Mathematical Programming Computation
https://doi.org/10.1007/s12532-026-00320-7

FULL LENGTH PAPER

Clarabel: An interior-point solver for conic programs with quadratic objectives

Paul J. Goulart[1] · **Yuwen Chen**[1]

Received: 23 May 2024 / Accepted: 3 April 2026

Abstract

We present a general-purpose interior-point solver for convex optimization problems with conic constraints. Our method is based on a homogeneous embedding method originally developed for general monotone complementarity problems and more recently applied to operator splitting methods, and here specialized to an interior-point method for problems with quadratic objectives. We allow for a variety of standard symmetric and non-symmetric cones, and provide support for chordal decomposition methods in the case of semidefinite cones. We describe the implementation of this method in the open-source solver CLARABEL, and provide a detailed numerical evaluation of its performance versus several state-of-the-art solvers on a wide range of standard benchmark problems. CLARABEL is faster than competing commercial and open-source solvers across a range of test sets with quadratic objectives, and remains competitive for problems with linear objectives even at large scale. CLARABEL is currently distributed as a default solver for the Python CVXPY optimization suite.

1 Introduction

We consider throughout the following convex conic optimization problem:

$$\begin{aligned} \min_{x,s} \quad & \tfrac{1}{2}x^\top P x + q^\top x \\ \text{subject to:} \quad & Ax + s = b \\ & s \in \mathcal{K}, \end{aligned} \tag{$\mathcal{P}$}$$

with decision variables $x \in \mathbb{R}^n$ and $s \in \mathbb{R}^m$, and problem data $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $q \in \mathbb{R}^n$ and $P \in \mathbb{R}^{n \times n}$. We assume that P is symmetric and positive semidefinite

✉ Paul J. Goulart
paul.goulart@eng.ox.ac.uk

Yuwen Chen
yuwen.chen@eng.ox.ac.uk

1 University of Oxford, Parks Road, Oxford OX1 3PJ, UK

Published online: 21 May 2026

(possibly zero) and that the set $\mathcal{K}$ is a closed and convex cone. We will denote the optimal value of this problem as p^* and an optimizer (when it exists) as (x^*, s^*).

It can be shown that the problem dual to $\mathcal{P}$ is

$$\begin{aligned} \max_{x,z} \quad & -\tfrac{1}{2}x^\top P x - b^\top z \\ \text{subject to:} \quad & Px + A^\top z = -q, \\ & z \in \mathcal{K}^* \end{aligned} \qquad (\mathcal{D})$$

where $\mathcal{K}^*$ is the dual cone of $\mathcal{K}$. We will denote its optimal value as d^* and an optimizer (when it exists) as (x^*, z^*). We will assume for convenience throughout that A is full row rank and strong duality holds between $\mathcal{P}$ and $\mathcal{D}$, i.e. we assume $p^* = d^*$ and satisfaction of Slater's condition.

The problem $\mathcal{P}$ is very general and can model most convex optimization problems of practical interest. Such problems appear in a huge range of applications including constrained optimal control [1] and moving horizon estimation [2], limit analysis of engineering structures [3, 4] and fluid flows [5], image processing [6], support vector machines [7], lasso problems [8, 9], circuit design [10], portfolio optimization [11, 12], and many others.

The numerical solution of convex problems in the form $\mathcal{P}$ also underpins many nonconvex optimization methods, including those based on sequential quadratic programming [13, Ch. 18][14] and branch-and-bound methods within mixed-integer methods [15, 16].

1.1 Solution methods

Considerable research effort has focused on the development of solution techniques for $\mathcal{P}$ over many decades. If the cone $\mathcal{K}$ is the nonnegative orthant, then $\mathcal{P}$ represents a linear program (LP) if $P = 0$ or a quadratic program (QP) otherwise. This case in particular has a long history, dating to the landmark work of Kantorovich [17] and Koopmans [18, 19] in the LP case, and Frank and Wolfe in the QP case [20]. Of particular significance was the development of numerical solution methods for such problems, and in particular *simplex* or active-set methods for the solution of LP [21] and QP [22] instances. Active set methods still form the basis of many modern LP and QP solvers, including commercial solvers such as Gurobi [23] and open-source solvers such as qpOASES [24] and HiGHS [25].

However, since active set methods are based on the iterative updating of a basis of active constraints and rely heavily on the polyhedral structure of systems of linear equalities and inequalities, they do not generalize well to the case of non-polyhedral constraints. Recent advances for more general *conic programs*, i.e. optimization problems where the set $\mathcal{K}$ is a more general convex cone, have therefore focussed instead on either operator splitting methods such as the alternating direction method of multipliers (ADMM) [26] or on interior-point methods [27]. ADMM methods in particular are attractive in cases where only moderate accuracy is required, since ADMM is known to be prone to poor 'tail convergence' at high precision [28]. Nevertheless, ADMM based methods are widely used in practice for both their scalability and simplicity of implementation and form the basis of several open-source solvers for $\mathcal{P}$, including the open source solvers OSQP [29], SCS [30] and COSMO [31].

Primal-dual interior-point methods In this work we focus instead on primal-dual interior-point methods [32], since such methods are known to perform better than operator splitting methods at high precision and, unlike active set methods, can support general conic constraints. A significant milestone in the development of solution algorithms for $\mathcal{P}$ was Karmarkar's method [33], which provided the first practically useful polynomial-time solution method for linear programs, followed by the path following method of Renegar [34].

An advantage of interior-point methods is that they can be extended to the case of general conic constraints $\mathcal{K}$ in $\mathcal{P}$. Of particular importance in the development of efficient conic interior-point solvers was the development of solvers based on cones and associated barrier functions that have the *self-scaling* property [35, 36]. The family of self-scaled cones include those that are generally considered the most important constraint types in practice, i.e. the nonnegative orthant, second-order cone and positive semidefinite (PSD) cone. Conic problems defined over self-scaled cones can be solved efficiently using primal-dual interior-point methods based on the *Nesterov-Todd (NT) scaling* strategy [35, 36].

Nonsymmetric cones that do *not* have the self-scaling property have been the subject of increasing recent interest for interior-point methods [37, 38]. The exponential cone and the power cone are the two most studied and are supported in commercial solvers such as Mosek [39]. In certain cases it is also attractive to formulate problems in terms of more exotic nonsymmetric cones that can be solved more efficiently by exploiting their special structure [40–42]. Since the NT scaling method does not apply to nonsymmetric cones, alternative strategies have been proposed based on either primal or dual iterates [41, 43, 44] or both primal and dual iterates together [39, 42].

Alongside these developments, a further innovation was the development of the *homogeneous self-dual embedding* (HSDE) [45] for conic problems. This problem reformulation method embeds a conic program with linear objectives into a slightly larger feasibility problem, which is *always* feasible regardless of the feasibility of the original problem. Once a feasible point is found, it can be used to construct either an optimizer for the original problem or a primal or dual certificate of infeasibility.

The combination of the HSDE and interior-point numerical methods is at the heart of many modern numerical optimization codes including CVXOPT [46], ECOS [47] and MOSEK [48]. A downside of this approach is that it does not naturally support problems with quadratic objectives, since the standard embedding method applied to such cases does not produce a self-dual problem.

Monotone complementarity problems (MCPs)

Extensions of the HSDE method were quickly developed for the linear complementarity problem (LCP) [49], and later for the more general monotone complementarity problem (MCP) [50] with linear constraints. The homogeneous embedding for MCPs was also extended to problems defined over more general symmetric cone constraints [51, 52].

The practical performance of the homogeneous embedding in the case of convex quadratically constrained quadratic programming (QCQP) problems was investigated in [53]. More recently, [54] proposed a maximal monotone operator specialized for the LCP based on the monotone operator in [50] to produce a first-order solver for problems with convex quadratic objectives and general conic constraints.

2 Algorithmic approach

In this paper we adopt the approach of [53, 54] by specializing the homogeneous embedding method of [50] to convex problems with quadratic objectives. We will use this embedding as the basis for a general purpose primal-dual interior-point solver that handles both symmetric and nonsymmetric cone constraints.

The standard Karush-Kuhn-Tucker (KKT) conditions for problem $\mathcal{P}$ are

$$\begin{aligned} Ax + s &= b \\ Px + A^\top z &= -q \\ s^\top z &= 0 \\ (x, s, z) &\in \mathbb{R}^n \times \mathcal{K} \times \mathcal{K}^*. \end{aligned} \tag{1}$$

We define the duality gap γ as

$$\begin{aligned} \gamma &:= \left(\frac{1}{2}x^\top Px + q^\top x\right) - \left(-\frac{1}{2}x^\top Px - b^\top z\right) \\ &= x^\top Px + q^\top x + b^\top z = s^\top z, \end{aligned}$$

where the final equality follows from substitution in the KKT conditions (1). The duality gap γ is nonnegative for any feasible point since (s, z) are constrained to the dual pair of cones $(\mathcal{K}, \mathcal{K}^*)$, and is zero at an optimal point (x^*, s^*, z^*) when strong duality holds.

Taking problems $\mathcal{P}$ and $\mathcal{D}$ together with an objective minimizing the duality gap, we can rewrite our problem in the form

$$\begin{aligned} \min_{(x,s,z)} \quad & s^\top z \\ \text{subject to:} \quad & x^\top Px + q^\top x + b^\top z = 0 \\ & Px + A^\top z + q = 0 \\ & Ax + s - b = 0 \\ & (x, s, z) \in \mathbb{R}^n \times \mathcal{K} \times \mathcal{K}^*. \end{aligned} \tag{2}$$

This problem is of course infeasible whenever either $\mathcal{P}$ or $\mathcal{D}$ is infeasible or the pair is not strongly dual.

If we define nonnegative scalars (τ, κ) and apply a change of variables $x \to x/\tau$, $z \to z/\tau$ and $s \to s/\tau$, then we can rewrite the feasibility problem (2) as

$$\begin{array}{ll} \min\limits_{(x,s,z,\tau,\kappa)} & s^\top z + \tau\kappa \\ \text{subject to:} & \frac{1}{\tau}x^\top P x + q^\top x + b^\top z = -\kappa \\ & Px + A^\top z + q\tau = 0 \\ & Ax + s - b\tau = 0 \\ & (x, s, z, \tau, \kappa) \in \mathbb{R}^n \times \mathcal{K} \times \mathcal{K}^* \times \mathbb{R}_+ \times \mathbb{R}_+. \end{array} \tag{$\mathcal{H}$}$$

The problem $\mathcal{H}$ is equivalent to the widely-used homogeneous self-dual embedding (HSDE) [55] in the special case $P = 0$. The most common approach to solving $\mathcal{P}$ is therefore to eliminate the quadratic term in the objective function, replacing it with an epigraphical upper bound and an additional second-order cone constraint in the objective [56]. This amounts to rewriting $\mathcal{P}$ as

$$\begin{array}{ll} \min\limits_{x,s,w} & w + q^\top x \\ \text{subject to:} & Ax + s = b \\ & \left\| \begin{bmatrix} 2P^{\frac{1}{2}}x \\ w - 2 \end{bmatrix} \right\|_2 \leq w + 2 \\ & s \in \mathcal{K},\ \ w \geq 0, \end{array} \tag{3}$$

which is a conic optimization problem with a purely linear cost and an additional second-order cone constraint. Additional care is needed in such a formulation if P is rank deficient, since in such cases regularization may be required to avoid loss of definiteness in floating point arithmetic.

If we instead keep the quadratic term in the objective then $\mathcal{H}$ remains homogeneous but is no longer self-dual, in addition to featuring the seemingly awkward $\frac{1}{\tau}x^\top Px$ term in the first equality. Our approach will nevertheless amount to a direct solution of $\mathcal{H}$ using a primal-dual type interior-point method, and we will show that significant performance improvements are possible with this approach in many cases relative to standard HSDE-based interior-point methods.

2.1 Existence of solutions and infeasibility detection

Before proceeding to the details of our algorithm, we address briefly the existence and interpretation of solutions to $\mathcal{H}$. As in the case of the HSDE, an advantage of our reformulation is that solutions will produce either a solution to the original problem $\mathcal{P}$ or $\mathcal{D}$, or (asymptotically) a certificate of either primal or dual infeasibility. We first define the following sets of infeasibility certificates for $\mathcal{P}$ and $\mathcal{D}$:

$$\mathbb{P} := \left\{ \bar{x} \;\middle|\; P\bar{x} = 0,\ A\bar{x} \in -\mathcal{K},\ \ q^\top \bar{x} < 0 \right\} \tag{4}$$

$$\mathbb{D} := \left\{ \bar{z} \;\middle|\; A^\top \bar{z} = 0, \quad \bar{z} \in \mathcal{K}^*,\ b^\top \bar{z} < 0 \right\}. \tag{5}$$

The problem $\mathcal{P}$ is infeasible if and only if there exists some $\bar{z} \in \mathbb{D}$, while the problem $\mathcal{D}$ is infeasible if and only if there exists some $\bar{x} \in \mathbb{P}$. Similar conditions have appeared in [31, 54, 57] in the context of conic optimization with quadratic objectives.

Solving the problem $\mathcal{H}$ amounts to finding a root of the nonlinear system of equations

$$G(x, z, s, \tau, \kappa) := \begin{bmatrix} 0 \\ s \\ \kappa \end{bmatrix} - \begin{bmatrix} P & A^\top & q \\ -A & 0 & b \\ -q^\top & -b^\top & 0 \end{bmatrix} \begin{bmatrix} x \\ z \\ \tau \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} x^\top P x \end{bmatrix} = 0 \tag{6}$$

subject to the conic constraint

$$(x, z, s, \tau, \kappa) \in \mathcal{C} := \mathbb{R}^n \times \mathcal{K}^* \times \mathcal{K} \times \mathbb{R}_+ \times \mathbb{R}_+.$$

These conditions are identical to those in [54, (4.1)], where they were developed in terms of specialization of the homogeneous embedding of monotone complementarity problems developed in [50] to an LCP. See also [51, 52] for a theoretical analysis of interior point methods applied to a general form of such an embedding for symmetric cones.

We can define a more compact notation $v := (x, z, s, \tau, \kappa)$ for convenience, so that $\mathcal{H}$ is equivalent to

$$\begin{aligned} \min_{v} \quad & s^\top z + \tau\kappa \\ \text{subject to:} \quad & G(v) = 0, \quad v \in \mathcal{C}. \end{aligned} \tag{7}$$

As in [50], we say that the problem $\mathcal{H}$ (equivalently (7)) is *asymptotically feasible* if there exists a bounded sequence $\{v^k\} \subset \mathcal{C}$, $k = 1, 2, \ldots,$ such that

$$\lim_{k \to \infty} G(v^k) = 0,$$

and call any limit point $\hat{v} := (\hat{x}, \hat{z}, \hat{s}, \hat{\tau}, \hat{\kappa})$ of such a sequence an *asymptotically feasible point*. We will call any such point with $\hat{s}^\top \hat{z} + \hat{\tau}\hat{\kappa} = 0$ an *(asymptotically) complementary* or optimal solution.

We can now develop some basic results about solutions to the problem $\mathcal{H}$ in the spirit of [50, Thm. 1]:

Lemma 2.1 *The problem $\mathcal{H}$ is (asymptotically) feasible and every (asymptotically) feasible point is (asymptotically) complementary.*

Proof e can construct an asymptotically feasible sequence by defining $x^k = (\frac{1}{2})^k \mathbf{1} \frac{1}{\sqrt{n}}$, $s^k = (\frac{1}{2})^k e_{\mathcal{K}}$, $z^k = (\frac{1}{2})^k e_{\mathcal{K}^*}$, $\tau^k = (\frac{1}{2})^k$ and $\kappa^k = (\frac{1}{2})^k$, where $(e_{\mathcal{K}}, e_{\mathcal{K}^*})$ is any vector pair in $\mathcal{K} \times \mathcal{K}^*$. It is then straightforward to show that $G(v^k) \to 0$, noting in particular that the sequence

$$\frac{1}{\tau^k}(x^k)^\top P x^k \le (\tfrac{1}{2})^k \bar{\sigma}(P) \to 0$$

where $\bar{\sigma}(P)$ is the maximum singular value of P, and is lower bounded by zero since P is positive semidefinite. To show the second part, observe that

$$\begin{bmatrix} x \\ z \\ \tau \end{bmatrix}^\top G(v) = s^\top z + \tau\kappa$$

for any $v \in \mathcal{C}$, so $(s^k)^T(z^k) + \tau^k\kappa^k \to 0$ since $G(v^k) \to 0$ and all sequences are bounded.

Lemma 2.2 *Suppose that $v^* := (x^*, z^*, s^*, \tau^*, \kappa^*)$ is an asymptotically complementary solution to $\mathcal{H}$. Then:*

i) If $\tau^* > 0$ then $(x^*/\tau^*, s^*/\tau^*)$ is an optimal solution to $\mathcal{P}$ and $(x^*/\tau^*, z^*/\tau^*)$ is an optimal solution to $\mathcal{D}$.
ii) If $\kappa^* > 0$ then at least one of the following holds:

- *$\mathcal{P}$ is infeasible and $z^* \in \mathbb{D}$.*
- *$\mathcal{D}$ is infeasible and $x^* \in \mathbb{P}$.*

Proof or (i), note that $\kappa^* \to 0$ since the solution is assumed asymptotically complementary. Then any point satisfying $G(v^*) = 0$ also satisfies the KKT conditions (1) following rescaling of (x^*, s^*, z^*) by τ^*.

For (ii), suppose that $\{v^k\} \in \mathcal{C}$, $k = 1, 2, \ldots,$ is any bounded sequence with limit v^*. Since $\tau^k \to 0$ by assumption, it follows that $(x^k)^\top P x^k \to 0$ since $\frac{1}{\tau^k}(x^k)^\top P x^k$ must remain bounded. Hence $Px^* = 0$ since P is assumed positive semidefinite, and both $Ax^* + s^* = 0$ and $A^\top z^* = 0$. Since $\{s^k, z^k\} \in \mathcal{K} \times \mathcal{K}^*$ by assumption and both cones are closed, $z^* \in \mathcal{K}^*$ and $-Ax^* \in \mathcal{K}$. Since $q^\top x^* + b^\top z^* = -\kappa^*$ with $\kappa^* > 0$, then at least one of the inner product terms must be negative. Consequently, if $q^\top x^* < 0$ then $x^* \in \mathbb{P}$ and $\mathcal{D}$ is infeasible, and if $b^\top z^* < 0$ then $z^* \in \mathbb{D}$ and $\mathcal{P}$ is infeasible.

2.2 Supported constraint types and barrier functions

Our interior-point method supports a variety of cones of both symmetric (i.e. self dual) and nonsymmetric type, and we allow the cone $\mathcal{K}$ in problem $\mathcal{P}$ to be any arbitrary composition of the basic cone types we describe in this section. For the symmetric case, we support the following cones:

- The *nonnegative cone* $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n \mid x \ge 0\}$.
- The *second order cone* $\mathcal{Q}_n := \left\{(u, v) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|v\|_2 \le u\right\}$.
- The *positive semidefinite cone* $\mathcal{S}_n := \left\{x \in \mathbb{R}^{n(n+1)/2} \mid \mathrm{mat}(x) \succeq 0\right\}$.

Our implementation for $\mathcal{S}_n$ uses the standard scaled and vectorized 'triangular' format. Given some symmetric matrix $M \in \mathbb{R}^{n\times n}$, we define an operator svec$(\cdot)$ that extracts the upper triangular part and scales off-diagonal terms by $\sqrt{2}$, i.e.

$$\begin{aligned} \text{svec}(M) := (M_{1,1}, \sqrt{2}M_{1,2}, M_{2,2}, \cdots, \\ \sqrt{2}M_{n-2,n-1}, M_{n-1,n-1}, \sqrt{2}M_{1,n}, \cdots, \sqrt{2}M_{n-1,n}, M_{nn}). \end{aligned}$$

The inverse operation mat$(\cdot)$ then restores the original matrix, i.e. $M = \text{mat}(\text{svec}(M))$. This scaling is necessary to preserve inner products, i.e. $\text{tr}(L^\top M) = \text{svec}(L)^\top \text{svec}(M)$ for matrices of compatible dimension.

We also support two nonsymmetric cones, both defined as subsets of $\mathbb{R}^3$:

- The *exponential cone* $\mathcal{K}_{\text{exp}} := \left\{(s_1, s_2, s_3) \in \mathbb{R}^3 \;\middle|\; s_2 \cdot e^{s_1/s_2} \le s_3, s_2 > 0\right\}$.
- The *power cone* $\mathcal{K}_{\text{pow}} := \left\{(s_1, s_2, s_3) \in \mathbb{R}^3 \;\middle|\; s_1^\alpha s_2^{1-\alpha} \ge |s_3|, s_1 \ge 0, s_2 \ge 0\right\}$.

For these two nonsymmetric cones the corresponding dual cones are

$$\begin{aligned} \mathcal{K}_{\text{exp}}^* &:= \left\{(z_1, z_2, z_3) \in \mathbb{R}^3 \;\middle|\; z_3 \ge -z_1 \cdot e^{z_2/z_1 - 1},\; z_3 > 0, z_1 < 0\right\} \\ \mathcal{K}_{\text{pow}}^* &:= \left\{(z_1, z_2, z_3) \in \mathbb{R}^3 \;\middle|\; \left(\frac{z_1}{\alpha}\right)^\alpha \cdot \left(\frac{z_2}{1-\alpha}\right)^{1-\alpha} \ge |z_3|,\; z_1, z_2 \ge 0\right\}. \end{aligned}$$

Finally, we support the zero cone $\mathcal{Z}_n := \{0\}^n$ to enable modelling of problems with equality constraints.

Barrier functions

For each nonzero cone we define a strictly convex *barrier function* $f : \mathcal{K} \to \mathbb{R}$ and its associated *conjugate barrier* $f^* : \mathcal{K}^* \to \mathbb{R}$. In particular, we use barrier functions that meet the following definition:

Definition 2.3 *([39, 58]) A function $f : \mathcal{K} \to \mathbb{R}$ is called a* logarithmically homogeneous self-concordant barrier *with degree $\nu \ge 1$ (ν–LHSCB) for a closed, convex and pointed cone $\mathcal{K} \subseteq \mathbb{R}^d$ if it satisfies the following conditions:*

$$\begin{aligned} |\nabla^3 f(x)[r, r, r]| &\le 2\left(\nabla^2 f(x)[r, r]\right)^{3/2} & \forall x \in \text{int}\,\mathcal{K}, r \in \mathbb{R}^d, \\ f(\lambda x) &= f(x) - \nu \log(\lambda) & \forall x \in \text{int}\,\mathcal{K}, \lambda > 0. \end{aligned}$$

The corresponding *conjugate function* $f^* : \mathcal{K}^* \to \mathbb{R}$ is defined as

$$f^*(y) := \sup_{x \in \mathcal{K}} \{-y^\top x - f(x)\}, \tag{8}$$

which is ν–LHSCB for $\mathcal{K}^*$ [58, Thm. 2.4.4]. The respective gradients ∇f and ∇f^* satisfy

$$x^\top \nabla f(x) = -\nu, \forall x \in \text{int}\,\mathcal{K}, \quad x^\top \nabla f^*(x) = -\nu, \forall x \in \text{int}\,\mathcal{K}^*. \tag{9}$$

In cases where the conjugate barrier function is known only through the definition (8) (i.e. rather than being representable in closed-form), we can compute its gradient as the solution to

$$\nabla f^*(y) := -\arg \sup_{x\in \operatorname{int} \mathcal{K}} \{-y^\top x - f(x)\}. \tag{10}$$

The relations (8)–(10) then collectively ensure that

$$\begin{aligned} f^*(y) &= -y^\top(-\nabla f^*(y)) - f(-\nabla f^*(y)) \\ &= -\nu - f(-\nabla f^*(y)), \end{aligned}$$

and also

$$-\nabla f^*(-\nabla f(x)) = x,\ \forall x \in \operatorname{int} \mathcal{K}, \quad -\nabla f(-\nabla f^*(y)) = y,\ \forall y \in \operatorname{int} \mathcal{K}^*.$$

2.3 The central path

We assume that $f : \mathcal{K} \to \mathbb{R}$ is a ν–LHSCB function on $\mathcal{K}$ with conjugate barrier f^* for some $\nu \geq 1$. Given any initial $v^0 \in \mathcal{C}$, we define the *central path* $v^*(\mu)$ as the unique solution to

$$G(v) = \mu G(v^0) \tag{11a}$$

$$s = -\mu\nabla f^*(z),\ z = -\mu\nabla f(s), \tag{11b}$$

which implies that

$$s^\top z/\nu = \mu.$$

If the cone $\mathcal{K}$ is symmetric, then we can replace the condition (11b) with a simpler condition defined in terms of the *Jordan product* [46, 59] on $\mathcal{K}$:

$$s \circ z = \mu \mathbf{e}, \tag{11c}$$

where $\mathbf{e}$ is the standard idempotent for $\mathcal{K}$.

There are three cases of interest when considering the existence of a central path:

- When $P = 0$, (7) reduces to a standard conic problem formulation with linear objective, our homogeneous embedding reduces to the classical homogeneous self-dual embedding, and the existence of a central path can be found in [44].
- When $P \neq 0$ and the cone $\mathcal{K}$ is symmetric, we can reformulate (7) as a linear complementarity problem, which is a special form of monotone complementarity problem. Existence of a central path in this case was shown in [51].
- When $P \neq 0$ and the cone $\mathcal{K}$ is nonsymmetric, we are not aware of any literature establishing a theoretical guarantee for the existence of a central path. If such guarantees are required for specific applications, one could apply a standard epigraphical formulation to remove the quadratic objective term.

The core of our interior-point algorithm amounts to a Newton-like method for computing a solution to the system of equations (11). We describe the calculation of step directions for this method in §3. Before doing so, we address solver initialization, termination and preprocessing steps in the remainder of this section.

2.4 Solver initialization

Before solving we perform an equilibration step on all matrix-valued data using the Ruiz equilibration technique described in [60]. We refer the reader to [31, §3.5] and [29, §5.1] for implementation details.

We have several strategies for choosing an initial iterate in the interior of our problem's conic constraints while being sufficiently near to the central path.

2.4.1 Symmetric cones

In the symmetric case, i.e. when $\mathcal{K}$ is self-dual, our method follows the approach of CVXOPT [46] for initialization, and we distinguish between the cases $P = 0$ and $P \neq 0$. Although the zero cone is not itself symmetric, we treat problems where $\mathcal{K}$ is a composition of symmetric and zero cones as symmetric for the purposes of initialization.

If $P \neq 0$, we first solve the linear system

$$\begin{bmatrix} P & A^\top \\ A & -I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix},$$

which is the solution to the problem

$$\min \quad \frac{1}{2} x^\top P x + q^\top x + \frac{1}{2}\|Ax - b\|_2^2.$$

We set $x^0 = x$ and

$$s^0 = \begin{cases} -z, & \alpha_p < -\epsilon \\ -z + (\epsilon + \alpha_p)\mathbf{e} & \text{otherwise}, \end{cases}$$

where $\alpha_p = \inf\{\alpha \mid -z + \alpha\mathbf{e} \in \mathcal{K}\}$. We introduce a threshold $\epsilon > 0$ to ensure that s^0 is in the strict interior of the cone $\mathcal{K}$. Likewise, z^0 is set to

$$z^0 = \begin{cases} z, & \alpha_d < -\epsilon \\ z + (\epsilon + \alpha_d)\mathbf{e} & \text{otherwise}, \end{cases}$$

where $\alpha_d = \inf\{\alpha \mid z + \alpha\mathbf{e} \in \mathcal{K}\}$.

If $P = 0$, we instead solve the linear system

$$\begin{bmatrix} 0 & A^\top \\ A & -I \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix},$$

in an effort to minimize the linear residuals. We set $x^0 = x$ and

$$s^0 = \begin{cases} -s, & \alpha_p < -\epsilon \\ -s + (\epsilon + \alpha_p)\mathbf{e} & \text{otherwise,} \end{cases}$$

where $\alpha_p = \inf\{\alpha \mid -s + \alpha\mathbf{e} \in \mathcal{K}\}$. We then solve the system

$$\begin{bmatrix} 0 & A^\top \\ A & -I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} -q \\ 0 \end{bmatrix},$$

which is equivalent to

$$\begin{aligned} \min_z \quad & \frac{1}{2}\|z\|_2^2 \\ s.t. \; & A^\top z + q = 0. \end{aligned}$$

Finally, we set

$$z^0 = \begin{cases} z, & \alpha_d < -\epsilon \\ z + (\epsilon + \alpha_d)\mathbf{e} & \text{otherwise,} \end{cases}$$

where $\alpha_d = \inf\{\alpha \mid z + \alpha\mathbf{e} \in \mathcal{K}\}$. In either case, we set the scalars (τ^0, κ^0) to 1.

2.4.2 Nonsymmetric cones

When $\mathcal{K}$ contains any nonsymmetric cone, we instead apply a unit initialization strategy as in [39, 43]. In this case, we initialize both primal and dual variables at a point satisfying $z = s = -\nabla f^*(z)$ (corresponding to $\mu^0 = 1$). This is equivalent to solving the unconstrained optimization

$$\min_z \; \frac{1}{2}z^2 + f^*(z),$$

which is strictly convex and has a unique solution. It yields $s^0_{sym} = z^0_{sym} = \mathbf{e}$ for symmetric cones,

$$s^0_{\text{exp}} = z^0_{\text{exp}} \approx (-1.051383, 0.556409, 1.258967)$$

for exponential cones, and

$$s^0_{\text{pow}} = z^0_{\text{pow}} = (\sqrt{1+\alpha}, \sqrt{2-\alpha}, 0)$$

for power cones of parameter α.

2.5 Chordal Decomposition

For semidefinite programs (SDPs) we follow the chordal decomposition and clique merging strategy developed as part of the COSMO solver [31]. We implement both a 'parent-child' merge strategy based on the clique-tree analysis method of [61] and the clique-graph merge strategy described in [31, §5]. We reformulate problems decomposed using either of these methods using the so-called 'compact' or 'range-space' conversion method described in [62, §5].

When implementing the clique-graph based merge strategy, we use the same edge-weight metric as in [31] to determine when candidate clique merges are accepted. Given a clique graph with V vertices and two cliques $\mathcal{C}_i$ and $\mathcal{C}_j$, we define an edge weight function $e : 2^V \times 2^V \to \mathbb{R}$ as

$$e(\mathcal{C}_i, \mathcal{C}_j) = |\mathcal{C}_i|^3 + |\mathcal{C}_j|^3 - |\mathcal{C}_i \cup \mathcal{C}_j|^3,$$

and merge candidate cliques when this metric is positive. We note that this metric was used in the ADMM-based solver [31, §5] because it relates directly to the computational complexity of projections onto the positive semidefinite cone. In our case the size of a Hessian block generated by a conic variable in $\mathcal{S}_n$ has dimension $\mathbb{R}^{(n(n+1)/2)\times(n(n+1)/2)}$, with the resulting KKT factorization time further complicated by the linking variables generated between overlapping cliques. We note therefore that it is likely that more efficient merge metrics are possible.

2.6 Termination Criteria

All termination criteria are based on unscaled problem data and iterates, i.e. after the Ruiz scaling described in §2.4 has been reverted. For checks of primal and dual feasibility, we first define normalized variables $\bar{x} = x/\tau$, $\bar{s} = s/\tau$, $\bar{z} = z/\tau$ and then define primal and dual residuals as

$$r_p := A\bar{x} + \bar{s} - b \tag{12a}$$
$$r_d := P\bar{x} + A^\top \bar{z} + q \tag{12b}$$

and primal and dual objectives as

$$g_p := \frac{1}{2}\bar{x}^\top P\bar{x} + q^\top \bar{x} \tag{13a}$$
$$g_d := -\frac{1}{2}\bar{x}^\top P\bar{x} - b^\top \bar{z}. \tag{13b}$$

We then declare convergence if each of the following three conditions holds:

$$\begin{aligned} \left\|r_p\right\| &< \epsilon_f \cdot \max\{1, \|b\|_\infty + \|\bar{x}\| + \|\bar{s}\|\} \\ \|r_d\| &< \epsilon_f \cdot \max\{1, \|q\|_\infty + \|\bar{x}\| + \|\bar{z}\|\} \\ |g_p - g_d| &< \epsilon_f \cdot \max\{1, \min\{|g_p|, |g_d|\}\}. \end{aligned}$$

We specify a default value of $\epsilon_f = 10^{-8}$ in our implementation, and a weaker threshold of $\epsilon = 10^{-5}$ when testing for 'near optimality' in cases of early termination (e.g. lack of progress, timeout, iterations limit).

When testing for infeasibility certificates, we do *not* normalize iterates, but rather work directly with the unscaled variables since infeasibility corresponds to the case where $\tau \to 0$. We declare primal infeasibility if

$$\begin{aligned}\left\|A^\top z\right\| &< -\epsilon_{i,r} \cdot \max(1, \|x\| + \|z\|) \cdot (b^\top z) \\ b^\top z &< -\epsilon_{i,a},\end{aligned}$$

and dual infeasibility if

$$\begin{aligned}\|Px\| &< -\epsilon_{i,r} \cdot \max(1, \|x\|) \cdot (b^\top z) \\ \|Ax + s\| &< -\epsilon_{i,r} \cdot \max(1, \|x\| + \|s\|) \cdot (q^\top x) \\ q^\top x &< -\epsilon_{i,a}.\end{aligned}$$

We again set $\epsilon_{i,r} = \epsilon_{i,a} = 10^{-8}$ as default values, and allow for weaker thresholds to declare 'near infeasibility' certificates in cases of early termination.

3 Computing step directions

Our interior-point method computes Newton-like search directions using a linearization of (11) given some right-hand side residual $d := (d_x, d_z, d_\tau, d_s, d_\kappa)$. This produces a linear system in the form

$$\begin{bmatrix} 0 \\ \Delta s \\ \Delta\kappa \end{bmatrix} - \begin{bmatrix} P & A^\top & q \\ -A & 0 & b \\ -(q + 2P\xi)^\top & -b^\top & \xi^\top P \xi \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \\ \Delta\tau \end{bmatrix} = - \begin{bmatrix} d_x \\ d_z \\ d_\tau \end{bmatrix} \tag{14a}$$

$$H\Delta z + \Delta s = -d_s, \quad \kappa\Delta\tau + \tau\Delta\kappa = -d_\kappa, \tag{14b}$$

where $\xi := x\tau^{-1}$ and $H \in \mathbb{R}^{m \times m}$ is a positive definite matrix that we will refer to as the *scaling matrix*. The choice of both the scaling matrix and the right-hand side terms d in (14) depend on the search direction, the symmetry or asymmetry of the cone $\mathcal{K}$ and the particular choice of scaling strategy. We defer more precise definitions of these terms to later in this section.

Our approach to solving (14) follows that of [46, 47], but differs in the fact that some blocks in the coefficient matrix of (14a) include an additional term when $P \neq 0$. We first eliminate the variables $(\Delta s, \Delta\kappa)$ to obtain the reduced system

$$\begin{bmatrix} P & A^\top & q \\ -A & H & b \\ -(q+2P\xi)^\top & -b^\top & \xi^\top P\xi + \frac{\kappa}{\tau} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \\ \Delta\tau \end{bmatrix} = \begin{bmatrix} d_x \\ d_z - d_s \\ d_\tau - d_\kappa/\tau \end{bmatrix} \tag{15a}$$

$$\Delta s = -d_s - H\Delta z, \quad \Delta\kappa = -(d_\kappa + \kappa\Delta\tau)/\tau. \tag{15b}$$

To solve (15) we solve a pair of linear systems with a common left-hand side

$$\begin{bmatrix} P & A^\top \\ A & -H \end{bmatrix} \begin{bmatrix} \Delta x_1 & \Delta x_2 \\ \Delta z_1 & \Delta z_2 \end{bmatrix} = \begin{bmatrix} d_x & -q \\ -(d_z - d_s) & b \end{bmatrix} \tag{16}$$

and then recover $(\Delta\tau, \Delta x, \Delta z)$ using

$$\begin{aligned} \Delta\tau &= \frac{d_\tau - d_\kappa/\tau + (2P\xi + q)^\top \Delta x_1 + b^\top \Delta z_1}{\kappa/\tau + \xi^\top P\xi - (2P\xi + q)^\top \Delta x_2 - b^\top \Delta z_2} \\ &= \frac{d_\tau - d_\kappa/\tau + q^\top \Delta x_1 + b^\top \Delta z_1 + 2\xi^\top P\Delta x_1}{\kappa/\tau + \|\Delta x_2 - \xi\|_P^2 - \|\Delta x_2\|_P^2 - q^\top \Delta x_2 - b^\top \Delta z_2}, \end{aligned} \tag{17}$$

and

$$\Delta x = \Delta x_1 + \Delta\tau \cdot \Delta x_2, \quad \Delta z = \Delta z_1 + \Delta\tau \cdot \Delta z_2.$$

Finally, we recover $(\Delta s, \Delta\kappa)$ from (15b). After obtaining the search direction $(\Delta s, \Delta z, \Delta\tau, \Delta\kappa)$, we compute the maximal step size α ensuring that the new update resides in the interior of conic constraints. In cases where any part of $\mathcal{K}$ is nonsymmetric, we also choose α sufficiently small so that the updated values $(s+\alpha\Delta s, z+\alpha\Delta z)$ remain within a neighborhood of the central path (11) using the proximity metric described in [39].

3.1 Scaling matrices

The choice of scaling matrix H in (14) depends on the way in which we linearize the central path equations as defined in Sect. 2.3. For symmetric cones the most common choice is the NT scaling [35, 36], which we also employ. For nonsymmetric cones, the central path is defined by the set of points satisfying (11b), and both symmetric [39] and nonsymmetric [44] scaling strategies have been implemented in state-of-the-art solvers. We set $H = 0$ for the zero cone.

3.1.1 Symmetric cones

For symmetric cones we linearize the central path equation (11c). The NT scaling method exploits the self-scaled property of symmetric cone $\mathcal{K}$ to define, for $(s, z) \in \mathcal{K}$, a unique scaling point $w \in \mathcal{K}$ satisfying

$$\bar{H}(w)s = z.$$

The matrix $\bar{H}(w)$ can be factorized as $\bar{H}^{-1}(w) = W^\top W$, and we set $H = \bar{H}^{-1}(w)$ in (14). The terms (w, W) are then computed following [46] except in the case of second-order cones, where we instead apply a modified version of the sparse factorization strategy of ECOS [47] for second-order cones of dimension 5 or greater, and the standard dense form for $\bar{H}^{-1}(w)$ from [46] otherwise. Similar sparsification techniques can also be applied for certain nonsymmetric cones, which we describe in detail in [63].

3.1.2 Nonsymmetric cones

In the nonsymmetric case the self-scaled property does not hold and the central path cannot be formulated as in (11c) due to the lack of a Jordan algebra, so we instead linearize (11b). A general primal-dual scaling strategy suitable for nonsymmetric cones was consequently proposed in [64], and used later in [39], which relies on the satisfaction of two secant equations

$$Hz = s, \quad H\nabla f(s) = \nabla f^*(z).$$

Suppose we define *shadow iterates* as

$$\tilde{z} := -\nabla f(s), \quad \tilde{s} := -\nabla f^*(z),$$

with

$$\tilde{\mu} = \langle \tilde{s}, \tilde{z} \rangle / \nu.$$

A scaling matrix H can then be obtained from the rank-4 Broyden-Fletcher-Goldfarb-Shanno (BFGS) update, which is commonly used in quasi-Newton methods,

$$H := H_{\text{BFGS}} := Z(Z^\top S)^{-1} Z^\top + H_a - H_a S (S^\top H_a S)^{-1} S^\top H_a,$$

where $Z := [z, \tilde{z}]$, $S := [s, \tilde{s}]$, $\tilde{z} := -\nabla f(s)$, $\tilde{s} := -\nabla f^*(z)$ and $H_a \succ 0$ is an approximation of the Hessian. In our implementation, we choose $H_a = \mu \nabla^2 f^*(z)$ following [39] and the calculation of H_{BFGS} reduces to a rank-3 update:

$$\begin{aligned} H_{\text{BFGS}} =& \mu \nabla^2 f^*(z) + \frac{1}{2\mu\nu} \delta_s \left(s + \mu\tilde{s} + \frac{1}{\mu\tilde{\mu} - 1} \delta_s \right)^T + \frac{1}{2\mu\nu} \left(s + \mu\tilde{s} + \frac{1}{\mu\tilde{\mu} - 1} \delta_s \right) \delta_s^T \\ & - \mu \frac{\left(\nabla^2 f^*(z)\tilde{z} - \tilde{\mu}\tilde{s} \right) \left(\nabla^2 f^*(z)\tilde{z} - \tilde{\mu}\tilde{s} \right)^T}{\left\langle \tilde{z}, \nabla^2 f^*(z)\tilde{z} \right\rangle - \nu\tilde{\mu}^2}, \end{aligned}$$

where $\delta_s := s - \mu\tilde{s}$. More details about the primal-dual scaling of nonsymmetric cones can be found in [39].

3.2 The affine and centering directions

At every interior-point iteration, our method requires us to solve the linear system (14) for each of two different right-hand side terms $(d_x, d_z, d_\tau, d_s, d_\kappa)$. This amounts to a single factorization of the condensed system (16) followed by three linear solves, noting the common right-hand-side term $(-q, b)$ in (16).

The first of these two choices produces the *affine step* direction, where we compute the search direction $(\Delta x, \Delta s, \Delta z, \Delta\kappa, \Delta\tau)$ to eliminate the linear residuals in (6), i.e. by computing a pure Newton step direction for (11) with $\mu = 0$. The second is called the *combined step*, which is the affine step plus a centering correction toward the central path. In practice, the estimated direction $(\Delta x, \Delta s, \Delta z, \Delta\kappa, \Delta\tau)$ from the affine step is used to determine the right-hand side $(d_x, d_z, d_\tau, d_s, d_\kappa)$ for the combined step to compute a higher-order correction for acceleration. In the language of predictor-corrector algorithms, the affine step is the *predictor* while the centering step as the *corrector*.

Computation of a higher-order correction term η is a heuristic technique that is known to accelerate the convergence of IPMs significantly. The choice of this term varies depending on the choice of the scaling matrix H and whether a given cone constraint is symmetric. For our method $(d_x, d_z, d_\tau, d_s, d_\kappa)$ is defined as

$$d_x = r_x, d_z = r_z, d_\tau = r_\tau, d_\kappa = \kappa\tau, d_s = s, \tag{18}$$

in the affine step, and

$$\begin{aligned} d_x = (1-\sigma)r_x, d_z = (1-\sigma)r_z, d_\tau = (1-\sigma)r_\tau, d_\kappa = \kappa\tau + \Delta\kappa\Delta\tau - \sigma\mu, \\ d_s = \begin{cases} W^\top(\lambda \backslash (\lambda \circ \lambda + \eta - \sigma\mu\mathbf{e})) & \text{(symmetric)} \\ s + \sigma\mu\nabla f^*(z) + \eta & \text{(nonsymmetric),} \end{cases} \end{aligned} \tag{19}$$

in the combined step. For symmetric cones we use the Mehrotra correction $\eta = (W^{-1}\Delta s) \circ (W\Delta z)$ [46], while for nonsymmetric cones we compute η using the 3rd-order correction method from [39], i.e.

$$\eta = -\frac{1}{2}\nabla^3 f^*(z)[\Delta z, \nabla^2 f^*(z)^{-1}\Delta s].$$

We choose $\sigma = (1-\alpha_{\text{aff}})^3$, where α_{aff} is the step size from the affine step.

3.3 Linear solve method

Nearly all of the cost of computing step directions in §3 typically comes when solving the symmetric linear system (16). To solve this system we add a small regularization term ϵ_s and then compute a direct factorization

$$K := \begin{bmatrix} P + \epsilon_s I & A^\top \\ A & -(H + \epsilon_s I) \end{bmatrix} = LDL^\top. \tag{20}$$

This *static regularization* ensures that the matrix K is quasidefinite [65] even when P or A are rank deficient, and consequently that the factor D is diagonal with $D_{ii} \neq 0$ when computing the Cholesky-like factorization LDL^T [66]. This approach also ensures that the sparsity pattern of the factor L depends only on the problem's sparsity pattern, so the method is allocation-free after the first factorization. When computing the LDL factorization we also employ a *dynamic regularization* strategy by bounding pivots away from zero by an amount ϵ_d to ensure that the factorization is numerically stable. We have extended the open-source package QDLDL, original developed for OSQP and based on [67], to support this regularization strategy.

The improved numerical efficiency of our scheme relative to the standard HSDE method can be seen by consideration of the linear system in (16). If we had started instead from the epigraphical reformulation (3), then the coefficient matrix in (16) would have been

$$\begin{bmatrix} 0 & A^T & [2P^{\frac{1}{2}}]^T \\ A & -H & \\ 2P^{\frac{1}{2}} & & -H_{\mathcal{K}} \end{bmatrix} \tag{21}$$

with $H_{\mathcal{K}}$ a scaling matrix appropriate for the additional second order cone constraint introduced in (3). Direct factorization of (21) can produce significantly more fill-in than for coefficient matrix (16), particularly when the matrix factor $P^{\frac{1}{2}}$ already has substantial fill-in.

4 Implementation: the Clarabel solver

We have implemented our approach in the CLARABEL solver, an open-source and liberally licensed software package with separate implementations in both the Rust [68] and Julia [69] programming languages. We provide a high-level sketch of the overall algorithm in Algorithm 1, with more detail on specific steps provided in the preceding sections. Both implementations, along with extensive documentation, are available under the Apache v2.0 license and are publicly available via our main project site at

https://clarabel.org/

Our Rust implementation is intended for most academic and industrial end users. This implementation can also be accessed via other common languages through standard foreign function interfaces (FFIs) and wrappers, and we currently provide such wrappers for Python, C, C++ and R. We also provide interfaces in Python to the standard modelling package CVXPY [70, 71]. Clarabel is installed as part of the standard CVXPY distribution [72] and is the default solver in CVXPY for linear and second-order cone programs as of version 1.5.

The Rust version of Clarabel provides its own internal implementation of most linear algebra functionality, including a standalone Rust reimplementation of the quasidefinite linear solver QDLDL with regularization features as described in §3.3. We use Rust interfaces to user-selectable BLAS [73] implementations (e.g. [74, 75]) for solving conic programs on the semidefinite cone. We allow for the use of alternative direct

Algorithm 1 Interior-point method implemented in CLARABEL

Require: Problem data (P, q, A, b); composite convex cone $\mathcal{K}$ with dual $\mathcal{K}^*$.

Compute chordal decomposition for problem data (see §2.5)
Compute Ruiz equilibration for problem data (see [31, §3.5], [29, §5.1])
Initialize variables $(x, s, z, \tau, \kappa)_0$ (see §2.4).
loop
  Compute residuals and duality gap from (12)–(13):
  if termination by certificate (§2.6) or iteration bound **then**
    break;
  Compute scaling matrices $H(w)$ (symmetric: (§3.1.1), nonsymmetric: (§3.1.2))
  Update and refactor the KKT system (20)
  Affine step direction:
    Compute affine step RHS terms from (18)
    Compute affine step direction $(\Delta x, \Delta s, \Delta z, \Delta\tau, \Delta\kappa)_{\text{aff}}$ from (15)–(16)
    Line search:
      $\alpha_{\text{aff}} = \max\{\alpha \in (0, 1) : (s, z) + \alpha(\Delta s, \Delta z)_{\text{aff}} \succ_{(\mathcal{K}\times\mathcal{K}^*)} 0, (\tau, \kappa) + \alpha(\Delta\tau, \Delta\kappa)_{\text{aff}} > 0\}$
  Centering parameter:. $\sigma = (1 - \alpha_{\text{aff}})^3$
  Combined step direction:
    Compute combined step RHS terms from (19)
    Compute combined step direction $(\Delta x, \Delta s, \Delta z, \Delta\tau, \Delta\kappa)$ from (15)–(16)
    Line search:
      $\alpha = \max\{\alpha \in (0, 1) : (s + \alpha\Delta s, z + \alpha\Delta z) \succ_{\mathcal{K}\times\mathcal{K}^*} 0, \tau + \alpha\Delta\tau > 0, \kappa + \alpha\Delta\kappa > 0\}$
  Update variables:
    $(x, s, z, \tau, \kappa) \leftarrow (x, s, z, \tau, \kappa) + \beta\alpha(\Delta x, \Delta s, \Delta z, \Delta\tau, \Delta\kappa), \beta \in (0, 1)$.
end loop

linear solvers, and provide support for a 3rd-party open-source multithreaded supernodal LDL factorization method implemented in native Rust as part of the `faer-rs` package [76]. We also provide optional support for the Intel MKL and Panua Pardiso [77] solvers.

The Julia implementation is intended for use both as a standalone solver for users of the Julia language and as a prototyping platform for future algorithmic development. The Julia implementation relies heavily on native Julia functions for most linear algebra functionality, aside from a Julia implementation of QDLDL which we provide as a standalone package. In Julia, we also provide the option of using alternative linear solve methods including CHOLMOD [78], Pardiso [77] and the HSL MA57 [79] solver.

Both implementations provide the same functionality and support the same set of conic constraints. We also provide support for different floating point data types in both languages, e.g. for standard 32- or 64-bit single or double precision floating point types [80] or for extended precision types such as the Julia BIGFLOAT type. Our implementation is inspired by the modular design pattern of the interior-point solver OOQP [81], in the sense that all internal data types are defined as abstract types that can be extended or customized by end users to specific problem classes to exploit domain-specific structure. In the Rust implementation this functionality relies heavily on Rust's trait-based type system and generics, while in Julia we instead rely on Julia's dynamic dispatch and "duck typing" [82].

For more detail we refer the reader to the documentation available on the CLARABEL project website.

5 Numerical Experiments

We have benchmarked our implementation of Clarabel against a variety of open-source and commercial solvers: the open-source solver ECOS [47] (v2.0.8), which uses an HSDE-based primal dual interior point method similar to that in CVXOPT [46], the open-source solver HiGHS [25] (v1.11.0), and the commercial solvers Mosek [48] (v11.0.20) and Gurobi [23] (v12.0.0). We explicitly select Gurobi's barrier method algorithm for all applicable test cases to enable direct comparison of iteration counts. Where applicable we have disabled crossover in all solvers for fairness of comparison. All benchmarks are executed with all default settings for all solvers enabled (using termination conditions and tolerances for Clarabel as reported in §2.6), but with pre-solve disabled where applicable to ensure that the solvers are solving equivalent problems. We do not impose any iteration limits other than those specified within each solver's internal defaults. We set the maximum solve time to 300 seconds unless otherwise stated.

The Julia implementation of Clarabel uses our native implementation of the direct LDL linear solver QDLDL for all tests. The QDLDL solver is relatively unsophisticated relative to the multithreaded methods used in commercial solvers, but is lightweight, open-source and does not rely on any external libraries. Our Rust implementation is configured to automatically choose between the simplicial QDLDL method and the supernodal method from the `faer-rs` package [76], with the choice determined by the ratio of estimated flop count to the number of nonzero entries allocated for the L factor. We automatically switch from the simplicial to the supernodal method when this ratio exceeds some threshold (nominally 40).

All experiments were carried out on a 36 core Intel Xeon w9-3475X testbed with 256 GB RAM. Unless otherwise stated, test problems in our benchmark results were run as single threaded process with available memory limited to 16 GB RAM. All benchmark tests are scripted in Julia and access solver interfaces via JuMP [83]. We use Rust compiler version 1.86.0 and Julia version 1.11.6. The code for all numerical examples is publicly available [84].

For each set of benchmark problems in our results we exclude problems for which none of the benchmarked solvers produced a valid solution. We provide a summary of the results for all benchmarked solvers appropriate to each problem class in the form of shifted geometric means and performance profiles in the remainder of this section.

For all benchmark test sets, we provide more detailed numerical results for our Rust and Julia implementations as well as the solvers Gurobi, Mosek and ECOS (where applicable) in Appendix A, including solve times and iteration counts. We include only this subset of solvers in our detailed reporting since all implement interior-point based methods and therefore have iteration counts that are directly comparable.

Shifted geometric means

We follow the standard benchmarking convention of using a normalized shifted geometric mean for comparison of solve time across different solvers [85]. For a set of N test problems, we define the shifted geometric mean solve time g_s for solver s as

$$g_s := \left[\prod_{p=1}^{N}(t_{p,s} + k)\right]^{\frac{1}{N}} - k,$$

where $t_{p,s}$ is the time in seconds for solver s to solve problem p, and $k = 1$ is the shift. The normalized shifted geometric mean is then defined as

$$r_s := \frac{g_s}{\min_s g_s},$$

so that the solver with the lowest shifted geometric mean solve time has a normalized score of 1. For those problems for which a given solver fails, we assign a solve time $t_{p,s}$ equal to the maximum allowable solve time for the relevant benchmark.

When reporting shifted geometric means, we report results based on the default tolerances for each solver (labelled as "Full Accuracy"). For cases where a solver does not solve to full accuracy, but determines that a problem is "almost solved" (or an equivalent status) at termination, we count the case as a "Low Accuracy" solution. We apply normalizations separately for the "Full Accuracy‘‘ and "Low Accuracy" cases for all test sets.

Performance profiles

We also provide performance profiles [86] to compare both the relative and absolute performance of different solvers. For a set of N test problems, we define the *relative performance ratio* for solver s and problem p as

$$u_{p,s} = \frac{t_{p,s}}{\min_s t_{p,s}}.$$

The performance profile for the solver s is then a plot of the function $f_s^{\text{rel}} : \mathbb{R}_+ \mapsto [0, 1]$ defined as

$$f_s^{\text{rel}}(\tau) := \frac{1}{N}\sum_p \mathcal{I}_{\le\tau}(u_{p,s}),$$

where $\mathcal{I}_{\le\tau}(u_{p,s}) = 1$ if $u_{p,s} \le \tau$ and $\mathcal{I}_{\le\tau}(u_{p,s}) = 0$ otherwise. The relative performance profile therefore shows, at each level τ, the fraction of problems solved by solver s in time within a factor τ of the solve time of the best solver.

Since the relative performance profile for a given solver can change depending on the overall collection of solvers being benchmarked, we further compute an *absolute performance profile* by plotting a function $f_s^{\text{abs}} : \mathbb{R}_+ \mapsto [0, 1]$ as

$$f_s^{\text{abs}}(\tau) := \frac{1}{N}\sum_p \mathcal{I}_{\le\tau}(t_{p,s}).$$

The absolute performance profile then shows, at each level τ, the fraction of problems solved by solver s within τ seconds and is independent of the other solvers benchmarked.

5.1 Benchmark problems with quadratic objectives

In this section we present benchmark results for quadratic programming (QP) problems in the standard form $\mathcal{P}$ with the set $\mathcal{K}$ restricted to the composition of the zero cone (i.e. modelling linear equality constraints) and the nonnegative orthant. We consider example problems taken or generated from standard open-source problem collections and covering a wide range of problem dimensions.

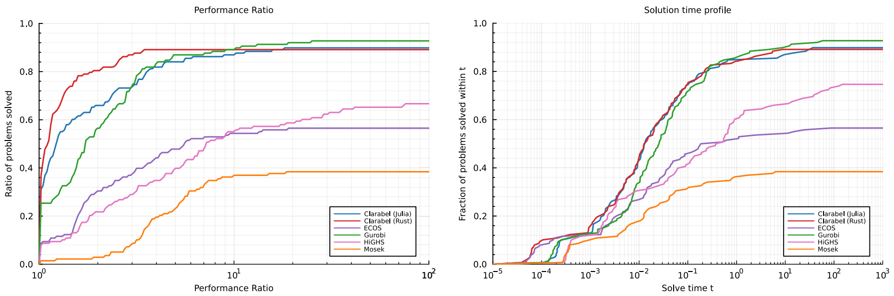
5.1.1 The Maros-Meszaros test set

We consider first the standard benchmark collection of 138 quadratic programs from the Maros-Meszaros test set [87]. This collection of QPs includes a wide range of problem sizes and contains a number of difficult test cases due to numerical ill-conditioning, rank deficiency or poor scaling.

Results Results for this benchmark set are shown in Figure 1 for all solvers. Clarabel is clearly faster than all solvers other than Gurobi on this benchmark set, with the Rust implementation marginally faster than the Julia one as expected. Compared to Gurobi, Clarabel is faster on a majority of instances, but with a slightly higher failure rate at full accuracy. We observe that the seemingly large gap in the relative performance profile of our Rust and Julia implementations is almost entirely attributable to faster solve times among the smallest examples in this test set. All solvers fail on at least some subset of these benchmarks, with Gurobi the lowest failure rate at full accuracy, and Clarabel the lowest failure rate at reduced accuracy.

Of particular note in this benchmark set is the high failure rate of the ECOS and Mosek solvers, since both are interior-point methods broadly similar to Clarabel. In the case of ECOS these failures are partly attributable to the solvers' requirement to reformulate QP problems in the conic form (21), since ECOS does not support quadratic objectives natively. This leads to immediate failures in a substantial number of cases due to ill-conditioning of the matrix P, resulting in a failed attempt to compute the Cholesky factor $P^{\frac{1}{2}}$ in (21) when P is either semidefinite or contains very small negative eigenvalues. Mosek handles this case more reliably, but is still not able to solve a substantial number of problems to full accuracy within the benchmark time limit.

In Figure 2 we show the performance of Clarabel compared to ECOS using both our own approach and the more standard one in which quadratic objectives are handled via an epigraphical reformulation that transforms the problem to an SOCP as in (3). In this case we observe that the performance of Clarabel's algorithm (labelled `Clarabel-HSDE`) is very close to that of ECOS, since in this case the underlying interior point method amounts to a very similar method. The avoidance of this epigraphical reformulation leads to a very substantial performance benefit in terms of



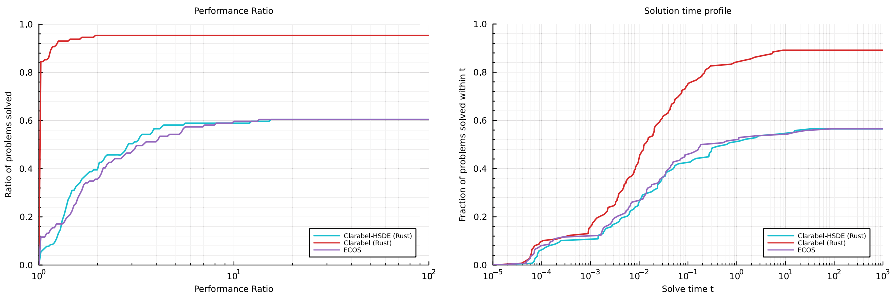
(a) Relative performance profile

(b) Absolute performance profile

		ClarabelRs	Clarabel	ECOS	Gurobi	HiGHS	Mosek
Shifted GM	Full Acc.	1.28	1.35	14.97	1.0	6.84	40.88
	Low Acc.	1.0	1.26	14.24	1.9	15.68	3.53
Failure Rate (%)	Full Acc.	10.9	10.1	43.5	7.2	25.4	61.6
	Low Acc.	2.9	2.9	28.3	5.8	25.4	10.1

(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 1 Performance profiles for the Maros-Meszaros problem set



(a) Relative performance profile

(b) Absolute performance profile

		ClarabelRs	ClarabelRsHSDE	ECOS
Shifted GM	Full Acc.	1.0	11.82	11.73
	Low Acc.	1.0	6.16	14.24
Failure Rate (%)	Full Acc.	10.9	43.5	43.5
	Low Acc.	2.9	17.4	28.3

(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 2 Performance profiles for the Maros-Meszaros (HSDE format) problem set

both numerical speed and the number of problems solvable. We observe very similar results in all QP test cases.

5.1.2 Least-squares problems with suitesparse matrices

We next consider a collection of 46 sparse least-squares problems $Ax \approx b$ derived from matrices taken from the SuiteSparse Matrix Collection [88], following the equivalent set of benchmark examples from [29]. For each case we compute an approximate solution by solving a constrained QP using two standard methods:

Huber Problem: The *Huber fitting* [89, 90] or *robust least squares* problem for a given matrix A and vector b is defined as

$$\min_{x} \sum_{i=1}^{m} \phi(a_i^T x - b_i),$$

where $a_i^\top$ is the i^{th} row of A and the *Huber loss function* $\phi : \mathbb{R} \to \mathbb{R}$ is defined as

$$\phi(w) = \begin{cases} w^2 & |w| \leq M \\ M(2|w| - M) & |w| > M. \end{cases}$$

We set $M = 1$ for all test cases. This problem is equivalent [91] to the following quadratic program:

$$\begin{array}{ll} \min\limits_{x,u,r,s} & u^\top u + 2M\mathbf{1}^\top(r+s) \\ \text{subject to:} & Ax - b - u = r - s \\ & (r,s) \geq 0. \end{array}$$

LASSO Problem: The *least absolute shrinkage and selection operator (LASSO)* problem [8, 9] for a given matrix A and vector b is defined as

$$\min_{x} \|Ax - b\|_2^2 + \lambda \|x\|_1 .$$

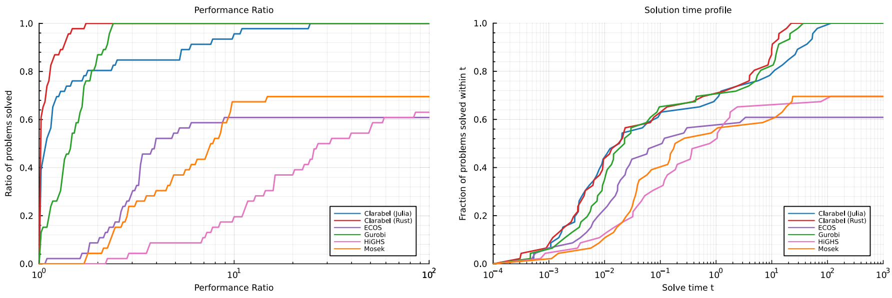
We set $\lambda = \left\|A^\top b\right\|_\infty$ for all test cases. This problem is equivalent [91] to the following quadratic program:

$$\begin{array}{ll} \min\limits_{y,x,t} & y^\top y + \lambda \mathbf{1}^\top t \\ \text{subject to:} & Ax - b = y \\ & -t \leq x \leq t. \end{array}$$

Results Results for this benchmark set of 46 problems are shown in Figure 3. The Rust implementation of Clarabel is the fastest solver overall, with Gurobi performance somewhere between our Rust and Julia implementations. The reason for this difference is that the Rust implementation benefits from a switch to a supernodal factorization method for the larger problems in the test set. In this test set only the Clarabel and Gurobi solvers are able to solve all cases to full accuracy.

5.1.3 Constrained optimal control

Finally, we consider finite-horizon constrained optimal control problems with quadratic objectives. Problems of this type are of particular interest in embedded control systems, since the repeated online solution of such problems is the basis of



(a) Relative performance profile

(b) Absolute performance profile

		ClarabelRs	Clarabel	ECOS	Gurobi	HiGHS	Mosek
Shifted GM	Full Acc.	1.0	1.64	9.25	1.19	7.49	7.77
	Low Acc.	1.0	1.64	9.25	1.19	7.49	2.08
Failure Rate (%)	Full Acc.	0.0	0.0	39.1	0.0	30.4	30.4
	Low Acc.	0.0	0.0	39.1	0.0	30.4	0.0

(c) Benchmark timings as shifted geometric mean and failure rates

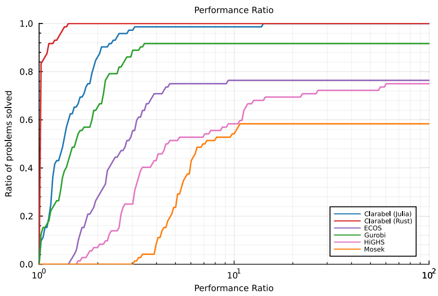
Fig. 3 Performance profiles for the SuiteSparse least-squares problem set

the model predictive control (MPC) method [1]. We consider a collection of 72 such problems taken from the benchmark collection of industrial and academic applications in [92]. Problems in this collection are in the form

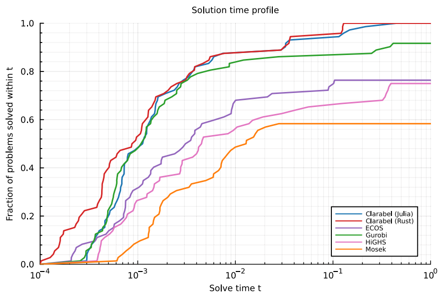
$$\begin{aligned}
\min_{\{x_i\},\{y_i\},\{u_i\}} \quad & \sum_{i=0}^{N-1} \begin{pmatrix} y_i - y_i^r \\ u_i - u_i^r \end{pmatrix} \begin{pmatrix} Q_k & S_k \\ S_k^T & R_k \end{pmatrix} \begin{pmatrix} y_i - y_i^r \\ u_i - u_i^r \end{pmatrix} + \begin{pmatrix} g_k^y \\ g_u^k \end{pmatrix}^\top \begin{pmatrix} y_i - y_i^r \\ u_i - u_i^r \end{pmatrix} \\
& + (x_N - x_N^r)^\top P (x_N - x_N^r) \\
\text{subject to:} \quad & \left.\begin{aligned} & x_{k+1} = A_k x_k + B_k u_k + f_k \\ & y_k = C_k x_k + D_k u_k + e_k \\ & d_k^\ell \leq M_k x_k + N_k u_k \leq d_k^u \\ & u_k \in \mathcal{U}_k, \;\; y_k \in \mathcal{Y}_k \end{aligned}\right\} \quad k = 0 \dots N-1 \\
& T x_N \in \mathcal{T},
\end{aligned}$$

where the constraint sets $\mathcal{U}_k$, $\mathcal{Y}_k$ and $\mathcal{T}$ are interval constraints. All problems have $Q_k \succeq 0$, $R_k \succeq 0$ and $P \succ 0$, which ensures that the problems are all convex QPs. As is typical of optimal control problems for embedded systems, the dimension of the states x_k and inputs u_k are relatively small (max 12 and 4, respectively), with horizons N up to 100.

Results Results for this benchmark set are shown in Figure 4. Clarabel is again the fastest solver overall, and is the only solver tested with a 100% success rate in solving problems to full accuracy.



(a) Relative performance profile



(b) Absolute performance profile

		ClarabelRs	Clarabel	ECOS	Gurobi	HiGHS	Mosek
Shifted GM	Full Acc.	1.0	1.61	287.47	63.7	329.31	981.27
	Low Acc.	1.0	1.61	12.51	39.77	329.31	4.89
Failure Rate (%)	Full Acc.	0.0	0.0	23.6	8.3	25.0	41.7
	Low Acc.	0.0	0.0	1.4	5.6	25.0	0.0

(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 4 Performance profiles for the optimal control problem set

5.2 Benchmark problems with linear objectives

In this section we present benchmark results for optimization problems *without* quadratic objective terms, i.e. with $P = 0$ in $\mathcal{P}$. We again consider example problems taken or generated from standard open-source problem collections and covering a wide range of problem dimensions. Our test set covers cases with constraints on the *non-negative orthant* (i.e. linear programs), as well as both second-order and exponential cone programs.

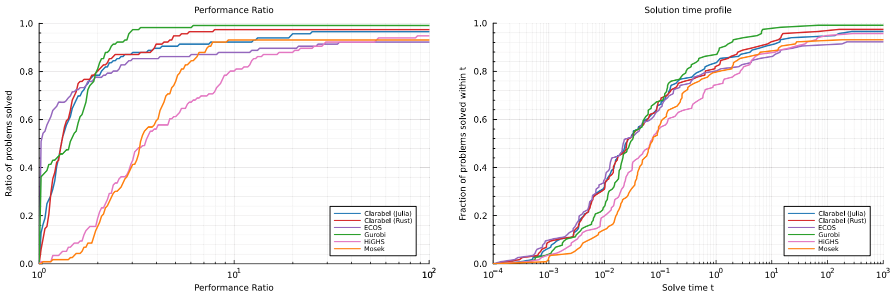
5.2.1 NETLIB LP problems

We first consider linear programming (LP) problems taken from the NETLIB collection, a standard collection of benchmark LPs [93, 94]. Our benchmark test set includes 117 feasible and 29 infeasible test cases.

Results for the feasible and infeasible test sets are shown in Figures 5 and 6, respectively. For these cases Clarabel has performance similar to both Mosek and Gurobi for the feasible set, and somewhat slower for the infeasible set. This result is to be expected since most of the potential performance advantage of our method arises from improved handling of quadratic objectives, but illustrates that our implementation is, in the LP case, still broadly comparable.

5.2.2 CBLIB exponential cone problems

We consider a collection of 47 exponential cone programs from the CBLIB benchmark collection [95] in order to test performance on nonsymmetric cone programs. As expected, for these problems our implementation has broadly similar performance to Mosek but with a generally higher iteration count.(See Fig. 7)

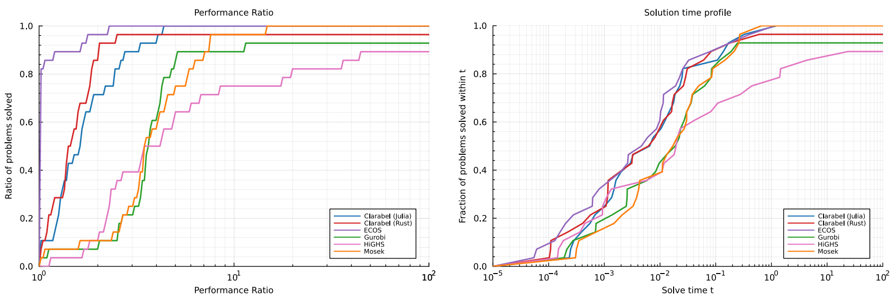


(a) Relative performance profile **(b)** Absolute performance profile

		ClarabelRs	Clarabel	ECOS	Gurobi	HiGHS	Mosek
Shifted GM	Full Acc.	1.91	2.22	3.46	1.0	3.54	3.13
	Low Acc.	1.69	1.98	1.94	1.0	3.54	1.38
Failure Rate (%)	Full Acc.	2.6	3.4	7.8	0.9	4.3	6.9
	Low Acc.	1.7	2.6	2.6	0.9	4.3	0.0

(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 5 Performance profiles for the NETLIB Feasible LP problem set



(a) Relative performance profile **(b)** Absolute performance profile

		ClarabelRs	Clarabel	ECOS	Gurobi	HiGHS	Mosek
Shifted GM	Full Acc.	4.81	1.09	1.0	9.94	25.37	1.17
	Low Acc.	1.29	1.09	1.0	9.94	25.37	1.17
Failure Rate (%)	Full Acc.	3.6	0.0	0.0	7.1	10.7	0.0
	Low Acc.	0.0	0.0	0.0	7.1	10.7	0.0

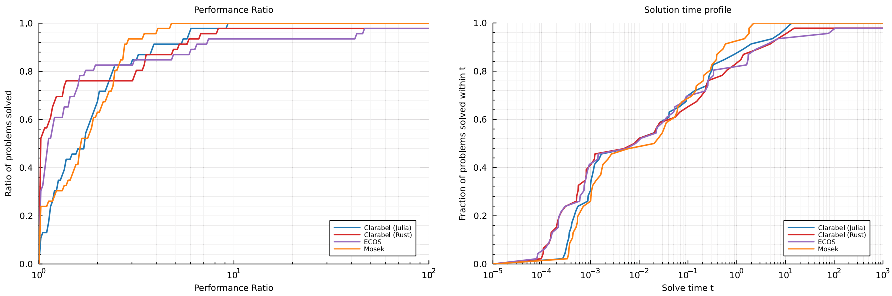
(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 6 Performance profiles for the NETLIB Infeasible LP problem set

5.2.3 Optimal power flow

We next consider a collection of optimal power flow problems based on power networks from IEEE PLS PGLib-OPF benchmark library [96] and constructed using the `PowerModels.jl` benchmark test framework [97]. This framework allows for the generation of optimal power flow problems with various modelling assumptions and convex relaxations applied [98, 99]. We consider in particular linear programming problems based on linearized (i.e. direct current (DC)) power flow models [100], and SOCPs arising from second-order cone relaxations of AC models [101]. Results from these benchmarks are shown in Figs. 8 and 9, respectively.

For both of these test sets Clarabel performs similarly to Gurobi, albeit with a slightly higher failure rate (i.e. 3.3% vs 0% at full accuracy for LP tests, 24.0% vs 20.4%

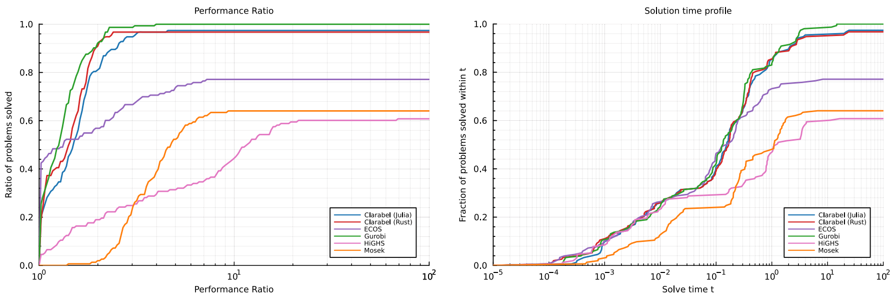


(a) Relative performance profile **(b)** Absolute performance profile

		ClarabelRs	Clarabel	ECOS	Mosek
Shifted GM	Full Acc.	3.25	2.04	4.02	1.0
	Low Acc.	2.2	2.04	4.02	1.0
Failure Rate (%)	Full Acc.	2.2	0.0	2.2	0.0
	Low Acc.	0.0	0.0	2.2	0.0

(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 7 Performance profiles for the CBLIB Exponential Cone problem set



(a) Relative performance profile **(b)** Absolute performance profile

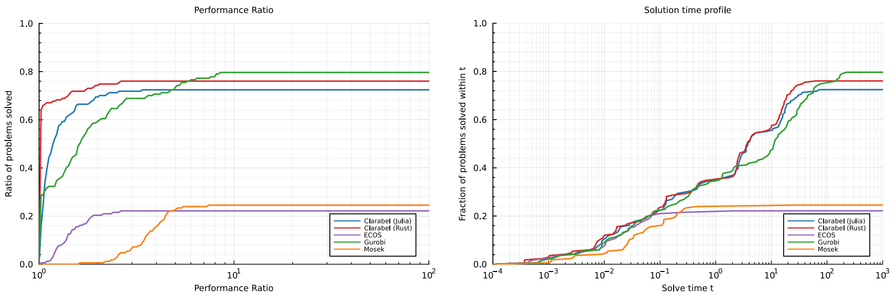
		ClarabelRs	Clarabel	ECOS	Gurobi	HiGHS	Mosek
Shifted GM	Full Acc.	1.67	1.57	8.8	1.0	30.37	23.47
	Low Acc.	1.2	1.23	4.22	1.0	30.37	2.9
Failure Rate (%)	Full Acc.	3.3	2.6	22.9	0.0	39.2	35.9
	Low Acc.	0.7	0.7	11.1	0.0	39.2	2.0

(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 8 Performance profiles for the LP Optimal Power Flow problem set

for SOCP tests). We note however that Gurobi only converges for some of the largest SOCP problems after more than 200 iterations, which we have set as our own maximum iteration limit. Other solvers in the SOCP benchmark tests fared substantially worse in terms of failure rate at full accuracy, although all solvers struggled to some extent on this test set.

We note also that our success rates differ slightly between our Julia and Rust implementations, even though the implementation of our algorithm is (nearly) identical between the cases. We believe that this difference is attributable to minor differences in compiled code vectorizations and optimizations, which in some very difficult problems lead to slight differences in behavior at high accuracy.



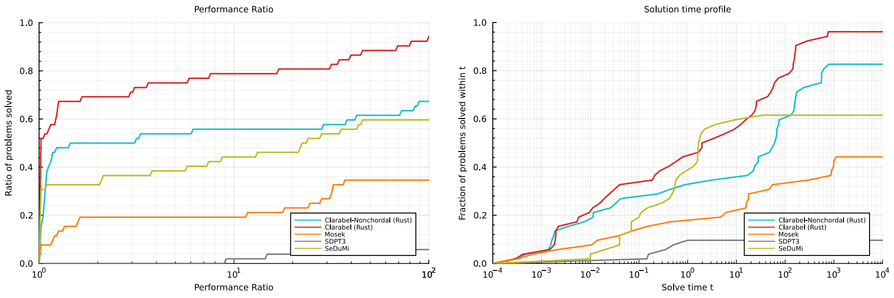
(a) Relative performance profile

(b) Absolute performance profile

		ClarabelRs	Clarabel	ECOS	Gurobi	Mosek
Shifted GM	Full Acc.	1.0	1.16	9.54	1.32	8.53
	Low Acc.	1.0	1.04	9.14	1.77	1.68
Failure Rate (%)	Full Acc.	24.0	27.5	77.8	20.4	75.4
	Low Acc.	2.4	2.4	59.3	0.0	0.0

(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 9 Performance profiles for the SOCP Optimal Power Flow problem set



(a) Relative performance profile

(b) Absolute performance profile

		ClarabelRs	ClarabelRs (Nonchordal)	Mosek	SDPT3	SeDuMi
Shifted GM	Full Acc.	1.0	4.08	25.36	103.04	2.91
	Low Acc.	6.2	19.0	8.75	214.44	1.0
Failure Rate (%)	Full Acc.	3.8	17.3	55.8	90.4	38.5
	Low Acc.	1.9	11.5	1.9	69.2	0.0

(c) Benchmark timings as shifted geometric mean and failure rates

Fig. 10 Performance profiles for the SDPLIB Semidefinite Programming problem

5.3 Semidefinite program benchmarks

Finally, we test performance on large-scale SDPs using the SDPLIB benchmark collection [102]. We consider 64 problems from this collection, where problems are included if they are known to be feasible and could be solved on our testing platform using the reference solver Mosek. Since some of these problems are large, we set a time limit of 30 minutes for each problem.

For this set of benchmarks we test only our Rust implementation of Clarabel and use the supernodal factorization method of [76]. We provide results for Clarabel both with and without the chordal decomposition described in §2.5 enabled. For this test we also include results for the solvers SeDuMi [103] and SDPT3 [104], two older interior-point solvers for SDPs commonly used in MATLAB, which we access via their respective JuMP [83] interfaces in Julia. (See Fig. 10)

For this set of benchmarks we find that Clarabel outperforms the other solvers overall, particularly when considering problems solved to full accuracy solutions. Clarabel also outperforms Mosek, sometimes very substantially, for those test cases with a considerable amount of sparsity that are amenable to chordal decomposition. We include in Appendix A a detailed breakdown of the results for this benchmark set, including performance both with and without chordal decomposition enabled. We also observe that the solver SeDuMi does exceptionally well on some problems in this test set, and in several cases is the fastest solver by a substantial margin. We believe that this performance difference is attributable to the linear solve strategy employed. SeDuMi reduces its KKT system (similar to (20)) to normal form, which can be considerably faster for some problems. We intend to support this approach in Clarabel in future releases.

For problems in which no chordal structure could be identified, there is very little difference in performance between our two solver configurations. However, some problems (e.g. the series of problems ‘ARCH0’–‘ARCH8’) show improvements in computation time of nearly 100x when chordal decomposition is enabled. We note that further improvements are likely possible in our implementation of chordal decomposition, particularly if more sophisticated clique-merging scoring methods are employed.

6 Conclusions

We have presented a novel interior-point solver for conic optimization problems with quadratic objectives. Our method uses a homogeneous embedding inspired by previous work on monotone complementarity problems, but not previously applied to interior-point conic optimization in any widely available solver. We have shown that our method is competitive with state-of-the-art solvers for a wide range of problem classes, and in particular outperforms state-of-the-art solvers in problems with quadratic objectives (QPs), large-scale SOCPs, and SDPs with significant sparsity structure.

Our implementation of Clarabel is available as open-source software in both Rust and Julia, with several other language interfaces, and is available as a standard solver in the CVXPY modelling package. Clarabel already has a growing base of both academic and industrial users and has been downloaded more than 40 million times since its initial release.

A Detailed benchmark results

See Table 1, 2, 3, 4, 5, 6, 7, 8, 9,

Table 1 Solve times and iteration counts for the Maros-Meszaros problem set

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
TAME	3	2	4	3	5	6	6	**3**	1.21e-05	**8.09e-06**	2.55e-05	9.02e-05	6.03e-05	**4.86e-05**	0.000153	0.000271
HS21	5	2	6	2	**9**	11	10	11	6.92e-06	**4.89e-06**	1.5e-05	3.41e-05	6.23e-05	**5.37e-05**	0.00015	0.000375
HS35	4	3	6	5	**7**	9	10	9	8.3e-06	**7.36e-06**	1.69e-05	3.51e-05	**5.81e-05**	6.63e-05	0.000169	0.000316
HS35MOD	4	3	6	5	**12**	15	14	14	5.91e-06	**4.84e-06**	1.5e-05	2.7e-05	**7.1e-05**	7.25e-05	0.00021	0.000378
QPTEST	5	2	7	3	**8**	9	10	**8**	7.73e-06	**6.43e-06**	1.71e-05	3.59e-05	6.19e-05	**5.78e-05**	0.000171	0.000287
HS51	3	5	7	7	**0**	9	2	9	**0**	7.83e-06	6.81e-05	3.19e-05	**4.48e-05**	7.05e-05	0.000136	0.000287
HS52	3	5	7	7	**0**	7	2	5	**0**	8.51e-06	7.75e-05	5.55e-05	**3.86e-05**	5.96e-05	0.000155	0.000278
ZECEVIC2	6	2	8	1	8	8	8	**7**	8.54e-06	**5.95e-06**	1.35e-05	4.96e-05	6.83e-05	**4.76e-05**	0.000108	0.000347
HS268	5	5	25	15	**12**	21	13	-	7.38e-06	**6.97e-06**	1.69e-05	-	**8.85e-05**	0.000146	0.000219	-
S268	5	5	25	15	**12**	21	13	-	7.63e-06	**6.9e-06**	1.96e-05	-	**9.15e-05**	0.000145	0.000255	-
HS76	7	4	14	6	**6**	9	10	8	9.7e-06	**7.87e-06**	1.83e-05	3.73e-05	**5.82e-05**	7.09e-05	0.000183	0.000299
GENHS28	8	10	24	19	**0**	10	2	6	**0**	8.58e-06	8.59e-05	5.91e-05	**5.12e-05**	8.58e-05	0.000172	0.000355
HS53	13	5	17	7	**6**	15	8	7	1.05e-05	**6.31e-06**	2.32e-05	5.31e-05	**6.32e-05**	9.46e-05	0.000186	0.000372
LOTSCHD	19	12	66	6	**9**	21	11	19	1.18e-05	**8.63e-06**	1.92e-05	3.74e-05	**0.000106**	0.000181	0.000211	0.00071
HS118	59	15	93	15	11	**10**	13	12	1.91e-05	**1.74e-05**	2.74e-05	4.79e-05	0.00021	**0.000174**	0.000356	0.000575

Table 1 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
QAFIRO	59	32	115	6	14	14	15	**13**	1.97e-05	**1.89e-05**	3.85e-05	7.55e-05	0.000276	**0.000265**	0.000577	0.000981
DPKLO1	77	133	1575	77	**0**	14	2	6	**0**	0.000189	0.000282	0.000618	**0.000379**	0.00264	0.000564	0.00371
DUAL4	151	75	225	2799	12	12	**11**	14	**0.000192**	0.000233	0.000386	0.000256	**0.00231**	0.00279	0.00425	0.00359
QPCBLEND	157	83	574	83	17	18	20	**15**	**6.49e-05**	9.22e-05	7.07e-05	0.000352	**0.0011**	0.00166	0.00141	0.00528
DUALC1	233	9	1953	45	**11**	33	15	-	**9.12e-05**	9.53e-05	0.000601	-	**0.001**	0.00314	0.00901	-
DUALC2	243	7	1617	28	**11**	-	**11**	-	**8.46e-05**	-	0.000653	-	**0.00093**	-	0.00718	-
QADLITTL	153	97	480	87	**14**	28	16	-	6.4e-05	**6.15e-05**	8.83e-05	-	**0.000896**	0.00172	0.00141	-
QSHARE2B	175	79	773	55	**16**	26	23	-	**7.46e-05**	7.72e-05	8.33e-05	-	**0.00119**	0.00201	0.00191	-
DUAL1	171	85	255	3558	12	15	13	**11**	**0.000254**	0.000296	0.000465	0.00038	**0.00305**	0.00445	0.00605	0.00418
DUAL2	193	96	288	4508	**11**	15	**11**	13	**0.000297**	0.000445	0.000642	0.000419	**0.00327**	0.00667	0.00706	0.00544
DUALC5	294	8	2240	36	**10**	17	13	15	0.000108	0.000104	0.000901	**8.72e-05**	**0.00108**	0.00176	0.0117	0.00131
CVXQP2_S	225	100	274	386	**10**	-	14	-	**9.04e-05**	-	0.000253	-	**0.000904**	-	0.00354	-
DUAL3	223	111	333	6108	**12**	18	**12**	13	**0.00041**	0.000546	0.00086	0.000966	**0.00491**	0.00983	0.0103	0.0126

Table 1 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CVXQP1_S	250	100	348	386	**9**	-	14	-	**0.000102**	-	0.000318	-	**0.000917**	-	0.00445	-
CVXQP3_S	275	100	422	386	**11**	-	14	-	**0.000115**	-	0.000439	-	**0.00126**	-	0.00615	-
QSCAGR7	269	140	560	25	**16**	36	25	-	8.53e-05	**7.98e-05**	9.38e-05	-	**0.00136**	0.00287	0.00235	-
PRIMAL1	86	325	5816	324	**10**	20	19	**10**	0.000539	0.000658	**0.000181**	0.0012	0.00539	0.0132	**0.00344**	0.012
QRECIPE	340	180	912	50	**17**	21	21	23	9.44e-05	**8.77e-05**	0.000115	0.00029	**0.0016**	0.00184	0.00242	0.00668
QPCBOEI2	382	143	1480	143	**20**	-	-	-	**0.000208**	-	-	-	**0.00417**	-	-	-
DUALC8	519	8	4040	36	**10**	-	13	-	**0.000191**	-	0.00274	-	**0.00191**	-	0.0356	-
QSHARE1B	342	225	1376	39	32	52	**31**	-	0.000116	**0.000114**	0.000144	-	**0.00371**	0.00592	0.00446	-
PRIMALC5	286	287	2574	286	14	14	**11**	14	0.000159	0.000172	**0.000129**	0.000596	0.00223	0.0024	**0.00142**	0.00835
VALUES	405	202	606	3822	**13**	-	-	-	**0.000273**	-	-	-	**0.00354**	-	-	-
QSC205	408	203	754	21	19	**18**	22	**18**	0.000108	**9.1e-05**	0.000129	0.000327	0.00206	**0.00164**	0.00283	0.00589
QBEACONF	435	262	3637	27	28	-	**19**	22	**0.000312**	-	0.00037	0.00057	0.00875	-	**0.00704**	0.0125
QBRANDY	469	249	2397	65	**19**	31	25	37	0.000256	**0.000228**	0.000321	0.00068	**0.00486**	0.00707	0.00804	0.0252
PRIMAL2	97	649	8043	648	**8**	16	18	**8**	0.000987	0.0011	**0.000258**	0.00171	0.0079	0.0176	**0.00465**	0.0137
QE226	505	282	2860	964	24	25	26	**22**	**0.000387**	0.000496	0.000456	0.000995	**0.0093**	0.0124	0.0118	0.0219
PRIMAL3	112	745	21548	744	**9**	18	18	10	0.00294	0.00202	**0.00059**	0.00362	0.0264	0.0363	**0.0106**	0.0362
QBORE3D	559	315	1755	78	27	25	26	**21**	0.000224	**0.000215**	0.000327	0.000694	0.00604	**0.00538**	0.00851	0.0146
KSIP	1001	20	19898	20	14	21	18	**12**	0.000881	0.000676	0.000664	**0.000388**	0.0123	0.0142	0.012	**0.00466**

Table 1 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
QGROW7	721	301	3193	357	22	22	**20**	-	0.000319	**0.000277**	0.000527	-	0.00703	**0.0061**	0.0105	-
QFORPLAN	604	421	5006	582	**21**	-	36	-	0.000482	-	**0.000448**	-	**0.0101**	-	0.0161	-
PRIMALC8	511	520	4663	519	**12**	-	-	-	**0.000379**	-	-	-	**0.00454**	-	-	-
QCAPRI	741	353	2237	894	32	-	**30**	-	**0.000387**	-	0.000644	-	**0.0124**	-	0.0193	-
QSCORPIO	746	358	1784	40	**11**	15	22	16	**0.000205**	0.000242	0.000289	0.000789	**0.00226**	0.00363	0.00635	0.0126
QSCFXM1	787	457	3046	733	**26**	-	31	-	**0.000432**	-	0.00065	-	**0.0112**	-	0.0201	-
QBANDM	777	472	2966	41	**21**	36	29	-	**0.000373**	0.000387	0.000591	-	**0.00782**	0.0139	0.0171	-
QSCTAP1	780	480	2172	153	**19**	24	24	24	**0.000249**	0.000274	0.000315	0.000676	**0.00472**	0.00658	0.00757	0.0162
QPCSTAIR	823	467	4323	467	**22**	-	**22**	-	**0.000554**	-	0.000689	-	**0.0122**	-	0.0152	-
QSTAIR	823	467	4323	1018	31	-	**27**	-	**0.000674**	-	0.000908	-	**0.0209**	-	0.0245	-
QPCBOEI1	980	384	4359	384	**17**	43	23	-	**0.000571**	0.000685	0.00064	-	**0.0097**	0.0295	0.0147	-
QSCAGR25	971	500	2054	128	**20**	-	24	-	**0.000269**	-	0.000359	-	**0.00537**	-	0.00863	-
PRIMAL4	76	1489	16032	1488	**10**	21	17	13	0.00201	0.00212	**0.000466**	0.00316	0.0201	0.0446	**0.00792**	0.0411
QSCSD1	837	760	3148	745	**10**	12	17	12	**0.000386**	0.00044	0.000436	0.000889	**0.00386**	0.00528	0.00742	0.0107
QETAMACR	1223	688	3232	4447	**20**	36	42	-	0.00272	0.00322	**0.0017**	-	**0.0545**	0.116	0.0714	-
QGROW15	1545	645	6865	500	22	22	**20**	-	0.00063	**0.000555**	0.0011	-	0.0139	**0.0122**	0.022	-
QFFFFF80	1378	854	7081	1916	**27**	52	56	-	0.0017	**0.00139**	0.00163	-	**0.0458**	0.0725	0.0911	-
MOSARQP2	1500	900	3830	945	**10**	22	16	18	**0.00084**	0.00159	0.000893	0.00273	**0.0084**	0.035	0.0143	0.0491
GOULDQP2	1747	699	2445	697	**14**	-	15	-	**0.000407**	-	0.000522	-	**0.0057**	-	0.00783	-
GOULDQP3	1747	699	2445	1395	**7**	-	10	-	**0.000469**	-	0.00105	-	**0.00328**	-	0.0105	-

Table 1 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations ClarabelRs	ECOS	Gurobi	Mosek	time per iteration(s) ClarabelRs	ECOS	Gurobi	Mosek	total time (s) ClarabelRs	ECOS	Gurobi	Mosek
QSCFXM2	1574	914	6097	1131	**32**	-	61	-	**0.000932**	-	0.00135	-	**0.0298**	-	0.0824	-
QSTANDAT	1538	1075	4210	804	**18**	28	-	20	0.000552	**0.000532**	-	0.00208	**0.00993**	0.0149	-	0.0417
QSCRS8	1659	1169	4351	121	33	**30**	35	31	0.000617	**0.000603**	0.00081	0.00136	0.0204	**0.0181**	0.0283	0.0421
QSCSD6	1497	1350	5666	1404	**13**	17	17	18	**0.000648**	0.000833	0.000901	0.00148	**0.00842**	0.0142	0.0153	0.0267
LASER	2000	1002	6000	3231	**11**	29	23	-	**0.00068**	0.000889	0.00113	-	**0.00748**	0.0258	0.0259	-
QGFRDXPN	1966	1092	3727	162	**22**	-	44	-	**0.000492**	-	0.00086	-	**0.0108**	-	0.0379	-
QSEBA	2057	1028	5902	646	30	-	**29**	-	**0.000636**	-	0.00083	-	**0.0191**	-	0.0241	-
QGROW22	2266	946	10078	852	27	29	**22**	-	0.00101	**0.000841**	0.00156	-	0.0272	**0.0244**	0.0344	-
CVXQP2_M	2250	1000	2749	3984	**10**	-	15	-	**0.00433**	-	0.00741	-	**0.0433**	-	0.111	-
QSHIP04S	1860	1458	5810	56	**15**	36	19	-	**0.000639**	0.000726	0.000746	-	**0.00959**	0.0261	0.0142	-
CVXQP1_M	2500	1000	3498	3984	**10**	-	15	-	**0.00751**	-	0.0147	-	**0.0751**	-	0.22	-
QSCFXM3	2361	1371	9148	1221	**34**	-	70	-	**0.00154**	-	0.00183	-	**0.0522**	-	0.128	-
CVXQP3_M	2750	1000	4247	3984	**12**	-	16	-	**0.00742**	-	0.0204	-	**0.089**	-	0.327	-
Q25FV47	2391	1571	11971	59499	**26**	-	30	-	**0.00856**	-	0.0142	-	**0.222**	-	0.425	-
YAO	2003	2002	6003	2002	-	-	**24**	-	-	-	**0.000562**	-	-	-	**0.0135**	-
QSHELL	2428	1775	5448	34790	**35**	-	-	-	**0.00547**	-	-	-	**0.191**	-	-	-
QSHIP04L	2520	2118	8450	56	**15**	38	20	-	**0.000977**	0.00115	0.00101	-	**0.0147**	0.0437	0.0202	-
QSCTAP2	2970	1880	8594	777	**12**	22	24	15	**0.00127**	0.00157	0.00141	0.00357	**0.0152**	0.0345	0.0338	0.0535
AUG3D	1000	3873	6546	2673	**0**	13	2	10	**0**	0.00381	0.00289	0.00558	**0.00374**	0.0495	0.00578	0.0558
AUG3DC	1000	3873	6546	3873	**0**	-	2	-	**0**	-	0.00327	-	**0.0035**	-	0.00655	-
QSHIP08S	3165	2387	9501	11677	**15**	34	22	-	**0.00266**	0.00411	0.00288	-	**0.0399**	0.14	0.0634	-
QPILOTNO	3487	2172	15569	485	**33**	-	52	-	0.00879	-	**0.00373**	-	0.29	-	**0.194**	-

Table 1 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
MOSARQP1	3200	2500	5922	2545	**10**	23	15	17	**0.00127**	0.00296	0.00145	0.00612	**0.0127**	0.0681	0.0218	0.104
QSCSD8	3147	2750	11334	2510	**11**	18	15	-	**0.0014**	0.00181	0.00187	-	**0.0154**	0.0326	0.028	-
QSCTAP3	3960	2480	11354	1047	**12**	22	24	-	**0.0017**	0.0022	0.00201	-	**0.0204**	0.0484	0.0483	-
QSHIP12S	3914	2763	10941	17403	**15**	38	26	-	**0.00287**	0.00464	0.0038	-	**0.0431**	0.176	0.0988	-
QSIERRA	5279	2036	11354	183	-	51	**23**	-	-	0.00169	**0.00167**	-	-	0.0864	**0.0384**	-
STADAT1	5999	2001	13997	2000	**15**	-	-	-	**0.00128**	-	-	-	**0.0193**	-	-	-
STADAT2	5999	2001	13997	2000	45	20	**18**	23	**0.00147**	0.00202	0.0015	0.00401	0.0661	0.0404	**0.027**	0.0923
AUG3DCQP	4873	3873	10419	3873	**11**	31	17	-	0.00274	0.00535	**0.00229**	-	**0.0301**	0.166	0.0389	-
AUG3DQP	4873	3873	10419	2673	**13**	35	18	-	0.00263	0.00528	**0.00225**	-	**0.0341**	0.185	0.0405	-
QSHIP08L	5061	4283	17085	34965	**16**	35	21	-	0.0123	0.0176	**0.00693**	-	0.197	0.616	**0.146**	-
CONT- 050	7595	2597	17199	2597	**9**	22	10	20	0.00681	0.00734	**0.00535**	0.0124	0.0613	0.161	**0.0535**	0.249
EXDATA	7501	3000	12000	1125750	22	21	**14**	16	**0.0876**	0.732	0.865	0.108	1.93	15.4	12.1	**1.72**
QSHIP12L	6578	5427	21597	62228	**16**	37	23	-	0.017	0.0299	**0.0103**	-	0.272	1.11	**0.237**	-
STCQP1	10246	4097	21532	26603	**7**	31	12	-	**0.0139**	0.0168	0.0201	-	**0.0976**	0.52	0.241	-
STCQP2	10246	4097	21532	26603	**9**	-	13	-	**0.0167**	-	0.0348	-	**0.15**	-	0.452	-
STADAT3	11999	4001	27997	4000	54	**21**	24	24	0.00307	0.00411	**0.00272**	0.00857	0.166	0.0863	**0.0654**	0.206
POWELL20	10000	10000	20000	10000	-	-	**92**	-	-	-	**0.00275**	-	-	-	**0.253**	-
HUES- MOD	10002	10000	30000	10000	**13**	-	14	-	0.0041	-	**0.00257**	-	0.0533	-	**0.036**	-
HUESTIS	10002	10000	30000	10000	**9**	-	15	-	0.00433	-	**0.00247**	-	0.039	-	**0.037**	-
LISWET1	10000	10002	30000	10002	-	-	**23**	-	-	-	**0.00315**	-	-	-	**0.0724**	-

Table 1 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
LISWET10	10000	10002	30000	10002	-	-	**91**	-	-	-	**0.00297**	-	-	-	**0.27**	-
LISWET11	10000	10002	30000	10002	-	-	**39**	-	-	-	**0.00289**	-	-	-	**0.113**	-
LISWET12	10000	10002	30000	10002	-	-	**375**	-	-	-	**0.00295**	-	-	-	**1.11**	-
LISWET2	10000	10002	30000	10002	18	-	**16**	-	0.00399	-	**0.00336**	-	0.0718	-	**0.0537**	-
LISWET3	10000	10002	30000	10002	23	-	**16**	-	0.00395	-	**0.00317**	-	0.0908	-	**0.0507**	-
LISWET4	10000	10002	30000	10002	27	-	**15**	-	0.00375	-	**0.00327**	-	0.101	-	**0.049**	-
LISWET5	10000	10002	30000	10002	**11**	-	14	-	0.00411	-	**0.0033**	-	**0.0452**	-	0.0462	-
LISWET6	10000	10002	30000	10002	20	-	**15**	-	0.00375	-	**0.0032**	-	0.075	-	**0.0481**	-
LISWET7	10000	10002	30000	10002	-	-	**24**	-	-	-	**0.00309**	-	-	-	**0.074**	-
LISWET8	10000	10002	30000	10002	-	-	**199**	-	-	-	**0.00281**	-	-	-	**0.559**	-
LISWET9	10000	10002	30000	10002	-	-	**317**	-	-	-	**0.00284**	-	-	-	**0.901**	-
DTOC3	10000	14999	34995	14997	**0**	-	-	9	**0**	-	-	0.021	**0.0103**	-	-	0.189
AUG2D	10000	20200	40000	19800	**0**	-	2	-	**0**	-	0.0141	-	**0.0239**	-	0.0281	-
AUG2DC	10000	20200	40000	20200	**0**	-	2	-	**0**	-	0.0129	-	**0.0221**	-	0.0257	-
CVXQP2_L	22500	10000	27499	39984	**10**	-	16	-	**0.131**	-	0.83	-	**1.31**	-	13.3	-
CVXQP1_L	25000	10000	34998	39984	**11**	-	16	-	**0.32**	-	2.85	-	**3.51**	-	45.6	-

Table 1 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CVXQP3_L	27500	10000	42497	39984	**11**	-	15	-	**0.214**	-	3.9	-	**2.36**	-	58.5	-
CONT- 100	30195	10197	69399	10197	**9**	19	11	14	0.0865	0.058	**0.0212**	0.065	0.778	1.1	**0.233**	0.909
CONT- 101	30492	10197	69993	2700	**10**	41	11	13	0.0915	0.062	**0.0245**	0.0572	0.915	2.54	**0.269**	0.744
UBH1	24018	18009	60018	6003	16	-	**12**	35	**0.0076**	-	0.00763	0.0226	0.122	-	**0.0915**	0.791
AUG2DCQP	30200	20200	60200	20200	**12**	-	17	-	0.0181	-	**0.0126**	-	0.217	-	**0.214**	-
AUG2DQP	30200	20200	60200	19800	**14**	-	19	-	0.0168	-	**0.0119**	-	0.236	-	**0.225**	-
CONT- 200	120395	40397	278799	40397	**11**	22	**11**	16	0.486	0.505	**0.171**	0.392	5.35	11.1	**1.88**	6.27
CONT- 201	120992	40397	279993	10400	**10**	39	11	18	0.552	0.65	**0.166**	0.303	5.52	25.4	**1.82**	5.45
BOYD1	93279	93261	652246	93261	-	-	**28**	-	-	-	**0.0493**	-	-	-	**1.38**	-
CONT- 300	271492	90597	629993	23100	**11**	29	12	-	0.809	3.01	**0.391**	-	8.9	87.4	**4.69**	-
BOYD2	279794	93263	517047	2	-	-	**70**	-	-	-	**0.126**	-	-	-	**8.79**	-

Table 2 Solve times and iteration counts for the Optimal Control problem set

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations ClarabelRs	ECOS	Gurobi	Mosek	time per iteration(s) ClarabelRs	ECOS	Gurobi	Mosek	total time (s) ClarabelRs	ECOS	Gurobi	Mosek
PENDULUM_3	28	23	83	23	**5**	11	7	11	2.04e-05	**1.9e-05**	4.06e-05	6.33e-05	**0.000102**	0.000209	0.000284	0.000697
FIORDOSEXAMPLE_1	32	27	67	18	**8**	12	15	12	**1.58e-05**	1.73e-05	2.9e-05	5.09e-05	**0.000127**	0.000208	0.000435	0.000611
FIORDOSEXAMPLE_3	32	27	67	18	**8**	-	12	-	**1.76e-05**	-	2.96e-05	-	**0.000141**	-	0.000355	-
FORCESEXAMPLE_1	38	29	83	28	**9**	14	12	11	1.75e-05	**1.67e-05**	2.78e-05	7.64e-05	**0.000158**	0.000234	0.000333	0.00084
FORCESEXAMPLE_2	38	29	83	27	**8**	13	10	10	1.71e-05	**1.66e-05**	2.57e-05	7.6e-05	**0.000136**	0.000216	0.000257	0.00076
FORCESEXAMPLE_3	42	32	92	31	**9**	12	**9**	11	**1.71e-05**	2.25e-05	3.72e-05	8.29e-05	**0.000154**	0.00027	0.000335	0.000912
TOYEXAMPLE_1	42	32	102	31	**7**	20	9	16	2.29e-05	**1.83e-05**	3.27e-05	8e-05	**0.00016**	0.000366	0.000294	0.00128
TOYEXAMPLE_3	42	32	102	31	9	-	**7**	-	**1.94e-05**	-	4.21e-05	-	**0.000174**	-	0.000295	-
HELICOPTER_1	48	30	96	39	**6**	20	11	21	2.87e-05	**1.92e-05**	4.7e-05	7.85e-05	**0.000172**	0.000383	0.000517	0.00165
FIORDOSEXAMPLE_2	52	27	87	18	**7**	13	11	13	2.3e-05	**1.75e-05**	3.35e-05	4.84e-05	**0.000161**	0.000228	0.000368	0.000629
NONLINEARCSTR_3	64	54	204	30	**10**	19	12	12	**2.67e-05**	2.98e-05	4.12e-05	0.000109	**0.000267**	0.000567	0.000494	0.00131
ROBOTARM_1	84	54	174	34	**6**	-	9	-	**4.18e-05**	-	5.21e-05	-	**0.000251**	-	0.000469	-
PENDULUM_1	78	63	243	63	**6**	12	8	**6**	4.11e-05	**3.93e-05**	6.05e-05	0.000175	**0.000246**	0.000472	0.000484	0.00105
PENDULUM_2	78	63	243	60	**7**	14	9	8	4.06e-05	4.11e-05	**4.04e-05**	0.000179	**0.000284**	0.000575	0.000364	0.00143
FORCESEXAMPLE_4	82	62	182	61	**7**	15	11	**7**	**3.23e-05**	3.36e-05	5.53e-05	0.000184	**0.000226**	0.000504	0.000608	0.00129
TOYEXAMPLE_2	82	62	202	61	**7**	18	9	-	**3.33e-05**	3.62e-05	5.47e-05	-	**0.000233**	0.000652	0.000492	-
AIRCRAFT_3	104	84	364	20	**8**	16	13	12	5.39e-05	4.68e-05	**4.38e-05**	0.000109	**0.000431**	0.000749	0.000569	0.00131
DOUBLEINVERTEDPENDULUM_1	104	84	364	50	**7**	17	10	9	6.17e-05	**5.01e-05**	6.71e-05	0.000187	**0.000432**	0.000852	0.000671	0.00169
DOUBLEINVERTEDPENDULUM_2	104	84	364	50	**8**	16	13	10	5.35e-05	5.17e-05	**4.78e-05**	0.000168	**0.000428**	0.000828	0.000622	0.00168
HELICOPTER_3	118	86	278	80	**7**	22	12	-	5.54e-05	6.15e-05	**5.07e-05**	-	**0.000388**	0.00135	0.000608	-
AIRCRAFT_1	144	84	404	20	**9**	13	12	**9**	4.91e-05	5.51e-05	**4.83e-05**	0.000165	**0.000442**	0.000717	0.00058	0.00148
AIRCRAFT_2	144	84	404	40	**9**	13	11	**9**	5.31e-05	5.65e-05	**4.91e-05**	0.000225	**0.000478**	0.000735	0.00054	0.00202

Table 2 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations ClarabelRs	ECOS	Gurobi	Mosek	time per iteration(s) ClarabelRs	ECOS	Gurobi	Mosek	total time (s) ClarabelRs	ECOS	Gurobi	Mosek
AIRCRAFT_4	144	84	404	20	**9**	13	12	**9**	5.04e-05	5.5e-05	**4.97e-05**	0.000161	**0.000453**	0.000715	0.000596	0.00145
DOUBLEINVERTEDPENDULUM_3	144	84	404	50	**6**	21	12	8	6.34e-05	6.11e-05	**6.03e-05**	0.000218	**0.00038**	0.00128	0.000724	0.00175
BALLONPLATE_1	152	77	257	47	**8**	15	10	10	**5.04e-05**	5.94e-05	5.61e-05	0.000249	**0.000403**	0.000891	0.000561	0.00249
BALLONPLATE_2	152	77	257	47	**7**	14	9	**7**	6.04e-05	5.37e-05	**5.36e-05**	0.00026	**0.000423**	0.000751	0.000482	0.00182
BALLONPLATE_3	152	77	257	47	**7**	17	9	9	6.12e-05	6.05e-05	**4.87e-05**	0.000239	**0.000429**	0.00103	0.000438	0.00216
DCMOTOR_1	144	94	424	32	**9**	-	10	-	5.86e-05	-	**5.49e-05**	-	**0.000528**	-	0.000549	-
AIRCRAFT_10	164	104	444	20	**9**	13	13	-	**5.65e-05**	5.74e-05	6.03e-05	-	**0.000509**	0.000747	0.000784	-
AIRCRAFT_11	164	104	444	64	**9**	15	10	-	**5e-05**	7.58e-05	6.82e-05	-	**0.00045**	0.00114	0.000682	-
AIRCRAFT_12	164	104	444	64	**9**	15	10	-	**5.7e-05**	7.45e-05	6.14e-05	-	**0.000513**	0.00112	0.000614	-
TOYEXAMPLE_4	202	152	502	151	**9**	29	**9**	-	**6.83e-05**	8.24e-05	8.98e-05	-	**0.000614**	0.00239	0.000808	-
BALLONPLATE_4	252	127	427	77	**9**	19	13	11	7.32e-05	7.66e-05	**6.42e-05**	0.000304	**0.000659**	0.00146	0.000835	0.00335
HELICOPTER_2	218	166	538	175	**6**	22	14	-	9.97e-05	0.000121	**8.71e-05**	-	**0.000598**	0.00267	0.00122	-
SHELL_1	249	159	559	60	**7**	15	12	8	0.000124	**9.06e-05**	9.93e-05	0.000443	**0.000871**	0.00136	0.00119	0.00354
SHELL_3	249	159	559	60	**11**	20	15	15	9.9e-05	**8.66e-05**	9.69e-05	0.000331	**0.00109**	0.00173	0.00145	0.00496
NONLINEARCSTR_1	244	204	804	120	**10**	-	12	22	**9.51e-05**	-	0.000113	0.000454	**0.000951**	-	0.00135	0.00998
NONLINEARCSTR_2	244	204	804	120	**10**	-	12	22	**8.59e-05**	-	0.000102	0.000407	**0.000859**	-	0.00122	0.00895
DCMOTOR_2	284	184	844	62	**10**	25	**10**	21	0.000127	0.000142	**9.44e-05**	0.000285	0.00127	0.00356	**0.000944**	0.00598
DCMOTOR_5	284	184	844	62	**10**	-	**10**	-	**0.000106**	-	0.00012	-	**0.00106**	-	0.0012	-
DCMOTOR_6	284	184	844	62	**9**	75	11	-	0.000108	0.000121	**9.8e-05**	-	**0.000971**	0.00904	0.00108	-

Table 2 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
ROBOTARM_2	324	204	684	124	**8**	-	13	-	0.000174	-	**0.000123**	-	**0.00139**	-	0.0016	-
SPACECRAFT_1	367	187	807	110	**10**	18	**10**	14	0.00011	**9.92e-05**	0.000113	0.000423	**0.0011**	0.00179	0.00113	0.00593
SPACECRAFT_2	367	187	807	110	**10**	18	**10**	14	0.000111	**9.66e-05**	0.000102	0.000524	0.00111	0.00174	**0.00102**	0.00734
BINARYDISTILLATIONCOLUMN_1	311	266	2666	90	**9**	18	10	13	0.000228	0.000213	**0.000158**	0.000551	0.00205	0.00383	**0.00158**	0.00716
BINARYDISTILLATIONCOLUMN_2	311	266	2666	90	**9**	18	10	13	0.000205	0.000214	**0.000166**	0.000517	0.00185	0.00384	**0.00166**	0.00672
QUADCOPTER_1	382	292	1162	110	**8**	16	13	10	**0.000149**	0.000192	0.000193	0.00071	**0.00119**	0.00308	0.0025	0.0071
QUADCOPTER_6	382	292	1162	172	**8**	17	14	12	**0.000161**	0.000205	0.000199	0.000654	**0.00129**	0.00348	0.00278	0.00785
TOYEXAMPLE_5	402	302	1002	301	10	-	**9**	-	**0.000128**	-	0.000152	-	**0.00128**	-	0.00137	-
SHELL_2	489	309	1109	120	**7**	15	12	9	0.000218	0.000193	**0.000185**	0.000928	**0.00153**	0.0029	0.00222	0.00836
TRIPLEINVERTEDPENDULUM_1	456	366	2166	201	**8**	-	13	16	0.000386	-	**0.000192**	0.001	0.00309	-	**0.0025**	0.016
SPRINGMASS_2	566	286	1646	181	**7**	20	9	18	0.000215	0.000219	**0.000211**	0.000853	**0.0015**	0.00437	0.0019	0.0154
SPRINGMASS_3	566	286	1646	166	**8**	21	11	-	0.000181	0.000216	**0.00017**	-	**0.00145**	0.00453	0.00187	-
TRIPLEINVERTEDPENDULUM_2	636	366	2346	201	**8**	29	14	13	0.000479	0.000314	**0.000196**	0.000989	0.00383	0.00911	**0.00274**	0.0129
TRIPLEINVERTEDPENDULUM_3	636	366	2346	201	**9**	31	16	-	0.000423	0.000306	**0.000201**	-	0.0038	0.0095	**0.00321**	-
QUADCOPTER_5	572	572	2212	220	**27**	-	-	-	**0.000283**	-	-	-	**0.00765**	-	-	-
AIRCRAFT_13	804	504	2204	304	**11**	23	13	-	**0.000228**	0.000405	0.000267	-	**0.00251**	0.00931	0.00348	-
QUADCOPTER_2	752	572	2312	220	**8**	18	14	11	**0.000274**	0.000423	0.000395	0.00118	**0.00219**	0.00761	0.00553	0.013
QUADCOPTER_4	752	572	2312	220	**12**	19	20	16	**0.000266**	0.000311	0.000432	0.00106	**0.00319**	0.00591	0.00865	0.017
SPRINGMASS_4	1126	566	3286	341	**8**	20	11	15	0.000422	0.000503	**0.000372**	0.00136	**0.00338**	0.0101	0.00409	0.0203
DCMOTOR_3	1404	904	4204	302	**11**	-	17	-	**0.000487**	-	0.00051	-	**0.00536**	-	0.00866	-
DCMOTOR_4	1404	904	4204	302	**11**	31	-	-	**0.000423**	0.000687	-	-	**0.00465**	0.0213	-	-

Table 2 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
QUADCOPTER_3	1862	1412	5762	550	**8**	20	14	10	**0.00069**	0.00118	0.000862	0.00279	**0.00552**	0.0237	0.0121	0.0279
NONLINEARCHAIN_13	2457	2397	30857	660	**9**	22	-	-	**0.00352**	0.00405	-	-	**0.0317**	0.0891	-	-
NONLINEARCHAIN_3	2457	2397	30857	660	**9**	22	-	-	**0.00327**	0.00412	-	-	**0.0294**	0.0907	-	-
NONLINEARCHAIN_14	2637	2397	31037	660	**10**	24	-	-	**0.00359**	0.00423	-	-	**0.0359**	0.101	-	-
NONLINEARCHAIN_4	2637	2397	31037	660	**10**	24	-	-	**0.00366**	0.00432	-	-	**0.0366**	0.104	-	-
SPRINGMASS_1	5606	2806	16406	1621	**12**	-	18	-	0.00294	-	**0.00154**	-	0.0353	-	**0.0277**	-
NONLINEARCHAIN_1	9657	9417	123257	2640	**8**	-	16	-	**0.0151**	-	0.0155	-	**0.121**	-	0.248	-
NONLINEARCHAIN_11	9657	9417	123257	2640	**8**	-	16	-	**0.0157**	-	0.0173	-	**0.125**	-	0.277	-
NONLINEARCHAIN_12	10377	9417	123977	2640	**9**	-	29	-	0.0143	-	**0.0133**	-	**0.129**	-	0.385	-
NONLINEARCHAIN_2	10377	9417	123977	2640	**9**	-	29	-	**0.0139**	-	0.0142	-	**0.125**	-	0.413	-

Table 3 Solve times and iteration counts for the SuiteSparse least-squares problem set

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
NYPA_MARAGAL_1_LASSO	60	60	322	32	**9**	15	10	10	**3.53e-05**	3.78e-05	4.66e-05	0.000112	**0.000318**	0.000568	0.000466	0.00112
NYPA_MARAGAL_1_HUBER	96	110	394	32	6	10	8	**5**	5.05e-05	**4.97e-05**	5.8e-05	0.000295	**0.000303**	0.000497	0.000464	0.00148
HB_ASH85_LASSO	255	255	948	85	**8**	19	9	16	**0.000128**	0.00014	0.000166	0.0005	**0.00102**	0.00266	0.00149	0.00799
HB_ASH85_HUBER	255	340	948	85	**6**	**6**	7	12	**0.000149**	0.000161	0.000164	0.000471	**0.000893**	0.000963	0.00115	0.00566
HB_ASH219_LASSO	389	389	997	219	**8**	19	10	16	**0.000147**	0.000194	0.000195	0.00063	**0.00117**	0.00368	0.00195	0.0101
HB_ASH331_LASSO	539	539	1409	331	**7**	19	8	15	**0.000215**	0.0003	0.000323	0.000965	**0.00151**	0.0057	0.00258	0.0145
HB_ABB313_LASSO	665	665	2574	313	**9**	21	10	-	**0.000307**	0.000432	0.000376	-	**0.00276**	0.00908	0.00376	-
HB_ASH219_HUBER	657	742	1533	219	**9**	16	10	16	**0.000211**	0.000295	0.000352	0.00104	**0.0019**	0.00471	0.00352	0.0166
HB_ASH292_LASSO	876	876	3668	292	**8**	23	10	30	**0.000535**	0.000836	0.000731	0.00192	**0.00428**	0.0192	0.00731	0.0575
HB_ASH608_LASSO	984	984	2576	608	**8**	24	10	-	**0.000412**	0.000538	0.000748	-	**0.00329**	0.0129	0.00748	-
HB_ASH292_HUBER	876	1168	3668	292	**6**	11	9	15	0.000629	**0.000568**	0.000646	0.0014	**0.00377**	0.00625	0.00582	0.021
HB_ABB313_HUBER	939	1115	3122	313	**9**	19	10	17	**0.000368**	0.000585	0.000484	0.00168	**0.00331**	0.0111	0.00484	0.0285
HB_ASH331_HUBER	993	1097	2317	331	**8**	16	10	15	**0.000313**	0.000465	0.000552	0.0016	**0.00251**	0.00744	0.00552	0.0241
NYPA_MARAGAL_2_LASSO	1255	1255	6312	555	**7**	15	8	11	**0.000918**	0.00104	0.00124	0.00323	**0.00642**	0.0155	0.00991	0.0355
HB_ASH958_LASSO	1542	1542	4042	958	**7**	24	8	19	**0.000644**	0.000969	0.00131	0.00219	**0.00451**	0.0232	0.0104	0.0415
HB_ILLC1033_LASSO	1673	1673	7032	1033	19	-	**8**	-	**0.000704**	-	0.00115	-	0.0134	-	**0.00917**	-
HB_WELL1033_LASSO	1673	1673	7045	1033	13	-	**7**	-	**0.000735**	-	0.00128	-	0.00955	-	**0.00898**	-

Table 3 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
NYPA_MARAGAL_2_HUBER	1665	2015	7132	555	**8**	16	10	12	**0.00102**	0.00106	0.00133	0.00312	**0.00817**	0.017	0.0133	0.0375
HB_ASH608_HUBER	1824	2012	4256	608	**9**	18	10	16	**0.000685**	0.000953	0.00111	0.00243	**0.00617**	0.0172	0.0111	0.0388
HB_ASH958_HUBER	2874	3166	6706	958	**9**	18	11	18	**0.00104**	0.00171	0.00165	0.00386	**0.00936**	0.0308	0.0181	0.0695
HB_ILLC1033_HUBER	3099	3419	9884	1033	7	13	8	**6**	**0.00175**	0.00217	0.00181	0.00523	**0.0122**	0.0282	0.0145	0.0314
HB_WELL1033_HUBER	3099	3419	9897	1033	7	11	8	**6**	**0.00131**	0.00223	0.0018	0.00549	**0.00916**	0.0245	0.0144	0.0329
HB_ILLC1850_LASSO	3274	3274	13334	1850	13	-	**8**	-	**0.0018**	-	0.00285	-	0.0234	-	**0.0228**	-
HB_WELL1850_LASSO	3274	3274	13453	1850	12	-	**7**	-	**0.00183**	-	0.00326	-	**0.0219**	-	0.0228	-
NYPA_MARAGAL_3_LASSO	3410	3410	23521	1690	**7**	15	8	9	0.00946	0.00715	**0.00701**	0.0151	0.0662	0.107	**0.0561**	0.135
NYPA_MARAGAL_4_LASSO	4032	4032	32819	1964	**6**	18	7	10	0.0127	0.0167	**0.0124**	0.0172	**0.0761**	0.3	0.0868	0.172
NYPA_MARAGAL_3_HUBER	5070	5930	26841	1690	**8**	14	10	11	0.0126	0.00846	**0.00663**	0.0167	0.101	0.118	**0.0663**	0.184
HB_ILLC1850_HUBER	5550	6262	17886	1850	**7**	12	8	13	**0.00298**	0.00508	0.00371	0.0116	**0.0209**	0.061	0.0297	0.151
HB_WELL1850_HUBER	5550	6262	18005	1850	**7**	12	8	13	**0.00251**	0.00477	0.0037	0.0114	**0.0176**	0.0573	0.0296	0.148
NYPA_MARAGAL_4_HUBER	5892	6926	36539	1964	**7**	15	9	14	0.0189	0.0176	**0.0106**	0.0192	0.132	0.264	**0.0957**	0.269
NYPA_MARAGAL_5_LASSO	11294	11294	111025	4654	**8**	26	10	9	0.0477	0.13	**0.0426**	0.0904	**0.382**	3.38	0.426	0.814

Table 3 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations ClarabelRs	ECOS	Gurobi	Mosek	time per iteration(s) ClarabelRs	ECOS	Gurobi	Mosek	total time (s) ClarabelRs	ECOS	Gurobi	Mosek
NYPA_MARAGAL_5_HUBER	13962	17282	116361	4654	**8**	20	11	12	0.0732	0.133	**0.0402**	0.0893	0.586	2.66	**0.442**	1.07
NYPA_MARAGAL_6_LASSO	41559	41559	599557	21255	**7**	-	9	8	**0.571**	-	0.575	0.861	**4**	-	5.17	6.89
NYPA_MARAGAL_6_HUBER	63765	73917	643969	21255	**7**	-	11	12	0.695	-	**0.579**	0.959	**4.87**	-	6.37	11.5
PEREYRA_LANDMARK_LASSO	77360	77360	1229616	71952	**8**	-	9	-	**0.174**	-	0.25	-	**1.39**	-	2.25	-
NYPA_MARAGAL_7_LASSO	99973	99973	1353638	46845	**7**	-	8	-	**1.22**	-	1.44	-	**8.56**	-	11.5	-
NYPA_MARAGAL_8_HUBER	99636	174713	1474475	33212	**8**	-	11	16	1.28	-	**0.962**	1.13	**10.2**	-	10.6	18.1
NYPA_MARAGAL_7_HUBER	140535	167099	1434762	46845	11	-	**10**	-	1.22	-	**1.19**	-	13.4	-	**11.9**	-
NYPA_MARAGAL_8_LASSO	183366	183366	1641935	33212	**7**	-	9	-	**1.42**	-	1.49	-	**9.93**	-	13.4	-
PEREYRA_LANDMARK_HUBER	215856	218560	1506608	71952	**9**	-	10	-	**0.268**	-	0.548	-	**2.42**	-	5.48	-
ANSYS_DELOR64K_HUBER	194157	1979502	975735	64719	**7**	-	10	14	0.566	-	**0.386**	1.1	3.96	-	**3.86**	15.4
ANSYS_DELOR338K_HUBER	1029708	1916766	5927779	343236	8	-	8	**4**	**1.24**	-	1.57	5.74	**9.94**	-	12.6	23
ANSYS_DELOR295K_HUBER	887202	2711130	3879993	295734	8	-	8	**4**	**1.12**	-	2.65	5.95	**8.97**	-	21.2	23.8
ANSYS_DELOR338K_LASSO	2117352	2117352	8103067	343236	**7**	-	**7**	-	**1.89**	-	3.01	-	**13.2**	-	21.1	-
ANSYS_DELOR64K_LASSO	3635409	3635409	7858239	64719	**8**	-	19	-	2.3	-	**1.54**	-	**18.4**	-	29.3	-
ANSYS_DELOR295K_LASSO	3943590	3943590	9992769	295734	**8**	-	9	-	**2.79**	-	4.1	-	**22.3**	-	36.9	-

Table 4 Solve times and iteration counts for the NETLIB Feasible LP problem set

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
AFIRO	78	51	153	0	**8**	9	9	10	2.23e-05	**1.51e-05**	4.53e-05	9.8e-05	0.000178	**0.000136**	0.000408	0.00098
KB2	120	68	390	0	16	17	17	**14**	4.09e-05	**3.32e-05**	5.05e-05	7.53e-05	0.000654	**0.000565**	0.000858	0.00105
SC50A	128	78	238	0	9	11	11	**8**	3.68e-05	**2.52e-05**	5.15e-05	0.000112	0.000331	**0.000278**	0.000567	0.0009
SC50B	128	78	226	0	**8**	10	9	10	3.71e-05	**2.42e-05**	6.29e-05	9.92e-05	0.000297	**0.000242**	0.000566	0.000992
BLEND	188	114	636	0	11	12	**10**	17	5.72e-05	**5.11e-05**	0.000108	0.000216	0.00063	**0.000613**	0.00108	0.00367
ADLITTLE	194	138	562	0	**12**	**12**	15	13	6.21e-05	**4.34e-05**	8.23e-05	0.000224	0.000746	**0.000521**	0.00123	0.00291
SHARE2B	258	162	939	0	**12**	13	17	19	7.56e-05	**6.64e-05**	0.00011	0.000252	0.000907	**0.000863**	0.00186	0.0048
SC105	268	163	503	0	**10**	12	11	11	7.13e-05	**4.85e-05**	0.000103	0.000225	0.000713	**0.000582**	0.00113	0.00248
STOCFOR1	282	165	666	0	16	**15**	**15**	39	6.75e-05	**5.63e-05**	0.000116	0.000201	0.00108	**0.000844**	0.00174	0.00785
SCAGR7	314	185	650	0	**15**	**15**	18	16	6.16e-05	**5.13e-05**	9.68e-05	0.000254	0.000924	**0.000769**	0.00174	0.00407
RECIPE	364	204	960	0	10	11	11	**7**	9.03e-05	**6.99e-05**	0.00012	0.000461	0.000903	**0.000768**	0.00132	0.00323
SHARE1B	370	253	1432	0	**22**	88	23	45	0.000127	**7.99e-05**	0.000135	0.000328	**0.00279**	0.00703	0.00311	0.0148
NUG05	435	225	1275	0	**7**	**7**	**7**	**7**	0.000485	0.000491	**0.000424**	0.000818	0.0034	0.00344	**0.00297**	0.00573
BEACONFD	468	295	3703	0	**9**	10	13	**9**	0.000318	**0.000273**	0.000358	0.000851	0.00286	**0.00273**	0.00466	0.00766
ISRAEL	490	316	2759	0	**19**	23	23	32	0.000223	**0.000134**	0.000364	0.000562	0.00424	**0.00308**	0.00838	0.018
BRANDY	523	303	2505	0	**15**	18	18	25	0.000227	**0.000194**	0.000323	0.000558	**0.00341**	0.00348	0.00581	0.014
SC205	522	317	982	0	**13**	**13**	**13**	**13**	0.000122	**9.19e-05**	0.000161	0.000298	0.00159	**0.0012**	0.00209	0.00388

Table 4 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
LOTFI	519	366	1502	0	19	32	**18**	**18**	0.000163	**8.48e-05**	0.000218	0.000507	0.00309	**0.00272**	0.00393	0.00912
BORE3D	578	334	1793	0	**19**	20	20	29	0.000208	**0.000173**	0.000331	0.000532	0.00395	**0.00347**	0.00662	0.0154
VTP_BASE	608	346	1461	0	31	34	73	**20**	0.000146	**8.7e-05**	0.000249	0.000589	0.00453	**0.00296**	0.0181	0.0118
GROW7	721	301	3193	0	13	**12**	15	**12**	0.000274	**0.000193**	0.000453	0.00126	0.00356	**0.00232**	0.0068	0.0151
E226	695	472	3240	0	22	24	**21**	31	0.000266	**0.000251**	0.000441	0.000682	**0.00585**	0.00601	0.00925	0.0212
BANDM	777	472	2966	0	**18**	21	**18**	22	0.000281	**0.00025**	0.000573	0.000877	**0.00505**	0.00525	0.0103	0.0193
SCORPION	854	466	2000	0	12	12	**11**	**11**	0.000232	**0.000161**	0.000338	0.000608	0.00279	**0.00193**	0.00372	0.00668
NUG06	858	486	2718	0	**7**	**7**	**7**	**7**	0.00204	0.00179	**0.00161**	0.00227	0.0143	0.0125	**0.0113**	0.0159
CAPRI	870	482	2495	0	21	22	**16**	18	0.000261	**0.000225**	0.000591	0.000928	0.00548	**0.00494**	0.00945	0.0167
SCFXM1	930	600	3332	0	**19**	21	20	**19**	0.000356	**0.000287**	0.000559	0.00103	0.00676	**0.00603**	0.0112	0.0196
STAIR	970	614	4617	0	22	23	17	**16**	0.000482	**0.000442**	0.000805	0.00202	0.0106	**0.0102**	0.0137	0.0324
SCSD1	837	760	3148	0	10	11	**7**	13	0.00027	**0.000193**	0.000485	0.000844	0.0027	**0.00212**	0.00339	0.011
TUFF	985	628	5213	0	21	-	**13**	18	**0.000575**	-	0.000811	0.00136	0.0121	-	**0.0105**	0.0245
SCTAP1	960	660	2532	0	26	25	**17**	24	0.0002	**0.000159**	0.000345	0.000712	0.00521	**0.00396**	0.00586	0.0171
AGG	1103	615	3477	0	45	46	**29**	47	0.000395	**0.000208**	0.000668	0.000605	0.0178	**0.00956**	0.0194	0.0285
SCAGR25	1142	671	2396	0	**17**	18	18	21	0.000223	**0.00018**	0.000351	0.000984	0.00379	**0.00323**	0.00631	0.0207

Table 4 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
DEGEN2	1201	757	4958	0	**11**	12	13	**11**	0.000932	**0.000892**	0.00127	0.00434	**0.0102**	0.0107	0.0166	0.0477
AGG2	1274	758	5498	0	26	28	**17**	32	0.00094	**0.000794**	0.00106	0.00148	0.0245	0.0222	**0.018**	0.0474
AGG3	1274	758	5514	0	25	27	**17**	29	0.000905	**0.000786**	0.00105	0.00161	0.0226	0.0212	**0.0178**	0.0466
ETAMACRO	1351	816	3488	0	**25**	47	**25**	36	0.000987	**0.000787**	0.000996	0.00168	**0.0247**	0.037	0.0249	0.0605
GROW15	1545	645	6865	0	**13**	**13**	15	**13**	0.000593	**0.000394**	0.000974	0.00301	0.00771	**0.00512**	0.0146	0.0392
NUG07	1533	931	5145	0	11	11	**9**	**9**	0.00454	0.00567	**0.00328**	0.00709	0.05	0.0624	**0.0295**	0.0638
FFFFF800	1552	1028	7429	0	**36**	45	47	-	0.00114	**0.00102**	0.00105	-	**0.0412**	0.0459	0.0492	-
FINNIS	1597	1064	3860	0	36	32	24	**23**	0.000424	**0.000324**	0.000859	0.00156	0.0153	**0.0104**	0.0206	0.0358
PILOT4	1692	1123	6546	0	38	71	94	**30**	0.0008	**0.000602**	0.000903	0.00223	**0.0304**	0.0427	0.0848	0.067
SCSD6	1497	1350	5666	0	13	13	**8**	13	0.000426	**0.000323**	0.000781	0.00144	0.00554	**0.00419**	0.00625	0.0187
STANDATA	1737	1274	4608	0	**12**	15	15	21	0.000514	**0.000362**	0.000674	0.00129	0.00617	**0.00542**	0.0101	0.0271
SCRS8	1765	1275	4563	0	28	29	**25**	55	0.000512	**0.000362**	0.000731	0.00135	0.0143	**0.0105**	0.0183	0.0743
SCFXM2	1860	1200	6669	0	22	23	**20**	**20**	0.000711	**0.000585**	0.0011	0.00202	0.0156	**0.0135**	0.022	0.0403
STANDMPS	1845	1274	5256	0	17	18	**15**	26	0.000522	**0.000387**	0.000776	0.00135	0.00888	**0.00696**	0.0116	0.035

Table 4 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
FIT1D	2099	1049	15502	0	28	30	**17**	29	0.00092	**0.000719**	0.00115	0.00104	0.0258	0.0216	**0.0196**	0.0301
GFRD_PNC	2034	1160	3863	0	**14**	17	22	32	0.000411	**0.000313**	0.000693	0.00122	0.00575	**0.00532**	0.0153	0.039
GROW22	2266	946	10078	0	14	14	15	**13**	0.000852	**0.000565**	0.00131	0.00466	0.0119	**0.00791**	0.0197	0.0606
STANDGUB	1848	1383	4825	0	**12**	15	15	19	0.000518	**0.000406**	0.000799	0.00139	0.00621	**0.00609**	0.012	0.0263
SHIP04S	1908	1506	5906	0	17	**16**	**16**	26	0.000595	**0.000435**	0.000941	0.00134	0.0101	**0.00695**	0.0151	0.0348
BNL1	2229	1586	7118	0	57	61	**28**	34	0.000904	**0.000785**	0.00117	0.0021	0.0515	0.0479	**0.0327**	0.0714
PEROLD	2309	1506	7832	0	**53**	-	86	-	**0.00155**	-	0.00157	-	**0.0824**	-	0.135	-
MODSZK1	2305	1620	4786	0	27	32	**21**	29	0.000634	**0.000387**	0.00128	0.00202	0.0171	**0.0124**	0.0269	0.0586
NUG08	2544	1632	8928	0	9	9	8	**7**	0.00983	0.0194	**0.00757**	0.0163	0.0885	0.175	**0.0606**	0.114
QAP8	2544	1632	8928	0	9	9	8	**7**	0.00963	0.0188	**0.00754**	0.0146	0.0866	0.17	**0.0603**	0.102
SHELL	2430	1777	5452	0	**17**	18	27	**17**	0.000598	**0.000535**	0.00118	0.00221	0.0102	**0.00962**	0.0318	0.0376
FIT1P	2703	1677	11944	0	18	23	**16**	23	0.0008	**0.000569**	0.00127	0.00182	0.0144	**0.0131**	0.0203	0.0419
25FV47	2697	1876	12581	0	27	31	27	**25**	0.00233	0.00225	**0.00224**	0.00365	0.0629	0.0697	**0.0606**	0.0913
SCFXM3	2790	1800	10006	0	22	24	21	**20**	0.00109	**0.000871**	0.00157	0.00285	0.024	**0.0209**	0.033	0.057
SHIP04L	2568	2166	8546	0	**17**	19	19	22	0.000794	**0.000648**	0.000933	0.00196	0.0135	**0.0123**	0.0177	0.043
MAROS	2812	1966	12103	0	28	80	**27**	31	0.00218	**0.00179**	0.00203	0.00337	0.061	0.143	**0.0548**	0.105
GANGES	3412	1706	9040	0	31	33	**15**	22	0.00108	**0.000893**	0.00184	0.00402	0.0334	0.0295	**0.0277**	0.0884
WOOD1P	2839	2595	72811	0	16	18	**10**	**10**	0.00573	0.00542	**0.00412**	0.0138	0.0917	0.0975	**0.0412**	0.138
SHIP08S	3245	2467	9661	0	**16**	17	17	22	0.00105	**0.000694**	0.00126	0.00236	0.0168	**0.0118**	0.0214	0.0519
PILOT_JA	3458	2267	17495	0	**61**	-	77	-	0.0103	-	**0.00288**	-	0.629	-	**0.222**	-
SCSD8	3147	2750	11334	0	11	12	**8**	12	0.000904	**0.00066**	0.00143	0.00315	0.00994	**0.00792**	0.0114	0.0378
SCTAP2	3590	2500	9834	0	**16**	**16**	**16**	23	0.000903	**0.000795**	0.00134	0.00283	0.0145	**0.0127**	0.0215	0.0651

Table 4 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
PILOTNOV	3761	2446	16117	0	21	23	**20**	26	0.0129	0.00365	**0.00342**	0.00635	0.271	0.084	**0.0684**	0.165
DEGEN3	4107	2604	28036	0	14	15	18	**11**	0.0136	0.00807	**0.00729**	0.021	0.19	**0.121**	0.131	0.231
PILOT_WE	3864	2928	12407	0	66	87	**37**	67	0.00157	**0.0013**	0.00208	0.00365	0.104	0.113	**0.0771**	0.245
SHIP12S	4020	2869	11153	0	21	23	**20**	27	0.00118	**0.000821**	0.00127	0.00273	0.0248	**0.0189**	0.0253	0.0736
CZPROB	4491	3562	14270	0	43	45	30	**26**	0.00131	**0.000976**	0.00159	0.00338	0.0563	**0.0439**	0.0478	0.088
SCTAP3	4820	3340	13074	0	16	16	**15**	25	0.00128	**0.00113**	0.0019	0.00371	0.0205	**0.0181**	0.0284	0.0929
STOCFOR2	5202	3045	12402	0	29	26	**20**	37	0.00149	**0.00131**	0.00243	0.00267	0.0432	**0.0341**	0.0485	0.0988
SIERRA	5978	2735	12752	0	18	19	**17**	28	0.00141	**0.00109**	0.00196	0.00459	0.0254	**0.0207**	0.0333	0.129
CYCLE	5344	3371	24675	0	38	-	**29**	-	0.00396	-	**0.00388**	-	0.151	-	**0.113**	-
SHIP08L	5141	4363	17245	0	18	19	**17**	23	0.00198	**0.00119**	0.002	0.00404	0.0356	**0.0226**	0.034	0.0929
BNL2	6810	4486	19482	0	34	44	34	**29**	0.019	0.00513	**0.00412**	0.00844	0.646	0.226	**0.14**	0.245
PILOT	7341	4860	50275	0	65	-	**54**	-	0.0252	-	**0.00797**	-	1.64	-	**0.43**	-
SHIP12L	6684	5533	21809	0	29	28	**21**	27	0.00224	**0.00148**	0.00247	0.00477	0.0649	**0.0415**	0.0519	0.129
D6CUBE	6599	6184	43888	0	-	-	22	**12**	-	-	**0.00514**	0.0225	-	-	**0.113**	0.27
KEN_07	9630	3602	15608	0	17	**16**	17	20	0.00188	**0.00127**	0.00276	0.00663	0.0319	**0.0203**	0.047	0.133
D2Q06C	8002	5831	38912	0	**32**	35	33	44	0.0298	0.011	**0.00761**	0.0125	0.952	0.384	**0.251**	0.548
GREENBEB	8277	5598	36955	0	66	67	**39**	50	0.028	0.00906	**0.00601**	0.0099	1.85	0.607	**0.235**	0.495
GREENBEA	8280	5598	36958	0	**42**	**42**	50	-	0.0275	0.00965	**0.00584**	-	1.16	0.405	**0.292**	-
CRE_C	9479	6411	22388	0	27	33	29	**25**	0.00333	**0.00241**	0.00445	0.00986	0.0898	**0.0797**	0.129	0.247
PILOT87	10288	6680	83207	0	**103**	-	-	-	**0.0423**	-	-	-	**4.35**	-	-	-
WOODW	9516	8418	45905	0	26	32	24	**18**	0.00534	**0.0042**	0.00501	0.0111	0.139	0.134	**0.12**	0.2

Table 4 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CRE_A	10764	7248	25416	0	**22**	26	29	32	0.00387	**0.00292**	0.00463	0.00971	0.0852	**0.0759**	0.134	0.311
TRUSS	9806	8806	36642	0	20	20	**15**	22	0.00507	0.00471	**0.00459**	0.0103	0.101	0.0942	**0.0688**	0.227
PDS_02	12803	7716	26421	0	34	32	**23**	24	0.00384	**0.00336**	0.0052	0.0129	0.13	**0.107**	0.12	0.31
NUG12	12048	8856	47160	0	20	20	**13**	16	0.122	0.72	**0.083**	0.127	2.45	14.4	**1.08**	2.04
QAP12	12048	8856	47160	0	20	20	**13**	16	0.118	0.702	**0.0882**	0.121	2.35	14	**1.15**	1.94
MAROS_R7	12544	9408	154256	0	14	15	**12**	14	0.0688	0.194	**0.0409**	0.0615	0.963	2.91	**0.491**	0.861
80BAU3B	17309	12061	38311	0	37	42	41	**25**	0.00551	**0.00409**	0.00749	0.0162	0.204	**0.172**	0.307	0.404
DFL001	18314	12230	47875	0	-	58	42	**33**	-	0.6	**0.0392**	0.0582	-	34.8	**1.65**	1.92
FIT2D	21049	10524	150066	0	**24**	27	27	27	0.00872	0.00822	0.0132	**0.0079**	**0.209**	0.222	0.357	0.213
FIT2P	24025	13525	71309	0	**21**	23	**21**	22	0.00755	**0.00473**	0.00736	0.0157	0.158	**0.109**	0.155	0.346
NUG15	28605	22275	117225	0	24	24	18	**17**	0.606	7.61	**0.381**	0.777	14.5	183	**6.86**	13.2
QAP15	28605	22275	117225	0	24	24	**17**	**17**	0.622	9.31	**0.384**	0.76	14.9	224	**6.53**	12.9
OSA_07	26185	25067	169879	0	**22**	24	26	30	0.0207	0.0149	**0.011**	0.0253	0.456	0.357	**0.285**	0.758
STOCFOR3	40216	23541	96262	0	42	53	**31**	51	0.0151	**0.0099**	0.0155	0.0241	0.635	0.524	**0.481**	1.23
PDS_06	48472	29351	101811	0	46	45	34	**32**	0.132	0.0947	**0.0305**	0.0744	6.06	4.26	**1.04**	2.38
KEN_11	57392	21349	91756	0	25	25	**20**	**20**	0.0152	**0.0137**	0.018	0.0351	0.38	**0.342**	0.361	0.703
OSA_14	57134	54797	371894	0	**24**	29	25	34	0.0503	0.0418	**0.0252**	0.079	1.21	1.21	**0.629**	2.69

Table 4 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
PDS_10	82638	49932	173685	0	61	64	41	**38**	0.259	0.38	**0.0753**	0.163	15.8	24.3	**3.09**	6.18
KEN_13	113950	42659	182564	0	27	27	25	**24**	0.0441	**0.0378**	0.0451	0.0858	1.19	**1.02**	1.13	2.06
CRE_D	82874	73948	320562	0	29	32	36	**28**	0.291	0.136	**0.0549**	0.144	8.44	4.36	**1.98**	4.04
NUG20	87840	72600	377400	0	27	-	21	**18**	5.96	-	**3.25**	10.2	161	-	**68.4**	184
CRE_B	86785	77137	337922	0	**30**	32	35	39	0.325	0.192	**0.0581**	0.15	9.75	6.14	**2.03**	5.85
OSA_30	108724	104374	708862	0	**24**	25	25	-	0.125	0.102	**0.0593**	-	3	2.54	**1.48**	-
PDS_20	176937	108175	375710	0	79	-	**47**	**47**	0.862	-	**0.302**	0.594	68.1	-	**14.2**	27.9
OSA_60	253526	243246	1651319	0	**30**	32	36	57	0.438	0.323	**0.152**	0.467	13.2	10.3	**5.46**	26.6
KEN_18	414525	154699	667569	0	-	35	**29**	33	-	0.329	**0.226**	0.45	-	11.5	**6.54**	14.9

Table 5 Solve times and iteration counts for the NETLIB Infeasible LP problem set

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
LPI_ITEST2	22	13	39	0	6	6	8	**4**	1.72e-05	**9.06e-06**	2.38e-05	7.91e-05	0.000103	**5.43e-05**	0.00019	0.000317
LPI_GALENET	30	14	44	0	6	5	9	**4**	1.83e-05	**1.22e-05**	2.37e-05	7.59e-05	0.00011	**6.09e-05**	0.000213	0.000304
LPI_ITEST6	28	17	46	0	5	8	21	**4**	2.17e-05	1.62e-05	**1.3e-05**	8.73e-05	**0.000108**	0.00013	0.000274	0.000349
LPI_BGPRTR	60	40	110	0	8	9	27	**7**	2.77e-05	**1.79e-05**	2.57e-05	0.000102	0.000221	**0.000161**	0.000694	0.000714
LPI_WOODINFE	138	89	243	0	7	8	16	**6**	5.83e-05	**2.52e-05**	4.45e-05	0.000249	0.000408	**0.000202**	0.000712	0.00149
LPI_KLEIN1	162	108	858	0	**18**	**18**	59	20	0.000105	**6.97e-05**	9.95e-05	0.000207	0.00189	**0.00125**	0.00587	0.00414
LPI_FOREST6	202	131	382	0	**8**	9	25	10	7.08e-05	**3.11e-05**	5.55e-05	0.00021	0.000567	**0.00028**	0.00139	0.0021
LPI_EX73A	404	211	668	0	7	**6**	20	16	0.000153	**0.000101**	0.000129	0.000237	0.00107	**0.000607**	0.00258	0.00379
LPI_EX72A	412	215	682	0	7	**6**	20	16	0.000163	**0.000103**	0.000128	0.00027	0.00114	**0.000619**	0.00256	0.00433
LPI_BOX1	492	261	912	0	8	6	19	**4**	0.000147	0.000135	**0.00013**	0.000819	0.00117	**0.000807**	0.00248	0.00328
LPI_QUAL	1032	464	2355	0	60	54	81	**50**	0.0003	**0.000211**	0.000476	0.000544	0.018	**0.0114**	0.0385	0.0272

Table 5 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
LPI_REFINERY	1032	464	2335	0	**19**	20	40	23	0.000327	**0.000232**	0.00047	0.000574	0.00621	**0.00464**	0.0188	0.0132
LPI_VOL1	1032	464	2355	0	52	51	**48**	**48**	0.000311	**0.000227**	0.000485	0.000789	0.0162	**0.0116**	0.0233	0.0379
LPI_KLEIN2	1008	531	5593	0	**14**	16	65	29	0.000579	**0.000372**	0.000465	0.000699	0.00811	**0.00596**	0.0302	0.0203
LPI_BGDBG1	1022	629	2336	0	9	**8**	15	9	0.000346	**0.000246**	0.000553	0.00124	0.00312	**0.00197**	0.0083	0.0111
LPI_PANG	1117	741	3689	0	**24**	**24**	55	25	0.00048	**0.000414**	0.000656	0.0012	0.0115	**0.00993**	0.0361	0.0301
LPI_CHEMCOM	1176	744	2478	0	**7**	8	15	**7**	0.00046	**0.000337**	0.000645	0.00157	0.00322	**0.0027**	0.00968	0.011
LPI_MONDOU2	1393	604	2289	0	**1**	14	27	24	0.00116	**0.000187**	0.000485	0.000707	**0.00116**	0.00262	0.0131	0.017
LPI_BGETAM	1351	816	3488	0	**7**	8	22	8	0.00138	0.00108	**0.00101**	0.00376	0.00968	**0.00865**	0.0222	0.0301
LPI_REACTOR	1674	808	3947	0	29	29	33	**21**	0.000572	**0.000354**	0.000825	0.00169	0.0166	**0.0103**	0.0272	0.0356
LPI_PILOT4I	1692	1123	6546	0	26	34	88	**23**	0.0011	**0.000753**	0.000946	0.00383	0.0287	**0.0256**	0.0832	0.0882
LPI_KLEIN3	2076	1082	14183	0	**18**	**18**	56	29	0.00146	**0.00106**	0.00113	0.0017	0.0263	**0.019**	0.0632	0.0493
LPI_GRAN	5707	2525	23160	0	**10**	14	-	44	0.00827	0.00569	-	**0.00487**	0.0827	**0.0797**	-	0.214
LPI_CERIA3D	7152	4400	24754	0	15	11	32	**10**	0.00404	**0.00298**	0.00373	0.0147	0.0606	**0.0328**	0.119	0.147
LPI_CPLEX1	8447	5224	16389	0	**6**	18	39	23	0.00514	**0.00125**	0.00217	0.00366	0.0308	**0.0224**	0.0846	0.0843
LPI_GREENBEA	8290	5596	36971	0	29	38	40	**18**	0.021	0.0108	**0.00619**	0.0145	0.609	0.409	**0.248**	0.261
LPI_BGINDY	13551	10880	77146	0	**7**	9	13	9	0.026	0.0199	**0.0146**	0.0301	0.182	**0.179**	0.19	0.27
LPI_GOSH	17005	13455	113166	0	-	36	-	**25**	-	0.0321	-	**0.0266**	-	1.16	-	**0.665**

Table 6 Solve times and iteration counts for the LP Optimal Power Flow problem set

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE3_LMBD	25	9	43	2	**8**	19	16	-	1.24e-05	**7.06e-06**	1.12e-05	-	**9.95e-05**	0.000134	0.000179	-
CASE3_LMBD__API	25	9	43	2	**8**	19	12	-	1.52e-05	**8.2e-06**	1.54e-05	-	**0.000121**	0.000156	0.000185	-
CASE3_LMBD__SAD	25	9	43	2	**8**	20	16	-	1.45e-05	**7.75e-06**	1.2e-05	-	**0.000116**	0.000155	0.000192	-
CASE5_PJM	46	16	82	0	7	8	**6**	8	2.44e-05	**1.46e-05**	2.93e-05	5.95e-05	0.000171	**0.000117**	0.000176	0.000476
CASE5_PJM__API	46	16	82	0	9	9	**7**	10	1.98e-05	**1.04e-05**	2.86e-05	4.54e-05	0.000178	**9.36e-05**	0.0002	0.000454
CASE14_IEEE	125	39	236	0	**7**	**7**	**7**	**7**	4.27e-05	**2.68e-05**	5.84e-05	0.000109	0.000299	**0.000188**	0.000409	0.000766
CASE14_IEEE__API	125	39	236	0	11	11	**6**	11	4.03e-05	**2.67e-05**	6.25e-05	6.93e-05	0.000443	**0.000294**	0.000375	0.000762
CASE30_AS	248	77	470	6	**8**	16	13	14	8.02e-05	6.66e-05	**6.52e-05**	0.000105	**0.000641**	0.00107	0.000848	0.00147
CASE30_AS__API	248	77	470	6	**9**	22	14	-	6.78e-05	**5.78e-05**	5.99e-05	-	**0.00061**	0.00127	0.000838	-
CASE30_IEEE	248	77	470	0	8	8	**6**	8	9.04e-05	**5.01e-05**	9.48e-05	0.000134	0.000723	**0.000401**	0.000569	0.00108
CASE30_IEEE__API	248	77	470	0	**7**	8	**7**	9	8.53e-05	**4.89e-05**	8.9e-05	0.000121	0.000597	**0.000391**	0.000623	0.00109
CASE24_IEEE_RTS	273	95	502	22	**9**	23	11	17	7.03e-05	7.51e-05	**6.84e-05**	0.00016	**0.000633**	0.00173	0.000752	0.00271
CASE24_IEEE_RTS__API	273	95	502	22	**11**	26	12	-	7.2e-05	7.07e-05	**6.36e-05**	-	0.000792	0.00184	**0.000763**	-
CASE24_IEEE_RTS__SAD	273	95	502	22	**12**	44	16	-	5.79e-05	6.98e-05	**5.63e-05**	-	**0.000695**	0.00307	0.000901	-
CASE39_EPRI	290	95	537	0	**9**	**9**	11	13	9.04e-05	**5.03e-05**	7.83e-05	0.000128	0.000814	**0.000453**	0.000861	0.00166
CASE39_EPRI__API	290	95	537	0	**9**	10	12	10	7.76e-05	**5.01e-05**	6.87e-05	0.000145	0.000699	**0.000501**	0.000824	0.00145
CASE39_EPRI__SAD	290	95	537	0	18	21	23	**13**	7.82e-05	**4.53e-05**	5.52e-05	0.000138	0.00141	**0.00095**	0.00127	0.00179
CASE57_IEEE	468	144	894	0	**8**	9	9	10	0.000127	**8.31e-05**	0.000137	0.000221	0.00101	**0.000748**	0.00123	0.00221

Table 6 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE57_IEEE__API	468	144	894	0	10	12	**9**	11	0.000121	**7.95e-05**	0.000141	0.000239	0.00121	**0.000954**	0.00127	0.00263
CASE60_C	515	171	974	0	**8**	**8**	9	9	0.000145	**9.76e-05**	0.00013	0.000339	0.00116	**0.000781**	0.00117	0.00305
CASE60_C__API	515	171	974	0	14	19	14	**13**	0.000126	**7.53e-05**	0.000105	0.000288	0.00176	**0.00143**	0.00147	0.00375
CASE73_IEEE_RTS	848	292	1570	66	**9**	27	12	16	0.000208	0.000253	**0.000175**	0.000603	**0.00187**	0.00684	0.0021	0.00965
CASE73_IEEE_RTS__API	848	292	1570	66	**12**	29	13	-	0.000183	0.000247	**0.000178**	-	**0.0022**	0.00716	0.00232	-
CASE73_IEEE_RTS__SAD	848	292	1570	66	**13**	73	18	-	0.000201	0.000254	**0.000153**	-	**0.00261**	0.0185	0.00276	-
CASE89_PEGASE	1156	311	2331	0	9	**8**	11	31	0.000285	**0.000211**	0.000607	0.000502	0.00256	**0.00169**	0.00668	0.0156
CASE89_PEGASE__API	1156	311	2331	0	**14**	**14**	17	31	0.000385	**0.000234**	0.00053	0.000494	0.00539	**0.00327**	0.00902	0.0153
CASE118_IEEE	1143	358	2181	0	12	12	**9**	12	0.000333	**0.000204**	0.000312	0.000655	0.00399	**0.00245**	0.0028	0.00787
CASE118_IEEE__API	1143	358	2181	0	14	14	**10**	16	0.000366	**0.000221**	0.000292	0.000585	0.00512	0.0031	**0.00292**	0.00936
CASE179_GOC	1471	471	2817	0	13	**12**	18	18	0.000311	**0.000237**	0.00033	0.000631	0.00405	**0.00284**	0.00594	0.0114

Table 6 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE179_GOC__API	1471	471	2817	0	**13**	**13**	22	23	0.000396	**0.000244**	0.000318	0.000596	0.00514	**0.00318**	0.00699	0.0137
CASE200_ACTIV	1502	483	2810	31	**10**	21	12	16	0.000332	**0.000308**	0.000473	0.000827	**0.00332**	0.00646	0.00568	0.0132
CASE200_ACTIV__API	1502	483	2810	31	**11**	19	14	31	0.00033	**0.000312**	0.000463	0.00074	**0.00362**	0.00593	0.00648	0.0229
CASE162_IEEE_DTC	1599	458	3145	0	14	14	**10**	17	0.000461	**0.000286**	0.000798	0.000824	0.00645	**0.004**	0.00798	0.014
CASE162_IEEE_DTC__API	1599	458	3145	0	16	14	**12**	16	0.000486	**0.000321**	0.000738	0.000795	0.00778	**0.0045**	0.00885	0.0127
CASE162_IEEE_DTC__SAD	1599	458	3145	0	18	19	18	**16**	0.000445	**0.000357**	0.000638	0.00081	0.00802	**0.00679**	0.0115	0.013
CASE197_SNEM	1572	518	3000	0	9	10	**7**	9	0.000378	**0.000256**	0.000472	0.000907	0.0034	**0.00256**	0.0033	0.00816
CASE197_SNEM__API	1572	518	3000	0	12	13	**8**	16	0.000424	**0.000281**	0.000443	0.00065	0.00509	0.00365	**0.00355**	0.0104
CASE300_IEEE	2490	780	4721	0	**12**	13	15	18	0.000651	**0.00047**	0.000653	0.000967	0.00781	**0.0061**	0.0098	0.0174
CASE300_IEEE__API	2490	780	4721	0	14	**13**	15	16	0.000765	**0.000556**	0.000686	0.0012	0.0107	**0.00723**	0.0103	0.0192
CASE300_IEEE__SAD	2490	780	4721	0	**14**	17	17	19	0.000773	**0.000504**	0.000675	0.00106	0.0108	**0.00856**	0.0115	0.0202
CASE240_PSERC	2567	831	4958	0	**13**	43	14	21	0.000953	**0.000377**	0.000719	0.00122	0.0124	0.0162	**0.0101**	0.0256
CASE240_PSERC__API	2567	831	4958	0	**12**	13	14	15	0.00069	**0.00044**	0.00076	0.00144	0.00829	**0.00571**	0.0106	0.0215
CASE588_SDET	4191	1369	7796	0	14	14	**13**	14	0.00137	**0.000864**	0.00124	0.00198	0.0192	**0.0121**	0.0162	0.0277
CASE588_SDET__API	4191	1369	7796	0	13	**12**	14	13	0.00148	**0.00104**	0.00113	0.00198	0.0192	**0.0125**	0.0158	0.0258
CASE500_GOC	4327	1399	8210	60	**15**	25	16	-	**0.000967**	0.00123	0.00126	-	**0.0145**	0.0307	0.0202	-

Table 6 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations ClarabelRs	iterations ECOS	iterations Gurobi	iterations Mosek	time per iteration(s) ClarabelRs	time per iteration(s) ECOS	time per iteration(s) Gurobi	time per iteration(s) Mosek	total time (s) ClarabelRs	total time (s) ECOS	total time (s) Gurobi	total time (s) Mosek
CASE500_GOC__API	4327	1399	8210	60	**18**	50	**18**	-	**0.00113**	0.00124	0.00127	-	**0.0203**	0.0621	0.0228	-
CASE793_GOC	5535	1803	10299	48	**16**	26	**16**	-	**0.00145**	0.0023	0.00163	-	**0.0232**	0.0597	0.0261	-
CASE793_GOC__API	5535	1803	10299	48	**15**	35	**15**	-	**0.00156**	0.00193	0.00174	-	**0.0235**	0.0675	0.0261	-
CASE1354_PEGASE	11268	3605	21558	0	**13**	**13**	**13**	29	0.00394	**0.00332**	0.00359	0.00521	0.0512	**0.0431**	0.0466	0.151
CASE1354_PEGASE__API	11268	3605	21558	0	**14**	**14**	15	24	0.00435	**0.0024**	0.00342	0.00587	0.0609	**0.0336**	0.0513	0.141
CASE1888_RTE	14678	4709	27820	0	**12**	13	**12**	35	0.00484	**0.00311**	0.00479	0.00641	0.058	**0.0405**	0.0575	0.224
CASE1888_RTE__API	14678	4709	27820	0	**16**	91	**16**	31	0.00586	**0.00227**	0.00425	0.00659	0.0938	0.206	**0.0679**	0.204
CASE1888_RTE__SAD	14678	4709	27820	0	**13**	14	14	27	0.00539	**0.00317**	0.00482	0.00682	0.0701	**0.0443**	0.0675	0.184
CASE1803_SNEM	15041	4828	29036	0	17	18	**11**	31	0.00571	**0.00358**	0.00426	0.00663	0.097	0.0644	**0.0469**	0.205
CASE1803_SNEM__API	15041	4828	29036	0	20	32	**13**	41	0.00637	**0.0036**	0.00411	0.00619	0.127	0.115	**0.0535**	0.254
CASE1951_RTE	15222	4913	28771	0	**14**	**14**	24	27	0.00492	**0.00339**	0.0043	0.00692	0.0689	**0.0474**	0.103	0.187
CASE1951_RTE__API	15222	4913	28771	0	**17**	33	19	35	0.00693	**0.00302**	0.00451	0.00665	0.118	0.0998	**0.0856**	0.233
CASE2312_GOC	17464	5551	33090	42	22	26	**21**	-	0.00616	0.00633	**0.00591**	-	0.136	0.165	**0.124**	-
CASE2312_GOC__API	17464	5551	33090	42	**18**	-	19	-	0.00595	-	**0.00548**	-	0.107	-	**0.104**	-
CASE2383WP_K	17498	5606	32798	0	21	-	**16**	30	0.00792	-	**0.00534**	0.00803	0.166	-	**0.0854**	0.241
CASE2383WP_K__API	17498	5606	32798	0	12	**10**	14	13	0.00676	**0.00511**	0.00543	0.0141	0.0812	**0.0511**	0.076	0.183
CASE2000_GOC	18988	5871	37370	122	**14**	37	16	-	0.00625	**0.00614**	0.00687	-	**0.0875**	0.227	0.11	-

Table 6 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE2000_GOC__API	18988	5871	37370	122	**18**	36	**18**	-	**0.0054**	0.00599	0.00665	-	**0.0972**	0.216	0.12	-
CASE2737SOP_K	19509	6225	36593	0	16	18	**12**	18	0.00871	**0.00468**	0.00647	0.0107	0.139	0.0842	**0.0777**	0.193
CASE2737SOP_K__API	19509	6225	36593	0	**15**	16	**15**	22	0.00733	**0.0049**	0.00623	0.00923	0.11	**0.0783**	0.0934	0.203
CASE2736SP_K	19610	6275	36746	0	**13**	**13**	27	23	0.00673	**0.00495**	0.00544	0.0088	0.0875	**0.0643**	0.147	0.202
CASE2736SP_K__API	19610	6275	36746	0	**19**	25	24	24	0.00945	**0.00482**	0.00562	0.00919	0.179	**0.12**	0.135	0.221
CASE2746WP_K	20042	6481	37414	0	15	17	**13**	29	0.00879	**0.00489**	0.0063	0.00923	0.132	0.0832	**0.0819**	0.268
CASE2746WP_K__API	20042	6481	37414	0	25	**21**	**21**	25	0.00675	**0.00454**	0.00639	0.0101	0.169	**0.0954**	0.134	0.252
CASE2746WOP_K	20128	6484	37639	0	14	15	**13**	21	0.00723	**0.00484**	0.00673	0.0102	0.101	**0.0726**	0.0875	0.215
CASE2746WOP_K__API	20128	6484	37639	0	18	22	**16**	22	0.00983	**0.00514**	0.00636	0.0102	0.177	0.113	**0.102**	0.224
CASE3012WP_K	21631	6969	40424	0	**16**	20	33	26	0.0111	0.00616	**0.00586**	0.0102	0.177	**0.123**	0.193	0.265
CASE3012WP_K__API	21631	6969	40424	0	**24**	81	32	27	0.0106	**0.00569**	0.00597	0.00988	0.254	0.461	**0.191**	0.267
CASE2848_RTE	22083	7135	41734	0	**13**	**13**	**13**	31	0.00774	**0.00494**	0.00672	0.0106	0.101	**0.0642**	0.0874	0.327
CASE2848_RTE__API	22083	7135	41734	0	**15**	16	**15**	32	0.00887	**0.00474**	0.00673	0.0106	0.133	**0.0758**	0.101	0.341
CASE3120SP_K	22164	7111	41482	0	17	25	**16**	27	0.0106	**0.00555**	0.00656	0.0103	0.181	0.139	**0.105**	0.279
CASE3120SP_K__API	22164	7111	41482	0	17	37	**15**	25	0.0109	**0.00619**	0.00658	0.0104	0.186	0.229	**0.0987**	0.259
CASE2868_RTE	22357	7237	42224	0	**14**	15	**14**	32	0.00912	**0.00498**	0.00737	0.0102	0.128	**0.0748**	0.103	0.327
CASE2868_RTE__API	22357	7237	42224	0	**17**	**17**	21	32	0.00943	**0.00471**	0.00637	0.0107	0.16	**0.0801**	0.134	0.343
CASE2853_SDET	23525	7593	44445	0	21	**17**	**17**	24	0.00828	**0.00578**	0.00734	0.0116	0.174	**0.0983**	0.125	0.279
CASE2853_SDET__API	23525	7593	44445	0	42	-	**18**	45	**0.00701**	-	0.0071	0.0111	0.295	-	**0.128**	0.498
CASE3022_GOC	23816	7484	45395	110	**22**	46	**22**	-	0.009	0.00974	**0.0084**	-	0.198	0.448	**0.185**	-

Table 6 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE3022_GOC__API	23816	7484	45395	110	**21**	100	**21**	54	**0.00727**	0.00997	0.0079	0.0166	**0.153**	0.997	0.166	0.895
CASE3022_GOC__SAD	23816	7484	45395	110	**21**	47	23	-	0.00893	0.00978	**0.00861**	-	**0.187**	0.46	0.198	-
CASE2742_GOC	25136	7597	49278	48	**15**	61	16	-	**0.00925**	0.0109	0.0102	-	**0.139**	0.667	0.163	-
CASE2742_GOC__API	25136	7597	49278	48	**19**	78	21	-	**0.00917**	0.0106	0.0102	-	**0.174**	0.831	0.214	-
CASE2742_GOC__SAD	25136	7597	49278	48	15	58	**14**	-	**0.0082**	0.00959	0.0104	-	**0.123**	0.556	0.146	-
CASE3375WP_K	24952	8014	46837	0	16	**15**	33	28	0.0112	**0.0065**	0.00674	0.0122	0.179	**0.0974**	0.223	0.34
CASE3375WP_K__API	24952	8014	46837	0	**18**	29	34	38	0.0102	**0.00608**	0.00678	0.012	0.184	**0.176**	0.231	0.456
CASE3375WP_K__SAD	24952	8014	46837	0	**17**	**17**	30	29	0.00985	**0.00567**	0.00681	0.0119	0.168	**0.0964**	0.204	0.345
CASE2869_PEGASE	25572	7961	49477	0	16	**15**	**15**	37	0.00953	**0.00581**	0.00835	0.0123	0.153	**0.0872**	0.125	0.454
CASE2869_PEGASE__API	25572	7961	49477	0	**16**	**16**	17	38	0.0112	**0.00602**	0.00808	0.012	0.18	**0.0963**	0.137	0.457
CASE4661_SDET	35603	11382	67156	0	**17**	62	18	31	0.0227	**0.00996**	0.0183	0.021	0.386	0.618	**0.329**	0.652
CASE4661_SDET__API	35603	11382	67156	0	**17**	57	19	32	0.0206	**0.0123**	0.0169	0.0226	0.35	0.701	**0.321**	0.724

Table 6 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations ClarabelRs	ECOS	Gurobi	Mosek	time per iteration(s) ClarabelRs	ECOS	Gurobi	Mosek	total time (s) ClarabelRs	ECOS	Gurobi	Mosek
CASE3970_GOC	36084	10994	70485	65	**15**	-	19	-	**0.0137**	-	0.0147	-	**0.206**	-	0.279	-
CASE3970_GOC__API	36084	10994	70485	65	**20**	-	21	-	**0.0141**	-	0.0146	-	**0.282**	-	0.307	-
CASE4020_GOC	37867	11360	74329	23	26	52	**23**	-	**0.0132**	0.0168	0.0173	-	**0.343**	0.871	0.398	-
CASE4020_GOC__API	37867	11360	74329	23	20	-	**19**	-	**0.0165**	-	0.017	-	0.33	-	**0.323**	-
CASE4917_GOC	38604	12210	73532	193	**20**	-	**20**	48	**0.0148**	-	0.0153	0.0226	**0.296**	-	0.307	1.09
CASE4917_GOC__API	38604	12210	73532	193	**22**	-	23	-	0.0143	-	**0.0126**	-	0.315	-	**0.29**	-
CASE4917_GOC__SAD	38604	12210	73532	193	**21**	-	**21**	-	0.0146	-	**0.0137**	-	0.306	-	**0.287**	-
CASE4601_GOC	39625	12208	76838	62	**16**	87	**16**	-	**0.0148**	0.0162	0.016	-	**0.237**	1.41	0.256	-
CASE4601_GOC__API	39625	12208	76838	62	**19**	-	**19**	-	**0.0153**	-	0.0159	-	**0.29**	-	0.301	-
CASE4601_GOC__SAD	39625	12208	76838	62	21	-	**20**	-	0.0175	-	**0.0159**	-	0.368	-	**0.318**	-
CASE4837_GOC	42041	12934	81840	50	**17**	53	18	-	**0.0151**	0.0161	0.0166	-	**0.257**	0.854	0.299	-
CASE4837_GOC__API	42041	12934	81840	50	**21**	-	**21**	45	0.0162	-	**0.0149**	0.0242	0.34	-	**0.312**	1.09
CASE4619_GOC	44438	13116	87440	29	**18**	73	**18**	-	**0.0151**	0.0186	0.0205	-	**0.272**	1.36	0.368	-
CASE4619_GOC__API	44438	13116	87440	29	**20**	-	**20**	-	**0.0203**	-	0.0209	-	**0.405**	-	0.418	-
CASE5658_EPIGRIDS	49549	15204	96379	0	17	17	**14**	39	0.0236	**0.0147**	0.02	0.0281	0.402	**0.249**	0.281	1.1

Table 6 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE5658_EPIGRIDS__API	49549	15204	96379	0	**14**	17	**14**	48	0.025	**0.0143**	0.022	0.0269	0.35	**0.243**	0.308	1.29
CASE6468_RTE	50397	15867	96458	0	**15**	**15**	16	67	0.0242	**0.015**	0.0168	0.0237	0.363	**0.225**	0.268	1.59
CASE6468_RTE__API	50397	15867	96458	0	18	**17**	18	55	0.0223	**0.0123**	0.0175	0.0241	0.401	**0.208**	0.316	1.33
CASE6468_RTE__SAD	50397	15867	96458	0	**15**	17	17	65	0.0242	**0.012**	0.0172	0.0232	0.362	**0.205**	0.293	1.51
CASE6495_RTE	51081	16194	97510	0	**18**	19	22	68	0.0234	**0.0121**	0.0166	0.0255	0.421	**0.23**	0.366	1.74
CASE6495_RTE__API	51081	16194	97510	0	**17**	-	22	54	0.0253	-	**0.0183**	0.0259	0.431	-	**0.403**	1.4
CASE6495_RTE__SAD	51081	16194	97510	0	**18**	19	25	63	0.0246	**0.0133**	0.0158	0.0255	0.444	**0.252**	0.395	1.6
CASE6470_RTE	51140	16236	97583	0	**16**	**16**	18	43	0.0241	**0.0124**	0.0176	0.0251	0.386	**0.198**	0.317	1.08
CASE6470_RTE__API	51140	16236	97583	0	**16**	**16**	**16**	40	0.0258	**0.012**	0.0165	0.0279	0.413	**0.192**	0.265	1.12
CASE6470_RTE__SAD	51140	16236	97583	0	**16**	**16**	20	44	0.0246	**0.0149**	0.0164	0.0255	0.393	**0.239**	0.328	1.12
CASE6515_RTE	51203	16236	97728	0	**18**	25	21	59	0.0234	**0.0117**	0.0155	0.024	0.422	**0.292**	0.326	1.41
CASE6515_RTE__API	51203	16236	97728	0	**18**	**18**	19	47	0.0249	**0.0123**	0.0173	0.0264	0.448	**0.222**	0.329	1.24
CASE6515_RTE__SAD	51203	16236	97728	0	**18**	26	23	52	0.0253	**0.0139**	0.0162	0.0247	0.455	**0.361**	0.372	1.29
CASE7336_EPIGRIDS	63100	19539	122362	0	**16**	18	17	39	0.0327	**0.0191**	0.0265	0.0362	0.523	**0.343**	0.451	1.41
CASE7336_EPIGRIDS__API	63100	19539	122362	0	**17**	19	**17**	40	0.0401	**0.0191**	0.0259	0.0377	0.681	**0.364**	0.44	1.51
CASE10000_GOC	79096	25209	149368	511	25	42	**23**	-	**0.0331**	0.0348	0.0341	-	0.828	1.46	**0.784**	-
CASE10000_GOC__API	79096	25209	149368	511	25	-	**23**	-	0.0337	-	**0.0316**	-	0.841	-	**0.728**	-
CASE8387_PEGASE	81791	24813	159503	0	23	26	**19**	30	0.0508	**0.0259**	0.0555	0.058	1.17	**0.674**	1.05	1.74

Table 6 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE8387_PEGASE__API	81791	24813	159503	0	23	28	**20**	36	0.0444	**0.0225**	0.0529	0.0531	1.02	**0.631**	1.06	1.91
CASE8387_PEGASE__SAD	81791	24813	159503	0	23	26	**18**	34	0.0509	**0.0225**	0.0554	0.0528	1.17	**0.584**	0.997	1.8
CASE9591_GOC	86151	25871	168669	55	**20**	66	**20**	-	**0.0432**	0.0527	0.0562	-	**0.864**	3.48	1.12	-
CASE9591_GOC__API	86151	25871	168669	55	24	-	**22**	-	**0.0461**	-	0.0529	-	**1.11**	-	1.16	-
CASE9241_PEGASE	88693	26735	173507	0	100	-	**22**	56	**0.0287**	-	0.061	0.0642	2.87	-	**1.34**	3.59
CASE9241_PEGASE__API	88693	26735	173507	0	19	22	**16**	50	0.0426	**0.0225**	0.0623	0.0533	0.809	**0.495**	0.997	2.66
CASE10192_EPIGRIDS	92051	27914	180020	697	**18**	-	19	-	**0.0455**	-	0.0537	-	**0.82**	-	1.02	-
CASE10192_EPIGRIDS__API	92051	27914	180020	697	**18**	-	19	-	**0.0437**	-	0.0558	-	**0.786**	-	1.06	-
CASE10480_GOC	99926	29816	196673	276	**21**	87	22	-	**0.0504**	0.0691	0.0643	-	**1.06**	6.01	1.41	-
CASE10480_GOC__API	99926	29816	196673	276	-	-	**20**	-	-	-	**0.0655**	-	-	-	**1.31**	-
CASE13659_PEGASE	120495	38218	230046	0	-	-	**19**	74	-	-	**0.0771**	0.0894	-	-	**1.46**	6.62
CASE13659_PEGASE__API	120495	38218	230046	0	45	-	**16**	37	**0.0475**	-	0.0844	0.09	2.14	-	**1.35**	3.33
CASE20758_EPIGRIDS	182928	56275	355508	1881	**19**	-	20	-	**0.109**	-	0.119	-	**2.07**	-	2.38	-
CASE20758_EPIGRIDS__API	182928	56275	355508	1881	**19**	-	**19**	-	**0.106**	-	0.109	-	**2.01**	-	2.08	-
CASE19402_GOC	184959	55077	364846	249	25	-	**22**	-	**0.113**	-	0.138	-	**2.83**	-	3.04	-
CASE19402_GOC__API	184959	55077	364846	249	26	-	**23**	-	**0.106**	-	0.137	-	**2.77**	-	3.15	-
CASE19402_GOC__SAD	184959	55077	364846	249	23	-	**21**	-	**0.1**	-	0.14	-	**2.31**	-	2.93	-
CASE24464_GOC	210481	63871	408258	348	-	-	**21**	-	-	-	**0.129**	-	-	-	**2.7**	-
CASE24464_GOC__API	210481	63871	408258	348	-	-	**21**	-	-	-	**0.128**	-	-	-	**2.7**	-
CASE24464_GOC__SAD	210481	63871	408258	348	-	-	**25**	-	-	-	**0.122**	-	-	-	**3.06**	-
CASE30000_GOC	213698	68919	399262	372	39	66	**32**	-	**0.106**	0.125	0.118	-	4.12	8.26	**3.79**	-

Table 6 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE30000_GOC__API	213698	68919	399262	372	31	-	**30**	-	0.114	-	**0.109**	-	3.54	-	**3.26**	-
CASE30000_GOC__SAD	213698	68919	399262	372	26	-	**24**	-	**0.106**	-	0.115	-	**2.75**	-	2.77	-
CASE78484_EPIGRIDS	685984	211266	1334253	0	**30**	-	32	-	0.715	-	**0.453**	-	21.4	-	**14.5**	-
CASE78484_EPIGRIDS__API	685984	211266	1334253	0	27	-	**24**	-	0.714	-	**0.488**	-	19.3	-	**11.7**	-
CASE78484_EPIGRIDS__SAD	685984	211266	1334253	0	**31**	-	33	-	0.769	-	**0.447**	-	23.9	-	**14.7**	-

Table 7 Solve times and iteration counts for the SOCP Optimal Power Flow problem set

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations ClarabelRs	ECOS	Gurobi	Mosek	time per iteration(s) ClarabelRs	ECOS	Gurobi	Mosek	total time (s) ClarabelRs	ECOS	Gurobi	Mosek
CASE3_LMBD	124	29	190	0	**13**	15	**13**	15	**2.83e-05**	2.89e-05	5.36e-05	6.88e-05	**0.000368**	0.000434	0.000696	0.00103
CASE3_LMBD__API	124	29	190	0	**13**	14	16	14	**2.85e-05**	3.18e-05	5.14e-05	7.97e-05	**0.000371**	0.000446	0.000823	0.00112
CASE3_LMBD__SAD	124	29	190	0	13	17	**11**	14	**2.85e-05**	3.19e-05	5.51e-05	7.96e-05	**0.000371**	0.000542	0.000606	0.00111
CASE5_PJM	220	51	352	0	**16**	18	**16**	27	**5.02e-05**	5.26e-05	7.31e-05	0.000104	**0.000803**	0.000946	0.00117	0.00281
CASE5_PJM__API	220	51	352	0	**17**	24	21	29	5.94e-05	**5.75e-05**	7.47e-05	0.000103	**0.00101**	0.00138	0.00157	0.00297
CASE5_PJM__SAD	220	51	352	0	**14**	19	18	22	7.38e-05	**6.27e-05**	7.73e-05	0.000119	**0.00103**	0.00119	0.00139	0.00262
CASE14_IEEE	676	144	1069	0	14	19	**12**	28	**0.000146**	0.000152	0.000266	0.000324	**0.00204**	0.00288	0.00319	0.00908
CASE14_IEEE__API	676	144	1069	0	16	25	**15**	30	**0.000162**	0.000183	0.000245	0.00038	**0.00259**	0.00458	0.00367	0.0114
CASE14_IEEE__SAD	676	144	1069	0	15	19	**13**	31	**0.000178**	0.000196	0.000287	0.000364	**0.00267**	0.00372	0.00373	0.0113
CASE30_IEEE	1374	288	2188	0	25	30	**19**	35	**0.000331**	0.000367	0.000526	0.000609	**0.00828**	0.011	0.01	0.0213
CASE30_IEEE__API	1374	288	2188	0	**22**	36	23	37	**0.000321**	0.00037	0.00049	0.000605	**0.00706**	0.0133	0.0113	0.0224
CASE30_IEEE__SAD	1374	288	2188	0	-	-	**20**	45	-	-	**0.000532**	0.000601	-	-	**0.0106**	0.027
CASE30_AS	1404	294	2218	0	21	24	**15**	32	**0.000317**	0.000389	0.000543	0.000724	**0.00665**	0.00934	0.00815	0.0232
CASE30_AS__API	1404	294	2218	0	**25**	29	26	40	**0.000294**	0.000367	0.00048	0.000613	**0.00736**	0.0106	0.0125	0.0245
CASE30_AS__SAD	1404	294	2218	0	19	28	**16**	46	0.000396	**0.000369**	0.000568	0.000731	**0.00752**	0.0103	0.00909	0.0336

Table 7 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE24_IEEE_RTS	1430	332	2253	0	18	21	**17**	40	**0.00036**	0.000387	0.000525	0.000649	**0.00648**	0.00813	0.00892	0.026
CASE24_IEEE_RTS__API	1430	332	2253	0	19	26	**16**	40	**0.000362**	0.000404	0.000531	0.000666	**0.00688**	0.0105	0.00849	0.0266
CASE24_IEEE_RTS__SAD	1430	332	2253	0	17	22	**16**	43	**0.000325**	0.000384	0.000568	0.000656	**0.00553**	0.00846	0.00908	0.0282
CASE39_EPRI	1576	335	2506	0	**24**	39	34	53	**0.000384**	0.000416	0.000476	0.000638	**0.00921**	0.0162	0.0162	0.0338
CASE39_EPRI__API	1576	335	2506	0	**25**	52	50	53	**0.000389**	0.000441	0.000431	0.000624	**0.00974**	0.023	0.0215	0.0331
CASE39_EPRI__SAD	1576	335	2506	0	**28**	42	39	61	**0.000354**	0.000401	0.000465	0.000638	**0.00992**	0.0168	0.0181	0.0389
CASE57_IEEE	2632	547	4171	0	**23**	37	29	43	**0.000646**	0.000719	0.00102	0.0012	**0.0149**	0.0266	0.0296	0.0518
CASE57_IEEE__API	2632	547	4171	0	**22**	36	32	41	**0.000677**	0.000753	0.00105	0.00122	**0.0149**	0.0271	0.0336	0.0499
CASE57_IEEE__SAD	2632	547	4171	0	26	59	**25**	58	**0.000623**	0.000713	0.00111	0.00116	**0.0162**	0.0421	0.0278	0.0675
CASE60_C	2780	602	4310	0	21	27	**20**	38	0.000771	**0.000722**	0.00106	0.00146	**0.0162**	0.0195	0.0211	0.0556
CASE60_C__API	2780	602	4310	0	**26**	41	38	-	**0.000646**	0.000653	0.00102	-	**0.0168**	0.0268	0.0386	-
CASE60_C__SAD	2780	602	4310	0	**24**	34	**24**	49	0.0007	**0.000615**	0.00106	0.00123	**0.0168**	0.0209	0.0254	0.0602
CASE73_IEEE_RTS	4474	1033	7067	0	**22**	27	**22**	45	**0.0011**	0.00131	0.00171	0.00251	**0.0243**	0.0354	0.0377	0.113

Table 7 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE73_IEEE_RTS__API	4474	1033	7067	0	-	-	**19**	43	-	-	**0.00183**	0.00252	-	-	**0.0347**	0.109
CASE73_IEEE_RTS__SAD	4474	1033	7067	0	19	26	**16**	47	**0.00136**	0.0014	0.0018	0.00248	**0.0259**	0.0365	0.0288	0.117
CASE118_IEEE	6184	1328	10052	0	24	27	**20**	63	**0.00189**	0.00192	0.00247	0.00321	**0.0454**	0.0519	0.0494	0.202
CASE118_IEEE__API	6184	1328	10052	0	**23**	30	27	62	0.0021	**0.00196**	0.0026	0.0032	**0.0484**	0.0587	0.0701	0.199
CASE118_IEEE__SAD	6184	1328	10052	0	**21**	27	**21**	66	**0.00166**	0.00167	0.00266	0.00287	**0.0348**	0.0452	0.0558	0.19
CASE89_PEGASE	6656	1365	11100	0	**38**	-	83	-	**0.00196**	-	0.00445	-	**0.0745**	-	0.37	-
CASE89_PEGASE__API	6656	1365	11100	0	**35**	-	72	-	**0.00193**	-	0.0048	-	**0.0675**	-	0.345	-
CASE89_PEGASE__SAD	6656	1365	11100	0	**45**	-	84	-	**0.0021**	-	0.00436	-	**0.0943**	-	0.366	-
CASE179_GOC	8230	1733	12996	0	25	32	**24**	63	0.00227	**0.00205**	0.00338	0.00391	**0.0568**	0.0657	0.0812	0.246
CASE179_GOC__API	8230	1733	12996	0	**54**	-	93	-	**0.00242**	-	0.00293	-	**0.131**	-	0.273	-
CASE179_GOC__SAD	8230	1733	12996	0	25	32	**23**	77	0.00232	**0.00217**	0.00321	0.00377	**0.0579**	0.0693	0.0738	0.29
CASE200_ACTIV	8457	1777	13527	0	38	**34**	42	42	0.0022	**0.00211**	0.00291	0.00412	0.0837	**0.0717**	0.122	0.173
CASE200_ACTIV__API	8457	1777	13527	0	**48**	-	81	68	**0.00239**	-	0.00282	0.00387	**0.115**	-	0.228	0.263
CASE200_ACTIV__SAD	8457	1777	13527	0	**40**	44	49	-	0.00223	**0.00222**	0.00282	-	**0.0891**	0.0975	0.138	-
CASE197_SNEM	8752	1857	14224	0	-	-	**17**	25	-	-	**0.00324**	0.00473	-	-	**0.055**	0.118
CASE197_SNEM__API	8752	1857	14224	0	**32**	-	43	49	**0.00288**	-	0.00295	0.00462	**0.0921**	-	0.127	0.227
CASE197_SNEM__SAD	8752	1857	14224	0	-	-	**21**	24	-	-	**0.00348**	0.00486	-	-	**0.073**	0.117
CASE162_IEEE_DTC	9168	1882	14920	0	**48**	-	66	-	**0.00273**	-	0.00439	-	**0.131**	-	0.29	-
CASE162_IEEE_DTC__API	9168	1882	14920	0	**47**	83	78	90	**0.00267**	0.00318	0.0043	0.00517	**0.125**	0.264	0.335	0.465

Table 7 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE162_IEEE_DTC__SAD	9168	1882	14920	0	**46**	-	74	-	**0.00274**	-	0.00432	-	**0.126**	-	0.32	-
CASE300_IEEE	13782	2900	21993	0	-	-	**90**	-	-	-	**0.00511**	-	-	-	**0.46**	-
CASE300_IEEE__API	13782	2900	21993	0	**50**	-	80	-	**0.00353**	-	0.00506	-	**0.176**	-	0.405	-
CASE300_IEEE__SAD	13782	2900	21993	0	**94**	-	115	-	**0.00388**	-	0.00535	-	**0.365**	-	0.615	-
CASE240_PSERC	13772	3014	22076	0	28	-	**25**	-	**0.00431**	-	0.00643	-	**0.121**	-	0.161	-
CASE240_PSERC__API	13772	3014	22076	0	**26**	-	34	-	**0.00446**	-	0.00626	-	**0.116**	-	0.213	-
CASE240_PSERC__SAD	13772	3014	22076	0	**29**	-	30	-	**0.00449**	-	0.00667	-	**0.13**	-	0.2	-
CASE588_SDET	23204	4876	37012	0	**78**	-	137	-	**0.00611**	-	0.00961	-	**0.476**	-	1.32	-
CASE588_SDET__API	23204	4876	37012	0	**69**	-	-	-	**0.00656**	-	-	-	**0.453**	-	-	-
CASE588_SDET__SAD	23204	4876	37012	0	**79**	-	-	-	**0.0062**	-	-	-	**0.49**	-	-	-
CASE500_GOC	23888	5114	38653	0	**45**	-	164	-	0.00875	-	**0.00833**	-	**0.394**	-	1.37	-
CASE500_GOC__API	23888	5114	38653	0	**37**	-	156	-	**0.00783**	-	0.00843	-	**0.29**	-	1.32	-
CASE500_GOC__SAD	23888	5114	38653	0	**43**	-	219	-	**0.00884**	-	0.00917	-	**0.38**	-	2.01	-
CASE793_GOC	31082	6495	49764	0	47	-	**36**	-	**0.0114**	-	0.0122	-	0.535	-	**0.44**	-
CASE793_GOC__API	31082	6495	49764	0	**56**	-	-	-	**0.0115**	-	-	-	**0.642**	-	-	-
CASE793_GOC__SAD	31082	6495	49764	0	**53**	-	65	-	**0.00851**	-	0.0113	-	**0.451**	-	0.736	-
CASE1354_PEGASE	62814	13258	103260	0	**103**	-	-	-	**0.0234**	-	-	-	**2.41**	-	-	-

Table 7 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE1354_PEGASE__API	62814	13258	103260	0	-	-	**259**	-	-	-	**0.0292**	-	-	-	**7.56**	-
CASE1354_PEGASE__SAD	62814	13258	103260	0	**102**	-	-	-	**0.0171**	-	-	-	**1.74**	-	-	-
CASE1888_RTE	81966	17208	131225	0	**69**	-	273	-	**0.0328**	-	0.0396	-	**2.26**	-	10.8	-
CASE1888_RTE__API	81966	17208	131225	0	**84**	-	-	-	**0.0276**	-	-	-	**2.32**	-	-	-
CASE1888_RTE__SAD	81966	17208	131225	0	**65**	-	-	-	**0.0325**	-	-	-	**2.12**	-	-	-
CASE1951_RTE	84496	17817	134660	0	**79**	-	-	-	**0.0325**	-	-	-	**2.57**	-	-	-
CASE1951_RTE__API	84496	17817	134660	0	**71**	-	-	-	**0.0342**	-	-	-	**2.43**	-	-	-
CASE1951_RTE__SAD	84496	17817	134660	0	**79**	-	-	-	**0.03**	-	-	-	**2.37**	-	-	-
CASE1803_SNEM	84794	17835	138430	0	**86**	-	-	-	**0.0241**	-	-	-	**2.08**	-	-	-
CASE1803_SNEM__SAD	84794	17835	138430	0	**95**	-	-	-	**0.0255**	-	-	-	**2.42**	-	-	-
CASE2383WP_K__API	97600	20393	155950	0	22	-	**19**	-	0.0457	-	**0.0389**	-	1.01	-	**0.738**	-
CASE2312_GOC	98556	20518	159516	0	-	-	**41**	-	-	-	**0.0453**	-	-	-	**1.86**	-
CASE2312_GOC__SAD	98556	20518	159516	0	-	-	**55**	-	-	-	**0.0417**	-	-	-	**2.29**	-
CASE2000_GOC	108628	22742	178538	0	**53**	-	303	-	0.0505	-	**0.0505**	-	**2.68**	-	15.3	-

Table 7 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE2000_GOC__API	108628	22742	178538	0	**52**	-	358	-	**0.0458**	-	0.0496	-	**2.38**	-	17.8	-
CASE2000_GOC__SAD	108628	22742	178538	0	**52**	-	328	-	**0.0506**	-	0.051	-	**2.63**	-	16.7	-
CASE2737SOP_K	109822	22777	176602	0	**87**	-	-	-	**0.0311**	-	-	-	**2.7**	-	-	-
CASE2737SOP_K__SAD	109822	22777	176602	0	**77**	-	-	-	**0.0421**	-	-	-	**3.24**	-	-	-
CASE2736SP_K	110022	22878	176901	0	**74**	-	-	-	**0.0332**	-	-	-	**2.46**	-	-	-
CASE2736SP_K__SAD	110022	22878	176901	0	**73**	-	-	-	**0.0459**	-	-	-	**3.35**	-	-	-
CASE2746WP_K	111106	23320	178556	0	**74**	-	-	-	**0.034**	-	-	-	**2.52**	-	-	-
CASE2746WP_K__API	111106	23320	178556	0	-	63	**31**	-	-	**0.0352**	0.0399	-	-	2.22	**1.24**	-
CASE2746WP_K__SAD	111106	23320	178556	0	**79**	-	-	-	**0.0474**	-	-	-	**3.74**	-	-	-
CASE2746WOP_K	111822	23434	179829	0	**68**	-	-	-	**0.0362**	-	-	-	**2.46**	-	-	-
CASE2746WOP_K__API	111822	23434	179829	0	-	-	**26**	-	-	-	**0.0416**	-	-	-	**1.08**	-
CASE2746WOP_K__SAD	111822	23434	179829	0	**64**	-	-	-	**0.0353**	-	-	-	**2.26**	-	-	-
CASE3012WP_K__API	120676	25202	193907	0	-	-	**27**	-	-	-	**0.0477**	-	-	-	**1.29**	-
CASE2848_RTE	122708	25858	197228	0	**80**	-	-	-	**0.0448**	-	-	-	**3.59**	-	-	-
CASE2848_RTE__API	122708	25858	197228	0	**82**	-	-	-	**0.0541**	-	-	-	**4.44**	-	-	-
CASE2848_RTE__SAD	122708	25858	197228	0	-	-	**156**	-	-	-	**0.0573**	-	-	-	**8.94**	-

Table 7 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE2868_RTE	123912	26164	198869	0	**89**	-	207	-	**0.0539**	-	0.0587	-	**4.79**	-	12.1	-
CASE2868_RTE__API	123912	26164	198869	0	**82**	-	-	-	**0.0515**	-	-	-	**4.23**	-	-	-
CASE2868_RTE__SAD	123912	26164	198869	0	**82**	-	173	-	**0.0503**	-	0.0542	-	**4.12**	-	9.38	-
CASE2853_SDET__API	128886	27445	206896	0	**128**	-	-	-	**0.0334**	-	-	-	**4.28**	-	-	-
CASE3022_GOC	134780	28060	217434	0	-	-	**32**	-	-	-	**0.0662**	-	-	-	**2.12**	-
CASE3022_GOC__API	134780	28060	217434	0	-	-	**117**	-	-	-	**0.0958**	-	-	-	**11.2**	-
CASE3022_GOC__SAD	134780	28060	217434	0	-	-	**33**	-	-	-	**0.0631**	-	-	-	**2.08**	-
CASE3375WP_K	139126	29112	223999	0	**100**	-	-	-	**0.0516**	-	-	-	**5.16**	-	-	-
CASE3375WP_K__API	139126	29112	223999	0	-	-	**460**	-	-	-	**0.0964**	-	-	-	**44.4**	-
CASE3375WP_K__SAD	139126	29112	223999	0	**97**	-	-	-	**0.0394**	-	-	-	**3.82**	-	-	-
CASE2869_PEGASE	143608	30153	236217	0	**56**	-	340	-	**0.062**	-	0.086	-	**3.47**	-	29.3	-
CASE2869_PEGASE__API	143608	30153	236217	0	**54**	-	-	-	**0.0651**	-	-	-	**3.51**	-	-	-
CASE2869_PEGASE__SAD	143608	30153	236217	0	**52**	-	317	-	**0.0661**	-	0.0865	-	**3.44**	-	27.4	-
CASE2742_GOC	144110	29856	236467	0	110	-	**78**	-	0.0908	-	**0.0901**	-	9.99	-	**7.03**	-
CASE2742_GOC__API	144110	29856	236467	0	110	-	**56**	-	**0.0869**	-	0.1	-	9.56	-	**5.61**	-
CASE2742_GOC__SAD	144110	29856	236467	0	109	-	**75**	-	**0.0653**	-	0.0731	-	7.12	-	**5.48**	-
CASE4661_SDET	198498	41599	320728	0	**143**	-	341	-	**0.0698**	-	0.168	-	**9.98**	-	57.4	-
CASE4661_SDET__API	198498	41599	320728	0	**132**	-	-	-	**0.0674**	-	-	-	**8.89**	-	-	-
CASE3970_GOC	205819	42789	337328	0	**141**	-	177	-	**0.089**	-	0.109	-	**12.6**	-	19.2	-
CASE3970_GOC__API	205819	42789	337328	0	-	-	**94**	-	-	-	**0.119**	-	-	-	**11.2**	-

Table 7 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE3970_GOC__SAD	205819	42789	337328	0	**143**	-	213	-	**0.092**	-	0.108	-	**13.2**	-	23	-
CASE4020_GOC	216455	44877	355706	0	-	-	**164**	-	-	-	**0.174**	-	-	-	**28.6**	-
CASE4020_GOC__API	216455	44877	355706	0	100	-	**59**	-	**0.121**	-	0.143	-	12.1	-	**8.42**	-
CASE4020_GOC__SAD	216455	44877	355706	0	188	-	**186**	-	**0.137**	-	0.242	-	**25.7**	-	44.9	-
CASE4917_GOC	218213	45522	352833	0	-	-	**36**	-	-	-	**0.103**	-	-	-	**3.69**	-
CASE4917_GOC__API	218213	45522	352833	0	-	-	**79**	-	-	-	**0.124**	-	-	-	**9.81**	-
CASE4917_GOC__SAD	218213	45522	352833	0	-	-	**33**	175	-	-	**0.117**	0.161	-	-	**3.85**	28.1
CASE4601_GOC	225588	46885	368445	0	181	-	**150**	-	**0.096**	-	0.123	-	**17.4**	-	18.4	-
CASE4601_GOC__API	225588	46885	368445	0	-	-	**69**	-	-	-	**0.135**	-	-	-	**9.31**	-
CASE4601_GOC__SAD	225588	46885	368445	0	**180**	-	182	-	**0.0919**	-	0.124	-	**16.5**	-	22.5	-
CASE4837_GOC	240160	49855	392884	0	136	-	**104**	-	0.136	-	**0.126**	-	18.5	-	**13.1**	-
CASE4837_GOC__API	240160	49855	392884	0	124	-	**91**	-	0.136	-	**0.129**	-	16.9	-	**11.7**	-
CASE4837_GOC__SAD	240160	49855	392884	0	137	-	**90**	-	0.138	-	**0.12**	-	18.8	-	**10.8**	-
CASE4619_GOC	254753	52616	419210	0	**151**	-	200	-	**0.147**	-	0.16	-	**22.1**	-	31.9	-
CASE4619_GOC__API	254753	52616	419210	0	135	-	**117**	-	**0.14**	-	0.169	-	**18.9**	-	19.8	-
CASE4619_GOC__SAD	254753	52616	419210	0	**152**	-	173	-	**0.141**	-	0.165	-	**21.5**	-	28.6	-
CASE5658_EPIGRIDS	282180	58620	461724	0	157	-	**67**	-	**0.157**	-	0.194	-	24.6	-	**13**	-
CASE5658_EPIGRIDS__API	282180	58620	461724	0	167	-	**55**	-	**0.173**	-	0.209	-	28.9	-	**11.5**	-

Table 7 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE5658_EPIGRIDS__SAD	282180	58620	461724	0	155	-	**71**	-	**0.173**	-	0.188	-	26.8	-	**13.4**	-
CASE6468_RTE	286248	59396	463599	0	**104**	-	153	-	**0.175**	-	0.188	-	**18.2**	-	28.7	-
CASE6468_RTE__API	286248	59396	463599	0	**90**	-	151	-	**0.151**	-	0.164	-	**13.6**	-	24.7	-
CASE6468_RTE__SAD	286248	59396	463599	0	**99**	-	125	-	**0.14**	-	0.159	-	**13.8**	-	19.9	-
CASE6470_RTE	287806	60144	465757	0	**124**	-	216	-	**0.136**	-	0.202	-	**16.9**	-	43.6	-
CASE6470_RTE__API	287806	60144	465757	0	**94**	-	-	-	**0.148**	-	-	-	**13.9**	-	-	-
CASE6470_RTE__SAD	287806	60144	465757	0	**100**	-	-	-	**0.152**	-	-	-	**15.2**	-	-	-
CASE6495_RTE	288050	60099	465939	0	**117**	-	-	-	**0.134**	-	-	-	**15.7**	-	-	-
CASE6495_RTE__API	288050	60099	465939	0	**106**	-	298	-	**0.139**	-	0.209	-	**14.7**	-	62.2	-
CASE6495_RTE__SAD	288050	60099	465939	0	**113**	-	-	-	**0.135**	-	-	-	**15.2**	-	-	-
CASE6515_RTE	288710	60239	466922	0	**99**	-	222	-	**0.15**	-	0.181	-	**14.9**	-	40.2	-
CASE6515_RTE__API	288710	60239	466922	0	**85**	-	-	-	**0.151**	-	-	-	**12.8**	-	-	-
CASE6515_RTE__SAD	288710	60239	466922	0	**88**	-	236	-	**0.182**	-	0.246	-	**16**	-	58.1	-
CASE7336_EPIGRIDS	358450	74618	585766	0	152	-	**72**	-	**0.24**	-	0.262	-	36.5	-	**18.9**	-
CASE7336_EPIGRIDS__API	358450	74618	585766	0	-	-	**50**	-	-	-	**0.214**	-	-	-	**10.7**	-
CASE7336_EPIGRIDS__SAD	358450	74618	585766	0	158	-	**61**	-	0.245	-	**0.243**	-	38.7	-	**14.8**	-
CASE10000_GOC	440997	92799	712177	0	-	-	**74**	-	-	-	**0.417**	-	-	-	**30.9**	-
CASE10000_GOC__API	440997	92799	712177	0	**68**	-	424	-	**0.332**	-	0.398	-	**22.5**	-	169	-
CASE10000_GOC__SAD	440997	92799	712177	0	-	-	**227**	-	-	-	**0.476**	-	-	-	**108**	-
CASE9591_GOC	495073	102120	811889	0	-	-	**143**	-	-	-	**0.401**	-	-	-	**57.3**	-
CASE9591_GOC__API	495073	102120	811889	0	-	-	**97**	-	-	-	**0.343**	-	-	-	**33.2**	-
CASE9591_GOC__SAD	495073	102120	811889	0	-	-	**129**	-	-	-	**0.447**	-	-	-	**57.7**	-

Table 7 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations				time per iteration(s)				total time (s)			
					ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek	ClarabelRs	ECOS	Gurobi	Mosek
CASE9241_PEGASE	502110	104741	827419	0	-	-	**353**	-	-	-	**0.465**	-	-	-	**164**	-
CASE9241_PEGASE__API	502110	104741	827419	0	-	-	**389**	-	-	-	**0.471**	-	-	-	**183**	-
CASE9241_PEGASE__SAD	502110	104741	827419	0	-	-	**318**	-	-	-	**0.468**	-	-	-	**149**	-
CASE10192_EPIGRIDS	529695	109758	866501	0	-	-	**88**	-	-	-	**0.376**	-	-	-	**33.1**	-
CASE10192_EPIGRIDS__API	529695	109758	866501	0	-	-	**105**	-	-	-	**0.345**	-	-	-	**36.3**	-
CASE10192_EPIGRIDS__SAD	529695	109758	866501	0	-	-	**106**	-	-	-	**0.369**	-	-	-	**39.1**	-
CASE10480_GOC	573754	118760	943278	0	-	-	**177**	-	-	-	**0.435**	-	-	-	**77**	-
CASE10480_GOC__API	573754	118760	943278	0	-	-	**115**	-	-	-	**0.444**	-	-	-	**51**	-
CASE10480_GOC__SAD	573754	118760	943278	0	163	-	**156**	-	**0.379**	-	0.433	-	**61.8**	-	67.5	-
CASE13659_PEGASE	662910	140961	1081042	0	-	-	**440**	-	-	-	**0.509**	-	-	-	**224**	-
CASE13659_PEGASE__API	662910	140961	1081042	0	-	-	**337**	-	-	-	**0.507**	-	-	-	**171**	-
CASE13659_PEGASE__SAD	662910	140961	1081042	0	-	-	**270**	-	-	-	**0.552**	-	-	-	**149**	-

Table 8 Solve times and iteration counts for the CBLIB Exponential Cone problem set

					iterations			time per iteration(s)			total time (s)		
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	ECOS	Mosek	ClarabelRs	ECOS	Mosek	ClarabelRs	ECOS	Mosek
BSS1	14	11	23	0	**9**	14	**9**	1.2e-05	**5.67e-06**	4.18e-05	0.000108	**7.93e-05**	0.000376
DEMB782	14	11	23	0	**6**	14	8	1.63e-05	**5.99e-06**	4.21e-05	9.8e-05	**8.39e-05**	0.000337
BSS2	20	15	33	0	**9**	14	**9**	1.19e-05	**8.54e-06**	4.04e-05	**0.000107**	0.00012	0.000364
DEMB781	26	19	40	0	10	16	**7**	1.42e-05	**8.68e-06**	5.11e-05	0.000142	**0.000139**	0.000358
GPTEST	32	24	48	0	**8**	16	9	1.96e-05	**1.03e-05**	4.77e-05	**0.000157**	0.000165	0.000429
RIJC781	32	24	48	0	**8**	16	9	1.92e-05	**9.71e-06**	4.91e-05	**0.000154**	0.000155	0.000442
RIJC784	40	29	66	0	11	17	**10**	1.82e-05	**1.29e-05**	5.43e-05	**0.000201**	0.000219	0.000543
RIJC785	44	33	83	0	13	17	**9**	1.96e-05	**1.32e-05**	6.01e-05	0.000255	**0.000225**	0.000541
RIJC786	44	33	83	0	9	18	**8**	2.34e-05	**1.22e-05**	6.33e-05	**0.000211**	0.000219	0.000507
RIJC782	56	40	87	0	**11**	17	**11**	2.06e-05	**1.48e-05**	5.66e-05	**0.000226**	0.000251	0.000623
RIJC783	74	53	124	0	**10**	17	11	3.07e-05	**1.78e-05**	6.48e-05	0.000307	**0.000302**	0.000713
BECK751	113	80	236	0	13	21	**11**	4.34e-05	**2.87e-05**	9.32e-05	**0.000565**	0.000602	0.00103
BECK752	113	80	236	0	14	23	**12**	3.75e-05	**3.24e-05**	9.29e-05	**0.000525**	0.000744	0.00111
BECK753	113	80	236	0	12	22	**11**	4.31e-05	**2.8e-05**	9.17e-05	**0.000517**	0.000616	0.00101
FIAC81B	122	87	201	0	14	25	**13**	3.98e-05	**2.95e-05**	8.13e-05	**0.000558**	0.000736	0.00106
FANG88	165	119	252	0	15	24	**14**	5.47e-05	**3.95e-05**	9.42e-05	**0.000821**	0.000949	0.00132
DEMB761	183	131	284	0	**15**	20	**15**	5.47e-05	**4.15e-05**	0.000114	**0.000821**	0.000831	0.0017
DEMB762	183	131	284	0	15	22	**14**	6.81e-05	**4.01e-05**	0.000116	0.00102	**0.000882**	0.00163

Table 8 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations ClarabelRs	ECOS	Mosek	time per iteration(s) ClarabelRs	ECOS	Mosek	total time (s) ClarabelRs	ECOS	Mosek
DEMB763	183	131	284	0	15	20	**14**	5.27e-05	**3.96e-05**	0.000129	**0.000791**	0.000791	0.0018
FIAC81A	289	191	496	0	13	22	**10**	9.26e-05	**6.45e-05**	0.000228	**0.0012**	0.00142	0.00228
RIJC787	296	200	510	0	13	23	**10**	9.51e-05	**6.27e-05**	0.000271	**0.00124**	0.00144	0.00271
CAR	865	601	1584	0	18	25	**10**	0.000267	**0.000213**	0.000591	**0.00481**	0.00532	0.00591
GP_DAVE_1	1441	705	3686	0	25	32	**21**	0.000404	**0.00034**	0.00117	**0.0101**	0.0109	0.0247
JHA88	1691	1131	3005	0	16	24	**13**	0.000495	**0.000351**	0.00155	**0.00791**	0.00843	0.0202
VARUN	2013	1346	6788	0	**24**	46	**24**	0.000853	**0.000656**	0.00136	**0.0205**	0.0302	0.0328
GP_DAVE_2	2489	1219	6752	0	31	36	**22**	0.000729	**0.00064**	0.00164	**0.0226**	0.0231	0.036
LogExpCR_N20_M400	3223	2022	13238	0	24	24	**18**	0.00111	**0.000926**	0.00165	0.0267	**0.0222**	0.0296
GP_DAVE_3	3537	1733	9818	0	-	40	**24**	-	**0.000917**	0.00249	-	**0.0367**	0.0597
LogExpCR_N100_M400	3303	2102	45314	0	30	26	**19**	0.00618	0.00357	**0.00282**	0.185	0.0928	**0.0535**
LogExpCR_N500_M400	3703	2502	205664	0	46	34	**30**	0.0585	0.047	**0.0173**	2.69	1.6	**0.52**
MRA01	5513	3681	10680	0	26	35	**14**	0.00208	**0.00154**	0.00445	0.0541	**0.0539**	0.0623
LogExpCR_N20_M800	6423	4022	26411	0	29	27	**21**	0.00236	**0.00189**	0.00363	0.0684	**0.0511**	0.0762
LogExpCR_N100_M800	6503	4102	90322	0	35	30	**23**	0.0144	0.00748	**0.00625**	0.502	0.224	**0.144**
LogExpCR_N500_M800	6903	4502	409866	0	41	34	**31**	0.119	0.127	**0.019**	4.89	4.32	**0.588**
LogExpCR_N20_M1200	9623	6022	39574	0	27	27	**21**	0.00369	**0.00307**	0.00556	0.0996	**0.0828**	0.117
LogExpCR_N100_M1200	9703	6102	135324	0	28	28	**22**	0.0225	0.012	**0.00946**	0.631	0.335	**0.208**
LogExpCR_N500_M1200	10103	6502	614001	0	34	36	**31**	0.204	0.195	**0.0465**	6.94	7.02	**1.44**
LogExpCR_N20_M1600	12823	8022	52729	0	28	28	**20**	**0.00537**	0.00921	0.00741	0.15	0.258	**0.148**
LogExpCR_N100_M1600	12903	8102	180333	0	28	30	**23**	0.0321	0.0568	**0.0125**	0.899	1.7	**0.288**
LogExpCR_N500_M1600	13303	8502	818214	0	36	36	**29**	0.29	2.11	**0.0617**	10.5	76.1	**1.79**

Table 8 continued

Problem	vars.	cons.	nnz(A)	nnz(P)	iterations			time per iteration(s)			total time (s)		
					ClarabelRs	ECOS	Mosek	ClarabelRs	ECOS	Mosek	ClarabelRs	ECOS	Mosek
LogExpCR_n20_m2000	16023	10022	65872	0	31	29	**23**	**0.00722**	0.0116	0.00913	0.224	0.335	**0.21**
LogExpCR_n100_m2000	16103	10102	225278	0	33	29	**24**	0.0416	0.0861	**0.0163**	1.37	2.5	**0.391**
LogExpCR_n500_m2000	16503	10502	1021846	0	39	39	**29**	0.386	2.68	**0.0779**	15.1	105	**2.26**
mra02	21965	14606	50050	0	30	-	**16**	**0.00865**	-	0.0193	**0.26**	-	0.309
cx02_100	31087	20693	56430	0	20	27	**13**	0.0118	**0.00947**	0.0294	**0.236**	0.256	0.382
cx02_200	122187	81393	222880	0	20	34	**13**	0.0608	**0.0508**	0.138	**1.22**	1.73	1.79

Table 9 Solve times and iteration counts for the SDPLIB Semidefinite Programming problem set

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	Mosek	SDPT3	SeDuMi	ClarabelRs	Mosek	SDPT3	SeDuMi	ClarabelRs	Mosek	SDPT3	SeDuMi
TRUSS1	19	6	25	0	11	**10**	18	13	**2.62e-05**	3.31e-05	0.0196	0.000769	**0.000289**	0.000331	0.353	0.01
TRUSS4	37	12	50	0	10	**9**	18	14	**4.19e-05**	7.29e-05	0.0282	0.000714	**0.000419**	0.000657	0.508	0.01
HINF1	41	13	92	0	**28**	-	-	-	**6.84e-05**	-	-	-	**0.00191**	-	-	-
HINF2	51	13	118	0	**17**	-	-	-	**8.69e-05**	-	-	-	**0.00148**	-	-	-
HINF3	51	13	118	0	**26**	-	-	-	**7.6e-05**	-	-	-	**0.00198**	-	-	-
HINF4	51	13	118	0	**24**	-	-	27	**8.15e-05**	-	-	0.00259	**0.00196**	-	-	0.07
HINF9	51	13	118	0	**26**	-	-	**26**	**8.49e-05**	-	-	0.00385	**0.00221**	-	-	0.1
CONTROL1	70	21	345	0	37	-	-	**30**	**0.000157**	-	-	0.000667	**0.0058**	-	-	0.02
TRUSS3	91	27	118	0	12	**10**	19	15	**0.000152**	0.000216	0.00767	0.00267	**0.00182**	0.00216	0.146	0.04
HINF12	120	43	579	0	43	-	-	**29**	**0.000223**	-	-	0.00241	**0.0096**	-	-	0.07
HINF14	227	73	2394	0	**22**	-	-	-	**0.000712**	-	-	-	**0.0157**	-	-	-
CONTROL2	265	66	2590	0	**25**	-	-	31	0.0016	-	-	**0.00129**	**0.0399**	-	-	0.04
TRUSS2	331	58	567	0	**14**	-	-	20	**0.000381**	-	-	0.0035	**0.00534**	-	-	0.07
QAP5	351	136	1026	0	**9**	**9**	27	14	0.00349	**0.00119**	0.0058	0.00286	0.0314	**0.0107**	0.157	0.04
TRUSS7	451	86	863	0	**22**	23	-	27	**0.000511**	0.000603	-	0.137	**0.0112**	0.0139	-	3.71
CONTROL3	585	136	8610	0	**29**	-	-	-	**0.00724**	-	-	-	**0.21**	-	-	-

Table 9 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	Mosek	SDPT3	SeDuMi	ClarabelRs	Mosek	SDPT3	SeDuMi	ClarabelRs	Mosek	SDPT3	SeDuMi
TRUSS6	901	172	1726	0	25	**22**	-	28	**0.000982**	0.00179	-	0.0243	**0.0245**	0.0394	-	0.68
CONTROL4	1030	231	20280	0	**32**	-	-	-	**0.0247**	-	-	-	**0.79**	-	-	-
THETA1	1275	104	153	0	12	**8**	26	19	0.286	0.0185	0.038	**0.00579**	3.43	0.148	0.988	**0.11**
TRUSS5	1816	208	2823	0	18	**16**	-	23	0.0231	**0.00448**	-	0.00652	0.416	**0.0717**	-	0.15
CONTROL7	3115	666	107940	0	**39**	-	-	-	**0.622**	-	-	-	**24.3**	-	-	-
QAP9	3403	748	8830	0	**29**	-	-	-	**0.891**	-	-	-	**25.8**	-	-	-
CONTROL8	4060	861	160960	0	**41**	-	-	-	**1.07**	-	-	-	**43.8**	-	-	-
MCP100	5050	100	100	0	11	**7**	-	17	0.0243	0.679	-	**0.0171**	**0.267**	4.76	-	0.29
GPP100	5050	101	5150	0	**26**	-	-	-	**1.95**	-	-	-	**50.8**	-	-	-
THETA2	5050	498	597	0	10	**9**	-	20	2.07	0.68	-	**0.0265**	20.7	6.12	-	**0.53**
QAP10	5151	1021	13101	0	**28**	-	-	-	**2.02**	-	-	-	**56.6**	-	-	-
CONTROL9	5130	1081	229005	0	**36**	-	-	-	**1.68**	-	-	-	**60.6**	-	-	-
TRUSS8	6271	496	8286	0	20	**14**	-	24	0.776	**0.0321**	-	0.0387	15.5	**0.449**	-	0.93
CONTROL10	6325	1326	313950	0	**44**	-	-	-	**2.74**	-	-	-	**120**	-	-	-
MCP124_1	7750	124	124	0	12	**7**	-	17	**0.00162**	2.25	-	0.0247	**0.0194**	15.8	-	0.42
MCP124_2	7750	124	124	0	11	**8**	-	18	0.0526	2.22	-	**0.0267**	0.578	17.8	-	**0.48**
MCP124_3	7750	124	124	0	14	**8**	-	18	0.656	2.21	-	**0.0306**	9.18	17.7	-	**0.55**
MCP124_4	7750	124	124	0	12	**7**	-	20	2.12	2.28	-	**0.0285**	25.5	16	-	**0.57**
GPP124_1	7750	125	7874	0	**30**	-	-	-	**5.24**	-	-	-	**157**	-	-	-
GPP124_2	7750	125	7874	0	**28**	-	-	-	**5.23**	-	-	-	**147**	-	-	-
GPP124_3	7750	125	7874	0	28	**24**	-	-	5.18	**1.98**	-	-	145	**47.6**	-	-
GPP124_4	7750	125	7874	0	**32**	-	-	-	**5.15**	-	-	-	**165**	-	-	-

Table 9 continued

					iterations				time per iteration(s)				total time (s)			
Problem	vars.	cons.	nnz(A)	nnz(P)	ClarabelRs	Mosek	SDPT3	SeDuMi	ClarabelRs	Mosek	SDPT3	SeDuMi	ClarabelRs	Mosek	SDPT3	SeDuMi
CONTROL11	7645	1596	417670	0	**66**	-	-	-	**4.27**	-	-	-	**282**	-	-	-
THETA3	11325	1106	1255	0	10	**9**	-	20	15.5	6.09	-	**0.112**	155	54.8	-	**2.24**
ARCH0	13215	174	3030	0	**22**	-	-	31	0.264	-	-	**0.0516**	5.8	-	-	**1.6**
ARCH2	13215	174	3420	0	**24**	-	-	28	0.0817	-	-	**0.0636**	1.96	-	-	**1.78**
ARCH4	13215	174	3420	0	**20**	-	-	29	0.0834	-	-	**0.0459**	1.67	-	-	**1.33**
ARCH8	13215	174	3420	0	**24**	-	-	34	0.082	-	-	**0.0468**	1.97	-	-	**1.59**
THETA4	20100	1949	2148	0	**10**	**10**	-	20	76.5	30.5	-	**0.472**	765	305	-	**9.44**
MCP250_1	31375	250	250	0	12	**9**	-	19	**0.0158**	110	-	0.0853	**0.19**	991	-	1.62
MCP250_2	31375	250	250	0	11	**8**	-	19	1.12	109	-	**0.0858**	12.4	872	-	**1.63**
MCP250_3	31375	250	250	0	13	**8**	-	19	12.6	112	-	**0.0868**	164	895	-	**1.65**
MCP250_4	31375	250	250	0	14	**9**	-	21	52.8	111	-	**0.089**	739	999	-	**1.87**
THETA5	31375	3028	3277	0	-	**10**	-	23	-	111	-	**1.54**	-	1.11e+03	-	**35.4**
SS30	43497	132	6642	0	**29**	-	-	-	**2.51**	-	-	-	**72.9**	-	-	-

Acknowledgements The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work.

Funding This work was supported by the Oxford University Clarendon Fund.

Data Availability statement Source code and data for this work are available at: Documentation: https://clarabel.org/ Julia implementation: https://github.com/oxfordcontrol/Clarabel.jl Rust implementation: https://github.com/oxfordcontrol/Clarabel.rs Benchmarking code:https://github.com/oxfordcontrol/ClarabelBenchmarks Datasets generated by the benchmarking tests in this study are available from the corresponding author.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Borrelli, F., Bemporad, A., Morari, M.: Predictive control for linear and hybrid systems. Cambridge University Press, (2017)
2. Allan, D. A., Rawlings, J. B.: "Moving horizon estimation," in Handbook of Model Predictive Control. Springer International Publishing, pp. 99–124 (2019)
3. Makrodimopoulos, A., Martin, C.M.: Lower bound limit analysis of cohesive-frictional materials using second-order cone programming. Int. J. Numer. Meth. Eng. **66**(4), 604–634 (2006)
4. Makrodimopoulos, A., Martin, C.M.: Upper bound limit analysis using simplex strain elements and second-order cone programming. Int. J. Numer. Anal. Meth. Geomech. **31**(6), 835–865 (2007)
5. Goulart, P.J., Chernyshenko, S.: Global stability analysis of fluid flows using sum-of-squares. Physica D **241**(6), 692–704 (2012)
6. Combettes, P.: "The convex feasibility problem in image recovery," in ser. Advances in Imaging and Electron Physics, P. W. Hawkes, Ed., vol. 95, Elsevier, pp. 155–270 (1996)
7. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
8. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. Roy. Stat. Soc. B **58**(1), 267–288 (1996)
9. Candés, E.J., Wakin, M.B., Boyd, S.: Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Anal. Appl. **14**(5), 877–905 (2008)
10. Boyd, S., Kim, S.-J., Vandenberghe, L., Hassibi, A.: A tutorial on geometric programming. Optim. Eng. **8**(1), 67–127 (2007)
11. Cornuejols, G., Tütüncü, R.: Optimization Methods in Finance (Mathematics, Finance and Risk). Cambridge University Press, (2006)
12. Markowitz, H.: Portfolio selection. J. Financ. **7**(1), 77–91 (1952)
13. Nocedal, J., Wright, S. J.: Numerical optimization (Springer Series in Operations Research and Financial Engineering). Springer, (2006)
14. Boggs, P.T., Tolle, J.W.: Sequential quadratic programming. Acta Numer **4**, 1–51 (1995)
15. Bertsimas, D., Weismantel, R.: Optimization over integers. Athena Scientific, (2005)
16. Morrison, D.R., Jacobson, S.H., Sauppe, J.J., Sewell, E.C.: Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discret. Optim. **19**, 79–102 (2016)

17. Kantorovich, L.: "Mathematical methods of organizing and planning production," Management Science, vol. 6, no. 4, pp. 366–422, English translation (1960)
18. Koopmans, T.: "Exchange ratios between cargoes on various routes (non-refrigerated dry cargoes)," Memorandum for the Combined Shipping Adjustment Board, 1942, Reprinted in Scientific Papers of Tjalling C. Koopmans, Springer Verlag, (1970)
19. Koopmans, T.: Optimum utilization of the transportation system. Econometrica **17**, 136–146 (1949)
20. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly **3**(1–2), 95–110 (1956)
21. Dantzig, G. B.: Linear programming and extensions. Princeton University Press, (1963)
22. Wolfe, P.: The simplex method for quadratic programming. Econometrica **27**(3), 382–398 (1959)
23. Gurobi Optimization, LLC, Gurobi Optimizer Reference Manual, (2023)
24. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. Math. Program. Comput. **6**(4), 327–363 (2014)
25. Huangfu, Q., Hall, J.A.J.: Parallelizing the dual revised simplex method. Math. Program. Comput. **10**(1), 119–142 (2018)
26. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning **3**(1), 1–122 (2011)
27. Wright, S. J.: Primal-Dual Interior-Point Methods. Society for Industrial and Applied Mathematics, (1997)
28. He, B., Yuan, X.: On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. SIAM J. Numer. Anal. **50**(2), 700–709 (2012)
29. Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: OSQP: An operator splitting solver for quadratic programs. Math. Program. Comput. **12**(4), 637–672 (2020)
30. O'Donoghue, B., Chu, E., Parikh, N., Boyd, S.: Conic optimization via operator splitting and homogeneous self-dual embedding. J. Optim. Theory Appl. **169**(3), 1042–1068 (2016)
31. Garstka, M., Cannon, M., Goulart, P.: COSMO: A conic operator splitting method for convex conic problems. J. Optim. Theory Appl. **190**(3), 779–810 (2021)
32. Nemirovski, A.S., Todd, M.J.: Interior-point methods for optimization. Acta Numer **17**, 191–234 (2008)
33. Karmarkar, N.: "A new polynomial-time algorithm for linear programming," in Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, ser. STOC '84, Association for Computing Machinery, pp. 302–311 (1984)
34. Renegar, J.: A polynomial-time algorithm, based on Newton's method, for linear programming. Math. Program. **40**(1), 59–93 (1988)
35. Nesterov, Y.E., Todd, M.J.: Self-scaled barriers and interior-point methods for convex programming. Math. Oper. Res. **22**(1), 1–42 (1997)
36. Nesterov, Y.E., Todd, M.J.: Primal-dual interior-point methods for self-scaled cones. SIAM J. Optim. **8**(2), 324–364 (1998)
37. Chares, P. R.: "Cones and interior-point algorithms for structured convex optimization involving powers and exponentials," Ph.D. dissertation, Université Catholique de Louvain, (2009)
38. Nesterov, Y.: Towards non-symmetric conic optimization. Optimization Methods and Software **27**(4–5), 893–917 (2012)
39. Dahl, J., Andersen, E.D.: A primal-dual interior-point algorithm for nonsymmetric exponential-cone optimization. Math. Program. **194**(1), 341–370 (2022)
40. Papp, D., Yildiz, S.: Sum-of-squares optimization without semidefinite programming. SIAM J. Optim. **29**(1), 822–851 (2019)
41. Coey, C., Kapelevich, L., Vielma, J.P.: Solving natural conic formulations with Hypatia.jl. INFORMS J. Comput. **34**(5), 2686–2699 (2022)
42. Karimi, M., Tunçel, L.: Domain-driven solver (DDS) version 2.1: A MATLAB-based software package for convex optimization problems in domain-driven form. Math. Program. Comput. **16**(1), 37–92 (2024)
43. Skajaa, A., Ye, Y.: A homogeneous interior-point algorithm for nonsymmetric convex conic optimization. Math. Program. **150**(2), 391–422 (2015)
44. Serrano, S. A.: "Algorithms for unsymmetric cone optimization and an implementation for problems with the exponential cone," Ph.D. dissertation, Stanford University, (2015)

45. Ye, Y., Todd, M.J., Mizuno, S.: An O($\sqrt{nL}$)-iteration homogeneous and self-dual linear programming algorithm. Math. Oper. Res. **19**(1), 53–67 (1994)
46. Vandenberghe, L.: The CVXOPT linear and quadratic cone program solvers, https://www.seas.ucla.edu/~vandenbe/publications/coneprog.pdf, (2010)
47. Domahidi, A., Chu, E., Boyd, S.: "ECOS: An SOCP solver for embedded systems," in. European Control Conference (ECC) **2013**, 3071–3076 (2013)
48. MOSEK ApS, MOSEK Optimization Suite 10.0, (2023)
49. Ye, Y.: On homogeneous and self-dual algorithms for LCP. Math. Program. **76**(1), 211–221 (1997)
50. Andersen, E.D., Ye, Y.: On a homogeneous algorithm for the monotone complementarity problem. Math. Program. **84**(2), 375–399 (1999)
51. Yoshise, A.: Interior point trajectories and a homogeneous model for nonlinear complementarity problems over symmetric cones. SIAM J. Optim. **17**(4), 1129–1153 (2007)
52. Yoshise, A.: Homogeneous algorithms for monotone complementarity problems over symmetric cones. Pacific Journal of Optimization **5**(02), 313–337 (2009)
53. Meszaros, C.: The practical behavior of the homogeneous self-dual formulations in interior point methods. CEJOR **23**(4), 913–924 (2015)
54. O'Donoghue, B.: Operator splitting for a homogeneous embedding of the linear complementarity problem. SIAM J. Optim. **31**(3), 1999–2023 (2021)
55. Nesterov, Y., Todd, M.J., Ye, Y.: Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. Math. Program. **84**(2), 227–267 (1999)
56. Lobo, M. S., Vandenberghe, L., Boyd, S., Lebret, H.: "Applications of second-order cone programming," Linear Algebra and its Applications, vol. 284, no. 1, pp. 193–228, (1998) International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing
57. Banjac, G., Goulart, P., Stellato, B., Boyd, S.: Infeasibility detection in the alternating direction method of multipliers for convex optimization. J. Optim. Theory Appl. **183**(2), 490–519 (2019)
58. Nesterov, Y., Nemirovskii, A.: Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics, (1994)
59. Sturm, J.F.: Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. Optimization Methods and Software **17**(6), 1105–1154 (2002)
60. Ruiz, D.: "A scaling algorithm to equilibrate both rows and columns norms in matrices," Rutherford Appleton Laboratory, Tech. Rep. RAL-TR-2001-034, (2001)
61. Sun, Y., Andersen, M.S., Vandenberghe, L.: Decomposition in conic optimization with partially separable structure. SIAM J. Optim. **24**(2), 873–897 (2014)
62. Kim, S., Kojima, M., Mevissen, M., Yamashita, M.: Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. Math. Program. **129**(1), 33–68 (2011)
63. Chen, Y., Goulart, P.: An efficient implementation of interior-point methods for a class of nonsymmetric cones. J. Optim. Theory Appl. **204**(2), 33 (2025)
64. Tunçel, L.: Generalization of primal-dual interior-point methods to convex optimization problems in conic form. Found. Comput. Math. **1**(3), 229–254 (2001)
65. Vanderbei, R.J.: Symmetric quasidefinite matrices. SIAM J. Optim. **5**(1), 100–113 (1995)
66. Gill, P.E., Saunders, M.A., Shinnerl, J.R.: On the stability of Cholesky factorization for symmetric quasidefinite systems. SIAM J. Matrix Anal. Appl. **17**(1), 35–46 (1996)
67. Davis, T.A.: Algorithm 849: A concise sparse Cholesky factorization package. ACM Transactions on Mathematical Software **31**(4), 587–591 (2005)
68. Klabnik, S., Nichols, C.: The Rust programming language. No Starch Press, (2023)
69. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. SIAM Rev. **59**(1), 65–98 (2017)
70. Diamond, S., Boyd, S.: CVXPY: A Python-embedded modeling language for convex optimization. J. Mach. Learn. Res. **17**(83), 1–5 (2016)
71. Agrawal, A., Verschueren, R., Diamond, S., Boyd, S.: A rewriting system for convex optimization problems. Journal of Control and Decision **5**(1), 42–60 (2018)
72. CVXPY, https://www.cvxpy.org/
73. Blackford, L.S., Petitet, A., Pozo, R., et al.: An updated set of basic linear algebra subprograms (BLAS). ACM Transactions on Mathematical Software **28**(2), 135–151 (2002)
74. Intel Corporation, Intel oneAPI Math Kernel Library (oneMKL), (2024)
75. OpenBLAS: An optimized BLAS library, https://www.openblas.net/

76. El Kazdadi, S.: A linear algebra foundation for the Rust programming language, https://github.com/sarah-ek/faer-rs
77. Schenk, O., Gärtner, K.: "PARDISO," in Encyclopedia of Parallel Computing, pp. 1458–1464. Springer, US (2011)
78. Chen, Y., Davis, T. A., Hager, W. W., Rajamanickam, S.: "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate," ACM Trans. Math. Softw., vol. 35, no. 3, (2008)
79. Duff, I.S.: MA57-a code for the solution of sparse symmetric definite and indefinite systems. ACM Trans. Math. Softw. **30**(2), 118–144 (2004)
80. Kahan, W.: IEEE standard 754 for binary floating-point arithmetic. Lecture Notes on the Status of IEEE **754**(94720–1776), 11 (1996)
81. Gertz, E.M., Wright, S.J.: Object-oriented software for quadratic programming. ACM Transactions on Mathematical Software (TOMS) **29**(1), 58–81 (2003)
82. Milojkovic, N., Ghafari, M., Nierstrasz, O.: "It's duck (typing) season!" In 2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC), pp. 312–315 (2017)
83. Lubin, M., Dowson, O., Garcia, J.D., Huchette, J., Legat, B., Vielma, J.P.: JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. Math. Program. Comput. **15**(3), 581–589 (2023)
84. Goulart, P., Chen, Y.: Clarabel solver Benchmark suite, https://github.com/oxfordcontrol/ClarabelBenchmarks, (2024)
85. Mittelmann, H.: Benchmarks for optimization software, http://plato.asu.edu/bench.html
86. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
87. Maros, I., Mészáros, C.: A repository of convex quadratic programming problems. Optimization Methods and Software **11**(1–4), 671–681 (1999)
88. Davis, T. A., Hu, Y.: "The University of Florida Sparse Matrix Collection," ACM Transactions on Mathematical Software, vol. 38, no. 1, 1:1–1:25, (2011)
89. Huber, P.J.: Robust estimation of a location parameter. Ann. Math. Stat. **35**(1), 73–101 (1964)
90. Huber, P. J.: Robust Statistics. John Wiley & Sons, (1981)
91. Mangasarian, O.L., Musicant, D.R.: Robust linear and support vector regression. IEEE Trans. Pattern Anal. Mach. Intell. **22**(9), 950–955 (2000)
92. Kouzoupis, D., Zanelli, A., Peyrl, H., Ferreau, H.J.: "Towards proper assessment of QP algorithms for embedded model predictive control," in. European Control Conference (ECC) **2015**, 2609–2616 (2015)
93. Dongarra, J.J., Grosse, E.: Distribution of mathematical software via electronic mail. Commun. ACM **30**(5), 403–407 (1987)
94. Netlib LP test collection, https://netlib.org/lp
95. Friberg, H.A.: CBLIB 2014: A benchmark library for conic mixed-integer and continuous optimization. Math. Program. Comput. **8**(2), 191–214 (2016)
96. Babaeinejadsarookolaee, S., Birchfield, A., Christie, R. D., et al.: The Power Grid Library for benchmarking AC optimal power flow algorithms, https://arxiv.org/abs/1908.02788, (2021)
97. Coffrin, C., Bent, R., Sundar, K., Ng, Y., Lubin, M.: "Powermodels.jl: An open-source framework for exploring power flow formulations," in 2018 Power Systems Computation Conference (PSCC), (2018), pp. 1–8
98. Molzahn, D.K., Hiskens, I.A.: A survey of relaxations and approximations of the power flow equations. Foundations and Trends in Electric Energy Systems **4**(1–2), 1–221 (2019)
99. Coffrin, C. J., Roald, L. A.: Convex relaxations in power system optimization, a brief introduction, (2018)
100. Stott, B., Jardim, J., Alsac, O.: DC power flow revisited. IEEE Trans. Power Syst. **24**(3), 1290–1300 (2009)
101. Jabr, R.A.: Radial distribution load flow using conic programming. IEEE Trans. Power Syst. **21**(3), 1458–1459 (2006)
102. Borchers, B.: SDPLIB 1.2, a library of semidefinite programming test problems. Optimization Methods and Software **11**(1–4), 683–690 (1999)
103. Sturm, J.F.: Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. Optimization Methods and Software **11**(1–4), 625–653 (1999)
104. Tütüncü, R.H., Toh, K.-C., Todd, M.J.: Solving semidefinite-quadratic-linear programs using sdpt3. Math. Program. **95**(2), 189–217 (2003)