Mathematical Programming Computation manuscript No.
(will be inserted by the editor)

Warmstarting the Homogeneous and Self-Dual Interior Point Method for Linear and Conic Quadratic Problems

Anders Skajaa · Erling D. Andersen ·
Yinyu Ye

Received: date / Accepted: date

Abstract We present two strategies for warmstarting primal-dual interior point methods for the homogeneous self-dual model when applied to mixed linear and quadratic conic optimization problems. Common to both strategies is their use of only the *final* (optimal) iterate of the initial problem and their negligible computational cost. This is a major advantage when compared to previously suggested strategies that require a pool of iterates from the solution process of the initial problem. Consequently our strategies are better suited for users who use optimization algorithms as black-box routines which usually only output the final solution. Our two strategies differ in that one assumes knowledge only of the final *primal* solution while the other assumes the availability of both primal *and dual* solutions. We analyze the strategies and deduce conditions under which they result in improved theoretical worst-case complexity. We present extensive computational results showing work reductions when warmstarting compared to coldstarting in the range 30%–75% depending on the problem class and magnitude of the problem perturbation. The computational experiments thus substantiate that the warmstarting strategies are useful in practice.

Keywords Warmstart · Interior Point Method · Homogeneous model · Conic Programming

A. Skajaa
Department of Informatics and Mathematical Modelling
Technical University of Denmark, DK-2800, Kgs. Lyngby, Denmark.
E-mail: andsk@imm.dtu.dk

E. D. Andersen
MOSEK ApS, Fruebjergvej 3, Box 16
DK-2100, Copenhagen, Denmark.
E-mail: e.d.andersen@mosek.com

Y. Ye
Department of Management Science and Engineering
Stanford University, CA 94305-4121, USA.
E-mail: yinyu-ye@stanford.edu

1 Introduction

The problem of *warmstarting* an optimization algorithm occurs when one needs to solve a sequence of different but presumably related optimization problems. Let x^* denote the solution to an optimization problem $\mathcal{P}$. The aim is then to use the information contained in x^* to initialize the optimization algorithm at a particularly good (or *warm*) point when solving $\hat{\mathcal{P}}$, a related but different problem. Hopefully this will enable us to solve $\hat{\mathcal{P}}$ using less computational effort than had we not known or used x^*.

It is widely perceived that it is hard to warmstart interior point methods (IPM). The main reason is that if the solution x^* of $\mathcal{P}$ is on the boundary of the feasible region, then x^* is also close to the boundary for $\hat{\mathcal{P}}$ but not well-centered. At an iterate that is close to the boundary but not well-centered, IPMs generally behave badly producing either ill conditioned linear systems or search directions that allow only tiny step sizes. For that reason, progress towards the solution of $\hat{\mathcal{P}}$ is very slow and often it would have been better to simply coldstart the IPM. For the problem classes usually considered (this work included) x^* is effectively always on the boundary of $\mathcal{P}$.

Different warmstarting strategies for IPMs have previously been studied in e.g. [6–8, 10–12, 30], most often for the case of Linear Programming (LP). Common to several of these approaches is the requirement of more information from the solution process of $\mathcal{P}$ than just the final solution x^*. In both [11] and [30], for example, a pool of primal and dual (non-final) iterates from the solution process of $\mathcal{P}$ is required. Other approaches include (a) further perturbing $\hat{\mathcal{P}}$ to move the boundary and in that way avoid tiny stepsizes [14] and (b) allowing decreasing infeasibility of nonnegativity constraints yielding an "exterior point" method, see e.g. [21]. Computational results from several of the above references are generally positive in that they obtain reductions in the number of interior point iterations on the order of about 50% when perturbations are not too large. A problem often incurred, however, is a relatively costly procedure to compute the warm point. This is in particular seen in the comparisons of different warmstarting schemes in [12]. Very recently, a warm-starting method based on a *slack-approach* was introduced in [8]. Extra artificial variables are introduced to avoid any of the two above mentioned drawbacks and the method exibits promising numerical results. For further information about previous work on warmstarting IPMs, see the thorough overview in [8].

The contribution of the present paper is to introduce two warmstart strategies that use *only the final* optimal iterate of the solution of $\mathcal{P}$ and has low computational complexity. One of the strategies, $\mathrm{W_P}$, uses only the primal optimal solution x^* while the other, $\mathrm{W_{PD}}$, uses the primal x^* and the dual optimal solution (y^*, s^*) of $\mathcal{P}$. There are several reasons motivating these schemes. Firstly, optimization software is often used as black-box subroutines that output only *final* iterates. Hence intermediate non-optimal iterates or internal algorithmic variables may not be available at all. In such a situation, both strategies are useful. Secondly, sometimes just one optimization problem is to be solved, but a user with technical insight into the particular problem may know a good guess for the optimal primal solution. This information should be possible to utilize without requiring a guess for the dual solution as well. In this situation, the strategy W_P is useful.

It seems sensible to be modest in our expectations about the gains from warmstarting an IPM. Let the linear program $\{\min_x c^T x, \text{ s.t. } Ax = b,\ x \geq 0\}$ be denoted by $\text{LP}(A, b, c)$ and let x^* be its optimal primal solution. Megiddo [15] showed in 1991 that the existence of a strongly polynomial time algorithm for $\{\text{given } x^*, \text{ solve } \text{LP}(A, b, c)\}$ would imply the existence of a strongly polynomial time algorithm for $\{\text{solve } \text{LP}(A, b, c)\}$. Here "solve" means (a) finding an optimal solution *and* a certificate of optimality or (b) certify that no such solution exists. Thus even *checking* whether a given point is primal optimal (even if the point actually is a primal optimal solution) is likely to be as hard as simply solving the problem from scratch.

In this paper we consider general convex conic optimization problems of the form

$$\begin{array}{rl} \min_x & c^T x \\ \text{s.t.} & Ax = b \\ & x \in \mathcal{K} \end{array} \tag{1}$$

where $x, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and $\mathcal{K} \subseteq \mathbb{R}^n$ is a proper cone of the form

$$\mathcal{K} = \mathbb{R}^{n_\ell}_+ \ \times\ \mathcal{K}^{(q_1)}_\text{q} \times \cdots \times \mathcal{K}^{(q_{n_q})}_\text{q}. \tag{2}$$

Here, $\mathbb{R}^{n_\ell}_+$ denotes the positive orthant of dimension n_ℓ and $\mathcal{K}^{(k)}_q$ denotes the standard quadratic cone (or the Lorentz cone) of dimension k defined by

$$\mathcal{K}^{(k)}_\text{q} = \left\{ x \in \mathbb{R}^k : x_1 \geq \|(x_2, \ldots, x_k)\|_2 \right\} \tag{3}$$

We are further assuming that $m \leq n$ and that A has full row-rank. We have $n = n_\ell + \sum_{j=1}^{n_q} q_j$ and we will be using the notation $\nu = n_\ell + n_q$. Notice that if $q_j = 0$ for all j, the problem (1) reduces to an LP in standard form. A problem of the kind (1) is defined by the data A, b and c along with the cone $\mathcal{K}$ defined by n_ℓ and $q_1, \ldots, q_{n_q}$. We will only consider cases where the cone in $\mathcal{P}$ is identical to the cone in $\hat{\mathcal{P}}$ so we need only consider changes in A, b and c.

We begin by presenting the Homogeneous and Self-Dual (HSD) model for (1) and its dual in Section 2. We then analyze our warmstarting strategies from a theoretical worst-case complexity viewpoint in Section 3. In Section 4, we describe our IPM. Readers familiar with the standard concepts of homogeneous primal-dual interior point methods for mixed linear and quadratic cones can safely skip Section 4. Finally, in Section 5, we present extensive computational results and conclude with directions for future work in Section 6.

2 Homogeneous Self-Dual Model

Convex optimization has a very strong duality theory that connects the primal problem (1) to its *dual*, see e.g. [18]. Because of the strong relations between these two problems, modern interior point methods solve simultaneously the two problems making use of information from one to help progress in the other.

$$\text{PRIMAL} \begin{cases} \min_x \; c^T x \\ \text{s.t.} \;\; Ax = b \\ \qquad x \in \mathcal{K} \end{cases} \qquad \text{DUAL} \begin{cases} \max_{y,s} \; b^T y \\ \text{s.t.} \;\; A^T y + s = c \\ \qquad s \in \mathcal{K}^*, \; y \in \mathbb{R}^m \end{cases} \tag{4}$$

Here, $\mathcal{K}^*$ denotes the *dual cone* of $\mathcal{K}$, but when $\mathcal{K}$ is of the form (2), we have $\mathcal{K} = \mathcal{K}^*$. Therefore we can employ the very efficient primal-dual symmetric interior point methods [19,20]. However, instead of solving (4) directly, we aim to solve the *homogeneous and self-dual* model [28] of problems (4). This problem is slightly larger than the original problem, but in our situation there are enough benefits to offset the modest extra cost incurred. We present this model for the case that $\mathcal{K} = \mathbb{R}^n$, i.e. for linear programming.

For brevity, we will write $z = (x, \tau, y, s, \kappa) \in S := \mathbb{R}^n_+ \times \mathbb{R}_+ \times \mathbb{R}^m \times \mathbb{R}^n_+ \times \mathbb{R}_+$ and we introduce

$$\begin{aligned} r_p(z) &= Ax - b\tau \\ r_d(z) &= -A^T y - s + c\tau \\ r_g(z) &= -c^T x + b^T y - \kappa \\ \mu(z) &= (x^T s + \tau\kappa)/(\nu + 1). \end{aligned}$$

Now let $z^0 = (x^0, \tau^0, y^0, s^0, \kappa^0) \in S$ be some initial point. Assume θ is a scalar variable. We then consider the problem

$$\begin{cases} \begin{array}{llllll} \min_{(z,\theta)} & \theta\mu(z^0) & & & & \\ \text{s.t.} & & Ax & -b\tau & & = \theta\, r_p(z^0) \\ & -A^T y & & +c\tau & -s & = \theta\, r_d(z^0) \\ & b^T y & -c^T x & & -\kappa & = \theta\, r_g(z^0) \\ & r_p(z^0)^T y & -r_d(z^0)^T x & +r_g(z^0)\tau & & = \mu(z^0) \\ & \multicolumn{5}{l}{(x,\tau) \geq 0, \;\; (s,\kappa) \geq 0, \;\; (y,\theta) \text{ free.}} \end{array} \end{cases} \tag{5}$$

The following lemma explains the advantages of solving (5) instead of (4):

Lemma 1 *Assume* (z, θ) *is a solution of* (5)*. Then* $\theta = 0$ *and*

(i) if $\tau > 0$ *then* $(x, y, s)/\tau$ *is optimal for* (4)*;*
(ii) if $\kappa > 0$ *then, one or both of* $b^T y > 0$ *and* $c^T x < 0$ *hold. If the first holds, then* (4) *is primal infeasible. If the second holds, then* (4) *is dual infeasible.*

So any solution to (5) with $\tau + \kappa > 0$ provides either an optimal solution to our original problems (4) or a certificate of infeasibility of (one of) the original problems. See [13,25,29] for a proof and further details.

Advantages of using the HSD-model thus include the ability to detect infeasibilities in the problem and particularly the ease of finding a suitable starting point will be of importance to us later. This latter property also eliminates the need for a Phase I procedure in the interior point method.

3 Warmstarting

Since (5) is a linear program, we may solve it using any algorithm for linear programming that generates a solution with $\tau + \kappa > 0$. The point z^0 used in generating the HSD-model is by construction a feasible point for the problem, so we can use a *feasible-start* IPM initialized in z^0 to solve the problem. To obtain the best known complexity, we could, for example, apply the Mizuno-Todd-Ye feasible predictor-corrector interior point algorithm [17]. If we initialize this algorithm in z^0, the worst-case iteration complexity to obtain a solution or an infeasibility-certificate to (4) at the tolerance ϵ (see Section 4.3) is given by

$$\mathcal{O}\left(\sqrt{n}\log\left(\Psi(z^0)/\epsilon\right)\right), \quad \text{where } \Psi(z) = \max\left\{\mu(z), \|r_p(z)\|, \|r_d(z)\|\right\}, \tag{6}$$

see particularly [29, pp. 169]. $\|\cdot\|$ denotes some norm. In practice when solving the HSD-model, one usually initializes the algorithm from the point $\mathrm{C} := (e, 1, 0, e, 1)$. Here e denotes the vector of all ones of length n and 0 denotes a zero-vector of length m. We will refer to starting from this point as a *cold start*. To obtain a better worst-case complexity, we would need to initialize the algorithm in a point z^0 satisfying $\Psi(z^0) < \Psi(\mathrm{C})$, which is certainly satisfied if

$$\mu(z^0) < \mu(\mathrm{C}), \quad \|r_p(z^0)\| < \|r_p(\mathrm{C})\|, \quad \|r_d(z^0)\| < \|r_d(\mathrm{C})\|. \tag{7}$$

For the above complexity result to hold, the initial point z^0 must lie in the central-path neighborhood $\mathcal{N}_2(\eta)$, defined by

$$\mathcal{N}_2(\eta) = \{z \in S : \ \|(x \circ s, \tau\kappa) - \mu(e, 1)\|_2 \le \eta\mu\}, \quad \text{for } \eta \in (0, 1). \tag{8}$$

where $\circ$ denotes *elementwise* product of vectors of equal length. That is, $(v \circ w)_i = v_i w_i$ for all i. Since $\mu(\mathrm{C}) = 1$, we clearly have $\mathrm{C} \in \mathcal{N}_2(\eta)$, but we must generally make sure that our initial point is in $\mathcal{N}_2(\eta)$.

3.1 Warm Starting Points

Now let x^* be the primal optimal solution and (y^*, s^*) the dual optimal solution of a linear program $\mathcal{P}$. Further let $\lambda \in [0, 1)$ and $\mu^0 > 0$ be (user chosen) parameters. We propose the following two starting points for the initialization of a related but different linear program $\hat{\mathcal{P}}$:

$$(\mathrm{W_P}) \begin{cases} x^0 = \lambda x^* + (1-\lambda)e \\ s^0 = \mu^0 (x^0)^{-1} \\ y^0 = 0 \\ \tau^0 = 1 \\ \kappa^0 = \mu^0 \end{cases} \qquad (\mathrm{W_{PD}}) \begin{cases} x^0 = \lambda x^* + (1-\lambda)e \\ s^0 = \lambda s^* + (1-\lambda)e \\ y^0 = \lambda y^* \\ \tau^0 = 1 \\ \kappa^0 = (x^0)^T s^0/n \end{cases} \tag{9}$$

Here, $(x^0)^{-1}$ denotes the *elementwise* reciprocal of x^0. Much computational experience [1,2,23] indicates that the starting point C seems to work well for the initialization of an interior point method to solve the HSD-model. We can view the starting point $\mathrm{W_{PD}}$ as a convex combination of (x^*, y^*, s^*) and C. Thus, hopefully, $\mathrm{W_{PD}}$ is a point closer (in some sense) to the solution of $\hat{\mathcal{P}}$, but incorporation of $(1-\lambda)\mathrm{C}$ introduces enough centrality to avoid tiny step sizes. The point $\mathrm{W_P}$ is identical to $\mathrm{W_{PD}}$ for the primal variable, but, as we restrict ourselves to using *only primal information*, we cannot do the same for the dual variables. Instead we choose s^0 so that the point is perfectly centered and has a prescribed duality gap, namely μ^0.

Since strategy $\mathrm{W_P}$ uses only the primal solution x^*, it is especially suited for situations where just one optimization problem is to be solved, but the user may have a qualified guess at a point close to the primal optimal solution or for some other reason, only the primal optimal solution to $\mathcal{P}$ is available. The strategy $\mathrm{W_{PD}}$ uses the full primal-dual solution (x^*, y^*, s^*) and is hence suited for the situation where a sequence of optimization problems is to be solved and a black-box routine for solving (4) that outputs (x^*, y^*, s^*) is used internally as a part of a larger program. We will see several examples of both in Section 5.

Our goal in the rest of this section is, for each of the two starting points, to deduce conditions under which they satisfy (7). We remark that (7) are sufficient conditions for $\Psi(z^0) < \Psi(\mathrm{C})$ but not necessary since no attention is paid to which term in (6) is the dominating one.

3.2 Comparison of Primal and Dual Residuals

We first introduce some notation consistent with that of [8]. The original LP instance $\mathcal{P}$ consists of the data triplet $d^\circ = (A^\circ, b^\circ, c^\circ)$. We will denote the perturbation by $\Delta d = (\Delta A, \Delta b, \Delta c)$ so that the perturbed problem $\hat{\mathcal{P}}$ has data

$$d = (A, b, c) = d^\circ + \Delta d = (A^\circ, b^\circ, c^\circ) + (\Delta A, \Delta b, \Delta c).$$

As in [8], we will measure the relative magnitude of perturbation between the two problems by the quantities $(\alpha,\ \alpha',\ \beta,\ \gamma)$, defined via:

$$\begin{aligned} \|\Delta A\| &\le \alpha \|A^\circ\| \\ \|\Delta A^T\| &\le \alpha' \|A^{\circ T}\| \\ \|\Delta b\| &\le \beta \|b^\circ\| \\ \|\Delta c\| &\le \gamma \|c^\circ\|. \end{aligned}$$

We are now ready to present three lemmas that, for both $\mathrm{W_P}$ and $\mathrm{W_{PD}}$, state when their primal and dual residuals are smaller than those of C.

Notice that $r_p(\mathrm{W_P}) = r_p(\mathrm{W_{PD}})$ so the lemma below applies to both points:

Lemma 2 *Define* $\delta_p = \max\{(\|x^*\| + \|e\|)\alpha,\ 2\beta\}$. *If*

$$\delta_p \le \frac{\|A^\circ e - b^\circ\|}{\|A^\circ\| + \|b^\circ\|}$$

then

$$\|r_p(\text{W}_\text{P})\| = \|r_p(\text{W}_\text{PD})\| \leq \|r_p(\text{C})\|.$$

Proof If $\lambda = 0$, $r_p(\text{W}_\text{P}) = r_p(\text{C})$ so the statement holds trivially. For $\lambda \in (0, 1)$,

$$\begin{aligned} r_p(\text{W}_\text{P}) &= Ax^0 - b\tau^0 \\ &= \lambda(Ax^* - b) + (1-\lambda)(Ae - b) \\ &= \lambda(\varDelta Ax^* - \varDelta b) + (1-\lambda) r_p(\text{C}). \end{aligned}$$

Therefore,

$$\|r_p(\text{W}_\text{P})\| \leq \lambda(\alpha\|A^\circ\|\|x^*\| + \beta\|b^\circ\|) + (1-\lambda)\|r_p(\text{C})\|.$$

Similarly,

$$\begin{aligned} \|r_p(\text{C})\| &= \|Ae - b\| = \|A^\circ e - b^\circ + \varDelta Ae - \varDelta b\| \\ &\geq \|A^\circ e - b^\circ\| - (\alpha\|A^\circ\|\|e\| + \beta\|b^\circ\|) \end{aligned}$$

and therefore,

$$\begin{aligned} \|r_p(\text{C})\| - \|r_p(\text{W}_\text{P})\| &\geq \|r_p(\text{C})\| - \lambda(\alpha\|A^\circ\|\|x^*\| + \beta\|b^\circ\|) - (1-\lambda)\|r_p(\text{C})\| \\ &= \lambda\|r_p(\text{C})\| - \lambda(\alpha\|A^\circ\|\|x^*\| + \beta\|b^\circ\|) \qquad \Rightarrow \\ \frac{1}{\lambda}\left(\|r_p(\text{C})\| - \|r_p(\text{W}_\text{P})\|\right) &\geq \|r_p(\text{C})\| - (\alpha\|A^\circ\|\|x^*\| + \beta\|b^\circ\|) \\ &\geq \|A^\circ e - b^\circ\| - (\alpha\|A^\circ\|\|e\| + \beta\|b^\circ\|) \\ &\quad - (\alpha\|A^\circ\|\|x^*\| + \beta\|b^\circ\|) \\ &= \|A^\circ e - b^\circ\| - (\|x^*\| + \|e\|)\alpha\|A^\circ\| - 2\beta\|b^\circ\| \\ &\geq \|A^\circ e - b^\circ\| - \delta_p\left(\|A^\circ\| + \|b^\circ\|\right). \end{aligned} \tag{10}$$

The statement then follows after a rearrangement of (10) ≥ 0. □

We remark that the preceding lemma is very similar in nature to the ones found in [8, sec. 3.1].

The dual parts of W_P and W_PD are not identical. Let us begin with W_P:

Lemma 3 *Define* $\psi = \|e\|^{-1}(\|c - e\| - \|c\|)$. *If*

$$\psi \geq \mu^0(1-\lambda)^{-1}$$

then

$$\|r_d(\text{W}_\text{P})\| \leq \|r_d(\text{C})\|.$$

Proof We have $\|r_d(\text{C})\| = \|c - e\|$ and

$$\begin{aligned} \|r_d(\text{W}_\text{P})\| &= \|c - \mu^0(x^0)^{-1}\| \\ &\leq \|c\| + \mu^0\|(x^0)^{-1}\| \\ &\leq \|c\| + \mu^0(1-\lambda)^{-1}\|e\| \end{aligned}$$

The statement follows by using this latter inequality to show that $\|r_d(\text{C})\| - \|r_d(\text{W}_\text{P})\| \geq 0$ by simple rearrangement. □

The statement for $r_d(\text{w}_{\text{PD}})$ is very similar to the one in Lemma 2:

Lemma 4 *Define* $\delta_d = \max\{\alpha'\|y^*\|,\ 2\gamma\}$. *If*

$$\delta_d \leq \frac{\|c^\circ - e\|}{\|A^{\circ T}\| + \|c^\circ\|}$$

then

$$\|r_d(\text{w}_{\text{PD}})\| \leq \|r_d(\text{c})\|.$$

Proof If $\lambda = 0$, $r_d(\text{w}_{\text{PD}}) = r_d(\text{c})$, so the statement holds trivially. For $\lambda \in (0,1)$, we get from manipulations similar to those in Lemma 2 that

$$\begin{aligned} r_d(\text{w}_{\text{PD}}) &= \lambda(-\Delta A^T y^* + \Delta c) + (1-\lambda) r_d(\text{c}) && \Rightarrow \\ \|r_d(\text{w}_{\text{PD}})\| &\leq \lambda(\alpha'\|A^{\circ T}\|\|y^*\| + \gamma\|c^\circ\|) + (1-\lambda)\|r_d(\text{c})\|. \end{aligned}$$

Similarly, $\|r_d(\text{c})\| = \|c^\circ + \Delta c - e\| \geq \|c^\circ - e\| - \gamma\|c^\circ\|$. Therefore,

$$\begin{aligned} \|r_d(\text{c})\| - \|r_d(\text{w}_{\text{PD}})\| &\geq \lambda\|r_d(\text{c})\| - \lambda(\alpha'\|A^{\circ T}\|\|y^*\| + \gamma\|c^\circ\|) \qquad \Rightarrow \\ \frac{1}{\lambda}\left(\|r_d(\text{c})\| - \|r_d(\text{w}_{\text{PD}})\|\right) &\geq \|r_d(\text{c})\| - (\alpha'\|A^{\circ T}\|\|y^*\| + \gamma\|c^\circ\|) \\ &\geq \|c^\circ - e\| - \gamma\|c^\circ\| - (\alpha'\|A^{\circ T}\|\|y^*\| + \gamma\|c^\circ\|) \\ &= \|c^\circ - e\| - \alpha'\|A^{\circ T}\|\|y^*\| - 2\gamma\|c^\circ\| \\ &\geq \|c^\circ - e\| - \delta_d\left(\|A^{\circ T}\| + \|c^\circ\|\right) \end{aligned} \tag{11}$$

The statement then follows after a rearrangement of (11) ≥ 0. □

The three preceding lemmas state conditions under which the primal and dual residuals, for each of the two points, are smaller in norm than those of c. Combined with the lemmas in the following section, this will allow us to present conditions under which we obtain an improved worst-case complexity.

3.3 Comparison of Centrality and Complementarity Gap

We also need results about the centrality and initial penalty value $\mu(z^0)$ of our warm points.

We start with w_P, for which the situation is particularly simple: We have $\mu(\text{w}_\text{P}) = \mu^0$ directly from the definition of w_P. So for a better initial complementarity gap than the cold start, we must choose $\mu^0 \leq 1 = \mu(\text{c})$. Now let us apply this to Lemma 3: Assume that $\psi \geq 0$. The condition in Lemma 3 states

$$\begin{aligned} \mu^0(1-\lambda)^{-1} &\leq \psi && \Leftrightarrow \\ (1-\lambda) &\geq \mu^0/\psi && \Leftrightarrow \\ \lambda &\leq 1 - \mu^0/\psi. \end{aligned} \tag{12}$$

Since we must take $\lambda \in [0,1)$, (12) implies that we must require $\mu^0 \leq \psi$. We remark that the condition of Lemma 3 can only be satisfied if $\psi > 0$, which is a rather strong requirement. Notice also that $\mathrm{W_P}$ is perfectly centered, so it automatically satisfies any neighborhood requirement imposed by the algorithm.

For $\mathrm{W_{PD}}$, the situation is more complicated: Define

$$\xi = e^T(x^* + s^*)/n.$$

We can then state the following lemma which expresses when the initial complementarity of $\mathrm{W_{PD}}$ is smaller than that of C:

Lemma 5 *If*

$$\begin{aligned} &\xi \in (0,2] \quad \textit{or} \\ &\xi > 2 \quad \textit{and} \quad \lambda \geq 1 - \frac{1}{\xi - 1} \end{aligned} \tag{13}$$

then

$$\mu(\mathrm{W_{PD}}) \leq 1.$$

Proof We have

$$\begin{aligned} x^0 \circ s^0 &= (\lambda x^* + (1-\lambda)e) \circ (\lambda s^* + (1-\lambda)e) \\ &= \lambda^2(x^* \circ s^*) + (1-\lambda)^2 e + \lambda(1-\lambda)(x^* + s^*) \\ &= \lambda(1-\lambda)(x^* + s^*) + (1-\lambda)^2 e \end{aligned} \tag{14}$$

where we used that $x^* \circ s^* = 0$. Therefore

$$\begin{aligned} \mu(\mathrm{W_{PD}}) &= \frac{(x^0)^T s^0 + \tau^0\kappa^0}{n+1} = \frac{(x^0)^T s^0 + \frac{(x^0)^T s^0}{n}}{n+1} = \frac{(x^0)^T s^0}{n} \\ &= \frac{1}{n} e^T(x^0 \circ s^0) = \lambda(1-\lambda)\xi + (1-\lambda)^2 \end{aligned} \tag{15}$$

$$\begin{aligned} &\mu(\mathrm{W_{PD}}) \leq 1 \quad \Leftrightarrow \\ &\lambda(1-\xi) \leq 2 - \xi \end{aligned} \tag{16}$$

Clearly, (16) holds for $\xi \in [0,2]$ because $\lambda \in (0,1)$. If $\xi > 2$, then (16) is equivalent to $\lambda \geq (2-\xi)/(1-\xi) = 1 - 1/(\xi - 1)$. □

Lemma 5 imposes a *lower* bound on λ when $\xi > 2$. Notice that as $\xi \to \infty$, the lower bound approaches 1, collapsing the width of the interval for λ to zero, because $\lambda \in [0,1]$.

The situation for $\mathrm{W_{PD}}$ is further complicated by the fact that it, unlike $\mathrm{W_P}$, is not necessarily in $\mathcal{N}_2(\eta)$. Let us define the quantity

$$\pi = \|\xi^{-1}(x^* + s^*) - e\|_2.$$

The following lemma gives conditions under which $\mathrm{W_{PD}}$ is sufficiently central.

Lemma 6 *If*

$$\lambda\xi(\pi-\eta) \leq \eta(1-\lambda) \tag{17}$$

then

$$\mathrm{W_{PD}} \in \mathcal{N}_2(\eta).$$

Proof First notice that $\tau^0\kappa^0 - \mu(\mathrm{W_{PD}}) = 0$ so this term does not contribute in the norm in (8). Now from (14) and (15) we obtain

$$\begin{aligned} x^0 \circ s^0 - \mu(\mathrm{W_{PD}})e &= \lambda(1-\lambda)(x^*+s^*) + (1-\lambda)^2 e \qquad (18)\\ &\quad - \lambda(1-\lambda)\xi e - (1-\lambda)^2 e \\ &= \lambda(1-\lambda)\left(x^*+s^*-\xi e\right) \qquad \Rightarrow \\ \|(x^0\circ s^0, \tau^0\kappa^0) - \mu(e,1)\| &= \lambda(1-\lambda)\xi\pi \end{aligned}$$

Therefore using (17):

$$\begin{aligned} \lambda\xi(\pi-\eta) &\leq \eta(1-\lambda) && \Rightarrow \\ \lambda\xi\pi &\leq \eta(\lambda\xi + (1-\lambda)) && \Rightarrow \\ \lambda(1-\lambda)\xi\pi &\leq \eta(\lambda(1-\lambda)\xi + (1-\lambda)^2) && \Rightarrow \\ \|(x^0\circ s^0, \tau\kappa) - \mu(e,1)\| &\leq \eta\mu(\mathrm{W_{PD}}) \end{aligned}$$

which is the statement. □

We now have a lower bound (13) and an upper bound (17) on λ so we can determine conditions under which there is a non-empty interval for λ which will imply that $\mathrm{W_{PD}}$ is sufficiently central and simultaneously has smaller initial complementary gap than C:

Lemma 7 *Define the following quantities:*

$$q = \frac{\eta}{\xi\pi + \eta(1-\xi)}, \quad \xi_1 = \frac{\xi-1}{\xi}, \quad \xi_2 = \frac{(\xi-1)^2}{\xi(\xi-2)}, \quad \xi_3 = \xi_1/\xi_2 = \frac{\xi-2}{\xi-1}.$$

We can then distinguish the following cases, all of which have the same conclusion, which is stated afterwards:

1:	*Assume* $0<\xi\leq 1$.		*If* $\lambda\in(0,q)$,
2(a):	*Assume* $1<\xi\leq 2$	*and* $\pi\leq\eta\xi_1$.	*If* $\lambda\in(0,1)$,
2(b):	*Assume* $1<\xi\leq 2$	*and* $\pi>\eta\xi_1$.	*If* $\lambda\in(0,q)$,
3(a):	*Assume* $\xi>2$	*and* $\pi\leq\eta\xi_1$.	*If* $\lambda\in(\xi_3,1)$,
3(b):	*Assume* $\xi>2$	*and* $\eta\xi_1<\pi\leq\eta$.	*If* $\lambda\in(\xi_3,1)$,
3(c):	*Assume* $\xi>2$	*and* $\eta<\pi<\eta\xi_2$.	*If* $\lambda\in(\xi_3,q)$,

then

$$\mu(\mathrm{W_{PD}}) \leq 1 \quad \textit{and} \quad \mathrm{W_{PD}} \in \mathcal{N}_2(\eta).$$

Proof First notice that if $\xi \leq 1$, then Lemma 5 imposes no restriction on λ, so the lower bound on λ is 0. If $\xi \leq 1$, then $1 - \xi \geq 0$ so (17) can be written (after some simple manipulation) as $\lambda \leq q$.

If $1 < \xi \leq 2$ then the lower bound on λ is still 0, for the same reason as above. However, (17) may now be written

$$\lambda \left[\xi\pi + \eta(1-\xi)\right] \leq \eta. \tag{19}$$

The expression in the hard brackets might be negative, which happens if $\pi \leq \eta(\xi - 1)/\xi = \eta\xi_1$. In this case, the condition (19) turns into $\lambda \geq q$, but then $q < 0$, so this is already satisfied for $\lambda \geq 0$. Thus if $\pi \leq \eta\xi_1$, we can allow $\lambda \in (0, 1)$. If on the other hand $\pi > \eta\xi_1$, the expression in the hard brackets of (19) is positive, and we can write it simply as $\lambda \leq q$.

If $\xi > 2$, Lemma 5 requires $\lambda \geq (\xi - 2)/(\xi - 1) = \xi_3$ while Lemma 6 only imposes an upper bound on λ if $\pi > \eta\xi_1$. In this case, the two lemmas require $\lambda \in (\xi_3, q)$, which is only a non-empty interval if $q > \xi_3$. This latter inequality holds precisely when $\pi < \eta\xi_2$. This accounts for all cases. □

3.4 Summary

Using all of the Lemmas 2–7, we can now summarize the conditions under which we get better worst-case complexity for each of the two points. We begin with $\mathrm{W_P}$:

Proposition 1 *If*

1. $\delta_p := \max\{(\|x^*\| + \|e\|)\alpha,\ 2\beta\} \leq (\|A^\circ\| + \|b^\circ\|)^{-1}\|A^\circ e - b^\circ\|$
2. $\|c - e\| \geq \|c\|$
3. *we choose* $\mu^0 \in (0, \psi)$ *and finally*
4. *we choose* $\lambda \in (0, 1 - \mu^0/\psi)$

then starting in $\mathrm{W_P}$ *results in a better worst-case complexity than a coldstart.*

Similarly for $\mathrm{W_{PD}}$:

Proposition 2 *If*

1. $\delta_p := \max\{(\|x^*\| + \|e\|)\alpha,\ 2\beta\} \leq (\|A^\circ\| + \|b^\circ\|)^{-1}\|A^\circ e - b^\circ\|$
2. $\delta_d := \max\{\alpha'\|y^*\|,\ 2\gamma\} \leq (\|{A^\circ}^T\| + \|c^\circ\|)^{-1}\|c^\circ - e\|$ *and*
3. *the conditions of one of the six cases of Lemma 7 are satisfied,*

then starting in $\mathrm{W_{PD}}$ *results in a better worst-case complexity than a coldstart.*

Thus we have established sufficient conditions under which we have improved worst-case complexity by warmstarting. We are, however, aware of the apparent gap between IPM complexity theory and state-of-the-art implementations, which in most cases perform much better than the worst case complexity estimates. Indeed, the algorithm described in the following sections is in practice usually superior to the predictor-corrector algorithm for which we have just derived complexity estimates relating warmstarts to coldstarts. It is therefore more fruitful to think of the results above as *conceptual* and purely theoretical justifications. That is, these statements should be seen as an attempt to show the existence of conditions under which the warmstarting strategies imply improved worst-case performance for the best-known algorithm in terms of theoretical complexity. However whether the warmstart strategies are effective in practice for the *practically* best-known algorithm shall be determined via computational experiments. For that reason, we devote the rest of the paper to such experiments. In the following section we present the actual algorithm used in experiments. Then, we show a series of computational evidences supporting the effectiveness of the warmstart strategies in Section 5.

4 Symmetric Primal-Dual Interior Point Algorithm

To carry out numerical experiments, we have implemented in MATLAB a symmetric primal-dual interior point method called CCOPT. It uses the Nesterov-Todd scaling and Mehrotra's second order correction. Following [2], we give in this section a brief overview of the algorithm. We consider first the case of linear programming, i.e. $\mathcal{K} = \mathbb{R}^n_+$, and then show how we handle the more general quadratic cones (3). A reader familiar with the standard ideas in this algorithm can safely skip this entire section. We use our own implementation instead of other public domain software because it is then easier to modify, control and monitor the algorithm. We remark that all of our source code is publicly available[1] and a reader can therefore reproduce and verify any of the following computational experiments.

4.1 Simplified Homogeneous Self-Dual Model

Instead of solving (5), our algorithm solves a slightly simpler version known as the *simplified* HSD-model [27]:

$$Ax - b\tau = 0 \tag{20}$$

$$-A^T y - s + c\tau = 0 \tag{21}$$

$$-c^T x + b^T y - \kappa = 0 \tag{22}$$

$$x \geq 0,\ s \geq 0,\ y \in \mathbb{R}^m,\ \tau \geq 0,\ \kappa \geq 0 \tag{23}$$

The HSD-model (5) and the simplied HSD-model (20)–(23) are closely related. See [25, 27] for results in this direction. The important points are that we retain the ability to detect infeasibility and our warmstarting strategies are still valid.

[1] `http://www2.imm.dtu.dk/~andsk/files/warmstart/downloadcode.html`.

4.2 Algorithm for Linear Programming

Assume $z^0 = (x^0, \tau^0, y^0, s^0, \kappa^0) \in \mathcal{K} \times \mathbb{R}_+ \times \mathbb{R}^m \times \mathcal{K} \times \mathbb{R}_+$ is the initial point and $\mu^0 = \mu(z^0)$ its complementarity gap. We then define the central path, parametrized by $\rho \in [0,1]$, for (20)–(23) by

$$Ax - b\tau = \rho(Ax^0 - b\tau^0) \tag{24}$$
$$-A^T y - s + c\tau = \rho(-A^T y^0 - s^0 + c\tau^0) \tag{25}$$
$$-c^T x + b^T y - \kappa = \rho(-c^T x^0 + b^T y^0 - \kappa^0) \tag{26}$$
$$x \circ s = \rho\mu^0 e \tag{27}$$
$$\tau\kappa = \rho\mu^0 \tag{28}$$

The idea of a primal-dual interior point algorithm for the simplified HSD-model is to loosely track the central path (24)–(28) towards a solution of (20)–(23). Notice that (24)–(26) are the feasibility equations while (27)–(28) are relaxed complementarity conditions. As $\rho \to 0$, we are guided towards an optimal solution for (20)–(23).

In each iteration we compute the direction $(d_x, d_\tau, d_y, d_s, d_\kappa)$ which is the solution to the system of linear equations (29)–(33):

$$Ad_x - bd_\tau = (\sigma - 1)(Ax - b\tau) \tag{29}$$
$$-A^T d_y - d_s + cd_\tau = (\sigma - 1)(-A^T y - s + c\tau) \tag{30}$$
$$-c^T d_x + b^T d_y - d_\kappa = (\sigma - 1)(-c^T x + b^T y - \kappa) \tag{31}$$
$$\tau d_\kappa + \kappa d_\tau = -\tau\kappa + \sigma\mu - d_{\tau\kappa} \tag{32}$$
$$x \circ d_s + s \circ d_x = -x \circ s + \sigma\mu e - d_{xs} \tag{33}$$

where (x, τ, y, s, κ) is the current iterate and μ its duality gap. The numbers σ and $d_{\tau\kappa}$ and the vector d_{xs} are computed by first solving (29)–(33) with $\sigma = d_{\tau\kappa} = 0$ and $d_{xs} = 0$. Let us denote the solution to this (pure Newton) system $(\hat{d}_x, \hat{d}_\tau, \hat{d}_y, \hat{d}_s, \hat{d}_\kappa)$. We then compute

$$\hat{\alpha} = \max_\alpha \left\{ \alpha : (x, \tau, y, s, \kappa) + \alpha(\hat{d}_x, \hat{d}_\tau, \hat{d}_y, \hat{d}_s, \hat{d}_\kappa) \geq 0 \right\} \tag{34}$$

and set

$$\sigma = (1 - \hat{\alpha}) \min\left(0.5, (1 - \hat{\alpha})^2\right) \tag{35}$$

The *Mehrotra second order correctors* [16] d_{xs} and $d_{\tau\kappa}$ are computed by

$$d_{\tau\kappa} = \hat{d}_\tau \hat{d}_\kappa \quad \text{and} \quad d_{xs} = \hat{d}_x \circ \hat{d}_s \tag{36}$$

After computing σ, $d_{\tau\kappa}$ and d_{xs} by (35)–(36) we compute the final search direction by solving (29)–(33) again but with a now altered right hand side. The iterate is then updated by taking a step of length α in this direction: $(x, \tau, y, s, \kappa) := (x, \tau, y, s, \kappa) + \alpha(d_x, d_\tau, d_y, d_s, d_\kappa)$. It should be stressed that only the right hand side changes so the factorization from the first solve can be used again. The step size α is chosen to be maximal under the conditions that the iterate stays feasible in the cone and that the iterates stay within a certain neighborhood of the central-path. See e.g. [29, pp. 128] for several reasonable definitions of such a neighborhood.

4.3 Termination

Assume (x, τ, y, s, κ) is the current iterate and consider the following inequalities:

$$\|Ax - \tau b\|_\infty \leq \epsilon \cdot \max\{1, \|[A, b]\|_\infty\} \tag{P}$$
$$\|A^T y + s - c\tau\|_\infty \leq \epsilon \cdot \max\{1, \left\|\left[A^T, I, -c\right]\right\|_\infty\} \tag{D}$$
$$\left|-c^T x + b^T y - \kappa\right| \leq \epsilon \cdot \max\{1, \|\left[-c^T, b^T, 1\right]\|_\infty\} \tag{G}$$
$$\left|c^T x/\tau - b^T y/\tau\right| \leq \epsilon \cdot \left(1 + \left|b^T y/\tau\right|\right) \tag{A}$$
$$\tau \leq \epsilon \cdot 10^{-2} \cdot \max\{1, \kappa\} \tag{T}$$
$$\tau \leq \epsilon \cdot 10^{-2} \cdot \min\{1, \kappa\} \tag{K}$$
$$\mu \leq \epsilon \cdot 10^{-2} \cdot \mu^0 \tag{M}$$

We then terminate and conclude as follows:

$$\begin{array}{rr}
(\text{OPT}) & (\text{P}) \wedge (\text{D}) \wedge (\text{A}) \Rightarrow \text{Feasible and optimal solution found} \\
(\text{INFEAS}) & (\text{P}) \wedge (\text{D}) \wedge (\text{G}) \wedge (\text{T}) \Rightarrow \text{Problem primal or dual infeasible} \\
(\text{ILLP}) & (\text{K}) \wedge (\text{M}) \Rightarrow \text{Problem ill-posed}
\end{array}$$

In case (OPT), the optimal solution $(x, y, s)/\tau$ is returned. If we find (INFEAS), the problem is dual infeasible if $c^T x < 0$ and primal infeasible if $b^T y > 0$. The number $\epsilon > 0$ is a user-specified tolerance.

4.4 Generalization to Quadratic Cones

In order to handle the more general quadratic cones alongside the positive orthant, it is necessary to modify only a few steps in the algorithm in Section 4.2. Notationally, this is facilitated by generalizing the product $\circ$ as follows (see e.g. [23] for many more details). First define

$$e_+^k := (1, 1, \dots, 1)^T \in \mathbb{R}^k$$
$$e_\mathrm{q}^k := (1, 0, \dots, 0)^T \in \mathbb{R}^k$$

and for $x \in \mathbb{R}^k$:

$$\mathrm{mat}_+(x) := \mathrm{diag}(x) \in \mathbb{R}^{k \times k}$$
$$\mathrm{mat}_\mathrm{q}(x) := \begin{pmatrix} x_1 & x_{2:k}^T \\ x_{2:k} & x_1 I_{k-1} \end{pmatrix} \in \mathbb{R}^{k \times k}$$

For an $x \in \mathbb{R}^{n_\ell}_+ \times \Pi_{j=1}^{n_q} \mathcal{K}_{\mathrm{q}}^{(q_j)}$ partitioned by $x = (x_+, x_{\mathrm{q}}^{(1)}, \dots, x_{\mathrm{q}}^{(n_q)})$ we then define

$$\mathrm{mat}(x) = \mathrm{mat}_+(x_+) \oplus \mathrm{mat}_{\mathrm{q}}(x_{\mathrm{q}}^{(1)}) \oplus \cdots \oplus \mathrm{mat}_{\mathrm{q}}(x_{\mathrm{q}}^{(n_q)})\,. \tag{37}$$

where $\oplus$ denotes direct matrix sum. So $\mathrm{mat}(x)$ is a block-diagonal matrix, where the blocks are the individual terms of the right-hand-side of (37). Similarly, we re-define $e := (e_+^{n_\ell}, e_{\mathrm{q}}^{q_1}, \dots, e_{\mathrm{q}}^{q_{n_q}})$. If $y \in \mathcal{K}$ is partitioned in the same manner as x, we finally re-define $\circ$ by

$$x \circ y := \mathrm{mat}(x)\, y$$

and the inverse

$$x^{-1} := \mathrm{mat}(x)^{-1} e.$$

It is easy to see that $x \circ x^{-1} = x^{-1} \circ x = e$.

When applying the algorithm to problems with mixed linear and quadratic cones, the search direction is instead the solution to the linear equations (29)-(32) and the equation

$$\Psi B^{-1} d_s + \Psi B d_x = -\psi \circ \psi + \sigma \mu e - d_{xs}. \tag{38}$$

Here we have introduced the notation $\Psi := \mathrm{mat}(\psi)$ and $\psi = Bx$, where B is a so called *scaling matrix*, chosen to ensure the primal-dual symmetry of the algorithm (see e.g. [24] for more details). Several different choices for B exist but in this algorithm we use the particularly interesting *Nesterov-Todd scaling* [19,20], determined such that B satisfies $Bx = B^{-1}s$. This scaling matrix has proven very efficient in practice [2,23]. The numbers σ and $d_{\tau\kappa}$ are determined as in Section 4.2, but now d_{xs} is computed by

$$d_{xs} = (B\hat{d}_x) \circ (B^{-1}\hat{d}_s). \tag{39}$$

We remark that all operations involving B can be carried out in $\mathcal{O}(n)$ floating point operations. Thus for example computing Bx or $B^{-1}\hat{d}_s$ is negligible in terms of computational effort. See [2] for more details. The termination criteria are unchanged.

4.5 Modelling Free Variables

Some of the problems in Section 5 contain unrestricted (free) variables. Our algorithm handles a free variable $x_f \in \mathbb{R}^{n_f}$ by introducing the extra variable t and adding another standard quadratic cone constraint $t \geq \|x_f\|_2$. The entry in c corresponding to t is set to zero. See [3] for a discussion of this approach.

4.6 Solving the Linear Systems

In each iteration of the homogeneous and self-dual interior point method, linear systems of the type (29)-(33) need to be solved. This system can be solved by block-reducing it to obtain the *normal equations*, a system of the form $ADA^Tv = r$ where D is diagonal and strictly positive and v is the unknown, see e.g. [1] for details. The matrix ADA^T is symmetric and positive definite, so we solve the equation using Cholesky factorization.

The matrix ADA^T becomes increasingly ill-conditioned as an optimal point is approached. For this reason, special handling of the factorization is usually employed as the optimal point is approached [26]. In our MATLAB-implemention of CCOPT, we switch from the standard `chol` to `cholinc` if numerical problems are encountered with `chol`. Essentially `cholinc` perturbs small pivots during the Cholesky factorization as is common practice, so the performance penalty is insignificant. This approach often enables us to obtain a higher accuracy of the solution than had we not switched to `cholinc`.

5 Numerical Results

In this section we present a series of computational results that support the effectiveness of our warmstarting strategies. We first describe the general methodology of our testing and then we present results for linear programs and for mixed linear and quadratic conic problems.

5.1 General Methodology

When conducting numerical experiments with CCOPT cold- and warmstarted, we use the following procedure. We first solve $\mathcal{P}$ using CCOPT and store the solution (x^*, y^*, s^*). We then perturb $\mathcal{P}$ to obtain the new problem $\hat{\mathcal{P}}$ — how we perturb depends on the type of problem and is described in each subsection below. We then solve $\hat{\mathcal{P}}$ using CCOPT coldstarted, denoted CCOPT(C) and CCOPT warmstarted using just x^* or (x^*, y^*, s^*) in the computation of the warm point, denoted CCOPT(W_P) and CCOPT(W_PD) respectively. For each warmstart, we use the measure

$$\mathcal{R} = \frac{\#\text{Iterations to solve } \hat{\mathcal{P}} \text{ warmstarted}}{\#\text{Iterations to solve } \hat{\mathcal{P}} \text{ coldstarted}}$$

to quantify the gain from warmstarting. If $\mathcal{R} < 1$ the warmstarted run was more efficient than the coldstarted and vice versa. For an entire set of problems $\mathcal{P}_1, \ldots, \mathcal{P}_K$, we define $\mathcal{G}$, the geometric mean of $\mathcal{R}_1, \ldots, \mathcal{R}_K$, i.e.

$$\mathcal{G} = \sqrt[K]{\mathcal{R}_1 \cdots \mathcal{R}_K}$$

Further, we use the following rules:

1. If the solution status of $\mathcal{P}$ was different from that of $\hat{\mathcal{P}}$, the problem was discarded. By solution status we mean either OPT, INFEAS or ILLP – see Section 4.3.
2. If $\mathcal{P}$ was primal or dual infeasible, the problem was discarded. In this case there is no reason to expect the final iterate of the algorithm to contain any valuable information for the solution of $\hat{\mathcal{P}}$.
3. If $\hat{\mathcal{P}}$ was solved completely by the *presolve procedures* described in [4], the problem was discarded. In this case, the number of main interior point iterations can be considered zero, making comparison meaningless. This happened only rarely for problems from the NETLIB-LP test set. We used MOSEK[2] to carry out this test.

For linear programs, we have tested our warmstarting strategies both when the solution (x^*, y^*, s^*) to $\mathcal{P}$ was generated by a coldstarted run of CCOPT and when it was generated by a simplex method[3], which, unlike the IPM, always returns a vertex (basic) solution. The warmstart strategies $\mathrm{W_P}$ and $\mathrm{W_{PD}}$ performed equally well for both cases. This suggests that the IPM is capable of truly using the *information* contained in the solution of $\mathcal{P}$, regardless of whether the final solution is an interior optimal or vertex solution and that the effectiveness of warmstart is not a result of some special "IPM property" of the specific solution produced by CCOPT.

5.2 The Parameters λ and μ^0

In all the following experiments, except the one presented in Section 5.6, we use $\lambda = 0.99$ and $\mu^0 = 1 - \lambda = 0.01$. There is no theoretically well-justified reason for this choice. It is a heuristic choice motivated by numerical experience. The experiment in Section 5.6 investigates the dependence on the parameter λ while using $\mu^0 = 1 - \lambda$. The experiment shows that particularly the performance of $\mathrm{W_P}$ is somewhat sensitive to the choice of λ. Therefore, it is an interesting topic of future interest to devise an adaptive method to choose the parameters λ and μ^0. In the present work, however, we use the static value of $\lambda = 0.99$ (except in Section 5.6) and always set $\mu^0 = 1 - \lambda$.

5.3 Netlib Linear Programs

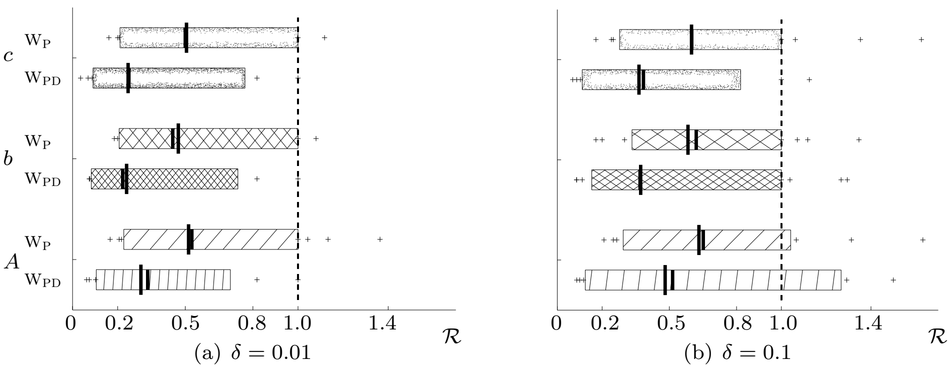
In this section we present results from running our warmstarted algorithm on the linear programs in the NETLIB[4] collection of test problems. We perturb the original problem in a manner similar to the one introduced in [6] and reused in [11]: Let v be a vector we want to perturb randomly (think of either b, c or the vector of nonzeros of A). Assume v has M elements. An element in v is changed if a $[0, 1]$-uniform randomly chosen number is less than $\min\{0.1, 20/M\}$.

---

[2] See `www.mosek.com`.

[3] We used the simplex solver in MOSEK.

[4] `http://www.netlib.org/lp/data/`



(a) $\delta = 0.01$ (b) $\delta = 0.1$

Fig. 1 *Results from the* NETLIB LP *test set with $\lambda = 0.99$ and $\mu^0 = 0.01$. The box contains 90% of the problems, plusses are the remaining 10%. The dashed line is $\mathcal{R} = 1.0$. The largest solid line is the geometric mean and the smaller solid line is the median. The accuracy used was $\epsilon = 10^{-6}$ (cf. Section 4.3). See text for further explanation of this figure.*

Thus on average, we change 10% but at most 20 elements of v. An element v_i is changed by setting

$$v_i := \begin{cases} \delta r & \text{if } |v_i| \leq 10^{-6} \\ (1+\delta r)v_i & \text{otherwise} \end{cases}$$

where r is a number chosen randomly from a uniform distribution on $[-1, 1]$. The scalar δ is a parameter that controls the pertubation magnitude.

We present results for the three cases where v is either b, c or the nonzeros of A. Figure 1 shows the value of $\mathcal{R}$ found for each problem in the test set. This was done for all three types of perturbations and for two values of δ. We observe that at these levels of δ, the gain in warmstarting using either strategy is significant. Overall, we see a reduction in the geometric mean of the number of iterations ranging from 34% to 52% when comparing CCOPT(C) to CCOPT($\mathrm{W_P}$) and 50% to 75% for CCOPT($\mathrm{W_{PD}}$). Usually about one in four problems were discarded because of rules 1–3, Section 5.1. Clearly the gain is smaller for the larger value of δ, compare Figures 1(a) and 1(b). Figure 2 on the next page shows the relation between the magnitude of the perturbation δ and reduction in the geometric mean of number of iterations. As expected, we clearly observe that the reduction depends crucially on δ. The size of the reduction is significant as long as δ is small enough. It is apparent that $\mathrm{W_{PD}}$ is consistently better than $\mathrm{W_P}$. This is of course reasonable since $\mathrm{W_{PD}}$ uses more information from the solution of $\mathcal{P}$ than $\mathrm{W_P}$. Notice, however, that the gap between $\mathrm{W_P}$ and $\mathrm{W_{PD}}$ narrows as δ grows. This too is reasonable, because as the problem is perturbed more, the information from the primal or the dual points can no longer be expected to be good. Thus both behave more and more like a coldstart.

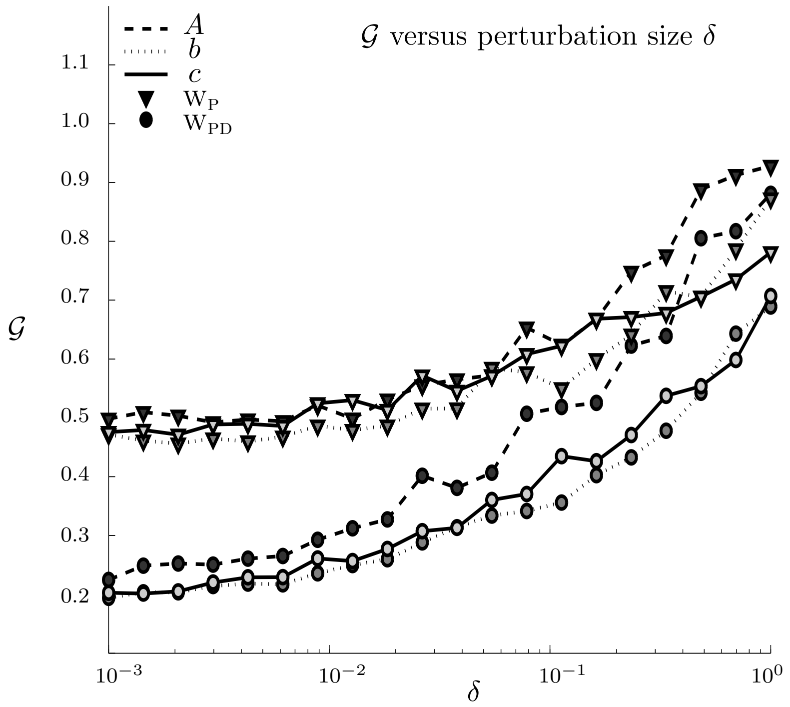


Fig. 2 *Results from the* NETLIB-LP *test set with* $\lambda = 0.99$ *and* $\mu^0 = 0.01$ *and varying* δ*. Each data point in the figure corresponds to solving the entire* NETLIB-LP *test set with the problem-perturbation specified in the legend for a certain value of* δ*. All problems were solved to the accuracy* $\epsilon = 10^{-6}$ *(cf. Section 4.3). See text for further explanation of this figure.*

5.4 Efficient Frontier Computation

An obvious candidate problem on which a warmstarting strategy should be employed is that of computing the efficient frontier in the Markowitz portfolio selection setting. The presentation here follows that of [5].

Assume that $r \in \mathbb{R}^n$ is a multivariate random variable modelling the return of n different assets. Assume further that the mean vector μ_r and covariance matrix Σ_r are known. If our initial holding in asset j is w_j^0 and we invest x_j, the portfolio after the investment period is $w^0 + x$ and thus the expected return of the investment is $r^T(w^0 + x)$. The risk of the investment is defined as the variance of the return of the investment, namely $(w^0 + x)^T \Sigma_r (w^0 + x) = \|R(w^0 + x)\|_2^2$ where R is a factor in the QR-factorization of $\Sigma = QR$. In the classical Markowitz portfolio selection problem, one seeks to minimize risk while fixing a certain return t. That is, we solve

$$\begin{array}{rll} \min_x & \|R(w^0+x)\|_2 & \\ \text{s.t.} & \bar{r}^T(w^0+x) & = t \\ & e^T x & = 0 \\ & w^0 + x & \geq 0 \end{array} \tag{40}$$

Here, $\bar{r}$ denotes the mean of observed historical realizations of r and R is the triangular factor from the QR-factorization of $\bar{X} = (N-1)^{-1/2}(X - e\bar{r}^T)$ where $X \in \mathbb{R}^{N \times n}$ contains the returns of each asset over time. Notice that $\bar{X}$ is a scaled zero-mean version of the observed data in X. We do not allow short-selling, so we also impose the constraint $w^0 + x \geq 0$. The problem (40) can be reformulated in conic form as

$$\begin{array}{rl} \min_{z,f,g} & f \\ \text{s.t.}\ \bar{r}^T z & = t \\ Rz & = g \\ e^T z & = e^T w^0 \\ f & \geq \|g\|_2 \\ z & \geq 0 \end{array} \tag{41}$$

and it is this version that we are solving using CCOPT. The solution x is then obtained via $z = x + w^0$. Let $f(t)$ denote the optimal value of (41) for a requested return of t. The set of points $(t, f(t))$ for $t \in [0, \max(\bar{r})]$ is called the *efficient frontier*. To compute this curve, we must solve a sequence of problems of the type (41) where only t varies from problem to problem – thus this entire computation is very well suited for a warmstarting scheme: Compute the optimal solution of (41) for the first value of t and compute a warm starting point using this solution as described in Section 3.1. Then solve (41) for the next value of t, initializing the algorithm in the warm starting point. We can then repeat this process for all following values of t using the solution of (41) for the previous value of t to compute a warm starting point for the next problem.

We use as the data matrix X the historically observed data from N daily prices for the 500 stocks in the S&P500 stock index[5]. With $N = 800$, (41) is a problem of the type (1) with $A \in \mathbb{R}^{502 \times 1002}$ and $\text{nnz}(A) = 126750$. The results are shown in Table 1. We see that the work is reduced by about 25% when using W_P and by about 60% if we use W_PD.

5.5 Frequent Robust Portfolio Rebalancing

The Markowitz portfolio selection problem presented in the previous section can be further generalized by assuming that the data X are uncertain but belong to known uncertainty sets. The *robust* portfolio selection problem consists in choosing the best possible portfolio while assuming that the worst case

[5] See e.g. `http://www.standardandpoors.com/indices/main/en/us`

CCOPT(C)			CCOPT(W_P)		CCOPT(W_{PD})	
t	$f(t)$	iters	iters	$\mathcal{R}$	iters	$\mathcal{R}$
1.00000	0.0042	14	14	1.00	14	1.00
1.00013	0.0037	16	16	1.00	8	0.50
1.00027	0.0038	14	13	0.93	8	0.57
1.00040	0.0042	14	12	0.86	7	0.50
1.00053	0.0050	16	14	0.88	6	0.38
1.00067	0.0058	15	13	0.87	6	0.40
1.00080	0.0068	14	14	1.00	7	0.50
1.00093	0.0078	14	12	0.86	7	0.50
1.00107	0.0089	14	12	0.86	6	0.43
1.00120	0.0101	19	11	0.58	6	0.32
1.00133	0.0114	16	12	0.75	6	0.38
1.00147	0.0127	14	11	0.79	5	0.36
1.00160	0.0141	14	10	0.71	6	0.43
1.00173	0.0158	19	9	0.47	6	0.32
1.00187	0.0177	15	10	0.67	5	0.33
1.00200	0.0197	14	9	0.64	5	0.36
1.00213	0.0219	14	10	0.71	5	0.36
1.00227	0.0242	14	8	0.57	5	0.36
1.00240	0.0265	13	10	0.77	4	0.31
1.00253	0.0289	14	9	0.64	4	0.29
1.00267	0.0313	11	9	0.82	4	0.36
1.00280	0.0338	12	10	0.83	5	0.42
1.00293	0.0363	12	8	0.67	4	0.33
1.00307	0.0388	12	8	0.67	5	0.42
1.00320	0.0414	12	8	0.67	5	0.42
	$\mathcal{G}$	14.1	10.7	**0.76**	5.7	**0.41**

Table 1 *Results from solving a series of Markowitz portfolio optimization problems, combined comprising an efficient frontier. We used $\lambda = 0.99$ and $\mu^0 = 0.01$. The third column shows the number of iterations spent solving the problem using* CCOPT *from a coldstart. The two column blocks to the right show the performance of* W_P *and* W_{PD}. *All problems were solved to the accuracy $\epsilon = 10^{-6}$ (cf. Section 4.3).*

scenario within the uncertainty sets is realized. The optimal such portfolio is the solution of a second order cone program (SOCP). For a complete description of the model, see [9] — here we omit a detailed description of the model as it is not the primary interest of this paper.

Instead, we focus on the following situation. On a certain trading day, we can estimate the return and variance of each asset and their uncertainty sets from historical data, for example from the past H trading days. This is done as in Section 5.4 (see [9] for estimation of the uncertainty sets). We can then compute a robust portfolio by solving the corresponding SOCP. A number of trading days later (say, k days), we repeat this procedure, estimating the relevant parameters over an equally long backwards time horizon, which is now shifted by k days. If k is small compared to H, the new estimates of the parameters are likely to be only slightly different from the previous ones. Therefore we can compute a warm starting point using the solution of the previous problem. This procedure can then be repeated.

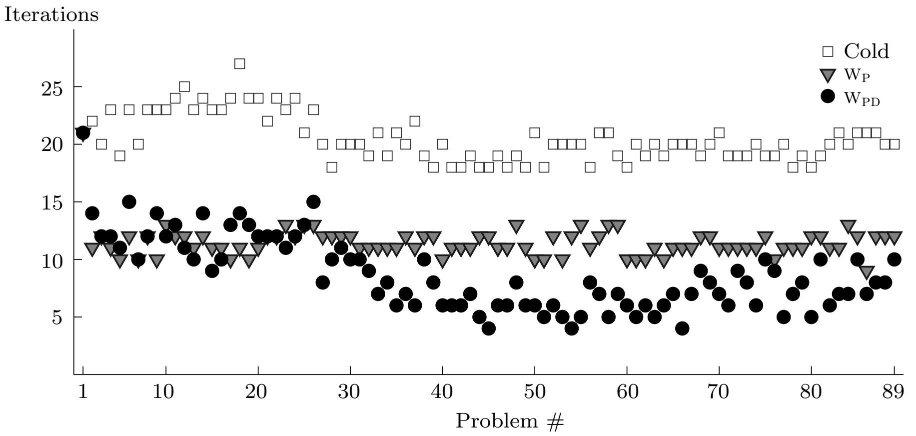


Fig. 3 *Results from the portfolio rebalancing problems. The problem set contains* 89 *problems. The figure shows the number of iterations spent solving each problem from a cold start (squares), the point* w_P *(triangles) and the point* w_{PD} *(circles) computed using the solution of the previous problem. We used* $\lambda = 0.99$ *and* $\mu^0 = 0.01$ *for all problems.*

To facilitate future research in the field of warmstarting optimization algorithms for SOCPs, we have generated a sequence of such problems using data from 2761 consecutive trading days from the stocks in the S&P500 stock index. Starting on day number 1001, we estimated returns and uncertainty sets over the past $H = 1000$ days and repeated this procedure for every $k = 20$ trading days. The result is a problem set consisting of 89 neighboring SOCPs each with 2531 variables and 1527 linear constraints, of which only two do not come from the introduction of slack variables. Of the 2531 variables, 2005 are non-negative and the rest belong to quadratic cones of dimensions 3, 21 and 502. The problems are stored in SeDuMi format (see [22]) in the MATLAB binary `.mat`-format and they are publicly available[6].

Figure 3 shows the performance of w_P and w_{PD} on this set of problems. We see that each problems is usually solved in about 20 iterations by CCOPT when started from a coldstart. Using warmstart from w_P reduces the number of iterations to about 10–13. Warmstarting from w_{PD} reduces the number even further to the range 4–15 iterations. The quantity $\mathcal{G}$ (defined in Section 5.1) for w_P and w_{PD} was 0.5590 and 0.3985 respectively. We can conclude that for these problems, our warmstarting strategies are highly effective.

5.6 Minimal Norm Vector in Convex Hull

In certain algorithms called *bundle methods* employed particularly in the field of nonsmooth optimization, a series of vectors (gradients at the iterates) are stored (in a *bundle*) and used in computing the next search direction and

[6] `http://www2.imm.dtu.dk/~andsk/files/warmstart/robpfrebalancing_probs.html`.

sometimes used to check stopping criteria. If the current bundle contains $g_1, \dots, g_k \in \mathbb{R}^n$, usually we will have $k \ll n$. At every iteration of these algorithms, the vector with minimal norm in the convex hull of the vectors $g_1, \dots, g_k$ is needed. At the end of each iteration, the bundle is updated, for example by removing one vector and replacing it by another one. We thus get a sequence of related optimization problems to solve – hence another suitable candidate for a warmstarting strategy.

Let $G \in \mathbb{R}^{n \times k}$ be a matrix with $g_1, \dots, g_k$ in the columns. The problem of finding the minimal norm vector in the convex hull of $g_1, \dots, g_k$ can be formulated as

$$\begin{cases} \min_x & \|Gx\|_2 \\ \text{s.t.} & e^T x = 1 \\ & x \geq 0 \end{cases} \quad \text{or} \quad \begin{cases} \min_{(x,t,y)} & t \\ \text{s.t.} & Gx = y \\ & e^T x = 1 \\ & x \geq 0 \\ & t \geq \|y\|_2 \end{cases} \tag{42}$$

The formulation on the right is in the standard conic form (1). If x^* solves this problem then Gx^* is the vector we seek. Using the notation of (1), we see that modifying G corresponds to changing the constraint matrix A of the problem. We experiment numerically with this problem by first generating $G \in \mathbb{R}^{n \times k}$ randomly from a $[-1, 1]$-uniform distribution and then solving the problem coldstarted – the solution is used in computing the warm points for the modified problem. We then change one entire column of G to a vector in $\mathbb{R}^n$ randomly chosen from the $[-1, 1]$-uniform distribution. The new problem is then solved both cold- and warmstarted for 20 equidistantly distributed $\lambda \in [0, 0.99]$. All this is done for 10 random instances, for $n = 80$ and two values of k. The results (geometric means over the 10 random instances) are shown in Figure 4. We clearly see, particularly for $\mathrm{W_P}$, that the best value of λ depends on the problem (in this case on k). Again $\mathrm{W_{PD}}$ consistently performs

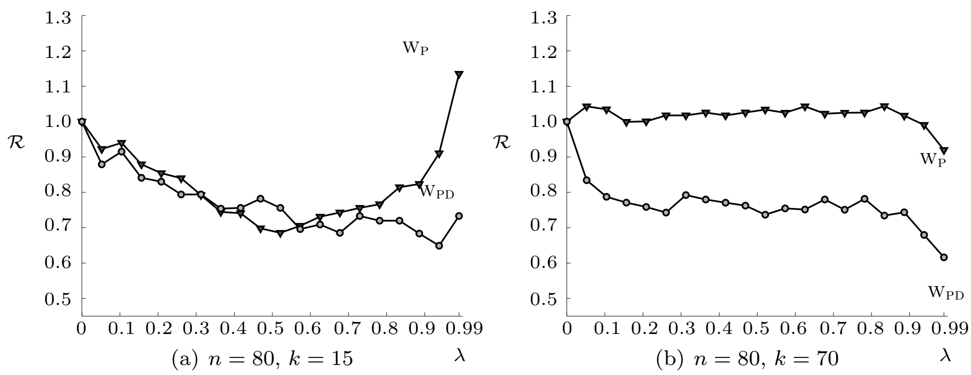


Fig. 4 *Results from solving* (42). *Geometric means of* $\mathcal{R}$ *over 10 random instances are shown. We used the tolerance* $\epsilon = 10^{-6}$ *(cf. Section 4.3) and always used* $\mu^0 = 1 - \lambda$. *Triangles denote* $\mathrm{W_P}$, *circles denote* $\mathrm{W_{PD}}$.

better than W_P, producing improvements in the range 20%–40% depending on problem and λ.

6 Conclusion and Future Work

In this paper, we have presented two new warmstarting strategies particularly well suited for homogeneous interior point methods to solve convex conic optimization problems involving linear and quadratic cones. We have analyzed them and given conditions under which each of them results in improved performance over a standard coldstart. In contrast to several previous warmstarting strategies, one of our strategies uses only the primal optimal point of the previous problem to solve the next. The other strategy uses only the primal and dual optimal solution but no intermediate iterates. This is significant in that it allows users of black-box optimization algorithms to apply our warmstarting strategy as part of a larger program where a series of related optimization problems are subproblems that need to be solved. A further benefit of our stategies is that they cost virtually nothing to compute.

We have presented extensive computational experiments with our warmstarting strategies showing work reductions in the range of 30%–75%. Thus the strategies are very effective in practice. This was shown both for linear programming problems and quadratic programming problems, which we formulated as general mixed linear and quadratic cone problems.

Our results apply to an interior point method used to solve the homogeneous model. It is an interesting question whether the presented warmstarting strategies would work equally well when used in a primal-dual interior point method applied to solve the original primal-dual pair of conic programs.

Using the general convex conic format, we expect to be able to easily generalize our warmstarting strategies to the context of semidefinite programming. This step simply involves the already known generalization of the Jordan product $\circ$ to the cone of symmetric and semidefinite matrices, similar to what was done in Section 4.4 for the quadratic cones. For that reason, we expect our strategies to also be useful in algorithms for solving combinatorial optimization problems. Here, problems are often reduced to solving a series of related simpler continuous problems such as linear programs, quadratic programs or semidefinite programs. Thus warmstarting is an obvious idea to improve computational performance. In this situation, the number of variables in $\mathcal{P}$ and $\hat{\mathcal{P}}$ may be different. In case it increases, we can add in components from the standard cold starting point C in appropriate places. If the number of variables on the other hand decreases, we simply drop those variables from the warm starting point.

References

1. Andersen, E.D., Andersen, K.D.: *The MOSEK interior point optimization for linear programming: an implementation of the homogeneous algorithm.* In: Frenk, H., Roos,

K., Terlaky, T., Zhang, S.: *High Performance Optimization*, 197–232. Kluwer (1999)
2. Andersen, E.D., Roos, C., Terlaky, T.: *On implementing a primal-dual interior-point method for conic quadratic optimization.* Math. Program. 95(2), 249–277 (2003).
3. Andersen, E.D.: *Handling free variables in primal-dual interior-point methods using a quadratic cone.* Available from `http://www.mendeley.com/c/4812865462/p/11467401/andersen-2002-handling-free-variables-in-methods-using-a-quadratic-cone/` (2002)
4. Andersen, E.D., Andersen, K.D.: *Presolving in linear programming.* Math. Program. 71(2), 221–245 (1995).
5. Andersen, E.D., Dahl, J., Friberg, H.A.: *Markowitz portfolio optimization using MOSEK.* MOSEK technical report: TR-2009-2 (2011)
6. Benson, H.Y., Shanno, D.F.: *An exact primal-dual penalty method approach to warm-starting interior-point methods for linear programming.* Comput. Optim. Appl. 38, 371–399 (2007).
7. Colombo, M., Gondzio, J., Grothey, A.: *A warm-start approach for large-scale stochastic linear programs.* Math. Program. 127(2), 371–397 (2011).
8. Engau, A., Anjos, M.F., Vannelli, A.: *On Interior-Point warmstarts for linear and combinatorial optimization.* SIAM J. Optim. 20(4), 1828–1861 (2010).
9. Goldfarb, D., Iyengar, G.: *Robust portfolio selection problems.* Math. Oper. Res. 28(1), 1–38 (2003).
10. Gondzio, J., Grothey, A.: *Reoptimization with the Primal-Dual interior point method.* SIAM J. Optim. 13, 842–864 (2002).
11. Gondzio, J., Grothey, A.: *A new unblocking technique to warmstart interior point methods based on sensitivity analysis.* SIAM J. Optim. 19(3), 1184–1210 (2008).
12. John, E., Yildirim, E.A.: *Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension.* Comput. Optim. Appl. 41, 151–183 (2008).
13. Luo, Z.Q., Sturm, J.F., Zhang, S.: *Conic convex programming and self-dual embedding.* Optim. Methods Softw. 14(3), 169–218 (2000).
14. Lustig, I.J., Marsten, R.E., Shanno, D.F.: *Interior point methods for linear programming: Computational state of the art.* ORSA J. Comput. 6(1), 1–14 (1994).
15. Megiddo, N.: *On finding primal- and dual-optimal bases.* ORSA J. Comput. 3(1), 63–65 (1991).
16. Mehrotra, S.: *On the implementation of a primal-dual interior point method.* SIAM J. Optim. 2(4), 575–601 (1992).
17. Mizuno, S., Todd, M.J., Ye, Y.: *On adaptive-step primal-dual interior-point algorithms for linear programming.* Math. Oper. Res. 18(4), 964–981 (1993).
18. Nesterov, Y.E., Nemirovski, A.S.: *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM (1994)
19. Nesterov, Y.E., Todd, M.J.: *Self-scaled barriers and interior-point methods for convex programming.* Math. Oper. Res. 22(1), 1–42 (1997).
20. Nesterov, Y.E., Todd, M.J.: *Primal-dual interior-point methods for self-scaled cones.* SIAM J. Optim. 8(2), 324–364 (1998).
21. Polyak, R.: *Modified barrier functions (theory and methods).* Math. Program. 54, 177–222 (1992).
22. Sturm, J.F.: *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones.* Optim. Methods Softw. 12, 625–653 (1999).
23. Sturm, J.F.: *Implementation of interior point methods for mixed semidefinite and second order cone optimization problems.* Optim. Methods Softw. 17(6), 1105–1154 (2002).
24. Tuncel, L.: *Primal-dual symmetry and scale invariance of interior-point algorithms for convex optimization.* Math. Oper. Res. 23(3), 708–718 (1998).
25. Wright, S.J.: *Primal-Dual Interior-Point Methods.* SIAM (1987)
26. Wright, S.J.: *Modified cholesky factorizations in Interior-Point algorithms for linear programming.* SIAM J. Optim. 9(4), 1159–1191 (1999).
27. Xu, X., Hung, P.F., Ye, Y.: *A simplified homogeneous and self-dual linear programming algorithm and its implementation.* Ann. Oper. Res. 62(1), 151–171 (1996).
28. Ye, Y., Todd, M.J., Mizuno, S.: *An O(sqrt(n)L)-iteration homogeneous and self-dual linear programming algorithm.* Math. Oper. Res. 19(1), 53–67 (1994).

29. Ye, Y.: *Interior Point Algorithms: Theory and Analysis.* Wiley-Interscience (1997)
30. Yildirim, E.A., Wright, S.J.: *Warm-start strategies in interior-point methods for linear programming.* SIAM J. Optim. 12(3), 782–810 (2002).