, Wen, Golfarb, and Yin developed an augmented Lagrangian alternating direction method for solving SDPs, SDPAD [19]. SDPAD uses the problem format

$$\begin{array}{llll} & \min \ \langle C, X\rangle & & \min \ -b^T y \\ (P) & A(X) = b, & (D) & A^T(y) + Z = C \\ & X \succeq 0 & & Z \succeq 0 \end{array} \tag{2.51}$$

which has (P) exactly the same as the standard definition of the primal problem (2.1), and (D) which can be rewritten as the standard definition of the dual problem (2.4) by the equivalence of min $-b^T y$ and max $b^T y$.

Much like the boundary point method and its generalization [14, 9], they calculate the augmented Lagrangian for the dual problem (2.4),

$$\mathcal{L}_\sigma \left(X, y, Z\right) = -b^T y + \left\langle X, A^T\left(y\right) + Z - C\right\rangle + \frac{\sigma}{2}\left\|A^T\left(y\right) + Z - C\right\|_F^2, \tag{2.52}$$

for $X \in \mathbb{S}^n$, $\sigma > 0$, identical to the augmented Lagrangian calculated there. Starting with $X^{(0)} = 0 \in \mathbb{S}^n$, the augmented Lagrangian method solves on the kth iteration:

$$\min_{\substack{y \in \mathbb{R}^m \\ Z \in \mathbb{S}^n}} \begin{array}{c} \mathcal{L}_\sigma\left(X^{(k)}, y, Z\right) \\ Z \succeq 0 \end{array}, \tag{2.53}$$

for $y^{(k+1)}$ and $Z^{(k+1)}$, and then updates the primal variable $X^{(k+1)}$ with

$$X^{(k+1)} = X^{(k)} + \sigma\left(A^T\left(y^{(k+1)}\right) + Z^{(k+1)} - C\right). \tag{2.54}$$

Because the joint minimization of $\mathcal{L}_\sigma\left(X^{(k)}, y, Z\right)$ with respect to both y and Z can be very slow and expensive, it is faster to instead approximate by minimizing with respect to first one and then the other. Then, the kth iteration of the augmented Lagrangian method is replaced with the steps:

$$y^{(k+1)} = \underset{y \in \mathbb{R}^m}{\operatorname{argmin}} \mathcal{L}_\sigma\left(X^{(k)}, y, Z^{(k)}\right), \tag{2.55}$$

$$Z^{(k+1)} = \underset{Z \in \mathbb{S}^n}{\operatorname{argmin}} \begin{array}{c} \mathcal{L}_\sigma\left(X^{(k)}, y^{(k+1)}, Z\right) \\ Z \succeq 0 \end{array}, \tag{2.56}$$

$$X^{(k+1)} = X^{(k)} + \sigma\left(A^T\left(y^{(k+1)}\right) + Z^{(k+1)} - C\right), \tag{2.57}$$

in any order. The first-order optimality conditions for (2.55) are

$$\nabla_y \mathcal{L}_\sigma\left(X^{(k)}, y^{(k+1)}, Z^{(k)}\right) = A\left(X^{(k)}\right) - b + \sigma A\left(A^T\left(y^{(k+1)}\right) + Z^{(k)} - C\right) = 0. \tag{2.58}$$

Because $\left(AA^T\right)$ is invertible, from (2.21), we obtain

$$\begin{gathered} y\left(X,Z\right) = -\left(AA^T\right)^{-1}\left(\tfrac{1}{\sigma}\left(A\left(X\right) - b\right) + A\left(Z - C\right)\right) \\ y^{(k+1)} = y\left(X^{(k)}, Z^{(k)}\right) \end{gathered} \tag{2.59}$$

The problem (2.56) can be rewritten as the projection

$$\begin{gathered} \min_{Z \in \mathbb{S}^n} \quad \left\|Z - V^{(k+1)}\right\|_F^2 \\ Z \succeq 0 \end{gathered}, \tag{2.60}$$

with

$$\begin{gathered} V\left(X,Z\right) = C - A^T\left(y\left(X,Z\right)\right) - \tfrac{1}{\sigma}X \\ V^{(k+1)} = V\left(X^{(k)}, Z^{(k)}\right), \end{gathered} \tag{2.61}$$

Letting $V = V_+ + V_-$ be the spectral decomposition of V gives

$$Z^{(k+1)} = V_+^{(k+1)}. \tag{2.62}$$

Then, (2.57) can be solved as

$$X^{(k+1)} = X^{(k)} + \sigma\left(A^T\left(y^{(k+1)}\right) + Z^{(k+1)} - C\right) = \sigma\left(Z^{(k+1)} - V^{(k+1)}\right) = \sigma V_-^{(k+1)}. \tag{2.63}$$

This $X^{(k+1)}$ is also the optimal solution of the projection

$$\begin{gathered} \min_{X \in \mathbb{S}^n} \quad \left\|\tfrac{1}{\sigma}X + V^{(k+1)}\right\|_F^2 \\ X \succeq 0. \end{gathered} \tag{2.64}$$

Then, the alternating direction augmented Lagrangian method involves selecting initial $X^{(0)} \succeq 0, Z^{(0)} \succeq 0$, and then calculating $y^{(k+1)}$ from (2.59), $V^{(k+1)}$ from (2.61), spectrally decomposing it to calculate $Z^{(k+1)}$ from (2.62), and calculating $X^{(k+1)}$ from (2.63), until a stopping condition is met. (Note that this is closely related to the BPM, and is exactly the same as the iterative update of primal and dual variables in its generalization MPRW as implemented.)

They proved that this algorithm converges eventually to an optimal solution for any penalty parameter $\sigma > 0$. To improve the algorithm's numerical performance, they tune σ as the algorithm progresses, to try to achieve and maintain

$$\left\| A\left(X^{(k+1)}\right) - b \right\|_2 \approx \left\| C - A^T\left(y^{(k+1)}\right) - Z^{(k+1)} \right\|_F . \tag{2.65}$$

From (2.63), (2.59), (2.61), and (2.21),

$$\begin{aligned} A\left(X^{(k+1)}\right) - b &= \sigma\left(Z^{(k+1)} - V^{(k+1)}\right) - b \\ = \sigma A\left(Z^{(k+1)} - C\right) + A\left(X^{(k)}\right) - b &+ \left(b - A\left(X^{(k)}\right)\right) + \sigma\left(C - Z^{(k)}\right) \\ &= \sigma A\left(Z^{(k+1)} - Z^{(k)}\right), \end{aligned} \tag{2.66}$$

and (2.63) can be rearranged to give

$$C - A^T\left(y^{(k+1)}\right) - Z^{(k+1)} = \frac{1}{\sigma}\left(X^{(k)} - X^{(k+1)}\right), \tag{2.67}$$

so changes to the primal and dual infeasibilities are proportional to σ and $\frac{1}{\sigma}$, respectively. Then, to keep them of approximately the same size, σ is decreased (increased) by a factor of γ ($\frac{1}{\gamma}$) if primal infeasibility is less than (greater than) a chosen multiple of the dual infeasibility for a chosen number of iterations in a row, with the additional restrictions that $0 < \gamma < 1$ and σ remains within a chosen interval $\left[\sigma_{\min}, \frac{1}{\sigma_{\max}}\right]$, where $0 < \sigma_{\min} < \sigma_{\max} < \infty$. This thesis's implementation of the algorithm checks whether either relative infeasibility is less than half of the other for 50 consecutive iterations, and initializes $\sigma = 0.2$ with $\sigma_{\min} = 10^{-4}$, $\sigma_{\max} = 10^4$, and $\gamma = 0.99$.

The algorithm stops when the relative primal and dual errors are below a chosen tolerance, or it runs out of CPU time or iterations.

Note: the paper uses primal error relative to $1 + \|b\|_2$, dual error relative to $1 + \|C\|_1$, while the code implemented and tested uses DIMACS errors with $1 + \|b\|_\infty$, $1 + \|c\|_\infty$, does not test for and stop with stagnation in dual gap, and instead, if primal and dual relative errors are within tolerance and the primal-dual gap error is not, it shifts the primal solution to become worse, along the direction away from the projection of $\mathrm{vec}_C\left(C\right)$ onto the null space of $\boldsymbol{A}$, until the gap decreases and then returns the result instead.

Algorithm 2.3 Alternating-Direction Augmented Lagrangian Method [19]

```
Precompute sparse Cholesky factorization of $\boldsymbol{AA}^T$.
Choose $X, Z \succeq 0$, $\sigma_{\min}, \sigma_{\max} \in (0, \infty)$,
$\sigma \in [\sigma_{\min}, \sigma_{\max}]$, $\gamma \in (0,1)$, tolerance $\varepsilon > 0$.
repeat until $\delta < \varepsilon$:

    Solve for and update $y$ using Cholesky factorization:
        $AA^T(y) = -\left(\frac{1}{\sigma}(A(X) - b) + A(Z - C)\right)$
    $V \leftarrow C - A^T(y) - \frac{1}{\sigma}X$
    $Z \leftarrow V_+$
    $X \leftarrow -\sigma V_-$
    $\delta_p \leftarrow \frac{\|A(X)-b\|_2}{1+\|b\|_\infty}$, $\delta_d \leftarrow \frac{\left\|C-Z-A^T(y)\right\|_2}{1+\|C\|_\infty}$, $\delta \leftarrow \max\left\{\delta_p, \delta_d\right\}$
    If $\frac{1}{2}\delta_p > \delta_d$ for 50 consecutive iterations:
        $\sigma \leftarrow \max(\sigma_{\min}, \gamma\sigma)$
    If $\delta_p < \frac{1}{2}\delta_d$ for 50 consecutive iterations:
        $\sigma \leftarrow \min\left(\sigma_{\max}, \frac{1}{\gamma}\sigma\right)$
end
If $\frac{\langle C,X\rangle - b^T y}{1+|\langle C,X\rangle|+\left|b^T y\right|} > \varepsilon$:

    move $X$ to improve dual gap, maintain $\frac{\|A(X)-b\|_2}{1+\|b\|_\infty} < \varepsilon$
```

2.1.4 Splitting Conic Solver (SCS)

In 2016, O'Donoghue, Chu, Parikh, and Boyd [12] developed a first order solver, Splitting Conic Solver (SCS), for large conic programming problems including SDPs, which uses the alternating direction method of multipliers to solve a self-dual homogeneous embedding of the problem. SCS uses the vectorization $\mathrm{vec}_S : \mathbb{S}^n \to \mathbb{R}^{\hat{n}}$, $\hat{n} = \frac{n(n+1)}{2}$ of symmetric matrices, defined earlier.

SCS uses the problem format:

$$\begin{array}{lrcl} \min & c^T x & & \\ (P) & \boldsymbol{A}x + s & = & b \\ & s & \in & \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \end{array} , \quad \begin{array}{lrrcl} \max & -b^T y & & & \\ (D) & -\boldsymbol{A}^T y & +r & = & c \\ & y & & \in & \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \\ & r & & \in & \{0\}^m \end{array} \tag{2.68}$$

To convert the SCS variables to the standard variable definitions, substitute $\boldsymbol{A}_{SCS}$=$\boldsymbol{A}^T$, $b_{SCS} = c = \mathrm{vec}_{\mathrm{S}}\left(C\right), c_{SCS} = -b, y_{SCS} = x = \mathrm{vec}_{\mathrm{S}}\left(X\right), x_{SCS} = y, s_{SCS} = z = \mathrm{vec}_{\mathrm{S}}\left(Z\right)$, $(P)_{SCS} = (D)$, and $(D)_{SCS} = (P)$:

$$\begin{array}{lrcl} \max & b^T y & & \\ (D) & \boldsymbol{A}^T y + z & = & c \\ & z & \succeq & 0 \end{array} , \quad \begin{array}{lrrcl} \min & c^T x & & & \\ (P) & \boldsymbol{A}x & -r & = & b \\ & x & & \in & \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \\ & r & & \in & \{0\}^m \end{array} \tag{2.69}$$

Although r is a new variable not in the standard notation, it must be $r = 0 \in \mathbb{R}^m$ for a feasible solution, so it does not change the problem:

$$\begin{array}{lrrcl} \min & c^T x & & & \\ (P) & \boldsymbol{A}x & -r & = & b \\ & x & & \in & \mathrm{vec}\left(\mathbb{S}^n_+\right) \\ & r & & \in & \{0\}^m \end{array} \Leftrightarrow \begin{array}{lrcl} \min & c^T x & & \\ (P) & \boldsymbol{A}x & = & b \\ & x & \in & \mathrm{vec}\left(\mathbb{S}^n_+\right) \end{array} \tag{2.70}$$

If strong duality holds, then by the theorem of strong alternatives [6, p260], exactly one of the pair

$$\mathcal{D} = \left\{ (y, z) : \boldsymbol{A}^T y + z = c, z \in \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \right\} \tag{2.71}$$

$$\mathcal{P} = \left\{ x : \boldsymbol{A}x = 0, x \in \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right), c^T x < 0 \right\} \tag{2.72}$$

is empty, where $\mathcal{D}$ encodes dual feasibility and $\mathcal{P}$ encodes dual unboundedness, so $\exists x \in \mathcal{P} \Rightarrow \mathcal{D} = \emptyset$ means that the dual problem is infeasible and x is a certificate of dual infeasibility. Similarly, also by the theorem of strong alternatives, exactly one of the pair

$$\tilde{\mathcal{D}} = \left\{ y : -\boldsymbol{A}^T y \in \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right), -b^T y < 0 \right\} \tag{2.73}$$

$$\tilde{\mathcal{P}} = \left\{ x : \boldsymbol{A}x = b, x \in \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \right\} \tag{2.74}$$

is empty, so any $y \in \tilde{\mathcal{D}}$ is a certificate of primal infeasibility [12].

Homogeneous Self-Dual embedding: Instead of having a separate primal and dual problem, SCS combines the primal-dual pair into a single feasibility problem, by embedding the KKT conditions into a system of equations and inclusions that solutions to (P) and (D) must jointly satisfy:

$$\begin{bmatrix} r \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{A} \\ -\boldsymbol{A}^T & 0 \\ -b^T & c^T \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} + \begin{bmatrix} -b \\ c \\ 0 \end{bmatrix} \tag{2.75}$$

and $(x, r, y, z) \in \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \times \{0\}^m \times \mathbb{R}^m \times \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right)$. Any (x, r, y, z) satisfying this embedding is primal-dual optimal, because it is equivalent to $\boldsymbol{A}x - b = r = 0 \in \mathbb{R}^m \Leftrightarrow \boldsymbol{A}x = b$, $c - \boldsymbol{A}^T y = z \Leftrightarrow \boldsymbol{A}^T y + z = c$, $-b^T y + c^T x = 0$, but if either (P) or (D) is infeasible, then the embedding has no solution, and there is no valid $(x, r, y, z) \in \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \times \{0\}^m \times \mathbb{R}^m \times \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right)$. However, this embedding does not give a certificate of infeasibility, to *show* that there is no solution.

A modified version of the homogeneous self-dual embedding:

$$\left\{ \begin{array}{c} \begin{bmatrix} r \\ z \\ \kappa \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{A} & -b \\ -\boldsymbol{A}^T & 0 & c \\ b^T & -c^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \\ \tau \end{bmatrix} \\ (y, z, r, x, \tau, \kappa) \in \mathbb{R}^m \times \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \times \{0\}^m \times \mathrm{vec}_{\mathrm{S}}\left(\mathbb{S}^n_+\right) \times \mathbb{R}_+ \times \mathbb{R}_+ \end{array} \right. \tag{2.76}$$

introduces new nonnegative complementary variables, κ and τ, so that at most one of the two is nonzero. $\langle (y, x, \tau), (r, z, \kappa) \rangle = 0$, $y^T r \geq 0$, $x^T z \geq 0$, and $\tau\kappa \geq 0$, by the definition of dual cones. For a solution $(y, z, r, x, \tau, \kappa)$ to this modified embedding, $\tau \neq 0$ is a scaling factor that can be used to recover solutions to the initial primal and dual problems, or $\kappa \neq 0$ can be used to recover a certificate of primal or dual infeasibility, which, if it exists, *proves* that the primal or dual problem is infeasible, respectively. This is because the system (2.76) is equivalent to

$$\begin{cases} \begin{bmatrix} r \\ z \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{A} \\ -\boldsymbol{A}^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} + \begin{bmatrix} -b \\ c \end{bmatrix} \tau \\ \kappa = b^T y - c^T x \end{cases} \tag{2.77}$$

i.e., scaling the added constants by τ and defining κ as the duality gap between the primal and dual objective values. This means that $\kappa > 0$ is a positive duality gap indicating that the current solution is not optimal, and $\tau > 0$ is a scaling factor which can be used to convert a feasible solution to the system into an optimal solution to the original problem.

There are three cases for any solution $(y, z, r, x, \tau, \kappa)$:

If $\tau > 0$ and $\kappa = 0$, then $(\hat{y}, \hat{x}, \hat{z}) = \frac{1}{\tau}(y, x, z)$ satisfies the original KKT conditions and is an optimal primal-dual solution.

If $\tau = 0$ and $\kappa > 0$, then there is a duality gap and the original problem is primal or dual infeasible. If $\tau = 0, \kappa > 0$, and $c^T x < 0$, then $\hat{x} = \frac{x}{c^T x}$ is a certificate of primal infeasibility: $\boldsymbol{A}\hat{x} = 0$, $\hat{x} \in \mathrm{vec}_{\mathrm{S}}(\mathbb{S}^n_+)$, $c^T\hat{x} = -1 \Rightarrow \hat{x} \in \mathcal{P}$. If $\tau = 0, \kappa > 0$, and $-b^T y < 0$, then $\hat{y} = \frac{y}{-b^T y}$ is a certificate of dual infeasibility (a certificate of primal unboundedness, in this case): $-\boldsymbol{A}^T(\hat{y}) \in \mathrm{vec}_{\mathrm{S}}(\mathbb{S}^n_+)$, $-b^T y = -1 \Rightarrow \hat{y} \in \tilde{\mathcal{D}}$. If $\tau = 0$ and $\kappa > 0$ and both $-b^T y < 0$ and $c^T x < 0$, then the problem is both primal and dual infeasible, but the assumption of strong duality has been violated.

If $\tau = \kappa = 0$ and $c^T x < 0$ or $-b^T y < 0$, then it can be used to derive a certificate of primal or dual infeasibility, as in the previous case. Otherwise, there is no information about the original problem; 0 is always a solution to the embedding, and the solving algorithm must take steps to avoid it.

The embedding is homogeneous: if

$$(y, z, r, x, \tau, \kappa) \in \mathbb{R}^m \times \mathrm{vec}_{\mathrm{S}}(\mathbb{S}^n_+) \times \{0\}^m \times \mathrm{vec}_{\mathrm{S}}(\mathbb{S}^n_+) \times \mathbb{R}_+ \times \mathbb{R}_+ \tag{2.78}$$

is a solution to the system, then

$$t(y, z, r, x, \tau, \kappa) = (ty, tz, tr, tx, t\tau, t\kappa)\, \forall t \geq 0 \tag{2.79}$$

is also a solution, yielding the same primal dual solution or infeasibility certificates. To simplify, let

$$u = \begin{bmatrix} y \\ x \\ \tau \end{bmatrix}, v = \begin{bmatrix} r \\ z \\ \kappa \end{bmatrix}, Q = \begin{bmatrix} 0 & \boldsymbol{A} & -b \\ -\boldsymbol{A}^T & 0 & c \\ b^T & -c^T & 0 \end{bmatrix} \tag{2.80}$$

and $C = \mathbb{R}^m \times \text{vec}_\text{S}\left(\mathbb{S}^n_+\right) \times \mathbb{R}_+$, with dual cone $C^* = \{0\}^m \times \text{vec}_\text{S}\left(\mathbb{S}^n_+\right) \times \mathbb{R}_+$. Then, searching for a solution to the modified homogeneous self-dual embedding can be expressed as finding $(u,v), \ni v = Qu,\ (u,v) \in C \times C^*$. $(u,v) = 0$ is always a *valid* solution to this, but it is not useful for the original problem, so we want to find a *nonzero* solution to this system. It can straightforwardly be shown that this problem is self-dual, from the Lagrangian

$$\mathcal{L}(u,v,\nu,\lambda,\mu) = \nu^T (Qu - v) - \lambda^T u - \mu^T v, \tag{2.81}$$

with dual variables ν and $\lambda \in C$, $\mu \in C^*$, using $Q^T = -Q$ and $\nu = -\mu$ to simplify the problem to the original problem, but with (μ, λ) replacing (u,v).

The feasibility method can be solved in multiple ways, but SCS uses the alternating direction method of multipliers (ADMM), a splitting method that solves convex problems of the form

$$\min f(x) + g(z), \text{ s.t.}\, x = z \tag{2.82}$$

with the $(k+1)$th iterative step

$$\begin{aligned} x^{(k+1)} &= \text{argmin}_x \left(f(x) + \tfrac{\rho}{2} \left\| x - z^{(k)} - \lambda^{(k)} \right\|_2^2 \right) \\ z^{(k+1)} &= \text{argmin}_z \left(g(z) + \tfrac{\rho}{2} \left\| x^{(k+1)} - z - \lambda^{(k)} \right\|_2^2 \right) \\ \lambda^{(k+1)} &= \lambda^{(k)} - x^{(k+1)} + z^{(k+1)} \end{aligned} \tag{2.83}$$

where $\rho > 0$ is a step-size parameter and λ is the scaled dual variable associated with constraint $x = z$, with arbitrary initial points $z^{(0)}, \lambda^{(0)}$, usually taken to be 0. This is the *alternating direction* method of multipliers because the augmented Lagrangian is minimized with respect to x and z sequentially, rather than jointly as in the method of multipliers. The residual between x and z, the primal objective value $f\left(x^{(k)}\right) + g\left(z^{(k)}\right)$, and the dual variable $\lambda^{(k)}$ converge, if f and g are closed, proper, and convex, and a saddle point exists [5].

To apply ADMM to the embedding, the simplified self-dual homogeneous embedding can be expressed as

$$\min \left(I_{C\times C^*}\left(u,v\right) + I_{Qu=v}\left(\tilde{u},\tilde{v}\right)\right), \text{ s.t. } \left(u,v\right) = \left(\tilde{u},\tilde{v}\right) \tag{2.84}$$

where

$$I_S\left(x\right) = \begin{cases} 0 & x \in S \\ +\infty & x \notin S \end{cases} \tag{2.85}$$

is the characteristic function for a set S. Then, direct substitution into the ADMM format gives:

$$\begin{gathered} \left(\tilde{u}^{(k+1)}, \tilde{v}^{(k+1)}\right) = \mathcal{P}_{Qu=v}\left(u^{(k)} + \lambda^{(k)}, v^{(k)} + \mu^{(k)}\right) \\ u^{(k+1)} = \mathcal{P}_C\left(\tilde{u}^{(k+1)} - \lambda^{(k)}\right) \\ v^{(k+1)} = \mathcal{P}_{C^*}\left(\tilde{v}^{(k+1)} - \mu^{(k)}\right) \\ \lambda^{(k+1)} = \lambda^{(k)} - \tilde{u}^{(k+1)} + u^{(k+1)} \\ \mu^{(k+1)} = \mu^{(k)} - \tilde{v}^{(k+1)} + v^{(k+1)} \end{gathered} \tag{2.86}$$

where $\mathcal{P}_S\left(x\right)$ denotes Euclidean projection of x onto set S, and λ and μ are the dual variables for the equality constraints on u and v, respectively.

If the variables are initialized as

$$\lambda^{(0)} = v^{(0)}, \; \mu^{(0)} = u^{(0)} \tag{2.87}$$

then

$$\lambda^{(k)} = v^{(k)}, \; \mu^{(k)} = u^{(k)} \tag{2.88}$$

for all subsequent iterations, allowing the simplification of the system and the elimination of the dual variables, using

$$\mathcal{Q} = \left\{(u,v) : Qu = v\right\} = \mathcal{Q}^{\perp} \tag{2.89}$$

because Q is skew-symmetric, so using $z = \mathcal{P}_{\mathcal{Q}}\left(z\right) + \mathcal{P}_{\mathcal{Q}^{\perp}}\left(z\right), \forall z :$

$$\left(\tilde{v}^{(k+1)}, \tilde{u}^{(k+1)}\right) = \mathcal{P}_{\mathcal{Q}^{\perp}}\left(u^{(k)} + v^{(k)}, u^{(k)} + v^{(k)}\right) \tag{2.90}$$

which gives

$$u^{(k)} + v^{(k)} = \tilde{u}^{(k+1)} + \tilde{v}^{(k+1)}. \tag{2.91}$$

The final algorithm becomes:

$$\begin{aligned} \tilde{u}^{(k+1)} &= (I+Q)^{-1}\left(u^{(k)} + v^{(k)}\right) \\ u^{(k+1)} &= \mathcal{P}_C\left(\tilde{u}^{(k+1)} - v^{(k)}\right) \\ v^{(k+1)} &= v^{(k)} - \tilde{u}^{(k+1)} + u^{(k+1)} \end{aligned} \tag{2.92}$$

where $I+Q$ is guaranteed to be invertible, because Q is skew-symmetric. The matrix inversion can be avoided by instead solving for $\tilde{u}^{(k+1)}$ as a linear system of equations:

$$\tilde{u}^{(k+1)} = (I+Q)^{-1}\left(u^{(k)} + v^{(k)}\right) \Leftrightarrow (I+Q)\,\tilde{u}^{(k+1)} = u^{(k)} + v^{(k)} \tag{2.93}$$

$$(I+Q)\,\tilde{u}^{(k+1)} = \begin{bmatrix} I_m & \boldsymbol{A} & -b \\ -\boldsymbol{A}^T & I_n & c \\ b^T & -c^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{u}_y \\ \tilde{u}_x \\ \tilde{u}_\tau \end{bmatrix}^{(k+1)} = \begin{bmatrix} u_y \\ u_x \\ u_\tau \end{bmatrix}^{(k)} + \begin{bmatrix} v_r \\ v_z \\ v_\kappa \end{bmatrix}^{(k)}. \tag{2.94}$$

The optimality conditions are $u^{(k)} \in C$, $v^{(k)} \in C^*$, $u^{(k)T}v^{(k)} = 0$, and $Qu^{(k)} - v^{(k)} = 0$. The first three conditions are true at every iteration, and the final condition holds asymptotically, as $k \to \infty$.

The projection $\mathcal{P}_C(w)$ is the Euclidean projection onto $C = \mathbb{R}^m \times \mathrm{vec}_S\left(\mathbb{S}^n_+\right) \times \mathbb{R}_+$, so

$$\mathcal{P}_C\left(\begin{bmatrix} w_y \\ w_x \\ w_\tau \end{bmatrix}\right) = \begin{bmatrix} \mathcal{P}_{\mathbb{R}^m}\left(w_y\right) \\ \mathcal{P}_{\mathrm{vec}_S\left(\mathbb{S}^n_+\right)}\left(w_x\right) \\ \mathcal{P}_{\mathbb{R}_+}\left(w_\tau\right) \end{bmatrix} = \begin{bmatrix} w_y \\ w_{x_+} \\ w_{\tau_+} \end{bmatrix} \tag{2.95}$$

where $w_x = w_{x_+} + w_{x_-}$ is the vec_S vectorization of spectral decomposition of

$\text{vec}_{\text{S}}{}^{-1}\left(w_x\right)$ and $w_{\tau_+} = \begin{cases} 0, & w_\tau < 0 \\ w_\tau & \text{otherwise} \end{cases}$.

Let $(u^\star, v^\star)$ be a nonzero solution to the system. Because it is a nonzero solution, either $u_\tau^\star > 0$ or $v_\kappa^\star > 0$, so it can be used to derive an optimal solution or a certificate of infeasibility. If an initial point $\left(u^{(0)}, v^{(0)}\right)$ is chosen with $u_\tau^{(0)} = 1$ and $v_\kappa^{(0)} = 1$, with all *other* entries zero, then

$$\left(u^\star, v^\star\right)^T \left(u^{(0)}, v^{(0)}\right) > 0. \tag{2.96}$$

Let $\phi : C \times C^* \to C \times C^*$ denote the mapping from one iteration of the algorithm to the next,

$$\left(u^{(k+1)}, v^{(k+1)}\right) = \phi\left(u^{(k)}, v^{(k)}\right) \tag{2.97}$$

which is nonexpansive

$$\left\|\phi\left(u, v\right) - \phi\left(\hat{u}, \hat{v}\right)\right\|_2 \leq \left\|\left(u, v\right) - \left(\hat{u}, \hat{v}\right)\right\|_2, \forall \left(u, v\right), \left(\hat{u}, \hat{v}\right) \tag{2.98}$$

and $(u^\star, v^\star)$ is a fixed point:

$$\phi\left(u^\star, v^\star\right) = \left(u^\star, v^\star\right). \tag{2.99}$$

Choosing initial $\left(u^{(0)}, v^{(0)}\right)$ such that $u_y^{(0)} = v_r^{(0)} = 0$, $u_x^{(0)} = v_z^{(0)} = 0$, and $u_\tau^{(0)} = v_\kappa^{(0)} = 1$ then forces the algorithm toward nonzero solutions.

To improve convergence, the algorithm adds over-relaxation to the u and v updates above, by replacing all occurrences of $\tilde{u}^{(k+1)}$ with $\alpha\tilde{u}^{(k+1)} + (1 - \alpha)\, u^{(k)}$, with some relaxation parameter $\alpha \in (1, 2)$, usually $\alpha \in [1.5, 1.8]$ [5]. Also, to reduce computational costs, rather than solving the linear system of equations $(I + Q)\, \tilde{u}^{(k+1)} = \left(u^{(k)} + v^{(k)}\right)$ for $\tilde{u}^{(k+1)}$, it uses the approximate projection of finding any $\tilde{u}^{(k+1)}$ that satisfies

$$\left\|(I + Q)\, \tilde{u}^{(k+1)} - \left(u^{(k)} + v^{(k)}\right)\right\|_2 \leq \zeta^{(k)} \tag{2.100}$$

where $\zeta^{(k)} > 0$ and $\Sigma_k \zeta^{(k)} < \infty$, bounding the norm of the residual between $(I + Q)\, \tilde{u}^{(k+1)}$ and $\left(u^{(k)} + v^{(k)}\right)$ rather than forcing it equal to 0.

Solving the linear system of equations $(I+Q)\,\tilde{u} = w$ for $\tilde{u}$ with some w is performed in each iteration:

$$\begin{bmatrix} I_m & \boldsymbol{A} & -b \\ -\boldsymbol{A}^T & I_{\hat{n}} & c \\ b^T & -c^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{u}_y \\ \tilde{u}_x \\ \tilde{u}_\tau \end{bmatrix} = \begin{bmatrix} w_y \\ w_x \\ w_\tau \end{bmatrix}. \tag{2.101}$$

To simplify notation, define

$$M = \begin{bmatrix} I_m & \boldsymbol{A} \\ -\boldsymbol{A}^T & I_{\hat{n}} \end{bmatrix}, h = \begin{bmatrix} -b \\ c \end{bmatrix}, \tag{2.102}$$

so

$$I + Q = \begin{bmatrix} M & h \\ -h^T & 1 \end{bmatrix}. \tag{2.103}$$

Because $M + hh^T$ is the Schur complement of the lower right block 1 in $I + Q$,

$$\begin{bmatrix} \tilde{u}_y \\ \tilde{u}_x \end{bmatrix} = \left(M + hh^T\right)^{-1} \left(\begin{bmatrix} w_y \\ w_x \end{bmatrix} - w_\tau h \right) \tag{2.104}$$

so applying the Sherman-Morrison-Woodbury formula [6, p650, p678] to $\left(M + hh^T\right)^{-1}$ gives

$$\begin{aligned} \left(M + hh^T\right)^{-1} &= M^{-1} - M^{-1}h\left(I_1 + h^T M^{-1} h\right)^{-1} h^T M^{-1} \\ &= M^{-1} - \tfrac{M^{-1}hh^T M^{-1}}{1 + h^T M^{-1} h}. \end{aligned} \tag{2.105}$$

Substituting this into the previous equation, and using $h^T M^{-1} = -\left(M^{-1}h\right)^T$

because M is skew-symmetric, gives:

$$\begin{aligned}\begin{bmatrix} \tilde{u}_y \\ \tilde{u}_x \end{bmatrix} &= \left(M^{-1} - \tfrac{M^{-1}hh^TM^{-1}}{1+h^TM^{-1}h}\right)\left(\begin{bmatrix} w_y \\ w_x \end{bmatrix} - w_\tau h\right) \\ &= M^{-1}\begin{bmatrix} w_y \\ w_x \end{bmatrix} - w_\tau\left(M^{-1}h\right) + \tfrac{\left(M^{-1}h\right)\left(M^{-1}h\right)^T}{1+h^T\left(M^{-1}h\right)}\left(\begin{bmatrix} w_y \\ w_x \end{bmatrix} - w_\tau h\right)\end{aligned} \tag{2.106}$$

$$\tilde{u}_\tau = w_\tau - b^T\tilde{u}_y + c^T\tilde{u}_x \tag{2.107}$$

Then, in the first iteration, $M^{-1}h$ is precomputed, and the bulk of subsequent iterations' effort is in computing

$$M^{-1}\begin{bmatrix} w_y \\ w_x \end{bmatrix} \tag{2.108}$$

and simple vector operations with precomputed quantities. Solving linear equations of the form

$$\begin{bmatrix} I_m & -\boldsymbol{A} \\ -\boldsymbol{A}^T & -I_{\hat{n}} \end{bmatrix}\begin{bmatrix} z_y \\ -z_x \end{bmatrix} = \begin{bmatrix} w_y \\ w_x \end{bmatrix} \tag{2.109}$$

for z is performed every iteration, either by solving a sparse permuted LDL^T factorization or indirectly with an iterative conjugate gradient method. The indirect conjugate gradient method approximately iteratively solves the system by rewriting it as

$$z_x = \left(I_m + \boldsymbol{A}\boldsymbol{A}^T\right)^{-1}\left(w_y - \boldsymbol{A}w_x\right),\, z_x = w_x + \boldsymbol{A}^T z_y \tag{2.110}$$

by elimination, which is then solved with the conjugate gradient method, which iteratively approximately solves for z by parallelly multiplying by $\boldsymbol{A}$ and $\boldsymbol{A}^T$in each iteration, avoiding actually constructing the $\boldsymbol{A}\boldsymbol{A}^T$ matrix, and halting when the residual satisfies (2.100) for some appropriate sequence $\zeta^{(k)}$.

However, the indirect conjugate gradient method is still generally slower than directly solving the system, and so was not used. Instead, the direct method, solving the system with a sparse permuted LDL^T factorization was used. This involves factoring

$$\begin{bmatrix} I_m & -\boldsymbol{A} \\ -\boldsymbol{A}^T & -I_{\hat{n}} \end{bmatrix} = PLDL^TP \tag{2.111}$$

where P is a permutation matrix, L is a lower triangular matrix with positive diagonal elements, and D is a block diagonal with nonsingular 1×1 and 2×2 blocks. The factorization takes $\frac{1}{3}(m + \hat{n})^3$ floating point operations *without* exploiting the structure, but there are methods for improving the factorization efficiency for sparse $\boldsymbol{A}$, using the sparsity pattern and nonzero values. The factorization only needs to be computed once, and then each iteration takes $O\left((m + \hat{n})^2\right)$ flops to solve for z by permutation, scaling, and forward and backward substitution. This allows each iteration's $\boldsymbol{A}\boldsymbol{A}^T$ linear system of equations to be solved in at most $O\left((m + \hat{n})^2\right)$ time [6, p672].

SCS uses ADMM, described above. A variant program written by the same group, SuperSCS, instead uses an algorithm called SuperMann to solve a different Douglas-Rachford Splitting of the problem [16]. SuperSCS generally performs worse than SCS on the limited set of problems tested in this thesis, and is not used here.

Algorithm 2.4 SCS [12]

```
    Set $k = 0$, $x^{(0)} = z^{(0)} = 0 \in \mathrm{vec}_S(\mathbb{S}^n_+)$,
    $y^{(0)} = r^{(0)} = 0 \in \mathbb{R}^m$,
    $\tau^{(0)} = \kappa^{(0)} = 1$.
    Choose tolerance $\varepsilon \geq 0$.
    Precompute sparse permuted $LDL^T$ factorization of $M$.
    Precompute $M^{-1} \begin{bmatrix} -b \\ c \end{bmatrix}$ using $LDL^T$ factorization.
    repeat until $\tau^{(k)} \leq \varepsilon$, $\delta \leq \varepsilon$:

        Solve for $\begin{bmatrix} \tilde{y} \\ \tilde{x} \\ \tilde{\tau} \end{bmatrix}$ using $M^{-1} \begin{bmatrix} -b \\ c \end{bmatrix}$ and $LDL^T$ factorization:
            $\begin{bmatrix} I_m & \boldsymbol{A} & -b \\ -\boldsymbol{A}^T & I_{\hat{n}} & c \\ b^T & -c^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{y} \\ \tilde{x} \\ \tilde{\tau} \end{bmatrix} = \begin{bmatrix} r \\ z \\ \kappa \end{bmatrix}^{(k)}$
        $\begin{bmatrix} y \\ x \\ \tau \end{bmatrix}^{(k+1)} = \begin{bmatrix} \tilde{y} \\ \tilde{x}_+ \\ \max(\tilde{\tau}, 0) \end{bmatrix}$
        $\begin{bmatrix} r \\ z \\ \kappa \end{bmatrix}^{(k+1)} = \begin{bmatrix} r \\ z \\ \kappa \end{bmatrix}^{(k)} - \begin{bmatrix} 0 \\ \tilde{x}_- \\ \min(\tilde{\tau}, 0) \end{bmatrix}$
        $\delta = \max\left( \frac{\left\|\boldsymbol{A}^T y^{(k)} + z^{(k)} - c\right\|_2}{1+\|c\|_2}, \frac{\left\|\boldsymbol{A} x^{(k)} - b\right\|_2}{1+\|b\|_2}, \frac{c^T x^{(k)} - b^T y^{(k)}}{1+\left|c^T x^{(k)}\right| + \left|b^T y^{(k)}\right|} \right)$
        $k = k + 1$

    end
```

CHAPTER 3

RESULTS

3.1 DIMACS Error Measures

This thesis uses six relative error measures for SDP solutions, five of which were defined by The Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) for its Seventh Implementation Challenge on Semidefinite and Related Optimization Problems [13], whose proceedings are archived online at [1], with names of the errors standardized and a sixth measure added in Mittelmann's benchmarking work [10]. These error measures describe how infeasible a solution is relative to its constraints and how suboptimal a solution is relative to its objective values. It is important to investigate all error measures, because solvers might sacrifice one measure of feasibility or optimality to improve performance in others.

The first and second error measures concern primal infeasibility. The first error measure is primal infeasibility for the affine equality constraint $A\left(X\right) = b$, with respect to the greatest absolute element of the vector b that defines its boundary:

$$\mathrm{err}_1 = \frac{\left\|A\left(X\right) - b\right\|_2}{1 + \left\|b\right\|_\infty}. \tag{3.1}$$

The second error measure is primal infeasibility for the conic inequality constraint $X \succeq 0$, i.e. $\lambda_{min}\left(X\right) \geq 0$, where λ_{min} denotes the least eigenvalue of a matrix. (Note: if X is broken into linear component X^ℓ and multiple diagonal blocks X^s, this is the minimum of the minimum element of X^ℓ and the minimum eigenvalues of each positive semidefinite sub-matrix of X^s). Then, the second error measure is the magnitude of the most negative eigenvalue of X with respect to the size of the greatest absolute element of b:

$$\mathrm{err}_2 = \max\left\{0, \frac{-\lambda_{min}\left(X\right)}{1 + \left\|b\right\|_\infty}\right\}. \tag{3.2}$$

The third and fourth error measures concern dual infeasibility. The third error measure is dual infeasibility with respect for the affine equality constraint $A^T(y) + Z = C$, with respect to the greatest absolute element of the matrix C that defines its boundary

$$\mathrm{err}_3 = \frac{\left\|A^T(y) + Z - C\right\|_2}{1 + \|c\|_\infty}. \tag{3.3}$$

The fourth error measure is dual infeasibility with respect to the conic inequality constraint $Z \succeq 0$, i.e. $\lambda_{min}(Z) \geq 0$, with respect to the size of the greatest absolute element of C:

$$\mathrm{err}_4 = \max\left\{0, \frac{-\lambda_{min}(Z)}{1 + \|c\|_\infty}\right\}. \tag{3.4}$$

The fifth and sixth error measures concern optimality. If strong duality holds, then for a feasible and optimal solution, the duality gap should be 0 and the complementary slackness condition should hold. A negative duality gap generally indicates that there has been some infeasibility in the solution. It can be taken as zero if negative, with the infeasibility instead described by earlier error measures, or left negative to communicate this. The fifth error measure is the duality gap between primal and dual objective values with respect to their magnitudes:

$$\mathrm{err}_5 = \frac{\langle C, X\rangle - b^T y}{1 + |\langle C, X\rangle| + |b^T y|} \tag{3.5}$$

The sixth error measure, defined only if the second and fourth errors are 0, i.e. X and Z are positive semidefinite, is the distance from the complementary slackness optimality condition with respect to the magnitudes of the objective values:

$$\mathrm{err}_6 = \frac{\langle X, Z\rangle}{1 + |\langle C, X\rangle| + |b^T y|}. \tag{3.6}$$

For feasible solutions, $\mathrm{err}_5 = \mathrm{err}_6$, because feasibility conditions give $\langle X, Z\rangle = \langle C, X\rangle - b^T y$. This can be shown using the feasibility conditions $A(X) = b$ and

$A^T(y) + Z = C:$

$$\begin{aligned} \langle X, Z\rangle &= \langle X, Z + C - C\rangle \\ &= \langle X, C\rangle + \langle X, Z - C\rangle \\ &= \langle C, X\rangle + \left\langle X, -A^T(y)\right\rangle \\ &= \langle C, X\rangle - X^T A^T(y) \\ &= \langle C, X\rangle - \langle A(X), y\rangle \\ &= \langle C, X\rangle - b^T y. \end{aligned} \tag{3.7}$$

Error	Equation
err$_1$	$\frac{\Vert A(X)-b\Vert_2}{1+\Vert b\Vert_\infty}$
err$_2$	$\max\left\{0, \frac{-\lambda_{min}(X)}{1+\Vert b\Vert_\infty}\right\}$
err$_3$	$\frac{\left\Vert A^T(y)+Z-C\right\Vert_2}{1+\Vert C\Vert_\infty}$
err$_4$	$\max\left\{0, \frac{-\lambda_{min}(Z)}{1+\Vert C\Vert_\infty}\right\}$
err$_5$	$\frac{\langle C,X\rangle-b^Ty}{1+\lvert\langle C,X\rangle\rvert+\left\lvert b^Ty\right\rvert}$
err$_6$	$\frac{\langle X,Z\rangle}{1+\lvert\langle C,X\rangle\rvert+\left\lvert b^Ty\right\rvert}$

Table 3.1: DIMACS Error Measures

3.2 Benchmarking

No known solver works well on all valid problems. The performance of solvers can be compared either to a standard or to whichever solver in a set performs best, depending on what is available, in a process known as benchmarking. The accuracy, speed, and robustness of solvers can be compared. Usually, there is a known acceptable error for a solution, which separates usable solutions from unusable solutions, so error measures of a solver on a given problem define its accuracy. Some solvers do not run at all on some problems, for example if the problem or operations are too large to fit in memory, or the solver crashes. It is also common to treat an unacceptably inaccurate solution as a failure. A solver's robustness is its ability to solve a variety of problems. Of solvers that work acceptably well on a problem, it is preferable to use a solver that works *quickly*, so speed is measured as the number of CPU seconds a solver takes to run on a problem. This will generally roughly reflect the computational complexity of the solver's algorithm on the problem. Most programs parallelize some independent steps of matrix and vector multiplication and addition, and the objective time taken might vary significantly depending on available computing resources, so CPU seconds spent by a solver on a problem is a better measure of the amount of computational work it performs than objective time.

3.2.1 Benchmarking through Shifted Geometric Means

Mittelmann's work in benchmarking SDP solvers stored the CPU runtime for each program on each problem in a library of test problems, and marked whether it crashed or was unacceptably or significantly inaccurate on any of the six error DIMACS error measures, defined above.

For comparison purposes, it is helpful to have a number for each solver describing its overall performance. To avoid rewarding quick but unacceptably inaccurate solutions, failures, or crashes, these are treated as taking a large amount of time.

The geometric mean is the *n*th root of the product of the *n* values:

$$\left(\prod_{i=1}^{n} v_i\right)^{1/n} = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln\left(v_i\right)\right). \tag{3.8}$$

This is preferable to the arithmetic mean

$$\frac{1}{n}\sum_{i=1}^{n} v_i \tag{3.9}$$

for positive values which are likely to be significantly different in scale, because it is less sensitive to large outliers. However, this is still susceptible to excessive influence by *small* outliers, and unfairly rewards the addition of particularly quickly solved problems. To reduce the effect of small outliers, a shift $s \geq 0$ is added to each of the values, as $(v_i + s)$, and subtracted from the final result. To prevent the mean from being *decreased* by a small new value, beyond the effect of increasing the number of values, each term is replaced with the maximum of the shifted value and 1, $\max\left(1, v_i + s\right)$, so including a particularly small term is not equivalent to multiplying by a small fraction. Then, the shifted geometric mean used is

$$\left(\prod_{i=1}^{n} \max\left(1, v_i + s\right)\right)^{1/n} - s = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln\left(\max\left(1, v_i + s\right)\right)\right) - s. \tag{3.10}$$

Mittelmann's work generally uses the shift $s = 10$ CPU seconds in his benchmarking work [10], which is also used in this thesis.

3.2.2 Performance Profiles

Another way to compare the performance of solvers on a set of problems is performance profiles. If each solver reports a statistic for each problem which is better when smaller, such as a sum of error measures or runtimes, set to $+\infty$ upon failure, then the performance ratio for each solver on a problem is its reported statistic divided by the best reported statistic on that problem, with problems *no* solvers can solve not included. Then, the performance profile of a solver is the graph on $f \geq 1$ of the fraction of total problems for which its performance ratio is $\leq f$. It is common to overlay the performance profiles of all solvers tested in a single plot, to compare them. However, although it usually makes clear which solver solves the *most* problems for a desired range of f, the apparent ranking of the worse solvers is misleading, and showing the best of the remaining solvers requires recomputing the performance profiles while excluding the best solver [8].

3.3 The Test Problems

The solvers were tested on two sets of benchmarking test problems, SDPLIB 1.2 library of SDP problems [3] and Mittelmann's collection of sparse and other SDP problems [11]. The SDPLIB set includes problems from truss topology design, control and system theory, graph equipartitioning, control system engineering, pregenerated random graph max cuts, graph max cuts, quadratic assignment problems, SDP relaxations of box constrained quadratic programming problems from pregenerated random graphs, Lovasz numbers, Lovasz theta numbers from pregenerated random graphs, and truss topology design problems. The infeasible problems were not included in the tests.

Mittelmann's library of problems are taken from maximal and maximum stable set problems, global polynomial minimization, copositivity determination for high-dimensional Hilbert matrices, SDP relaxations minibisection and maxcut problems for Gset graphs, efficient graph cuts for image segmentation, sum of squares bounding and relaxation problems, generalized problem of moments on polynomial systems, Rosenbrock functions, sensor locations, sparse example problems from various lectures, constrained polynomial problems, and other sources, including the DIMACS Seventh Implementation Challenge. Of Mittelmann's problems, diamond_patch, G60_mb, theta102, and theta123 were not included, nor were the prob* problems, which were too large to read as files rather than generate with separate MATLAB programs. Generally in both libraries, problems with the same name followed by a larger number are structurally similar but larger or otherwise more difficult.

3.4 Results

The solvers were run on all problems, and were killed or halted after an hour. (All code except SCS was possible to modify to accept a maximum CPU time rather than a maximum number of iterations, but SCS could only be killed and could not save its status on exit.)

Of the SDPLIB problems, all solvers failed to solve control3 through control11 of the eleven control* family control theory problems, maxG32, maxG55, and maxG60 of the five maxG* family max cut problems, both qpG11 and qpG51 of the qp* family problems from relaxations of quadratic programming, and thetaG51, the larger of the two thetaG* family theta number problems. Of the Mittelmann Library problems, all solvers failed to solve 1zc.1024, the last of the four 1*1024 family stable set problems, all of buck3 through buck5 of the buck* family problems all six G*mb and G*mc family problems from minbisection and maxcut problems, inc_1200, the larger of the two inc* family problems, mater:4 through mater:6, the largest three of the four mater* family problems, both neosfbr* family problems, neu2g, a middling-size problem from the seven neu* family problems, sensor_1000, the larger of the two sensor* family problems, all shmup3 through shmup5 of the three shmup* family problems, trto5, the largest of the three trto* family problems, and all of vibra3 through vibra5 of the vibra* family problems. Additionally, there were seven problems in the Mittelmann Library not from families of related problems that no solvers could solve in time: biggs, checker_1.5, foot, hand, ice_2.0, p_auss2_3.0, rendl1_2000_1e:6, swissroll, and tiger_texture.

Additionally, BPM was the only solver to solve control1 or control2, its generalization, MPRW, was the only solver to solve 1dc.1024, 1et.1024, or 1tc.1024, SDPAD was the only solver to solve inc_600 or neu2c, and SCS was the only solver to solve arch8, gpp500:1, gpp500:2, ss30, thetaG11, trto3, or trto4. MPRW crashed due to memory issues on the Mittelmann Library problem neosfbr25, and SDPAD crashed due to memory issues on neosfbr30e8, which were two particularly large problems whose AA^T systems were very dense and whose structures were respectively a single 25×25 block with 14376 constraint matrices and a single 842×842 block with 25201 constraint matrices, which none of the solvers were able to solve.

Over all, BPM was the least robust with 49 failures, followed by SCS with 29 failures, MPRW with 27 failures, and SDPAD, the most robust, with 19 failures, of the 112 problems that were solved by any solver at all, not counting the 49 problems that were not. Of the problems it was able to solve, SCS was often the fastest or not much slower than the fastest, and SDPAD overall outperformed BPM and MPRW. SCS might have performed better and more robustly if there were a way to modify it to accept a maximum time, or to cause it to use the same error measures as stopping criteria. The robustness of the solvers by total number of problems solved with the requested accuracy, solved with accuracy issues, or failed, is shown in Table 3.2.

Solver	Solved within 10^{-3}	Solved Between 10^{-3} and 10^{-2}	Failed
BPM	8	55	98
MPRW	62	23	76
SDPAD	75	18	68
SCS	69	14	78

Table 3.2: Robustness of solvers in number of problems solved

The different solvers had different failure modes, for giving inaccurate solutions or failing to solve problems. BPM was most likely to reach an error within tolerance on its own error measures and terminate before reaching the requested accuracy of 10^{-3} on DIMACS errors, although in these cases it often still stopped with a worst error better than 10^{-2}. MPRW generally solved to higher accuracy than BPM, and was more robust to failure. MPRW generally either solved a problem acceptably accurately, solved it with low accuracy by the time it ran out of time, or did not solve in time, although it incorrectly identified six problems as solved and returned solutions with unacceptably large primal or dual infeasibility. On some problems, it did return fairly quickly with low accuracy, usually on problems where it was not able to normalize with respect to b or C (on truss2, truss6, and truss7 of SDPLIB), or where it reached a solution such that the duality gap, which it does not use as a stopping condition, was larger than tolerance even though the relative infeasibilities were not (on cnhil8 and cnhil10 of Mittelmann's Library). SDPAD was the most robust to failure but was the *fastest* solver least often; it never exited before it was out of time unless it had a solution within tolerance. It generally solved acceptably or with low accuracy most problems reasonably quickly, and did not ever exit early with a low accuracy solution, which was helped by its checking whether a solution with low infeasibility had a large dual gap and adjusting to reduce it. SCS was the fastest solver most often by a large margin, and often solved with the requested accuracy and exited quickly, or exited early with a worst error between 10^{-2} and 10^{-3}. It generally either solved problems particularly quickly, exited early with low accuracy due to achieving errors within tolerance on its own error measures, or was killed for running out of time. However, SCS reported that the Mittelmann library problem Bex2_1_5 was infeasible, which is incorrect; this was treated as a failure, but no other solvers incorrectly reported problems as infeasible, because no other solvers were designed to detect infeasibility to begin with, and it is a concerning error to make. On many of the problems where it was killed for running out of time, it still had very large relative errors by its own measures, on the order of 10^{1} or larger, but was generally still improving rather than oscillating between decreasing each of two sets of error measures.

A performance profile of the solvers, shown in Figure 3.1, was calculated for the problems that any solvers could solve. The problems no solvers could solve were included in the calculation of shifted geometric mean but not included in the calculation of the performance profile. The performance profile is calculated for each solver relative to the best only for problems where *any* solver was able to solve it, because relative performance between failures is meaningless in this context, while the shifted geometric mean of time for each solver is *not* calculated relative to the other solvers, and accounts for failures differently, with a penalty of time. SCS was the fastest solver most often, by a large margin, while BPM and MPRW were fastest an approximately equal but smaller number of times, and SDPAD was fastest least often. However, eventually SDPAD overtook SCS on the performance profile, by being more robust and able to solve the most problems despite being slower.

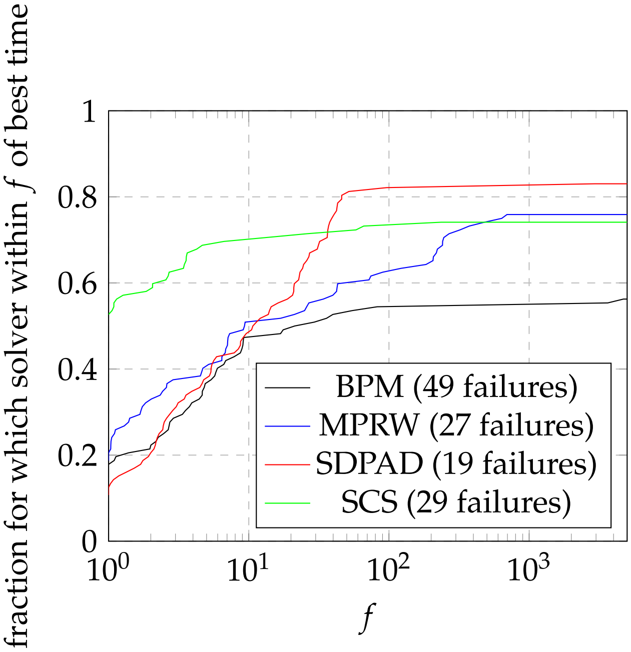

Figure 3.1: A performance profile, showing the fraction of the 113 problems that any solver was able to solve for which each solver's time was within a given fraction of the best solver's time. The boundary point method BPM in green is least robust, with 49 failures, and generally does not perform quickly, while its generalization MPRW in blue, with 27 failures, is somewhat faster and more robust. The alternating direction augmented Lagrangian method SDPAD in red is the most robust, with 19 failures, and generally solves within about 46 times the time of the fastest solver about 80% of the time, surpassing the less robust splitting conic solver SCS in green, with 29 failures, at about 36 times longer than the best time about 71% of the time, where SCS is most often faster before then.

The shifted geometric means, were calculated with the formula (3.10) using a shift of 10 s, cut off at 1 hour, and treated as taking 1 hour for failures including having any error worse than 10^{-2}, and are shown in Table 3.3.

Even despite its non-robustness, SCS performed best in shifted geometric mean of time overall, likely because this measure penalizes taking the entire time equivalently to failures, and SCS took the shortest time on the most problems. The other three solvers performed about equivalently with each other overall. On the SDPLIB problems, SCS performed the best, followed by BPM and SDPAD about equivalently, followed by MPRW. On the Mittelmann Library problems, MPRW and SCS performed best about equivalently, followed by SDPAD, followed by BPM.

Solver	Overall	SDPLIB	Mittelmann Library
BPM	1001.27	468.81	2480.51
MPRW	998.59	710.36	1503.26
SDPAD	926.59	559.97	2028.13
SCS	513.23	189.01	1667.92

Table 3.3: Shifted Geometric Mean (CPU s)

CHAPTER 4

DISCUSSION AND CONCLUSIONS

None of the four solvers generally produced high accuracy solutions within an hour of computations. Even with 10^{-3} tolerance, the solvers were only able to solve between 8 and 75 of the 161 problems within the requested tolerance, and were only able to solve within better tolerance than 10^{-2} for between 63 and 93 of the problems. More problems might have been solved with requested or acceptable accuracy if the solvers had been given more CPU time in which to solve, because running out of time was the most common failure, and most solutions were generally continuing to improve in accuracy, rather than stagnating or oscillating by alternately improving and worsening each of two error measures, at the time the solver was killed or exited due to running out of time. This indicates that future work in SDP solver development should focus on producing more robust and faster solvers. Additionally, BPM and SCS might have performed better on the problems for which they exited with remaining time with an inaccurate solution if they had used the same error measures as stopping criteria.

BPM, MPRW, and SDPAD were fundamentally performing the same operations and calculations in every iteration, with the difference that BPM performed several updates to the dual solution before updating the primal solution in each iteration and used different stopping criteria, whereas MPRW and SDPAD updated all variables once per iteration but used different dynamic adjustments of the scaling parameter and handled the duality gap differently. All three of these solvers solve a symmetric positive definite system of equations using a Cholesky factorization to update the variables. SCS on the other hand solves a different and larger indefinite system of equations to update the variables in each iteration, using an LDL^T factorization. However, the system can be algebraically reduced to a smaller positive definite system in one variable which can then be solved with a Cholesky factorization, from which the other variables can be calculated. In general, such a change could be expected to improve performance, and the SCS paper does not discuss any reason to expect it to be ill-conditioned or otherwise discuss the possibility, so future work might investigate whether an otherwise-identical implementation of SCS which instead used a Cholesky factorization to solve a reduced positive definite system would perform faster. Another significant difference between SCS and the other solvers is that all of the computations of SCS are performed by calling code in MEX files, MATLAB executable files which run compiled C code, while the other programs performed some computations in calls to MEX files, generally the spectral decompositions, but performed other operations directly in MATLAB, which is generally much slower than optimized C code. Future work in comparing the underlying algorithms might investigate whether the same operations of the first three algorithms would perform more competitively as compiled code.

APPENDIX A

DETAILED TABLES OF RESULTS

A.1 BPM Results

BPM had an overall shifted geometric mean of 1001.27 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

A.1.1 BPM Results on SDPLIB

On the SDPLIB problems, BPM had a shifted geometric mean of 468.81 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
arch0	Solved	Acc.	771.59	5.30E-03	5.87E-17	2.62E-03	6.40E-13	2.95E-03	-6.54E-13
arch2	Solved	Acc.	350.44	4.37E-03	3.49E-17	2.95E-03	1.38E-12	-3.70E-03	-1.50E-12
arch4	Solved	Acc.	2523.02	1.12E-03	7.80E-17	2.95E-03	2.17E-14	-5.57E-03	1.17E-12
arch8	Time	Fail	3600.07	2.96E-01	2.40E-16	1.30E-01	1.76E-12	3.99E-01	3.99E-11
control10	Time	Fail	3600.09	2.34E-01	2.24E-16	8.33E+00	5.73E-11	1.21E-01	-3.72E-11
control11	Time	Fail	3600.12	3.35E+00	1.59E-16	2.78E+01	5.31E-11	4.73E-01	-3.77E-12
control1	Solved	Solved	3584.89	1.00E-03	1.35E-15	1.67E-04	1.31E-10	7.77E-04	-1.27E-10
control2	Solved	Acc.	2698.95	6.03E-04	3.06E-16	2.08E-03	3.10E-11	-3.78E-04	5.97E-12
control3	Time	Fail	3600.06	3.03E-02	1.48E-16	4.19E-01	1.62E-10	-2.44E-01	-8.02E-11

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
control4	Time	Fail	3600.07	5.22E-01	2.52E-15	3.45E-01	1.72E-09	-5.80E-02	-1.09E-09
control5	Time	Fail	3600.06	8.54E-02	9.37E-17	6.75E-01	1.22E-10	-3.22E-01	3.37E-11
control6	Time	Fail	3600.08	1.54E+00	5.50E-16	7.01E-01	3.10E-11	-5.00E-01	1.21E-10
control7	Time	Fail	3600.04	9.85E-02	2.04E-16	5.46E-01	9.49E-11	-2.84E-01	-8.83E-12
control8	Time	Fail	3600.06	4.19E-02	1.82E-16	1.44E+00	3.02E-11	-1.74E-01	-2.10E-12
control9	Time	Fail	3600.04	5.99E-02	9.65E-17	7.93E-01	1.31E-10	-2.15E-01	4.39E-13
equalG11	Solved	Acc.	530.32	8.35E-03	3.17E-15	9.91E-04	2.72E-15	-2.24E-02	5.75E-15
equalG51	Solved	Acc.	560.66	2.79E-03	3.66E-15	1.01E-03	4.08E-16	-6.52E-03	1.48E-16
gpp100	Solved	Acc.	65.39	5.50E-03	1.94E-15	6.85E-04	5.75E-14	-2.78E-04	3.50E-14
gpp124:1	Solved	Acc.	162.74	6.07E-03	2.85E-15	9.46E-04	7.25E-14	-3.59E-03	1.66E-12
gpp124:2	Solved	Acc.	99.14	6.07E-03	3.83E-15	1.16E-03	8.16E-14	-1.12E-03	1.18E-13
gpp124:3	Solved	Acc.	54.59	6.07E-03	3.59E-15	9.82E-04	4.72E-13	-3.52E-04	3.19E-13
gpp124:4	Solved	Acc.	92.51	6.07E-03	2.29E-15	5.16E-03	4.76E-13	-1.54E-03	-1.12E-14
gpp250:1	Solved	Acc.	952.98	8.41E-03	7.13E-15	8.48E-04	1.45E-12	-2.30E-03	-4.27E-12
gpp250:2	Solved	Acc.	776.85	8.41E-03	4.16E-15	1.11E-03	6.07E-13	-1.00E-03	-2.41E-12
gpp250:3	Solved	Acc.	753.55	8.41E-03	3.51E-15	1.36E-03	3.15E-13	-5.38E-04	-2.31E-13

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
gpp250:4	Solved	Acc.	768.78	8.41E-03	4.45E-15	2.03E-03	2.51E-13	-4.35E-04	1.22E-13
gpp500:1	Time	Fail	3600.07	3.79E-02	6.64E-15	3.71E-03	5.47E-13	-1.75E-02	-1.11E-12
gpp500:2	Solved	Fail	2658.48	1.17E-02	9.69E-15	3.38E-03	3.50E-12	-2.80E-03	1.79E-12
gpp500:3	Solved	Fail	2045.5	1.17E-02	5.70E-15	2.33E-03	9.30E-13	-8.54E-04	1.06E-12
gpp500:4	Solved	Fail	2268.06	1.17E-02	5.29E-15	1.62E-03	1.36E-13	-3.56E-04	-1.37E-15
hinf10	Solved	Solved	6.38	1.00E-03	4.72E-15	1.90E-04	1.85E-14	-4.03E-02	-1.46E-13
hinf11	Solved	Solved	3.76	1.00E-03	2.81E-15	2.01E-04	6.45E-15	-4.42E-02	-1.15E-13
hinf12	Solved	Solved	12.01	1.00E-03	1.10E-14	1.91E-04	1.42E-14	-1.71E-01	9.27E-13
hinf13	Solved	Acc.	1392.36	1.79E-04	8.44E-14	2.00E-03	9.74E-15	-2.44E-01	9.73E-13
hinf14	Solved	Acc.	39.98	2.06E-04	1.02E-14	2.52E-03	3.22E-14	-8.22E-02	3.01E-12
hinf15	Solved	Acc.	382.31	3.58E-04	2.19E-14	2.39E-03	2.04E-13	-1.44E-01	-3.95E-12
hinf1	Solved	Solved	0.84	1.00E-03	3.95E-17	1.86E-04	3.70E-15	-5.66E-03	-5.06E-15
hinf2	Solved	Acc.	10.56	9.05E-04	7.17E-15	1.81E-03	4.08E-14	-5.45E-02	-8.96E-13
hinf3	Solved	Acc.	45.55	3.43E-04	3.70E-14	1.57E-03	9.41E-15	-7.52E-02	-1.63E-13
hinf4	Solved	Solved	3.27	1.00E-03	2.94E-16	1.72E-04	4.93E-14	-2.25E-04	7.29E-14
hinf5	Solved	Solved	303.06	1.00E-03	1.75E-14	9.31E-04	5.20E-13	-1.61E-02	-7.36E-12

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
hinf6	Solved	Acc.	315.48	5.93E-04	4.22E-14	1.67E-03	1.27E-14	-1.12E-01	8.59E-12
hinf7	Solved	Acc.	10.54	1.01E-04	2.55E-15	1.85E-03	2.08E-15	-7.15E-03	5.55E-14
hinf8	Solved	Acc.	130.45	4.12E-04	4.78E-14	1.58E-03	9.56E-16	-8.35E-02	2.26E-11
hinf9	Solved	Acc.	0.41	9.83E-04	4.75E-18	1.70E-03	5.40E-16	-1.15E-02	1.30E-15
maxG11	Solved	Fail	289.86	1.79E-03	9.88E-15	1.05E-02	6.87E-15	-9.81E-04	-7.36E-16
maxG32	Time	Fail	3601.93	3.85E-03	1.40E-14	1.66E-02	1.40E-14	-1.19E-03	-2.07E-16
maxG51	Solved	Acc.	508.54	1.96E-03	1.09E-14	3.51E-03	2.49E-15	-1.91E-03	-8.43E-16
maxG55	Time	Fail	3614.77	0.00E+00	0.00E+00	8.66E+00	0.00E+00	-2.85E-04	0.00E+00
maxG60	Time	Fail	3687.43	0.00E+00	0.00E+00	1.03E+01	0.00E+00	-3.50E-05	0.00E+00
mcp100	Solved	Acc.	5.83	5.17E-04	2.15E-15	4.16E-03	2.89E-15	-6.61E-04	6.71E-16
mcp124:1	Solved	Acc.	8.83	2.32E-03	2.32E-15	3.54E-03	1.17E-15	-7.98E-04	7.90E-16
mcp124:2	Solved	Acc.	8.47	1.37E-03	4.72E-15	4.25E-03	2.72E-15	-1.05E-03	1.14E-16
mcp124:3	Solved	Acc.	7.77	1.55E-03	2.04E-15	5.28E-03	1.40E-15	-6.85E-04	2.34E-15
mcp124:4	Solved	Acc.	8.49	1.73E-03	4.66E-15	6.97E-03	2.53E-15	-1.24E-03	-2.41E-16
mcp250:1	Solved	Acc.	40.15	2.35E-03	7.45E-15	5.00E-03	3.96E-15	-1.61E-03	-1.06E-15
mcp250:2	Solved	Acc.	31.29	1.28E-03	4.79E-15	5.91E-03	2.95E-15	-1.04E-03	1.97E-16

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
mcp250:3	Solved	Acc.	29.56	1.30E-03	3.38E-15	7.50E-03	9.87E-15	-1.07E-03	-5.69E-16
mcp250:4	Solved	Acc.	29.6	3.69E-03	5.80E-15	9.53E-03	2.97E-16	-1.03E-03	4.72E-15
mcp500:1	Solved	Acc.	162.61	2.12E-03	6.47E-15	6.12E-03	3.71E-15	-1.53E-03	-5.52E-16
mcp500:2	Solved	Acc.	141.33	2.22E-03	6.05E-15	8.38E-03	1.14E-14	-1.44E-03	-1.22E-15
mcp500:3	Solved	Fail	127.48	1.64E-03	6.93E-15	1.07E-02	1.02E-14	-1.29E-03	-1.16E-15
mcp500:4	Solved	Fail	121.32	1.92E-03	5.38E-15	1.27E-02	2.14E-14	-1.36E-03	-2.81E-15
qap10	Solved	Fail	3.51	3.36E-04	2.37E-18	2.93E-02	1.95E-14	-3.56E-03	-1.45E-15
qap5	Solved	Acc.	0.65	7.55E-04	5.31E-17	6.12E-03	5.13E-16	2.73E-03	-4.65E-16
qap6	Solved	Fail	1.73	6.58E-04	5.98E-18	1.08E-02	6.81E-15	-5.57E-03	-1.42E-15
qap7	Solved	Fail	2.3	3.46E-04	7.17E-18	1.17E-02	6.85E-15	-4.23E-03	-2.68E-15
qap8	Solved	Fail	3.11	5.63E-04	4.92E-18	1.87E-02	4.99E-15	-5.26E-03	2.04E-15
qap9	Solved	Fail	4.48	3.65E-04	3.89E-18	2.94E-02	1.03E-14	-3.08E-03	5.65E-16
qpG11	Solved	Fail	2439.8	1.95E-03	1.02E-14	2.85E-02	5.82E-14	-7.01E-04	1.38E-16
qpG51	Time	Fail	3600.65	5.61E-02	2.67E-13	1.73E+00	1.48E-12	-1.06E-01	-9.26E-14
ss30	Time	Fail	3600.15	4.26E-01	3.98E-15	5.55E-01	0.00E+00	7.46E-01	4.24E-12
theta1	Solved	Fail	1.09	7.21E-04	7.25E-17	2.52E-02	1.38E-14	5.15E-05	2.50E-16

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
theta2	Solved	Fail	2.34	5.61E-04	1.47E-17	4.99E-02	3.93E-14	1.01E-03	-2.51E-16
theta3	Solved	Fail	4.95	7.06E-04	1.70E-17	6.75E-02	5.50E-14	-1.48E-03	9.84E-16
theta4	Solved	Fail	7.92	7.52E-04	1.20E-17	9.25E-02	7.26E-14	-1.59E-03	-4.85E-16
theta5	Solved	Fail	12.5	8.58E-04	1.48E-17	8.88E-02	6.41E-14	-1.02E-03	1.50E-16
theta6	Solved	Fail	15.57	2.32E-04	1.10E-17	1.50E-01	1.41E-13	8.18E-04	4.71E-16
thetaG11	Solved	Fail	1269.73	1.07E-02	4.87E-14	1.22E-02	3.13E-13	-1.41E-03	-6.90E-14
thetaG51	Time	Fail	3600.25	3.45E+00	6.30E-14	1.11E-01	3.39E-11	9.92E-01	9.50E-14
truss1	Solved	Acc.	1.06	1.08E-03	5.31E-17	3.95E-04	8.88E-16	-2.07E-03	-6.99E-16
truss2	Solved	Acc.	97.36	1.54E-03	2.39E-15	1.04E-04	3.18E-14	-2.43E-03	6.15E-15
truss3	Solved	Acc.	1.97	1.08E-03	1.01E-15	9.81E-04	1.65E-16	-1.78E-03	-1.25E-15
truss4	Solved	Acc.	1.37	1.08E-03	6.67E-16	2.09E-04	1.98E-16	-1.87E-03	1.25E-15
truss5	Solved	Acc.	171.38	6.36E-04	9.63E-15	1.00E-03	5.97E-16	2.42E-03	-1.11E-13
truss6	Time	Acc.	3600.08	1.97E-03	4.88E-14	1.89E-04	3.11E-13	-3.13E-03	-1.45E-13
truss7	Solved	Acc.	3012.07	1.08E-03	2.96E-16	1.06E-04	1.12E-13	-1.72E-03	2.11E-14
truss8	Error	Fail							

A.1.2 BPM Results on Mittelmann Library

On the Mittelmann Library problems, BPM had a shifted geometric mean of 2480.51 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
1dc.1024	Solved	Fail	396.16	2.05E-04	4.63E-17	5.10E-01	4.82E-13	-6.72E-04	-1.05E-15
1et.1024	Solved	Fail	344.89	7.03E-04	5.53E-17	5.08E-01	2.01E-12	-3.19E-03	-1.49E-15
1tc.1024	Solved	Fail	490.81	3.07E-04	5.88E-17	5.10E-01	3.43E-13	-1.05E-03	6.57E-16
1zc.1024	Solved	Fail	335.99	7.75E-04	1.49E-16	4.49E-01	1.04E-12	-1.41E-03	-1.40E-15
Bex2_1_5	Solved	Fail	69.8	1.27E-03	9.15E-17	2.79E-02	1.55E-14	-6.17E-03	-5.77E-15
biggs	Time	Fail	3600.1	6.28E+02	2.34E-10	1.24E+02	8.17E-10	-9.97E-01	7.46E-08
broyden25	Time	Acc.	3600.36	6.59E-03	1.80E-14	1.13E-03	2.49E-14	-2.20E-01	-2.66E-13
Bst_jcbpaf2	Solved	Fail	129.12	9.89E-04	2.99E-17	2.78E-02	2.21E-16	-2.91E-02	-3.09E-16
buck3	Time	Fail	3600.2	1.63E-01	5.60E-14	4.24E-02	9.01E-14	-3.78E-01	1.19E-11
buck4	Time	Fail	3600.54	9.05E-02	1.73E-14	4.50E-02	1.13E-15	-3.21E-01	-1.19E-14
buck5	Time	Fail	3602.27	2.90E-01	7.14E-16	7.94E-02	1.57E-15	-2.25E-01	9.34E-14
butcher	Solved	Acc.	2024.2	3.88E-03	8.84E-15	1.15E-03	8.84E-15	-4.58E-02	-4.81E-14
cancer_100	Solved	Fail	135.53	3.10E-02	1.78E-13	3.97E-02	2.16E-14	-2.00E-03	2.60E-15

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
checker_1.5	Time	Fail	3617.39	0.00E+00	0.00E+00	6.09E-02	0.00E+00	-6.82E-03	0.00E+00
chs_5000	Solved	Fail	81.62	1.98E-03	1.15E-16	3.48E-02	1.59E-16	2.13E-02	1.56E-15
cnhil10	Error	Fail							
cnhil8	Error	Fail							
cphil10	Error	Fail							
cphil12	Error	Fail							
e_moment_quadknap_17_100_0.5_2_2	Solved	Acc.	62.92	1.93E-03	1.03E-15	9.96E-04	6.24E-17	6.05E-04	2.49E-16
e_moment_stable_17_0.5_2_2	Solved	Acc.	52.54	5.68E-04	1.11E-12	1.36E-03	2.20E-16	-2.51E-02	2.43E-09
foot	Time	Fail	3600.43	3.04E+00	0.00E+00	2.50E+02	5.21E-12	-8.49E-01	1.69E-15
G40_mb	Time	Fail	3603.55	3.60E-01	8.98E-15	1.84E-01	9.58E-14	-2.98E-02	-4.38E-15
G40mc	Time	Fail	3600.44	1.74E-02	4.66E-14	2.61E-02	1.76E-14	-8.12E-03	-2.29E-15

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
G48_mb	Time	Fail	3629.23	5.76E-02	4.46E-15	6.66E+00	1.46E-15	-2.12E-01	-2.27E-16
G48mc	Time	Fail	3622.07	1.29E-01	3.62E-15	6.29E+00	3.93E-13	-6.02E-02	5.39E-16
G55mc	Time	Fail	3660.44	0.00E+00	0.00E+00	9.30E+00	0.00E+00	-6.04E-04	0.00E+00
G59mc	Time	Fail	3602.2	0.00E+00	0.00E+00	8.39E+00	0.00E+00	-1.49E-01	0.00E+00
hand	Time	Fail	3600.32	2.14E-01	2.56E-14	3.23E+00	1.54E-11	-2.68E-02	-1.06E-13
ice_2.0	Time	Fail	3912.86	0.00E+00	0.00E+00	2.71E-02	0.00E+00	-6.80E-05	0.00E+00
inc_1200	Solved	Fail	3364.28	6.22E-04	1.69E-15	3.65E-02	1.28E-13	-1.89E-01	-6.03E-13
inc_600	Solved	Fail	721.28	7.85E-04	1.48E-15	2.56E-02	5.59E-14	-2.09E-01	2.10E-14
mater:3	Time	Acc.	3600.14	2.20E-03	1.86E-15	4.90E-04	4.25E-14	-3.75E-03	-2.18E-15
mater:4	Time	Fail	3600.17	1.54E-01	3.06E-15	3.15E-02	1.40E-14	-1.08E-01	-1.97E-15
mater:5	Time	Fail	3600.61	3.97E-02	3.35E-16	2.92E-02	1.82E-12	1.02E-01	-2.00E-12
mater:6	Time	Fail	3600.26	9.75E-03	6.22E-16	5.41E-02	4.55E-13	6.93E-02	-4.85E-13
neosfbr25	Time	Fail	3601.91	4.53E-04	1.10E-16	1.83E-02	1.98E-15	-8.58E-03	3.12E-16
neosfbr30e8	Error	Fail							
neu1	Time	Fail	3600.09	2.36E-02	7.42E-16	8.03E-03	8.17E-13	1.00E+00	3.13E-15
neu1g	Time	Fail	3600.5	2.32E-01	4.85E-15	5.07E-02	2.48E-12	1.00E+00	1.10E-14

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
neu2c	Time	Fail	3600.22	3.83E-02	1.69E-13	1.98E-02	2.51E-13	1.00E+00	-2.59E-11
neu2	Time	Fail	3600.53	2.58E-03	1.08E-16	3.15E-04	7.65E-13	1.00E+00	-8.55E-14
neu2g	Time	Fail	3600.1	3.92E-01	5.86E-11	4.57E-01	1.06E-09	1.00E+00	-1.86E-09
neu3	Solved	Solved	0.91	2.40E-06	0.00E+00	3.06E-04	0.00E+00	1.68E-05	3.39E-20
neu3g	Solved	Fail	549.63	1.39E-03	1.32E-16	1.01E-04	4.61E-13	2.29E-01	7.16E-13
nonc_500	Solved	Fail	19.1	2.69E-03	1.87E-16	1.17E-02	3.58E-16	-2.36E-02	-1.32E-14
p_auss2_3.0	Time	Fail	4083.77	0.00E+00	0.00E+00	7.98E-02	0.00E+00	-5.91E-05	0.00E+00
rabmo	Solved	Acc.	394.48	8.30E-04	2.92E-15	1.56E-03	1.98E-15	-7.68E-02	4.07E-16
reimer5	Solved	Acc.	1600.5	1.56E-03	4.11E-14	2.16E-03	2.90E-16	-6.90E-02	-3.97E-15
rendl1_2000_1e:6	Time	Fail	3605.04	5.96E-01	5.83E-15	1.39E+00	1.82E-12	-5.24E-02	9.13E-16
ros_2000	Solved	Fail	108.03	3.12E-03	1.23E-16	2.26E-02	6.47E-15	-1.63E-01	-1.06E-15
ros_500	Solved	Fail	31.19	2.40E-03	1.03E-15	1.16E-02	7.55E-15	1.65E-01	-2.68E-14
rose13	Solved	Fail	1216.17	4.23E-03	3.84E-13	8.56E-04	5.67E-14	9.99E-01	3.79E-09
rose15	Solved	Fail	3290.85	4.57E-03	1.95E-12	5.93E-04	1.89E-13	9.98E-01	5.08E-08
sensor_1000	Time	Fail	3600.73	2.67E-04	4.52E-15	8.45E-02	1.68E-12	-4.21E-01	4.73E-12

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
sensor_500	Solved	Fail	457.4	4.47E-04	2.37E-15	3.02E-02	1.99E-13	-5.37E-02	-3.92E-13
shmup3	Solved	Fail	761.86	6.22E-03	2.35E-15	1.96E-02	2.45E-16	-4.41E-01	-2.11E-13
shmup4	Solved	Fail	983.44	8.91E-03	2.85E-16	2.77E-02	5.90E-17	-6.07E-01	-4.78E-13
shmup5	Time	Fail	3614.37	9.70E-07	0.00E+00	1.14E-01	3.17E-17	-2.40E-02	4.39E-15
spar060:020:1 _LS	Solved	Acc.	379.87	5.09E-04	2.00E-15	9.08E-03	3.36E-14	-2.44E-03	-1.74E-14
swissroll	Time	Fail	3600.09	4.45E-01	2.47E-13	7.88E-01	3.93E-11	8.28E-01	1.27E-13
taha1a	Solved	Acc.	115.41	1.80E-04	8.51E-15	1.48E-03	3.46E-16	-1.24E-03	1.78E-12
taha1b	Solved	Acc.	473.86	6.14E-04	1.00E-16	1.71E-03	2.72E-14	-1.32E-02	3.18E-14
taha1c	Solved	Acc.	400.18	1.54E-04	8.29E-15	1.48E-03	5.43E-16	-1.08E-03	-7.83E-13
theta12	Solved	Fail	57.51	5.34E-04	2.56E-17	2.97E-01	2.43E-13	-1.30E-03	-3.20E-16
tiger_texture	Time	Fail	3601.82	3.49E+00	2.57E-14	2.26E+02	4.63E-12	-9.92E-01	-3.84E-13
trto3	Time	Fail	3600.08	9.46E-02	2.13E-13	5.36E-02	5.19E-16	-6.42E-01	1.50E-14
trto4	Time	Fail	3600.38	9.51E-02	4.22E-14	4.28E-02	3.24E-16	-6.44E-01	6.38E-14
trto5	Time	Fail	3600.15	3.16E-01	3.79E-15	4.68E-02	5.71E-16	-3.14E-01	1.01E-14
vibra3	Time	Fail	3600.11	8.54E-02	7.00E-14	2.96E-02	5.88E-16	-3.43E-01	1.93E-13

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
vibra4	Time	Fail	3600.65	2.26E-01	1.53E-14	3.29E-02	7.04E-16	-3.23E-01	4.68E-14
vibra5	Time	Fail	3604.1	4.92E-01	8.64E-16	7.45E-02	8.24E-16	-2.71E-01	6.97E-14
yalsdp	Time	Fail	3600.23	1.35E-02	1.98E-16	3.91E-02	6.73E-14	2.99E-02	-6.93E-14

A.2 MPRW Results

MPRW had an overall shifted geometric mean of 998.59 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

A.2.1 MPRW Results on SDPLIB

On the SDPLIB problems, MPRW had a shifted geometric mean of 710.36 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
arch0	Time	Fail	3600.1	1.61E-05	1.27E-17	1.25E-01	3.18E-11	-9.07E-02	-1.84E-11
arch2	Time	Fail	3600.09	1.39E-05	1.66E-17	1.55E-01	2.69E-11	-1.30E-01	7.03E-13
arch4	Time	Fail	3600.08	1.12E-05	1.74E-17	1.71E-01	2.79E-11	-1.35E-01	1.88E-11

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
arch8	Time	Fail	3600.08	1.70E-04	1.31E-16	1.39E-01	1.56E-11	-4.86E-01	4.10E-10
control10	Time	Fail	3600.1	1.18E+00	6.49E-16	4.13E+02	2.57E-08	9.81E-01	-2.21E-11
control11	Time	Fail	3600.11	1.52E+00	6.32E-16	6.99E+02	3.69E-08	9.87E-01	-5.15E-11
control1	Time	Fail	3600.04	3.75E-03	4.76E-16	3.87E-01	9.35E-11	-3.79E-01	3.20E-09
control2	Time	Fail	3600.06	1.14E-03	6.23E-17	6.57E-01	5.97E-12	-2.72E-01	5.72E-13
control3	Time	Fail	3600.05	2.40E-01	2.54E-18	1.64E+00	3.47E-09	9.94E-01	-1.26E-12
control4	Time	Fail	3600.33	1.62E-01	2.61E-16	7.29E+00	6.96E-09	9.31E-01	1.12E-10
control5	Time	Fail	3600.05	1.83E-01	7.74E-17	6.98E+00	4.35E-09	9.63E-01	1.33E-10
control6	Time	Fail	3600.05	8.52E-02	6.96E-16	2.58E+01	6.85E-09	9.26E-01	1.77E-10
control7	Time	Fail	3600.05	9.21E-02	4.08E-16	4.73E+01	8.73E-09	9.43E-01	-8.21E-11
control8	Time	Fail	3600.07	1.44E-01	6.52E-16	9.13E+01	2.37E-08	9.56E-01	-4.43E-10
control9	Time	Fail	3600.1	7.26E-01	5.55E-16	2.90E+02	1.30E-08	9.78E-01	3.41E-11
equalG11	Time	Acc.	3600.51	3.05E-03	0.00E+00	5.56E-03	0.00E+00	-1.20E-01	-2.63E-13
equalG51	Time	Acc.	3601.11	4.46E-03	0.00E+00	8.24E-03	0.00E+00	-2.43E-02	-1.60E-17
gpp100	Solved	Solved	2232.96	3.30E-05	1.95E-15	8.75E-04	3.82E-12	-3.12E-04	1.45E-12
gpp124:1	Solved	Solved	1300.27	9.51E-04	4.80E-15	5.94E-04	2.32E-13	-2.21E-03	-3.17E-12

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
gpp124:2	Solved	Solved	1027.8	5.50E-04	1.94E-15	8.02E-04	3.69E-13	-7.66E-04	-3.60E-13
gpp124:3	Solved	Solved	818.24	3.74E-04	2.23E-15	9.18E-04	8.45E-13	-3.20E-04	-9.30E-13
gpp124:4	Solved	Solved	1277.13	5.67E-04	3.66E-15	1.00E-03	1.61E-12	-2.46E-04	2.14E-12
gpp250:1	Time	Acc.	3600.23	2.46E-03	3.77E-15	1.62E-03	6.31E-13	-6.22E-03	-3.38E-12
gpp250:2	Time	Acc.	3600.38	1.58E-04	7.55E-15	2.11E-03	8.94E-12	-1.82E-03	1.34E-10
gpp250:3	Time	Acc.	3600.07	1.36E-03	3.95E-15	1.57E-03	2.48E-12	-6.07E-04	-7.54E-13
gpp250:4	Time	Acc.	3600.1	1.16E-03	3.90E-15	2.66E-03	9.40E-13	-5.51E-04	1.80E-12
gpp500:1	Time	Fail	3600.13	8.50E-03	8.47E-15	1.63E-02	8.42E-13	-5.27E-02	1.49E-13
gpp500:2	Time	Fail	3600.14	9.74E-03	5.28E-15	2.45E-02	4.20E-13	-2.27E-02	3.65E-13
gpp500:3	Time	Fail	3600.1	4.94E-03	5.15E-15	2.84E-02	8.33E-13	-1.06E-02	-4.69E-13
gpp500:4	Time	Fail	3600.04	5.60E-03	5.25E-15	3.05E-02	5.67E-13	-5.96E-03	3.98E-14
hinf10	Solved	Solved	7.14	2.58E-04	2.30E-15	1.00E-03	1.62E-14	-1.87E-02	3.49E-13
hinf11	Solved	Solved	3.9	2.41E-04	6.07E-16	1.00E-03	2.82E-14	-2.19E-02	3.65E-13
hinf12	Solved	Solved	13.95	2.67E-04	1.52E-15	1.00E-03	5.06E-14	-1.04E-01	5.74E-12
hinf13	Time	Acc.	3600.1	1.11E-04	1.15E-13	1.66E-03	1.26E-14	-2.23E-01	2.01E-11
hinf14	Solved	Solved	249.79	4.09E-05	7.44E-15	1.00E-03	1.04E-13	-3.51E-02	1.11E-11

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
hinf15	Solved	Solved	2458.27	4.61E-05	5.56E-14	1.00E-03	2.97E-13	-9.37E-02	7.69E-11
hinf1	Solved	Solved	0.87	3.76E-04	2.25E-16	9.98E-04	1.33E-14	-2.59E-03	2.13E-14
hinf2	Solved	Solved	123.66	2.17E-05	1.68E-14	1.00E-03	2.97E-13	-2.69E-02	-7.38E-12
hinf3	Solved	Solved	155	8.56E-05	4.39E-14	1.00E-03	1.62E-15	-4.98E-02	-4.80E-14
hinf4	Solved	Solved	6.01	2.24E-04	7.58E-16	1.00E-03	1.35E-15	-2.14E-03	1.52E-12
hinf5	Solved	Solved	1337.49	1.28E-04	4.67E-14	1.00E-03	7.90E-13	-1.54E-02	1.87E-11
hinf6	Solved	Solved	989.29	1.72E-04	7.06E-14	1.00E-03	1.55E-12	-7.13E-02	-6.27E-11
hinf7	Solved	Solved	99.26	1.92E-05	2.83E-16	1.00E-03	1.02E-13	-1.93E-02	-1.63E-12
hinf8	Solved	Solved	2640.2	1.23E-05	1.73E-14	1.00E-03	1.74E-12	-6.60E-02	2.72E-10
hinf9	Solved	Solved	0.58	1.59E-04	2.75E-16	1.00E-03	1.30E-15	-7.58E-04	9.38E-15
maxG11	Time	Fail	3600.09	1.76E-06	0.00E+00	5.33E-02	0.00E+00	-4.20E-03	1.79E-14
maxG32	Time	Fail	3604.8	4.65E-04	0.00E+00	1.28E+00	0.00E+00	-4.21E-02	1.43E-14
maxG51	Time	Fail	3600.76	1.09E-05	0.00E+00	3.74E-02	0.00E+00	-8.33E-03	-2.61E-14
maxG55	Time	Fail	3606.58	0.00E+00	0.00E+00	8.66E+00	0.00E+00	-3.63E-02	0.00E+00
maxG60	Time	Fail	3810.73	0.00E+00	0.00E+00	1.03E+01	0.00E+00	-1.12E-02	0.00E+00
mcp100	Solved	Solved	207.4	1.22E-07	2.38E-15	7.82E-04	4.51E-14	-9.79E-05	1.97E-14

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
mcp124:1	Solved	Solved	804.43	3.69E-08	3.58E-15	5.94E-04	5.27E-14	-1.25E-04	5.55E-16
mcp124:2	Solved	Solved	415.89	9.70E-08	3.30E-15	8.02E-04	1.09E-13	-1.11E-04	-3.92E-14
mcp124:3	Solved	Solved	402.25	9.89E-08	3.56E-15	9.17E-04	1.33E-13	-8.04E-05	-4.88E-14
mcp124:4	Solved	Solved	410.97	6.63E-08	2.77E-15	1.00E-03	9.94E-14	-9.14E-05	-3.63E-14
mcp250:1	Time	Acc.	3600.08	5.55E-08	8.14E-15	1.75E-03	1.90E-13	-3.62E-04	5.42E-14
mcp250:2	Solved	Solved	3307.7	2.95E-08	3.84E-15	7.17E-04	2.01E-13	-8.95E-05	4.65E-14
mcp250:3	Solved	Solved	3045.65	3.25E-08	3.05E-15	9.30E-04	5.09E-13	-9.56E-05	1.21E-13
mcp250:4	Time	Acc.	3600.12	2.67E-08	4.87E-15	1.41E-03	3.20E-13	-1.41E-04	2.13E-13
mcp500:1	Time	Fail	3600.2	4.29E-07	3.13E-15	1.69E-02	5.32E-13	-3.15E-03	-1.18E-14
mcp500:2	Time	Fail	3600.18	5.26E-07	6.57E-15	2.33E-02	6.61E-13	-3.19E-03	6.16E-14
mcp500:3	Time	Fail	3600.1	5.11E-07	5.24E-15	2.88E-02	8.69E-13	-3.03E-03	-1.59E-14
mcp500:4	Time	Fail	3600.19	5.09E-07	3.67E-15	3.33E-02	1.01E-12	-2.89E-03	1.37E-13
qap10	Solved	Solved	77.6	3.20E-05	1.90E-18	9.99E-04	2.62E-14	-5.25E-04	2.11E-15
qap5	Solved	Solved	2.26	2.37E-04	2.59E-17	2.96E-04	0.00E+00	-1.80E-04	6.65E-15
qap6	Solved	Solved	21.98	5.09E-05	8.61E-18	1.00E-03	3.12E-15	-9.40E-04	4.30E-15
qap7	Solved	Solved	39	3.23E-05	5.56E-18	1.00E-03	2.74E-14	-7.22E-04	-5.22E-15

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
qap8	Solved	Solved	22.45	7.29E-05	3.88E-18	9.99E-04	4.75E-14	-9.73E-04	-6.74E-15
qap9	Solved	Solved	69.23	2.69E-05	6.43E-18	1.00E-03	2.94E-14	-2.96E-04	1.61E-16
qpG11	Time	Fail	3601.64	2.24E-04	0.00E+00	6.79E-01	0.00E+00	-1.07E-02	-2.82E-14
qpG51	Time	Fail	3604.16	1.04E-03	0.00E+00	5.91E+00	0.00E+00	-7.81E-02	-3.05E-14
ss30	Time	Fail	3600.09	4.62E-03	8.85E-16	7.89E-02	4.25E-12	-2.31E-01	2.01E-11
theta1	Solved	Solved	0.78	6.42E-05	5.62E-17	9.56E-04	1.99E-14	8.49E-05	6.44E-15
theta2	Solved	Solved	5.72	6.25E-06	3.30E-17	9.98E-04	2.30E-14	-7.75E-06	1.78E-16
theta3	Solved	Solved	8.9	2.74E-06	1.58E-17	9.93E-04	6.29E-14	-6.80E-06	6.49E-16
theta4	Solved	Solved	15.72	1.86E-06	1.33E-17	9.89E-04	9.36E-14	-2.26E-06	6.73E-16
theta5	Solved	Fail	1.78	5.00E-01	0.00E+00	4.44E-14	0.00E+00	9.89E-01	0.00E+00
theta6	Solved	Fail	2.22	5.00E-01	0.00E+00	4.40E-14	0.00E+00	9.91E-01	0.00E+00
thetaG11	Time	Fail	3600.23	3.83E-04	0.00E+00	1.19E-01	0.00E+00	-1.36E-02	-1.49E-13
thetaG51	Time	Fail	3601.03	6.09E-03	0.00E+00	3.49E-01	0.00E+00	1.12E-01	1.63E-11
truss1	Solved	Solved	0.63	7.49E-04	3.15E-16	6.59E-04	1.67E-15	-1.86E-03	-1.64E-15
truss2	Solved	Acc.	24.97	9.40E-04	4.77E-15	4.16E-04	5.74E-14	1.56E-03	5.46E-14
truss3	Solved	Solved	3.32	3.88E-05	3.69E-16	6.63E-04	3.63E-16	-2.40E-04	9.24E-16

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
truss4	Solved	Solved	0.68	6.09E-04	3.78E-16	6.64E-04	6.22E-15	-1.68E-03	-1.83E-15
truss5	Solved	Solved	407.65	2.33E-05	1.02E-14	4.12E-04	1.48E-16	-4.64E-05	9.47E-14
truss6	Solved	Acc.	602.78	3.06E-03	1.34E-15	6.59E-04	2.62E-11	4.04E-03	2.62E-11
truss7	Solved	Acc.	408.07	2.95E-03	7.79E-16	6.67E-04	5.67E-12	3.80E-03	-4.21E-13
truss8	Solved	Solved	1693.93	8.41E-06	9.35E-15	4.07E-04	2.22E-16	-1.56E-04	5.43E-14

A.2.2 MPRW Results on Mittelmann Library

On the Mittelmann Library problems, MPRW had a shifted geometric mean of 1503.26 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
1dc.1024	Time	Acc.	3600.82	2.62E-06	0.00E+00	2.77E-03	0.00E+00	-1.13E-05	2.36E-15
1et.1024	Time	Acc.	3600.16	1.32E-06	0.00E+00	1.77E-03	0.00E+00	-5.69E-06	2.71E-14
1tc.1024	Time	Acc.	3600.92	3.08E-06	0.00E+00	5.54E-03	0.00E+00	-1.49E-05	-2.22E-15
1zc.1024	Solved	Fail	21.2	5.00E-01	0.00E+00	0.00E+00	0.00E+00	9.96E-01	0.00E+00
Bex2_1_5	Solved	Solved	253.61	2.21E-05	5.40E-18	9.66E-04	1.25E-14	-1.50E-04	2.26E-15

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
biggs	Time	Fail	3600.09	8.48E-01	3.98E-13	1.24E+00	2.34E-12	-1.00E+00	6.85E-15
broyden25	Time	Acc.	3600.3	5.96E-03	8.66E-16	3.88E-03	2.69E-14	-2.20E-01	-3.90E-14
Bst_jcbpaf2	Solved	Solved	710.44	3.60E-06	1.81E-17	1.00E-03	8.63E-16	-3.55E-04	3.00E-16
buck3	Time	Fail	3600.1	1.69E-01	6.44E-14	4.22E-02	1.02E-13	-3.78E-01	1.43E-11
buck4	Time	Fail	3600.23	1.99E-01	0.00E+00	3.62E-02	0.00E+00	-3.33E-01	8.88E-14
buck5	Time	Fail	3603.07	7.71E-01	0.00E+00	5.69E-02	0.00E+00	-1.80E-01	-1.68E-14
butcher	Time	Acc.	3600.12	4.81E-05	4.02E-15	4.59E-03	7.66E-16	-9.84E-02	4.75E-16
cancer_100	Time	Fail	3600.21	1.50E-04	8.53E-14	5.55E-02	0.00E+00	-2.56E-04	2.46E-13
checker_1.5	Time	Fail	3611.47	7.85E-05	0.00E+00	5.40E-02	1.30E-17	-1.14E-02	5.84E-18
chs_5000	Solved	Solved	62.39	2.30E-06	9.54E-17	9.95E-04	2.22E-16	-1.44E-05	2.87E-14
cnhil10	Solved	Acc.	25.47	1.04E-03	1.12E-16	4.04E-04	9.40E-15	-5.22E-04	5.34E-16
cnhil8	Solved	Acc.	6.83	1.03E-03	1.64E-16	4.81E-04	2.45E-15	-2.13E-04	4.51E-15
cphil10	Solved	Solved	1.16	3.88E-16	0.00E+00	5.27E-16	0.00E+00	1.72E-16	0.00E+00
cphil12	Solved	Solved	1.89	5.75E-16	0.00E+00	6.25E-16	0.00E+00	-2.47E-16	0.00E+00

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
e_moment_quadknap_17_100_0.5_2_2	Time	Acc.	3600.09	1.59E-05	2.80E-16	3.09E-03	2.44E-16	-4.18E-02	1.08E-16
e_moment_stable_17_0.5_2_2	Solved	Solved	1573.43	2.99E-06	3.95E-16	1.66E-04	1.09E-15	-3.20E-03	7.82E-15
foot	Time	Fail	3605.67	6.53E+00	3.49E-15	2.50E+02	6.26E-12	-9.57E-01	-5.28E-16
G40_mb	Time	Fail	3605	3.26E-01	3.92E-15	6.83E-01	6.58E-13	-7.74E-02	-7.89E-15
G40mc	Time	Fail	3601.45	2.19E-04	4.37E-15	6.77E-01	3.18E-13	-7.30E-02	2.13E-14
G48_mb	Time	Fail	3621.33	6.82E-04	3.34E-15	7.62E+00	1.46E-15	-1.85E-01	-2.07E-16
G48mc	Time	Fail	3602.05	4.17E-04	6.49E-15	1.01E+01	1.91E-15	-8.20E-02	-9.09E-17
G55mc	Time	Fail	3669.6	0.00E+00	0.00E+00	9.30E+00	0.00E+00	-4.77E-02	0.00E+00
G59mc	Time	Fail	3609.58	5.45E-03	7.80E-15	7.71E+00	1.06E-15	-7.59E-01	-9.74E-17
hand	Time	Fail	3600.24	1.84E-01	2.76E-14	3.37E+00	5.00E-12	-2.70E-02	3.64E-13
ice_2.0	Time	Fail	3776.53	3.77E-15	0.00E+00	2.71E-02	0.00E+00	-3.69E-04	0.00E+00

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
inc_1200	Time	Fail	3600.14	1.04E-04	0.00E+00	8.00E-02	2.82E-13	-2.02E-01	3.55E-13
inc_600	Time	Fail	3600.17	2.23E-05	0.00E+00	2.30E-02	4.12E-13	-2.08E-01	-9.92E-14
mater:3	Solved	Solved	1648.32	7.14E-04	1.86E-15	5.00E-04	9.05E-14	-1.57E-03	2.25E-13
mater:4	Time	Fail	3600.34	1.95E-02	2.18E-16	1.12E-02	4.55E-13	-3.79E-02	-4.30E-13
mater:5	Solved	Fail	33.07	5.05E+00	1.03E-17	4.92E-04	5.46E-16	-8.82E-01	-1.43E-17
mater:6	Solved	Fail	85.67	5.06E+00	1.11E-17	4.92E-04	2.53E-16	-8.58E-01	6.21E-17
neosfbr25	Time	Fail	3600.81	3.89E-05	9.26E-17	3.04E-02	2.39E-15	-1.10E-02	-1.06E-15
neosfbr30e8	Error	Fail							
neu1	Solved	Solved	116.35	5.82E-04	1.97E-16	6.61E-04	3.62E-14	-1.50E-01	-2.33E-12
neu1g	Solved	Solved	1259.18	1.34E-04	5.30E-16	1.09E-04	2.58E-14	-2.51E-04	2.33E-13
neu2c	Time	Fail	3600.15	5.20E-04	7.70E-16	1.40E-02	6.45E-13	-3.98E-03	-5.44E-14
neu2	Solved	Solved	487.57	1.90E-04	2.33E-16	7.01E-04	4.82E-14	-3.38E-02	1.57E-13
neu2g	Time	Fail	3600.12	5.54E-04	1.51E-16	7.14E-02	2.64E-12	6.52E-02	3.17E-11
neu3	Solved	Solved	1.03	1.17E-04	0.00E+00	2.92E-04	6.28E-16	8.15E-04	-8.16E-15
neu3g	Solved	Solved	202.32	1.32E-05	3.56E-16	4.01E-04	8.87E-17	-1.44E-03	-2.15E-15
nonc_500	Solved	Solved	1688.78	1.77E-05	9.35E-17	9.99E-04	1.12E-14	-4.45E-03	-2.35E-14

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
p_auss2_3.0	Time	Fail	3813.79	1.11E-16	0.00E+00	7.98E-02	0.00E+00	-5.23E-04	0.00E+00
rabmo	Time	Acc.	3600.14	7.76E-06	2.82E-15	2.01E-03	9.87E-17	-8.48E-02	-1.10E-17
reimer5	Time	Acc.	3600.05	3.99E-05	2.04E-15	7.75E-03	2.13E-16	-7.62E-02	8.90E-16
rendl1_2000_1e:6	Time	Fail	3600.07	4.22E-01	4.55E-15	2.15E+00	1.76E-12	-7.38E-02	-1.98E-14
ros_2000	Solved	Solved	446.34	1.98E-05	6.11E-17	9.98E-04	2.36E-14	-9.83E-05	2.28E-14
ros_500	Solved	Solved	97.13	4.52E-04	2.31E-16	8.07E-04	2.19E-15	-1.91E-03	-1.71E-15
rose13	Solved	Solved	383.97	1.23E-06	2.94E-16	1.18E-04	1.34E-15	-8.88E-04	-2.57E-13
rose15	Solved	Solved	600.49	8.35E-07	4.46E-16	1.55E-04	2.57E-15	-2.06E-02	6.36E-12
sensor_1000	Time	Fail	3600.38	1.14E-04	0.00E+00	1.24E-01	2.34E-12	-3.91E-01	1.01E-12
sensor_500	Solved	Solved	1702.08	3.42E-06	2.95E-15	9.99E-04	1.91E-12	-5.94E-04	-5.91E-12
shmup3	Time	Fail	3600.07	1.07E-05	0.00E+00	3.03E-02	1.29E-16	-4.52E-01	-2.27E-15
shmup4	Time	Fail	3603.98	1.61E-10	8.60E-17	3.50E-02	2.06E-17	-6.50E-01	5.87E-16
shmup5	Time	Fail	3627.71	1.34E-06	0.00E+00	8.50E-02	2.78E-17	-1.69E-01	-2.84E-16
spar060:020:1_LS	Time	Acc.	3600.12	1.99E-14	1.85E-15	2.38E-03	3.96E-14	-4.93E-05	1.17E-13

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
swissroll	Time	Fail	3600.16	2.25E-02	1.61E-14	5.22E+00	4.86E-11	-1.00E+00	-1.94E-12
taha1a	Solved	Solved	169.26	2.73E-05	5.33E-17	9.70E-04	7.58E-16	-5.75E-04	1.18E-15
taha1b	Solved	Solved	1075.78	5.05E-04	5.91E-17	1.00E-03	6.02E-15	-1.17E-02	-4.86E-15
taha1c	Solved	Solved	499.21	4.60E-05	4.25E-17	9.70E-04	4.02E-16	-4.78E-04	2.68E-15
theta12	Solved	Fail	8.8	5.00E-01	0.00E+00	1.02E-13	0.00E+00	9.94E-01	0.00E+00
tiger_texture	Time	Fail	3602.83	2.93E+01	1.18E-14	8.02E+01	6.29E-13	-9.79E-01	1.30E-14
trto3	Time	Fail	3600.09	1.47E-01	3.38E-13	5.13E-02	1.41E-15	-6.29E-01	-1.39E-14
trto4	Time	Fail	3600.26	1.22E-01	0.00E+00	4.15E-02	1.07E-15	-6.49E-01	-7.22E-14
trto5	Time	Fail	3600.96	2.32E-01	0.00E+00	3.72E-02	2.74E-16	-3.72E-01	-3.70E-15
vibra3	Time	Fail	3600.18	7.01E-02	5.51E-14	2.81E-02	3.81E-15	-3.38E-01	3.24E-14
vibra4	Time	Fail	3600.11	1.16E-01	0.00E+00	2.89E-02	7.92E-16	-3.09E-01	3.93E-14
vibra5	Time	Fail	3600.31	5.90E-01	0.00E+00	4.87E-02	1.68E-15	-2.05E-01	6.30E-14
yalsdp	Time	Fail	3600.32	6.34E-07	9.48E-17	3.55E-02	2.11E-13	-1.93E-02	2.08E-13

A.3 SDPAD Results

SDPAD had an overall shifted geometric mean of 926.59 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

A.3.1 SDPAD Results on SDPLIB

On the SDPLIB problems, SDPAD had a shifted geometric mean of 559.97 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
arch0	Time	Fail	3600.12	4.59E-03	4.44E-17	3.46E-02	1.78E-12	-6.41E-02	1.88E-11
arch2	Solved	Solved	1972.02	7.26E-04	2.40E-17	9.92E-04	2.56E-14	-1.66E-03	-2.79E-14
arch4	Time	Acc.	3600.09	8.97E-04	4.36E-17	1.71E-03	1.07E-13	-3.49E-03	2.04E-12
arch8	Time	Fail	3600.1	7.42E-01	3.15E-16	4.04E-01	1.09E-11	8.59E-01	5.02E-11
control10	Time	Fail	3600.15	2.68E+02	1.87E-14	1.66E+01	2.71E-10	7.53E-01	6.30E-11
control11	Time	Fail	3600.1	3.91E+02	7.37E-15	1.56E+01	4.98E-10	7.10E-01	-3.44E-11
control1	Time	Fail	3600.09	2.51E-02	1.59E-15	2.13E-03	3.02E-11	2.61E-02	7.64E-12
control2	Time	Fail	3600.08	7.72E-02	1.26E-15	4.74E-03	4.75E-11	1.80E-01	-3.68E-11
control3	Time	Fail	3600.1	2.52E-01	2.52E-15	2.01E-02	1.29E-10	2.76E-01	-8.24E-11

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
control4	Time	Fail	3600.11	7.65E-01	1.31E-15	1.23E-01	1.35E-10	-2.08E-01	-9.61E-11
control5	Time	Fail	3600.08	5.06E-01	2.13E-15	2.47E-01	6.33E-11	-1.80E-01	-4.19E-11
control6	Time	Fail	3600.11	3.35E+00	5.71E-15	9.28E-01	6.64E-11	-1.31E-01	1.82E-10
control7	Time	Fail	3600.09	4.40E+00	2.84E-15	1.39E+00	6.13E-11	2.78E-02	1.06E-10
control8	Time	Fail	3600.07	2.22E+01	1.01E-14	2.64E+00	2.17E-10	3.44E-01	1.98E-10
control9	Time	Fail	3600.11	1.01E+02	1.12E-14	9.62E+00	7.58E-10	6.70E-01	-6.39E-11
equalG11	Time	Fail	3600.42	1.84E-02	6.49E-15	6.86E-05	3.41E-15	-2.24E-03	1.99E-14
equalG51	Time	Fail	3600.64	1.50E-01	1.13E-14	1.45E-04	9.88E-16	-1.85E-03	7.40E-15
gpp100	Solved	Solved	204.7	1.00E-03	2.56E-15	7.94E-04	3.84E-13	-2.96E-04	-7.17E-13
gpp124:1	Solved	Solved	913.79	1.00E-03	2.47E-15	8.19E-04	3.75E-13	-2.93E-03	-1.79E-12
gpp124:2	Solved	Solved	589.44	1.00E-03	3.79E-15	3.74E-04	1.79E-12	-2.00E-04	-1.17E-12
gpp124:3	Solved	Solved	274.82	1.00E-03	3.52E-15	6.37E-04	4.11E-13	-2.03E-04	2.90E-12
gpp124:4	Solved	Solved	363.09	1.00E-03	3.48E-15	5.03E-04	6.80E-12	-3.22E-05	3.17E-13
gpp250:1	Time	Acc.	3600.08	2.34E-03	6.41E-15	1.83E-03	7.19E-13	-6.74E-03	-1.14E-12
gpp250:2	Time	Acc.	3600.11	1.13E-03	7.38E-15	1.31E-03	1.15E-12	-1.18E-03	2.57E-12
gpp250:3	Time	Acc.	3600.13	1.41E-03	4.24E-15	1.44E-03	1.23E-12	-5.64E-04	-6.94E-13

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
gpp250:4	Time	Acc.	3600.15	5.74E-04	5.14E-15	1.10E-03	8.08E-13	-2.37E-04	-1.46E-13
gpp500:1	Time	Fail	3600.15	1.75E-02	5.92E-15	1.03E-02	2.15E-13	-4.66E-02	4.63E-13
gpp500:2	Time	Fail	3600.16	1.37E-02	6.58E-15	1.09E-02	6.04E-13	-1.02E-02	3.47E-13
gpp500:3	Time	Acc.	3600.13	9.21E-03	5.52E-15	6.41E-03	1.07E-12	-2.42E-03	-8.28E-13
gpp500:4	Time	Acc.	3600.2	5.27E-03	6.94E-15	3.87E-03	4.61E-12	-8.08E-04	-1.93E-12
hinf10	Solved	Solved	13.44	1.00E-03	7.19E-15	3.56E-04	3.76E-14	-3.98E-02	-3.21E-13
hinf11	Solved	Solved	13.4	1.00E-03	1.86E-15	1.75E-04	3.66E-15	-4.28E-02	2.57E-13
hinf12	Solved	Solved	18.87	1.00E-03	2.28E-15	3.61E-04	3.35E-15	-1.70E-01	5.53E-12
hinf13	Solved	Solved	3194.3	1.00E-03	2.14E-13	3.84E-04	3.87E-14	-1.24E-01	5.08E-11
hinf14	Solved	Solved	35.45	1.00E-03	3.34E-15	4.29E-04	9.53E-14	-1.85E-02	-3.26E-12
hinf15	Solved	Solved	3008.6	7.45E-04	6.43E-13	7.12E-04	2.08E-14	-2.42E-01	-6.82E-11
hinf1	Solved	Solved	1.28	1.00E-03	1.91E-17	1.49E-04	3.16E-15	-5.71E-03	1.77E-15
hinf2	Solved	Solved	13.61	1.00E-03	4.78E-15	8.03E-04	3.30E-15	-3.12E-02	1.11E-12
hinf3	Solved	Solved	34.5	1.00E-03	1.72E-14	4.56E-04	1.33E-13	-2.73E-02	-1.60E-12
hinf4	Solved	Solved	11.33	1.00E-03	9.66E-16	4.86E-05	1.62E-13	-1.25E-04	-2.77E-13
hinf5	Solved	Solved	534.04	6.96E-04	2.46E-14	8.36E-04	2.52E-14	-1.63E-02	1.16E-11

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
hinf6	Solved	Solved	362.76	1.00E-03	4.46E-15	3.58E-04	1.47E-12	-2.97E-02	-8.64E-11
hinf7	Solved	Solved	22.72	5.00E-04	1.41E-14	7.24E-04	1.04E-15	-3.72E-02	3.99E-13
hinf8	Solved	Solved	162.79	1.00E-03	2.87E-14	5.87E-04	3.63E-13	-4.70E-02	-3.72E-11
hinf9	Solved	Solved	4.95	1.00E-03	9.71E-15	5.20E-04	2.05E-15	-1.60E-02	-5.14E-14
maxG11	Time	Acc.	3600.26	2.06E-05	1.03E-14	6.35E-03	6.53E-14	-6.23E-04	3.77E-15
maxG32	Time	Fail	3601.2	1.16E-03	8.48E-15	2.38E-01	1.67E-13	-1.01E-02	1.03E-15
maxG51	Time	Acc.	3600.68	6.94E-05	1.87E-14	1.27E-03	5.76E-15	-7.67E-04	4.68E-15
maxG55	Time	Fail	3755.08	7.98E-02	4.11E-15	2.68E+00	1.02E-13	-4.01E-02	-8.36E-17
maxG60	Time	Fail	3793.04	2.47E-01	4.53E-15	5.52E+00	1.15E-15	-6.70E-02	1.78E-17
mcp100	Solved	Solved	45.98	9.27E-06	2.07E-15	1.00E-03	0.00E+00	-1.23E-04	7.64E-15
mcp124:1	Solved	Solved	119.83	6.97E-06	2.68E-15	1.00E-03	8.98E-15	-2.01E-04	-2.73E-15
mcp124:2	Solved	Solved	77.45	1.08E-05	2.31E-15	1.00E-03	1.12E-14	-1.41E-04	-1.30E-15
mcp124:3	Solved	Solved	44.8	1.53E-05	2.78E-15	9.99E-04	9.65E-15	-8.80E-05	-2.91E-15
mcp124:4	Solved	Solved	29.7	2.21E-05	4.34E-15	1.00E-03	7.79E-15	-9.19E-05	1.99E-15
mcp250:1	Solved	Solved	500.62	1.79E-05	5.83E-15	1.00E-03	1.36E-14	-1.99E-04	-3.70E-15
mcp250:2	Solved	Solved	345.79	1.61E-05	4.92E-15	1.00E-03	1.63E-14	-1.27E-04	-3.16E-15

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
mcp250:3	Solved	Solved	229.83	2.11E-05	6.18E-15	1.00E-03	5.47E-15	-1.04E-04	3.36E-15
mcp250:4	Solved	Solved	195.74	2.68E-05	3.87E-15	1.00E-03	1.55E-14	-9.30E-05	-3.50E-15
mcp500:1	Solved	Solved	2240.98	5.66E-05	1.96E-14	1.00E-03	2.20E-14	-2.11E-04	-4.38E-15
mcp500:2	Solved	Solved	1664.98	4.47E-05	8.12E-15	1.00E-03	3.20E-14	-1.56E-04	-3.54E-15
mcp500:3	Solved	Solved	1135.73	3.79E-05	9.41E-15	1.00E-03	2.20E-14	-1.02E-04	2.60E-15
mcp500:4	Solved	Solved	802.4	4.63E-05	6.49E-15	1.00E-03	2.27E-14	-9.09E-05	-1.52E-15
qap10	Solved	Solved	84.05	1.00E-03	1.93E-17	6.29E-06	3.74E-15	-5.27E-03	-5.26E-16
qap5	Solved	Solved	0.81	9.99E-04	7.02E-17	1.26E-04	2.37E-16	-2.45E-03	8.16E-18
qap6	Solved	Solved	20.09	1.00E-03	3.58E-17	1.41E-05	2.07E-15	-5.75E-03	-1.12E-15
qap7	Solved	Solved	23.66	1.00E-03	2.32E-17	1.29E-05	1.33E-15	-5.67E-03	-1.85E-16
qap8	Solved	Solved	50.04	1.00E-03	1.91E-17	8.87E-06	2.46E-15	-5.90E-03	1.19E-15
qap9	Solved	Solved	84.46	1.00E-03	1.68E-17	7.88E-06	4.72E-15	-4.08E-03	-8.11E-17
qpG11	Time	Fail	3601.1	2.31E-03	5.79E-15	1.47E-01	1.51E-13	-3.07E-03	1.44E-15
qpG51	Time	Fail	3602.53	5.13E-02	1.03E-13	2.15E+00	7.04E-13	-9.70E-02	-2.97E-14
ss30	Time	Fail	3600.13	2.27E-02	4.81E-15	7.80E-02	7.62E-12	-4.52E-01	-4.88E-11
theta1	Solved	Solved	1.32	1.00E-03	2.11E-16	3.43E-04	6.69E-15	-1.12E-03	-6.32E-16

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
theta2	Solved	Solved	49.87	1.00E-03	1.99E-16	6.32E-05	1.51E-14	-3.20E-03	8.92E-16
theta3	Solved	Solved	128.54	9.99E-04	2.46E-16	4.78E-05	3.01E-14	-2.84E-03	6.12E-17
theta4	Solved	Solved	254.68	1.00E-03	2.38E-16	5.22E-05	3.51E-14	-2.44E-03	1.13E-15
theta5	Solved	Solved	492.43	9.89E-04	2.00E-16	4.48E-04	4.46E-14	-2.51E-03	1.86E-15
theta6	Solved	Solved	716.7	1.00E-03	2.06E-16	4.75E-05	5.66E-14	-2.12E-03	9.31E-16
thetaG11	Time	Fail	3600.34	6.99E-04	1.46E-14	4.09E-02	1.32E-13	-6.06E-03	-4.28E-15
thetaG51	Time	Fail	3601.86	3.92E-01	1.99E-14	1.46E-01	1.86E-13	-8.79E-04	6.08E-13
truss1	Solved	Solved	1.06	9.29E-04	0.00E+00	5.98E-04	4.94E-15	-8.49E-04	-3.47E-15
truss2	Solved	Solved	43.77	7.13E-04	7.43E-04	1.56E-04	1.16E-14	1.00E-03	-2.96E-04
truss3	Solved	Solved	7.54	4.53E-05	3.47E-16	4.41E-04	3.63E-16	-2.00E-04	3.33E-15
truss4	Solved	Solved	1.3	3.22E-04	1.69E-16	4.55E-04	5.28E-15	-2.88E-05	2.11E-15
truss5	Solved	Solved	156.31	8.53E-04	7.99E-04	9.48E-05	2.93E-16	1.00E-03	-2.97E-04
truss6	Solved	Solved	3235.08	1.00E-03	3.05E-14	1.82E-04	3.18E-13	-1.68E-03	3.72E-13
truss7	Solved	Solved	1885.27	1.00E-03	2.24E-14	2.64E-04	1.20E-13	-1.77E-03	3.88E-13
truss8	Solved	Solved	360.05	7.58E-04	7.86E-04	2.03E-04	1.93E-16	7.39E-04	-2.91E-04

A.3.2 SDPAD Results on Mittelmann Library

On the Mittelmann Library problems, SDPAD had a shifted geometric mean of 2028.13 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
1dc.1024	Time	Fail	3601.82	0.312	1.79E-15	0.00122	1.15E-13	0.994	-8.95E-14
1et.1024	Time	Fail	3601.49	0.463	3.24E-15	0.000143	2.43E-13	0.997	-2.15E-13
1tc.1024	Time	Fail	3601.03	0.482	3.32E-15	1.9E-05	2.93E-13	0.997	-2.65E-14
1zc.1024	Time	Fail	3601.87	0.281	2.01E-15	3.32E-07	1.09E-12	0.996	-3.33E-13
Bex2_1_5	Solved	Solved	3563.7	0.00096	9.99E-16	0.000322	1.24E-14	-0.00541	1.96E-13
biggs	Time	Fail	3600.16	5.49	7.73E-10	7.62	9.69E-11	-1	-9E-09
broyden25	Time	Acc.	3600.17	0.00858	4.57E-15	0.000945	1.21E-14	-0.0695	-1.67E-13
Bst_jcbpaf2	Solved	Solved	1420.23	0.000992	4.68E-16	0.000218	9.85E-16	-0.005	1.07E-15
buck3	Time	Fail	3600.26	0.0984	2.9E-14	0.0586	7.64E-16	-0.423	1.88E-14
buck4	Time	Fail	3600.24	0.106	7.66E-15	0.0501	9.76E-16	-0.312	5E-14
buck5	Time	Fail	3601.27	2.2	2.24E-15	0.0646	1.4E-15	-0.346	1.05E-13
butcher	Time	Acc.	3600.1	0.00328	5.28E-15	0.00408	3.52E-16	-0.105	2.63E-15
cancer_100	Time	Acc.	3600.31	0.00235	1.13E-13	0.00123	0	-9.01E-05	1.2E-13

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
checker_1.5	Time	Fail	3606.77	0.352	1.69E-14	0.00733	1.6E-16	-0.0237	4.21E-15
chs_5000	Solved	Solved	39.23	1.88E-05	2.42E-16	5.84E-05	1.67E-16	-7.04E-07	-9.97E-14
cnhil10	Error	Fail							
cnhil8	Error	Fail							
cphil10	Error	Fail							
cphil12	Error	Fail							
e_moment_quadknap_17_100_0.5_2_2	Solved	Solved	213.77	1.87E-05	4.71E-16	0.000708	7.39E-17	-0.000906	2.26E-16
e_moment_stable_17_0.5_2_2	Solved	Solved	1318.54	3.04E-05	1.05E-14	0.000239	2.1E-16	-0.00473	-1.74E-13
foot	Time	Fail	3603.63	1200	7.62E-13	31.7	7.97E-12	-0.0865	-3.2E-14
G40_mb	Time	Fail	3603.52	2.2	9.17E-15	0.167	8.2E-14	-0.0279	6.76E-15
G40mc	Time	Fail	3604.4	0.00359	1.19E-14	0.0994	3.96E-14	-0.0188	1.58E-15

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
G48_mb	Time	Fail	3628.51	0.00552	7.5E-15	1.97	2.52E-13	-0.203	1.29E-14
G48mc	Time	Fail	3621.98	0.0167	9.96E-15	1.43	2.07E-13	-0.0177	-1.71E-15
G55mc	Time	Fail	3735.52	0.155	4.65E-15	2.19	6.69E-14	-0.0358	-3.69E-16
G59mc	Time	Fail	3628.01	0.304	5.05E-15	1.64	3.02E-14	-0.101	-1.32E-15
hand	Time	Fail	3602.28	2.01	1.6E-13	0.607	1.88E-12	-0.00743	4.03E-14
ice_2.0	Time	Fail	3674.2	4.19	9.31E-15	0.0122	7.86E-18	0.387	-6.5E-17
inc_1200	Time	Fail	3602.18	0.00504	2.25E-15	0.0223	3.95E-14	-0.203	2.95E-13
inc_600	Time	Acc.	3600.41	0.00179	3.4E-15	0.00475	2.43E-14	-0.169	1.1E-13
mater:3	Time	Acc.	3600.16	0.00219	1.88E-15	0.000383	3.08E-14	-0.00363	1.19E-13
mater:4	Time	Fail	3600.07	0.116	7.2E-16	0.0199	2.27E-13	-0.133	-1.94E-13
mater:5	Time	Fail	3600.27	0.393	2.01E-16	0.0508	8.56E-15	-0.259	4.43E-16
mater:6	Time	Fail	3600.74	0.908	2.38E-16	0.0222	2.84E-14	-0.29	-2.04E-14
neosfbr25	Error	Fail							
neosfbr30e8	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neu1	Solved	Solved	333.52	0.000338	1.35E-16	0.000889	2.71E-14	-0.379	8.7E-12
neu1g	Solved	Solved	723.49	0.001	5.19E-16	0.000674	1.13E-13	-0.00192	5.98E-13

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
neu2c	Time	Acc.	3600.43	0.00493	1.33E-14	0.00294	4.31E-14	-0.0106	3.09E-12
neu2	Solved	Solved	936.5	0.000149	6.79E-16	0.000721	2.1E-14	-0.0462	-9.83E-13
neu2g	Time	Fail	3600.17	0.00946	3.31E-14	0.0302	5.43E-12	1	5.34E-11
neu3	Solved	Solved	2.47	5.66E-05	0	0.000524	0	0.000396	0
neu3g	Solved	Solved	634.08	6.36E-06	3.15E-16	0.000999	3.96E-17	-0.00315	3.5E-14
nonc_500	Solved	Solved	1102.43	0.000252	2.68E-15	0.000118	2.52E-16	-0.000752	6.06E-14
p_auss2_3.0	Time	Fail	4484.81	6.14	1.11E-14	0.038	1.22E-17	0.093	1.03E-14
rabmo	Solved	Solved	2591.14	0.000776	3.98E-15	0.000766	2.26E-15	-0.0447	-3.94E-15
reimer5	Time	Acc.	3600.21	0.000943	2.07E-15	0.0087	2.75E-16	-0.0795	-6.05E-16
rendl1_2000_1e:6	Time	Fail	3610.7	94.4	9.65E-14	0.00184	5.73E-15	0.00577	8.47E-16
ros_2000	Time	Acc.	3600.32	0.00378	1.47E-15	0.000253	9.57E-16	-0.153	-4.33E-14
ros_500	Solved	Solved	200.75	0.000998	1.15E-15	9.42E-05	8.52E-16	-0.00308	-1.82E-14
rose13	Solved	Solved	190.21	9.4E-06	3.32E-16	0.001	2.78E-15	-0.00446	1.71E-12
rose15	Solved	Solved	358.88	1.03E-05	7.38E-16	0.000997	2.93E-15	-0.11	5.18E-11
sensor_1000	Time	Fail	3600.7	0.00742	8.3E-15	0.0186	9.08E-14	-0.376	1.16E-12

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
sensor_500	Solved	Solved	590.18	0.000998	2.94E-15	0.000585	1.13E-13	-0.000495	-3.65E-13
shmup3	Time	Fail	3602.44	0.00519	6.96E-15	0.0124	2.43E-16	-0.382	1.07E-14
shmup4	Time	Fail	3602.33	0.0967	8.96E-15	0.0138	1.26E-16	-0.456	-2.65E-15
shmup5	Time	Fail	3711.79	0.439	1.02E-15	0.066	1.02E-16	-0.639	-2.82E-13
spar060:020:1_LS	Solved	Solved	90.46	0.00078	1.48E-15	0.000387	1.51E-15	-3.75E-05	-1.04E-15
swissroll	Time	Fail	3600.63	0.026	1.72E-14	1.68	8.27E-12	-0.472	-5.08E-13
taha1a	Solved	Solved	461.23	1.69E-05	2.01E-16	0.000282	6.05E-16	-0.000223	-1.26E-15
taha1b	Solved	Solved	2648.16	0.000595	5.8E-16	0.000238	1.46E-15	-0.0111	-9.03E-16
taha1c	Solved	Solved	1153.1	2.85E-05	4.15E-16	0.000285	4.25E-16	-0.000185	-3.47E-15
theta12	Time	Acc.	3600.57	0.00235	2.2E-16	7.41E-05	1.06E-13	-0.000994	-2.36E-15
tiger_texture	Time	Fail	3601.27	34.8	1.6E-14	0.152	2.49E-13	-0.287	-2.66E-13
trto3	Time	Fail	3600.23	0.0383	8.65E-14	0.0628	2.37E-16	-0.647	-7.37E-14
trto4	Time	Fail	3600.81	0.0629	1.98E-14	0.0492	9.23E-17	-0.589	3.79E-14
trto5	Time	Fail	3600.88	0.397	1.65E-15	0.0464	2.64E-16	-0.309	5.98E-14
vibra3	Time	Fail	3600.31	0.0303	9.52E-15	0.0404	1.7E-15	-0.34	-3.21E-14

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
vibra4	Time	Fail	3600.75	0.0405	1.14E-14	0.0337	3.47E-16	-0.302	2.75E-16
vibra5	Time	Fail	3602.36	1.29	4.05E-15	0.0492	9.31E-16	-0.311	-1.31E-14
yalsdp	Solved	Solved	665.6	0.000968	2.33E-16	0.000672	7.27E-15	-0.000771	2.19E-14

A.4 SCS Results

SCS had an overall geometric mean of 513.23 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

A.4.1 SCS Results on SDPLIB

On the SDPLIB problems, SCS had a geometric mean of 189.01 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
arch0	Solved	Solved	385.26	7.17E-05	1.73E-19	1.09E-07	3.26E-22	-4.69E-09	3.84E-13
arch2	Solved	Solved	444.52	6.94E-05	2.76E-19	1.20E-07	7.70E-22	-7.38E-10	3.61E-13
arch4	Solved	Solved	668.7	6.33E-05	1.94E-19	1.32E-07	2.82E-11	-2.10E-09	-5.07E-13

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
arch8	Solved	Solved	1566.79	5.24E-05	3.46E-19	8.06E-08	1.48E-11	5.01E-10	-3.15E-13
control10	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control11	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control1	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control2	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control3	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control4	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control5	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control6	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control7	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control8	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
control9	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
equalG11	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
equalG51	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
gpp100	Solved	Solved	8.32	1.96E-05	3.68E-15	1.78E-05	5.26E-10	1.50E-06	-7.20E-10
gpp124:1	Solved	Solved	48.44	5.44E-05	3.30E-15	2.69E-05	2.30E-09	6.28E-07	2.27E-07

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
gpp124:2	Solved	Solved	11.38	4.37E-05	2.50E-15	3.05E-05	6.26E-10	4.25E-06	2.98E-10
gpp124:3	Solved	Solved	11.35	2.97E-05	3.68E-15	3.87E-05	6.60E-10	-6.85E-07	-3.84E-10
gpp124:4	Solved	Solved	17.19	5.01E-05	1.80E-15	4.12E-05	2.74E-09	2.19E-06	-1.57E-09
gpp250:1	Solved	Solved	139.07	4.83E-05	4.76E-15	3.71E-05	1.10E-09	3.28E-06	2.22E-10
gpp250:2	Solved	Solved	85.13	7.44E-05	5.47E-15	5.84E-05	5.91E-10	5.59E-06	-1.91E-09
gpp250:3	Solved	Solved	84.43	7.02E-05	6.75E-15	7.33E-05	6.50E-10	4.81E-06	-6.36E-10
gpp250:4	Solved	Solved	83.6	7.09E-05	4.11E-15	8.63E-05	5.00E-10	5.36E-07	2.83E-10
gpp500:1	Solved	Acc.	1256.39	1.16E-04	1.76E-14	5.56E-05	3.88E-10	2.65E-06	-5.71E-09
gpp500:2	Solved	Solved	708.36	8.57E-05	1.05E-14	4.67E-05	8.01E-10	-2.42E-07	2.01E-09
gpp500:3	Solved	Acc.	376.1	1.09E-04	1.13E-14	7.16E-05	1.00E-10	-1.78E-07	-1.04E-10
gpp500:4	Solved	Acc.	411.73	1.12E-04	8.06E-15	9.60E-05	6.23E-11	-1.68E-07	3.80E-12
hinf10	Solved	Solved	1503.47	6.32E-06	4.90E-16	1.74E-05	3.00E-06	-1.69E-06	6.83E-05
hinf11	Solved	Solved	97.38	1.00E-05	3.33E-16	1.24E-05	1.11E-05	1.17E-07	-4.24E-05
hinf12	Solved	Solved	0.41	3.70E-07	3.20E-18	1.19E-05	0.00E+00	2.23E-06	8.49E-05
hinf13	Solved	Acc.	2866.07	9.77E-06	1.05E-13	1.94E-05	0.00E+00	2.37E-06	4.20E-04
hinf14	Solved	Solved	2061.73	9.12E-06	4.29E-14	1.37E-05	9.61E-08	6.73E-07	-2.91E-06

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
hinf15	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
hinf1	Solved	Fail	8.01	4.15E-06	2.53E-17	1.04E-05	2.88E-02	1.50E-06	-2.43E-03
hinf2	Solved	Solved	0.6	5.63E-06	7.11E-15	8.01E-06	2.21E-08	1.74E-07	-2.09E-07
hinf3	Solved	Solved	1.27	4.22E-06	1.72E-14	6.11E-06	7.93E-08	3.21E-07	5.98E-06
hinf4	Solved	Solved	11.35	7.21E-06	3.41E-16	8.12E-06	1.23E-05	1.26E-09	3.36E-06
hinf5	Solved	Solved	5.54	7.72E-06	1.33E-13	9.39E-06	8.83E-07	1.46E-07	-6.76E-05
hinf6	Solved	Solved	139.9	8.79E-06	1.87E-14	5.35E-06	4.07E-07	1.39E-07	-1.63E-05
hinf7	Solved	Fail	0.8	9.72E-06	2.27E-12	1.68E-05	9.64E-06	-3.41E-08	1.17E-02
hinf8	Solved	Solved	14.37	5.81E-06	4.74E-14	3.91E-06	1.14E-07	5.93E-08	-1.62E-06
hinf9	Solved	Solved	27.08	8.08E-06	1.92E-12	1.73E-05	5.02E-06	1.51E-07	-3.45E-02
maxG11	Solved	Acc.	1184.37	1.10E-04	7.64E-14	3.09E-07	5.57E-12	-2.26E-07	7.48E-13
maxG32	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
maxG51	Solved	Solved	175.4	6.89E-05	2.90E-14	6.26E-08	2.35E-12	-1.56E-07	-3.56E-12
maxG55	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
maxG60	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
mcp100	Solved	Solved	1	4.17E-05	2.31E-15	1.14E-06	1.53E-11	-1.13E-06	7.28E-12

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
mcp124:1	Solved	Solved	3.27	2.70E-05	4.47E-15	5.01E-07	1.96E-11	-1.50E-06	6.06E-12
mcp124:2	Solved	Solved	1.91	1.78E-05	3.10E-15	3.22E-07	1.22E-11	-1.10E-06	1.70E-12
mcp124:3	Solved	Solved	1.67	1.52E-05	2.25E-15	3.03E-07	1.76E-11	3.90E-07	1.02E-11
mcp124:4	Solved	Solved	1.28	4.04E-05	2.68E-15	1.32E-06	2.85E-11	-6.19E-07	6.97E-12
mcp250:1	Solved	Solved	16.25	2.95E-05	6.49E-15	2.43E-07	1.34E-11	-8.57E-08	-2.77E-12
mcp250:2	Solved	Solved	8.85	3.68E-05	3.37E-15	3.86E-07	1.36E-11	9.08E-07	-5.68E-12
mcp250:3	Solved	Solved	6.34	5.85E-05	2.99E-15	7.59E-07	1.93E-11	-1.34E-06	-1.27E-12
mcp250:4	Solved	Solved	5.18	3.66E-05	6.41E-15	5.92E-07	2.71E-11	9.43E-07	5.75E-12
mcp500:1	Solved	Solved	177.41	6.29E-05	1.47E-14	2.30E-07	7.55E-12	3.51E-07	1.16E-14
mcp500:2	Solved	Solved	53.05	7.63E-05	7.28E-15	5.36E-07	1.16E-11	-9.40E-07	-2.76E-12
mcp500:3	Solved	Solved	35.45	7.19E-05	7.05E-15	4.00E-07	1.45E-11	9.34E-07	-6.77E-13
mcp500:4	Solved	Solved	38.33	4.25E-05	1.38E-14	2.60E-07	1.48E-11	4.67E-07	-1.99E-12
qap10	Solved	Acc.	82.13	1.18E-05	2.18E-18	1.92E-04	4.96E-09	-6.21E-06	-1.51E-10
qap5	Solved	Solved	0.33	2.64E-06	3.42E-17	4.07E-06	0.00E+00	1.22E-07	3.95E-11
qap6	Solved	Solved	16.99	1.40E-05	7.31E-18	5.84E-05	6.20E-09	-9.92E-06	-1.55E-10
qap7	Solved	Solved	22.91	1.27E-05	6.27E-18	9.21E-05	4.00E-09	3.73E-07	4.74E-10

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
qap8	Solved	Acc.	41.68	7.42E-06	5.08E-18	1.07E-04	4.58E-09	-7.95E-06	7.50E-10
qap9	Solved	Acc.	62.85	1.33E-05	4.31E-18	1.35E-04	3.75E-09	2.95E-07	-2.69E-10
qpG11	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
qpG51	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
ss30	Solved	Solved	2023.58	7.49E-05	3.10E-17	4.81E-08	6.55E-12	2.00E-07	3.70E-12
theta1	Solved	Solved	0.89	4.00E-06	3.10E-17	3.83E-05	8.61E-10	1.54E-06	2.02E-11
theta2	Solved	Solved	15.22	9.31E-06	3.64E-17	8.79E-05	6.49E-10	-1.92E-07	5.36E-12
theta3	Solved	Acc.	22.82	5.72E-06	1.92E-17	1.99E-04	8.75E-10	1.70E-06	-1.58E-12
theta4	Solved	Acc.	56.35	7.16E-06	1.13E-17	2.97E-04	9.79E-10	-6.50E-06	-3.90E-12
theta5	Solved	Acc.	83.55	9.55E-06	1.11E-17	2.60E-04	1.12E-09	2.23E-06	1.43E-12
theta6	Solved	Solved	136.85	1.70E-06	1.47E-17	6.47E-05	1.25E-09	-1.96E-06	4.84E-12
thetaG11	Solved	Acc.	2619.85	1.98E-04	1.51E-13	7.54E-07	0.00E+00	4.19E-08	3.22E-11
thetaG51	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
truss1	Solved	Solved	0.27	7.81E-07	1.85E-17	2.69E-07	1.60E-16	-1.07E-08	-1.51E-16
truss2	Solved	Solved	1.18	1.44E-05	2.49E-15	1.21E-06	5.64E-09	2.01E-08	-1.29E-09
truss3	Solved	Solved	0.72	6.90E-06	2.07E-16	2.43E-06	4.67E-10	1.89E-07	-1.08E-10

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
truss4	Solved	Solved	0.28	1.12E-07	1.00E-16	3.32E-08	7.62E-10	5.90E-09	-2.39E-10
truss5	Solved	Solved	568.42	1.46E-05	1.08E-16	6.44E-07	2.44E-09	1.06E-08	9.77E-11
truss6	Solved	Solved	657.07	9.31E-06	1.21E-17	2.34E-06	4.98E-08	-3.73E-10	5.62E-09
truss7	Solved	Solved	0.64	7.87E-06	9.25E-18	1.58E-06	4.96E-14	-1.96E-10	1.21E-15
truss8	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN

A.4.2 SCS Results on Mittelmann Library

On the Mittelmann Library problems, SCS had a geometric mean of 1667.92 CPU seconds, with shift 10 s, cut off at 1 hour, treated as taking 1 hour for failures, including having any error worse than 10^{-2}.

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
1dc.1024	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
1et.1024	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
1tc.1024	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
1zc.1024	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
Bex2_1_5	Infeasible	Fail	23.23	8.22E-01	0.00E+00	NaN	NaN	NaN	NaN

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
biggs	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
broyden25	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
Bst_jcbpaf2	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
buck3	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
buck4	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
buck5	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
butcher	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
cancer_100	Solved	Acc.	2048.5	6.38E-04	9.75E-14	2.20E-05	1.67E-07	-7.55E-07	1.09E-08
checker_1.5	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
chs_5000	Solved	Solved	8.77	9.76E-06	5.05E-17	6.21E-06	5.09E-11	-4.10E-06	-3.44E-10
cnhil10	Solved	Solved	87.08	2.37E-05	9.90E-17	6.13E-12	1.17E-18	-1.16E-13	-7.08E-18
cnhil8	Solved	Solved	29.04	2.32E-05	8.59E-17	8.89E-12	2.48E-18	-2.59E-14	1.49E-17
cphil10	Solved	Solved	2.39	1.27E-10	0.00E+00	3.69E-17	8.29E-19	-1.05E-17	-2.04E-18
cphil12	Solved	Solved	6.75	5.86E-09	0.00E+00	1.90E-15	7.43E-19	-2.15E-16	-4.32E-19

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
e_moment_quadknap_17_100_0.5_2_2	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
e_moment_stable_17_0.5_2_2	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
foot	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
G40_mb	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
G40mc	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
G48_mb	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
G48mc	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
G55mc	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
G59mc	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
hand	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
ice_2.0	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
inc_1200	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
inc_600	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
mater:3	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
mater:4	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
mater:5	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
mater:6	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neosfbr25	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neosfbr30e8	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neu1	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neu1g	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neu2c	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neu2	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neu2g	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
neu3	Solved	Solved	0.99	1.64E-13	0.00E+00	5.98E-14	6.33E-18	-5.34E-14	3.38E-16
neu3g	Solved	Solved	942.64	1.38E-05	1.52E-16	1.02E-06	3.12E-13	-2.61E-09	-1.23E-12
nonc_500	Solved	Solved	231.22	5.44E-05	6.21E-17	4.69E-05	4.90E-11	-7.78E-06	7.14E-10

problem	reported	status	time (s)	err_1	err_2	err_3	err_4	err_5	err_6
p_auss2_3.0	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
rabmo	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
reimer5	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
rendl1_2000_1e:6	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
ros_2000	Solved	Solved	98.83	6.84E-06	8.61E-18	4.15E-05	1.94E-10	3.08E-06	-8.65E-11
ros_500	Solved	Solved	18.77	6.49E-05	7.32E-17	3.46E-05	1.68E-10	6.25E-06	-9.12E-11
rose13	Solved	Solved	1265.64	4.06E-05	1.36E-16	3.71E-06	1.31E-10	-2.22E-06	5.47E-09
rose15	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
sensor_1000	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
sensor_500	Solved	Solved	1595.92	1.63E-05	5.30E-16	9.68E-06	4.71E-11	-1.08E-07	-1.33E-12
shmup3	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
shmup4	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
shmup5	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
spar060:020:1_LS	Solved	Solved	1193.32	6.87E-06	6.90E-16	6.06E-05	6.87E-10	7.07E-08	-9.56E-11

problem	reported	status	time (s)	err$_1$	err$_2$	err$_3$	err$_4$	err$_5$	err$_6$
swissroll	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
taha1a	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
taha1b	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
taha1c	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
theta12	Solved	Acc.	427.51	9.35E-06	2.10E-17	2.78E-04	1.04E-09	5.21E-06	-6.55E-13
tiger_texture	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
trto3	Solved	Solved	130.69	1.56E-07	3.83E-18	1.23E-10	3.67E-12	9.53E-10	-2.71E-13
trto4	Solved	Solved	1147.15	2.11E-07	6.77E-18	5.56E-11	1.82E-12	-4.02E-09	2.51E-12
trto5	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
vibra3	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
vibra4	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
vibra5	Time	Fail	3600	NaN	NaN	NaN	NaN	NaN	NaN
yalsdp	Solved	Solved	121.86	3.41E-06	4.54E-17	4.66E-05	4.62E-11	-1.85E-06	4.43E-11

REFERENCES

[1] (2000). Seventh DIMACS Implementation Challenge: Semidefinite and Related Optimization Problems. http://dimacs.rutgers.edu/archive/Challenges/Seventh/.

[2] Borchers, B. (1999a). CSDP, A C library for semidefinite programming. *Optimization Methods and Software*, 11(1-4):613–623.

[3] Borchers, B. (1999b). SDPLIB 1.2, library of semidefinite programming test problems. *Optimization Methods and Software*, 11:683–690. http://euler.nmt.edu/~brian/sdplib/.

[4] Boyd, S. (2002). Advances in Convex Optimization: Interior-point Methods, Cone Programming, and Applications. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 4, Las Vegas. IEEE.

[5] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.

[6] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[7] Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

[8] Gould, N. and Scott, J. (2016). A Note on Performance Profiles for Benchmarking Software. *ACM Transactions on Mathematical Software*, 43(2):1–5.

[9] Malick, J., Povh, J., Rendl, F., and Wiegele, A. (2009). Regularization Methods for Semidefinite Programming. *SIAM Journal on Optimization*, 20(1):336–356.

[10] Mittelmann, H. (2003). An independent benchmarking of SDP and SOCP solvers. *Mathematical Programming, Series B*, 95(2):407–430.

[11] Mittelmann, H. (2017). Sparse and other SDP problems. http://plato.asu.edu/ftp/sdp/.

[12] O'Donoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068.

[13] Pataki, G. (2003). Foreword. *Mathematical Programming*, 95(2):187–188.

[14] Povh, J., Rendl, F., and Wiegele, A. (2006). A Boundary Point Method to Solve Semidefinite Programs. *Computing*, 78(3):277–286.

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

[16] Themelis, A. and Patrinos, P. (2019). SuperMann: A Superlinearly Convergent Algorithm for Finding Fixed Points of Nonexpansive Operators. *IEEE Transactions on Automatic Control*, 64(12):4875–4890.

[17] Tütüncü, R. H., Toh, K.-C., and Todd, M. J. (2003). Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2):189–217.

[18] Vandenberghe, L. and Boyd, S. (1996). Semidefinite Programming. *SIAM Review*, 38(1):49–95.

[19] Wen, Z., Goldfarb, D., and Yin, W. (2010). Alternating direction augmented Lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2(3):203–230.

[20] Yamashita, M., Fujisawa, K., Nakata, K., Nakata, M., Fukuda, M., Kobayashi, K., and Goto, K. (2010). A high-performance software package for semidefinite programs: SDPA 7.

A COMPARISON OF THE PERFORMANCE OF SDP SOLVERS

by

Celia Flicker

NEW MEXICO TECH
SCIENCE • ENGINEERING • RESEARCH UNIVERSITY