A Feasible Direction Algorithm for Nonlinear Second-Order Cone Optimization Problems

Alfredo Canelas* Miguel Carrasco† Julio López‡

August 21, 2014

Abstract

In this work we present a new feasible direction algorithm for solving smooth nonlinear second-order cone programs. These problems consist of minimizing a nonlinear differentiable objective function subject to some nonlinear second-order cone constraints. Given a point interior to the feasible set defined by the nonlinear constraints, the proposed approach computes a feasible and descent direction for the objective function. The search direction is computed by using a formulation that is similar to the algorithm FDIPA for nonlinear programming. A line search along the search direction finds a new feasible point that has a lower value of the objective function. Repeating this process, the algorithm generates a feasible sequence with a monotone decrease of the objective function. Under mild assumptions we prove that the present algorithm converge globally to stationary points of the nonlinear second-order cone program. We test our algorithm with several instances of robust classification of support vector machines.

Keywords: Feasible direction, second-order cone programming, interior point algorithm, support vector machines.

1 Introduction

In this paper, we consider the following *nonlinear second-order cone programming problem* (NSOCP):

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ \text{s.t.} & g^j(x) \succeq_{\mathcal{K}^{m_j}} 0,\ j=1,\dots,J, \end{cases} \tag{NSOCP}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, $g^j\colon \mathbb{R}^n \to \mathbb{R}^{m_j}$, $j=1,\dots,J$, are continuously differentiable functions, and $z \succeq_{\mathcal{K}^{m_j}} 0$ means $z \in \mathcal{K}^{m_j}$. The set $\mathcal{K}^m$ denotes the second-order cone (SOC) (also called the Lorentz cone or ice-cream cone) of dimension m, i.e. if $m=1$, $\mathcal{K}^1$ denotes the set $\mathbb{R}_+$ of nonnegative reals and if $m \geq 2$, $\mathcal{K}^m := \{(y_1,\bar{y}) \in \mathbb{R}\times\mathbb{R}^{m-1} : \ \|\bar{y}\| \leq y_1\}$, where $\|\cdot\|$ denotes the Euclidean norm. Since the norm is not differentiable at 0, this problem does not belong to the class of smooth nonlinear programs.

*Instituto de Estructuras y Transporte, Facultad de Ingeniería, Universidad de la República, Julio Herrera y Reissig 565, Montevideo, Uruguay (acanelas@fing.edu.uy).

†Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Mons. Álvaro del Portillo 12.455, Las Condes, Santiago, Chile. (migucarr@uandes.cl).

‡Facultad de Ingeniería, Universidad Diego Portales, Ejército 441, Santiago, Chile (julio.lopez@udp.cl).

On the other hand, a second-order cone can be rewritten using smooth nonconvex constraints: $\mathcal{K}^m = \{y \in \mathbb{R}^m : y_2^2 + \ldots + y_m^2 \leq y_1^2,\, y_1 \geq 0\}$. However, these constraints are not qualified at 0 (see, e.g. [11, Definition 3.20]).

In recent years, linear SOCP have received considerable attention because of its wide range of applications such as antenna array weight design, support vector machines and data classification, combinatorial optimization, control, and robust optimization (see for instance [1, 2, 20, 26] and references there in). It is known that $\mathcal{K}^m$, like $\mathbb{R}^m_+$ and the cone $\mathcal{S}^m_+$ of $m \times m$ real symmetric positive semidefinite matrices, belongs to the class of symmetric cones to which a Jordan algebra may be associated [8]. Using this connection, interior-point methods have been proposed for solving linear programs with SOC constraints [1, 20, 22, 28] and software implementing those methods has been also developed, see e.g. [2, 27]. On the other hand, linear SOCP problems are essentially a specific case of linear semidefinite programming (SDP) problems, see e.g. [1], and hence can be efficiently solved by using an algorithm for SDP problems. However, it is pointed out by Nesterov and Nemirovskii, see [23], that an interior-point method (IPM) that solves the SOCP problem directly has much better complexity than an IPM applied to the semidefinite formulation of the SOCP problem.

The study of nonlinear SOCP is much more recent than the linear one. For instance, theoretical properties or associated reformulations have been presented in [4, 6], primal-dual interior-point method has been studied in [29], a sequential quadratic programming type method has been proposed in [16], an augmented Lagrangian method in [18, 19], a semismooth Newton method without strict complementarity condition in [15] and recently a method based on differentiable exact penalty function has been proposed in [9].

In this paper we propose an algorithm to find a stationary point of (NSOCP) by generating a sequence $\{x^{k+1}\}_{k\in\mathbb{N}}$ that is interior to the feasible set of the (NSOCP), i.e. $\{x^{k+1}\}_{k\in\mathbb{N}} \subset \operatorname{int}(\Omega)$, where $\operatorname{int}(\Omega)$ denotes the interior of the feasible set

$$\Omega := \{x \in \mathbb{R}^n \;:\; g^j(x) \in \mathcal{K}^{m_j},\; j = 1, \ldots, J\}.$$

The sequence reduces monotonically the value of the objective function, i.e. $f(x^{k+1}) \leq f(x^k)$, $k \in \mathbb{N}$. Algorithms of this type are very useful in Engineering applications, where a feasible point represents a feasible design and functions evaluation is in general very expensive, so that each iteration represents a considerable amount of time. Since any point in the sequence is feasible, the iteration can safely stopped when the objective reduction per iteration becomes small enough. On the other hand, this approach is essential to solve problems in which the objective function, or some of the constraints are not defined or cannot be computed by using the available codes at infeasible points [13]. This is a typical case when one of the variables represent quantities that are physically meaningful only in the case that some of the constraints are satisfied.

This paper is organized as follows. Section 2 is devoted to the preliminaries and it is split into two subsections. The first one recalls some basic notions and properties associated with SOC, while the second one recalls the notion of feasible direction. In Section 3 the feasible directions interior point algorithm for NSOCP is introduced and its global convergence to stationary points is proved. In Section 4 some numerical examples are studied. Finally, concluding remarks are given in Section 5.

2 Preliminaries

The following notation and terminology are used throughout the paper. The superscript $\top$ denotes transpose operator. For two matrices A and B, we define

$$A \oplus B := \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}.$$

The set $\mathrm{int}(\mathcal{K}^m) = \{(y_1, \bar{y}) \in \mathbb{R} \times \mathbb{R}^{m-1} : \ \|\bar{y}\| < y_1\}$ denotes the topological interior of the closed convex self-dual cone $\mathcal{K}^m$. If $y \in \mathrm{int}(\mathcal{K}^m)$, we say that $y \succ_{\mathcal{K}^m} 0$. The boundary of $\mathcal{K}^m$ is the set $\partial\mathcal{K}^m = \{y \in \mathcal{K}^m : y_1 = \|\bar{y}\|\}$. Let $\mathcal{K}$ be the Cartesian product of several second-order cones, namely, $\mathcal{K} = \mathcal{K}^{m_1} \times \ldots \times \mathcal{K}^{m_J}$. Define $\mathbf{g}(x) := (g^1(x), \ldots, g^J(x)) \in \mathbb{R}^m$, where $g^j : \mathbb{R}^n \to \mathbb{R}^{m_j}$ and $\mathbf{y} = (y_1, \ldots, y_J) \in \mathbb{R}^m$, $y_j \in \mathbb{R}^{m_j}$, where $m = \sum_{j=1}^{J} m_j$. For any scalar differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ denotes its gradient vector. For a differentiable mapping $g : \mathbb{R}^n \to \mathbb{R}^m$, $Dg : \mathbb{R}^n \to \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$ denotes its differential mapping defined by $Dg(x)h = \nabla g(x)h$, for all $h \in \mathbb{R}^n$, where $\nabla g(x) \in \mathbb{R}^{m\times n}$ denotes the Jacobian of g at $x \in \mathbb{R}^n$. Finally, $\mathcal{S}^m_+$ and $\mathcal{S}^m_{++}$ denote the sets of positive semidefinite and definite matrices.

2.1 Algebra preliminaries

Let us recall some basic concepts and properties about the Jordan algebra associated with the second-order cone $\mathcal{K}^m$ with $m \geq 2$ (see [8] for more details). The *Jordan product* of any pair $v = (v_1, \bar{v})$, $w = (w_1, \bar{w}) \in \mathbb{R} \times \mathbb{R}^{m-1}$ is defined by $v \circ w = (v^\top w, v_1\bar{w} + w_1\bar{v})$. The bilinear mapping $(v, w) \mapsto v \circ w$ has as the unit element $e = (1, 0, \ldots, 0) \in \mathbb{R}^m$, satisfies $\langle v \circ w, z\rangle = \langle v, w \circ z\rangle$ for all $u, w, z \in \mathbb{R}^m$, is commutative but not associative in general, which is a main source of complication in the analysis of SOCP. However, $\circ$ is power associative, that is, for all $w \in \mathbb{R}^m$, w^k can be unambiguously defined as $w^k = w^p \circ w^q$ for any $p, q \in \mathbb{N}$ with $p + q = k$. If $w \in \mathcal{K}^m$, then there exists a unique vector in $\mathcal{K}^m$, denoted by $w^{1/2}$, such that $(w^{1/2})^2 = w^{1/2} \circ w^{1/2} = w$.

Given a vector $v = (v_1, \bar{v}) \in \mathbb{R} \times \mathbb{R}^{m-1}$, the *Arrow Matrix* of v is

$$\mathrm{Arw}(v) := \begin{pmatrix} v_1 & \bar{v}^\top \\ \bar{v} & v_1 I_{m-1} \end{pmatrix},$$

which can be viewed as a linear mapping from $\mathbb{R}^m$ to $\mathbb{R}^m$. It is not hard to verify that $\mathrm{Arw}(v)w = v \circ w$.

The *spectral factorization* of vectors in $\mathbb{R}^m$ associated with $\mathcal{K}^m$ is, for any $w = (w_1, \bar{w}) \in \mathbb{R} \times \mathbb{R}^{m-1}$, the decomposition

$$w = \lambda_1(w)u_1(w) + \lambda_2(w)u_2(w), \tag{2.1}$$

where $\lambda_i(w)$ and $u_i(w)$ for $i = 1, 2$ are the *spectral values* and *spectral vectors* of w given by

$$\lambda_i(w) = w_1 + (-1)^i\|\bar{w}\| \quad \text{and} \quad u_i(w) = \begin{cases} \frac{1}{2}(1, (-1)^i \frac{\bar{w}}{\|\bar{w}\|}) & , \text{ if } \bar{w} \neq 0, \\ \frac{1}{2}(1, (-1)^i\bar{v}) & , \text{ if } \bar{w} = 0, \end{cases} \tag{2.2}$$

with $\bar{v}$ being any unit vector in $\mathbb{R}^{m-1}$ (satisfying $\|\bar{v}\| = 1$). If $\bar{w} \neq 0$, the decomposition is unique. Notice that $\lambda_1(w) \leq \lambda_2(w)$. We also denote $\lambda_{\min}(w) = \lambda_1(w)$, $\lambda_{\max}(w) = \lambda_2(w)$. Some basic properties of these definitions are summarized below (see [1, 8, 10]).

Proposition 2.1. *For any $w = (w_1, \bar{w}) \in \mathbb{R} \times \mathbb{R}^{m-1}$, we have:*

(a) $\|u_i(w)\| = \frac{1}{\sqrt{2}}$ *and* $u_i(w) \in \partial\mathcal{K}^m$ *for* $i = 1, 2$.

(b) $u_1(w)$ *and* $u_2(w)$ *are orthogonal for the Jordan product:* $u_1(w) \circ u_2(w) = 0$.

(c) $u_i(w)$ *is idempotent for the Jordan product:* $u_i(w) \circ u_i(w) = u_i(w)$ *for* $i = 1, 2$.

(d) $\lambda_{\min}(w), \lambda_{\max}(w)$ *are nonnegative (resp. positive) if and only if* $w \in \mathcal{K}^m$ *(resp.* $w \in \operatorname{int}(\mathcal{K}^m)$*).*

(e) *The Euclidean norm of* w *can be represented as* $\|w\|^2 = \frac{1}{2}(\lambda_{\min}(w)^2 + \lambda_{\max}(w)^2)$ *and it satisfies* $\|w^2\| \leq \sqrt{2}\|w\|^2$.

(f) $\operatorname{Arw}(w)$ *is positive semidefinite (resp. definite) if and only if* $w \in \mathcal{K}^m$ *(resp.* $w \in \operatorname{int}(\mathcal{K}^m)$*). If* $\operatorname{Arw}(w)$ *is invertible,*

$$\operatorname{Arw}(w)^{-1} = \frac{1}{\det(w)} \begin{pmatrix} w_1 & -\bar{w}^\top \\ -\bar{w} & \frac{\det(w)}{w_1} I_{m-1} + \frac{1}{w_1}\bar{w}\bar{w}^\top \end{pmatrix},$$

where $\det(w) := \lambda_{\min}(w)\lambda_{\max}(w) = w_1^2 - \|\bar{w}\|^2$ *denotes the determinant of* w.

Any pair of vectors $\{u_1, u_2\}$ satisfying properties (b), (c) of Proposition 2.1 and $u_1 + u_2 = e$ is called a *Jordan frame*, which is always of the form (2.2).

The next result provides some interesting properties that will be usefull in Section 3 (see [1, Theorem 6 and Lemma 15]).

Proposition 2.2. *The following results hold:*

(a) $x \in \mathcal{K}^m$ *if and only if* $\langle x, y\rangle \geq 0$ *holds for all* $y \in \mathcal{K}^m$*. Moreover,* $x \in \operatorname{int}(\mathcal{K}^m)$ *if and only if* $\langle x, y\rangle > 0$ *for all* $y \in \mathcal{K}^m \setminus \{0\}$.

(b) *For* $v, w \in \mathbb{R}^m$ *the following conditions are equivalent:*

   (i) $v, w \in \mathcal{K}^m$*, and* $\langle v, w\rangle = 0$.

   (ii) $v, w \in \mathcal{K}^m$*, and* $v \circ w = 0$.

   In each case, the elements v *and* w *operator commute, that is,* $v = \lambda_1 u_1 + \lambda_2 u_2$ *and* $w = \beta_1 u_1 + \beta_2 u_2$*, where* $\{u_1, u_2\}$ *is a Jordan frame. This is equivalent to saying* $\operatorname{Arw}(v)\operatorname{Arw}(w) = \operatorname{Arw}(w)\operatorname{Arw}(v)$.

A vector $w = (w_1, \bar{w}) \in \mathbb{R} \times \mathbb{R}^{m-1}$ is said to be nonsingular if $\det(w) \neq 0$. If w is nonsingular, then there exists a unique $v = (v_1, \bar{v}) \in \mathbb{R} \times \mathbb{R}^{m-1}$ such that $w \circ v = v \circ w = e$. We call this v the inverse of w and denote it by w^{-1}. Direct calculations yields $w^{-1} = \frac{1}{w_1^2 - \|\bar{w}\|^2}(w_1, -\bar{w}) = \frac{1}{\det(w)}(\operatorname{tr}(w)e - w)$, where $\operatorname{tr}(w) = \lambda_1(w) + \lambda_2(w) = 2w_1$ denotes the trace of w.

Definition 2.1. *Let* $\mathbf{v} = (v_1, \ldots, v_J)$, $\mathbf{w} = (w_1, \ldots, w_J)$ *and* $v_j, w_j \in \mathbb{R}^{m_j}$ *for* $j = 1, \ldots, J$*. Then,*

(a) $\mathbf{v} \circ \mathbf{w} = (v_1 \circ w_1, \ldots, v_J \circ w_J)$.

(b) $\operatorname{Arw}(\mathbf{v}) = \operatorname{Arw}(v_1) \oplus \ldots \oplus \operatorname{Arw}(v_J)$.

(c) $\mathbf{v}$ *and* $\mathbf{w}$ *operator commute if and only if* v_j *and* w_j *operator commute for all* $j = 1, \ldots, J$.

2.2 Feasible direction

Definition 2.2. *The vector $d \in \mathbb{R}^n$ is a* **feasible direction** *at $x \in \Omega$, if for some $\theta > 0$ we have $x + td \in \Omega$ for all $t \in [0, \theta]$.*

Note that any vector is a feasible direction at $x \in \text{int}(\Omega)$. When g^j is a convex function, with respect to the convex cone $\mathcal{K}^{m_j}$ for $j = 1, \ldots, J$; i.e. for all $x, y \in \mathbb{R}^n$ and $t \in [0, 1]$, $g^j(tx + (1-t)y) \succeq_{\mathcal{K}} tg^j(x) + (1-t)g^j(y)$, a weaker condition for a vector d to be a feasible direction is that $g^j(x + \theta d) \in \mathcal{K}^{m_j}$ for some $\theta > 0$, for all $j = 1, \ldots, J$. In fact, because of the convexity of g^j for $j \in \{1, \ldots, J\}$ we have for any $t \in [0, \theta]$:

$$g^j(x + td) = g^j\left(\frac{t}{\theta}(x + \theta d) + \left(1 - \frac{t}{\theta}\right)x\right) \succeq_{\mathcal{K}^{m_j}} \frac{t}{\theta} g^j(x + \theta d) + \left(1 - \frac{t}{\theta}\right) g^j(x).$$

The following result provides a verifiable condition to show the feasibility of a given direction when $\mathcal{K} = \mathcal{K}^m$ (the case of one single cone). The extension of this result to several cones is direct.

Proposition 2.3. *Let $d \in \mathbb{R}^n$ and $x \in \Omega$. Suppose that $g \colon \mathbb{R}^n \to \mathbb{R}^m$ is differentiable at x and that $g(x) \in \partial\mathcal{K}^m$. Let $\alpha_1 u_1 + \alpha_2 u_2$ be the spectral decomposition of $g(x)$.*

(a) If $\nabla g(x)d \in \text{int}(\mathcal{K}^m)$ when $g(x) = 0$, or

(b) If $\langle \nabla g(x)d, u_1 \rangle > 0$, when $g(x) \in \partial\mathcal{K}^m \setminus \{0\}$,

then d is a feasible direction for the problem (NSOCP) at x.

Proof. Since g is differentiable at x, the following expansion holds

$$g(x + td) = g(x) + t\nabla g(x)d + o(t), \quad \text{with } \lim_{t \to 0} \frac{o(t)}{t} = 0.$$

Then, in the case $g(x) = 0$ we have $g(x + td) = t(\nabla g(x)d + o(t)/t)$. Since $\nabla g(x)d \in \text{int}(\mathcal{K}^m)$ and $\lim_{t\to 0} o(t)/t = 0$ there exists $\theta > 0$ such that $\nabla g(x)d + o(t)/t \in \mathcal{K}^m$ for all $t \in (0, \theta]$. Then, $g(x + td) \in \mathcal{K}^m$ for all $t \in (0, \theta]$ and the case (a) is proven. To prove the case (b) we note first that $\lambda_1(w)$ is differentiable with gradient $\nabla\lambda_1(w) = 2u_1(w)$ at $w \in \partial\mathcal{K}^m \setminus \{0\}$. In addition, the condition $g(x) \in \partial\mathcal{K}^m \setminus \{0\}$ can be equivalently expressed as $\lambda_1(g(x)) = 0$ and $\lambda_2(g(x)) > 0$. Therefore, considering the function $c(x) = \lambda_1(g(x))$ we get that if $\langle \nabla c(x), d \rangle > 0$, then, d is a feasible direction at x. Consecuently, by using the chain rule we obtain $\langle \nabla c(x), d \rangle = \langle \nabla\lambda_1(g(x)), \nabla g(x)d \rangle = 2\langle u_1, \nabla g(x)d \rangle$ and the case (b) is proven. □

3 Feasible direction interior point algorithm for NSOCP

Let $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, with $m = \sum_{j=1}^J m_j$, be the Lagrangian function of problem (NSOCP) defined by

$$L(x, \mathbf{y}) = f(x) - \sum_{j=1}^J \langle g^j(x), y^j \rangle,$$

where $\mathbf{y} \in \mathbb{R}^m$ is the Lagrange multiplier vector. Then Karush-Kuhn-Tucker (KKT) conditions for the optimality problem (NSOCP) are given by

$$\nabla_x L(x, \mathbf{y}) = \nabla f(x) - \sum_{j=1}^{J} \nabla g^j(x)^\top y^j = 0, \tag{3.1}$$

$$\langle g^j(x), y^j \rangle = 0, \quad j = 1, \dots, J, \tag{3.2}$$

$$g^j(x),\ y^j \in \mathcal{K}^{m_j}, \quad j = 1, \dots, J. \tag{3.3}$$

Note that, from Proposition 2.2, Part (b), the relation (3.2) can be replaced by $g^j(x) \circ y^j = 0$, for $j = 1, \dots, J$. Also, the KKT conditions are necessary optimality conditions under the constraint qualification of Assumption 3.4 below [5].

Definition 3.1. *A feasible point $x \in \Omega$ of (NSOCP) is called a stationary point if there exist $\mathbf{y} \in \mathbb{R}^m$ such that the (KKT) conditions* (3.1)–(3.2) *are satisfied at* $(x, \mathbf{y})$. *Additionally, if* (3.3) *holds, the stationary point is called KKT point.*

From now on, we suppose that the following assumptions hold true:

Assumption 3.1. *There exists a real number a such that the set $\Omega_a = \{x \in \Omega : f(x) \leq a\}$ is compact and* $\operatorname{int}(\Omega_a) \neq \emptyset$.

Assumption 3.2. *Each $x \in \operatorname{int}(\Omega_a)$ satisfies* $\mathbf{g}(x) \succ_{\mathcal{K}} 0$.

Assumption 3.3. *The mappings $\nabla f \colon \mathbb{R}^n \to \mathbb{R}^n$, $(\nabla g^j)^\top \colon \mathbb{R}^n \to \mathcal{L}(\mathbb{R}^{m_j}, \mathbb{R}^n)$, for $j = 1, \dots, J$, are Lipschitz continuous.*

Assumption 3.4. *(Nondegeneracy condition) Each feasible point $x^* \in \Omega_a$ of (NSOCP) is nondegenerate [5, Definition 4.70], i.e.*

$$D\mathbf{g}(x^*)(\mathbb{R}^n) + \operatorname{Lin}(\mathcal{T}_{\mathcal{K}}(\mathbf{g}(x^*))) = \mathbb{R}^m, \tag{3.4}$$

where $\mathcal{T}_{\mathcal{K}}(\mathbf{g}(x^*))$ is the tangent cone of $\mathcal{K}$ at $\mathbf{g}(x^*)$ in the sense of convex analysis, see [5], and $\operatorname{Lin}(\mathcal{C})$ is the linearity space of $\mathcal{C}$, i.e. the largest linear space contained in $\mathcal{C}$.

3.1 The Newton-like iteration

A Newton-like iteration to solve the nonlinear system of equations (3.1)-(3.2) can be stated as the following block matrix form

$$\begin{pmatrix} B & -\nabla \mathbf{g}(x)^\top \\ \operatorname{Arw}(\mathbf{y})\nabla \mathbf{g}(x) & \operatorname{Arw}(\mathbf{g}(x)) \end{pmatrix} \begin{pmatrix} x_a - x \\ \mathbf{y}_a - \mathbf{y} \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - \nabla \mathbf{g}(x)^\top \mathbf{y} \\ \operatorname{Arw}(\mathbf{g}(x))\mathbf{y} \end{pmatrix}, \tag{3.5}$$

where $\operatorname{Arw}(\cdot)$ is meant in the direct sum sense (cf. Definition 2.1), $(x, \mathbf{y})$ is the starting (interior) point of the iteration and $B \in \mathbb{R}^{n \times n}$ is any positive definite matrix. Typically, B is chosen as a quasi-Newton estimate of $\nabla_x^2 L(x, \mathbf{y})$. In particular, when $B = \nabla_x^2 L(x, \mathbf{y})$, we get the well known Newton iteration of (3.1)-(3.2). Defining $d_a = x_a - x$ the direction in the primal space, then, we obtain from (3.5) the following equivalent linear system:

$$\begin{pmatrix} B & -\nabla \mathbf{g}(x)^\top \\ \operatorname{Arw}(\mathbf{y})\nabla g(x) & \operatorname{Arw}(\mathbf{g}(x)) \end{pmatrix} \begin{pmatrix} d_a \\ \mathbf{y}_a \end{pmatrix} = - \begin{pmatrix} \nabla f(x) \\ 0 \end{pmatrix}. \tag{3.6}$$

Remark 3.1. *If the solution of* (3.6) *with* $x \in \operatorname{int}(\Omega)$ *satisfies* $d_a = 0$*, then* $\mathbf{y}_a = 0$ *and hence* x *is a stationary point of (NSOCP).*

The next result shows that if the direction d_a, solution to (3.6), satisfies $d_a \neq 0$, then d_a is a descent direction of the objective function.

Lemma 3.2. *Let* $(x, \mathbf{y}) \in \Omega \times \operatorname{int}(\mathcal{K})$*, and suppose that* y *and* $g(x)$ *operator commute. Then, for any positive definite matrix* B*, the direction* d_a *solution to the system* (3.6) *is a descent direction of* f*.*

Proof. Scalar multiplication by d_a on both sides of the first equality in (3.6) gives

$$\langle Bd_a, d_a\rangle - \langle \mathbf{y}_a, \nabla \mathbf{g}(x) d_a\rangle = - \langle \nabla f(x), d_a\rangle . \tag{3.7}$$

On the other hand, as $\mathbf{y} \in \operatorname{int}(\mathcal{K})$, we have that the diagonal block matrix $\operatorname{Arw}(\mathbf{y})$ is a positive definite symmetric matrix (see Proposition 2.1(f)). Then, from the second equality in (3.6) we get

$$\nabla \mathbf{g}(x) d_a = -[\operatorname{Arw}(\mathbf{y})]^{-1} \operatorname{Arw}(\mathbf{g}(x))\mathbf{y}_a. \tag{3.8}$$

Since $(x, \mathbf{y}) \in \Omega \times \operatorname{int}(\mathcal{K}^m)$ and $\mathbf{g}(x)$ and $\mathbf{y}$ operator commute, from Proposition 2.2, Proposition 2.1(f) and [14, Exercise 7.6.10] it follows that the diagonal block matrix $\operatorname{Arw}(\mathbf{y})^{-1} \operatorname{Arw}(\mathbf{g}(x))$ is a positive semidefinite symmetric matrix. Hence,

$$- \langle \mathbf{y}_a, \nabla \mathbf{g}(x) d_a\rangle = \left\langle \mathbf{y}_a, [\operatorname{Arw}(\mathbf{y})]^{-1} \operatorname{Arw}(\mathbf{g}(x))\mathbf{y}_a\right\rangle \geq 0 \tag{3.9}$$

Using this inequality and the fact that B is positive definite, we get from (3.7), that d_a is a descent direction of the objective function f. □

This descent direction d_a cannot be taken as a search direction, since it is not always a feasible direction when x is at the boundary of the feasible set. In fact, let $x \in \partial\Omega$ and $\mathbf{y} \in \operatorname{int}(\mathcal{K})$ such that $\mathbf{y}$ and $\mathbf{g}(x)$ operator commute. We take any $j \in \{1, \ldots, J\}$, and let $g^j(x) = \lambda_1^j u_1^j + \lambda_2^j u_2^j$ be the spectral decomposition of $g^j(x)$. We have the following cases: (i) $\lambda_1^j = \lambda_2^j = 0$ and (ii) $\lambda_1^j = 0$, $\lambda_2^j > 0$. In the first case, we do not have a condition to ensure that d_a is a feasible direction, because $\nabla g^j(x) d_a = 0$ (cf. (3.8)). Now, let us consider the second case. It follows from the second equality of (3.6) that

$$\begin{aligned} 0 =& \langle \operatorname{Arw}(y^j)\nabla g^j(x) d_a, u_1^j\rangle + \langle \operatorname{Arw}(g^j(x)) y_a^j, u_1^j\rangle \\ =& \langle \nabla g^j(x) d_a, y^j \circ u_1^j\rangle + \langle y_a^j, g^j(x) \circ u_1^j\rangle \\ =& \langle \nabla g^j(x) d_a, y^j \circ u_1^j\rangle, \end{aligned}$$

where we have used Proposition 2.1(b) in the third equality. Since y^j and $g^j(x)$ operator commute, from the above equality, we deduce that $\langle \nabla g^j(x) d_a, u_1^j\rangle = 0$ (cf. Proposition 2.2 and Proposition 2.1(c)). Hence, the assumption of Proposition 2.3 is not true in the set $\partial\Omega$.

To obtain a feasible direction, we add a positive vector in the right side of the second equality of (3.6). Consider $\rho > 0$ and the following linear system:

$$\begin{pmatrix} B & -\nabla \mathbf{g}(x)^\top \\ \operatorname{Arw}(\mathbf{y})\nabla \mathbf{g}(x) & \operatorname{Arw}(\mathbf{g}(x)) \end{pmatrix} \begin{pmatrix} d \\ \hat{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} -\nabla f(x) \\ \rho \mathbf{y} \end{pmatrix}. \tag{3.10}$$

The following result shows that d, solution to the above system, is a feasible direction.

Lemma 3.3. *Let $(x, \mathbf{y}) \in \Omega \times \text{int}(\mathcal{K})$. Suppose that $\mathbf{y}$ and $\mathbf{g}(x)$ operator commute. Then, the solution d to* (3.10) *is a feasible direction at x.*

Proof. Under Assumption 3.1 the case $x \in \text{int}(\Omega)$ is trivial since in that point any direction is feasible. Consider $x \in \partial\Omega$. In order to prove that d is a feasible direction, we show that d satisfies the conditions of Proposition 2.3. We take any $j \in \{1, \dots, J\}$, and let us consider $g^j(x) = \lambda_1(g^j(x))u_1^j + \lambda_2(g^j(x))u_2^j$ the spectral decomposition of $g^j(x)$. Since $x \in \partial\Omega$ we have the following cases: (i) $\lambda_1^j = \lambda_2^j = 0$ and (ii) $\lambda_1^j = 0$, $\lambda_2^j > 0$. If case (i) holds, then, from the second equality of (3.10), we have

$$\text{Arw}(y^j)\nabla g^j(x)d = \rho y^j \, ,$$

hence $\nabla g^j(x)d = \rho \ \text{Arw}(y^j)^{-1}y^j = \rho e$, consecuently, $\nabla g^j(x)d \in \text{int}(\mathcal{K}^{m_j})$.

In the case (ii), scalar multiplication of both sides of the $j-$th equation of second equality of (3.10) by u_1^j implies that

$$\langle \nabla g^j(x)d, u_1^j \rangle = \langle \nabla g^j(x)d, y^j \circ u_1^j \rangle = \rho \langle y^j, u_1^j \rangle = \frac{\rho}{2}, \tag{3.11}$$

where we have used Proposition 2.1, parts (a), (b) and (c), and Proposition 2.2. Hence, in both cases, Proposition 2.3 implies that d is a feasible direction. □

The feasible direction d is not necessarily a descent direction for all $\rho > 0$, since the addition of a positive vector in the right side of the second equality of (3.6) produces the effect of deflecting d_a into the feasible region and this deflection of d_a grows with ρ. To ensure that the vector d be a descent direction, we need to impose a convenient upper bound on ρ. This bound is obtained by imposing the following condition (see [13]):

$$\langle d, \nabla f(x) \rangle \leq \xi \langle d_a, \nabla f(x) \rangle, \quad \xi \in (0, 1). \tag{3.12}$$

Clearly, (3.12) implies that d is a descent direction at x whenever d_a be also a descent direction.

3.2 Algorithm FDIPA for solving NSOCP

The algorithm proposed to solve the problem (NSOCP) is the following:

Algorithm FDIPA-NSOCP: Choose the parameters $\xi, \eta, \nu \in (0, 1)$ and $\varphi > 0$.

Step 0: Start with initial $x^0 \in \text{int}(\Omega_a)$, $\mathbf{y}^0 \in \text{int}(\mathcal{K})$ such that operator commutes with $\mathbf{g}(x^0)$, and $B^0 \in \mathcal{S}^n_{++}$. Set $k = 0$.

Step 1: Computation of the search direction.

(i) Compute d_a^k and $\mathbf{y}_a^k$ by solving the linear system

$$\begin{pmatrix} B^k & -\nabla \mathbf{g}(x^k)^\top \\ \text{Arw}(\mathbf{y}^k)\nabla \mathbf{g}(x^k) & \text{Arw}(\mathbf{g}(x^k)) \end{pmatrix} \begin{pmatrix} d_a^k \\ \mathbf{y}_a^k \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) \\ 0 \end{pmatrix}. \tag{3.13}$$

If $d_a^k = 0$, stop.

(*ii*) Compute d_b^k and $\mathbf{y}_b^k$ by solving the linear system

$$\begin{pmatrix} B^k & -\nabla \mathbf{g}(x^k)^\top \\ \mathrm{Arw}(\mathbf{y}^k)\nabla \mathbf{g}(x^k) & \mathrm{Arw}(\mathbf{g}(x^k)) \end{pmatrix} \begin{pmatrix} d_b^k \\ \mathbf{y}_b^k \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{y}^k \end{pmatrix}. \tag{3.14}$$

(*iii*) If $\langle d_b^k, \nabla f(x^k)\rangle > 0$, set

$$\rho^k = \min\left\{\varphi\|d_a^k\|^2, (\xi-1)\frac{\langle d_a^k, \nabla f(x^k)\rangle}{\langle d_b^k, \nabla f(x^k)\rangle}\right\}.$$

Otherwise, set

$$\rho^k = \varphi\|d_a^k\|^2.$$

(*iv*) Compute

$$d^k = d_a^k + \rho^k d_b^k \quad \text{and} \quad \hat{\mathbf{y}}^k = \mathbf{y}_a^k + \rho^k \mathbf{y}_b^k.$$

Step 2: (Armijo line search): compute t^k as the first number of the sequence $\{1, \nu, \nu^2, \ldots\}$ satisfying

$$\begin{aligned} &f(x^k + t^k d^k) \le f(x^k) + t^k \eta \nabla f(x^k)^\top d^k, \text{ and} \\ &\mathbf{g}(x^k + t^k d^k) \succ_{\mathcal{K}} 0. \end{aligned}$$

Step 3: Updates. Set $x^{k+1} = x^k + t^k d^k$. Define $\mathbf{y}^{k+1} \in \mathrm{int}(\mathcal{K})$, such that operator commutes with $\mathbf{g}(x^{k+1})$, and $B^{k+1} \in \mathcal{S}_{++}^n$.

Replace k by $k+1$ and go to **Step 1.**

For the sake of simplicity, global convergence of FDIPA-NSOCP is proven considering one simple second-order cone, i.e. $\mathcal{K} = \mathcal{K}^m$. In addition, the updating rules are required to satisfy the following assumptions:

Assumption 3.5. *There exist $\sigma_1, \sigma_2 > 0$ such that*

$$\sigma_1\|d\|^2 \le d^\top B d \le \sigma_2 \|d\|^2, \text{ for all } d \in \mathbb{R}^n.$$

Assumption 3.6. *There exist $c^I, c^S > 0$ such that $c^I e \preceq y^j \preceq c^S e$, for $j = 1, \ldots, J$.*

The following result shows that the linear systems defined in Step 1 always have a unique solution.

Lemma 3.4. *Let B be a positive definite symmetric matrix, $x \in \Omega$ and $y \in \mathrm{int}(\mathcal{K}^m)$. Suppose that $g(x)$ and y operator commute. Then, the matrix*

$$M(x, y, B) = \begin{pmatrix} B & -\nabla g(x)^\top \\ \mathrm{Arw}(y)\nabla g(x) & \mathrm{Arw}(g(x)) \end{pmatrix},$$

is nonsingular.

Proof. To prove that $M(x, y, B)$ is nonsingular, it is enough to show that the homogeneous system

$$\begin{pmatrix} B & -\nabla g(x)^\top \\ \operatorname{Arw}(y)\nabla g(x) & \operatorname{Arw}(g(x)) \end{pmatrix} \begin{pmatrix} d \\ \hat{y} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3.15}$$

has the unique solution $(d, \hat{y}) = (0, 0)$. It follows from the first equation of (3.15) that

$$d^\top Bd - \hat{y}^\top \nabla g(x)d = 0.$$

Since $\operatorname{Arw}(y) \in \mathcal{S}^m_{++}$, it follows from the second equation of (3.15) that

$$\hat{y}^\top \nabla g(x)d = -\hat{y}^\top \operatorname{Arw}^{-1}(y) \operatorname{Arw}(g(x))\hat{y}.$$

Since $(x, y) \in \Omega \times \operatorname{int}(\mathcal{K}^m)$ and $g(x)$ and y operator commute, from Proposition 2.2, Proposition 2.1(f) and [14, Exercise 7.6.10] it follows that $\operatorname{Arw}(y)^{-1} \operatorname{Arw}(g(x)) \in \mathcal{S}^m_{+}$. From both equalities above we conclude that $d^\top Bd \leq 0$ and therefore $d = 0$, because B is a positive definite symmetric matrix. Since $d = 0$, we obtain from (3.15) that

$$\nabla g(x)^\top \hat{y} = 0, \tag{3.16}$$
$$\operatorname{Arw}(g(x))\hat{y} = 0. \tag{3.17}$$

On the other hand, since $x \in \Omega$, we have the following cases: (i) $g(x) \in \operatorname{int}(\mathcal{K}^m)$, (ii) $g(x) = 0$ and (iii) $g(x) \in \partial\mathcal{K}^m \setminus \{0\}$. If the case (i) holds, then $\operatorname{Arw}(g(x)) \in \mathcal{S}^m_{++}$ and therefore $\hat{y} = 0$ from (3.17). Suppose that case (ii) holds. In this case, the tangent cone is $\mathcal{T}_{\mathcal{K}^m}(g(x)) = \{0\}$. By using Assumption 3.4, we get that the linear application $\nabla g(x)\colon \mathbb{R}^n \to \mathbb{R}^m$ is onto. Thus, (3.16) implies that $\hat{y} = 0$. Finally, suppose that case (iii) holds and that $g(x)$ has the following spectral decomposition $g(x) = \alpha_1 u_1 + \alpha_2 u_2$. Hence, it follows that $\alpha_1 = 0$, and $\alpha_2 > 0$. Additionally,

$$\langle \hat{y}^2, g(x)\rangle = \langle \hat{y} \circ \hat{y}, g(x)\rangle = \langle \hat{y}, g(x) \circ \hat{y}\rangle = 0.$$

This implies that $\hat{y}^2$ and $g(x)$ operator commute (cf. Proposition 2.2), because $\hat{y}^2 \in \mathcal{K}^m$ (see [1, page 17]). From this, it follows that $\hat{y}^2$ has the following form $\hat{y}^2 = \lambda u_1$ with $\lambda > 0$. Then, $\hat{y} = \sqrt{\lambda} u_1$. Note that in that case, the tangent cone is $\mathcal{T}_{\mathcal{K}^m}(g(x)) = \{z : z^\top u_1 = 0\}$ (see [1, page. 30]), therefore, relation (3.4) implies the existence of $z_1, z_2 \in \mathbb{R}^n$, with $z_2 \in \mathcal{T}_{\mathcal{K}^m}(g(x))$ such that

$$z_1^\top \nabla g(x)^\top \hat{y} + z_2^\top \hat{y} = \|\hat{y}\|^2. \tag{3.18}$$

Replacing (3.16) in (3.18) and since $z_2^\top \hat{y} = \sqrt{\lambda}(z_2^\top u_1) = 0$, we get $\hat{y} = 0$. □

Remark 3.5. *It follows from the above lemma that the maps* $(x, y, B) \mapsto (d_a, y_a)$ *and* $(x, y, B) \mapsto (d_b, y_b)$ *given by*

$$\begin{pmatrix} B & -\nabla g(x)^\top \\ \operatorname{Arw}(y)\nabla g(x) & \operatorname{Arw}(g(x)) \end{pmatrix} \begin{pmatrix} d_a & d_b \\ y_a & y_b \end{pmatrix} = \begin{pmatrix} \nabla f(x) & 0 \\ 0 & y \end{pmatrix}, \tag{3.19}$$

are continuous. Hence, (d_a^k, y_a^k) *and* (d_b^k, y_b^k) *of FDIPA-NSOCP are well defined. In addition, from* (3.7), (3.9) *and Assumption 3.5 we have* $\langle d_a, \nabla f(x)\rangle \leq -\sigma_1\|d_a\|^2$.

Remark 3.6. *Following the same analysis of [13], we deduce that the updating rule for* ρ *satisfies*

$$\varphi_0\|d_a\|^2 \leq \rho \leq \varphi\|d_a\|^2, \tag{3.20}$$

for some $\varphi_0 > 0$. *We can also see that the map* $(x, y, B) \mapsto \rho$ *is continuous. In fact, the map* $(x, y, B) \mapsto \varphi\|d_a\|^2$ *is continuous everywhere, and* ρ *is taken as* $\rho = (\xi - 1)\langle d_a, \nabla f(x)\rangle / \langle d_b, \nabla f(x)\rangle$ *only at points where* $\langle d_b, \nabla f(x)\rangle > 0$ *and* $(\xi - 1)\langle d_a, \nabla f(x)\rangle / \langle d_b, \nabla f(x)\rangle \leq \varphi\|d_a\|^2$, *i.e., at points* (x, y, B) *where*

$$\langle d_b, \nabla f(x)\rangle \geq \frac{(\xi - 1)\langle d_a, \nabla f(x)\rangle}{\varphi\|d_a\|^2} \geq \frac{-(\xi - 1)\sigma_1}{\varphi} > 0.$$

Lemma 3.7. *The map* $(x, y, B) \mapsto (d, \hat{y})$ *of Step 1(iv) of the algorithm FDIPA-NSOCP is continuous. In addition, the search direction* d *constitutes a continuous descent direction field satisfying* $\langle d, \nabla f(x)\rangle \leq \xi\langle d_a, \nabla f(x)\rangle \leq -\xi\sigma_1\|d_a\|^2$ *at each point* $x \in \Omega$.

Proof. By Remarks 3.5 and 3.6 the map $(x, y, B) \mapsto (d, \hat{y})$ is continuous since $d = d_a + \rho d_b$ and $\hat{y} = y_a + \rho y_b$. It follows from **Step 1**(iii) of FDIPA-NSOCP that

$$\langle d, \nabla f(x)\rangle \leq \begin{cases} \xi\langle d_a, \nabla f(x)\rangle, & \text{if } \langle d_b, \nabla f(x)\rangle > 0, \\ \langle d_a, \nabla f(x)\rangle, & \text{otherwise.} \end{cases}$$

Then, $\langle d, \nabla f(x)\rangle \leq \xi\langle d_a, \nabla f(x)\rangle \leq -\xi\sigma_1\|d_a\|^2$ at each point $x \in \Omega$ (cf. Remark 3.5). Hence d is a descent direction at x. □

The following result extends the given in [13, Proposition 4.1]:

Lemma 3.8. *Let* $\Phi\colon \mathbb{R}^n \to \mathbb{R}^m$ *be a continuously differentiable function. Suppose that each gradient* $\nabla\Phi_i\colon \mathbb{R}^n \to \mathbb{R}^n$ *is Lipschitz continuous in the convex open set* $U \subset \mathbb{R}^n$. *Then, if* $x, y \in U$, *there exists* $\kappa > 0$ *such that*

$$\langle \Phi(y) - \Phi(x) - \nabla\Phi(x)(y - x), z\rangle \geq -\kappa\|y - x\|^2\|z\|, \quad \forall z \in \mathbb{R}^m.$$

Proof. By using the multivariate mean value theorem, we get

$$\Phi(y) = \Phi(x) + A(y - x),$$

where the i-th row of $A \in \mathbb{R}^{m\times n}$ is given by $\nabla\Phi_i(x + \xi_i(y - x))$ for some $\xi_i \in (0, 1)$. Then, for all $z \in \mathbb{R}^m$ we have

$$\begin{aligned}
\langle \Phi(y) - \Phi(x) - \nabla\Phi(x)(y - x), z\rangle &\geq -\|(A - \nabla\Phi(x))(y - x)\|\|z\| \\
&\geq -\|A - \nabla\Phi(x)\|\|y - x\|\|z\| \\
&\geq -\sum_{i=1}^{m} \|\nabla\Phi_i(x + \xi_i(y - x)) - \nabla\Phi_i(x)\|\|y - x\|\|z\| \\
&\geq -\sum_{i=1}^{m} L_i\xi_i\|y - x\|^2\|z\|,
\end{aligned}$$

where we have used the Cauchy-Schwarz inequality in the first inequality and the Lipschitz condition (with $L_i > 0$ the Lipschitz constant) in the fourth one. Taking $\kappa = \sum_{i=1}^{m} L_i\xi_i$, the result follows. □

The following result show that the search direction of the present algorithm constitutes an uniformly feasible directions field.

Lemma 3.9. *There exists a map $(x, y, B) \mapsto \tau$, which is continuous at each point (x, y, B) where $d \neq 0$, and satisfies $\tau > 0$ and both conditions in* **Step 2** *of algorithm FDIPA-NSOCP for any $t \in [0, \tau]$.*

Proof. The existence of $\tau_f > 0$ such that the first condition holds for any $t \in [0, \tau_f]$ is proven in [13, Lemma 4.4]. For the second condition, we will prove that for any (x, y, B) there exists $\tau_g(x, y, B) > 0$ such that

$$\langle g(x + td), z\rangle > 0, \quad \forall t \in [0, \tau_g], \quad \forall z \in \mathcal{K}^m, \ \|z\| = 1. \tag{3.21}$$

Since Assumption 3.3 holds, it follows from Lemma 3.8 that there exists $\kappa > 0$ such that

$$\langle g(x + td), z\rangle \geq \langle g(x), z\rangle + t\langle \nabla g(x)d, z\rangle - \kappa t^2 \|d\|^2, \quad \forall z \in \mathcal{K}^m, \|z\| = 1\,.$$

Then, for any point (x, y, B) such that $d_a \neq 0$ we can define $\tau_g(x, y, B)$ as

$$\tau_g := \inf_{z \in \mathcal{K}^m, \|z\|=1} \frac{\langle \nabla g(x)d, z\rangle + \sqrt{\langle \nabla g(x)d, z\rangle^2 + 4\kappa \|d\|^2 \langle g(x), z\rangle}}{\kappa \|d\|^2}. \tag{3.22}$$

Note that the infimum is positive, since the expression is continuous with respect to z, it is positive when $\langle g(x), z\rangle > 0$ and when $\langle g(x), z\rangle = 0$ we have $g(x) = \lambda_2(g(x))u_2 \in \partial\mathcal{K}^m$, $z = u_1 \in \mathcal{K}^m$, and from (3.11) and Remark 3.6 we have $\langle \nabla g(x)d, z\rangle \geq \rho/2 \geq \varphi_0 \|d_a\|^2$. □

Remark 3.10. *Taking into account Lemmas 3.7 and 3.9, we deduce that the line search of* **Step 2** *in Algorithm (FDIPA-NSOCP) completes successfully. Hence, the algorithm FDIPA-NSOCP is well defined.*

Theorem 3.11. *(Global convergence of Algorithm FDIPA-NSOCP) Let $x^0 \in \Omega_a$. Then, any accumulation point of the sequence $\{x^k\}_{k\in\mathbb{N}}$ is a stationary point of Problem (NSOCP).*

Proof. By Remark 3.10 a sequence $\{x^k\}_{k\in\mathbb{N}}$ is computed by Algorithm (FDIPA-NSOCP), satisfying both conditions in **Step 2** of Algorithm (FDIPA-NSOCP), so that $\{x^k\}_{k\in\mathbb{N}} \subset \Omega$ and $f(x^{k+1}) \leq f(x^k)$ for each $k \in \mathbb{N}$. Then $\{x^k\}_{k\in\mathbb{N}} \subset \Omega_a$ and since Ω_a is compact, $\{x^k\}_{k\in\mathbb{N}}$ has an accumulation point $x^* \in \Omega_a$.

By Assumptions 3.5 and 3.6, $\{B^k\}_{k\in\mathbb{N}}$ and $\{y^k\}_{k\in\mathbb{N}}$ are bounded. Then, by Remark 3.5 $\{d_a^k\}_{k\in\mathbb{N}}$, $\{d_b^k\}_{k\in\mathbb{N}}$, $\{y_a^k\}_{k\in\mathbb{N}}$ and $\{y_b^k\}_{k\in\mathbb{N}}$ are bounded. Since $1 \geq t^k \geq 0$ the sequence $\{t^k\}_{k\in\mathbb{N}}$ is also bounded. Then, the sequence $\{(x^k, B^k, y^k, d_a^k, d_b^k, y_a^k, y_b^k, t^k)\}_{k\in\mathbb{N}}$ is bounded and has an accumulation point $(x^*, B^*, y^*, d_a^*, d_b^*, y_a^*, y_b^*, t^*)$. Let $\{(x^{k_i}, B^{k_i}, y^{k_i}, d_a^{k_i}, d_b^{k_i}, y_a^{k_i}, y_b^{k_i}, t^{k_i})\}_{i\in\mathbb{N}}$ be a convergent subsequence. If $d_a^* = 0$, then taking the limit of (3.13) as $k_i \to \infty$, we have that x^* is a stationary point for the Lagrange multiplier y_a^*. The other case will imply a contradiction. In effect, if $d_a^* \neq 0$, then by Lemma 3.7 the limit d^* of $\{d^{k_i}\}_{i\in\mathbb{N}}$ satisfies $\langle d^*, \nabla f(x^*)\rangle \leq -\xi\sigma_1 \|d_a^*\|^2 < 0$, so that $d^* \neq 0$. By Lemma 3.9 we have $t^* > 0$ (a step size $t^{k_i} \geq \hat{\tau}$ will be accepted in the line search for any $0 < \hat{\tau} < \nu\tau(x^*, y^*, B^*)$ and every large enough k_i). However, since $f(x^k)_{k\in\mathbb{N}}$ is bounded and monotone, it converges to $f(x^*)$. Then

$$\begin{aligned} f(x^*) &= \lim_{i\to\infty} f(x^{k_i+1}) \\ &\leq \lim_{i\to\infty} f(x^{k_i}) + t^{k_i}\eta\nabla f(x^{k_i})^\top d^{k_i} \\ &= f(x^*) + t^*\eta\nabla f(x^*)^\top d^* \\ &\leq f(x^*) - t^*\eta\xi\sigma_i \|d_a^*\|^2\,, \end{aligned}$$

which constitutes a contradiction. □

3.3 Problems with equality constraints

Let us consider now the (NSOCP) problem with nonlinear equality constraints:

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ \text{s.t.} & g^j(x) \succeq_{\mathcal{K}^{m_j}} 0, \; j=1,\dots,J, \\ & h(x)=0, \end{cases} \tag{NSOCP}$$

where $h:\mathbb{R}^m \to \mathbb{R}^p$ is a differentiable nonlinear function. In this case Algorithm FDIPA-NSOCP can be modified following the ideas presented in [13, Section 6]. The main modifications are related to:

1. The feasible set Ω, which is now defined as

$$\Omega := \{x \in \mathbb{R}^n \;:\; g^j(x) \in \mathcal{K}^{m_j}, \; j=1,\dots,J, \quad h(x) \geq 0\}.$$

2. The potential function: since a given point in Ω could be infeasible with respect to the equality constraints, its objective value could actually be lower than the optimal value attained in the set of the fully feasible points. The algorithm should then be able to increase the objective function when necessary and a potential function must be used for monitoring the convergence. Following [13], the potential function is defined as:

$$\phi_c(x) := f(x) + \sum_{j=1}^{p} c_i |h_i(x)|.$$

where $c \in \mathbb{R}^p$ is a vector of parameters.

3. The search direction: in this case the Lagrangian $L\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ and the optimality conditions are given by

$$L(x,\mathbf{y},\mu) = f(x) - \sum_{j=1}^{J} \langle g^j(x), y^j\rangle - \langle h(x), \mu\rangle,$$

and

$$\nabla_x L(x,\mathbf{y},\mu) = \nabla f(x) - \sum_{j=1}^{J} \nabla g^j(x)^\top y^j - \nabla h(x)^\top \mu = 0, \tag{3.23}$$

$$\langle g^j(x), y^j\rangle = 0, \quad j=1,\dots,J, \tag{3.24}$$

$$g^j(x),\; y^j \in \mathcal{K}^{m_j}, \quad j=1,\dots,J, \tag{3.25}$$

$$h(x) = 0. \tag{3.26}$$

So that the quasi-Newton direction and the direction of deflection are

$$\begin{pmatrix} B & -\nabla \mathbf{g}(x)^\top & -\nabla h(x)^\top \\ \mathrm{Arw}(y)\nabla \mathbf{g}(x) & \mathrm{Arw}(\mathbf{g}(x)) & 0 \\ \nabla h(x) & 0 & 0 \end{pmatrix} \begin{pmatrix} d_a & d_b \\ \mathbf{y}_a & \mathbf{y}_b \\ \mu_a & \mu_b \end{pmatrix} = \begin{pmatrix} \nabla f(x) & 0 \\ 0 & \mathbf{y} \\ -h(x) & \mathbf{1} \end{pmatrix}. \tag{3.27}$$

In consequence, based on this ideas the algorithm is the following:

Algorithm FDIPA-NSOCP: Choose the parameters $\xi, \eta, \nu \in (0,1)$ and $\varphi > 0$.

Step 0: Start with initial $x^0 \in \text{int}(\Omega_a)$, $\mathbf{y}^0 \in \text{int}(\mathcal{K})$ such that operator commutes with $\mathbf{g}(x^0)$, $B^0 \in \mathcal{S}^n_{++}$, $\mu^0 \in \mathbb{R}^p$ and $c > 0$. Set $k = 0$.

Step 1: Computation of the search direction.

(*i*) Compute d_a^k, $\mathbf{y}_a^k$ and μ_a^k by solving the linear system

$$\begin{pmatrix} B^k & -\nabla \mathbf{g}(x^k)^\top & -\nabla h(x^k) \\ \text{Arw}(\mathbf{y}^k)\nabla \mathbf{g}(x^k) & \text{Arw}(\mathbf{g}(x^k)) & 0 \\ \nabla h(x^k) & 0 & 0 \end{pmatrix} \begin{pmatrix} d_a^k \\ \mathbf{y}_a^k \\ \mu_a^k \end{pmatrix} = \begin{pmatrix} -\nabla f(x^k) \\ 0 \\ -h(x^k) \end{pmatrix}.$$

If $d_a^k = 0$, stop.

(*ii*) Compute d_b^k and $\mathbf{y}_b^k$ by solving the linear system

$$\begin{pmatrix} B^k & -\nabla \mathbf{g}(x^k)^\top & -\nabla h(x^k) \\ \text{Arw}(\mathbf{y}^k)\nabla \mathbf{g}(x^k) & \text{Arw}(\mathbf{g}(x^k)) & 0 \\ \nabla h(x^k) & 0 & 0 \end{pmatrix} \begin{pmatrix} d_b^k \\ \mathbf{y}_b^k \\ \mu_b^k \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{y}^k \\ \mathbf{1} \end{pmatrix}.$$

(*iii*) If $c_i < 1.2(\mu_a^k)_i$, then set $c_i = 2(\mu_a^k)_i$.

(*iv*) If $\langle d_b^k, \nabla \phi_c(x^k)\rangle > 0$, set

$$\rho^k = \min\left\{\varphi \|d_a^k\|^2, (\xi - 1)\frac{\langle d_a^k, \nabla \phi_c(x^k)\rangle}{\langle d_b^k, \nabla \phi_c(x^k)\rangle}\right\}.$$

Otherwise, set

$$\rho^k = \varphi \|d_a^k\|^2.$$

(*v*) Compute

$$d^k = d_a^k + \rho^k d_b^k, \quad \hat{\mathbf{y}}^k = \mathbf{y}_a^k + \rho^k \mathbf{y}_b^k \text{ and } \hat{\mu}^k = \mu_a^k + \rho^k \mu_b^k$$

Step 2: (Armijo line search): compute t^k as the first number of the sequence $\{1, \nu, \nu^2, \ldots\}$ satisfying

$$\begin{aligned} &\phi_c(x^k + t^k d^k) \le \phi_c(x^k) + t^k \eta \nabla \phi_c(x^k)^\top d^k, \text{ and} \\ &\mathbf{g}(x^k + t^k d^k) \succ_{\mathcal{K}} 0. \end{aligned}$$

Step 3: Updates. Set $x^{k+1} = x^k + t^k d^k$. Define $\mathbf{y}^{k+1} \in \text{int}(\mathcal{K})$, such that operator commutes with $\mathbf{g}(x^{k+1})$, and $B^{k+1} \in \mathcal{S}^n_{++}$.

Replace k by $k+1$ and go to Step 1.

4 Implementation and numerical results

4.1 About the numerical implementation

The implementation of updating rules for B_k and y_k satisfying Assumptions 3.5–3.6 (see p. 9) are of main importance to obtain a reasonably efficient algorithm for practical applications. In some specific convex problems the use of the Hessian matrix could both guarantee the validity of Assumption 3.5 as well as quick convergence to the stationary point. However, in general nonlinear problems the Hessian matrix is not positive definite, and computation of second order derivatives is usually too expensive in terms of number of operations and computation time in most engineering applications. In these cases the use of quasi-Newton rules is a standard approach that provides positive definite matrices from the knowledge of just the first order derivatives. For example, the BFGS formula with the Han-Powell modification reads as (see e.g. [21, Section 14.7]):

$$B^{k+1} = B^k - \frac{B^k p^k (p^k)^\top B^k}{(p^k)^\top B^k p^k} + \frac{r^k (r^k)^\top}{(p^k)^\top r^k},$$

where

$$\begin{aligned}
p^k &= x^{k+1} - x^k, \\
r^k &= \theta^k q^k + (1-\theta^k) B^k p^k, \\
q^k &= \nabla^x L(x^{k+1}, y^{k+1}) - \nabla^x L(x^k, y^{k+1}), \\
\theta^k &= \begin{cases} 1 & \text{if } (p^k)^\top q^k \geq (0.2)(p^k)^\top B^k p^k, \\ \dfrac{(0.8)(p^k)^\top B^k p^k}{(p^k)^\top B^k p^k - (p^k)^\top q^k} & \text{if } (p^k)^\top q^k < (0.2)(p^k)^\top B^k p^k. \end{cases}
\end{aligned}$$

The initial value $B^0 = I$ can be used. Restarting the method using $B^k = I$ every ℓ steps, with $\ell \leq n$, is necessary since the BFGS rule cannot ensure satisfaction of Assumption 3.5 at all iterations. The first step after restarting can be viewed as a spacer step that ensures convergence of the whole process, see [21, Section 7.10].

In the case of the Lagrange multipliers, the updating rule must provide the multiplier y^{k+1} such that operator commutes with $g(x^{k+1})$ and satisfies Assumption 3.6. The main idea here is to modify the solution y_a^k as less as possible. Let $\lambda_1 u_1 + \lambda_2 u_2$ be the spectral decomposition of $g(x^{k+1})$. The multiplier y^{k+1} operator commutes with $g(x^{k+1})$ if and only if it belongs to the linear subspace generated by u_1 and u_2. If we call $\tilde{y}$ to the projection of y_a^k in this subspace, then its spectral decomposition $\tilde{y} = \tilde{y}_1 u_1 + \tilde{y}_2 u_2$ satisfy the inequalities $c^I e \succeq \tilde{y} \succeq c^S e$ if and only if $c^I \leq \tilde{y}_i \leq c^S$, $i = 1, 2$. If this last inequalities are not satisfied by $\tilde{y}$, then we proceed to project $\tilde{y}$ into the convex region $\{y : c^I e \succeq y \succeq c^S e\}$. The proposed updating rules are:

$$y^{k+1} := y_1^{k+1} u_1 + y_2^{k+1} u_2,$$

where

$$\begin{aligned}
y_i^{k+1} &:= \max\{c^I, \min\{c^S, \tilde{y}_i\}\} \quad i = 1, 2, \\
\tilde{y}_i &:= 2\langle y_a^k, u_i \rangle \quad i = 1, 2.
\end{aligned}$$

4.2 Numerical results

In this section we discuss some numerical results on specific instance of class of SOCP: robust classification by hyperplanes under data uncertainty (see formulation (4.3) below), and a nonlinear convex SOCP taken from [12] (see formulation (4.4) below). Our goal is to show how our algorithm FDIPA-NSOCP works in practice and verify the theoretical results obtained in the last section. The first formulation was chosen because it can be formulated as a linear SOCP. This allows us to compare our results with those obtained by SeDuMi 1.21 toolbox for MATLAB, which implements a primal-dual interior point method for solving LSOCPs (see [27]). The second one shows the behavior of the algorithm when we use a highly nonlinear differentiable objective function.

The algorithm was implemented in MATLAB 7.8, Release 2009b and the numerical experiments were performed on a personal computer with an Intel Core 2 Duo CPU 2.20GHz processor and 4GB of RAM, running Microsoft Windows 7 Home Premium.

The set of values of the parameters in our experiments were taken as follows:

$$\varphi = 1, \quad \xi = 0.7, \quad \eta = 0.5, \quad \nu = 0.7, \quad c^I = 10^{-9}, \quad c^S = 10^9,$$

and the stopping rule as $\|d_a^k\| \leq 10^{-6}$.

4.2.1 Support vector machines under uncertainty

Let us consider the following general binary classification problem: from some training data points in $\mathbb{R}^n$, each of which belongs to one of two classes, the goal is to determine some way of deciding which class new data points will be in. Suppose that the training data consists of two sets of points whose elements are labeled by either 1 or -1 to indicate the class they belong to. If there exists a strictly separating $(n-1)$-dimensional hyperplane between the two data sets, namely $H(w,b) = \{\mathbf{x} \in \mathbb{R}^n : w^\top \mathbf{x} - b = 0\}$, then the standard Support Vector Machine (SVM) approach is based on constructing a *linear classifier* according to the function $f(x) = \operatorname{sgn}(w^\top \mathbf{x} - b)$, where $\operatorname{sgn}(\cdot)$ denotes the sign function. As there might be many hyperplanes that classify the data, in order to minimize misclassification one picks the hyperplane which maximizes the separation (margin) between the two classes, so that the distance from the hyperplane to the nearest data point is maximized. In fact, if we have a set $\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ of m training data points in $\mathbb{R}^n \times \{-1, 1\}$, the *maximum-margin* hyperplane problem can be formulated as the following Quadratic Programming (QP) optimization problem [7]:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &\text{s.t. } y_i(w^\top \mathbf{x}_i - b) \geq 1, \quad i = 1, \ldots, m. \end{aligned} \tag{4.1}$$

If this problem is feasible then we say that the training data set $\mathcal{T}$ is *linearly separable*. The linear equations $w^\top \mathbf{x} - b = 1$ and $w^\top \mathbf{x} - b = -1$ describe the so-called *supporting* hyperplanes.

Suppose that $\mathbf{X}_1$ and $\mathbf{X}_2$ are random vector variables that generate samples of the positive and negative classes respectively. In order to construct a maximum margin linear classifier such that the false-negative and false-positive error rates do not exceed $\eta_1 \in (0,1)$ and $\eta_2 \in (0,1)$ respectively, Nath and Bhattacharyya [25] suggested consider the following Quadratic

Chance-Constrained Programming (QCCP) problem:

$$\begin{aligned}
&\min_{w,b} \frac{1}{2}\|w\|^2 \\
&\text{s.t. } \operatorname{Prob}\{w^\top \mathbf{X}_1 - b < 0\} \le \eta_1, \\
&\qquad \operatorname{Prob}\{w^\top \mathbf{X}_2 - b > 0\} \le \eta_2.
\end{aligned} \tag{4.2}$$

In other words, we require that the random variable $\mathbf{X}_i$ lies on the correct side of the hyperplane with probability greater than $1 - \eta_i$ for $i = 1, 2$. Assume that for $i = 1, 2$ we *only know* the mean $\mu_i \in \mathbb{R}^n$ and covariance matrix $\Sigma_i \in \mathbb{R}^{n \times n}$ of the random vector $\mathbf{X}_i$. In this case, for each $i = 1, 2$ we want to be able to classify correctly, up to the rate η_i, even for the *worst distribution* in the class of distributions which have common mean and covariance, i.e. $\mathbf{X}_i \sim (\mu_i, \Sigma_i)$, replacing the probability constraints in (4.2) with their *robust* counterparts

$$\sup_{\mathbf{X}_1 \sim (\mu_1, \Sigma_1)} \operatorname{Prob}\{w^\top \mathbf{X}_1 - b \le -1\} \le \eta_1, \qquad \sup_{\mathbf{X}_2 \sim (\mu_2, \Sigma_2)} \operatorname{Prob}\{w^\top \mathbf{X}_2 - b \ge 1\} \le \eta_2.$$

Thanks to an appropriate application of the multivariate Chebyshev inequality [17, Lemma 1], this worst distribution approach leads to the following QSOCP, which is a deterministic formulation of (4.2) (see [25] for all details):

$$\begin{aligned}
&\min_{w,b} \frac{1}{2}\|w\|^2 \\
&\text{s.t. } w^\top \mu_1 - b \ge 1 + \kappa_1 \|S_1^\top w\|, \\
&\qquad b - w^\top \mu_2 \ge 1 + \kappa_2 \|S_2^\top w\|,
\end{aligned} \tag{4.3}$$

where $\Sigma_i = S_i S_i^\top$ (for instance, Cholesky factorization) for $i = 1, 2$, and η_i and κ_i are related via the formula $\kappa_i = \sqrt{\frac{1-\eta_i}{\eta_i}}$.

The numerical algorithm presented here requires an initial feasible point. This point can be found depending on the nature of the problem. In order to obtain an initial point of formulation (4.3) for applied FDIPA-NSOCP algorithm, we solve the following auxiliary conic optimization problem:

$$\begin{aligned}
&\min_{w,b,\zeta} \zeta \\
&\text{s.t. } w^\top \mu_1 - b \ge 1 - \zeta + \kappa_1 \|S_1^\top w\|, \\
&\qquad b - w^\top \mu_2 \ge 1 - \zeta + \kappa_2 \|S_2^\top w\|, \\
&\qquad -\zeta + \zeta^* \ge 0,
\end{aligned}$$

where $\zeta^* \in \mathbb{R}$ is a constant. The reasoning behind this choice is based on [24, Example 6.1].

Next, we describe three benchmark data sets that will be used in order to solve numerically the formulation (4.3). More information on these data sets can be found in the UCI Repository [3].

- **Wisconsin Breast Cancer (WBC)**: This data set contains $m = 569$ observations of tissue samples (212 diagnosed as malignant and 357 diagnosed as benign tumors) described by $n = 30$ continuous features, computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image, such as the perimeter, the area, the symmetry, and the number of concave portions of the contour.

- **Pima Indians Diabetes (DIA)**: The Pima Indians Diabetes data set presents $n = 8$ features and $m = 768$ instances (500 tested negative for diabetes and 268 tested positive). All patients are females at least 21 years old of Pima Indian heritage. The features include age, number of times pregnant, diastolic blood pressure and body mass index, among others.

- **German Credit (GC)**: This data set presents $m = 1000$ granted loans, 700 good and 300 bad payers in terms of repayment, described by $n = 24$ attributes. The variables include loan information (amount, duration, and purpose), credit history, personal information (sex, marital status, number of years in present employment) and other variables to assess financial capacity and willingness to pay (properties, telephone, among others).

Tables 1, 2 and 3 report the results of our experiments and provide some comparisons with SeDuMi. In these tables, the first and second columns show the error rates, the third and fifth columns show the number of iterations when B^k is the identity matrix I and B^k use the BFGS rule, respectively, the fourth and sixth columns report the CPU time by using our implementation in MATLAB, the seventh and eighth columns provide the value of the objective function at the output solution obtained by FDIPA-NSOCP algorithm, and the optimal value given by SeDuMi. Finally, the last column shows the CPU time required by SeDuMi toolbox using its default configuration.

Table 1: Numerical comparisons with SeDuMi applied to Wisconsin Breast cancer data set: Benigno vs Malignant.

		$B = I$		B =approx. Hessian				
η_1	η_2	# iter.	CPU Time	# iter.	CPU Time	val_{fdipa}	val_{sdm}	CPU time SeDuMi
0.1	0.9	51	11".18	23	05".96	32.995793	32.995793	1".04
0.1	0.7	50	12".09	21	05".69	115.094729	115.094729	1".23
0.3	0.7	134	38".18	20	05".11	14.741665	14.741665	0".79
0.5	0.7	107	26".31	20	05".16	8.903124	8.903124	0".90

Table 2: Numerical comparisons with SeDuMi applied to Pima Indian Diabetes data set.

		$B = I$		B =approx. Hessian				
η_1	η_2	# iter.	CPU Time	# iter.	CPU Time	val_{fdipa}	val_{sdm}	CPU time SeDuMi
0.9	0.9	22	07".81	31	11".83	169.389431	169.389431	1".04
0.9	0.8	21	07".50	30	10".96	302.246324	302.246323	0".82
0.9	0.7	19	06".54	19	06".83	608.031244	608.031242	0".93
0.7	0.9	22	07".89	22	07".79	619.895090	619.895087	1".23

Table 3: Numerical comparisons with SeDuMi applied to German Credit data set.

		$B = I$		B =approx. Hessian				
η_1	η_2	# iter.	CPU Time	# iter.	CPU Time	val_{fdipa}	val_{sdm}	CPU time SeDuMi
0.6	0.9	43	28".45	19	12".01	150.850277	150.850277	0".49
0.7	0.8	37	25".00	19	16".13	232.590326	232.590326	0".33
0.8	0.8	49	32".34	20	12".37	60.153169	60.153169	0".45
0.9	0.7	50	34".60	17	10".57	64.901024	64.901024	0".47

4.2.2 Nonlinear convex SOCP

Let us consider the following nonlinear convex SOCP [12]:

$$\begin{aligned} &\min_z \exp(z_1 - z_3) + 3(2z_1 - z_2)^4 + \sqrt{1 + (3z_2 + 5z_3)^2} \\ &\text{s.t.} \begin{pmatrix} 4 & 6 & 3 \\ -1 & 7 & -5 \end{pmatrix} z + \begin{pmatrix} -1 \\ 2 \end{pmatrix} \in \mathcal{K}^2, \quad z \in \mathcal{K}^3. \end{aligned} \tag{4.4}$$

In the following Table we list the numerical results obtained when we apply FDIPA Algorithm for solving this test problem from several starting points (see [24]). In this table, the first column provide the starting point, the second and fifth columns show the number of iterations when B^k is the identity matrix I and B^k use the BFGS rule, respectively, the third and sixth columns provide the value of the objective function at the output solution obtained by FDIPA-NSOCP algorithm, and the fourth and eighth columns report the CPU time by using our implementation in MATLAB.

Table 4: Numerical results of Problem (4.4).

	$B = I$			B = aprox. Hessian		
z^0	# iter.	f_{end}	CPU Time	# iter.	f_{end}	CPU Time
$(1.8860, -0.1890, -0.4081)^\top$	25	2.597575	0.27"	21	2.597577	0.35"
$(4.3425, 0.0875, -0.2332)^\top$	32	2.597575	0.61"	28	2.597575	0.44"
$(4.6972, -0.4294, -1.3931)^\top$	31	2.597575	0.65"	38	2.597575	0.84"
$(3.2266, -0.7353, -1.5477)^\top$	31	2.597575	0.17"	29	2.597575	0.37"
$(3.7282, 0.2875, 0.2737)^\top$	30	2.597575	0.59"	28	2.597575	0.37"

5 Concluding remarks

We have proposed an algorithm to solve smooth nonlinear SOCP problems. Then, we have obtained theoretical results about convergence. Finally, we have applied our method to robust classification problems obtaining encouraging numerical results.

Acknowledgements

This research was supported by CONICYT Anillo ACT1106 (Julio López) and FONDECYT Grants 1130905 (Miguel Carrasco), 11110188 (Julio López). Alfredo Canelas thanks the Uruguayan Councils ANII and CSIC for the financial support.

References

[1] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Math. Program.*, 95(1, Ser. B):3–51, 2003.

[2] F. Alvarez, J. López, and H. Ramírez C. Interior proximal algorithm with variable metric for second-order cone programming: applications to structural optimization and support vector machines. *Optim. Methods Softw.*, 25(4-6):859–881, 2010.

[3] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[4] J.F. Bonnans and H. Ramírez C. Perturbation analysis of second-order cone programming problems. *Math. Program.*, 104(2):205–227, 2005.

[5] J.F. Bonnans and A. Shapiro. *Perturbation analysis of optimization problems.* Springer Series in Operations Research. Springer-Verlag, New York, 2000.

[6] J.S. Chen, X. Chen, and P. Tseng. Analysis of nonsmooth vector-valued functions associated with second-order cones. *Math. Program.*, 101(1):95–117, 2004.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[8] J. Faraut and A. Korányi. *Analysis on symmetric cones.* Oxford Mathematical Monographs. The Clarendon Press Oxford University Press, New York, 1994. Oxford Science Publications.

[9] E.H. Fukuda, P.J.S. Silva, and M. Fukushima. Differentiable exact penalty functions for nonlinear second-order cone programs. *SIAM J. Optim.*, 22(4):1607–1633, 2012.

[10] M. Fukushima, Z.Q. Luo, and P. Tseng. Smoothing functions for second-order-cone complementarity problems. *SIAM J. Optim.*, 12(2):436–460, 2001/02.

[11] J. Ch. Gilbert. Élements d'Optimisation Différentiable - Théorie et Algorithmes. Syllabus de cours à I'ENSTA, Paris. Avaliable at http://www-rocq.inria.fr/∼gilbert/ensta/optim.html, 2012.

[12] S. Hayashi, N. Yamashita, and M. Fukushima. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM J. Optim.*, 15(2):593–615, 2005.

[13] J. Herskovits. Feasible direction interior-point technique for nonlinear optimization. *J. Optim. Theory Appl.*, 99(1):121–146, 1998.

[14] R.A. Horn and C.R. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

[15] C. Kanzow, I. Ferenczi, and M. Fukushima. On the local convergence of semismooth newton methods for linear and nonlinear second-order cone programs without strict complementarity. *SIAM J. Optim.*, 20(1):297–320, 2009.

[16] H. Kato and M. Fukushima. An sqp-type algorithm for nonlinear second-order cone programs. *Optim. Lett.*, 1(2):129–144, 2007.

[17] G. Lanckriet, L.E. Ghaoui, C. Bhattacharyya, and M. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2003.

[18] Y.J. Liu and L.W. Zhang. Convergence analysis of the augmented lagrangian method for nonlinear second-order cone optimization problems. *Nonlinear Anal.*, 67(5):1359 – 1373, 2007.

[19] Y.J. Liu and L.W. Zhang. Convergence of the augmented lagrangian method for nonlinear optimization problems over second-order cones. *J. Optim. Theory Appl.*, 139(3):557–575, 2008.

[20] M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra Appl.*, 284(1-3):193–228, 1998.

[21] David G. Luenberger. *Linear and nonlinear programming*. Kluwer Academic Publishers, Boston, MA, second edition, 2003.

[22] R.D.C. Monteiro and T. Tsuchiya. Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions. *Math. Program.*, 88(1, Ser. A):61–83, 2000.

[23] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[24] S. Pan and J.S. Chen. A class of interior proximal-like algorithms for convex second-order cone programming. *SIAM J. Optim.*, 19(2):883–910, 2008.

[25] J. Saketha N. and C. Bhattacharyya. Maximum margin classifiers with specified false positive and false negative error rates. In *Proceedings of the Seventh SIAM International Conference on Data Mining*, pages 26–28, Minneapolis, Minnesota, April 2007. Available at http://www.siam.org/meetings/proceedings/2007/datamining/papers/004Jagarlapudi.pdf.

[26] P.K. Shivaswamy, C. Bhattacharyya, and A.J. Smola. Second order cone programming approaches for handling missing and uncertain data. *J. Mach. Learn. Res.*, 7:1283–1314, 2006.

[27] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11/12(1-4):625–653, 1999.

[28] T. Tsuchiya. A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming. *Optim. Methods Softw.*, 11/12(1-4):141–182, 1999.

[29] H. Yamashita and H. Yabe. A primal-dual interior point method for nonlinear optimization over second-order cones. *Optim. Methods Softw.*, 24(3):407–426, 2009.