Computational Optimization and Applications (2019) 72:363–390
https://doi.org/10.1007/s10589-018-0045-8

A primal-dual interior-point method based on various selections of displacement step for symmetric optimization

Baha Alzalg[1,2]

Received: 12 June 2017 / Published online: 8 November 2018

Abstract
In this paper, we develop a primal-dual central trajectory interior-point algorithm for symmetric programming problems and establish its complexity analysis. The main contribution of the paper is that it uniquely equips the central trajectory algorithm with various selections of the displacement step while solving symmetric programming. To show the efficiency of the proposed algorithm, these selections of calculating the displacement step are compared in numerical examples for second-order cone programming, which is a special case of symmetric programming.

Keywords Symmetric programming · Interior-point methods · Primal-dual methods · Central trajectory methods · Jordan algebras

Mathematics Subject Classification 90C30 · 90C46 · 90C51 · 17A15

1 Introduction

Symmetric programming problems [1,2] are convex optimization problems in which we optimize a linear objective function over the intersection of an affine linear manifold with the Cartesian product of symmetric cones. Linear programming, second-order cone programming [3,4] and semidefinite programming [5,6] are well known and important special cases of symmetric programs. Interior-point methods are considered one of the effective methods developed for general symmetric programming problems (see for example [1,2,7–9]). For certain classes of optimization problems over symmetric cones complete interior point methods were developed. See, for example, [10–13] for linear programming, [3,4,14–18] for second-order cone programming, and [5,6,19–22] for semidefinite programming.

✉ Baha Alzalg
b.alzalg@ju.edu.jo

1 Department of Mathematics, The University of Jordan, Amman 11942, Jordan

2 School of Mathematical Sciences, Rochester Institute of Technology, Rochester, NY 14623, USA

Interior-point methods consist of two elements: namely, calculating a step or search direction that places one in the direction of a central path, and calculating a displacement step-size in order to enforce some sort of feasibility (more specifically, the positive-definiteness) of the subsequent iterate. Throughout this paper, we use the term "displacement step" instead of "displacement step-size" for the sake of simplicity. In fact, the main challenge to be faced in establishing an iteration in an interior-point algorithm is that the determination and computation of the displacement step along the search direction. Calculating the displacement step using classical line search methods is undesirable and even generally impossible [20,23]. Touil et al. [23] developed an interior-point algorithm for semidefinite programming and presented some strategies for determining the displacement step for the developed algorithm. Extensions of interior-point methods for semidefinite programming to general symmetric cones can be found in the literature (see for example [14–18]), but only in terms of calculating the search direction, not preserving geometric feasibility conditions thereafter. This paper presents some strategies for determining the displacement step for a particular interior-point method on general symmetric cones. So, our focus in this paper is not on the search direction, but on the displacement step. In other words, we are not developing a new step, but a new step size.

We can separate two types of interior-point methods: projective methods [10–12,24,25] and (feasible or infeasible) central trajectory methods [19,23,26,27]. The interior-point method developed by Touil et al. [23] for semidefinite programming is a feasible central trajectory algorithm. The focus in this paper is on solving symmetric programming by feasible central trajectory methods. More specifically, this work exploits the Jordan algebraic structure of the symmetric cone to derive a feasible primal-dual central trajectory algorithm for symmetric programming and to present some strategies for determining the displacement step for the proposed algorithm by extending the work of Touil et al. [23] for the semidefinite case to the general symmetric case.

In this paper, we associate a perturbed problem to symmetric programming problem and apply Newton's method to treat the corresponding perturbed equations in order to obtain a descent search direction. The main contribution of this paper is that it uniquely proposes four different selections to calculate the displacement step along the search direction while solving symmetric programming. Our work is not a trivial generalization of that in [23], however, because a deep understanding of the Euclidean Jordan algebras is very important to establish the analysis of this paper. The main difficulties in developing our analysis are caused by establishing and proving the selections of determining the displacement step in the framework of Euclidean Jordan algebras.

There are widely available software packages than can handle symmetric programs in general, and second-order cone programs in particular. Optimization solvers such as MOSEK [28] and SDPT3 [29] are based on the interior-point methods and are well suited for symmetric programming problems. The efficiency of our proposed algorithms is shown by presenting numerical experiments on second-order cone programs comparing with SDPT3 and MOSEK.

This paper is organized as follows. In Sect. 2, we briefly review the basics of the Euclidean Jordan algebra associated with the symmetric cone and provide some

preliminary results. In Sect. 3, we introduce the problem formulation and make some assumptions, then we apply Newton's method to the perturbed problem. In Sect. 4, we propose four different selections of calculating the appropriate displacement step. Section 5 is devoted to present a feasible primal-dual central trajectory algorithm. The complexity analysis of the proposed algorithm is presented in Sect. 6. In Sect. 7, we compare the four selections of computing the displacement step and show the efficiency of the proposed algorithm in some numerical examples for second-order cone programs. Section 8 contains some concluding remarks.

2 Background and preliminary results

In this section, we provide the necessary background for the subsequent sections. First, we review some statistical inequalities needed for deriving some results in the work. Then, we briefly review the definitions of self-duality, homogeneity and symmetric cones. Finally, we review the basics of the theory of Euclidean Jordan algebras.

2.1 Preliminary statistical inequalities

In this part, we review some statistical inequalities needed for the derivation of some of the results in the paper.

Let $x_1, x_2, \ldots, x_n \in \mathbb{R}$ be a sample of size n, then its *mean* $\bar{x}$ and its *standard deviation* $\sigma_{\boldsymbol{x}}$ are respectively defined as

$$\bar{x} := \frac{1}{n}\sum_{k=1}^{n} x_k \quad \text{and} \quad \sigma_{\boldsymbol{x}}^2 := \frac{1}{n}\sum_{k=1}^{n} x_k^2 - \bar{x}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \bar{x})^2.$$

The first statement in the following proposition is due to [30] and the second statement is due to [20, Theorem 5].

Proposition 2.1 *Assume that* $\boldsymbol{x} \in \mathbb{R}^n$, *then we have*

$$\bar{x} - \sigma_{\boldsymbol{x}}\sqrt{n-1} \le \min_{1\le k\le n} x_k \le \bar{x} - \frac{\sigma_{\boldsymbol{x}}}{\sqrt{n-1}}, \quad \textit{and}$$
$$\bar{x} + \frac{\sigma_{\boldsymbol{x}}}{\sqrt{n-1}} \le \max_{1\le k\le n} x_k \le \bar{x} + \sigma_{\boldsymbol{x}}\sqrt{n-1}.$$

In particular, if $x_k > 0$ *for all* $k = 1, 2, \ldots, n$, *then we also have*

$$n\ln\left(\bar{x} - \sigma_{\boldsymbol{x}}\sqrt{n-1}\right) \le A \le \sum_{k=1}^{n}\ln(x_k) \le B \le n\ln(\bar{x}), \quad \textit{where}$$
$$A = (n-1)\ln\left(\bar{x} + \frac{\sigma_{\boldsymbol{x}}}{\sqrt{n-1}}\right) + \ln\left(\bar{x} - \sigma_{\boldsymbol{x}}\sqrt{n-1}\right), \quad \textit{and}$$
$$B = (n-1)\ln\left(\bar{x} - \frac{\sigma_{\boldsymbol{x}}}{\sqrt{n-1}}\right) + \ln\left(\bar{x} + \sigma_{\boldsymbol{x}}\sqrt{n-1}\right).$$

2.2 Preliminary conical definitions

A cone is said to be *closed* iff it is closed under the taking of sequential limits, *convex* iff it is closed under taking convex combinations, *pointed* iff it does not contain two opposite nonzero vectors (so the origin is an extreme point), *solid* iff it has a nonempty interior, and *regular* iff it is a closed, convex, pointed, solid cone.

Definition 2.1 Let $\mathcal{V}$ be a finite-dimensional Euclidean vector space over $\mathbb{R}$ with an inner product "$\langle \cdot, \cdot \rangle$". The dual cone of a regular cone $\mathcal{K} \subset \mathcal{V}$ is denoted by $\mathcal{K}^\star$ and is defined as

$$\mathcal{K}^\star := \{\boldsymbol{s} \in \mathcal{V} : \langle \boldsymbol{x}, \boldsymbol{s} \rangle \geq 0, \ \forall \boldsymbol{x} \in \mathcal{K}\}.$$

A regular cone $\mathcal{K}$ is said to be self-dual if it coincides with its dual cone, i.e., $\mathcal{K} = \mathcal{K}^\star$.

The *general linear group* of degree n over $\mathbb{R}$ is denoted by $GL(n, \mathbb{R})$ and is defined to be the set of all nonsingular matrices of order n with entries from $\mathbb{R}$, together with the operation of ordinary matrix multiplication. For a regular cone $\mathcal{K} \subset \mathcal{V}$, we denote by $\mathrm{Aut}(\mathcal{K})$ the *automorphism group* of $\mathcal{K}$, i.e.,

$$\mathrm{Aut}(\mathcal{K}) := \{\varphi \in GL(n, \mathbb{R}) : \varphi(\mathcal{K}) = \mathcal{K}\}.$$

Definition 2.2 Let $\mathcal{V}$ be a finite-dimensional real Euclidean space. A regular cone $\mathcal{K} \subset \mathcal{V}$ is said to be homogeneous if for each $\boldsymbol{u}, \boldsymbol{v} \in \operatorname{int} \mathcal{K}$, there exists an nonsingular linear map $F : \mathcal{V} \longrightarrow \mathcal{V}$ such that

$$F(\mathcal{K}) = \mathcal{K} \text{ (i.e., } F \text{ is an automorphism of } \mathcal{K}), \quad \text{and} \quad F(\boldsymbol{u}) = \boldsymbol{v}.$$

In other words, $\mathrm{Aut}(\mathcal{K})$ acts transitively on the interior of $\mathcal{K}$.

A regular $\mathcal{K}$ cone is said to be *symmetric* if it is both self-dual and homogeneous.

2.3 Preliminaries of Euclidean Jordan algebras

In this subsection, we review some basic definitions and notions from the theory of Jordan algebras. The text of Faraut and Korányi [31] covers the foundations of this theory. Our presentation in this subsection follows that of [1, Section 2]. In order to make our presentation concrete, we give some examples from the algebra of the second-order cone as an interesting paradigm for understanding the theory of Jordan algebras.

We use "," for adjoining scalars, vectors and matrices in a row, and use ";" for adjoining them in a column. So, for instance, if $\boldsymbol{u}$ and $\boldsymbol{v}$ are vectors, we have

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} = (\boldsymbol{u}^{\mathsf{T}}, \boldsymbol{v}^{\mathsf{T}})^{\mathsf{T}} = (\boldsymbol{u}; \boldsymbol{v}).$$

Let $\mathcal{J}$ be a finite-dimensional vector space over $\mathbb{R}$. A map $\circ : \mathcal{J} \times \mathcal{J} \longrightarrow \mathcal{J}$ is called *bilinear* if $(\alpha \boldsymbol{x} + \beta \boldsymbol{y}) \circ \boldsymbol{z} = \alpha(\boldsymbol{x} \circ \boldsymbol{z}) + \beta(\boldsymbol{y} \circ \boldsymbol{z})$ and $\boldsymbol{x} \circ (\alpha \boldsymbol{y} + \beta \boldsymbol{z}) = \alpha(\boldsymbol{x} \circ \boldsymbol{y}) + \beta(\boldsymbol{x} \circ \boldsymbol{z})$ for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{J}$ and $\alpha, \beta \in \mathbb{R}$. The vector space $\mathcal{J}$ over $\mathbb{R}$ is called an *algebra* over $\mathbb{R}$ if a bilinear map $\circ : \mathcal{J} \times \mathcal{J} \longrightarrow \mathcal{J}$ exists.

Let $\boldsymbol{x}$ be an element in an algebra $\mathcal{J}$, then we define $\boldsymbol{x}^{(1)} := \boldsymbol{x}$ and define $\boldsymbol{x}^{(n)}$ recursively by $\boldsymbol{x}^{(n)} := \boldsymbol{x} \circ \boldsymbol{x}^{(n-1)}$ for $n \geq 2$.

Definition 2.3 Let $\mathcal{J}$ be an algebra over $\mathbb{R}$ with a bilinear product $\circ : \mathcal{J} \times \mathcal{J} \longrightarrow \mathcal{J}$. Then $(\mathcal{J}, \circ)$ is called a Jordan algebra if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{J}$ we have

$$\begin{aligned} &\boldsymbol{x} \circ \boldsymbol{y} = \boldsymbol{y} \circ \boldsymbol{x} \text{ (commutativity), and} \\ &\boldsymbol{x} \circ (\boldsymbol{x}^{(2)} \circ \boldsymbol{y}) = \boldsymbol{x}^{(2)} \circ (\boldsymbol{x} \circ \boldsymbol{y}) \text{ (Jordan's axiom).} \end{aligned}$$

The product $\boldsymbol{x} \circ \boldsymbol{y}$ between two elements $\boldsymbol{x}$ and $\boldsymbol{y}$ of a Jordan algebra $(\mathcal{J}, \circ)$ is called the *Jordan multiplication* between $\boldsymbol{x}$ and $\boldsymbol{y}$. A Jordan algebra $(\mathcal{J}, \circ)$ has an *identity element* if there exists a (necessarily unique) element $\boldsymbol{e} \in \mathcal{J}$ such that $\boldsymbol{x} \circ \boldsymbol{e} = \boldsymbol{x}$ for all $\boldsymbol{x} \in \mathcal{J}$. A Jordan algebra $(\mathcal{J}, \circ)$ is not necessarily associative, that is, $\boldsymbol{x} \circ (\boldsymbol{y} \circ \boldsymbol{z}) = (\boldsymbol{x} \circ \boldsymbol{y}) \circ \boldsymbol{z}$ may not generally hold. However, it is power associative, i.e., $\boldsymbol{x}^{(p)} \circ \boldsymbol{x}^{(q)} = \boldsymbol{x}^{(p+q)}$ for all integers $p, q \geq 1$.

Example 2.1 For each vector $\boldsymbol{x} \in \mathbb{R}^n$ whose first entry is indexed with 0, we write $\widetilde{\boldsymbol{x}}$ for the subvector consisting of entries 1, through $n-1$ (therefore $\boldsymbol{x} = (x_0; \widetilde{\boldsymbol{x}}) \in \mathbb{R} \times \mathbb{R}^{n-1}$). We let $\mathcal{E}^n$ denote the n^{th} dimensional real vector space $\mathbb{R} \times \mathbb{R}^{n-1}$ whose vectors $\boldsymbol{x}$ are indexed from 0. Consider the multiplication $\circ : \mathcal{E}^n \times \mathcal{E}^n \longrightarrow \mathcal{E}^n$ defined as

$$\boldsymbol{x} \circ \boldsymbol{y} := \begin{bmatrix} x_0 y_0 + \widetilde{\boldsymbol{x}}^{\mathsf{T}} \widetilde{\boldsymbol{y}} \\ x_0 \widetilde{\boldsymbol{y}} + y_0 \widetilde{\boldsymbol{x}} \end{bmatrix}, \tag{1}$$

for $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{E}^n$. It is easy to see that the space $\mathcal{E}^n$ with the multiplication "$\circ$" forms a Jordan algebra with the identity vector $\boldsymbol{e} := (1; \boldsymbol{0})$.

Definition 2.4 A Jordan algebra $\mathcal{J}$ is called Euclidean if there exists a map $\bullet : (\mathcal{J}, \circ) \times (\mathcal{J}, \circ) \longrightarrow \mathbb{R}$ such that for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{J}$ we have

(1) $\boldsymbol{x} \bullet \boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq \boldsymbol{0}$ (positive definiteness);
(2) $\boldsymbol{x} \bullet \boldsymbol{y} = \boldsymbol{y} \bullet \boldsymbol{x}$ (symmetry);
(3) $\boldsymbol{x} \bullet (\boldsymbol{y} \circ \boldsymbol{z}) = (\boldsymbol{x} \circ \boldsymbol{y}) \bullet \boldsymbol{z}$ (associativity).

That is, $\mathcal{J}$ admits a positive definite, symmetric, quadratic form which is also associative.

The *degree* of an element $\boldsymbol{x} \in \mathcal{J}$ (denoted by $\deg(\boldsymbol{x})$) is the smallest integer d such that the set $\left\{\boldsymbol{e}, \boldsymbol{x}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(d)}\right\}$ is linearly independent. The *rank* of a Jordan algebra $\mathcal{J}$ (denoted by $\text{rank}(\mathcal{J})$) is the largest $\deg(\boldsymbol{x})$ of any element $\boldsymbol{x} \in \mathcal{J}$. An element $\boldsymbol{x} \in \mathcal{J}$ is called *regular* if $\deg(\boldsymbol{x}) = \text{rank}(\mathcal{J})$. Throughout the rest of this section, we let $\mathcal{J}$ be a rank-r Euclidean Jordan algebra.

Let $\boldsymbol{x}$ be an element of degree d in a Euclidean Jordan algebra $\mathcal{J}$. Since $\{\boldsymbol{e}, \boldsymbol{x}, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^d\}$ is linearly dependent, there are real numbers $a_1(\boldsymbol{x}), a_2(\boldsymbol{x}), \ldots, a_d(\boldsymbol{x})$, not all zero, such that

$$q(\boldsymbol{x}) := \boldsymbol{x}^d - a_1(\boldsymbol{x})\boldsymbol{x}^{d-1} + a_2(\boldsymbol{x})\boldsymbol{x}^{d-2} + \cdots + (-1)^d a_d(\boldsymbol{x})\boldsymbol{e} = \boldsymbol{0}.$$

We call q the *minimal polynomial* of $\boldsymbol{x}$. If $\boldsymbol{x}$ is a regular element of a Euclidean Jordan algebra, then we define its characteristic polynomial to be equal to its minimal polynomial. Since the set of regular elements is dense in $\mathcal{J}$, we can extend characteristic polynomials to all $\boldsymbol{x}$ in $\mathcal{J}$. So, the minimal polynomial coincides with the characteristic polynomial for regular elements and divides the characteristic polynomial of non-regular elements. Let $\boldsymbol{x}$ be an element in a rank-r algebra $\mathcal{J}$, then its *eigenvalues* are the roots $\lambda_1, \lambda_2, \ldots, \lambda_r$ of its characteristic polynomial $p(\lambda) = \lambda^r - a_1(\boldsymbol{x})\lambda^{r-1} + a_2(\boldsymbol{x})\lambda^{r-2} + \cdots + (-1)^r a_r(\boldsymbol{x})$.

Two elements $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathcal{J}$ are said to be *orthogonal* if $\boldsymbol{c}_1 \circ \boldsymbol{c}_2 = \boldsymbol{0}$. A set of elements of $\mathcal{J}$ is orthogonal if all its elements are mutually orthogonal to each other. An element $\boldsymbol{c} \in \mathcal{J}$ is said to be an *idempotent* if $\boldsymbol{c}^{(2)} = \boldsymbol{c}$. An idempotent is *primitive* if it is non-zero and cannot be written as a sum of two (necessarily orthogonal) non-zero idempotents. A subset $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ of $\mathcal{J}$ is called a *Jordan frame* if it is a complete system (i.e., $\boldsymbol{c}_1 + \boldsymbol{c}_2 + \cdots + \boldsymbol{c}_k = \boldsymbol{e}$) of orthogonal primitive idempotents.

Note that Jordan frames always have r primitive idempotents in them, so in Jordan frames $k = r$.

Theorem 2.1 (Spectral decomposition I, [31, Theorem 6]). *Let $\mathcal{J}$ be a Euclidean Jordan algebra. Then for each $\boldsymbol{x} \in \mathcal{J}$ there exist unique real numbers $\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \ldots, \lambda_k(\boldsymbol{x})$, all distinct, and a unique complete system of orthogonal idempotents $\boldsymbol{c}_1(\boldsymbol{x}), \boldsymbol{c}_2(\boldsymbol{x}), \ldots, \boldsymbol{c}_k(\boldsymbol{x})$ such that*

$$\boldsymbol{x} = \lambda_1(\boldsymbol{x})\boldsymbol{c}_1(\boldsymbol{x}) + \cdots + \lambda_k(\boldsymbol{x})\boldsymbol{c}_k(\boldsymbol{x}).$$

Theorem 2.2 (Spectral decomposition II, [31, Theorem 7]). *Let $\mathcal{J}$ be a rank-r Euclidean Jordan algebra. Then for each $\boldsymbol{x} \in \mathcal{J}$ there exist real numbers $\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \ldots, \lambda_r(\boldsymbol{x})$ and a Jordan frame $\boldsymbol{c}_1(\boldsymbol{x}), \boldsymbol{c}_2(\boldsymbol{x}), \ldots, \boldsymbol{c}_r(\boldsymbol{x})$ such that*

$$\boldsymbol{x} = \lambda_1(\boldsymbol{x})\boldsymbol{c}_1(\boldsymbol{x}) + \cdots + \lambda_r(\boldsymbol{x})\boldsymbol{c}_r(\boldsymbol{x}), \tag{2}$$

and $\lambda_1, \lambda_2, \ldots, \lambda_r$ are the eigenvalues of $\boldsymbol{x}$.

The elements $\boldsymbol{x}$ and $\boldsymbol{y}$ of a Jordan algebra are called *simultaneously decomposed* if they share a Jordan frame, i.e., $\boldsymbol{x} = \lambda_1(\boldsymbol{x})\,\boldsymbol{c}_1 + \cdots + \lambda_r(\boldsymbol{x})\,\boldsymbol{c}_r$ and $\boldsymbol{y} = \lambda_1(\boldsymbol{y})\,\boldsymbol{c}_1 + \cdots + \lambda_r(\boldsymbol{y})\,\boldsymbol{c}_r$ for a Jordan frame $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r\}$ (hence, we have $\boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{c}_i(\boldsymbol{y})$ for each $i = 1, \ldots, r$). Two elements $\boldsymbol{x}$ and $\boldsymbol{y}$ are said to *operator commute* if for all $\boldsymbol{z}$, we have that $\boldsymbol{x} \circ (\boldsymbol{y} \circ \boldsymbol{z}) = \boldsymbol{y} \circ (\boldsymbol{x} \circ \boldsymbol{z})$. Two elements of a Euclidean Jordan algebra are simultaneously decomposed iff they operator commute [1, Theorem 27].

Let $\boldsymbol{x}$ be an element in a Jordan algebra $\mathcal{J}$ with the spectral decomposition given in (2). Then $\mathrm{trace}(\boldsymbol{x}) := \lambda_1(\boldsymbol{x}) + \cdots + \lambda_r(\boldsymbol{x})$ is called the *trace* of $\boldsymbol{x}$ in $\mathcal{J}$, and $\det(\boldsymbol{x}) := \lambda_1(\boldsymbol{x}) \cdots \lambda_r(\boldsymbol{x})$ is the *determinant* of $\boldsymbol{x}$ in $\mathcal{J}$.

For a Euclidean Jordan algebra $\mathcal{J}$, we define the *Frobenius inner product* between two elements $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{J}$ as

$$\boldsymbol{x} \bullet \boldsymbol{y} := \operatorname{trace}(\boldsymbol{x} \circ \boldsymbol{y}).$$

Due to the bilinearity of "$\circ$", one can show that the map "$\bullet$" satisfies the three conditions in Definition 2.4. (See [1, Section 2]).

Example 2.2 One can easily show that the algebra $\mathcal{E}^n$, with the Jordan multiplication "$\circ$" defined in (1), forms a Euclidean Jordan algebra under the inner product $\boldsymbol{x} \bullet \boldsymbol{y} = 2\boldsymbol{x}^{\mathsf{T}}\boldsymbol{y} = 2\left(x_0 y_0 + \widetilde{\boldsymbol{x}}^{\mathsf{T}}\widetilde{\boldsymbol{y}}\right)$.

Now, for $\boldsymbol{x} \in \mathcal{J}$ having the spectral decomposition given in (2), we can define $\boldsymbol{x}^2$ as

$$\boldsymbol{x}^2 := (\lambda_1(\boldsymbol{x}))^2 \boldsymbol{c}_1(\boldsymbol{x}) + \cdots + (\lambda_r(\boldsymbol{x}))^2 \boldsymbol{c}_r(\boldsymbol{x}).$$

One can easily see that $\boldsymbol{x}^2 = \boldsymbol{x} \circ \boldsymbol{x} = \boldsymbol{x}^{(2)}$.

If $\det(\boldsymbol{x}) \neq 0$ (i.e., all $\lambda_i(\boldsymbol{x}) \neq 0$), then we say that $\boldsymbol{x}$ is *nonsingular*. In this case, the *inverse* of $\boldsymbol{x}$, which is denoted by $\boldsymbol{x}^{-1}$, is the unique element that satisfies $\boldsymbol{x}^{-1} \circ \boldsymbol{x} = \boldsymbol{e}$. Therefore,

$$\boldsymbol{x}^{-1} := \frac{1}{\lambda_1(\boldsymbol{x})}\boldsymbol{c}_1(\boldsymbol{x}) + \cdots + \frac{1}{\lambda_r(\boldsymbol{x})}\boldsymbol{c}_r(\boldsymbol{x}).$$

More generally, if $f : \mathbb{R} \to \mathbb{R}$ is continuous, then it is also possible to extend the above definition to define $f(\boldsymbol{x})$ as

$$f(\boldsymbol{x}) := f(\lambda_1(\boldsymbol{x}))\boldsymbol{c}_1(\boldsymbol{x}) + \cdots + f(\lambda_r(\boldsymbol{x}))\boldsymbol{c}_r(\boldsymbol{x}).$$

In particular, the *square root* of $\boldsymbol{x}$ is defined as

$$\boldsymbol{x}^{1/2} := \sqrt{\lambda_1(\boldsymbol{x})}\boldsymbol{c}_1(\boldsymbol{x}) + \cdots + \sqrt{\lambda_r(\boldsymbol{x})}\boldsymbol{c}_r(\boldsymbol{x}),$$

provided that $\lambda_i(\boldsymbol{x}) \geq 0, i = 1, 2, \ldots, r$.

The *cone of squares* of $\mathcal{J}$ is defined as $\mathcal{K}_{\mathcal{J}} := \left\{\boldsymbol{x}^2 : \boldsymbol{x} \in \mathcal{J}\right\}$. The *logarithmic barrier function*, which is defined on the interior of $\mathcal{K}_{\mathcal{J}}$ (denoted by $\operatorname{int}\mathcal{K}_{\mathcal{J}}$) as $\ell(\boldsymbol{x}) := -\ln\det(\boldsymbol{x})$, will play an important role for our subsequent development.

In the following theorem, we give the Jordan algebraic characterization of symmetric cones.

Theorem 2.3 [1, Theorem 2] *A regular cone is symmetric iff it is the cone of squares of some Euclidean Jordan algebra.*

Example 2.3 The cone of squares of the algebra $(\mathcal{E}^n, \circ)$, with "$\circ$" defined in (1), is the second-order cone, which is defined as

$$\mathcal{E}^n_+ := \{\boldsymbol{x} : x_0 \geq \|\widetilde{\boldsymbol{x}}\|\}.$$

The mappings introduced in following definition play an important role in the development of the interior point methods for conic programming [1].

Definition 2.5 Let $\boldsymbol{x}$ be an element in a Jordan algebra $\mathcal{J}$. Then

(a) The linear map $L(\boldsymbol{x}) : \mathcal{J} \longrightarrow \mathcal{J}$ is defined as $L(\boldsymbol{x})\boldsymbol{y} := \boldsymbol{x} \circ \boldsymbol{y}$ for all $\boldsymbol{y} \in \mathcal{J}$.
(b) The quadratic representation of $\boldsymbol{x}$, $Q_{(\boldsymbol{x})} : \mathcal{J} \longrightarrow \mathcal{J}$, is defined as $Q_{(\boldsymbol{x})} := 2L(\boldsymbol{x})^2 - L(\boldsymbol{x}^2)$.

Note that $Q_{(\boldsymbol{x})}$ is also a linear operator in $\mathcal{J}$. Note also that $L(\boldsymbol{x})\boldsymbol{e} = \boldsymbol{x}$, $L(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{x}^2$, $L(\boldsymbol{e}) = Q_{(\boldsymbol{e})} = I$, $\mathrm{trace}(\boldsymbol{e}) = 2$ and $\det(\boldsymbol{e}) = 1$ (since all the eigenvalues of $\boldsymbol{e}$ are equal to one).

The *Frobenius norm* of an element $\boldsymbol{x} \in \mathcal{J}$ is defined as $\|\boldsymbol{x}\|_F := \sqrt{\boldsymbol{x} \bullet \boldsymbol{x}} = \sqrt{\sum_{i=1}^{r} \lambda_i^2(\boldsymbol{x})}$. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{J}$, we have

$$\|\boldsymbol{x} \circ \boldsymbol{y}\|_F \leq \|\boldsymbol{x}\| \, \|\boldsymbol{y}\|_F \leq \|\boldsymbol{x}\|_F \, \|\boldsymbol{y}\|_F \quad \text{and} \quad \|\boldsymbol{x}+\boldsymbol{y}\|_F^2 = \|\boldsymbol{x}\|_F^2 + \|\boldsymbol{y}\|_F^2 + 8\boldsymbol{x} \bullet \boldsymbol{y}. \tag{3}$$

All the above notions are also used in the block sense as follows. Let $(\mathcal{J}_1, \circ_1, \bullet_1)$, $(\mathcal{J}_2, \circ_2, \bullet_2), \ldots, (\mathcal{J}_q, \circ_q, \bullet_q)$, be Euclidean Jordan algebras with identities $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_q$ and cone of squares $\mathcal{K}_{\mathcal{J}_1}, \mathcal{K}_{\mathcal{J}_2}, \cdots, \mathcal{K}_{\mathcal{J}_q}$, respectively. Let also $\boldsymbol{x} := (\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_q)$ and $\boldsymbol{y} := (\boldsymbol{y}_1; \boldsymbol{y}_2; \ldots; \boldsymbol{y}_q)$ with $\boldsymbol{x}_i, \boldsymbol{y}_i \in \mathcal{J}_i$ for $i = 1, 2, \ldots, q$. Then

(a) $\mathcal{K}_{\mathcal{J}} := \mathcal{K}_{\mathcal{J}_1} \times \mathcal{K}_{\mathcal{J}_2} \times \cdots \times \mathcal{K}_{\mathcal{J}_q}$ is the cone of squares of $\mathcal{J} := \mathcal{J}_1 \times \mathcal{J}_2 \times \cdots \times \mathcal{J}_q$.
(b) $\boldsymbol{e} := \left(\boldsymbol{e}_1; \boldsymbol{e}_2; \ldots; \boldsymbol{e}_q\right)$ is the identity vector of $\mathcal{J}$.
(c) $\boldsymbol{x} \circ \boldsymbol{y} := (\boldsymbol{x}_1 \circ_1 \boldsymbol{y}_1; \boldsymbol{x}_2 \circ_2 \boldsymbol{y}_2; \ldots; \boldsymbol{x}_q \circ_q \boldsymbol{y}_q)$.
(d) $\boldsymbol{x} \bullet \boldsymbol{y} := \boldsymbol{x}_1 \bullet_1 \boldsymbol{y}_1 + \boldsymbol{x}_2 \bullet_2 \boldsymbol{y}_2 + \cdots + \boldsymbol{x}_q \bullet_q \boldsymbol{y}_q$.
(e) $\mathrm{L}(\boldsymbol{x}) := \mathrm{L}(\boldsymbol{x}_1) \oplus \mathrm{L}(\boldsymbol{x}_2) \oplus \cdots \oplus \mathrm{L}(\boldsymbol{x}_q)$[1].
(f) $Q_{(\boldsymbol{x})} := Q_{(\boldsymbol{x}_1)} \oplus Q_{(\boldsymbol{x}_2)} \oplus \cdots \oplus Q_{(\boldsymbol{x}_q)}$.
(g) $f(\boldsymbol{x}) := (f(\boldsymbol{x}_1); f(\boldsymbol{x}_2); \ldots; f(\boldsymbol{x}_q))$. In particular, $\boldsymbol{x}^{-1} := \left(\boldsymbol{x}_1{}^{-1}; \boldsymbol{x}_2{}^{-1}; \ldots; \boldsymbol{x}_q{}^{-1}\right)$.
(h) $\|\boldsymbol{x}\|_F^2 := \|\boldsymbol{x}_1\|_F^2 + \|\boldsymbol{x}_2\|_F^2 + \cdots + \|\boldsymbol{x}_q\|_F^2$.
(i) The logarithmic barrier is defined on $\mathrm{int}\,\mathcal{K}_{\mathcal{J}}$ as $\ell(\boldsymbol{x}) := -\sum_{i=1}^{q} \ln \det(\boldsymbol{x}_i)$.
(j) $\boldsymbol{x}$ and $\boldsymbol{y}$ operator commute iff $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ operator commute for each $i = 1, 2, \ldots, q$.

We write $\boldsymbol{x} \succeq_{\mathcal{K}_{\mathcal{J}}} \boldsymbol{0}$ (or simply $\boldsymbol{x} \succeq \boldsymbol{0}$ when $\mathcal{K}_{\mathcal{J}}$ is known from the context) to mean that $\boldsymbol{x} \in \mathcal{K}_{\mathcal{J}}$ (i.e., $\boldsymbol{x}$ is positive semidefinite). We also write $\boldsymbol{x} \succ \boldsymbol{0}$ to mean that $\boldsymbol{x} \in \mathrm{int}\mathcal{K}_{\mathcal{J}}$ (i.e., $\boldsymbol{x}$ is positive definite), and write $\boldsymbol{x} \succeq \boldsymbol{y}$ ($\boldsymbol{x} \succ \boldsymbol{y}$) to mean that $\boldsymbol{x} - \boldsymbol{y} \succeq \boldsymbol{0}$ ($\boldsymbol{x} - \boldsymbol{y} \succ \boldsymbol{0}$). Note that $\boldsymbol{x} \succeq \boldsymbol{0}$ ($\boldsymbol{x} \succ \boldsymbol{0}$) iff $\lambda_i(\boldsymbol{x}) \geq 0$ ($\lambda_i(\boldsymbol{x}) > 0$) for $i = 1, 2, \ldots, r$.

3 Problem formulation and Newton's method

In this section, we introduce the primal and dual forms of symmetric programming problem and the perturbed problem. Then we apply Newton's method to the perturbed problem.

[1] The direct sum of two square matrices A and B is the block diagonal matrix $A \oplus B := \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$.

3.1 Problem formulation and assumptions

Let $q \geq 1$ be an integer. For each $i = 1, 2, \ldots, q$, let n, n_i, r and r_i be positive integers such that $n := \sum_{i=1}^{q} n_i$ and $r := \sum_{i=1}^{q} r_i$. Let $(\mathcal{J}_1, \circ_1, \bullet_1), (\mathcal{J}_2, \circ_2, \bullet_2), \ldots, (\mathcal{J}_q, \circ_q, \bullet_q)$ be Euclidean Jordan algebras with dimensions $n_1, n_2, \ldots, n_q$, ranks $r_1, r_2, \ldots, r_q$, identities $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_q$ and cone of squares $\mathcal{K}_{\mathcal{J}_1}, \mathcal{K}_{\mathcal{J}_2}, \ldots \mathcal{K}_{\mathcal{J}_q}$, respectively, and let

$$\begin{aligned}(\mathcal{J}, \circ, \bullet) &:= (\mathcal{J}_1, \circ_1, \bullet_1) \times (\mathcal{J}_2, \circ_2, \bullet_2) \times \cdots \times (\mathcal{J}_q, \circ_q, \bullet_q),\\ \mathcal{K}_{\mathcal{J}} &:= \mathcal{K}_{\mathcal{J}_1} \times \mathcal{K}_{\mathcal{J}_2} \times \cdots \times \mathcal{K}_{\mathcal{J}_q}.\end{aligned}$$

Let also $\boldsymbol{y}, \boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s} \in \mathcal{J}$ and $A \in \mathbb{R}^{m \times n}$ be such that they are conformally partitioned as

$$\begin{aligned}&\boldsymbol{x} := (\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_q),\ \boldsymbol{s} := (\boldsymbol{s}_1; \boldsymbol{s}_2; \ldots; \boldsymbol{s}_q),\ \boldsymbol{c} := (\boldsymbol{c}_1; \boldsymbol{c}_2; \ldots; \boldsymbol{c}_q),\ \text{and}\\ &A := (A_1, A_2, \ldots, A_q),\end{aligned}$$

where $\boldsymbol{x}_i, \boldsymbol{s}_i, \boldsymbol{c}_i \in \mathcal{J}_i$ and $A_i \in \mathbb{R}^{m \times n_i}$ for $i = 1, 2, \ldots, q$. More specifically, the submatrix A_i consists of m rows $\boldsymbol{a}_1^{(i)}, \boldsymbol{a}_2^{(i)}, \ldots, \boldsymbol{a}_m^{(i)} \in \mathcal{J}_i$ and acts as a linear transformation that maps $\boldsymbol{x}_i$ to the m^{th}-dimensional vector whose j^{th} component is $\boldsymbol{a}_j^{(i)} \bullet_i \boldsymbol{x}_i$ for each $i = 1, 2, \ldots, q$.

The *symmetric programming problem and its dual* in multi-block structures are defined as [1, Section 3]

$$\begin{array}{lll} & \min\ \boldsymbol{c}_1 \bullet_1 \boldsymbol{x}_1 + \cdots + \boldsymbol{c}_q \bullet_q \boldsymbol{x}_q & \\ (\mathrm{P}_i) & \text{s.t.}\ A_1\boldsymbol{x}_1 + \cdots + A_q\boldsymbol{x}_q = \boldsymbol{b}, & (\mathrm{D}_i) \\ & \quad \boldsymbol{x}_i \succeq \boldsymbol{0},\ i = 1, \ldots, q; & \end{array} \qquad \begin{array}{l} \max\ \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.}\ A_i{}^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s}_i = \boldsymbol{c}_i,\ i = 1, \ldots, q, \\ \quad \boldsymbol{s}_i \succeq \boldsymbol{0},\ \ i = 1, \ldots, q. \end{array}$$

We can rewrite the pair (P_i, D_i) compactly as

$$\begin{array}{lll} & \min\ \boldsymbol{c} \bullet \boldsymbol{x} & \\ (\mathrm{P}) & \text{s.t.}\ A\boldsymbol{x} = \boldsymbol{b}, & (\mathrm{D}) \\ & \quad \boldsymbol{x} \succeq \boldsymbol{0}; & \end{array} \qquad \begin{array}{l} \max\ \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.}\ A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}, \\ \quad \boldsymbol{s} \succeq \boldsymbol{0}. \end{array}$$

The general scheme of the central trajectory methods is as follows. We associate the perturbed problems to symmetric programming problem (P) and (D), then we draw a path of the centers defined by the perturbed KKT optimality conditions. After that, Newton's method is applied to treat the corresponding perturbed equations in order to obtain a descent search direction. As we mentioned earlier, we propose four different selections of calculating the displacement step.

Let $\mu > 0$ be a barrier parameter and $\sigma \in (0, 1)$ be the centering parameter. The *perturbed primal problem* corresponding to the primal problem (P) is

$$\begin{array}{ll} \min & f_\mu(\boldsymbol{x}) := \boldsymbol{c} \bullet \boldsymbol{x} + \sigma\mu\,\ell(\boldsymbol{x}) + r\mu\ln\mu \\ (\mathrm{P}_\mu) \quad \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \succ \boldsymbol{0}, \end{array}$$

and the *perturbed dual problem* corresponding to the dual problem (D) is

$$(\mathrm{D}_\mu)\qquad \begin{array}{ll} \max & g_\mu(\boldsymbol{y},\boldsymbol{s}) := \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} - \sigma\mu\,\ell(\boldsymbol{s}) - r\mu\ln\mu \\ \text{s.t.} & A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}, \\ & \boldsymbol{s} \succ \boldsymbol{0}. \end{array}$$

Now, we define the following feasibility sets:

$$\begin{aligned} \mathcal{F}_P &:= \{\boldsymbol{x} \in \mathcal{J} : A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succeq \boldsymbol{0}\}, \\ \mathcal{F}_D &:= \left\{(\boldsymbol{y};\boldsymbol{s}) \in \mathbb{R}^m \times \mathcal{J} : A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{s} \succeq \boldsymbol{0}\right\}, \\ \mathcal{F}_P^\circ &:= \{\boldsymbol{x} \in \mathcal{J} : A\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succ \boldsymbol{0}\}, \\ \mathcal{F}_D^\circ &:= \left\{(\boldsymbol{y};\boldsymbol{s}) \in \mathbb{R}^m \times \mathcal{J} : A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{s} \succ \boldsymbol{0}\right\}, \\ \mathcal{F}^\circ &:= \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ. \end{aligned}$$

Next, we make two assumptions about the primal-dual pair (P, D).

Assumption 3.1 The m rows of the matrix A are linearly independent.

Assumption 3.2 The set $\mathcal{F}^\circ$ is nonempty.

Assumption 3.1 is for convenience. Assumption 3.2 requires that Problem (P_μ) and its dual (D_μ) have strictly feasible solutions, which guarantees strong duality for the symmetric programming problem. Note that the feasible region for Problems (P_μ) and (D_μ) is described implicitly by $\mathcal{F}_0$. Due to the coercivity of the function f_μ on the feasible set of P_μ, Problem (P_μ) has an optimal solution.

The following lemma proves the convergence of the optimal solution of Problem (P_μ) to the optimal solution of Problem (P) when μ approaches zero.

Lemma 3.1 *Let $\bar{\boldsymbol{x}}_\mu$ be an optimal primal solution of Problem (P_μ), then $\bar{\boldsymbol{x}} = \lim_{\mu\to 0}\bar{\boldsymbol{x}}_\mu$ is an optimal solution of Problem (P).*

Proof Let $f_\mu(\boldsymbol{x}) := f(\boldsymbol{x},\mu)$ and $f(\boldsymbol{x}) := f(\boldsymbol{x},0)$. Due to the coercivity of the function f_μ on the feasible set of P_μ, Problem (P_μ) has an optimal solution, say $\bar{\boldsymbol{x}}_\mu$, such that

$$\nabla_{\boldsymbol{x}} f_\mu(\bar{\boldsymbol{x}}_\mu) = \nabla_{\boldsymbol{x}} f(\bar{\boldsymbol{x}}_\mu,\mu) = \boldsymbol{0}.$$

Then, for all $\boldsymbol{x} \in \mathcal{F}_P^\circ$, we have that

$$\begin{aligned} f(\boldsymbol{x}) &\geq f(\bar{\boldsymbol{x}}_\mu,\mu) + (\boldsymbol{x}-\bar{\boldsymbol{x}}_\mu)\bullet\nabla_{\boldsymbol{x}} f(\bar{\boldsymbol{x}}_\mu,\mu) + (0-\mu)\frac{\partial}{\partial\mu} f(\bar{\boldsymbol{x}}_\mu,\mu) \\ &\geq f(\bar{\boldsymbol{x}}_\mu,\mu) + \mu\ln\det\bar{\boldsymbol{x}}_\mu - r\mu\ln\mu - r\mu \\ &\geq \boldsymbol{c}\bullet\bar{\boldsymbol{x}}_\mu - \mu\ln\det\bar{\boldsymbol{x}}_\mu + r\mu\ln\mu + \mu\ln\det\bar{\boldsymbol{x}}_\mu - r\mu\ln\mu - r\mu \\ &\geq \boldsymbol{c}\bullet\bar{\boldsymbol{x}}_\mu - r\mu. \end{aligned}$$

Since $\boldsymbol{x}$ was arbitrary in $\mathcal{F}_P^\circ$, this implies that $\min_{\boldsymbol{x}\in\mathcal{F}_P^\circ} f(\boldsymbol{x}) \geq \boldsymbol{c}\bullet\bar{\boldsymbol{x}}_\mu - r\mu \geq \boldsymbol{c}\bullet\bar{\boldsymbol{x}}_\mu = f(\bar{\boldsymbol{x}}_\mu)$. On the other side, we have $f(\bar{\boldsymbol{x}}_\mu) \geq \min_{\boldsymbol{x}\in\mathcal{F}_P^\circ} f(\boldsymbol{x})$. As μ goes to 0, it immediately follows that $f(\bar{\boldsymbol{x}}) = \min_{\boldsymbol{x}\in\mathcal{F}_P^\circ} f(\boldsymbol{x})$. Thus, $\bar{\boldsymbol{x}}$ is an optimal solution of Problem (P). The proof is complete. □

3.2 Newton's method and commutative directions

As we mentioned, the objective function of Problem (P_μ) is strictly convex, hence the KKT conditions are necessary and sufficient to characterize an optimal solution of Problem (P_μ). Consequently, the points $\bar{\boldsymbol{x}}_\mu$ and $(\bar{\boldsymbol{y}}_\mu, \bar{\boldsymbol{s}}_\mu)$ are optimal solutions of (P_μ) and (D_μ) respectively iff they satisfy the perturbed nonlinear system

$$\begin{aligned} A\boldsymbol{x} &= \boldsymbol{b}, & \boldsymbol{x} &\succ \boldsymbol{0}, \\ A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} &= \boldsymbol{c}, & \boldsymbol{s} &\succ \boldsymbol{0}, \\ \boldsymbol{x} \circ \boldsymbol{s} &= \sigma\mu\boldsymbol{e}, & \mu &> 0, \end{aligned} \tag{4}$$

where $\boldsymbol{e} := \left(\boldsymbol{e}_{n_1}; \boldsymbol{e}_{n_2}; \ldots; \boldsymbol{e}_{n_q}\right)$ is the identity vector of $\mathcal{J}$.

We call the set of all solutions of system (4), denoted by $(\boldsymbol{x}_\mu, \boldsymbol{y}_\mu, \boldsymbol{s}_\mu)$ with $\mu > 0$, the *central trajectory* or *central path*. We say that a point $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ is near to the central trajectory if it belongs to the set $\mathcal{N}_\theta(\mu)$, which is defined as

$$\begin{aligned} \mathcal{N}_\theta(\mu) &:= \left\{(\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{s}) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ : d_F(\boldsymbol{x}, \boldsymbol{s}) \le \theta\mu\right\}, \quad \text{where} \\ d_F(\boldsymbol{x}, \boldsymbol{s}) &:= \|Q_{(\boldsymbol{x}^{1/2})}\boldsymbol{s} - \mu\boldsymbol{e}\|_F \quad \text{and} \quad \theta \in (0, 1). \end{aligned}$$

Now, we can apply Newton's method to system (4) and obtain the following linear system

$$\begin{aligned} A\Delta\boldsymbol{x} &= \boldsymbol{0}, \\ A^{\mathsf{T}}\Delta\boldsymbol{y} + \Delta\boldsymbol{s} &= \boldsymbol{0}, \\ \boldsymbol{x} \circ \Delta\boldsymbol{s} + \Delta\boldsymbol{x} \circ \boldsymbol{s} &= \sigma\mu\boldsymbol{e} - \boldsymbol{x} \circ \boldsymbol{s}. \end{aligned} \tag{5}$$

where $(\Delta\boldsymbol{x}; \Delta\boldsymbol{y}; \Delta\boldsymbol{s}) \in \mathcal{J} \times \mathbb{R}^m \times \mathcal{J}$ is the search direction and $\mu = \frac{1}{r}\boldsymbol{x} \bullet \boldsymbol{s}$ is the normalized duality gap corresponding to $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$.

Note that the strict second-order cone inequalities $\boldsymbol{x}, \boldsymbol{s} \succ \boldsymbol{0}$ imply that $d_F(\boldsymbol{x}, \boldsymbol{s}) \le \|\boldsymbol{x} \circ \boldsymbol{s} - \mu\boldsymbol{e}\|_F$ with equality holds when $\boldsymbol{x}$ and $\boldsymbol{s}$ operator commute [1, Lemma 30]. In fact, it is known that many interesting properties become apparent for the analysis of interior-point methods when $\boldsymbol{x}$ and $\boldsymbol{s}$ operator commute. The *commutative class* is denoted by $C(\boldsymbol{x}, \boldsymbol{s})$ and is defined as

$$C(\boldsymbol{x}, \boldsymbol{s}) := \left\{\boldsymbol{p} \in (\mathcal{J}, \circ, \bullet) : \boldsymbol{p} \text{ is nonsingular, } Q_{(\boldsymbol{p})}\boldsymbol{x} \text{ and } Q_{(\boldsymbol{p}^{-1})}\boldsymbol{s} \text{ operator commute}\right\}. \tag{6}$$

From [1, Lemma 28], the equality $\boldsymbol{x} \circ \boldsymbol{s} = \mu\boldsymbol{e}$ holds iff the equality $(Q_{(\boldsymbol{p})}\boldsymbol{x}) \circ (Q_{(\boldsymbol{p}^{-1})}\boldsymbol{s}) = \mu\boldsymbol{e}$ holds, for any nonsingular vector $\boldsymbol{p}$ in $\mathcal{J}$. Therefore, for any given nonsingular vector $\boldsymbol{p} \in (\mathcal{J}, \circ, \bullet)$, the system (4) is equivalent to the system

$$\begin{aligned} A\boldsymbol{x} &= \boldsymbol{b}, & \boldsymbol{x} &\succ \boldsymbol{0}, \\ A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} &= \boldsymbol{c}, & \boldsymbol{s} &\succ \boldsymbol{0}, \\ (Q_{(\boldsymbol{p})}\boldsymbol{x}) \circ (Q_{(\boldsymbol{p}^{-1})}\boldsymbol{s}) &= \sigma\mu\boldsymbol{e}, & \mu &> 0. \end{aligned} \tag{7}$$

Let $\boldsymbol{v} \in (\mathcal{J}, \circ, \bullet)$. With respect to a nonsingular vector $\boldsymbol{p} \in (\mathcal{J}, \circ, \bullet)$, we define the scaling vectors $\overline{\boldsymbol{v}}$ and $\underline{\boldsymbol{v}}$ and the scaling matrix $\underline{A}$ as

$$\overline{\boldsymbol{v}} := Q_{(\boldsymbol{p})}\boldsymbol{v}, \ \underline{\boldsymbol{v}} := Q_{(\boldsymbol{p}^{-1})}\boldsymbol{v}, \ \text{ and } \ \underline{A} := Q_{(\boldsymbol{p})}A.$$

Using this change of variables and the fact that $Q_{(\boldsymbol{p})}(\mathcal{J}) = \mathcal{J}$, we conclude that the system (5) is equivalent to the following Newton system

$$\begin{aligned} \underline{A}\overline{\Delta \boldsymbol{x}} &= \boldsymbol{b} - \underline{A}\overline{\boldsymbol{x}}, \\ \underline{A}^{\mathsf{T}}\Delta \boldsymbol{y} + \underline{\Delta \boldsymbol{s}} &= \underline{\boldsymbol{c}} - \underline{\boldsymbol{s}} - \underline{A}^{\mathsf{T}}\boldsymbol{y}, \\ \overline{\boldsymbol{x}} \circ \underline{\Delta \boldsymbol{s}} + \overline{\Delta \boldsymbol{x}} \circ \underline{\boldsymbol{s}} &= \sigma\mu\boldsymbol{e} - \overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}. \end{aligned} \tag{8}$$

Here, the normalized duality gap is $\mu = \frac{1}{r}\overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} = \frac{1}{r}\boldsymbol{x} \bullet \boldsymbol{s}$. In fact,

$$\overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} = \left(Q_{(\boldsymbol{p})}\boldsymbol{x}\right) \bullet Q_{(\boldsymbol{p}^{-1})}\boldsymbol{s} = \boldsymbol{x} \bullet Q_{(\boldsymbol{p})}Q_{(\boldsymbol{p}^{-1})}\boldsymbol{s} = \boldsymbol{x} \bullet \boldsymbol{s}. \tag{9}$$

Solving the scaled Newton system (8) yields the search direction $\left(\overline{\Delta \boldsymbol{x}}; \Delta \boldsymbol{y}; \underline{\Delta \boldsymbol{s}}\right)$. Then, we apply the inverse scaling to $\left(\overline{\Delta \boldsymbol{x}}; \underline{\Delta \boldsymbol{s}}\right)$ to obtain the Newton direction $(\Delta \boldsymbol{x}; \Delta \boldsymbol{s})$. Note that the search direction $\left(\overline{\Delta \boldsymbol{x}}; \Delta \boldsymbol{y}; \underline{\Delta \boldsymbol{s}}\right)$ belongs to the so-called the *MZ family of directions* (due to Monteiro [21] and Zhang [13]). In fact, such a way of scaling originally proposed for semidefinite programming by Monteiro [21] and Zhang [13], and after that it was generalized for general symmetric cone programming by Schmieta and Alizadeh [1].

Clearly, the set $C(\boldsymbol{x}, \boldsymbol{s})$ defined in (6) is a subclass of the MZ family of search directions. Our focus in this paper is in vectors $\boldsymbol{p} \in C(\boldsymbol{x}, \boldsymbol{s})$. We discuss the following three choices of $\boldsymbol{p}$ (see [1, Section 3]): We may choose $\boldsymbol{p} = \boldsymbol{x}^{1/2}$ to obtain $\overline{\boldsymbol{x}} = \boldsymbol{e}$, and we may choose $\boldsymbol{p} = \boldsymbol{s}^{-1/2}$ to obtain $\underline{\boldsymbol{s}} = \boldsymbol{e}$. These two choices of directions are called the *HRVW/KSH/M directions* (due to Helmberg et al. [22], Monteiro [21] and Kojima et al. [32]). The third choice of $\boldsymbol{p}$ is given by

$$\boldsymbol{p} = \left(Q_{(\boldsymbol{x}^{1/2})}\left(Q_{(\boldsymbol{x}^{1/2})}\boldsymbol{s}\right)^{-1/2}\right)^{-1/2},$$

which yields $Q^2_{(\boldsymbol{p})}\boldsymbol{x} = \boldsymbol{s}$, and therefore $\underline{\boldsymbol{s}} = Q_{(\boldsymbol{p}^{-1})}\boldsymbol{s} = Q_{(\boldsymbol{p})}\boldsymbol{x} = \overline{\boldsymbol{x}}$. This choice of directions is called the *NT direction* (due to Nesterov and Todd [7]).

4 Selections of the displacement step

We start by motivating the need for introducing the so-called displacement step. Note that the positive definiteness of the vectors $\boldsymbol{x}^+ = \boldsymbol{x} + \Delta \boldsymbol{x}$ and $\boldsymbol{s}^+ = \boldsymbol{s} + \Delta \boldsymbol{s}$ is not always achieved. In order to circumvent this difficulty, we introduce a parameter $\alpha > 0$, for which we call the *displacement step-size* or simply the *displacement step*, then we redefine $\boldsymbol{x}^+$, $\boldsymbol{y}^+$ and $\boldsymbol{s}^+$ as

$$\boldsymbol{x}^+ := \boldsymbol{x} + \alpha\Delta \boldsymbol{x}, \ \boldsymbol{y}^+ := \boldsymbol{y} + \alpha\Delta \boldsymbol{y}, \ \text{ and } \boldsymbol{s}^+ := \boldsymbol{s} + \alpha\Delta \boldsymbol{s}.$$

Calculating the displacement step using classical line search methods is undesirable and even generally impossible [20,23]. Strategies for determining the displacement step and preserving positive definiteness exists in the literature for semidefinite programming case only [23]. To the best of our knowledge, no such strategies available for general symmetric programming. From this stems the importance of the results of this paper. Calculating the search direction for any Newton method for problems over symmetric cones has been well established in the literature, but there are gaps in terms of maintaining feasibility. In order to fill such gaps, this section proposes four different selections of computing the appropriate displacement step for symmetric optimization. Lemmas 4.1, 4.2, 4.3 and 4.4 below generalize Lemmas 3.3, 3.4, 3.5 and 3.6, respectively, in [23, Section 3]. Each one of the following four lemmas gives a selection to calculate the displacement step α.

Lemma 4.1 (The first selection lemma). *Let* $(\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{s}) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ$. *If* $\alpha = \rho \min\{\alpha_{\boldsymbol{x}}, \alpha_{\boldsymbol{s}}\}$ *with* $0 < \rho < 1$, *then* $\boldsymbol{x}^+, \boldsymbol{s}^+ \succ \mathbf{0}$, *where for* $\boldsymbol{v} \in \{\boldsymbol{x}, \boldsymbol{s}\}$ *we have*

$$\alpha_{\boldsymbol{v}} = \begin{cases} \dfrac{-1}{\bar{\lambda}_{\boldsymbol{v}} - \delta_{\boldsymbol{v}}\sqrt{r-1}} - \varepsilon, & \text{if } \left(\dfrac{-1}{\bar{\lambda}_{\boldsymbol{v}} - \delta_{\boldsymbol{v}}\sqrt{r-1}} > 0 \text{ and } \min\limits_{1\le i\le r} \lambda_i \left(\boldsymbol{v}^{-1/2} \circ \left(\Delta\boldsymbol{v} \circ \boldsymbol{v}^{-1/2} \right) \right) < 0 \right); \\ \varepsilon, & \text{if } \left(\dfrac{-1}{\bar{\lambda}_{\boldsymbol{v}} - \delta_{\boldsymbol{v}}\sqrt{r-1}} < 0 \text{ and } \min\limits_{1\le i\le r} \lambda_i \left(\boldsymbol{v}^{-1/2} \circ \left(\Delta\boldsymbol{v} \circ \boldsymbol{v}^{-1/2} \right) \right) < 0 \right); \\ 1, & \text{if } \min\limits_{1\le i\le r} \lambda_i \left(\boldsymbol{v}^{-1/2} \circ \left(\Delta\boldsymbol{v} \circ \boldsymbol{v}^{-1/2} \right) \right) > 0, \end{cases} \tag{10}$$

where

$$\bar{\lambda}_{\boldsymbol{v}} = \frac{1}{r} \sum_{i=1}^{r} \lambda_i \left(\boldsymbol{v}^{-1/2} \circ \left(\Delta\boldsymbol{v} \circ \boldsymbol{v}^{-1/2} \right) \right),$$

$$\delta_{\boldsymbol{v}}^2 = \frac{1}{n} \sum_{i=1}^{r} \lambda_i^2 \left(\boldsymbol{v}^{-1/2} \circ \left(\Delta\boldsymbol{v} \circ \boldsymbol{v}^{-1/2} \right) \right) - \bar{\lambda}_{\boldsymbol{v}}^2,$$

and ϵ *is a small positive real. Here,* $\lambda_i \left(\boldsymbol{v}^{-1/2} \circ \left(\Delta\boldsymbol{v} \circ \boldsymbol{v}^{-1/2} \right) \right), i = 1, 2, \ldots, r$, *are the eigenvalues of the vector* $\boldsymbol{v}^{-1/2} \circ \left(\Delta\boldsymbol{v} \circ \boldsymbol{v}^{-1/2} \right)$.

Proof Since $\boldsymbol{x} \succ \mathbf{0}$, the vectors $\boldsymbol{x}^{\pm 1/2}$ are well-defined and we have $\boldsymbol{x}^{\pm 1/2} \succ \mathbf{0}$. Note that

$$\begin{aligned} \boldsymbol{x}^+ &= \boldsymbol{x} + \alpha\ \Delta\boldsymbol{x} = \boldsymbol{x}^{1/2} \circ \boldsymbol{x}^{1/2} + \alpha\ \Delta\boldsymbol{x} \\ &= \boldsymbol{x}^{1/2} \circ \left(\left(\boldsymbol{e} + \alpha\ \boldsymbol{x}^{-1/2} \circ \left(\Delta\boldsymbol{x} \circ \boldsymbol{x}^{-1/2} \right) \right) \circ \boldsymbol{x}^{1/2} \right). \end{aligned}$$

Therefore, as $\boldsymbol{x}^{-1/2} \succ \mathbf{0}$, we have that

$$\begin{aligned} \boldsymbol{x}^+ \succ \mathbf{0} &\iff \boldsymbol{x}^{-1/2} \circ \boldsymbol{x}^+ \succ \mathbf{0} \\ &\iff \left(\boldsymbol{x}^{-1/2} \circ \boldsymbol{x}^+ \right) \circ \boldsymbol{x}^{-1/2} \succ \mathbf{0} \\ &\iff \boldsymbol{e} + \alpha\ \boldsymbol{x}^{-1/2} \circ \left(\Delta\boldsymbol{x} \circ \boldsymbol{x}^{-1/2} \right) \succ \mathbf{0} \\ &\iff 1 + \alpha\lambda_i \left(\boldsymbol{x}^{-1/2} \circ \left(\Delta\boldsymbol{x} \circ \boldsymbol{x}^{-1/2} \right) \right) > 0, \text{ for } i = 1, 2, \ldots, r. \end{aligned}$$

Hence,

$$1+\alpha \min_{1\le i\le r} \lambda_i\left(\boldsymbol{x}^{-1/2}\circ\left(\Delta \boldsymbol{x}\circ \boldsymbol{x}^{-1/2}\right)\right)>0$$
$$\Longrightarrow \quad \alpha \min_{1\le i\le r} \lambda_i\left(\boldsymbol{x}^{-1/2}\circ\left(\Delta \boldsymbol{x}\circ \boldsymbol{x}^{-1/2}\right)\right)>-1.$$

If $\min_{1\le i\le r}\lambda_i\left(\boldsymbol{x}^{-1/2}\circ\left(\Delta \boldsymbol{x}\circ \boldsymbol{x}^{-1/2}\right)\right)<0$, then $\alpha<-1/\min_{1\le i\le r}\lambda_i\left(\boldsymbol{x}^{-1/2}\circ\left(\Delta \boldsymbol{x}\circ \boldsymbol{x}^{-1/2}\right)\right)$. Using Proposition 2.1, we conclude that

$$\alpha<\frac{-1}{\min\limits_{1\le i\le r} \lambda_i\left(\boldsymbol{x}^{-1/2}\circ\left(\Delta \boldsymbol{x}\circ \boldsymbol{x}^{-1/2}\right)\right)} \quad \text{and} \quad \frac{-1}{\bar{\lambda}_{\boldsymbol{x}}-\delta_{\boldsymbol{x}}\sqrt{r-1}}$$
$$\le \frac{-1}{\min\limits_{1\le i\le r} \lambda_i\left(\boldsymbol{x}^{-1/2}\circ\left(\Delta \boldsymbol{x}\circ \boldsymbol{x}^{-1/2}\right)\right)}.$$

This gives the expression of $\alpha_{\boldsymbol{x}}$. Applying the same procedure, we obtain $\alpha_{\boldsymbol{s}}$. Finally, we choose $\alpha=\rho\min\{\alpha_{\boldsymbol{x}},\alpha_{\boldsymbol{s}}\}$ with $0<\rho<1$. The proof is complete. □

Lemma 4.2 (The second selection lemma). *Let* $(\boldsymbol{x};\boldsymbol{y};\boldsymbol{s})\in\mathcal{F}_P^\circ\times\mathcal{F}_D^\circ$. *If* $\alpha=\rho\min\{\alpha_{\boldsymbol{x}},\alpha_{\boldsymbol{s}}\}$ *with* $0<\rho<1$, *then* $\boldsymbol{x}^+,\boldsymbol{s}^+\succ\boldsymbol{0}$, *where for* $\boldsymbol{v}\in\{\boldsymbol{x},\boldsymbol{s}\}$ *we have*

$$\alpha_{\boldsymbol{v}}=\begin{cases}\dfrac{-1}{\bar{\lambda}_{\boldsymbol{v}}-\delta_{\boldsymbol{v}}\sqrt{r-1}}-\varepsilon, & \text{if } \left(\dfrac{-1}{\bar{\lambda}_{\boldsymbol{v}}-\delta_{\boldsymbol{v}}\sqrt{r-1}}>0 \text{ and } \min\limits_{1\le i\le r}\lambda_i\left(\boldsymbol{v}^{-1}\circ\Delta\boldsymbol{v}\right)<0\right);\\ \varepsilon, & \text{if } \left(\dfrac{-1}{\bar{\lambda}_{\boldsymbol{v}}-\delta_{\boldsymbol{v}}\sqrt{r-1}}<0 \text{ and } \min\limits_{1\le i\le r}\lambda_i\left(\boldsymbol{v}^{-1}\circ\Delta\boldsymbol{v}\right)<0\right);\\ 1, & \text{if } \min\limits_{1\le i\le r}\lambda_i\left(\boldsymbol{v}^{-1}\circ\Delta\boldsymbol{v}\right)>0,\end{cases} \tag{11}$$

where

$$\bar{\lambda}_{\boldsymbol{v}}=\frac{1}{r}\sum_{i=1}^{r}\lambda_i\left(\boldsymbol{v}^{-1}\circ\Delta\boldsymbol{v}\right),\quad \delta_{\boldsymbol{v}}^2=\frac{1}{n}\sum_{i=1}^{r}\lambda_i^2\left(\boldsymbol{v}^{-1}\circ\Delta\boldsymbol{v}\right)-\bar{\lambda}_{\boldsymbol{v}}^2,$$

and ϵ *is a small positive real. Here,* $\lambda_i\left(\boldsymbol{v}^{-1}\circ\Delta\boldsymbol{v}\right)$, $i=1,2,\ldots,r$, *are the eigenvalues of the vector* $\boldsymbol{v}^{-1}\circ\Delta\boldsymbol{v}$.

Proof Since $\boldsymbol{x}\succ\boldsymbol{0}$, the vector $\boldsymbol{x}^{-1}$ is invertible and positive definite (i.e., $\boldsymbol{x}^{-1}\succ\boldsymbol{0}$). Note that

$$\boldsymbol{x}^+=\boldsymbol{x}+\alpha\ \Delta\boldsymbol{x}=\boldsymbol{x}\circ\left(\boldsymbol{e}+\alpha\ \boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\right).$$

Therefore, as $\boldsymbol{x}^{-1}\succ\boldsymbol{0}$, we have that

$$\begin{aligned}\boldsymbol{x}^+\succ\boldsymbol{0} &\Longleftrightarrow \boldsymbol{x}^{-1}\circ\boldsymbol{x}^+\succ\boldsymbol{0}\\ &\Longleftrightarrow \boldsymbol{e}+\alpha\ \boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\succ\boldsymbol{0}\\ &\Longleftrightarrow 1+\alpha\lambda_i\left(\boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\right)>0,\ \text{for } i=1,2,\ldots,r.\end{aligned}$$

Hence,

$$1+\alpha \min_{1\le i\le r} \lambda_i\left(\boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\right)>0 \implies \alpha \min_{1\le i\le r} \lambda_i\left(\boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\right)>-1.$$

If $\min_{1\le i\le r}\lambda_i\left(\boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\right) < 0$, then $\alpha < -1/\min_{1\le i\le r}\lambda_i\left(\boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\right)$. Using Proposition 2.1, we conclude that

$$\alpha < \frac{-1}{\min\limits_{1\le i\le r}\lambda_i\left(\boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\right)} \quad\text{and}\quad \frac{-1}{\bar{\lambda}_{\boldsymbol{x}}-\delta_{\boldsymbol{x}}\sqrt{r-1}} \le \frac{-1}{\min\limits_{1\le i\le r}\lambda_i\left(\boldsymbol{x}^{-1}\circ\Delta\boldsymbol{x}\right)}.$$

This gives the expression of $\alpha_{\boldsymbol{x}}$. Applying the same procedure, we obtain $\alpha_{\boldsymbol{s}}$. Finally, we choose $\alpha=\rho\min\{\alpha_{\boldsymbol{x}},\alpha_{\boldsymbol{s}}\}$ with $0<\rho<1$. The proof is complete. □

Lemma 4.3 (The third selection lemma). *Let* $(\boldsymbol{x};\boldsymbol{y};\boldsymbol{s}) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ$. *If* $\alpha = \rho\min\{\alpha_{\boldsymbol{x}},\alpha_{\boldsymbol{s}}\}$ *with* $0<\rho<1$, *then* $\boldsymbol{x}^+,\boldsymbol{s}^+\succ\boldsymbol{0}$, *where for* $\boldsymbol{v}\in\{\boldsymbol{x},\boldsymbol{s}\}$ *we have*

$$\alpha_{\boldsymbol{v}}=\begin{cases} -\dfrac{\bar{\lambda}_{\boldsymbol{v}}-\delta_{\boldsymbol{v}}\sqrt{r-1}}{\bar{\lambda}_{\Delta\boldsymbol{v}}-\delta_{\Delta\boldsymbol{v}}\sqrt{r-1}}-\varepsilon, & \text{if } \left(-\dfrac{\bar{\lambda}_{\boldsymbol{v}}-\delta_{\boldsymbol{v}}\sqrt{r-1}}{\bar{\lambda}_{\Delta\boldsymbol{v}}-\delta_{\Delta\boldsymbol{v}}\sqrt{r-1}}>0 \text{ and } \min\limits_{1\le i\le r}\lambda_i(\Delta\boldsymbol{v})<0\right);\\ \varepsilon, & \text{if } \left(-\dfrac{\bar{\lambda}_{\boldsymbol{v}}-\delta_{\boldsymbol{v}}\sqrt{r-1}}{\bar{\lambda}_{\Delta\boldsymbol{v}}-\delta_{\Delta\boldsymbol{v}}\sqrt{r-1}} \text{ and } \min\limits_{1\le i\le r}\lambda_i(\Delta\boldsymbol{v})<0\right);\\ 1, & \text{if } \min\limits_{1\le i\le r}\lambda_i(\Delta\boldsymbol{v})>0, \end{cases} \tag{12}$$

where

$$\bar{\lambda}_{\boldsymbol{v}}=\frac{1}{r}\sum_{i=1}^{r}\lambda_i(\boldsymbol{v}),\quad \bar{\lambda}_{\Delta\boldsymbol{v}}=\frac{1}{r}\sum_{i=1}^{r}\lambda_i(\Delta\boldsymbol{v}),\quad \delta_{\boldsymbol{v}}^2=\frac{1}{r}\sum_{i=1}^{r}\lambda_i^2(\boldsymbol{v})-\bar{\lambda}_{\boldsymbol{v}}^2,$$
$$\delta_{\Delta\boldsymbol{v}}^2=\frac{1}{r}\sum_{i=1}^{r}\lambda_i^2(\Delta\boldsymbol{v})-\bar{\lambda}_{\Delta\boldsymbol{v}}^2,$$

and ϵ *is a small positive real. Here,* $\lambda_i(\boldsymbol{v}),\ i=1,2,\ldots,r$, *are the eigenvalues of the vector* $\boldsymbol{v}$, *and* $\lambda_i(\Delta\boldsymbol{v}),\ i=1,2,\ldots,r$, *are the eigenvalues of the vector* $\Delta\boldsymbol{v}$,

Proof It is known that $\boldsymbol{x}^+=\boldsymbol{x}+\alpha\Delta\boldsymbol{x}\succ\boldsymbol{0}$ iff $\min_{1\le i\le r}\lambda_i(\boldsymbol{x}^+)>0$. Here, $\lambda_i(\boldsymbol{x}^+),\ i=1,2,\ldots,r$, are the eigenvalues of the vector $\boldsymbol{x}^+$. It is also known (see for example [33]) that

$$\min_{1\le i\le r}\lambda_i(\boldsymbol{x}^+)\ge\min_{1\le i\le r}\lambda_i(\boldsymbol{x})+\min_{1\le i\le r}\lambda_i(\Delta\boldsymbol{x}).$$

Then, it is enough to find α such that

$$\min_{1\le i\le r}\lambda_i(\boldsymbol{x}^+)\ge\min_{1\le i\le r}\lambda_i(\boldsymbol{x})+\alpha\min_{1\le i\le r}\lambda_i(\Delta\boldsymbol{x})>0.$$

Hence

$$\min_{1\le i\le r} \lambda_i(\Delta \boldsymbol{x}) < 0 \implies \alpha < -\frac{\min_{1\le i\le r} \lambda_i(\boldsymbol{x})}{\min_{1\le i\le r} \lambda_i(\Delta \boldsymbol{x})}.$$

Using Proposition 2.1, we find

$$\alpha < -\frac{\min_{1\le i\le r} \lambda_i(\boldsymbol{x})}{\min_{1\le i\le r} \lambda_i(\Delta \boldsymbol{x})} \quad \text{and} \quad -\frac{\bar{\lambda}_{\boldsymbol{x}} - \delta_{\boldsymbol{x}}\sqrt{r-1}}{\bar{\lambda}_{\Delta \boldsymbol{x}} - \delta_{\Delta \boldsymbol{x}}\sqrt{r-1}} \le -\frac{\min_{1\le i\le r} \lambda_i(\boldsymbol{x})}{\min_{1\le i\le r} \lambda_i(\Delta \boldsymbol{x})}.$$

This gives the expression of $\alpha_{\boldsymbol{x}}$. Applying the same procedure, we obtain $\alpha_{\boldsymbol{s}}$. Finally, we choose $\alpha = \rho \min\{\alpha_{\boldsymbol{x}}, \alpha_{\boldsymbol{s}}\}$ with $0 < \rho < 1$. The proof is complete. □

Lemma 4.4 (The fourth selection lemma). *Let* $(\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{s}) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ$. *If* $\alpha = \rho \min\{\alpha_{\boldsymbol{x}}, \alpha_{\boldsymbol{s}}\}$ *with* $0 < \rho < 1$, *then* $\boldsymbol{x}^+, \boldsymbol{s}^+ \succ \boldsymbol{0}$, *where for* $\boldsymbol{v} \in \{\boldsymbol{x}, \boldsymbol{s}\}$ *we have*

$$\alpha_{\boldsymbol{v}} = \begin{cases} \min\limits_{i\in I_{\boldsymbol{v}}} \dfrac{\left(\sum\limits_{j\neq k=1}^{n_i} \left|\{L(\boldsymbol{v}_i)\}_{jk}\right|\right) - \{L(\boldsymbol{v}_i)\}_{jj}}{\{L(\Delta \boldsymbol{v}_i)\}_{jj} - \sum\limits_{j\neq k=1}^{n_i} \left|\{L(\Delta \boldsymbol{v}_i)\}_{jk}\right|}, & \text{if } I_{\boldsymbol{v}} \neq \emptyset; \\ +\infty, & \text{if } I_{\boldsymbol{v}} = \emptyset, \end{cases} \tag{13}$$

where

$$I_{\boldsymbol{v}} := \left\{ i \in \{1, 2, \ldots, q\} : \{L(\Delta \boldsymbol{v}_i)\}_{jj} - \sum_{j\neq k=1}^{n_i} \left|\{L(\Delta \boldsymbol{v}_i)\}_{jk}\right| < 0 \right\}.$$

Proof Note that a vector $\boldsymbol{w} \in \mathcal{J}$ is positive definite (i.e., $\boldsymbol{w} \succ \boldsymbol{0}$) iff $\mathrm{L}(\boldsymbol{w}_i)$ is positive definite (i.e., $\boldsymbol{w}_i \succ \boldsymbol{0}$) for $i = 1, 2, \ldots, q$ [1, Lemma 12]. Then, $\boldsymbol{x}^+ = \boldsymbol{x} + \alpha \Delta \boldsymbol{x} \succ \boldsymbol{0}$ if

$$\{\mathrm{L}(\boldsymbol{x}_i^+)\}_{jj} > \sum_{j\neq k=1}^{n_i} \left|\{\mathrm{L}(\boldsymbol{x}_i^+)\}_{jk}\right|, \quad \forall \ i = 1, 2, \ldots, q. \tag{14}$$

Observe that $\mathrm{L}(\boldsymbol{x}_i^+) = \mathrm{L}(\boldsymbol{x}_i + \alpha \Delta \boldsymbol{x}_i) = \mathrm{L}(\boldsymbol{x}_i) + \alpha \mathrm{L}(\Delta \boldsymbol{x}_i)$ for $i = 1, 2, \ldots, q$. So, the inequality (14) is equivalent to

$$\{\mathrm{L}(\boldsymbol{x}_i)\}_{jj} + \alpha\{\mathrm{L}(\Delta \boldsymbol{x}_i)\}_{jj} > \sum_{j\neq k=1}^{n_i} \left|\{\mathrm{L}(\boldsymbol{x}_i)\}_{jk} + \alpha\{\mathrm{L}(\Delta \boldsymbol{x}_i)\}_{jk}\right|, \quad \forall \ i = 1, 2, \ldots, q.$$

Because, for $i = 1, 2, \ldots, q$, we have

$$\sum_{j\neq k=1}^{n_i} \left(\left|\{\mathrm{L}(\boldsymbol{x}_i)\}_{jk}\right| + \alpha \left|\{\mathrm{L}(\Delta \boldsymbol{x}_i)\}_{jk}\right|\right) \ge \sum_{j\neq k=1}^{n_i} \left|\{\mathrm{L}(\boldsymbol{x}_i)\}_{jk} + \alpha\{\mathrm{L}(\Delta \boldsymbol{x}_i)\}_{jk}\right|,$$

it is enough to find α such that

$$\{\mathrm{L}(\boldsymbol{x}_i)\}_{jj} + \alpha\{\mathrm{L}(\Delta\boldsymbol{x}_i)\}_{jj} > \sum_{j\neq k=1}^{n_i} \left(\left|\{\mathrm{L}(\boldsymbol{x}_i)\}_{jk}\right| + \alpha\left|\{\mathrm{L}(\Delta\boldsymbol{x}_i)\}_{jk}\right|\right), \quad \forall\ i = 1, 2, \ldots, q,$$

or equivalently

$$\alpha\left(\{\mathrm{L}(\Delta\boldsymbol{x}_i)\}_{jj} - \sum_{j\neq k=1}^{n_i}\left|\{\mathrm{L}(\Delta\boldsymbol{x}_i)\}_{jk}\right|\right) > \left(\sum_{j\neq k=1}^{n_i}\left|\{\mathrm{L}(\boldsymbol{x}_i)\}_{jk}\right|\right) - \{\mathrm{L}(\boldsymbol{x}_i)\}_{jj}, \quad \forall\ i = 1, 2, \ldots, q.$$

This gives the expression of $\alpha_{\boldsymbol{x}}$.

Applying the same procedure for the vector $\boldsymbol{s}^+ = \boldsymbol{s} + \alpha\Delta\boldsymbol{s}$, we obtain $\alpha_{\boldsymbol{s}}$. Finally, we choose $\alpha = \rho\min\{\alpha_{\boldsymbol{x}}, \alpha_{\boldsymbol{s}}\}$ with $0 < \rho < 1$. The proof is complete. □

5 The central trajectory algorithm

The central trajectory algorithm for solving symmetric programming problem is formally stated in Algorithm 5.1. Algorithm 5.1 selects a sequence of displacement steps $\{\alpha^{(k)}\}$ and centrality parameters $\{\sigma^{(k)}\}$ according to the following rule: for all $k \geq 0$, we take $\sigma^{(k)} = 1 - \delta/\sqrt{r}$, where $\delta \in \left[0, \sqrt{r}\right)$ as it will be seen in the next section.

Algorithm 5.1. The central trajectory algorithm for symmetric programming problem.

Begin algorithm

1: Initialize $k = 0, \boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{s}^{(0)}, \mu^{(0)}, \epsilon, \sigma^{(k)}, \theta$

Ensure: $\left(\boldsymbol{x}^{(0)}; \boldsymbol{y}^{(0)}; \boldsymbol{s}^{(0)}\right) \in \mathcal{N}_\theta\left(\mu^{(0)}\right), \epsilon > 0,\ \sigma^{(0)}, \theta \in (0, 1)$

2: **While** $\boldsymbol{x}^{(k)} \bullet \boldsymbol{s}^{(k)} \geq \epsilon$ **do**

3: Choose $\boldsymbol{p}^{(k)} \in C\left(\boldsymbol{x}^{(k)}, \boldsymbol{s}^{(k)}\right)$

4: Compute $\left(\overline{\boldsymbol{x}^{(k)}}; \boldsymbol{y}^{(k)}; \underline{\boldsymbol{s}^{(k)}}\right)$ by applying scaling to $\left(\boldsymbol{x}^{(k)}; \boldsymbol{y}^{(k)}; \boldsymbol{s}^{(k)}\right)$

5: Let $\mu^{(k)} := \frac{1}{r}\overline{\boldsymbol{x}}^{(k)} \bullet \underline{\boldsymbol{s}}^{(k)},\ \boldsymbol{h}^{(k)} := \sigma^{(k)}\mu^{(k)}\boldsymbol{e} - \overline{\boldsymbol{x}^{(k)}} \circ \underline{\boldsymbol{s}^{(k)}}$ and $\Psi^{(k)} := \frac{1}{\mu}\underline{A}\ \overline{\boldsymbol{x}^{(k)^2}}\ \underline{A}^{\mathsf{T}}$

6: Compute $\left(\overline{\Delta\boldsymbol{x}^{(k)}}; \Delta\boldsymbol{y}^{(k)}; \underline{\Delta\boldsymbol{s}^{(k)}}\right)$ by solving the scaled Newton system (8) to get

$$\left(\overline{\Delta\boldsymbol{x}^{(k)}};\ \Delta\boldsymbol{y}^{(k)};\ \underline{\Delta\boldsymbol{s}^{(k)}}\right) := \left(\left(\boldsymbol{h}^{(k)} - \overline{\boldsymbol{x}^{(k)}} \circ \underline{\Delta\boldsymbol{s}^{(k)}}\right) \circ \underline{\boldsymbol{s}^{(k)}}^{-1};\ -\Psi^{(k)^{-1}}\underline{A}\left(\underline{\boldsymbol{s}^{(k)}}^{-1} \circ \boldsymbol{h}^{(k)}\right);\ -\underline{A}^{\mathsf{T}}\Delta\boldsymbol{y}^{(k)}\right)$$

7: Compute $\left(\Delta\boldsymbol{x}^{(k)}; \Delta\boldsymbol{y}^{(k)}; \Delta\boldsymbol{s}^{(k)}\right)$ by applying inverse scaling to $\left(\overline{\Delta\boldsymbol{x}^{(k)}}; \Delta\boldsymbol{y}^{(k)}; \underline{\Delta\boldsymbol{s}^{(k)}}\right)$

8: Calculate the displacement step $\alpha^{(k)}$ by one of the selections in (10), (11), (12) or (13)

9: Set the new iterate according to

$$\left(\boldsymbol{x}^{(k+1)};\ \boldsymbol{y}^{(k+1)};\ \boldsymbol{s}^{(k+1)}\right) := \left(\boldsymbol{x}^{(k)} + \alpha^{(k)}\Delta\boldsymbol{x}^{(k)};\right.$$
$$\left.\boldsymbol{y}^{(k)} + \alpha^{(k)}\Delta\boldsymbol{y}^{(k)};\ \boldsymbol{s}^{(k)} + \alpha^{(k)}\Delta\boldsymbol{s}^{(k)}\right)$$

10: Set $k = k + 1$

11: **End while**

End algorithm

In the rest of this section, we prove that the complementary gap and the function f_μ decrease for a given displacement step. The proof of this result depends essentially on the following lemma.

Lemma 5.1 *Let* $(\boldsymbol{x};\boldsymbol{y};\boldsymbol{s}) \in int\,\mathcal{K}_\mathcal{J} \times \mathbb{R}^m \times int\,\mathcal{K}_\mathcal{J}$, $\left(\overline{\boldsymbol{x}};\boldsymbol{y};\underline{\boldsymbol{s}}\right)$ *be obtained by applying scaling to* $(\boldsymbol{x};\boldsymbol{y};\boldsymbol{s})$, *and* $\left(\overline{\Delta\boldsymbol{x}};\Delta\boldsymbol{y};\underline{\Delta\boldsymbol{s}}\right)$ *be a solution of the system* (8). *Then we have*

(a) $\overline{\Delta\boldsymbol{x}} \bullet \underline{\Delta\boldsymbol{s}} = 0$.

(b) $\overline{\boldsymbol{x}} \bullet \underline{\Delta\boldsymbol{s}} + \overline{\Delta\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} = \mathrm{trace}(\boldsymbol{h})$, *where* $\boldsymbol{h} = \sigma\mu\boldsymbol{e} - \overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}$ *such that* $\sigma \in (0,1)$ *and* $\mu = \frac{1}{r}\overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}}$.

(c) $\overline{\boldsymbol{x}}^+ \bullet \underline{\boldsymbol{s}}^+ = \left(1 - \alpha\left(1 - \frac{\sigma}{2}\right)\right)\ \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}},\ \ \forall\alpha \in \mathbb{R}$, *where* $\overline{\boldsymbol{x}}^+ = \overline{\boldsymbol{x}} + \alpha\overline{\Delta\boldsymbol{x}}$ *and* $\underline{\boldsymbol{s}}^+ = \underline{\boldsymbol{s}} + \alpha\underline{\Delta\boldsymbol{s}}$.

(d) $\boldsymbol{x}^+ \bullet \boldsymbol{s}^+ = \left(1 - \alpha\left(1 - \frac{\sigma}{2}\right)\right)\ \boldsymbol{x} \bullet \boldsymbol{s},\ \ \forall\alpha \in \mathbb{R}$, *where* $\boldsymbol{x}^+ = \boldsymbol{x} + \alpha\Delta\boldsymbol{x}$ *and* $\boldsymbol{s}^+ = \boldsymbol{s} + \alpha\Delta\boldsymbol{s}$.

Proof By the first two equations of the system (8), we get

$$\overline{\Delta\boldsymbol{x}} \bullet \underline{\Delta\boldsymbol{s}} = -\overline{\Delta\boldsymbol{x}} \bullet \underline{A}^{\mathsf{T}}\Delta\boldsymbol{y} = -\left(\underline{A}\overline{\Delta\boldsymbol{x}}\right)^{\mathsf{T}}\Delta\boldsymbol{y} = 0.$$

This proves item (a).

We prove item (b) by noting that

$$\begin{aligned}\mathrm{trace}(\boldsymbol{h}) &= \mathrm{trace}\left(\sigma\mu\boldsymbol{e} - \overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}\right)\\ &= \mathrm{trace}\left(\overline{\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}} + \overline{\Delta\boldsymbol{x}} \circ \underline{\boldsymbol{s}}\right)\\ &= \mathrm{trace}\left(\overline{\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}}\right) + \mathrm{trace}\left(\overline{\Delta\boldsymbol{x}} \circ \underline{\boldsymbol{s}}\right)\\ &= \overline{\boldsymbol{x}} \bullet \underline{\Delta\boldsymbol{s}} + \overline{\Delta\boldsymbol{x}} \bullet \underline{\boldsymbol{s}},\end{aligned}$$

where we used the last equation of the system (8) to obtain the first equality. To prove item (c), note that

$$\begin{aligned}\overline{\boldsymbol{x}}^+ \bullet \underline{\boldsymbol{s}}^+ &= \left(\overline{\boldsymbol{x}} + \alpha\overline{\Delta\boldsymbol{x}}\right) \bullet \left(\underline{\boldsymbol{s}} + \alpha\underline{\Delta\boldsymbol{s}}\right)\\ &= \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} + \alpha\left(\overline{\Delta\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} + \overline{\boldsymbol{x}} \bullet \underline{\Delta\boldsymbol{s}}\right) + \alpha^2\overline{\Delta\boldsymbol{x}} \bullet \underline{\Delta\boldsymbol{s}}\\ &= \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} + \tfrac{1}{2}\alpha\ \mathrm{trace}\left(\sigma\mu\boldsymbol{e} - \overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}\right)\\ &= \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} + \tfrac{1}{2}\alpha\sigma\mu\ \mathrm{trace}\,(\boldsymbol{e}) - \tfrac{1}{2}\alpha\ \mathrm{trace}\left(\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}\right)\\ &= \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} + \alpha\sigma\mu r - \alpha\overline{\boldsymbol{x}} \bullet \overline{\boldsymbol{s}}\\ &= \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} + \tfrac{1}{2}\alpha\sigma\overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} - \alpha\overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}}\\ &= \left(1 - \alpha\left(1 - \tfrac{\sigma}{2}\right)\right)\ \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}},\end{aligned}$$

where the third equality follows from items (a) and (b).

Finally, item (d) follows from item (c) and the fact that $\overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} = \boldsymbol{x} \bullet \boldsymbol{s}$ (see (9)), and similarly that $\overline{\boldsymbol{x}}^+ \bullet \underline{\boldsymbol{s}}^+ = \boldsymbol{x}^+ \bullet \boldsymbol{s}^+$. The proof is complete. □

The following result generalizes the corresponding one in [23, Lemmas 4.2 and 4.3].

Lemma 5.2 *Let* $(\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{s})$ *and* $\left(\boldsymbol{x}^+; \boldsymbol{y}^+; \boldsymbol{s}^+\right)$ *be strictly feasible solutions of the pair of problems* (P_μ, D_μ) *with* $\left(\boldsymbol{x}^+; \boldsymbol{y}^+; \boldsymbol{s}^+\right) = (\boldsymbol{x} + \alpha\Delta\boldsymbol{x};\ \boldsymbol{y} + \alpha\Delta\boldsymbol{y};\ \boldsymbol{s} + \alpha\Delta\boldsymbol{s})$*, where* α *is a displacement step and* $(\Delta\boldsymbol{x}; \Delta\boldsymbol{y}; \Delta\boldsymbol{s})$ *is the Newton direction. Then we have*

(a) $\boldsymbol{x}^+ \bullet \boldsymbol{s}^+ < \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}}$.
(b) $f_\mu(\boldsymbol{x}^+) < f_\mu(\boldsymbol{x})$.

Proof Note that

$$\boldsymbol{x}^+ \bullet \boldsymbol{s}^+ = \left(1 - \alpha\left(1 - \frac{\sigma}{2}\right)\right)\overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}} < \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}},$$

where the equality follows from item (d) of Lemma 5.1 and the strict inequality follows from $\left(1 - \alpha\left(1 - \frac{\sigma}{2}\right)\right) < 1$ (as $\alpha > 0$ and $\sigma \in (0, 1)$). This proves item (a).

To prove item (b), note that

$$f_\mu(\boldsymbol{x}^+) \simeq f_\mu(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} f_\mu(\boldsymbol{x}) \bullet (\boldsymbol{x}^+ - \boldsymbol{x}),$$

and hence

$$f_\mu(\boldsymbol{x}^+) - f_\mu(\boldsymbol{x}) \simeq \alpha\nabla_{\boldsymbol{x}} f_\mu(\boldsymbol{x}) \bullet \Delta\boldsymbol{x}.$$

Since

$$\nabla_{\boldsymbol{x}} f_\mu(\boldsymbol{x}) = -\nabla^2_{\boldsymbol{xx}} f_\mu(\boldsymbol{x})\Delta\boldsymbol{x},$$

we have

$$f_\mu(\boldsymbol{x}^+) - f_\mu(\boldsymbol{x}) \simeq -\alpha\ \Delta\boldsymbol{x} \bullet \nabla^2_{\boldsymbol{xx}} f_\mu(\boldsymbol{x})\Delta\boldsymbol{x} < 0,$$

where the strict inequality follows from the positive definiteness of the Hessian matrix $\nabla^2_{\boldsymbol{xx}} f_\mu(\boldsymbol{x})$ (as f_μ is strictly convex). Thus, $f_\mu(\boldsymbol{x}^+) < f_\mu(\boldsymbol{x})$. The proof is complete. □

6 Complexity analysis

In this section, we analyze the complexity of the proposed central trajectory algorithm for symmetric programming. More specifically, we prove that the iteration-complexity of Algorithm 5.1 is bounded by

$$O\left(\sqrt{r}\ln\left[\epsilon^{-1}\,\boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}\right]\right).$$

Our proof depends essentially on the following two lemmas.

Lemma 6.1 *Let* $(\boldsymbol{x};\boldsymbol{y};\boldsymbol{s}) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ$, $\left(\overline{\boldsymbol{x}};\boldsymbol{y};\underline{\boldsymbol{s}}\right)$ *be obtained by applying scaling to* $(\boldsymbol{x};\boldsymbol{y};\boldsymbol{s})$ *with* $\boldsymbol{h} = \sigma\mu\boldsymbol{e} - \overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}$, *and* $\left(\overline{\Delta\boldsymbol{x}};\Delta\boldsymbol{y};\underline{\Delta\boldsymbol{s}}\right)$ *be a solution of the system (8). For any* $\alpha \in \mathbb{R}$, *we set*

$$\begin{aligned}
&(\boldsymbol{x}(\alpha);\boldsymbol{y}(\alpha);\boldsymbol{s}(\alpha)) := (\overline{\boldsymbol{x}};\boldsymbol{y};\underline{\boldsymbol{s}}) + \alpha(\overline{\Delta\boldsymbol{x}};\Delta\boldsymbol{y};\underline{\Delta\boldsymbol{s}}),\\
&\mu(\alpha) := \tfrac{1}{r}\boldsymbol{x}(\alpha) \bullet \boldsymbol{s}(\alpha),\\
&\boldsymbol{v}(\alpha) := \boldsymbol{x}(\alpha) \circ \boldsymbol{s}(\alpha) - \mu(\alpha)\boldsymbol{e}.
\end{aligned}$$

Then

$$\boldsymbol{v}(\alpha) = (1-\alpha)(\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}} - \mu\boldsymbol{e}) + \alpha^2\, \overline{\Delta\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}}. \tag{15}$$

Proof Given $\alpha \in \mathbb{R}$, using item (c) of Lemma 5.1, we have

$$\boldsymbol{x}(\alpha) \bullet \boldsymbol{s}(\alpha) = (1 - \alpha + \alpha\sigma)\,\overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}}, \quad \text{and hence} \quad \mu(\alpha) = (1 - \alpha + \alpha\sigma)\mu.$$

Thus, we get

$$\begin{aligned}
\boldsymbol{v}(\alpha) &= \boldsymbol{x}(\alpha) \circ \boldsymbol{s}(\alpha) - \mu(\alpha)\boldsymbol{e}\\
&= \left(\overline{\boldsymbol{x}} + \alpha\overline{\Delta\boldsymbol{x}}\right) \circ \left(\underline{\boldsymbol{s}} + \alpha\underline{\Delta\boldsymbol{s}}\right) - (1 - \alpha + \alpha\sigma)\mu\boldsymbol{e}\\
&= (1-\alpha)(\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}} - \mu\boldsymbol{e}) + \alpha(\underbrace{\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}} - \sigma\mu\boldsymbol{e}}_{-\boldsymbol{h}})\\
&\quad + \alpha\underbrace{\left(\overline{\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}} + \overline{\Delta\boldsymbol{x}} \circ \underline{\boldsymbol{s}}\right)}_{\boldsymbol{h}} + \alpha^2\, \overline{\Delta\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}}\\
&= (1-\alpha)(\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}} - \mu\boldsymbol{e}) + \alpha^2\, \overline{\Delta\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}}.
\end{aligned}$$

This completes the proof. □

Lemma 6.2 *Let* $(\boldsymbol{x};\boldsymbol{y};\boldsymbol{s}) \in \mathcal{F}_P^\circ \times \mathcal{F}_D^\circ$, $\left(\overline{\boldsymbol{x}};\boldsymbol{y};\underline{\boldsymbol{s}}\right)$ *be obtained by applying scaling to* $(\boldsymbol{x};\boldsymbol{y};\boldsymbol{s})$ *such that* $\left\|\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}} - \mu\boldsymbol{e}\right\| \le \theta\mu$, *for some* $\theta \in [0,1)$ *and* $\mu > 0$. *Let also* $\left(\overline{\Delta\boldsymbol{x}};\Delta\boldsymbol{y};\underline{\Delta\boldsymbol{s}}\right)$ *be a solution of the system (8),* $\boldsymbol{h} = \sigma\mu\boldsymbol{e} - \overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}$, $\delta_{\boldsymbol{x}} := \mu\left\|\overline{\Delta\boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1}\right\|_F$, $\delta_{\boldsymbol{s}} := \left\|\overline{\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}}\right\|_F$. *Then, we have*

$$\delta_{\boldsymbol{x}}\delta_{\boldsymbol{s}} \le \frac{1}{2}\left(\delta_{\boldsymbol{x}}^2 + \delta_{\boldsymbol{s}}^2\right) \le \frac{\|\boldsymbol{h}\|_F^2}{2(1-\theta)^2}. \tag{16}$$

Proof By the last equation of system (8) and from the operator commutativity, we have

$$\boldsymbol{h} = \overline{\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}} + \overline{\Delta\boldsymbol{x}} \circ \underline{\boldsymbol{s}} = \overline{\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}} + \mu\overline{\Delta\boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1} + \left(\overline{\Delta\boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1}\right) \circ \left(\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}} - \mu\boldsymbol{e}\right).$$

It immediately follows that

$$\begin{aligned}\|\boldsymbol{h}\|_F &\geq \left\|\overline{\boldsymbol{x}} \circ \underline{\Delta \boldsymbol{s}} + \mu\, \overline{\Delta \boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1}\right\|_F - \left\|\overline{\Delta \boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1}\right\|_F \left\|\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}} - \mu \boldsymbol{e}\right\| \\ &\geq \left\|\overline{\boldsymbol{x}} \circ \underline{\Delta \boldsymbol{s}} + \mu\, \overline{\Delta \boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1}\right\|_F - \theta \delta_{\boldsymbol{x}} \\ &\geq \sqrt{\left\|\overline{\boldsymbol{x}} \circ \underline{\Delta \boldsymbol{s}}\right\|_F^2 + \left\|\mu\, \overline{\Delta \boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1}\right\|_F^2} - \theta \delta_{\boldsymbol{x}} \\ &= \sqrt{\delta_{\boldsymbol{x}}^2 + \delta_{\boldsymbol{s}}^2} - \theta \delta_{\boldsymbol{x}} \\ &\geq (1-\theta)\sqrt{\delta_{\boldsymbol{x}}^2 + \delta_{\boldsymbol{s}}^2},\end{aligned} \tag{17}$$

where the second inequality follows from the assumption that $\left\|\overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}} - \mu \boldsymbol{e}\right\| \leq \theta\mu$, and the third inequality follows from (3) the fact that

$$\begin{aligned}\left(\overline{\boldsymbol{x}} \circ \underline{\Delta \boldsymbol{s}}\right) \bullet \left(\overline{\Delta \boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1}\right) &= \operatorname{trace}\left(\left(\overline{\boldsymbol{x}} \circ \underline{\Delta \boldsymbol{s}}\right) \circ \left(\overline{\Delta \boldsymbol{x}} \circ \overline{\boldsymbol{x}}^{-1}\right)\right) \\ &= \operatorname{trace}\left(\underline{\Delta \boldsymbol{s}} \circ \overline{\Delta \boldsymbol{x}}\right) = \overline{\Delta \boldsymbol{x}} \bullet \underline{\Delta \boldsymbol{s}},\end{aligned}$$

which is essentially zero due to item (a) of Lemma 5.1.

The right-hand side inequality in (16) follows by noting that $(\delta_{\boldsymbol{x}} - \delta_{\boldsymbol{s}})^2 \geq 0$, and the left-hand side inequality in (16) follows from the last inequality in (17). The proof is complete. □

The following theorem analyzes the behavior of one iteration of Algorithm 5.1. This theorem generalizes [23, Theorem 7.1].

Theorem 6.1 *Let $\theta \in (0, 1)$ and $\delta \in \left[0, \sqrt{r}\right)$ be given such that*

$$\frac{\theta^2 + \delta^2}{2(1-\theta)^2\left(1 - \frac{\delta}{\sqrt{r}}\right)} \leq \theta \leq \frac{1}{2}. \tag{18}$$

Suppose that $\left(\overline{\boldsymbol{x}}; \boldsymbol{y}; \underline{\boldsymbol{s}}\right) \in \mathcal{N}_\theta(\mu)$ and let $\left(\overline{\Delta \boldsymbol{x}}; \Delta \boldsymbol{y}; \underline{\Delta \boldsymbol{s}}\right)$ denote the solution of system (8) with $\boldsymbol{h} = \sigma\mu\boldsymbol{e} - \overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}$ and $\sigma = 1 - \frac{\delta}{\sqrt{r}}$. Then, we have

(a) $\overline{\boldsymbol{x}}^+ \bullet \underline{\boldsymbol{s}}^+ = \left(1 - \frac{\delta}{\sqrt{r}}\right) \overline{\boldsymbol{x}} \bullet \underline{\boldsymbol{s}}$.
(b) $\left(\overline{\boldsymbol{x}}^+; \boldsymbol{y}^+; \underline{\boldsymbol{s}}^+\right) = \left(\overline{\boldsymbol{x}}; \boldsymbol{y}; \underline{\boldsymbol{s}}\right) + \left(\overline{\Delta \boldsymbol{x}}; \Delta \boldsymbol{y}; \underline{\Delta \boldsymbol{s}}\right) \in \mathcal{N}_\theta(\mu)$.
(c) $\left(\boldsymbol{x}^+; \boldsymbol{y}^+; \boldsymbol{s}^+\right) = (\boldsymbol{x}; \boldsymbol{y}; \boldsymbol{s}) + (\Delta \boldsymbol{x}; \Delta \boldsymbol{y}; \Delta \boldsymbol{s}) \in \mathcal{N}_\theta(\mu)$.

Proof Item (a) follows directly from item (c) of Lemma 5.1 with $\alpha = 1$ and $\sigma = 1 - \frac{\delta}{\sqrt{r}}$. We now prove item (b). Define

$$\mu^+ := \frac{1}{r}\, \overline{\boldsymbol{x}}^+ \bullet \underline{\boldsymbol{s}}^+ = \left(1 - \frac{\delta}{r}\right) \mu \tag{19}$$

and let $\left(\overline{\boldsymbol{x}}; \boldsymbol{y}; \underline{\boldsymbol{s}}\right) \in \mathcal{N}_\theta(\mu)$, we then have

$$\left\|\sigma\mu\boldsymbol{e}-\overline{\boldsymbol{x}}\circ\underline{\boldsymbol{s}}\right\|_F^2 \le \|(\sigma-1)\mu\boldsymbol{e}\|_F^2+\left\|\mu\boldsymbol{e}-\overline{\boldsymbol{x}}\circ\underline{\boldsymbol{s}}\right\|_F^2 \le \left((\sigma-1)^2r+\theta^2\right)\mu^2=\left(\delta^2+\theta^2\right)\mu^2. \tag{20}$$

Since $\|\overline{\boldsymbol{x}}\circ\underline{\boldsymbol{s}}-\mu\boldsymbol{e}\|\le\theta\mu$ and $\boldsymbol{h}=\sigma\mu\boldsymbol{e}-\overline{\boldsymbol{x}}\circ\underline{\boldsymbol{s}}$, using Lemma 6.2 it follows that

$$\left\|\overline{\Delta\boldsymbol{x}}\circ\overline{\boldsymbol{x}}^{-1}\right\|_F\left\|\overline{\boldsymbol{x}}\circ\underline{\Delta\boldsymbol{s}}\right\|_F\le\frac{\left\|\sigma\mu\boldsymbol{e}-\overline{\boldsymbol{x}}\circ\underline{\boldsymbol{s}}\right\|_F^2}{2(1-\theta)^2\mu}. \tag{21}$$

Defining $\boldsymbol{v}^+:=\boldsymbol{v}(1)=\overline{\boldsymbol{x}}^+\circ\underline{\boldsymbol{s}}^+-\mu^+\boldsymbol{e}$ and using (15) with $\alpha=1$, (21), (20), (18) and (19), we get

$$\begin{aligned}\|\boldsymbol{v}^+\|_F&=\left\|\overline{\Delta\boldsymbol{x}}\circ\underline{\Delta\boldsymbol{s}}\right\|_F\le\left\|\overline{\Delta\boldsymbol{x}}\circ\overline{\boldsymbol{x}}^{-1}\right\|_F\left\|\overline{\boldsymbol{x}}\circ\underline{\Delta\boldsymbol{s}}\right\|_F\\&\le\frac{\left\|\sigma\mu\boldsymbol{e}-\overline{\boldsymbol{x}}\circ\underline{\boldsymbol{s}}\right\|_F^2}{2(1-\theta)^2\mu}\le\frac{(\delta^2+\theta^2)\mu}{2(1-\theta)^2}\le\theta\left(1-\frac{\delta}{\sqrt{r}}\right)\mu=\theta\mu^+.\end{aligned}$$

Consequently,

$$\left\|\overline{\boldsymbol{x}}^+\circ\underline{\boldsymbol{s}}^+-\mu^+\boldsymbol{e}\right\|_F\le\theta\mu^+. \tag{22}$$

By using the right-hand side inequality in (16), and using (20) and (18), we have

$$\left\|\overline{\Delta\boldsymbol{x}}\circ\overline{\boldsymbol{x}}^{-1}\right\|_F\le\frac{\left\|\sigma\mu\boldsymbol{e}-\overline{\boldsymbol{x}}\circ\underline{\boldsymbol{s}}\right\|_F}{(1-\theta)\mu}\le\frac{\sqrt{\delta^2+\theta^2}}{(1-\theta)}\le\sqrt{2\theta\left(1-\frac{\delta}{\sqrt{r}}\right)}<1,$$

where the strict inequality follows from $\theta\le\frac{1}{2}$ and $0<1-\frac{\delta}{\sqrt{r}}<1$.

One can easily see that $\left\|\overline{\Delta\boldsymbol{x}}\circ\overline{\boldsymbol{x}}^{-1}\right\|_F<1$ implies that $\boldsymbol{e}+\overline{\Delta\boldsymbol{x}}\circ\overline{\boldsymbol{x}}^{-1}\succ\boldsymbol{0}$, and therefore

$$\overline{\boldsymbol{x}}^+=\overline{\Delta\boldsymbol{x}}+\overline{\boldsymbol{x}}=\left(\boldsymbol{e}+\overline{\Delta\boldsymbol{x}}\circ\overline{\boldsymbol{x}}^{-1}\right)\circ\overline{\boldsymbol{x}}\succ\boldsymbol{0}.$$

Note that, from (22), we have $\lambda_{\min}\left(\overline{\boldsymbol{x}}^+\circ\underline{\boldsymbol{s}}^+\right)\ge(1-\theta)\mu^+>0$, and therefore $\overline{\boldsymbol{x}}^+\circ\underline{\boldsymbol{s}}^+\succ\boldsymbol{0}$. Since $\overline{\boldsymbol{x}}^+\succ\boldsymbol{0}$ and $\overline{\boldsymbol{x}}^+$ and $\underline{\boldsymbol{s}}^+$ operator commute, we conclude that $\underline{\boldsymbol{s}}^+\succ\boldsymbol{0}$. Using the first equation of system (8), we get

$$\underline{A}\overline{\boldsymbol{x}}^+=\underline{A}\left(\overline{\boldsymbol{x}}+\overline{\Delta\boldsymbol{x}}\right)=\underline{A}\overline{\boldsymbol{x}}+\underline{A}\overline{\Delta\boldsymbol{x}}=\boldsymbol{b},\quad\text{and hence}\quad\overline{\boldsymbol{x}}^+\in\mathcal{F}_P^\circ.$$

By using the second equation of system (8), we get

$$\begin{aligned}\underline{A}^{\mathsf{T}}\boldsymbol{y}^++\overline{\boldsymbol{s}}^+&=\underline{A}^{\mathsf{T}}(\boldsymbol{y}+\Delta\boldsymbol{y})+\left(\overline{\boldsymbol{s}}+\overline{\Delta\boldsymbol{s}}\right)=\underline{A}^{\mathsf{T}}\boldsymbol{y}+\overline{\boldsymbol{s}}\\&\quad+\underline{A}^{\mathsf{T}}\Delta\boldsymbol{y}+\overline{\Delta\boldsymbol{s}}=\boldsymbol{c},\quad\text{and hence}\quad\left(\boldsymbol{y}^+;\overline{\boldsymbol{s}}^+\right)\in\mathcal{F}_D^\circ.\end{aligned}$$

Thus, in view of (22), we deduce that $\left(\overline{\boldsymbol{x}}^+;\boldsymbol{y}^+;\overline{\boldsymbol{s}}^+\right)\in\mathcal{N}_\theta^\circ(\mu)$. Item (b) is therefore established. Item (c) follows from item (b) and [1, Proposition 29]. The proof is now complete. □

Corollary 6.1 *Let θ and δ as given in Theorem 6.1 and $\left(\boldsymbol{x}^0; \boldsymbol{y}^0; \boldsymbol{s}^{(0)}\right) \in \mathcal{N}_\theta(\mu)$. Then Algorithm 5.1 generates a sequence of points $\left\{\left(\boldsymbol{x}^k; \boldsymbol{y}^k; \boldsymbol{s}^{(k)}\right)\right\} \subset \mathcal{N}_\theta(\mu)$ such that*

$$\boldsymbol{x}^{(k)} \bullet \boldsymbol{s}^{(k)} = \left(1 - \frac{\delta}{\sqrt{r}}\right)^k \boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}, \quad \forall k \geq 0.$$

Moreover, given a tolerance $\epsilon > 0$, Algorithm 5.1 computes an iterate $\left\{\left(\boldsymbol{x}^k; \boldsymbol{y}^k; \boldsymbol{s}^{(k)}\right)\right\}$ satisfying $\boldsymbol{x}^{(k)} \bullet \boldsymbol{s}^{(k)} \leq \epsilon$ in at most $K = O\left(\sqrt{r} \ln\left[\epsilon^{-1} \boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}\right]\right)$ iterations.

Proof Looking recursively at item (a) of Theorem 6.1, for each k we have that

$$\boldsymbol{x}^{(k)} \bullet \boldsymbol{s}^{(k)} = \left(1 - \frac{\delta}{\sqrt{r}}\right)^k \boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)} \leq \epsilon.$$

By taking natural algorithm of both sides, we get

$$k \ln\left(1 - \frac{\delta}{\sqrt{r}}\right) \leq \ln\left(\frac{\epsilon}{\boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}}\right),$$

which holds only if

$$k\left(-\frac{\delta}{\sqrt{r}}\right) \leq \ln\left(\frac{\epsilon}{\boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}}\right), \quad \text{or equivalently,} \quad k \geq K \geq \left\lceil \delta^{-1}\sqrt{r}\ \ln\left(\frac{\boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}}{\epsilon}\right)\right\rceil.$$

The result is established. □

7 Numerical experiments

In this section, we present two numerical examples for second-order cone programs to demonstrate the efficiency of our algorithm. Our numerical experiments are carried out on a PC with Intel(R) Dual CPU at 2.20 GHz and 2 GB of physical memory. The PC runs MATLAB Version: 7.4.0.287 (R2007a) on Windows XP Enterprise 32-bit operating system. We denote by

1st Sel.: the first selection for calculating the displacement step,
2nd Sel.: the second selection for calculating the displacement step,
3rd Sel.: the third selection for calculating the displacement step,
4th Sel.: the fourth selection for calculating the displacement step,
(n, m): the size of problems,
Iter.: the number of iterations taken to obtain the optimal solution,
CPU(s): the time (in seconds) required to obtain the optimal solution.

We point out that the matrices used in our examples have full row rank. In all of our experiments, we used the NT direction for choosing the scaling vector

$\boldsymbol{p}^{(k)} \in C\left(\boldsymbol{x}^{(k)}, \boldsymbol{s}^{(k)}\right)$ because this direction takes the advantage of being primal-dual symmetric. The following example is taken from the literature, see [15, Section 6].

Example 7.1 Let n and m be positive integers such that $n = 2m$. We consider the second-order cone programming problem and its dual

$$(P) \quad \begin{array}{ll} \min & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \in \mathcal{E}^n_+; \end{array} \qquad (D) \quad \begin{array}{ll} \max & \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.} & \boldsymbol{c} - A^{\mathsf{T}}\boldsymbol{y} \in \mathcal{E}^n_+, \end{array}$$

where $\mathcal{E}^n_+$ is the second-order cone defined in Example 2.3, and

$$\begin{array}{l} \boldsymbol{c} = 10\boldsymbol{e} - 2\,\mathbf{ones}(n,1) + 4\,\mathbf{rand}(n,1) \in \mathcal{E}^n, \\ \boldsymbol{b} = 10\boldsymbol{e} - 2\,\mathbf{ones}(m,1) + 4\,\mathbf{rand}(m,1) \in \mathbb{R}^m, \\ A = \left[\hat{A} \vdots \text{Randn}(m, n-m)\right] \in \mathbb{R}^{m\times n}, \end{array} \quad \text{and} \quad \hat{a}_{ij} = \begin{cases} 2 & \text{if } i = j-1, \\ 100 & \text{if } i = j, \\ -2 & \text{if } i = j+1, \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } 1 \le i, j \le m.$$

Here, $\mathbf{ones}(\ell, 1)$ is a vector of ones of length ℓ. We take $\boldsymbol{x}^{(0)} = \boldsymbol{e} \in \mathcal{E}^n$ and $\boldsymbol{y}^{(0)} = \boldsymbol{0} \in \mathbb{R}^m$ as our initial strictly feasible points. We also take $\epsilon = 10^{-6}, \sigma = 0.1$ and $\rho = 0.99$. As a well-known interior-point method for solving second-order cone programming, MOSEK [28] and SDPT3 [29] solvers were used in our experiments for comparison purposes. The numerical results related to this example are displayed in Table 1.

Due to the similarity in results between Examples 7.1 and 7.2, we analyze the numerical results of Table 1 after Example 7.2.

Example 7.2 In this example, the proposed primal-dual central trajectory algorithm was tested on randomly generated second-order cone programming problems. We generate a random matrix $A \in \mathbb{R}^{n\times m}$ with full row rank and random vectors $\boldsymbol{x}, \boldsymbol{s} \in \mathcal{E}^n_+$ and $\boldsymbol{y} \in \mathbb{R}^m$. We let $\boldsymbol{b} = A\boldsymbol{x}$ and $\boldsymbol{c} = A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s}$. Let $\boldsymbol{x}^{(0)} = \boldsymbol{e} \in \mathcal{E}^n$, $\boldsymbol{y}^{(0)} = \boldsymbol{0} \in \mathbb{R}^m$ and $\boldsymbol{s}^{(0)} = \boldsymbol{e} \in \mathcal{E}^n$ be initial points. Since the set of strictly feasible solutions of (P) and (D) are non-empty, the generated test problems (P) and (D) have optimal solutions and their optimal values are equal. The parameters used in Algorithm 5.1 were as follows: $\epsilon = 1.3500e - 06, \sigma = 0.1$ and $\rho = 0.99$.

In our experiments, we generated test problems with size $n(= 2m)$ from 100 to 1200 and $r = 1$. The random test problems of each n are generated 5 times, and hence we have 60 random problems in total. In this example, we use MOSEK [28] and SDPT3 [29] solvers in our experiments for comparison purposes as another well-known interior-point method for solving second-order cone programming. We present the results of our numerical results in Table 2. Note that the values of "Iter" and "CPU(s)" are the average of 5 runs for each n.

In view of Tables 1 and 2, we can see that Algorithm 5.1 with the first, second or fourth selections are able to give an optimal solution with a favorable running time and requires less number of iterations than the MOSEK and SDPT3 solvers; this is

Table 1 The numerical results of Example 7.1

Prob. size	1st Sel.		2nd Sel.		3rd Sel.		4th Sel.		MOSEK		SDPT3	
(m,n)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)
(5, 10)	5	0.0078	6	0.0062	9	0.1113	5	0.0058	6	0.0052	7	0.0050
(10, 20)	7	0.0178	7	0.0142	11	0.1235	6	0.0094	6	0.0081	7	0.0073
(20, 40)	6	1.0160	7	0.0186	13	0.1732	6	0.0139	7	0.0167	8	0.0145
(40, 80)	8	0.0456	9	0.0617	18	0.2718	7	0.0336	9	0.0528	8	0.0436
(80, 160)	12	0.1900	11	0.1862	21	0.3675	9	0.1414	11	0.1677	11	0.1510
(120, 240)	15	0.2398	16	0.2405	34	0.8152	13	0.2273	16	0.2194	18	0.2107

Table 2 The numerical results of Example 7.2

Prob. size	1st Sel.		2nd Sel.		3rd Sel.		4th Sel.		MOSEK		SDPT3	
(m,n)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)
(50,100)	5.6	0.0931	5.6	0.0993	8.3	0.1539	5.2	0.0881	6.1	0.1259	6.5	0.1132
(100,200)	5.8	0.3107	5.7	0.2981	8.4	0.6913	5.4	0.2840	6.3	0.5439	6.6	0.4347
(150,300)	6.1	1.0134	6.2	0.9408	9.4	1.3423	5.8	0.7761	7.1	1.1901	6.6	1.1759
(200,400)	6.3	2.2150	6.4	2.0935	8.9	2.8031	6.2	1.8420	7.0	2.3494	7.2	2.1036
(250,500)	6.4	2.6019	6.5	2.7142	9.5	3.2401	6.1	2.2245	7.2	3.1780	7.5	2.9518
(300,600)	6.4	3.8210	6.4	3.9101	9.8	6.1309	6.3	3.2091	7.1	5.2717	7.9	4.7671
(350,700)	6.5	6.0134	6.6	6.2013	9.4	7.8431	6.4	5.5219	7.8	7.5198	7.7	6.9383
(400,800)	6.8	8.1394	6.7	8.3405	9.7	9.8410	6.4	7.3490	7.8	11.841	8.4	9.2521
(450,900)	6.8	10.951	6.8	11.302	10.0	14.028	6.7	9.1918	8.1	13.990	8.5	12.3401
(500,1000)	6.9	15.213	6.9	16.029	10.2	18.721	6.7	13.879	7.9	18.084	7.9	17.310
(550,1100)	7.0	16.090	7.2	16.881	11.1	22.411	6.9	15.109	7.9	21.468	8.3	19.894
(600,1200)	7.3	20.132	7.4	21.094	11.4	27.913	7.1	18.938	8.8	25.567	8.7	23.431

probably due to the fact that these solvers implement infeasible algorithms and use infeasible initial points. We can also see that the fourth selection for calculating the displacement step has remarkable superiority to other three selections in terms of number of iterations and running time. It is interesting to note that our conclusion for second-order cone programming problems exactly matches the same conclusion obtained from the numerical experiments in [23, Section 5] for solving semidefinite programming problems.

8 Conclusions

In this paper, we have presented a primal-dual central trajectory interior-point method for solving symmetric programming problem. We have proven the convergence of the optimal solution of the corresponding perturbed problem to the optimal solution of the original problem when the barrier parameter goes to zero. Then, we have applied Newton's method to find a new iterative point by computing a good descent direction. The inconvenience lies in the high computational cost motivated us to avoid using several methods, such as the line search methods, to calculate the displacement step. Alternatively, in this paper, we have proposed a new approach based on four new selections to calculate the displacement step.

After stating the central trajectory algorithm, we have analyzed the convergence of the proposed algorithm and have proven that the complexity for short-step is bounded by $O(\sqrt{r}\ln[\epsilon^{-1}\boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}])$ iterations, where r is the rank of the Cartesian product of the corresponding Euclidean Jordan algebras. Our numerical results for second-order cone program have demonstrated the efficiency of our approach and have shown the convergence of the four proposed selections to the optimal solution of the problem. By looking at the computed time and number of iterations, the numerical results on second-order cone programs have shown that the fourth selection is the best selection that can be chosen to reach the optimal solution.

Acknowledgements A part of this work was performed while the author was visiting The Center for Applied and Computational Mathematics at Rochester Institute of Technology, NY, USA. The work of the author was supported in part by the Deanship of Scientific Research at the University of Jordan. The author thanks the two anonymous expert referees for their valuable suggestions. The constructive comments from the referees have greatly enhanced the paper.

References

1. Schmieta, S.H., Alizadeh, F.: Extension of primal-dual interior point methods to symmetric cones. Math. Program. Ser. A **96**, 409–438 (2003)
2. Schmieta, S.H., Alizadeh, F.: Associative and Jordan algebras, and polynomial time interior point algorithms for symmetric cones. Math. Oper. Res. **26**(3), 543–564 (2001)
3. Alizadeh, F., Goldfarb, D.: Second-order cone programming. Math. Program. Ser. B **95**, 3–51 (2003)
4. Alzalg, B.: Stochastic second-order cone programming: application models. Appl. Math. Model. **36**, 5122–5134 (2012)
5. Todd, M.J.: Semidefinite optimization. ACTA Numer. **10**, 515–560 (2001)
6. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Rev. **38**, 49–95 (1996)
7. Nesterov, Y.E., Todd, M.J.: Primal-dual interior-point methods for self-scaled cones. SIAM J. Optim. **8**(2), 324–364 (1998)

8. Alzalg, B., Ariyawansa, K.A.: Logarithmic barrier decomposition-based interior point methods for stochastic symmetric programming. J. Math. Anal. Appl. **409**, 973–995 (2014)
9. Alzalg, B., Maggiono, F., Vitali, S.: Homogeneous self-dual methods for symmetric cones under uncertainty. Far East J. Math. Sci. **99**(11), 1603–1778 (2016)
10. Benterki, D., Leulmi, A.: An improving procedure of the interior projective method for linear programming. Appl. Math. Comput. **199**, 811–819 (2008)
11. Dodani, M., Babu, A.: Karmarkar's projective method for linear programming: a computational appraisal. Comput. Ind. Eng. **16**, 189–206 (1989)
12. Todd, M., Wang, Y.: On combined phase 1-phase 2 projective methods for linear programming. Algorithmica **9**, 64–83 (1993)
13. Zhang, Y.: On extending primal-dual interior-point algorithms from linear programming to semidefinite programming. SIAM J. Optim. **8**, 356–386 (1998)
14. Hans, Y.-J., Mittelmann, D.: Interior point methods for second-order cone programming and OR applications. Comput. Optim. Appl. **28**, 255–285 (2004)
15. Tang, J., He, G., Dong, L., Fang, L.: A new one-step smoothing newton method for second-order cone programming. Appl. Math. **57**, 311–331 (2012)
16. Alzalg, B.: Homogeneous self-dual algorithms for stochastic second-order cone programming. J. Optim. Theory Appl. **163**(1), 148–164 (2014)
17. Alzalg, B.: Decomposition-based interior point methods for stochastic quadratic second-order cone programming. Appl. Math. Comput. **249**, 1–18 (2014)
18. Alzalg, B.: Volumetric barrier decomposition algorithms for stochastic quadratic second-order cone programming. Appl. Math. Comput. **256**, 494–508 (2015)
19. Kettab, S., Benterki, D.: A relaxed logarithmic barrier method for semidefinite programming. RAIRO Oper. Res. **42**, 555–568 (2015)
20. Crouzeix, J.P., Merikhi, B.: A logarithm barrier method for semidefinite programming. RAIRO Oper. Res. **42**, 123–139 (2008)
21. Monteiro, R.D.: Primal-dual path-following algorithms for semidefinite programming. SIAM J. Optim. **7**, 663–678 (1997)
22. Helmberg, C., Rendl, F., Vanderbei, R.J., Wolkowicz, H.: An interior-point methods for stochastic semidefinite programming. SIAM J. Optim. **6**, 342–361 (1996)
23. Touil, I., Benterki, D., Yassine, A.: A feasible primal-dual interior point method for linear semidefinite programming. J. Comput. Appl. Math. **312**, 216–230 (2017)
24. Kebbiche, Z., Keraghel, A., Yassine, A.: Extension of a projective interior point method for linearly constrained convex programming. Appl. Math. Comput. **193**, 553–559 (2007)
25. Gahinet, P., Nemirovski, A.: The projective method for solving linear matrix inequalities. Math. Prog. **77**, 163–190 (1997)
26. Behling, R., Gonzaga, C., Haeser, G.: Primal-dual relationship between Levenberg–Marquardt and central trajectories for linearly constrained convex optimization. J. Optim. Theory Appl. **162**, 705–717 (2014)
27. Gould, N., Orban, D., Robinson, D.: Trajectory-following methods for large-scale degenerate convex quadratic programming. Math. Prog. Comput. **5**, 113–142 (2013)
28. MOSEK is an optimization software designed to solve large-scale mathematical optimization problems. http://www.mosek.com/. Accessed 5 Oct 2017
29. Toh, K.C., Todd, M.J., Tutuncu, R.H.: SDPT3 version 4.0–a MATLAB software for semidefinite-quadratic-linear programming, (2009). Available online at: http://www.math.nus.edu.sg/~mattohkc/sdpt3.html
30. Wolkowicz, H., Styan, G.-P.-H.: Bounds for eigenvalues using traces. Linear Algebra Appl. **29**, 471–506 (1980)
31. Faraut, J., Korányi, A.: Analysis on Symmetric Cones. Oxford University Press, Oxford (1994)
32. Kojima, M., Shindoh, S., Hara, S.: Interior-point methods for the monotone linear complementarity problem in symmetric matrices. SIAM J. Optim. **7**(9), 86–125 (1997)
33. Lütkepohl, H.: Handbook of Matrices. Humboldt-Universität zu Berlin, Germany (1996)