A computational study of the homogeneous algorithm for large-scale convex optimization

Erling D. Andersen *

Yinyu Ye †

May 1996 (Revised marts 1997)

Abstract

Recently the authors have proposed a homogeneous and self-dual algorithm for solving the monotone complementarity problem (MCP) [5]. The algorithm is a single phase interior-point type method, nevertheless it yields either an approximate optimal solution or detects a possible infeasibility of the problem. In this paper we specialize the algorithm to the solution of general smooth convex optimization problems that also possess nonlinear inequality constraints and free variables. We discuss an implementation of the algorithm for large-scale sparse convex optimization. Moreover, we present computational results for solving quadratically constrained quadratic programming and geometric programming problems, where some of the problems contain more than 100,000 constraints and variables. The results indicate that the proposed algorithm is also practically efficient.

*Department of Management, Odense University, Campusvej 55, DK-5230 Odense M, Denmark. E-mail: eda@busieco.ou.dk

†Department of Management Sciences, The University of Iowa, Iowa City, Iowa 52242, USA. E-mail: yyye@dollar.biz.uiowa.edu. This author is supported in part by NSF Grants DDM-9207347 and DMI-9522507.

Key words: Monotone complementarity problem, homogeneous and self-dual model, interior-point algorithms, large-scale convex optimization.

1 Introduction

In 1984 Karmarkar [31] presented an interior-point method for linear programming (LP) and since then interior-point algorithms enjoyed great publicity for two reasons. First, these algorithms solve LP problems in polynomial time, as proved by Karmarkar and many others. Secondly, interior-point algorithms have demonstrated excellent practical performance when solving large-scale LP problems, see Lustig et al. [37].

It was soon realized (see Gill et al. [25]) that Karmarkar's method was closely related to the logarithmic barrier algorithm for general nonlinear programming studied by Fiacco and McCormick [23] and others in the sixties. Hence, it is natural to investigate the efficiency of the interior-point methods for solving more general classes of problems. In general good complexity results could only be expected for solving convex optimization problems.

Interior-point methods for general convex optimization have been developed by many researchers, and probably the most important theory has been presented by Nesterov and Nemirovskii [40]. They have shown that if the functions satisfy a certain smoothness condition, then the logarithmic barrier algorithm solves convex optimization problems in polynomial time. Hence, interior-point methods could be efficient for the solution of certain convex optimization problems.

Since the most successful interior-point method for LP is the Kojima et al. [33] primal-dual type algorithm, several researchers have investigated the theoretical and practical behavior of the primal-dual interior-point methods for solving general or specific classes of convex optimization problems. Kortanek et al. [34] analyzed a primal-dual type method for linearly constrained convex programming with a feasible starting point. Vial [46] presented a similar primal-dual algorithm for general smooth convex optimization and obtained numerical results, which demonstrated that the algorithm could solve several classes of problems efficiently. Later, Vial and Anstreicher [8] presented a globally convergent version of Vial's algorithm. El-Bakry et al. [22] proposed an algorithm of the similar type for general smooth, possibly nonconvex, optimization. They also gave a global convergence proof and some promising numerical results. For the same class of problems Yamashita and Yabe [49] study the local convergence properties of the primal-dual algorithm.

Several researchers developed and implemented interior-point algorithms for solving special classes of convex optimization problems. Vanderbei [45] developed a primal-dual algorithm to solve convex quadratic programs with linear constraints. Kortanek et al. [36] developed a primal-dual infeasible-start algorithm for solving the linearly constrained convex problems and applied the algorithm to solve dual geometric programming problems. Andersen [7] and Xue and Ye [48] developed efficient interior-point algorithms for minimizing a sum of Euclidean norms, which is a minimization problem over the second-order cone. Jarre et al. [30] developed an excellent implementation for solving quadratically constrained minimization problems arising in engineering designs. Also the important class of positive semi-definite programming problems has received great attention. We refer the reader to the paper by Alizadeh [1], Nesterov and Todd [41], and Vandenberghe and Boyd [43].

The primal-dual algorithm is based on directly solving the Karush-Kuhn-Tucker (KKT) equations and requires second order information of the (objective and constraint) functions. An alternative approach is to work with a linear approximation to the convex optimization problem and successively refine the approximation by adding linear cuts. This so-called cutting plane idea has been explored computationally by several researchers, see [9, 18] and the references therein. Recently Gondzio et al. [27, 26] report good computational results of this approach for solution of some large-scale nonlinear multicommodity network flows.

More recently, Andersen and Ye [5] proposed a "homogeneous and self-dual" algorithm for solving the monotone complementarity problem. Observe that, by using the KKT conditions, a convex optimization problem can be formulated as a monotone complementarity problem. The proposed algorithm was applied to solve the linearly constrained convex optimization problem and some promising numerical results were presented. The main advantage of the algorithm is that it can detect whether the problem is feasible or not without using any regularization or "big-M" construction. Moreover, if the monotone complementarity problem is polynomially solvable from a given interior feasible starting point, then it can also be solved polynomially without knowing and using such a starting point.

A homogeneous model and algorithm was first suggested by Ye et al. [50] for the case of LP. This model was later simplified by Xu et al. [47]. A homogeneous model and algorithm for the special case of semi-definite programming is proposed in [42, 16].

The purpose of this paper is to specialize the homogeneous algorithm to solution of large-scale convex optimization problems that also possess nonlinear inequality constraints and free variables. Moreover, we investigate the efficiency of the proposed algorithm for solving large-scale and sparse convex optimization problems.

The outline of the paper is as follows. First we present some relevant theories and notations. Next we state the homogeneous and self-dual model for the monotone complementarity problem and its basic properties. Then we develop an interior-point algorithm to solve this model. This is followed by a specialization of the algorithm to general smooth convex optimization. Finally, we present extensive numerical results for solving several large-scale test problems, which include quadratically constrained quadratic programming and geometric programming problems.

2 Notation

Consider the monotone complementarity problem (MCP) with nonlinear equality constraints in the following form:

$$
\begin{array}{lll}
(MCP) & \text{minimize} & x^T s \\
& \text{subject to} & \begin{pmatrix} 0 \\ s \end{pmatrix} = f(y,x) := \begin{pmatrix} h(y,x) \\ g(y,x) \end{pmatrix}, \ (x,s) \geq 0,
\end{array}
$$

where m and n is the dimension of y and x respectively. Similarly $s \in \mathbf{R}^n$ and is a vector of slack variables. $f(y,x)$ is assumed to be a continuous *monotone* mapping from $\mathbf{R}^m \times \mathbf{R}^n_{++}$ to $\mathbf{R}^{m+n}$, where we use the definitions $\mathbf{R}_+ := \{x : \ x \geq 0\}$ and $\mathbf{R}_{++} := \{x : \ x > 0\}$, and ^{T} denotes transpose. Finally, we have $h : \ \mathbf{R}^{m+n} \to \mathbf{R}^m$ and $g : \ \mathbf{R}^{m+n} \to \mathbf{R}^n$.

In other words, for every $(y^1, x^1), (y^2, x^2) \in \mathbf{R}^m \times \mathbf{R}^n_+$, we have (e.g., [15])

$$
((y^1; x^1) - (y^2; x^2))^T (f(y^1, x^1) - f(y^2, x^2)) =
$$

$$
(y^1 - y^2)^T (h(y^1, x^1) - h(y^2, x^2)) + (x^1 - x^2)^T (g(y^1, x^1) - g(y^2, x^2)) \geq 0.
$$

Denote by ∇f the Jacobian matrix of f, which is positive semi-definite in $\mathbf{R}^m \times \mathbf{R}^n_{++}$.

Here and in the following we use the Matlab-inspired notation $(x; y)$ to denote a vector that is the concatenation of the vectors x and y. A similar notation is used for matrices. Also, if $x \in \mathbf{R}^n$, then $\min(x) := \min_j(x_j)$.

(MCP) is said to be (asymptotically) feasible if and only if there is a *bounded* sequence $(y^t, x^t, s^t) \in \mathbf{R}^m \times \mathbf{R}^n_+ \times \mathbf{R}^n_+$, $t = 1, 2, ...$, such that

$$
\lim_{t \to \infty} \quad (0; s^t) - f(y^t, x^t) \to 0,
$$

where any limit point $(\hat{y}, \hat{x}, \hat{s})$ of the sequence is called an (asymptotically) feasible point for (MCP). (MCP) has an interior feasible point if it has an (asymptotically) feasible point $(\hat{y}, \hat{x}, \hat{s}) \in \mathbf{R}^m \times \mathbf{R}^n_{++} \times \mathbf{R}^n_{++}$. (MCP) is said to be (asymptotically) solvable if there is an (asymptotically) feasible $(\hat{y}, \hat{x}, \hat{s})$ such that $\hat{x}^T \hat{s} = 0$, where $(\hat{y}, \hat{x}, \hat{s})$ is called the "optimal" or "complementary" solution for (MCP).

Note even if (MCP) is feasible, it does not imply that (MCP) has a solution. If (MCP) has a solution, then it has a maximal solution (y^*, x^*, s^*) where the number of positive components in (x^*, s^*) is maximal. The indices of those positive components are invariant among all maximal solutions for (MCP) (see Güler [28]).

3 A homogeneous and self-dual MCP model

In [5, 6] the authors present a homogeneous and self-dual model for (MCP). For convenience of the reader we state here this model and several relevant results. The homogeneous model to (MCP) is:

$$
\begin{array}{lll}
(HMCP) & \text{minimize} & x^T s + \tau\kappa \\
& & \\
& \text{subject to} & \begin{pmatrix} 0 \\ s \\ \kappa \end{pmatrix} = \begin{pmatrix} \tau h(y/\tau, x/\tau) \\ \tau g(y/\tau, x/\tau) \\ -y^T h(y/\tau, x/\tau) - x^T g(y/\tau, x/\tau) \end{pmatrix}, \\
& & (x, \tau, s, \kappa) \geq 0.
\end{array} \tag{1}
$$

Let $\psi(y, x, \tau) : \ \mathbf{R}^m \times \mathbf{R}^{n+1}_{++} \to \mathbf{R}^m \times \mathbf{R}^{n+1}$ be defined by

$$
\psi(y, x, \tau) := \begin{pmatrix} \tau f(y/\tau, x/\tau) \\ -(y; x)^T f(y/\tau, x/\tau) \end{pmatrix} = \begin{pmatrix} \tau h(y/\tau, x/\tau) \\ \tau g(y/\tau, x/\tau) \\ -y^T h(y/\tau, x/\tau) - x^T g(y/\tau, x/\tau) \end{pmatrix}. \tag{2}
$$

We have that

$$
\nabla\psi(y, x, \tau) :=
$$

$$
\begin{pmatrix} \nabla f(y/\tau, x/\tau) & f(y/\tau, x/\tau) - \nabla f(y/\tau, x/\tau)\frac{(y;x)}{\tau} \\ -f(y/\tau, x/\tau)^T - \frac{(y;x)}{\tau}^T \nabla f(y/\tau, x/\tau) & (y/\tau; x/\tau)^T \nabla f(y/\tau, x/\tau)\frac{(y;x)}{\tau} \end{pmatrix}. \tag{3}
$$

Then, it is easy to verify the following lemma.

Lemma 3.1 *Let ψ be defined by (2) then*

i. $(y; x; \tau)^T \psi(y, x, \tau) = 0$.

ii. $\nabla\psi(y, x, \tau)(y; x; \tau) = \psi(y, x, \tau)$.

iii. $(y; x; \tau)^T \nabla\psi(y, x, \tau) = -\psi(y, x, \tau)^T$.

iv. *Let ∇f be positive semi-definite in $\mathbf{R}^m \times \mathbf{R}^n_+$. Then $\nabla\psi$ is positive semi-definite in $\mathbf{R}^m \times \mathbf{R}^{n+1}_{++}$.*

Furthermore, we have the following theorem.

Theorem 3.1 *Let ψ be given by (2) and let $(y^*, x^*, \tau^*, s^*, \kappa^*)$ be a maximal complementary solution for $(HMCP)$. Then (MCP) has a solution if and only if $\tau^* > 0$. In this case, $(y^*/\tau^*, x^*/\tau^*, s^*/\tau^*)$ is a complementary solution for (MCP).*

Therefore instead of solving the original problem we can compute a maximal solution to the homogeneous model $(HMCP)$. Such a maximal solution will provide the information needed about the original problem.

We now present an interior-point algorithm for solution of $(HMCP)$. For simplicity, in what follows we let

$$x := (x; \tau) \in \mathbf{R}^{n+1},$$

and

$$s := (s; \kappa) \in \mathbf{R}^{n+1}.$$

4 The homogeneous interior-point algorithm

At iteration k with iterate $(y, x, s) \in \mathbf{R}^m \times \mathbf{R}^{n+1}_{++} \times \mathbf{R}^{n+1}_{++}$, the algorithm solves for direction (d_y, d_x, d_s) from the following system of linear equations

$$(0; d_s) - \nabla\psi(y, x)(d_y; d_x) \quad = \quad \eta r \tag{4}$$

and

$$Xd_s + Sd_x \;=\; \gamma\mu e - Xs, \tag{5}$$

where η and γ are proper given parameters between 0 and 1, and

$$r := (r_h; r_g) := -(0; s) + \psi(y, x) \quad \text{and} \quad \mu := \frac{x^T s}{n+1}.$$

In the following lemma we prove that the Newton equation system (4) and (5) always has a solution.

Lemma 4.1 *The set of linear equations (4) and (5) always has a solution.*

Proof: Let $(d_y; d_x; d_s) = -\eta(y; x; s) + (d'_y; d'_x; d'_s)$. Using the fact that $\nabla\psi(y, x)(y; x) = \psi(y, x)$ we obtain (4) and (5) are equivalent to

$$(0; d'_s) - \nabla\psi(y, x)(d'_y; d'_x) \;=\; 0 \tag{6}$$

and

$$Xd'_s + Sd'_x \;=\; (2\eta - 1)Xs + \gamma\mu e. \tag{7}$$

By eliminating d'_s from (6) and (7) we obtain

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} d'_y \\ d'_x \end{bmatrix} = \begin{bmatrix} 0 \\ X^{-1}((2\eta - 1)Xs + \gamma\mu e) \end{bmatrix}, \tag{8}$$

where

$$M := \begin{bmatrix} A & B \\ C & D \end{bmatrix} := \nabla\psi(y, x) + \begin{bmatrix} 0 & 0 \\ 0 & X^{-1}S \end{bmatrix}. \tag{9}$$

Clearly M is positive semi-definite and D is positive definite.

First, suppose that the submatrix $[A\ B]$ is not of full row rank. Then a non-singular matrix $T \in \mathbf{R}^{m\times m}$ exists such that

$$e_i^T T[A\ B] = 0,$$

where e_i is the ith unit vector. Let $d''_y = T^{-1}d'_y$. Then (8) is equivalent to

$$\begin{bmatrix} TAT & TB \\ CT & D \end{bmatrix} \begin{bmatrix} d''_y \\ d'_x \end{bmatrix} = \begin{bmatrix} 0 \\ X^{-1}((2\eta - 1)Xs + \gamma\mu e) \end{bmatrix}. \tag{10}$$

Now the matrix

$$\begin{bmatrix} TAT & TB \\ CT & D \end{bmatrix} \tag{11}$$

is also positive semi-definite. Therefore, for any $j \in \{1, \dots, m\}$ and for all $\alpha, \beta \in$ $\mathbf{R}$ we have

$$(\alpha e_i + \beta e_j)^T TAT(\alpha e_i + \beta e_j) = \alpha\beta e_j^T TATe_i + \beta^2 e_j^T TATe_j \geq 0.$$

This implies $(TAT)_{ji} = 0$ for all $j = 1, \dots, m$. Similarly, for any $j \in \{1, \dots, n\}$ and for all $\alpha, \beta \in \mathbf{R}$ we have

$$(\alpha e_i; \beta e_j)^T \begin{bmatrix} TAT & TB \\ CT & D \end{bmatrix} (\alpha e_i; \beta e_j) = \alpha\beta e_j^T CTe_i + \beta^2 e_j^T De_j \geq 0,$$

which implies $(CT)_{ji} = 0$ for all $j = 1, \dots, n$. Hence, all elements in the ith row and column of the matrix (11) are identically zero. Note that the ith element in the right hand side of (10) is zero. Therefore, we can remove the ith row and column of the coefficient matrix in (10) by setting $e_i^T d_y'' = 0$, and solve a lower dimensional system of linear equations. We will therefore without loss of generality assume $[A\ B]$ is of full row rank.

Now assume that $[A\ B]$ has full row rank and let $(\bar{x}; \bar{y})$ be a solution to the homogeneous system

$$M^T(\bar{y}; \bar{x}) = \begin{bmatrix} A^T & C^T \\ B^T & D^T \end{bmatrix} \begin{bmatrix} \bar{y} \\ \bar{x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{12}$$

Multiplying both sides of (12) by $(\bar{y}; \bar{x})$ gives

$$(\bar{y}; \bar{x})^T \nabla\psi(y, x)(\bar{y}; \bar{x}) + \bar{x}^T X^{-1} S\bar{x} = 0,$$

because of (9). From the fact $\nabla\psi(y, x)$ is positive semi-definite it follows that $\bar{x} = 0$. This implies that $\bar{y} = 0$, because $[A\ B]^T$ is of full column rank. Therefore, (12) has the unique solution 0 which implies M is of full row rank and then the reduced Newton equation system (8) has a solution.

□

After the search direction is computed, the variables are updated. A simple update is the linear update

$$(y^+; x^+; s^+) = (y; x; s) + \alpha(d_y; d_x; d_s), \tag{13}$$

where $\alpha \geq 0$ is a given step-size. Next, we discuss how to choose the step-size α using this update. First, define the merit function

$$\phi(y, x, s) := \theta x^T s + \|(0; s) - \psi(y, x)\|, \tag{14}$$

where θ is a positive parameter. Clearly if the merit function is reduced to zero (while $\tau > 0$), then a complementary solution is obtained. Therefore, the merit function will be used to measure the progress in the algorithm. The purpose of θ in (14) is to balance the complementarity ($x^T s$) and infeasibility measures ($\|(0; s) - \psi(y, x)\|$).

It can be verified that

$$\nabla\phi(y, x, s)^T = \theta(0; s; x) + \frac{(-\nabla\psi(y, x)^T; I)((0; s) - \psi(y, x))}{\|(0; s) - \psi(y, x)\|}.$$

Hence,

$$\begin{aligned} \nabla\phi(y, x, s)(d_y; d_x; d_s) &= \theta(s^T d_x + x^T d_s) \\ &\quad + \tfrac{((0;s)-\psi(y,x))^T(-\nabla\psi(y,x)^T;I)^T(d_y;d_x;d_s)}{\|(0;s)-\psi(y,x)\|} \\ &= \theta(\gamma - 1)x^T s - \eta\|(0; s) - \psi(y, x)\|, \end{aligned} \tag{15}$$

which is obtained using (4) and (5). Therefore, for a suitable choice of γ and η the search direction is a descent direction for the merit function.

In each iteration the step-size is selected such that all iterates satisfies the following three conditions. Let $\beta \in R^6_+$ be a positive constant vector. Then the first condition

$$\frac{(x^+)^T(s^+)}{(x^0)^T s^0} \geq \beta_1 \frac{\|(0; s^+) - \psi(y^+, x^+)\|}{\|(0; s^0) - \psi(y^0, x^0)\|} \tag{16}$$

prevents the iterates from converging towards a complementarity solution before feasibility is reached. The second condition

$$x_j^+ s_j^+ \geq \beta_2 (x^+)^T s^+/n, \quad \text{for} \quad j = 1, \ldots, n+1 \tag{17}$$

prevents the iterates from converging towards the boundary of the positive orthant prematurely. The third condition

$$\phi(y^+, x^+, s^+) \leq \phi(y, x, s) + \beta_3 \alpha \nabla \phi(y, x, s)(d_y; d_x; d_s) \tag{18}$$

requires the merit function to be reduced in all iterations. The condition (18) is well-known from the Goldstein-Armijo rule, see Dennis and Schnable [19, p. 118]. (β_4, β_5 and β_6 will be specified later.)

Related conditions have been employed by Kojima et al. [32] in their global convergence proof of the primal-dual algorithm for LP. El-Bakry et al. [22] have also used similar conditions in their algorithm for nonlinear optimization. However, they have used a slightly different merit function.

The step-size α is computed using a simple backtracking line-search. First, $\alpha^{\max}$ is computed from

$$\alpha^{\max} = \operatorname{argmax}\{\alpha \in [0, 1] : (x; s) + \alpha(d_x; d_s) \geq 0\}.$$

Next, the smallest positive integer k is chosen such that $\alpha = (\beta_4)^k \alpha^{\max}$ and the update (y^+, x^+, s^+) satisfies the conditions (16), (17), and (18), where $\beta_4 \in (0, 1)$.

In the paper [5] we have proposed another update of the variables s; instead of using (13), we let

$$\begin{aligned} s^+ &= s + \alpha d_s - (\psi_g(y^+, x^+) - \psi_g(y, x) - \alpha \nabla \psi_g(y, x)(d_y; d_x)) \\ &= s + \alpha d_s + (1 - \alpha\eta) r_g, \end{aligned} \tag{19}$$

and all the remaining variables are updated using the simple linear update (13). Using this update we have shown

$$r_g^+ = (1 - \alpha\eta) r_g.$$

Hence, the residual vector r_g is reduced linearly in the step-size. If there is no nonlinear equality constraint in the problem (MCP), i.e. h is an affine function, then we can show that the complementarity gap is also reduced linearly in the step-size. However, this is not the case, when h is a nonlinear function in (MCP). In our implementation we use both updates, where in each iteration one of them is chosen depending on which reduces the merit function more.

In the case $m = 0$ and given certain assumptions, then polynomial complexity of the homogeneous algorithm is proved in [5].

5 An implementation for convex optimization

In this section we specialize the homogeneous algorithm presented in the previous section to solve convex optimization problems and discuss an implementation of the algorithm.

The problem is

$$\begin{array}{lllll} \text{minimize} & c(x,\bar{x}) & & & \\ \text{subject to} & a_i(x,\bar{x}) & \geq & 0, & i = 1,\ldots,m_1, \\ & a_i(x,\bar{x}) & = & 0, & i = m_1+1,\ldots,m, \\ & x \geq 0, & & & \end{array} \tag{20}$$

where $x \in \mathbf{R}^{n_1}$, $\bar{x} \in \mathbf{R}^{n_2}$, $a(x,\bar{x}) \in \mathbf{R}^m$, and $z \in \mathbf{R}^{m_1}$. Also let $n := n_1 + n_2$. The function $c : \mathbf{R}^n \to \mathbf{R}$ is assumed to be convex, and the component functions $a_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1,\ldots,m_1$ are assumed to be concave whereas the component functions a_i, $\quad i = m_1+1,\ldots,m$ are assumed to be affine. Hence, the problem (20) minimizes a convex function over a convex set. All functions are assumed to be at least twice differentiable on the set $\dot{\mathcal{F}}_x$, where

$$\mathcal{F}_x := \{(x;\bar{x}) : (x;\bar{x}) \in \mathbf{R}^{n_1}_{+} \times \mathbf{R}^{n_2}\} \quad \text{and} \quad \dot{\mathcal{F}}_x := \{(x;\bar{x}) : (x;\bar{x}) \in \mathbf{R}^{n_1}_{++} \times \mathbf{R}^{n_2}\}.$$

Next, define the Lagrange function

$$L(x,\bar{x},y,\bar{y}) := c(x,\bar{x}) - (y;\bar{y})^T a(x,\bar{x}),$$

where

$$(y, \bar{y}) \in \mathcal{F}_y := \{(y; \bar{y}) : (y; \bar{y}) \in \mathbf{R}_+^{m_1} \times \mathbf{R}^{(m-m_1)}\}.$$

For future reference we also define the set

$$\dot{\mathcal{F}}_y := \{(y; \bar{y}) : (y; \bar{y}) \in \mathbf{R}_{++}^{m_1} \times \mathbf{R}^{(m-m_1)}\}.$$

For simplicity we make the following assumption.

Assumption 5.1 *For all $(x; \bar{x}; y; \bar{y}) \in \dot{\mathcal{F}}_x \times \dot{\mathcal{F}}_y$ the matrix*

$$\begin{bmatrix} \nabla^2_{(x,\bar{x})} L(x, \bar{x}, y, \bar{y}) + \begin{bmatrix} D_x & 0 \\ 0 & 0 \end{bmatrix} & -\nabla a(x, \bar{x})^T \\ \nabla a(x, \bar{x}) & \begin{bmatrix} D_z & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix}$$

is non-singular for any positive diagonal matrices $D_x \in \mathbf{R}^{n_1 \times n_1}$ and $D_z \in \mathbf{R}^{m_1 \times m_1}$.

The dual problem corresponding to (20) is

$$\begin{array}{ll} \text{maximize} & L(x, \bar{x}, y, \bar{y}) - x^T s \\ \text{subject to} & \nabla_{(x,\bar{x})} L(x, \bar{x}, y, \bar{y})^T = (s; 0), \\ & x, y, s \geq 0, \end{array} \tag{21}$$

where $s \in \mathbf{R}^{n_1}$ is the vector of dual slacks. This problem is equivalent to

$$\begin{array}{ll} \text{maximize} & L(x, \bar{x}, y, \bar{y}) - \nabla_{(x,\bar{x})} L(x, \bar{x}, y, \bar{y})(x; \bar{x}) \\ \text{subject to} & \nabla_{(x,\bar{x})} L(x, \bar{x}, y, \bar{y})^T = (s; 0), \\ & x, y, s \geq 0. \end{array} \tag{22}$$

Combining (20) and (21) give us the optimality conditions to (20) which are

$$\begin{array}{llcl} \text{minimize} & (x; y)^T (s; z) & & \\ \text{subject to} & \nabla_{(x,\bar{x})} L(x, \bar{x}, y, \bar{y})^T & = & (s; 0), \\ & a(x, \bar{x}) & = & (z; 0), \\ & x, z, y, s \geq 0. & & \end{array} \tag{23}$$

The vector $z \in \mathbf{R}^{m_1}$ is a set of surplus variables included in the problem for convenience of the algorithmic development. The problem (23) is a monotone complementarity problem with equality constraints and free variables. Hence, we can use the algorithm presented in the previous section to solve this problem. Introducing the homogenization variable we obtain the homogenized monotone complementarity problem

$$\begin{array}{lll} \text{minimize} & (x; y; \tau)^T (s; z; \kappa) & \\ \text{subject to} & \tau \nabla_{(x,\bar{x})} L(x/\tau, \bar{x}/\tau, y/\tau, \bar{y}/\tau)^T & = \ (s; 0), \\ & \tau a(x/\tau, \bar{x}/\tau) & = \ (z; 0), \\ & -(x; \bar{x})^T \nabla_{(x,\bar{x})} L(x/\tau, \bar{x}/\tau, y/\tau, \bar{y}/\tau)^T - (y; \bar{y})^T a(x/\tau, \bar{x}/\tau) & = \ \kappa, \\ & x, z, \tau, y, s, \kappa \geq 0. & \end{array} \tag{24}$$

Note that

$$\begin{aligned} \frac{\kappa}{\tau} &= -(x/\tau; \bar{x}/\tau)^T \nabla_{(x,\bar{x})} L(x/\tau, \bar{x}/\tau, y/\tau, \bar{y}/\tau)^T - (y/\tau; \bar{y}/\tau)^T a(x/\tau, \bar{x}/\tau) \\ &= -(c(x/\tau, \bar{x}/\tau) \\ &\quad -(L(x/\tau, \bar{x}/\tau, y/\tau, \bar{y}/\tau) - \nabla_{(x,\bar{x})} L(x/\tau, \bar{x}/\tau, y/\tau, \bar{y}/\tau)(x/\tau; \bar{x}/\tau))). \end{aligned}$$

Therefore, κ is equal to the negative duality gap multiplied by τ.

5.1 The search direction

Next we show how to compute the search direction. We use the notation that

$$w := (x; \bar{x}; z; \tau; y; \bar{y}; s; \kappa).$$

Also we define the set

$$\mathcal{F}_w := \{w : w \in \mathcal{F}_x \times \mathbf{R}_+^{m_1+1} \times \mathcal{F}_y \times \mathbf{R}_+^{n_1+1}\}, \tag{25}$$

and its interior

$$\dot{\mathcal{F}}_w := \{w : w \in \dot{\mathcal{F}}_x \times \mathbf{R}_{++}^{m_1+1} \times \dot{\mathcal{F}}_y \times \mathbf{R}_{++}^{n_1+1}\}. \tag{26}$$

Let $w \in \dot{\mathcal{F}}_w$ be the current iterate, then the residual vector is given by

$$r(w) := \begin{bmatrix} r_D \\ r_G \\ r_P \end{bmatrix} := \begin{bmatrix} (s;0) - \tau g \\ \kappa + (x;\bar{x})^T g + (y;\bar{y})^T a(x/\tau, \bar{x}/\tau) \\ (z;0) - \tau a(x/\tau, \bar{x}/\tau) \end{bmatrix}, \tag{27}$$

where

$$\begin{aligned} J &:= \nabla a(x/\tau, \bar{x}/\tau), \\ g &:= \nabla_{(x,\bar{x})} L(x/\tau, \bar{x}/\tau, y/\tau, \bar{y}/\tau)^T = \nabla c(x/\tau, \bar{x}/\tau)^T - J^T(y/\tau; \bar{y}/\tau), \\ H &:= \nabla^2_{(x,\bar{x})} L(x/\tau, \bar{x}/\tau, y/\tau, \bar{y}/\tau). \end{aligned}$$

Since $y \in \mathbf{R}^{m_1}_+$ and the previous assumptions, then H is positive semi-definite. The search direction from (4) and (5) is, in this case, equivalent to

$$\begin{bmatrix} H & v^2 & -J^T \\ (v^1)^T & H_g & -(v^3)^T \\ J & v^3 & 0 \end{bmatrix} \begin{bmatrix} (d_x; d_{\bar{x}}) \\ d_\tau \\ (d_y; d_{\bar{y}}) \end{bmatrix} - \begin{bmatrix} (d_s; 0) \\ d_\kappa \\ (d_z; 0) \end{bmatrix} = \eta \begin{bmatrix} r_D \\ r_G \\ r_P \end{bmatrix} \tag{28}$$

and

$$\begin{aligned} S d_x + X d_s &= -Xs + \gamma\mu e, \\ Y d_z + Z d_y &= -Zy + \gamma\mu e, \\ \kappa d_\tau + \tau d_\kappa &= -\tau\kappa + \gamma\mu, \end{aligned} \tag{29}$$

where

$$\begin{aligned} v^1 &:= \nabla c(x/\tau, \bar{x}/\tau)^T - H(x/\tau; \bar{x}/\tau), \\ v^2 &:= -\nabla c(x/\tau, \bar{x}/\tau)^T - H(x/\tau; \bar{x}/\tau), \\ v^3 &:= a(x/\tau, \bar{x}/\tau) - J(x/\tau; \bar{x}/\tau), \\ H_g &:= (x/\tau; \bar{x}/\tau)^T H(x/\tau; \bar{x}/\tau), \\ \mu &:= (x; z; \tau)^T (s; y; \kappa)/(n_1 + m_1 + 1). \end{aligned} \tag{30}$$

η and γ are parameters to be chosen subsequently. If (29) is used to eliminate $(d_z; d_s; d_\kappa)$, then the following reduced Newton equation system is obtained

$$\begin{bmatrix} D_x & v^2 & -J^T \\ (v^1)^T & D_g & -(v^3)^T \\ J & v^3 & D_z \end{bmatrix} \begin{bmatrix} (d_x; d_{\bar{x}}) \\ d_\tau \\ (d_y; d_{\bar{y}}) \end{bmatrix} = \begin{bmatrix} \eta r_D + (X^{-1}(-Xs + \gamma\mu e); 0) \\ \eta r_G + \tau^{-1}(-\tau\kappa + \gamma\mu) \\ \eta r_P + (Z^{-1}(-Zy + \gamma\mu e); 0) \end{bmatrix}, \tag{31}$$

where

$$D_x := H + \begin{bmatrix} X^{-1}S & 0 \\ 0 & 0 \end{bmatrix}, \quad D_z := \begin{bmatrix} Z^{-1}Y & 0 \\ 0 & 0 \end{bmatrix}, \quad \text{and} \quad D_g := H_g + \tau^{-1}\kappa.$$

This elimination might be numerically unstable and could be stabilized, see Freund and Jarre [24].

Observe that the coefficient matrix in (31) is in general not symmetric and cannot be made symmetric by a simple row or column scaling. Although in the special case c and a are affine functions, then the matrix (31) can be made symmetric by a simple scaling.

Let

$$K := \begin{bmatrix} D_x & -J^T \\ J & D_z \end{bmatrix}. \tag{32}$$

Then (31) can be stated in the form

$$\begin{bmatrix} K & (v^2; v^3) \\ (v^1; -v^3)^T & D_g \end{bmatrix} \begin{bmatrix} (d_x; d_{\bar{x}}; d_y; d_{\bar{y}}) \\ d_\tau \end{bmatrix} = \begin{bmatrix} \hat{r} \\ \eta r_G + \tau^{-1}(-\tau\kappa + \gamma\mu) \end{bmatrix}, \tag{33}$$

where

$$\hat{r} := \begin{bmatrix} \eta r_D + (X^{-1}(-Xs + \gamma\mu e); 0) \\ \eta r_P + (Z^{-1}(-Zy + \gamma\mu e); 0) \end{bmatrix}.$$

Given Assumption 5.1 the matrix K is non-singular and therefore define p and q as the unique solutions to the linear systems

$$Kp = (v^2; v^3) \tag{34}$$

and

$$Kq = \hat{r}. \tag{35}$$

In the linear system (31) K can be used as a block pivot to eliminate $(v^1; -v^3)^T$, which reduces the system to

$$(D_g - (v^1; -v^3)^T p) d_\tau = \eta r_G + \tau^{-1}(-\tau\kappa + \gamma\mu e) - (v^1; -v^3)^T q. \tag{36}$$

From the previous discussion it is known that the Newton equation system always has a solution, which implies the equation (36) has a solution. When d_τ has been obtained from (36) the remaining part of the search direction can be computed as follows. First compute

$$(d_x; d_{\bar{x}}; d_y; d_{\bar{y}}) = p - q d_\tau,$$

and then the vector $(d_z; d_s; d_\kappa)$ is obtained from relation (29).

Now define the matrix

$$\bar{K} = \begin{bmatrix} D_x & J^T \\ J & -D_z \end{bmatrix}, \tag{37}$$

which is symmetric and in general not positive definite. Clearly the inverse of $\bar{K}$ can be used to solve (34) and (35). Therefore, we compute a suitable matrix factorization of $\bar{K}$ and use this factorization to solve the linear systems (34) and (35). Our approach is similar to Vanderbei's quasi-definite method, see [45, 44]. The method exploits $\bar{K}$ being symmetric and sparse to save storage and to perform the factorization efficiently.

6 Update of the variables and the choice of algorithmic parameters

Let $\beta \in \mathbf{R}^6_+$ be the positive constant vector again and define

$$g(w) := x^T s + z^T y + \tau\kappa, \tag{38}$$

then the merit function is given by

$$\phi(w) = \phi(x, \bar{x}, z, \tau, y, \bar{y}, s, \kappa) := \theta g(w) + \|r(w)\|. \tag{39}$$

Recall $\theta \geq 0$ is the balancing parameter. Also define the set

$$\begin{aligned} \mathcal{N}(\beta_1, \beta_2) := \{w \in \mathcal{F}_w : \;& g(w)/g(w^0) \geq \beta_1 \|r(w)\|/\|r(w^0)\|, \\ & \min((Xs; Zy; \tau\kappa)) \geq \beta_2 g(w)/(m_1 + n_1 + 1)\}. \end{aligned} \tag{40}$$

When the search direction has been computed, then the new point w^+ is computed using

$$[w^+] = \mathit{update}(w, d_w, \beta),$$

where *update* is the procedure

1. $\mathit{procedure}\ [w^+] = \mathit{update}(w, d_w, \beta)$
2. $\alpha^0 := \operatorname{argmax}\{\alpha \in [0, 1/\beta_5] : w + \alpha d_w \in \mathcal{F}_w\}$
3. $\mathit{for}\ k = 0 : \infty$
4. $\quad \alpha := \beta_5 (\beta_4)^{2^k} \alpha^0$
5. $\quad w^l := w + \alpha d_w$
6. $$\quad w^n := \begin{bmatrix} x^+ \\ z^+ \\ \tau^+ \\ y^+ \\ \bar{y}^+ \\ s^+ \\ \kappa^+ \end{bmatrix} = \begin{bmatrix} x + \alpha d_x \\ ((1-\alpha\eta) r_P + \tau^+ a(x^+/\tau^+, \bar{x}^+/\tau^+))_{1,\ldots,m_1} \\ \tau + \alpha d_\tau \\ y + \alpha d_y \\ \bar{y} + \alpha d_{\bar{y}} \\ ((1-\alpha\eta) r_D + g^+)_{1,\ldots,n_1} \\ \kappa + \alpha d_\kappa \end{bmatrix}$$
7. $\quad \chi^l := \phi(w^l)$
8. $\quad \chi^n := \phi(w^n)$
9. $\quad \mathit{if}\ (\ w^l \notin \mathcal{N}(\beta_1, \beta_2)\ )$
10. $\quad\quad \chi^l := \infty$
11. $\quad \mathit{if}\ (\ w^n \notin \mathcal{N}(\beta_1, \beta_2)\ )$
12. $\quad\quad \chi^n := \infty$
13. $\quad \mathit{if}\ (\ \chi^l \leq \chi_n\ \mathit{and}\ \chi^l \leq \phi(w) + \alpha\beta_3 \nabla\phi(w) d_w\ )$
14. $\quad\quad \mathit{return}\ (w^l)$
15. $\quad \mathit{if}\ (\ \chi^n \leq \phi(w) + \alpha\beta_3 \nabla\phi(w) d_w\ )$
16. $\quad\quad \mathit{return}\ (w^n)$
17. $\mathit{end\ for}$

In step 6 of *update* g^+ denotes g evaluated at $(x^+, \bar{x}^+, \tau^+, y^+, \bar{y}^+)$. In each iteration of the procedure *update* two trial points w^l and w^n are computed using the linear update (13) and the nonlinear update (19). The procedure is terminated whenever an acceptable point is computed. In the case both trial points become acceptable in the same iteration of the procedure, the trial point that reduces the merit function more is chosen. It should be emphasized that *update* can be implemented such that all the functions are evaluated only once in each iteration of *update* even though two trial points are computed.

In each iteration of the homogeneous algorithm the parameters γ and η should be chosen. If $\eta = 1 - \gamma$, then the infeasibility and the complementarity gap tend to be reduced at the same rate, which is theoretically desirable. This fact is exact in the LP case i.e. when all functions in (20) are affine. Therefore, we adapt this choice of η. The parameter γ can be interpreted as a targeted reduction factor in the infeasibility and complementarity, which is the case in solving LP problems with a unit step-size.

In the following a heuristic adapted from Mehrotra [38] to choose γ is presented in detail. Let d_w^a be the pure Newton direction that is the solution to (28) and (29) for $\gamma = 0$ and $\eta = 1$. Next let

$$\nu := \frac{\phi(update(w, d_w^a, \beta))}{\phi(w)},$$

which measure the reduction in the merit function along the pure Newton direction. Finally, we let

$$\gamma := \nu \min(\nu, \beta_6),$$

The idea of this heuristic is to reduce the parameter γ more when the algorithm can make good progress towards optimality in the pure Newton direction. Moreover, $\beta_6 \in (0, 1)$ is the minimum desired reduction. Using this estimated γ and $\eta = 1 - \gamma$ we compute the final search direction d_w.

A common practice in solving LP problems is to use a high-order method to compute the search direction, proposed by Mehrotra [38]. We use the same method to perform correction on the complementarity conditions.

6.1 Simple bounds

In general (20) is specified as

$$\begin{array}{lllll} \text{minimize} & c(x, \bar{x}) & & & \\ \text{subject to} & a_i(x, \bar{x}) & \geq & 0, & i =, \ldots, m_1, \\ & a_i(x, \bar{x}) & = & 0, & i = m_1 + 1, \ldots, m, \\ & u \geq x \geq l, & & & \end{array} \tag{41}$$

where $l,\ u \in R^{n_1}$ are given lower and upper bounds. All lower bounds are assumed to be finite, but some upper bounds may be infinite. Moreover, it is assumed that $l < u$. For simplicity we will in the following assume that all upper bounds are finite. Therefore, using the transformation $x = x' + l$ and the slack variables t, then (41) is equivalent to

$$\begin{array}{lllll} \text{minimize} & c(x' + l, \bar{x}) & & & \\ \text{subject to} & a_i(x' + l, \bar{x}) & \geq & 0, & i = 1, \ldots, m_1, \\ & a_i(x' + l, \bar{x}) & \geq & 0, & i = m_1 + 1, \ldots, m, \\ & x' + t + l - u & = & 0, & \\ & x', t \geq 0, & & & \end{array} \tag{42}$$

which brings (41) to the required form. In general it can only be assumed that the functions c and a are differentiable for values of x within its bounds. It can be verified that if the initial value of $x',\ t$, and τ satisfies

$$x' + t - (u - l)\tau = 0, \tag{43}$$

then all iterates generated by the algorithm also satisfy (43). This implies that the functions will be evaluated at points strictly within their bounds.

Our code exploits the simple structure of the upper bounds constraints in (43) so that the linear algebra is handled more efficiently.

6.2 Presolve

In [2, 3] several methods for reducing the size of an LP problem is presented. The reduction phase is called presolve. We have implemented a small subset of these methods in our code. Upon termination of the optimization phase a postsolve procedure is invoked to restore the optimal solution to the original problem.

6.3 Starting point and stopping criteria

If all upper bounds in the problem (42) are infinite or if the problem functions are known to be differentiable everywhere, then the starting point

$$(x^0, \bar{x}^0, z^0, \tau^0, y^0, \bar{y}^0, s^0, \kappa^0) = (e, 0, 1, e, 0, e, 1)$$

is used. If some finite upper bounds are present we modify the starting point such that it satisfies (43).

An important issue is when to terminate the homogeneous algorithm. Clearly the algorithm cannot be terminated before an approximately optimal solution to (23) or (24) has been obtained. Let $(x^k, \bar{x}^k, z^k, \tau^k, y^k, \bar{y}^k, s^k, \kappa^k)$ be the kth iterate generated by the homogeneous algorithm. Also let r_P^k, r_D^k, and r_G^k be the vectors of residuals corresponding to the kth iterate, see (27). To measure the infeasibility the code employs the following measures

$$\rho_P^k := \frac{\left\| r_P^k \right\|}{\max(1, \|r_P^0\|)}, \tag{44}$$

$$\rho_D^k := \frac{\left\| r_D^k \right\|}{\max(1, \|r_D^0\|)}, \tag{45}$$

and

$$\rho_G^k := \frac{|r_G^k|}{\max(1, |r_G^0|)} \tag{46}$$

which essentially measure the relative reduction in the primal, dual, and gap infeasibility, respectively. The kth iterate is considered primal and dual feasible if

$$\rho_P^k \le \bar{\rho}_P \quad \text{and} \quad \rho_D^k \le \bar{\rho}_D,$$

where $\bar{\rho}_P, \bar{\rho}_D \in (0,1)$ are small user specified constants. Also define the primal and dual objective value by

$$pobj^k := c(x^k/\tau^k, \bar{x}^k/\tau^k)$$

and

$$\begin{aligned} dobj^k \;&:=\; L(x^k/\tau^k, \bar{x}^k/\tau^k, y^k/\tau^k, \bar{y}^k/\tau^k) \\ &\quad -\nabla_{(x,\bar{x})} L(x^k/\tau^k, \bar{x}^k/\tau^k, y^k/\tau^k, \bar{y}^k/\tau^k)\tfrac{(x^k;\bar{x}^k)}{\tau}, \end{aligned}$$

respectively. Using this notation then

$$\rho_A^k := \frac{|pobj^k|}{1+|dobj^k|} \tag{47}$$

measures the number of significant figures in the objective value. Therefore, if the current iterate is primal and dual feasible and the relative gap satisfies

$$\rho_A^k \le \bar{\rho}_A$$

then the algorithm is terminated. In this case the solution

$$(x^*, \bar{x}^*, z^*, y^*, \bar{y}^*, s^*) = (x^k, \bar{x}^k, z^k, y^k, \bar{y}^k s^k)/\tau^k$$

is reported to be an approximate primal-dual optimal solution to the problem (20).

Moreover, the algorithm is terminated if

$$\rho_P^k \le \bar{\rho}_P, \quad \rho_D^k \le \bar{\rho}_D, \quad \rho_G^k \le \bar{\rho}_G, \quad \text{and} \quad \mu^k \le \bar{\rho}_\mu \mu^0.$$

In this case an approximate optimal solution to the homogeneous model has been computed. If τ^k is significantly smaller than κ^k that is

$$\tau^k \le \bar{\rho}_I \min(1, \kappa^k)$$

then it is concluded that the problem (20) is either infeasible or unbounded. Otherwise, it is concluded that (20) does not have a finite optimal solution. $\bar{\rho}_A, \bar{\rho}_G, \bar{\rho}_\mu, \bar{\rho}_I \in (0,1)$ are all user specified constants.

Parameter	Value	Parameter	Value
β_1	1e-4	ε_p	1e-8
β_2	1e-8	ε_d	1e-8
β_3	1e-4	ε_o	1e-8
β_4	9e-1	ε_μ	1e-14
β_5	9.99e-1	ε_i	1e-8
β_6	0.5 or 0.75	θ	$1/\sqrt{m_1+n_1+1}$

Table 1: Default values of algorithmic parameters.

7 Computational results

In this section we report our computational results obtained with the homogeneous algorithm. Our implementation of the homogeneous algorithm is coded in ANSI C and the test is conducted on a SGI computer with 24 MIPS R4400 processors (250 mhz) and 2048 Mbytes of memory. We have not used the parallel capability of the computer in our test, that is, the program is run on one CPU only. Moreover, all reported timing results are measured in CPU seconds.

All problems are solved using the same default tolerances, which are shown in Table 1. Note β_6 is set identical to 0.5 for problems containing only linear or quadratic functions. Otherwise we use $\beta_6 = 0.75$. The computational performance for an individual problem can possibly be improved significantly by changing few of the algorithmic parameters. However, we have not been able to achieve better computational results uniformly for all the test problems using one different set of algorithmic parameters.

7.1 Quadratically constrained quadratic programming

In the first test we apply the homogeneous algorithm to solving the so-called quadratically constrained quadratic programming (QCQP) problem, which can be stated as follows

$$
\begin{array}{llll}
(QCQP) & \text{minimize} & q_0(x,\bar{x}) & \\
& \text{subject to} & q_i(x,\bar{x}) \le 0, \quad \text{for} \quad i=1,\ldots,m_1, & \\
& & A(x;\bar{x}) = b, & \\
& & l \le x \le u, &
\end{array} \tag{48}
$$

where

$$q_i(x,\bar{x}) = 1/2(x;\bar{x})^T Q_i(x;\bar{x}) + c_i^T(x;\bar{x}) + f_i.$$

Furthermore, $b \in \mathbf{R}^{m-m_1}$, $(x;\bar{x}), c_i \in \mathbf{R}^n$, $f_i \in \mathbf{R}$, and $A \in \mathbf{R}^{(m-m_1)\times n}$. It is assumed that $Q_i \in \mathbf{R}^{n\times n}$ is positive semi-definite for all $i = 0,\ldots,m_1$. This problem has applications in finance, location theory, and engineering, see [10, 21, 7, 48]. Also several researchers have developed polynomial time algorithms for solving this class of problems, see [29, 39] and [40, pp. 220-229].

We have obtained several QCQP test problems from Andersen [7] and Ben-Tal and Nemirovskii [13]. The test problems from Andersen contain both linear and quadratic constraints and all the Q matrices are diagonal matrices having two or three non-zeroes on the diagonal. The test problems by Ben-Tal and Nemirovskii are so-called single load truss topology design problems. In this case all the Q matrices are rank one matrices.

The test problems have been converted into the QMPS format, see Andersen and Xu [4]. The QMPS format is an extension of the MPS format, which allows an easy specification of the problem $(QCQP)$. These test problems are available in this format to other researchers upon request.

In Table 2 the test problems are described. The table shows the number of constraints and variables before and after presolve. (The number of constraints before presolve includes the objective function.) The number of variables does not include slack variables. The final three columns show the number iterations, the computer time required to achieve the required accuracy, and the total time spend in factoring the matrix (37).

	Before presolve.		After presolve.		Itera-	Sol.	Fac.
Name	cons.	vars.	cons.	vars.	tions	time	time
cl30a13l1	4581	4501	4579	4501	29	15.90	5.89
cl60a13l1	18161	18001	18159	18001	23	74.60	31.53
cl90a13l1	40741	40501	40739	40501	28	264.33	120.99
cl120a13l1	72321	72001	72319	72001	35	758.06	394.07
cl180a13l1	162481	162001	162479	162001	39	2392.84	1361.43
t63-1234s	1235	113	1228	113	20	2.70	0.50
t81-2040s	2041	145	2032	145	19	4.44	0.94
t96-2852s	2853	161	2837	161	17	5.84	1.28
t121-4492s	4493	221	4482	221	22	12.45	3.01
t171-8958s	8959	305	8940	305	21	29.23	7.00
t225-15556	15557	421	15542	421	30	87.66	25.08
qssp30l1	4745	5767	4744	5767	26	15.30	4.72
qssp60l1	18485	22327	18484	22327	19	61.26	27.07
qssp90l1	41225	49687	41224	49687	21	191.33	94.50
qssp120l1	72965	87847	72964	87847	29	537.77	289.96
qssp180l1	163445	196567	163444	196567	24	1320.47	814.17

Table 2: Results for solving the QCQP problems.

Name	ρ_p	ρ_d	Primal obj.	Dual obj.
cl30a13l1	3.8e-09	4.5e-09	-9.4602849378488107e-01	-9.4602850921624126e-01
cl60a13l1	1.4e-12	2.1e-09	-9.3505294166867114e-01	-9.3505295746871198e-01
cl90a13l1	2.2e-12	5.5e-10	-9.3138315652711212e-01	-9.3138317144560767e-01
cl120a13l1	6.7e-11	2.5e-10	-9.2955022926894237e-01	-9.2955024129555708e-01
cl180a13l1	2.5e-10	2.3e-10	-9.2772863671149697e-01	-9.2772863578905596e-01
t63-1234s	2.0e-11	1.2e-12	-2.0345064434013960e+02	-2.0345064627301187e+02
t81-2040s	2.7e-09	1.0e-14	-3.8241975306374269e+02	-3.8241975305447420e+02
t96-2852s	6.2e-10	1.0e-16	-4.0000000000259701e+02	-3.9999999997827081e+02
t121-4492s	1.1e-10	2.6e-17	-5.9645654722790448e+02	-5.9645654722619804e+02
t171-8958s	1.6e-09	1.3e-14	-2.5599999999396292e+02	-2.5599999989610524e+02
t225-15556	3.2e-15	9.3e-14	-5.9343439479233894e+02	-5.9343439613853207e+02
qssp30l1	1.1e-11	8.7e-11	-6.2953155460764316e+00	-6.2953155463485588e+00
qssp60l1	5.9e-09	4.5e-10	-6.3821035086883739e+00	-6.3821035066463496e+00
qssp90l1	2.0e-11	2.8e-13	-6.4237731924500254e+00	-6.4237731924492065e+00
qssp120l1	3.8e-10	1.3e-11	-6.4501427585196129e+00	-6.4501427519507875e+00
qssp180l1	6.6e-11	2.2e-12	-6.4835088374724759e+00	-6.4835088374640213e+00

Table 3: Accuracy results for solving the QCQP problems.

In general these problems are solved in a fairly low number of iterations. Moreover, the number of iterations grows slowly with the increase in the size of the problem. For some problems, notably the t* problems, the factorization time is relative small compared to the total solution time. This indicates that the computational cost of the line-search for these problems is high.

In Table 3 the accuracy measures are presented. These columns show the feasibility measures (44) and (45) and the optimal objective values. All the measures are very good. Hence, we conclude that our code solves the QCQP test problems successfully.

7.2 Geometric programming

The geometric programming (GP) problem is a well-known optimization problem, which has applications in economics, statistics, engineering design, and manufacturing. We refer the reader to the references [20, 12, 36].

The GP problem can be stated as follows

$$\begin{array}{lll} (GP) & \text{minimize} & g_0(t) \\ & \text{subject to} & g_k(t) \le 1, \quad k = 1, \ldots, p, \\ & & t \in R^m_{++}, \end{array}$$

where

$$g_0(t) = \sum_{i \in I_0} c_i \prod_{j=1}^{m} t_i^{a_{ij}} \quad \text{and} \quad g_k(t) = \sum_{i \in I_k} c_i \prod_{j=1}^{m} t_i^{a_{ij}}, \quad k = 1, \ldots, p.$$

The exponents a_{ij} are arbitrary real constants and coefficients c_i are positive. The index sets $I_0, I_1, \ldots, I_p$ are mutually disjoint and $I_0 \cup I_1 \cup \ldots \cup I_p = \{1, \ldots, T\}$. After a transformation the dual problem

$$\begin{array}{ll} \text{maximize} & f(x) \\ \text{subject to} & Ax = 0, \\ & \sum_{k \in I_0} x_k = 1, \\ & x \ge 0, \end{array} \tag{49}$$

is obtained, where

$$f(x) = \sum_{i=1}^{m} x_i \ln(c_i / x_i) + \sum_{i=1}^{p} (\sum_{k \in I_i} x_k) \ln(\sum_{k \in I_i} x_k).$$

The function f is concave and infinitely differentiable on the interior of the positive orthant. Moreover, the Hessian of f has a block diagonal structure. Our code automatically exploits this structure. A solution to (GP) can easily be recovered from the dual solution to (49), see Bazaraa et al. [11] for details. The main difficulty in solving (49) is, f is not differentiable at any solution x when at least one component of x is zero. In general if (GP) has an optimal solution, where some of the constraints are not binding, then some components in the optimal solution to (49) are zero, see Dembo [17, pp. 160-161]. This occurs very often for large problems.

We have obtained a set of GP test problems that has been used in previous studies. Computational results for most of the problems are reported in [36, 35]. Kortanek et al. [36] presented a generalization of the Kojima et al. type primal-dual algorithm to solving GP problems. Hence, their algorithm is fairly similar to the algorithm presented here, except that our algorithm is based on the homogeneous model and it is a general purpose algorithm. We have not adjusted algorithmic parameters such that the best possible performance for solving GP problems may be obtained. However, the work performed in each iteration of the algorithm presented in [36] and our algorithm is comparable. In [35] it is shown that the public widely available solvers MINOS 5.4 and LINGO-NL cannot solve all the GP problems due to numerical difficulties.

Several researchers have applied cutting planes algorithms to solving GP problems, see [18, 9]. However, they only present results for solving small-sized GP problems and the results presented in [36] indicate that these methods may not be as efficient as "direct" primal-dual methods.

In Table 4 results for the GP dual (49) of the test problems are presented. First, the table shows the number of constraints and variables before and after presolve. (The number of constraints before presolve includes the objective function.) Secondly, it shows the number of interior-point iterations and computing time to achieve the required accuracy.

In general the problems are not reduced a lot by the presolve procedure, although the problem demb782 is solved completely by the presolve procedure. (Therefore, we have not presented any solution information for this problem in Table 5.) All problems are solved in few iterations and none of the problems require more than 30 iterations. Moreover, none of the problems require more than 50 cpu seconds of computing time.

Name	Before presolve. cons.	vars.	After presolve. cons.	vars.	Iterations.	Sol. time	Fac. time
beck751	9	18	8	18	9	0.02	0.01
beck752	9	18	8	18	11	0.03	0.00
beck753	9	18	8	18	10	0.03	0.00
car	39	142	37	141	14	0.21	0.04
cx02-100	102	5247	101	5247	22	6.77	0.99
cx02-200	202	20497	201	20497	29	45.63	6.48
demb761	13	31	7	26	11	0.03	0.00
demb762	13	31	7	26	13	0.04	0.01
demb763	13	31	7	26	12	0.04	0.01
demb781	4	4	3	4	5	0.01	0.00
demb782							
fang88	13	28	7	23	14	0.04	0.01
fiac81a	24	73	23	73	12	0.06	0.00
fiac81b	12	20	11	20	18	0.05	0.02
jha88	32	305	31	305	14	0.24	0.03
mra01	63	906	61	905	18	2.04	0.62
mra02	128	3621	126	3620	20	15.79	7.30
rijc781	6	6	3	4	5	0.01	0.00
rijc782	5	9	4	9	7	0.01	0.00
rijc783	6	12	5	12	10	0.02	0.00
rijc784	6	8	5	8	7	0.01	0.01
rijc785	10	12	9	12	9	0.01	0.00
rijc786	10	12	9	12	6	0.01	0.00
rijc787	9	48	8	48	14	0.06	0.01

Table 4: Results for solving the dual GP problems.

Name	ρ_p	ρ_d	Primal obj.	Dual obj.
beck751	3.2e-11	4.3e-11	-7.5009521423565744e+00	-7.5009521640328529e+00
beck752	4.1e-11	5.7e-10	-6.8155090251553263e+00	-6.8155090527805768e+00
beck753	9.3e-15	6.2e-10	-6.2983386900385199e+00	-6.2983386900661786e+00
car	2.9e-13	1.5e-11	-3.2794477580013117e+00	-3.2794477579793431e+00
cx02-100	1.5e-17	1.9e-09	-7.7292740128748125e+00	-7.7292740734430643e+00
cx02-200	5.4e-19	1.1e-09	-9.1265800729255080e+00	-9.1265801321819708e+00
demb761	6.4e-18	0.0e+00	-2.2310862849459880e+01	-2.2310862888650291e+01
demb762	4.5e-18	9.3e-19	-1.1545067230808250e+00	-1.1545067356150565e+00
demb763	4.5e-18	4.7e-19	-1.1579029958285805e+00	-1.1579029999809780e+00
demb781	2.3e-11	2.3e-11	-6.9314717944658133e-01	-6.9314718147393162e-01
demb782				
fang88	1.8e-20	1.4e-19	-1.1328847125026869e+00	-1.1328847140308298e+00
fiac81a	1.3e-11	2.9e-10	-7.5130579739647967e+00	-7.5130579858194775e+00
fiac81b	5.6e-11	6.0e-11	-1.7292843760999482e+01	-1.7292843776350093e+01
jha88	2.7e-14	5.1e-12	-1.0389427942984188e+01	-1.0389427948893942e+01
mra01	5.0e-12	1.9e-09	-3.4206497461132104e+00	-3.4206497419227366e+00
mra02	7.9e-13	1.5e-09	-4.3179836839850863e+00	-4.3179836811629242e+00
rijc781	1.1e-14	1.1e-14	4.4142865366935284e+00	4.4142865355062524e+00
rijc782	4.7e-14	2.8e-10	-8.7482799006310490e+00	-8.7482799017344526e+00
rijc783	2.9e-13	3.9e-13	-1.1746440470341867e+01	-1.1746440470486460e+01
rijc784	2.2e-13	7.6e-12	-1.3342702802629788e+01	-1.3342702803170582e+01
rijc785	1.2e-15	5.2e-10	-3.3751779233021040e+00	-3.3751779242561186e+00
rijc786	3.0e-13	1.9e-11	-3.3750741630763139e+00	-3.3750741630835082e+00
rijc787	2.3e-13	1.9e-09	-5.1844648959440409e+00	-5.1844648960923214e+00

Table 5: Accuracy results for the dual GP problems.

In Table 6 we report the size of (GP) and the accuracy of the solution recovered to (GP). This solution is obtained using a simple conversion of the optimal dual solution to (49).

Table 6 shows the number of constraints (p), variables (m), and terms (T) in (GP). The last two columns show the maximum violation among all the primal constraints and the primal objective value. It can be seen that very good feasibility accuracy is obtained. Moreover, it can be verified that at least 7 figures of the primal objective values are identical to those reported in [35]. Hence, we conclude that the GP problems are also solved successfully by our code.

7.3 Infeasible problems

An important feature of the homogeneous algorithm is that it can detect a possible infeasible or unbounded status of the optimization problem. In this section we test this feature.

The test problem is adapted from Bretthauer and Shetty [14] and has the form

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} d_j/x_j \\ \text{subject to} & \sum_{j=1}^{n} a_j x_j \leq b, \\ & l \leq x \leq u, \\ & x \text{ integer}. \end{array} \tag{50}$$

This problem arises in stratified sampling. Note that the problem is an integer programming problem which can be solved by the branch and bound algorithm. During the branch and bound phase the problem (50) is solved repeatedly while a subset of the variables is fixed at integer values and the integer restriction for the remaining variables is relaxed. The fixing of some of the variables may cause the problem to become infeasible. Hence, infeasible problems of type (50) are realistic test problems. We have randomly generated data to make the problem infeasible. Following the suggestion in [14] we generate the problem parameters

Name	Cons.	Vars.	Terms	Max. viol.	Primal obj.
cx02-200	398	200	20497	6.5e+01	3.2397502176e+04
cx02-200	398	200	20497	2.2e-10	9.1965167260e+03
cx02-200	398	200	20497	0.0e+00	9.1965167280e+03
beck751	4	7	18	0.0e+00	1.8097647826e+03
beck752	4	7	18	0.0e+00	9.1188058739e+02
beck753	4	7	18	0.0e+00	5.4366795848e+02
car	19	37	142	2.4e-13	2.6561100493e+01
cx02-100	198	100	5247	0.0e+00	2.2739507513e+03
cx02-200	398	200	20497	0.0e+00	9.1965167280e+03
demb761	3	11	31	0.0e+00	4.8919796194e+09
demb762	3	11	31	0.0e+00	3.1724581709e+00
demb763	3	11	31	0.0e+00	3.1832509948e+00
demb781	1	2	4	0.0e+00	2.0000000000e+00
demb782	1	2	3	0.0e+00	2.0000000000e+00
fang88	3	11	28	0.0e+00	3.1045994766e+00
fiac81a	36	22	73	0.0e+00	1.8318066167e+03
fiac81b	7	10	20	0.0e+00	3.2373274272e+07
jha88	40	30	305	0.0e+00	3.2514061784e+04
mra01	83	61	906	1.7e-10	3.0589283706e+01
mra02	245	126	3621	1.0e-10	7.5037176783e+01
rijc781	2	4	6	0.0e+00	1.2103186239e-02
rijc782	1	3	9	0.0e+00	6.2998424331e+03
rijc783	1	4	12	0.0e+00	1.2630317800e+05
rijc784	3	4	8	0.0e+00	6.2324987638e+05
rijc785	7	8	12	5.8e-11	2.9229483957e+01
rijc786	7	8	12	0.0e+00	2.9226451225e+01
rijc787	7	7	48	0.0e+00	1.7847792001e+02

Table 6: Results for the primal GP.

Name	Before presolve. cons.	vars.	After presolve. cons.	vars.	Itera-tions.	Sol. time	Fac. time
is10	2	10	1	10	5	0.01	0.00
is100	2	100	1	100	5	0.03	0.01
is1000	2	1000	1	995	4	0.16	0.01
is10000	2	10000	1	9904	4	1.59	0.06
is100000	2	100000	1	99033	4	27.03	0.86

Table 7: Results for solving the infeasible problems.

Name	ρ_p	ρ_d	κ^*/τ^*
is10	8.4e-12	8.4e-12	1.2e+11
is100	1.1e-13	1.1e-13	9.5e+12
is1000	3.0e-11	3.0e-11	3.4e+10
is10000	2.5e-11	2.5e-11	4.0e+10
is100000	2.4e-11	2.4e-11	4.1e+10

Table 8: Feasibility measures for the infeasible problems.

from intervals $d_j \in [10,000, 20,000]$, $a_j \in [10, 50]$, and $l_j, u_j \in [100, 200]$. Finally, we set $b = \sum_{j=1}^{n} a_j l_j - 1$ which implies that the problem is infeasible.

In Table 7 we show the solution summary of the relaxation of problems (50). All the problems are correctly detected infeasible by the algorithm. Furthermore, it can be seen that the algorithm has used a low number of iterations to detect the infeasibility of the problems. Although in general such a low number of iterations may not be expected.

In Table 8 we have shown the optimal feasibility measures and the ratio between κ/τ in the optimal solution. These results indicate that a feasible solution to the homogeneous model has been computed and τ^* is significantly smaller than κ^* implying the problem is infeasible.

8 Conclusion

In this paper we study an extension of the homogeneous algorithm for solution of general monotone complementarity problems that possess equality constraints and free variables. We specialize the homogeneous algorithm to smooth convex optimization and discuss in detail how this algorithm can be implemented. Then, we apply this specialization to solution of some large-scale quadratically constrained quadratic programming and GP programming problems. The proposed algorithm solves these test problems in a fairly low number of iterations, even though several of the test problems contain more than 100,000 constraints and variables. Hence, we could conclude that the homogeneous algorithm is a promising algorithm for solution of general smooth convex optimization problems.

Acknowledgement We thank K. D. Andersen, K. O. Kortanek, A. Nemirovskii, and X. Xu for letting us use their test problems and for their assistance in this study.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. on Optim.*, 5(1), February 1995.

[2] E. D. Andersen. Finding all linearly dependent rows in large-scale linear programming. *Optimization Methods and Software*, 6:219–227, 1995.

[3] E. D. Andersen and K. D. Andersen. Presolving in linear programming. *Math. Programming*, 71(1):221–245, 1995.

[4] E. D. Andersen and X. Xu. The QMPS format for quadratically constrained optimization problems. Technical report, Department of Man-

agement, Odense University, Denmark, March 1996. Available from: http://www.ou.dk/busieco/man/faculty/eda.html.

[5] E. D. Andersen and Y. Ye. On a homogeneous algorithm for the monotone complementarity problem. Technical report, Department of Management Sciences, The University of Iowa, 1995. Revised november 1996, to appear Mathematical Programming.

[6] E. D. Andersen and Y. Ye. On a homogeneous algorithm for a monotone complementarity problem with nonlinear equality constraints. In M. C. Ferris and J. S. Pang, editors, *Complementarity and Variational Problems: State of the Art.* SIAM, Philadelphia, Pennsylvania, USA, 1997.

[7] K. D. Andersen. *Minimizing a Sum of Norms (Large Scale solutions of symmetric positive definite linear systems).* PhD thesis, Odense University, 1995.

[8] K. Anstreicher and J. -P. Vial. On the convergence of an infeasible primal-dual interior-point method for convex programming. *Optimization Methods and Software*, 3:273–283, 1994.

[9] O. Bahn, J. L. Goffin, J. -Ph. Vial, and O. Du Merle. Experimental behaviour of an interior point cutting plane algorithm for convex programming: an application to geometric programming. *Discrete Appl. Math.*, 49:2–23, 1994.

[10] D. P. Baron. Quadratic programming with quadratic constraints. *Naval Res. Logist. Quart.*, 19:253–260, 1972.

[11] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: Theory and algorithms.* John Wiley and Sons, New York, 2 edition, 1993.

[12] C. Beightler and D. T. Phillips. *Applied geometric programming.* John Wiley and Sons, New York, 1976.

[13] A. Ben-Tal and A. Nemirovskii. Potential reduction polynomial time method for truss topology design. *SIAM J. on Optim.*, 4(3), 1994.

[14] K. M. Bretthauer and B. Shetty. The nonlinear resource allocation problem. *Oper. Res.*, 43(4):670–683, 1995.

[15] R. W. Cottle, J. -S. Pang, and R. E. Stone. *The Linear Complementarity Problem.* Computer Science and Scientific Computing, Academic Press, San Diego, 1992.

[16] E. de Klerk, C. Roos, and T. Terlaky. Initialization in semidefinite programming via a self-dual skew-symmetric embedding. Technical Report 96-10, Faculty of Mathematics and Informatics, Delft University of Technology, January 1996.

[17] R. S. Dembo. Second order algorithms for posynomial geometric programming dual, part I: analysis. *Math. Programming*, 17:156–175, 1979.

[18] D. den Hertog, J. Kaliski, C. Roos, and T. Terlaky. A logarithmic barrier cutting plane method for convex programming. In I. Maros and G. Mitra, editors, *Annals of Operations Research*, volume 58, pages 69–98. Baltzer Science Publishers, Amsterdam, The Netherlands, 1995.

[19] J. E. Dennis Jr. and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, USA, 1983.

[20] R. J. Duffin, E. L. Peterson, and C. Zener. *Geometric programming.* John Wiley and Sons, New York, 1969.

[21] E. Phan-huy-Hao. Quadratically constrained quadratic programming: Some applications and a method for solution. *Zeitschrift für Operations Research*, 26:105–119, 1982.

[22] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the primal-dual Newton interior-point method for nonlinear programming. *J. Optim. Theory Appl.*, 89(3):507–541, June 1996.

[23] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* John Wiley and Sons, New York, 1968.

[24] R. W. Freund and F. Jarre. A QMR-based interior-point algorithm for solving linear programs. *Math. Programming*, 76(1):183–210, 1997.

[25] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. On the projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Math. Programming*, 36:183–209, 1986.

[26] J. -L. Goffin, J. Gondzio, R. Sarkissian, and J. -P Vial. Solving nonlinear multicommodity flow problems by the analytic center cutting plane method. *Math. Programming*, 76(1):131–154, 1997.

[27] J. Gondzio, R. Sarkissian, and J. -P. Vial. Using an interior point method for the master problem in a decomposition approach. Technical Report 1995.30, Logilab, HEC Geneva, Section of Management Studies, University of Geneva, October 1995. To appear in European Journal of Operations Research.

[28] O. Güler. Existence of interior points and interior paths in nonlinear monotone complementarity problems. *Math. Oper. Res.*, 18:148–162, 1993.

[29] F. Jarre. On the convergence of the method of analytic centers when applied to convex quadratic programs. *Math. Programming*, 49:341–358, 1991.

[30] F. Jarre, M. Kocvara, and J. Zowe. Interior point methods for mechanical design problems. Technical Report 173, Institut für Angewandte Mathematik, Universität Erlangen-Nürnburg, 1996.

[31] N. Karmarkar. A polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[32] M. Kojima, N. Megiddo, and S. Mizuno. A primal-dual infeasible-interior-point algorithm for linear programming. *Math. Programming*, 61:263–280, 1993.

[33] M. Kojima, S. Mizuno, and A. Yoshise. A primal-dual interior point algorithm for linear proramming. In N. Megiddo, editor, *Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods*, pages 29–47. Springer Verlag, Berlin, 1989.

[34] K. O. Kortanek, F. Potra, and Y. Ye. On some efficient interior point methods for nonlinear convex programming. *Linear Algebra Appl.*, 152:169–189, 1991.

[35] K. O. Kortanek and X. Xu. Numerical experiments with XGP-An optimizer for geometric programming and some comparisons with other solvers. Technical report, Department of Management Sciences, The University of Iowa, October 1995.

[36] K. O. Kortanek, X. Xu, and Y. Ye. An infeasible interior-point algorithm for solving primal and dual geometric programs. *Math. Programming*, 76(1):155–181, January 1997.

[37] I. J. Lustig, R. E. Marsten, and D. F. Shanno. Interior point methods for linear programming: Computational state of the art. *ORSA J. on Comput.*, 6(1):1–15, 1994.

[38] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. on Optim.*, 2(4):575–601, 1992.

[39] S. Mehrotra and J. Sun. A method of analytic centers for quadratically constrained convex quadratic programming. *SIAM J. Numer. Anal.*, 28(2):529–544, April 1991.

[40] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, PE, 1 edition, 1994.

[41] Y. Nesterov and M. Todd. Self-scaled barries and interior point methods for convex programming. Technical report, CORE, Lovain-la-Neuve, 1994. Discussion paper 9462.

[42] F. A. Potra and R. Sheng. Homogeneous interior-point algorithms for semidefinite programming. Technical Report 82/1995, Department of Mathematics, The University of Iowa, November 1995.

[43] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, March 1996.

[44] R. J. Vanderbei. Symmetric quasi-definite matrices. *SIAM J. on Optim.*, 5(1), February 1995.

[45] R. J. Vanderbei and T. J. Carpenter. Symmetric indefinite systems for interior point methods. *Math. Programming*, 58:1–32, 1993.

[46] J. -P. Vial. Computational experience with a primal-dual algorithm for smooth convex programming. *Optimization Methods and Software*, 3:273–283, 1994.

[47] X. Xu, P. -F. Hung, and Y. Ye. A simplified homogeneous and self-dual linear programming algorithm and its implementation. *Annals of Operations Research*, 62:151–171, 1996.

[48] G. Xue and Y. Ye. An efficient algorithm for minimizing a sum of Euclidean norms with applications. Technical report, Department of Computer Science and Electrical Engineering, The University of Vermont, June 1995. University of Vermont Research Report, CSEE/955/01-01, to appear in SIAM J. on Optimization.

[49] H. Yamashita and H. Yabe. Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization. *Math. Programming*, 75(3):377–397, 1996.

[50] Y. Ye, M. J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$ - iteration homogeneous and self-dual linear programming algorithm. *Math. Oper. Res.*, 19:53–67, 1994.