computation

MDPI

Article

An Algebraic-Based Primal–Dual Interior-Point Algorithm for Rotated Quadratic Cone Optimization

Karima Tamsaouete † **and Baha Alzalg** *,†

Department of Mathematics, The University of Jordan, Amman 11942, Jordan
* Correspondence: b.alzalg@ju.edu.jo
† These authors contributed equally to this work.

Abstract: In rotated quadratic cone programming problems, we minimize a linear objective function over the intersection of an affine linear manifold with the Cartesian product of rotated quadratic cones. In this paper, we introduce the rotated quadratic cone programming problems as a "self-made" class of optimization problems. Based on our own Euclidean Jordan algebra, we present a glimpse of the duality theory associated with these problems and develop a special-purpose primal–dual interior-point algorithm for solving them. The efficiency of the proposed algorithm is shown by providing some numerical examples.

Keywords: rotated quadratic cone; Jordan algebras; interior-point methods; convex programming

check for updates

Citation: Tamsaouete, K.; Alzalg, B. An Algebraic-Based Primal–Dual Interior-Point Algorithm for Rotated Quadratic Cone Optimization. *Computation* **2023**, *11*, 50. https://doi.org/10.3390/computation11030050

Academic Editor: Demos T. Tsahalis

Received: 21 January 2023
Revised: 18 February 2023
Accepted: 22 February 2023
Published: 2 March 2023

1. Introduction

Rotated quadratic cone programming (RQCP) problems are convex conic optimization problems [1–6] in which we minimize a linear objective function over the intersection of an affine linear manifold with the Cartesian product of rotated quadratic cones, where the nth-dimensional rotated quadratic cone is defined as

$$\mathcal{R}^n_+ \triangleq \{\boldsymbol{x} = (x_1; x_2; \bar{x}) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}^{n-2} : x_1^{1/2} x_2^{1/2} \geq \|\bar{x}\|\}, \tag{1}$$

where, $\mathbb{R}_+$ is the set of positive real numbers and $\|\cdot\|$ is the Euclidean norm.

Many optimization problems are formulated as RQCPs (see Section 2.3 in [7] and Section 4 in [8], for example, but not limited to the problem of minimizing the harmonic mean of positive affine functions, the problem of maximizing the geometric mean of non-negative affine functions, the logarithmic Tchebychev approximation problem, problems involving fractional quadratic functions, problems with inequalities involving rational powers, problems with inequalities involving p-norms, and problems involving pairs of quadratic forms (also called minimum-volume covering ellipsoid problems).

It is known that the rotated quadratic cone is converted into the second-order cone under a linear transformation. In fact, the restricted hyperbolic constraint $x_1 x_2 \geq \|\bar{x}\|^2$ is equivalent to the set of linear and second-order cone constraints: $u = x_1 + x_2, v = x_1 - x_2, \boldsymbol{w} = (v; 2\bar{x})$, and $u \geq \|\boldsymbol{w}\|$. Based on this observation, all earlier work on RQCP problems has converted them to second-order cone programming problems, but while doing this can be easier than developing special-purpose algorithms for solving RQCPs, this approach may not always be the cheapest one in terms of computational cost.

Mathematical optimization together with evolutionary algorithms are today a state-of-the-art methodology in solving hard problems in machine learning and artificial intelligence, see for example [9–11]. Going back years in time, the introduction of the interior-point methods (IPMs) during in the 1980s perhaps was one of the most notable developments in the field of mathematical programming since its origination in the 1940s. Karmarker [12] was the first to propose them for linear programming, where their work generated a stir due to the superiority of their polynomial complexity results over that of the simplex method. It then seemed natural to expand these methods created for linear programming to solve nonlinear programs.

Nesterov and Nemirovskii [3] laid the groundwork for IPMs to solve convex programming problems, where primal (and dual) IPMs based on the so-called self-concordant barrier functions were taken into consideration. Nesterov and Todd [4] later presented symmetric primal–dual IPMs for problems over a specific class of cones termed as self-scaled cones, allowing for a symmetric approach to the primal and dual problems.

We point out that Nesterov and Todd's work [4] did not take a Jordan algebraic approach, but rather Güler's work [13] is credited with being the first to link Jordan algebras and optimization. Güler [13] noted that the family of self-scaled cones are the same as the family of the so-called symmetric cones for which a thorough classification theory is available [14]. The characteristics of these algebras act as a key toolkit for the analysis of IPMs for optimization over symmetric cones. Due to their diverse applications, the most important classes of symmetric cone optimization problems are linear programming, second-order cone programming [7], and semi-definite programming [15] (see also Part IV in [16] which gives a thorough presentation of these three classes of optimization problems). Several IPMs have been developed for these classes of conic optimization problems; for example, [2,7,15,17–24].

There are two classes of IPMs for solving linear and non-linear convex optimization problems. The first class solely uses dual or primal methods (see, for example, [25–27]). The second class is based on primal–dual methods, which were developed by [23] and [24] and are more useful and efficient than the first. These methods involve applying Newton's method to the Karush–Kuhn–Tucker (KKT) system up until a convergence condition is satisfied.

In [28], the authors have set up the Euclidean Jordan algebra (EJA) associated with the rotated quadratic cone and presented several spectral and algebraic characteristics of this EJA, where the authors have found that the rotated quadratic cone is the cone of squares of some EJAs (confirming that it is a symmetric cone). To our knowledge, no specialized algorithms for RQCPs that make use of the EJA of the underlying rotated quadratic cones. This paper is an attempt to introduce RQCPs as another self-contained paradigm of symmetric cone optimization, where we introduce RQCP as a "self-made" class of optimization problems and develop special purpose primal–dual interior-point algorithms (second class of IPMs) for solving RQCP problems based on the EJA in [28], which offers a useful set of tools for the analysis of IPMs related to RQCPs.

The so-called commutative class of primal–dual IPMs was designed by Monteiro and Zhang [29] for semi-definite programming, and by Alizadeh and Goldfarb [7] for second-order cone programming, and then extended by Schmieta and Alizadeh [30] for symmetric cone programming. This paper uses the machinery of EJA built in [28] to carry out an extension of this commutative class to RQCP. We prove polynomial complexities for versions of the short-, semi-long-, and long-step path-following IPMs using NT, HRVW/KSH/M, or dual HRVW/KSH/M directions (equivalent to NT, XS, or SX directions in semi-definite programming).

This paper is organized as follows: In Section 2, we calculate the derivatives of the logarithmic barrier function associated with the rotated quadratic cone and prove the corresponding self-concordance property. The formulation of the RQCP problems with the optimality conditions are provided in Section 3. Section 4 applies Newton's method and discusses the commutative direction. The proposed path-following algorithm for RQCP and its complexity are given in Section 5. Section 6 shows the efficiency of the proposed algorithm by providing some numerical results. We close this paper with Section 7, which contains some concluding remarks.

2. The Algebra and the Logarithmic Barrier of the Rotated Quadratic Cone

In Table 1, we summarize the Jordan algebraic notions associated with the cone $\mathcal{R}^n_+$.

Table 1. The algebraic notions and concepts associated with $\mathcal{R}^n_+$ (for more details, refer to [28]).

Notion/Concept	Definition
In single-setting	
Space	$\mathcal{R}^n \triangleq \{x = (x_1; x_2; \bar{x}) : x_1, x_2 \in \mathbb{R}_+, \bar{x} \in \mathbb{R}^{n-2}\}$
Rotated quadratic cone	$\mathcal{R}^n_+ \triangleq \{\boldsymbol{x} \in \mathcal{R}^n : x_1^{1/2} x_2^{1/2} \geq \lVert\bar{\boldsymbol{x}}\rVert\}$
Half-identity matrix	$H_n \triangleq \begin{bmatrix} \frac{1}{2} & 0 & \mathbf{0}^\mathsf{T} \\ 0 & \frac{1}{2} & \mathbf{0}^\mathsf{T} \\ \mathbf{0} & \mathbf{0} & I_{n-2} \end{bmatrix}$
Rotation–reflection matrix	$R_n \triangleq \begin{bmatrix} 0 & 1 & \mathbf{0}^\mathsf{T} \\ 1 & 0 & \mathbf{0}^\mathsf{T} \\ \mathbf{0} & \mathbf{0} & -I_{n-2} \end{bmatrix}$
Positive semi-definiteness	$\boldsymbol{x} \succeq_{\mathcal{R}^n_+} \mathbf{0}$ (or simply $\boldsymbol{x} \succeq \mathbf{0}$), which means that $\boldsymbol{x} \in \mathcal{R}^n_+$, i.e., $x_1^{1/2} x_2^{1/2} \geq \lVert\bar{\boldsymbol{x}}\rVert$
Positive definiteness	$\boldsymbol{x} \succ_{\mathcal{R}^n_+} \mathbf{0}$ (or simply $\boldsymbol{x} \succ \mathbf{0}$), which means that $\boldsymbol{x} \in \text{Int}(\mathcal{R}^n_+)$, i.e., $x_1^{1/2} x_2^{1/2} > \lVert\bar{\boldsymbol{x}}\rVert$
Eigenvalues	$\lambda_{1,2}(\boldsymbol{x}) \triangleq \boldsymbol{x}_1 \boldsymbol{x}_2 \pm \lVert\bar{\boldsymbol{x}}\rVert^2$
Eigenvectors	$\boldsymbol{c}_{1,2}(\boldsymbol{x}) \triangleq \left(\frac{1}{2} \pm \frac{x_1 - x_2}{2\sqrt{(x_1-x_2)^2 + 4\bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{x}}}};\ \frac{1}{2} \mp \frac{x_1 - x_2}{2\sqrt{(x_1-x_2)^2 + 4\bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{x}}}};\ \frac{\pm\bar{\boldsymbol{x}}}{\sqrt{(x_1-x_2)^2 + 4\bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{x}}}}\right)$
Trace	$\text{trace}(\boldsymbol{x}) \triangleq \lambda_1(\boldsymbol{x}) + \lambda_2(\boldsymbol{x}) = \boldsymbol{x}_1 + \boldsymbol{x}_2$
Determinant	$\det(\boldsymbol{x}) \triangleq \lambda_1(\boldsymbol{x})\lambda_2(\boldsymbol{x}) = \boldsymbol{x}_1\boldsymbol{x}_2 - \lVert\bar{\boldsymbol{x}}\rVert^2$
Identity	$\boldsymbol{e} \triangleq \boldsymbol{c}_1(\boldsymbol{x}) + \boldsymbol{c}_2(\boldsymbol{x}) = (1; 1; \mathbf{0})$
Spectral decomposition of $f(x)$	$f(\boldsymbol{x}) \triangleq f(\lambda_1(\boldsymbol{x}))c_1(\boldsymbol{x}) + f(\lambda_2(\boldsymbol{x}))c_2(\boldsymbol{x})$; f is any real valued continuous func.
Inverse	$\boldsymbol{x}^{-1} \triangleq \lambda_1^{-1}(\boldsymbol{x})\boldsymbol{c}_1(\boldsymbol{x}) + \lambda_2^{-1}(\boldsymbol{x})\boldsymbol{c}_2(\boldsymbol{x}) = \frac{1}{\det(\boldsymbol{x})} R\boldsymbol{x}$
Linear representation	$\mathrm{L}(\boldsymbol{x}) \triangleq \begin{bmatrix} x_1 & 0 & \bar{\boldsymbol{x}}^\mathsf{T} \\ 0 & x_2 & \bar{\boldsymbol{x}}^\mathsf{T} \\ \frac{1}{2}\bar{\boldsymbol{x}} & \frac{1}{2}\bar{\boldsymbol{x}} & \frac{1}{2}(x_1 + x_2)I_{n-2} \end{bmatrix}$
Quadratic representation	$\mathrm{Q}_{\boldsymbol{x}} \triangleq 2\mathrm{L}(\boldsymbol{x})^2 - \mathrm{L}(\boldsymbol{x}^2) = \begin{bmatrix} x_1^2 & \lVert\bar{\boldsymbol{x}}\rVert^2 & 2x_1\bar{\boldsymbol{x}}^\mathsf{T} \\ \lVert\bar{\boldsymbol{x}}\rVert^2 & x_2^2 & 2x_2\bar{\boldsymbol{x}}^\mathsf{T} \\ x_1\bar{\boldsymbol{x}} & x_2\bar{\boldsymbol{x}} & 2\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^\mathsf{T} + \det(\boldsymbol{x})I_{n-2} \end{bmatrix}$
Quadratic operator $\mathrm{Q}_{\boldsymbol{x},\boldsymbol{y}} : \mathcal{R}^n \longrightarrow \mathcal{R}^n$	$\mathrm{Q}_{\boldsymbol{x},\boldsymbol{y}} \triangleq \mathrm{L}(\boldsymbol{x})L(\boldsymbol{y}) + L(\boldsymbol{y})\mathrm{L}(\boldsymbol{x}) - L(\boldsymbol{x} \circ \boldsymbol{y})$ $= \begin{bmatrix} x_1y_1 & \bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{y}} & x_1\bar{\boldsymbol{y}}^\mathsf{T} + y_1\bar{\boldsymbol{x}}^\mathsf{T} \\ \bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{y}} & x_2y_2 & x_2\bar{\boldsymbol{y}}^\mathsf{T} + y_2\bar{\boldsymbol{x}}^\mathsf{T} \\ \frac{y_1}{2}\bar{\boldsymbol{x}} + \frac{x_1}{2}\bar{\boldsymbol{y}} & \frac{y_2}{2}\bar{\boldsymbol{x}} + \frac{x_2}{2}\bar{\boldsymbol{y}} & \bar{\boldsymbol{x}}\bar{\boldsymbol{y}}^\mathsf{T} + \bar{\boldsymbol{y}}\bar{\boldsymbol{x}}^\mathsf{T} + \left(\frac{x_1y_2 + x_2y_1}{2} - \bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{y}}\right)I_{n-2} \end{bmatrix}$
Jordan product $\circ : \mathcal{R}^n \times \mathcal{R}^n \longrightarrow \mathcal{R}^n$	$\boldsymbol{x} \circ \boldsymbol{y} \triangleq \mathrm{L}(\boldsymbol{x})\boldsymbol{y} = \left(x_1y_1 + \bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{y}};\ x_2y_2 + \bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{y}};\ \frac{1}{2}(y_1 + y_2)\bar{\boldsymbol{x}} + \frac{1}{2}(x_1 + x_2)\bar{\boldsymbol{y}}\right)$
Inner product $\bullet : \mathcal{R}^n \times \mathcal{R}^n \longrightarrow \mathbb{R}$	$\boldsymbol{x} \bullet \boldsymbol{y} \triangleq\ \frac{1}{2}\text{trace}\ (\boldsymbol{x} \circ y) = \boldsymbol{x}^\mathsf{T}H\boldsymbol{y} = \frac{1}{2}(x_1y_1 + x_2y_2) + \bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{y}}$
Frobenius norm	$\lVert\boldsymbol{x}\rVert_\mathrm{F}\ \triangleq \sqrt{\lambda_1^2(\boldsymbol{x}) + \lambda_2^2(\boldsymbol{x})} = \sqrt{x_1^2 + x_2^2 + 2\lvert\lvert\bar{\boldsymbol{x}}\rvert\rvert^2} = \sqrt{2\,\boldsymbol{x} \bullet \boldsymbol{x}}$
Rank	$\mathsf{rk}(\mathcal{R}^n) \triangleq 2$
In block-setting (r blocks)	
Space	$\mathcal{R} \triangleq \mathcal{R}^{n_1} \times \mathcal{R}^{n_2} \times \cdots \times \mathcal{R}^{n_r}$
Cone	$\mathcal{R}_+ \triangleq \mathcal{R}^{n_1}_+ \times \mathcal{R}^{n_2}_+ \times \cdots \times \mathcal{R}^{n_r}_+$
Elements/vectors	$\boldsymbol{x} = (\boldsymbol{x}^{(1)}; \boldsymbol{x}^{(2)}; \ldots; \boldsymbol{x}^{(r)})$, with $\boldsymbol{x}^{(i)} \in \mathcal{R}^{n_i}_+$ for each $i = 1, 2, \ldots, r$
Half-identity matrix	$H \triangleq H_{n_1} \oplus H_{n_2} \oplus \cdots \oplus H_{n_r}$
Rotation–reflection matrix	$R \triangleq R_{n_1} \oplus R_{n_2} \oplus \cdots \oplus R_{n_r}$
Positive semi-definiteness	$\boldsymbol{x} \succeq \mathbf{0}$, which means $\boldsymbol{x} \in \mathcal{R}^n_+$, i.e., $\boldsymbol{x}^{(i)} \in \mathcal{R}^{n_i}_+$ for each $i = 1, 2, \ldots, r$
Positive definiteness	$\boldsymbol{x} \succ \mathbf{0}$, which means $\boldsymbol{x} \in \text{Int}(\mathcal{R}^n_+)$, i.e., $\boldsymbol{x}^{(i)} \in \text{Int}(\mathcal{R}^{n_i}_+)$ for each $i = 1, 2, \ldots, r$
Trace	$\text{trace}(\boldsymbol{x}) \triangleq \sum_{i=1}^r \text{trace}(\boldsymbol{x}^{(i)})$
Determinant	$\det(\boldsymbol{x}) \triangleq \prod_{i=1}^r \det(\boldsymbol{x}^{(i)})$
Identity	$\boldsymbol{e} \triangleq (\boldsymbol{e}^{(1)}; \boldsymbol{e}^{(2)}; \ldots; \boldsymbol{e}^{(r)})$.
Vector function $f(x)$	$f(\boldsymbol{x}) \triangleq (f(\boldsymbol{x}^{(1)}); f(\boldsymbol{x}^{(2)}); \ldots; f(\boldsymbol{x}^{(r)}))$; f is any real valued continuous func.
Jordan product	$\boldsymbol{x} \circ \boldsymbol{y} \triangleq (\boldsymbol{x}^{(1)} \circ \boldsymbol{y}^{(1)}; \boldsymbol{x}^{(2)} \circ \boldsymbol{y}^{(2)}; \ldots; \boldsymbol{x}^{(r)} \circ \boldsymbol{y}^{(r)})$
Inner product	$\boldsymbol{x} \bullet \boldsymbol{y} \triangleq \boldsymbol{x}^{(1)} \bullet \boldsymbol{y}^{(1)} + \boldsymbol{x}^{(2)} \bullet \boldsymbol{y}^{(2)} + \cdots + \boldsymbol{x}^{(n_r)} \bullet \boldsymbol{y}^{(n_r)}$
Dot product	$\boldsymbol{x}^\mathsf{T}\boldsymbol{y} \triangleq \boldsymbol{x}^{(1)\mathsf{T}}\boldsymbol{y}^{(1)} + \cdots + \boldsymbol{x}^{(r)\mathsf{T}}\boldsymbol{y}^{(r)}$
Linear representation	$\mathrm{L}(\boldsymbol{x}) \triangleq \mathrm{L}(\boldsymbol{x}^{(1)}) \oplus \mathrm{L}(\boldsymbol{x}^{(2)}) \oplus \cdots \oplus \mathrm{L}(\boldsymbol{x}^{(r)})$
Quadratic representation	$\mathrm{Q}_{\boldsymbol{x}} \triangleq \mathrm{Q}_{\boldsymbol{x}^{(1)}} \oplus \mathrm{Q}_{\boldsymbol{x}^{(2)}} \oplus \cdots \oplus \mathrm{Q}_{\boldsymbol{x}^{(r)}}$
Quadratic operator	$\mathrm{Q}_{\boldsymbol{x},\boldsymbol{y}} \triangleq \mathrm{Q}_{\boldsymbol{x}^{(1)},\boldsymbol{y}^{(1)}} \oplus \mathrm{Q}_{\boldsymbol{x}^{(2)},\boldsymbol{y}^{(2)}} \oplus \cdots \oplus \mathrm{Q}_{\boldsymbol{x}^{(r)},\boldsymbol{y}^{(r)}}$
Frobenius norm	$\lVert\boldsymbol{x}\rVert_\mathrm{F} \triangleq \sum_{i=1}^r \lVert\boldsymbol{x}^{(i)}\rVert_\mathrm{F}$
Rank	$\mathsf{rk}(\mathcal{R}) \triangleq 2r$

In this section, we compute the derivatives of the logarithmic barrier associated with our cone and use them to prove the self-concordance of this barrier. To obtain these results, we do not use concepts outside of the EJA established in [28] and summarized in Table 1.

Associated with the cone $\mathcal{R}^n_+$, the logarithmic barrier function $b : \mathrm{Int}(\mathcal{R}^n_+) \to \mathbb{R}$ is defined as

$$b(\boldsymbol{x}) \triangleq -\ln \det(\boldsymbol{x}) = -\ln\left(x_1 x_2 - \|\bar{\boldsymbol{x}}\|^2\right). \tag{2}$$

We provide a proof the following lemma, which is a useful tool in order to prove some fundamental properties of our barrier. The inner product $\bullet$, inverse x^{-1}, norm $\|\cdot\|_{\mathrm{F}}$, and matrices $\mathrm{L}(\cdot)$ and $\mathrm{Q}_{\cdot}$ used in Lemma 1 are defined in Table 1.

Lemma 1. *Let $\boldsymbol{x} \in \mathcal{R}^n_+$ having $\boldsymbol{x} \succ \mathbf{0}$ and $\boldsymbol{h} \in \mathcal{R}^n_+$ be a non-zero vector. Then,*

(i) *The gradient $\nabla_{\boldsymbol{x}} b(\boldsymbol{x}) = -2H\boldsymbol{x}^{-1}$. Therefore, $\nabla_{\boldsymbol{x}} b(\boldsymbol{x})[\boldsymbol{h}] = -2\,\boldsymbol{h} \bullet \boldsymbol{x}^{-1}$.*

(ii) *The Hessian $\nabla^2_{\boldsymbol{xx}} b(\boldsymbol{x}) = 2HQ_{\boldsymbol{x}^{-1}}$. Therefore, $\nabla^2_{\boldsymbol{xx}} b(\boldsymbol{x})\ [\boldsymbol{h},\boldsymbol{h}] = \|Q_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}\|^2_{F}$.*

(iii) *The third derivative $\nabla^3_{\boldsymbol{xxx}} b(\boldsymbol{x})[\boldsymbol{h},\boldsymbol{h},\boldsymbol{h}] = -4(Q_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}) \bullet (Q_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})^2$.*

Proof of Lemma 1. For item (i), we have

$$\nabla_{\boldsymbol{x}} b(\boldsymbol{x}) = \frac{-1}{\det(\boldsymbol{x})}\begin{bmatrix} x_2 \\ x_1 \\ -2\bar{\boldsymbol{x}} \end{bmatrix} = \frac{-1}{\det(\boldsymbol{x})} R \begin{bmatrix} x_1 \\ x_2 \\ 2\bar{\boldsymbol{x}} \end{bmatrix} = \frac{-2}{\det(\boldsymbol{x})} R \begin{bmatrix} x_1/2 \\ x_2/2 \\ \bar{\boldsymbol{x}} \end{bmatrix} = \frac{-2}{\det(\boldsymbol{x})} RH \begin{bmatrix} x_1 \\ x_2 \\ \bar{\boldsymbol{x}} \end{bmatrix} = -2H\boldsymbol{x}^{-1}.$$

Item (ii) follows by using item (i) and noting that the Jacobean of $\boldsymbol{x}^{-1}$ is

$$\mathrm{J}_x \boldsymbol{x}^{-1} = \mathrm{J}_x \begin{bmatrix} \dfrac{x_2}{x_1x_2 - \|\bar{\boldsymbol{x}}\|^2} \\ \dfrac{x_1}{x_1x_2 - \|\bar{\boldsymbol{x}}\|^2} \\ \dfrac{\bar{\boldsymbol{x}}}{x_1x_2 - \|\bar{\boldsymbol{x}}\|^2} \end{bmatrix} = \frac{-1}{\det(\boldsymbol{x})^2}\begin{bmatrix} -x_2^2 & -||\bar{\boldsymbol{x}}||^2 & 2x_2\bar{\boldsymbol{x}}^{\mathsf{T}} \\ -||\bar{\boldsymbol{x}}||^2 & -x_1^2 & 2x_1\bar{\boldsymbol{x}}^{\mathsf{T}} \\ x_2\bar{\boldsymbol{x}} & x_1\bar{\boldsymbol{x}} & -2\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^{\mathsf{T}} - \det(\boldsymbol{x}) I_{n-2} \end{bmatrix} = -\mathrm{Q}_{\boldsymbol{x}^{-1}},$$

and that

$$\nabla^2_{\boldsymbol{xx}} b(\boldsymbol{x})\ [\boldsymbol{h},\boldsymbol{h}] = 2\boldsymbol{h}^{\mathsf{T}}(H\mathrm{Q}_{\boldsymbol{x}^{-1}})\boldsymbol{h} = 2\ (\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})^{\mathsf{T}} H(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}) = 2\ (\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}) \bullet (\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}) = \|\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}\|^2_{\mathrm{F}}.$$

For item (iii), note that

$$\begin{aligned} \nabla_{\boldsymbol{x}}\mathrm{Q}_{\boldsymbol{x}^{-1}}[\boldsymbol{h},\boldsymbol{h}] &= \nabla_{\boldsymbol{x}}\left(2\mathrm{L}\left(\boldsymbol{x}^{-1}\right)^2 - L\left(\boldsymbol{x}^{-2}\right)\right)[\boldsymbol{h},\boldsymbol{h}] \\ &= -2\left(\mathrm{L}(\mathrm{Q}_{\boldsymbol{x}^{-1}}\boldsymbol{h})\mathrm{L}\left(\boldsymbol{x}^{-1}\right) + \mathrm{L}\left(\boldsymbol{x}^{-1}\right)\mathrm{L}(\mathrm{Q}_{\boldsymbol{x}^{-1}}\boldsymbol{h}) - \mathrm{L}\left(\mathrm{Q}_{\boldsymbol{x}^{-1}}\boldsymbol{h} \circ \boldsymbol{x}^{-1}\right)\right)\boldsymbol{h} \\ &= -2\mathrm{Q}_{\mathrm{Q}_{\boldsymbol{x}^{-1}}\boldsymbol{h},\boldsymbol{x}^{-1}}\boldsymbol{h} \ = \ -2\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\mathrm{L}(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}. \end{aligned}$$

It follows that

$$\begin{aligned} \nabla^3_{\boldsymbol{xxx}} b(\boldsymbol{x})[\boldsymbol{h},\boldsymbol{h},\boldsymbol{h}] &= \nabla_{\boldsymbol{x}}\left(\boldsymbol{h}^{\mathsf{T}}\nabla^2_{\boldsymbol{xx}} b(\boldsymbol{x})\boldsymbol{h}\right)[\boldsymbol{h}] \\ &= 2\nabla_{\boldsymbol{x}}\left(\boldsymbol{h}^{\mathsf{T}} H\mathrm{Q}_{\boldsymbol{x}^{-1}}\boldsymbol{h}\right)[\boldsymbol{h}] \\ &= -4\boldsymbol{h}^{\mathsf{T}} H\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\mathrm{L}(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h} \\ &= -4(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})^{\mathsf{T}} H\mathrm{L}(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h} \ = \ -4(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}) \bullet (\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})^2, \end{aligned}$$

where we used the fact that $\mathrm{L}(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h} = (\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})^2$ to obtain the last equality. The proof is complete. □

The notion of self-concordance introduced by Nesterov and Nemirovskii [3] is essential to the existence of polynomial-time interior-point algorithms for convex conic programming problems. We have the following definition.

Definition 1 (Definition 2.1.1 in [3])**.** *Let V be a finite-dimensional real vector space, G be an open non-empty convex subset of V, and let f be a C^3, convex mapping from G to $\mathbb{R}$. Then, f is* *called α-self-concordant on G with the parameter $\alpha > 0$ if for every $\boldsymbol{x} \in G$ and $\boldsymbol{h} \in V$, the following inequality holds*

$$\left| \nabla^3_{xxx} f(\boldsymbol{x}) \, [\boldsymbol{h}, \boldsymbol{h}, \boldsymbol{h}] \right| \leq \frac{2}{\sqrt{\alpha}} \left(\nabla^2_{xx} f(\boldsymbol{x}) \, [\boldsymbol{h}, \boldsymbol{h}] \right)^{3/2}. \tag{3}$$

An α-self-concordant function f on G is called strongly α-self-concordant if f tends to infinity for any sequence approaching a boundary point of G.

In the proof of the next theorem, we use the inequalities that

$$|\boldsymbol{x} \bullet \boldsymbol{y}| \leq \frac{1}{2} \|\boldsymbol{x}\|_{\mathrm{F}} \|\boldsymbol{y}\|_{\mathrm{F}}, \text{ and } \|\boldsymbol{x}^2\|_{\mathrm{F}} \leq \|\boldsymbol{x}\|^2_{\mathrm{F}},$$

for any $\boldsymbol{x}$ and $\boldsymbol{y}$ residing a Jordan algebra. These two inequalities can be seen by noting that

$$\|\boldsymbol{x}^2\|_{\mathrm{F}} = \sqrt{\lambda_1^4(\boldsymbol{x}) + \lambda_2^4(\boldsymbol{x})} \leq \lambda_1^2(\boldsymbol{x}) + \lambda_2^2(\boldsymbol{x}) = \|\boldsymbol{x}\|^2_{\mathrm{F}},$$

and

$$|\boldsymbol{x} \bullet \boldsymbol{y}| = |\boldsymbol{x}^{\mathsf{T}} H \boldsymbol{y}| = \left| \left(H^{1/2}\boldsymbol{x}\right)^{\mathsf{T}} H^{1/2}\boldsymbol{y} \right| \leq \|H^{1/2}\boldsymbol{x}\|_2 \|H^{1/2}\boldsymbol{y}\|_2 = \sqrt{\boldsymbol{x}^{\mathsf{T}} H \boldsymbol{x}} \sqrt{\boldsymbol{y}^{\mathsf{T}} H \boldsymbol{y}} = \frac{1}{2} \|\boldsymbol{x}\|_{\mathrm{F}} \|\boldsymbol{y}\|_{\mathrm{F}}.$$

We are now ready to provide a proof for the following theorem.

Theorem 1. *The logarithmic barrier function $b(\boldsymbol{x})$ is a 1-strongly self-concordant barrier on $Int(\mathcal{R}^n_+)$.*

Proof of Theorem 1. Note that for any sequence approaching the $\mathcal{R}^n_+$ boundary point, $b(\cdot)$ goes to infinity. Using items (ii) and (iii) in Lemma 1, we have

$$\begin{array}{rcl} \left|\nabla^3_{xxx} b(\boldsymbol{x})[\boldsymbol{h}, \boldsymbol{h}, \boldsymbol{h}]\right| & = & 4\left|(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}) \bullet (\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})^2\right| \\ & \leq & 2\|\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}\|_{\mathrm{F}} \|(\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h})^2\|_{\mathrm{F}} \\ & \leq & 2\|\mathrm{Q}_{\boldsymbol{x}^{-1/2}}\boldsymbol{h}\|^3_{\mathrm{F}} \\ & = & 2\left(\boldsymbol{h}^{\mathsf{T}} \nabla^2_{xx} b(\boldsymbol{x}) \boldsymbol{h}\right)^{3/2} = 2\left(\left|\nabla^2_{xx} b(\boldsymbol{x})[\boldsymbol{h}, \boldsymbol{h}]\right|\right)^{3/2}. \end{array} \tag{4}$$

Thus, the inequality in (3) holds. Hence, the logarithmic barrier $b(\boldsymbol{x})$ on $\mathrm{Int}(\mathcal{R}^n_+)$ is 1-strongly self-concordant. □

3. Rotated Quadratic Cone Programming Problem and Duality

In this section, we introduce and define the RQCP problem along with a discussion of the duality theory and the optimality conditions for these problems.

Let $A \in \mathbb{R}^{m \times n}$ be a real matrix whose m rows reside in the Euclidean Jordan algebra $(\mathcal{R}^n, \circ, \bullet)$, and let A^{T} be its transpose. Associated with A, we define the matrix–vector product "$\mathcal{A}\cdot$" as

$$\mathcal{A}\boldsymbol{x} \triangleq AH\boldsymbol{x} = (\boldsymbol{a}_1 \bullet \boldsymbol{x}; \boldsymbol{a}_2 \bullet \boldsymbol{x}; \ldots; \boldsymbol{a}_m \bullet \boldsymbol{x}),$$

where $\boldsymbol{x} \in \mathcal{R}^n$, $\boldsymbol{a}_i \in \mathcal{R}^n$ is the ith-row of A for $i = 1, 2, \ldots, m$, and H is defined in Table 1. The operator $\mathcal{A}$ is the half-identity matrix defined to map $(\mathcal{R}^n, \circ, \bullet)$ into $\mathbb{R}^m$, while the transpose A^{T} is defined to map $\mathbb{R}^m$ into $(\mathcal{R}^n, \circ, \bullet)$. If $\boldsymbol{x} \in \mathcal{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$, one can easily show that

$$(\mathcal{A}\boldsymbol{x})^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{x} \bullet (A^{\mathsf{T}}\boldsymbol{y}) \tag{5}$$

An RQCP problem in the primal form is the problem

$$(P)\left\{\begin{array}{ll} \min & \boldsymbol{c} \bullet \boldsymbol{x} \\ \text{s.t.} & \mathcal{A}\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \succeq \boldsymbol{0}, \end{array}\right. \text{ and its dual is } (D)\left\{\begin{array}{ll} \max & \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.} & A^{\mathsf{T}}\boldsymbol{y} \preceq \boldsymbol{c}, \end{array}\right. \text{or } (D)\left\{\begin{array}{ll} \max & \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.} & A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}, \\ & \boldsymbol{s} \succeq \boldsymbol{0}, \end{array}\right.$$

where $\boldsymbol{x} \in \mathcal{R}^n$ is the primal variable, $\boldsymbol{y} \in \mathbb{R}^m$ is the dual variable, and $\boldsymbol{s} \in \mathcal{R}^n$ is the dual slack variable.

Let $\mathcal{F} \triangleq \mathcal{F}_{\mathrm{P}} \times \mathcal{F}_{\mathrm{D}}$ and $\mathcal{F}^{\circ} \triangleq \mathcal{F}_{\mathrm{P}}^{\circ} \times \mathcal{F}_{\mathrm{D}}^{\circ}$ denote the feasible and strictly feasible primal-dual sets for the pair (P, D), respectively, where

$$\begin{array}{lcll} \mathcal{F}_{\mathrm{P}} & \triangleq & \left\{\boldsymbol{x} \in \mathcal{R}^n : \mathcal{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succeq \boldsymbol{0}\right\} & \text{(the primal feasibility set),} \\ \mathcal{F}_{\mathrm{D}} & \triangleq & \left\{(\boldsymbol{y}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathcal{R}^n : A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{s} \succeq \boldsymbol{0}\right\} & \text{(the dual feasibility set),} \\ \mathcal{F}_{\mathrm{P}}^{\circ} & \triangleq & \left\{\boldsymbol{x} \in \mathcal{R}^n : \mathcal{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \succ \boldsymbol{0}\right\} & \text{(the strict primal feasibility set),} \\ \mathcal{F}_{\mathrm{D}}^{\circ} & \triangleq & \left\{(\boldsymbol{y}, \boldsymbol{s}) \in \mathbb{R}^m \times \mathcal{R}^n : A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{s} \succ \boldsymbol{0}\right\} & \text{(the strict dual feasibility set).} \end{array}$$

Problem (P) (respectively, Problem (D)) is said to be strictly feasible if $\mathcal{F}_{\mathrm{P}}^{\circ} \neq \emptyset$ (respectively, $\mathcal{F}_{\mathrm{D}}^{\circ} \neq \emptyset$). Now, we make two assumptions about the pair (P, D): First, we assume that the matrix A has a full row rank. This assumption is standard and is added for convenience. Second, we assume that the strictly feasible set $\mathcal{F}^0$ is non-empty. This assumption guarantees that the strong duality holds for the RQCP problem and suggests that both Problems (P) and (D) have a unique solution.

We give with a proof the following weak duality result.

Lemma 2 (Weak duality)**.** *If $\boldsymbol{x} \in \mathcal{F}_P$ and $(\boldsymbol{y}, \boldsymbol{s}) \in \mathcal{F}_D$, then the duality gap is $\boldsymbol{c} \bullet \boldsymbol{x} - \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{x} \bullet \boldsymbol{s} \geq 0$.*

Proof of Theorem 2. Let $\boldsymbol{x} \in \mathcal{F}_{\mathrm{P}}$ and $(\boldsymbol{y}, \boldsymbol{s}) \in \mathcal{F}_{\mathrm{D}}$, then $A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}, \mathcal{A}\boldsymbol{x} = \boldsymbol{b}$, and $\boldsymbol{x}, \boldsymbol{s} \succeq \boldsymbol{0}$. Then, using (5), it follows that

$$\boldsymbol{c} \bullet \boldsymbol{x} - \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} = (A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s}) \bullet \boldsymbol{x} - (\mathcal{A}\boldsymbol{x})^{\mathsf{T}}\boldsymbol{y} = (A^{\mathsf{T}}\boldsymbol{y}) \bullet \boldsymbol{x} + \boldsymbol{x} \bullet \boldsymbol{s} - (\mathcal{A}\boldsymbol{x})^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{x} \bullet \boldsymbol{s}.$$

Because $\boldsymbol{x}, \boldsymbol{s} \succeq \boldsymbol{0}$, we have that

$$x_1 x_2 s_1 s_2 \geq \|\bar{x}\|^2\ \|\bar{s}\|^2. \tag{6}$$

Applying the arithmetic inequality to $x_1 s_1$ and $x_2 s_2$, we obtain

$$\frac{1}{4}(x_1 s_1 + x_2 s_2)^2 \geq x_1 x_2 s_1 s_2. \tag{7}$$

Combining (6) and (7), we have $(x_1 s_1 + x_2 s_2)^2/4 \geq \|\bar{x}\|^2\ \|\bar{s}\|^2$. Taking the square root of both sides and applying the Cauchy–Schwartz inequality, we obtain

$$\boldsymbol{x} \bullet \boldsymbol{s} = \frac{1}{2}(x_1 s_1 + x_2 s_2) + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} \geq \|\bar{x}\|\ \|\bar{s}\| + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} \geq \left|\bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}}\right| + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} \geq 0.$$

The proof is complete. □

It is known that the strong duality property can fail in general conic programming problems, but a slightly weaker property can be shown for them [3]. Using the Karush–Kuhn–Tucker (KKT) conditions, we provide a proof of the following semi-strong duality result, which provides conditions for such a slightly weaker property to hold in RQCP.

Lemma 3 (Semi-strong duality)**.** *Let Problems (P) and (D) be strictly feasible. If Problem (P) is solvable, then so is its dual (D) and their optimal values are equal.*

Proof of Theorem 3. Since (P) is strictly feasible and solvable, then there is a solution $\boldsymbol{x} \in \mathcal{F}_{\mathrm{P}}$ to which the KKT conditions can be applied. According to this, there must be Lagrange multiplier vectors $\boldsymbol{y}$ and $\boldsymbol{s}$ such that $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ satisfies the conditions

$$\begin{aligned} \mathcal{A}\boldsymbol{x} &= \boldsymbol{b}, \\ A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} &= \boldsymbol{c}, \\ \boldsymbol{x} \bullet \boldsymbol{s} &= 0, \\ \boldsymbol{x}, \boldsymbol{s} &\succeq \boldsymbol{0}. \end{aligned} \tag{8}$$

It follows that (D) can be solved using $(\boldsymbol{y}, \boldsymbol{s})$. Let us suppose that $(\boldsymbol{u}, \boldsymbol{v})$ is any feasible solution to the dual problem (D); then,

$$\boldsymbol{b}^{\mathsf{T}}\boldsymbol{u} \leq \boldsymbol{c} \bullet \boldsymbol{x} = \boldsymbol{x} \bullet \boldsymbol{s} + \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y},$$

where the inequality was obtained using weak duality (Lemma 2) and the last equality was obtained using the last equation in (8). Since $(\boldsymbol{u}, \boldsymbol{v})$ was chosen arbitrarily, the optimal solution to Problem (D) is $(\boldsymbol{y}, \boldsymbol{s})$ and $\boldsymbol{c} \bullet \boldsymbol{x} = \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y}$. The proof is complete. □

The strong duality result in the following lemma can be obtained by applying the duality relations to our problem formulation.

Theorem 2 (Strong duality)**.** *Let Problems (P) and (D) be strictly feasible; then, they must also have optimal solutions, say $\boldsymbol{x}^{\star}$ and $(\boldsymbol{y}^{\star}, \boldsymbol{s}^{\star})$, respectively, and $\boldsymbol{c} \bullet \boldsymbol{x}^{\star} = \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y}^{\star}$ (i.e., $\boldsymbol{x}^{\star} \bullet \boldsymbol{s}^{\star} = 0$).*

As one of the optimality conditions of RQCP, we describe the complementarity condition in the following lemma.

Lemma 4 (Complementarity condition)**.** *Let $\boldsymbol{x}, \boldsymbol{s} \in \mathcal{R}^n$ have $\boldsymbol{x}, \boldsymbol{s} \succeq \boldsymbol{0}$. Then $\boldsymbol{x} \bullet \boldsymbol{s} = 0$ if and only if $\boldsymbol{x} \circ \boldsymbol{s} = \boldsymbol{0}$.*

Proof of Theorem 4. Let $\boldsymbol{x}, \boldsymbol{s} \in \mathcal{R}^n$ have $\boldsymbol{x}, \boldsymbol{s} \succeq \boldsymbol{0}$. First, we prove the direction from left to right. Assume that $\boldsymbol{x} \bullet \boldsymbol{s} = 0$. To show that $\boldsymbol{x} \circ \boldsymbol{s} = \boldsymbol{0}$, we must show that (see the definition of the Jordan product "$\circ$" used in Table 1)

(i) $x_1 s_1 + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = 0$; (ii) $x_2 s_2 + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = 0$; (iii) $\frac{1}{2}(s_1 + s_2)\bar{\boldsymbol{x}} + \frac{1}{2}(x_1 + x_2)\bar{\boldsymbol{s}} = \boldsymbol{0}$.

If $(x_1 = 0$ and $x_2 = 0)$ or $(s_1 = 0$ and $s_2 = 0)$, then $\boldsymbol{x} = \boldsymbol{0}$ in the first case and $\boldsymbol{s} = \boldsymbol{0}$ in the second one, indicating that (i), (ii), and (iii) trivially hold. As a result, we need to consider $x_1, x_2 > 0$ and $s_1, s_2 > 0$. Then, by taking the square root of both sides in (7), using the fact that $\boldsymbol{x}, \boldsymbol{s} \succeq \boldsymbol{0}$, and applying the Cauchy–Schwartz inequality, we obtain

$$-\frac{1}{2}(x_1 s_1 + x_2 s_2) \leq -(x_1 x_2 s_1 s_2)^{1/2} \leq -\|\bar{\boldsymbol{x}}\|\, \|\bar{\boldsymbol{s}}\| \leq \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}}. \tag{9}$$

Therefore, $\boldsymbol{x} \bullet \boldsymbol{s} = \frac{1}{2}(x_1 s_1 + x_2 s_2) + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = 0$ if and only if $\bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = -\frac{1}{2}(x_1 s_1 + x_2 s_2)$. This is true if and only if the inequalities in (9) are satisfied as equalities. This simply holds true if and only if either $\boldsymbol{x} = \boldsymbol{0}$ or $\boldsymbol{s} = \boldsymbol{0}$, in which (i), (ii), and (iii) are trivially held, or

$$\boldsymbol{x} \gneqq \boldsymbol{0},\ \boldsymbol{s} \gneqq \boldsymbol{0},\ \bar{\boldsymbol{x}} = -\beta\bar{\boldsymbol{s}},\ x_1 = \|\bar{\boldsymbol{x}}\| = \beta\|\bar{\boldsymbol{s}}\| = \beta s_2, \text{ and } x_2 = \|\bar{\boldsymbol{x}}\| = \beta\|\bar{\boldsymbol{s}}\| = \beta s_1, \tag{10}$$

where $\beta > 0$.

Note that the first equation in (10), or equivalently $\bar{\boldsymbol{x}} + \beta\bar{\boldsymbol{s}} = \boldsymbol{0}$, implies that $\|\bar{\boldsymbol{x}}\|^2 + \beta\bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = 0$. Using (10), this can be written as

$$\beta s_2^2 + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = 0 \text{ and } \beta s_1^2 + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = 0, \quad \text{or as } \ x_1 s_2 + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = 0 \text{ and } x_2 s_1 + \bar{\boldsymbol{x}}^{\mathsf{T}}\bar{\boldsymbol{s}} = 0.$$

From (10), we have that $s_1 = s_2$. Then, $x_1 s_1 + \bar{\boldsymbol{x}}^{\mathsf{T}} \bar{\boldsymbol{s}} = 0$ and $x_2 s_2 + \bar{\boldsymbol{x}}^{\mathsf{T}} \bar{\boldsymbol{s}} = 0$, as desired in (i) and (ii). For (iii), using (10) again, we have

$$\bar{\boldsymbol{x}} + \beta \bar{\boldsymbol{s}} = \bar{\boldsymbol{x}} + \frac{x_1}{s_2} \bar{\boldsymbol{s}} = \boldsymbol{0} \text{ and } \bar{\boldsymbol{x}} + \beta \bar{\boldsymbol{s}} = \bar{\boldsymbol{x}} + \frac{x_2}{s_1} \bar{\boldsymbol{s}} = \boldsymbol{0}.$$

This implies that $(x_1 + x_2)\bar{\boldsymbol{s}} + (s_1 + s_2)\bar{\boldsymbol{x}} = \boldsymbol{0}$, or as desired in (iii).

Now, we prove the direction from right to left. Let us assume that $\boldsymbol{x} \circ \boldsymbol{s} = \boldsymbol{0}$. From (i) and (ii), we have that $x_1 s_1 + x_2 s_2 + 2\bar{\boldsymbol{x}}^{\mathsf{T}} \bar{\boldsymbol{s}} = 0$, or $\boldsymbol{x} \bullet \boldsymbol{s} = \frac{1}{2}(x_1 s_1 + x_2 s_2) + \bar{\boldsymbol{x}}^{\mathsf{T}} \bar{\boldsymbol{s}} = 0$ as desired. The proof is complete. □

From the above results, the following corollary is now immediate.

Corollary 1 (Optimality conditions)**.** *Let us assume that both Problems (P) and (D) are strictly feasible. Then,* $(\boldsymbol{x}, (\boldsymbol{y}, \boldsymbol{s})) \in \mathcal{R}^n \times \mathbb{R}^m \times \mathcal{R}^n$ *is an optimal solution to the pair (P, D) if and only if*

$$\begin{aligned}
&\mathcal{A}\boldsymbol{x} = \boldsymbol{b},\\
&A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\\
&\boldsymbol{x} \circ \boldsymbol{s} = \boldsymbol{0},\\
&\boldsymbol{x}, \boldsymbol{s} \succeq \boldsymbol{0}.
\end{aligned}$$

4. The Newton System and Commutative Directions

In this section, we present the logarithmic barrier problems for the pair (P, D) and the Newton system corresponding to them, as well as a subclass of the MZ family of search directions known as the commutative directions.

The logarithmic barrier problems associated with the pair (P, D) are the problems

$$(\mathrm{P}_\mu)\begin{cases} \min & \boldsymbol{c} \bullet \boldsymbol{x} - \mu \ln\det(\boldsymbol{x}) \\ \text{s.t.} & \mathcal{A}\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \succ \boldsymbol{0}, \end{cases} \qquad (\mathrm{D}_\mu)\begin{cases} \max & \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} + \mu \ln\det(\boldsymbol{s}) \\ \text{s.t.} & A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}, \\ & \boldsymbol{s} \succ \boldsymbol{0}, \end{cases}$$

where $\mu \triangleq \frac{1}{2}\boldsymbol{x} \bullet \boldsymbol{s}$ is a small positive scalar and is typically referred to as the barrier parameter.

The solutions of the pair (P_μ, D_μ) can be characterized by the following perturbed KKT optimality conditions.

$$\begin{aligned}
&\mathcal{A}\boldsymbol{x} = \boldsymbol{b},\\
&A^{\mathsf{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\\
&\boldsymbol{x} \circ \boldsymbol{s} = \delta\mu\boldsymbol{e},\\
&\boldsymbol{x}, \boldsymbol{s} \succ \boldsymbol{0},
\end{aligned} \tag{11}$$

where $\boldsymbol{e} = (1; 1; \boldsymbol{0})$ is the identity vector of $\mathcal{R}^n$ as defined in Table 1, and $\delta \in (0, 1)$ is a centering parameter that reduces the barrier term μ. For any $\mu > 0$, System (11) has a unique solution, indicated by $(\boldsymbol{x}_\mu, \boldsymbol{y}_\mu, \boldsymbol{s}_\mu)$, where $\boldsymbol{x}_\mu$ is called the μ-center for (P) and the pair $(\boldsymbol{y}_\mu, \boldsymbol{s}_\mu)$ is called the μ-center for (D). The set of all μ-centers that solve the perturbed KKT system (11) is called the central path of the pair (P, D), and is defined as

$$\mathcal{CP} \triangleq \big\{(\boldsymbol{x}_\mu, \boldsymbol{y}_\mu, \boldsymbol{s}_\mu) \in \mathcal{F}^0 : \boldsymbol{x} \circ \boldsymbol{s} = \delta\mu\boldsymbol{e},\ \mu > 0\big\}.$$

Due the assumption that $\mathcal{F}^0 \neq 0$, the central path is well defined. As μ gets closer to zero, the μ-central point $(\boldsymbol{x}_\mu, \boldsymbol{y}_\mu, \boldsymbol{s}_\mu)$ converges toward an ϵ-approximate solution $(\boldsymbol{x}^\star, \boldsymbol{y}^\star, \boldsymbol{s}^\star)$ of (P, D).

Now, we reformulate the complementary condition $\boldsymbol{x} \circ \boldsymbol{s} = \delta\mu\boldsymbol{e}$ in (11), which is a direct consequence of Lemma 28 in [30].

Lemma 5. *Let $\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{p} \in (\mathcal{R}^n, \circ, \bullet)$ be such that $\boldsymbol{x}, \boldsymbol{s} \succ \boldsymbol{0}$, and $\boldsymbol{p}$ is invertible (i.e.,* $\det(\boldsymbol{p}) \neq 0$*). Then $\boldsymbol{x} \circ \boldsymbol{s} = \delta\mu\boldsymbol{e}$ if and only if $Q_{\boldsymbol{p}}\boldsymbol{x} \circ Q_{\boldsymbol{p}^{-1}}\boldsymbol{s} = \delta\mu\boldsymbol{e}$.*

In order to solve System (11), we apply Newton's method to this system and obtain

$$\begin{aligned} &\mathcal{A}\Delta\boldsymbol{x} = \boldsymbol{b} - \mathcal{A}\boldsymbol{x}, \\ &A^{\mathsf{T}}\Delta\boldsymbol{y} + \Delta\boldsymbol{s} = \boldsymbol{c} - A^{\mathsf{T}}\boldsymbol{y} - \boldsymbol{s}, \\ &\mathrm{Q}_{\boldsymbol{p}}\boldsymbol{x} \circ \mathrm{Q}_{\boldsymbol{p}^{-1}}\Delta\boldsymbol{s} + \mathrm{Q}_{\boldsymbol{p}}\Delta\boldsymbol{x} \circ \mathrm{Q}_{\boldsymbol{p}^{-1}}\boldsymbol{s} = \delta\mu\boldsymbol{e} - \mathrm{Q}_{\boldsymbol{p}}\boldsymbol{x} \circ \mathrm{Q}_{\boldsymbol{p}^{-1}}\boldsymbol{s}, \end{aligned} \tag{12}$$

where $(\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{s}) \in (\mathcal{R}^n, \circ, \bullet) \times R^m \times (\mathcal{R}^n, \circ, \bullet)$ is called the Newton search direction.

In the theory of Jordan algebra, two elements of a Jordan algebra operator commute if they share the same set of eigenvectors. In particular, two vectors $\boldsymbol{u}, \boldsymbol{v}s. \in \mathcal{R}^n$ operator commute if $\boldsymbol{c}_i = \boldsymbol{c}_i(\boldsymbol{u}) = \boldsymbol{c}_i(\boldsymbol{v})$ for $i = 1, 2$, i.e., $\boldsymbol{u} = \lambda_1(\boldsymbol{u})\boldsymbol{c}_1 + \lambda_2(\boldsymbol{u})\boldsymbol{c}_2$ and $\boldsymbol{v} = \lambda_1(\boldsymbol{v})\boldsymbol{c}_1 + \lambda_2(\boldsymbol{v})\boldsymbol{c}_2$. The vectors $\boldsymbol{x}$ and $\boldsymbol{s}$ in System (12) may not operator commute. We need now to scale the underlying system so that the scaled vectors operator commute. In practice, this scaling is needed to guarantee that we iterate in the interior of the quadratic rotated cone.

Let $\mathcal{C}(\boldsymbol{x}, \boldsymbol{s})$ be the set of nonsingular elements so that the scaled vectors operator commute, i.e.,

$$\mathcal{C}(\boldsymbol{x}, \boldsymbol{s}) = \left\{\boldsymbol{p} \in \mathcal{R}^n : \boldsymbol{p} \text{ nonsingular and } \mathrm{Q}_{\boldsymbol{p}}\boldsymbol{x} \text{ and } \mathrm{Q}_{\boldsymbol{p}^{-1}}\boldsymbol{s} \text{ operator commute}\right\}.$$

We call the set of directions $(\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{s})$ arising by choosing $\boldsymbol{p}$ from $\mathcal{C}(\boldsymbol{x}, \boldsymbol{s})$ the commutative class of directions for RQCP, and call a direction in this class a commutative direction.

As mentioned earlier, the commutative class of primal-dual IPMs was designed by Monteiro and Zhang [29] for semidefinite programming, and by Alizadeh and Goldfarb [7] for second-order cone programming, and then extended by Schmieta and Alizadeh [30] for symmetric cone programming. We concentrate on three prominent choices of $\boldsymbol{p}$, and each choice leads to a different direction in the commutative class of search directions: First, the choice $\boldsymbol{p} = \boldsymbol{s}^{1/2}$ is referred to as the HRVW/KSH/M direction, and is equivalent to the XS direction in semidefinite programming (introduced by Helmberg, Rendl, Vanderbei, and Wolkowicz [31], and Kojima, Shindoh, and Hara [32] independently, and then rediscovered by Monteiro [29]). Second, the choice $\boldsymbol{p} = \boldsymbol{x}^{-1/2}$ is referred to as the dual HRVW/KSH/M direction, and is equivalent to the SX direction in semidefinite programming. Third, the choice $\boldsymbol{p} = (\mathrm{Q}_{\boldsymbol{x}^{1/2}}(\mathrm{Q}_{\boldsymbol{x}^{1/2}}\boldsymbol{s})^{-1/2})^{-1/2} = (\mathrm{Q}_{\boldsymbol{s}^{-1/2}}(\mathrm{Q}_{\boldsymbol{s}^{1/2}}\boldsymbol{x})^{1/2})^{-1/2}$ is equivalent to the NT direction in semidefinite programming (introduced by Nesterov and Todd [4]).

Now, associated with $\boldsymbol{p} \in \mathcal{C}(\boldsymbol{x}, \boldsymbol{s})$, we make the following change of variables:

$$\boldsymbol{x} \to \overline{\boldsymbol{x}} \triangleq \mathrm{Q}_{\boldsymbol{p}}\boldsymbol{x}, \;\; \boldsymbol{s} \to \underline{\boldsymbol{s}} \triangleq \mathrm{Q}_{\boldsymbol{p}^{-1}}\boldsymbol{s}, \;\; \boldsymbol{c} \to \underline{\boldsymbol{c}} \triangleq \mathrm{Q}_{\boldsymbol{p}^{-1}}\boldsymbol{c}, \;\; A \to \underline{A} \triangleq A\mathrm{Q}_{\boldsymbol{p}^{-1}}, \;\; \mathcal{A} \to \underline{\mathcal{A}} \triangleq AH\mathrm{Q}_{\boldsymbol{p}^{-1}}.$$

Because $\mathrm{Q}_{\boldsymbol{p}^{-1}} = \mathrm{Q}_{\boldsymbol{p}}^{-1}$ (see [28, Theorem 4.3]), System (12) is equivalent to

$$\begin{aligned} \underline{\mathcal{A}}\overline{\Delta\boldsymbol{x}} &= \boldsymbol{r}_{\mathrm{p}}, \\ \underline{A}^{\mathsf{T}}\Delta\boldsymbol{y} + \underline{\Delta\boldsymbol{s}} &= \boldsymbol{r}_{\mathrm{d}}, \\ \overline{\boldsymbol{x}} \circ \underline{\Delta\boldsymbol{s}} + \overline{\Delta\boldsymbol{x}} \circ \underline{\boldsymbol{s}} &= \boldsymbol{r}_{\mathrm{c}}, \end{aligned} \quad \text{or equivalently} \quad \begin{bmatrix} \underline{\mathcal{A}} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \underline{A}^{\mathsf{T}} & I \\ \mathrm{L}(\underline{\boldsymbol{s}}) & \boldsymbol{O} & \mathrm{L}(\overline{\boldsymbol{x}}) \end{bmatrix} \begin{bmatrix} \overline{\Delta\boldsymbol{x}} \\ \Delta\boldsymbol{y} \\ \underline{\Delta\boldsymbol{s}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{\mathrm{p}} \\ \boldsymbol{r}_{\mathrm{d}} \\ \boldsymbol{r}_{\mathrm{c}} \end{bmatrix}, \tag{13}$$

where $\boldsymbol{r}_{\mathrm{p}}, \boldsymbol{r}_{\mathrm{d}}$, and $\boldsymbol{r}_{\mathrm{c}}$ are given by

$$\boldsymbol{r}_{\mathrm{p}} \triangleq \boldsymbol{b} - \underline{\mathcal{A}}\overline{\boldsymbol{x}}, \;\; \boldsymbol{r}_{\mathrm{d}} \triangleq \underline{\boldsymbol{c}} - \underline{A}^{\mathsf{T}}\boldsymbol{y} - \underline{\boldsymbol{s}}, \;\; \boldsymbol{r}_{\mathrm{c}} \triangleq \delta\mu\boldsymbol{e} - \overline{\boldsymbol{x}} \circ \underline{\boldsymbol{s}}.$$

Applying block Gaussian elimination to (13), we obtained the Newton search directions $(\overline{\Delta\boldsymbol{x}}, \Delta\boldsymbol{y}, \underline{\Delta\boldsymbol{s}})$:

$$\begin{aligned} \overline{\Delta\boldsymbol{x}} &= \mathrm{L}^{-1}(\underline{\boldsymbol{s}})\left(\boldsymbol{r}_{\mathrm{c}} - \mathrm{L}(\overline{\boldsymbol{x}})\underline{\Delta\boldsymbol{s}}\right), \\ \Delta\boldsymbol{y} &= \left(\underline{\mathcal{A}}\mathrm{L}^{-1}(\underline{\boldsymbol{s}})\mathrm{L}(\overline{\boldsymbol{x}})\underline{A}^{\mathsf{T}}\right)^{-1}\left(\boldsymbol{r}_{\mathrm{p}} + \underline{\mathcal{A}}\mathrm{L}^{-1}(\underline{\boldsymbol{s}})(\mathrm{L}(\overline{\boldsymbol{x}})\boldsymbol{r}_{\mathrm{d}} - \overline{r}_{\mathrm{c}})\right), \\ \underline{\Delta\boldsymbol{s}} &= \boldsymbol{r}_{\mathrm{d}} - \underline{A}^{\mathsf{T}}\overline{\Delta\boldsymbol{y}}. \end{aligned}$$

To obtain the search directions Δx and Δs for (12), we apply inverse scaling to $\overline{\Delta x}$ and $\underline{\Delta s}$ as follows:

$$\Delta x \triangleq Q_{p^{-1}}\overline{\Delta x} \ \text{ and } \ \Delta s \triangleq Q_p \underline{\Delta s}.$$

Finally, we take a step size α so that the new point $(x^+, y^+, s^+) \triangleq (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$ is generated in the neighborhood of the central path; see Figure 1.

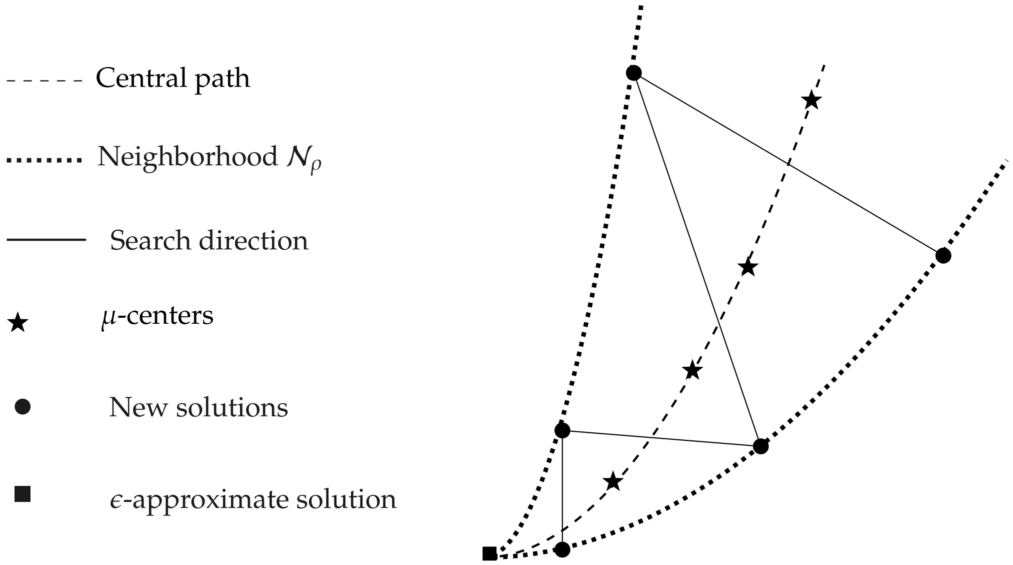

Figure 1. The alteration between steps to follow the central path.

5. Path-Following Interior-Point Algorithms

Primal–dual path-following IPMs for solving the pair (P, D) are introduced in this section in three different lengths: short, semi-long, and long-step.

The generation of each iteration $(x_\mu^{(k)}, y_\mu^{(k)}, s_\mu^{(k)})$ in the neighborhood of the central path $\mathcal{CP}$ is one of the main issues in the path-following IPMs, and we use proximity measure functions to handle this. With our adherence to the central path, the duality gap sequence $\{\mu_k\}$ will converge, and a bound on the number of iterations needed to obtain the optimal solution of the pair problem (P, D) will be polynomial.

The standard way to classify the proximity measures is to measure the distance to a specific point on the central path $\mathcal{CP}$. More specifically, the proximity measures for $x, s \succ \mathbf{0}$ are given as

$$\begin{array}{lclcl} d_{\mathrm{F}}(x,s) & \triangleq & \|Q_{x^{1/2}}s - \mu e\}\|_{\mathrm{F}} & = & \sqrt{\lambda_1(Q_{x^{1/2}}s-\mu)^2 + \lambda_2(Q_{x^{1/2}}s-\mu)^2}, \\ d_2(x,s) & \triangleq & \|Q_{x^{1/2}}s - \mu e\}\|_2 & = & \max\{|\lambda_1(Q_{x^{1/2}}s) - \mu|, |\lambda_2(Q_{x^{1/2}}s) - \mu|\}, \\ d_{-\infty}(x,s) & \triangleq & \mu - \lambda_{\min}(Q_{x^{1/2}}s) & = & \mu - \min\{\lambda_1(Q_{x^{1/2}}s), \lambda_2(Q_{x^{1/2}}s)\}. \end{array} \tag{14}$$

The three different distances in (14) lead to the following three neighborhoods along the central path:

$$\begin{array}{lcl} \mathcal{N}_{\mathrm{F}}(\gamma) & \triangleq & \{(x,y,s) \in \mathcal{F}^\circ : d_{\mathrm{F}}(x,s) \le \gamma\mu\}, \\ \mathcal{N}_2(\gamma) & \triangleq & \{(x,y,s) \in \mathcal{F}^\circ : d_2(x,s) \le \gamma\mu\}, \\ \mathcal{N}_{-\infty}(\gamma) & \triangleq & \{(x,y,s) \in \mathcal{F}^\circ : d_{-\infty}(x,s) \le \gamma\mu\}, \end{array} \tag{15}$$

where $\gamma \in (0, 1)$ is a constant known as the neighborhood parameter.

By Proposition 21 in [30] , both $Q_{x^{1/2}}s$ and $Q_{\overline{x}^{1/2}}\underline{s}$ have the same eigenvalues, and since all neighborhoods $\mathcal{N}_\rho(\gamma)$ may be described in terms of the eigenvalue of $Q_{x^{1/2}}s$, one can see that the neighborhoods defined in (15) are scaling-invariant, i.e., $(x, s) \in$

$\mathcal{N}_\rho(\gamma)$ if and only if $(\overline{\boldsymbol{x}}, \underline{\boldsymbol{s}}) \in \mathcal{N}_\rho(\gamma)$, where ρ can be selected as $F, 2$, or $-\infty$.

Furthermore, given the eigenvalue characterization of $d_\rho(\boldsymbol{x}, \boldsymbol{s})$, we can find that $d_{-\infty}(\boldsymbol{x}, \boldsymbol{s}) \leq d_2(\boldsymbol{x}, \boldsymbol{s}) \leq d_{\mathrm{F}}(\boldsymbol{x}, \boldsymbol{s})$, and hence $\mathcal{N}_{\mathrm{F}}(\gamma) \subseteq \mathcal{N}_2(\gamma) \subseteq \mathcal{N}_{-\infty}(\gamma)$.

The performance of the path-following IPMs for RQCP problems greatly depends on the neighborhood $\mathcal{N}_\rho(\gamma)$ of the central path and the centering parameter δ that we select. These options allow us to divide the path-following IPMs for our problem into three categories: Short, semi-long, and long-step. More specifically:

- Selecting $\mathcal{N}_{\mathrm{F}}(\gamma)$ as the neighborhood yields the short-step algorithm;
- Selecting $\mathcal{N}_2(\gamma)$ as the neighborhood yields the semi-long-step algorithm;
- Selecting $\mathcal{N}_{-\infty}(\gamma)$ as the neighborhood yields the long-step algorithm.

We indicate that the long-step version of the algorithm seems to outperform the short-step version of the algorithm in practical performance. In Table 2, we compare some of the categorized versions of this algorithm. The proposed path-following interior-point algorithm for solving the pair of problems (P, D) is described in Algorithm 1 and Figure 2.

Table 2. Contrasting some features of path-following IPMs in short, semi-long, and long-step versions.

Features	**Short-Step Algorithm**	**Semi-Long-Step Algorithm**	**Long-Step Algorithm**
Neighborhood $\mathcal{N}_\rho(\gamma)$	$\mathcal{N}_{\mathrm{F}}(\gamma)$	$\mathcal{N}_{-\infty}(\gamma)$	$\mathcal{N}_2(\gamma)$
Factor σ	$(0,1)$	$(0,1)$	$(0,1)$
Centering parameter δ	$1-\sigma/\sqrt{2r}$	$(0,1)$	$(0,1)$
No. of iter. in NT direction	$\mathcal{O}(\sqrt{2r}\log(1/\epsilon))$	$\mathcal{O}(2r\log(1/\epsilon))$	$\mathcal{O}(2r\log(1/\epsilon))$
No. of iter. in HRVW/KSH/M direction or its dual	$\mathcal{O}(\sqrt{2r}\log(1/\epsilon))$	$\mathcal{O}(2r\log(1/\epsilon))$	$\mathcal{O}((2r)^{3/2}\log(1/\epsilon))$

Algorithm 1: A path-following interior algorithm for solving RQCP problem.

```
Initialize: a tolerance $\epsilon$, a neighborhood parameter $\gamma$, a centering parameter $\delta$;
Ensure: $\epsilon > 0, \gamma \in (0,1), \delta \in (0,1)$;
choose an initial feasible solution $(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{s}^{(0)}) \in \mathcal{N}_\rho(\gamma)$;
set $\mu_0 = \frac{1}{2}\boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}$;
set $i = 0$;
while $\mu_i \geq \epsilon\mu_0$ do
    choose a scaling vector $\boldsymbol{p} \in \mathcal{C}(\boldsymbol{x}^{(i)}, \boldsymbol{s}^{(i)})$ ;
    find the Newton direction $(\overline{\Delta\boldsymbol{x}^{(i)}}, \Delta\boldsymbol{y}^{(i)}, \underline{\Delta\boldsymbol{s}^{(i)})}$ by solving System (13) ;
    compute $(\Delta\boldsymbol{x}^{(i)}, \Delta\boldsymbol{s}^{(i)})$ by applying inverse scaling to $(\overline{\Delta\boldsymbol{x}^{(i)}}, \underline{\Delta\boldsymbol{s}^{(i)})}$;
    choose a step length $\alpha^i$;
    obtain a new step from the previous step
      $(\boldsymbol{x}^{(i+1)}, \boldsymbol{y}^{(i+1)}, \boldsymbol{s}^{(i+1)}) = (\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \boldsymbol{s}^{(i)}) + \alpha^i(\Delta\boldsymbol{x}^{(i)}, \Delta\boldsymbol{y}^{(i)}, \Delta\boldsymbol{s}^{(i)}) \in \mathcal{N}_\rho(\gamma)$;
    reduce barrier parameter $\mu_{i+1} = \frac{1}{2}\boldsymbol{x}^{(i+1)} \bullet \boldsymbol{s}^{(i+1)}$;
    increment $i \leftarrow i+1$;
end while
```

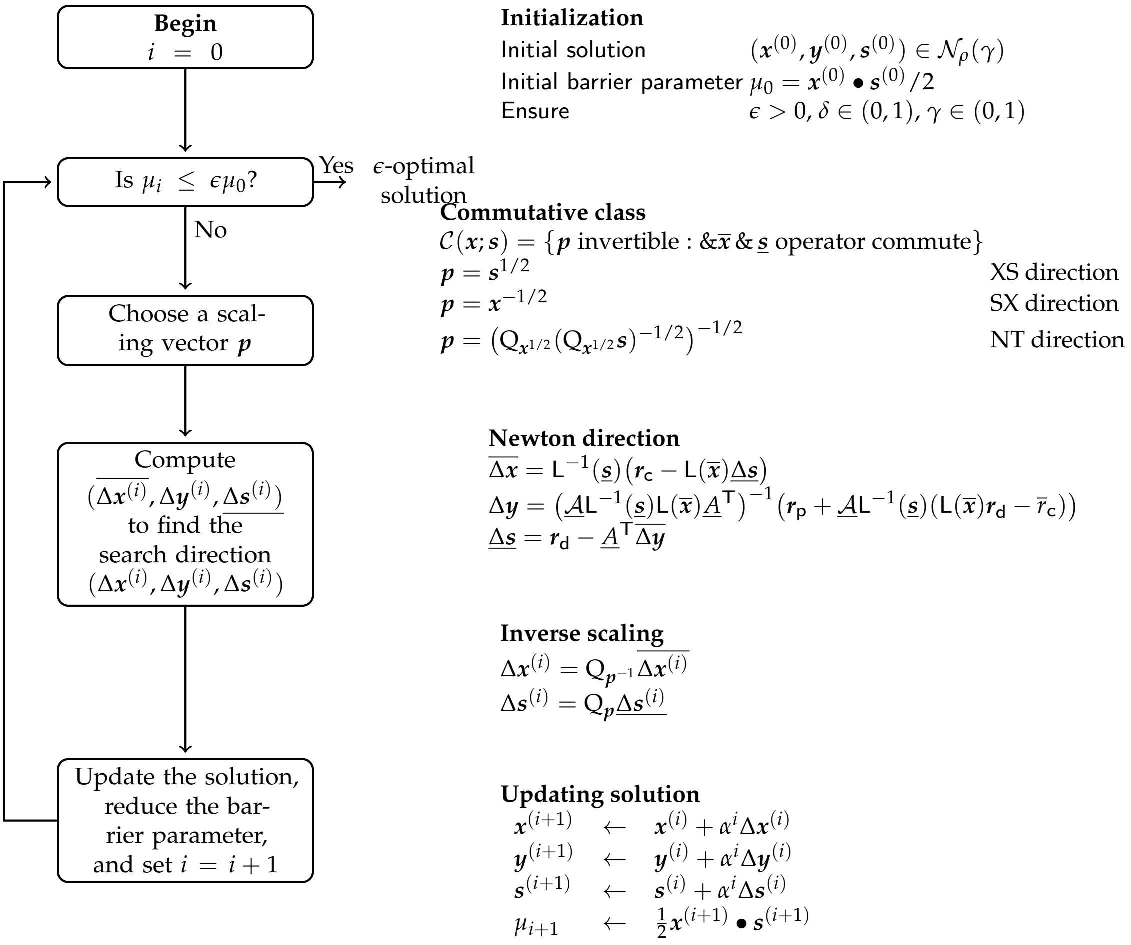
Begin
$i = 0$
Is $\mu_i \leq \epsilon\mu_0$?
Yes
ϵ-optimal solution
No
Choose a scaling vector $\boldsymbol{p}$
Compute $(\overline{\Delta \boldsymbol{x}^{(i)}}, \Delta \boldsymbol{y}^{(i)}, \underline{\Delta \boldsymbol{s}^{(i)}})$ to find the search direction $(\Delta \boldsymbol{x}^{(i)}, \Delta \boldsymbol{y}^{(i)}, \Delta \boldsymbol{s}^{(i)})$
Update the solution, reduce the barrier parameter, and set $i = i + 1$
Initialization
Initial solution $(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{s}^{(0)}) \in \mathcal{N}_\rho(\gamma)$
Initial barrier parameter $\mu_0 = \boldsymbol{x}^{(0)} \bullet \boldsymbol{s}^{(0)}/2$
Ensure $\epsilon > 0, \delta \in (0,1), \gamma \in (0,1)$
Commutative class
$\mathcal{C}(\boldsymbol{x};\boldsymbol{s}) = \{\boldsymbol{p}$ invertible : $\&\overline{x}\,\&\,\underline{s}$ operator commute$\}$
$\boldsymbol{p} = \boldsymbol{s}^{1/2}$ XS direction
$\boldsymbol{p} = \boldsymbol{x}^{-1/2}$ SX direction
$\boldsymbol{p} = \left(\mathrm{Q}_{\boldsymbol{x}^{1/2}}(\mathrm{Q}_{\boldsymbol{x}^{1/2}}\boldsymbol{s})^{-1/2}\right)^{-1/2}$ NT direction
Newton direction
$\overline{\Delta \boldsymbol{x}} = \mathsf{L}^{-1}(\underline{\boldsymbol{s}})\left(\boldsymbol{r}_{\mathsf{c}} - \mathsf{L}(\overline{\boldsymbol{x}})\underline{\Delta \boldsymbol{s}}\right)$
$\Delta \boldsymbol{y} = \left(\underline{A}\mathsf{L}^{-1}(\underline{\boldsymbol{s}})\mathsf{L}(\overline{\boldsymbol{x}})\underline{A}^{\mathsf{T}}\right)^{-1}\left(\boldsymbol{r}_{\mathsf{p}} + \underline{A}\mathsf{L}^{-1}(\underline{\boldsymbol{s}})(\mathsf{L}(\overline{\boldsymbol{x}})\boldsymbol{r}_{\mathsf{d}} - \overline{\boldsymbol{r}}_{\mathsf{c}})\right)$
$\underline{\Delta \boldsymbol{s}} = \boldsymbol{r}_{\mathsf{d}} - \underline{A}^{\mathsf{T}}\overline{\Delta \boldsymbol{y}}$
Inverse scaling
$\Delta \boldsymbol{x}^{(i)} = \mathrm{Q}_{\boldsymbol{p}^{-1}}\overline{\Delta \boldsymbol{x}^{(i)}}$
$\Delta \boldsymbol{s}^{(i)} = \mathrm{Q}_{\boldsymbol{p}}\underline{\Delta \boldsymbol{s}^{(i)}}$
Updating solution
$\boldsymbol{x}^{(i+1)} \leftarrow \boldsymbol{x}^{(i)} + \alpha^i \Delta \boldsymbol{x}^{(i)}$
$\boldsymbol{y}^{(i+1)} \leftarrow \boldsymbol{y}^{(i)} + \alpha^i \Delta \boldsymbol{y}^{(i)}$
$\boldsymbol{s}^{(i+1)} \leftarrow \boldsymbol{s}^{(i)} + \alpha^i \Delta \boldsymbol{s}^{(i)}$
$\mu_{i+1} \leftarrow \frac{1}{2}\boldsymbol{x}^{(i+1)} \bullet \boldsymbol{s}^{(i+1)}$

Figure 2. A pseudocode for Algorithm 1.

The convergence and time complexity for Algorithm 1 is given in the following theorem. This theorem is a consequence of Theorem 1 that we verified in Section 2 and Theorem 37 in [30] after taking the rotated quadratic cone $\mathcal{R}^n_+$ as our underlying symmetric cone.

Theorem 3. *If each iteration in Algorithm 1 follows the NT direction, then the short-step algorithm terminates in $\mathcal{O}((2r)^{1/2}\log(1/\epsilon))$ iterations, and the semi-long and long-step algorithms terminate in $\mathcal{O}(2r\log(1/\epsilon))$ iterations. If each iteration in Algorithm 1 follows the HRVW/KSH/M direction or dual HRVW/KSH/M direction, then the short-step algorithm terminates in $\mathcal{O}((2r)^{1/2}\log(1/\epsilon))$ iterations, the semi-long-step algorithm terminates in $\mathcal{O}(2r\log(1/\epsilon))$ iterations, and the long-step algorithm terminates in $\mathcal{O}((2r)^{3/2}\log(1/\epsilon))$ iterations.*

Before closing this section and moving forward to our numerical results below, we indicate that the above analysis can be carried out and extended word-by-word if Problem (P, D) is given in multiple block-setting by using the notations introduced in Table 1.

6. Numerical Results

In this section, we will assess how well the proposed method performs when implemented to some RQCP problems: The problem of minimizing the harmonic mean of positive affine function, and some randomly generating problems. We also contrast the numerical results of the proposed algorithm for the randomly generated problems with those of the two symmetric cone programming software package systems: CVX [33] and MOSEK [34].

We produced our numerical results using MATLAB version R2021a on a PC with an Intel (R) Core (TM) i3-1005G1 processor operating at 1.20 GHz and 4 GB of physical memory. In the first numerical example, we consider a problem in multiple block setting. The dimensions of our test problems are denoted by the letters m and n, and the number of block splittings is denoted by r. "Iter" stands for the typical number of iterations needed to obtain ϵ-optimal solutions, while "CPU" stands for the typical CPU time needed to reach an ϵ-optimal solution to the underlying problem.

In all of our tests, we use the long-step path-following version of Algorithm 1, and choose the dual HRVW/KSH/M direction (i.e., our scaling vector is $\boldsymbol{p} = \boldsymbol{x}^{-1/2}$). Furthermore, when the condition of convergence is not satisfied at an iteration step, it will generate a new solution that must be in the neighborhood of the central path. We choose the step size of α that was proposed in [30].

Example 1 (Minimizing the harmonic mean of affine functions)**.** *We consider the problem of minimizing the harmonic mean of (positive) affine functions of $\boldsymbol{x} \in \mathbb{R}^n$ [7]:*

$$\begin{array}{llcll}
\min & \displaystyle\sum_{i=1}^{r} \frac{1}{\boldsymbol{a}_i^{\mathsf{T}}\boldsymbol{x} + \beta_i} & & \min & \displaystyle\sum_{i=1}^{r} u_i \\
\text{s.t.} & \boldsymbol{a}_i^{\mathsf{T}}\boldsymbol{x} + \beta_i > 0,\ i = 1,2,\ldots,r, & \textit{which can be cast as} & \text{s.t.} & u_i(\boldsymbol{a}_i^{\mathsf{T}}\boldsymbol{x} + \beta_i) \geq 1,\ i = 1,2,\ldots,r, \\
& \boldsymbol{d}_j^{\mathsf{T}}\boldsymbol{x} + h_j \geq 0,\ j = 1,2,\ldots,m; & \textit{the RQCP problem} & & u_i \geq 0,\ i = 1,2,\ldots,r, \\
& & & & \boldsymbol{d}_j^{\mathsf{T}}\boldsymbol{x} + h_j \geq 0,\ j = 1,2,\ldots,m.
\end{array}$$

We implement this problem for sizes $n = 6, 12, 24, 30, 36$, $m = 3, 9, 15$, and the number of blocks $r = 5, 10, 15, 20$. We generate the coefficients $\boldsymbol{a}_i, \beta_i, \boldsymbol{d}_j$ and h_j randomly from a list of numbers uniformly distributed between -1 and 1 for all $i = 1,2,\ldots,r$ and $j = 1,2,\ldots,m$. We take the parameters $\epsilon = 10^{-6}$, $\sigma = 0.55$, and $\gamma = 0.8$. The initial solutions of $\boldsymbol{x}^{(0)}, \boldsymbol{s}^{(0)}$ are also chosen randomly from a list of uniformly distributed numbers between -1 and 1, while $\boldsymbol{y}^{(0)}$ is chosen as the zero vector, and $u_i, i = 1, 2, \cdots, r$, are all set to take values between 0 and 1. We display the numerical results obtained for this example in Table 3, and visualize them graphically in Figure 3.

Example 2 (Randomly generated problems)**.** *In this example, the coefficients A and b are generated at random from a list of uniformly distributed numbers between -1 and 1. We set $\boldsymbol{b} = \mathcal{A}\boldsymbol{x}^{(0)}$ and $\boldsymbol{c} = A^{\mathsf{T}}\boldsymbol{y}^{(0)} + \boldsymbol{s}^{(0)}$, and choose the parameters $\epsilon = 10^{-4}$, $\sigma = 0.33$ and $\gamma = 0.9$. The size of the problem is given so that $n = 2m$, where m is ranging from 5 to 1000. The initial solutions are chosen as follows: $\boldsymbol{x}^{(0)} = \boldsymbol{e}$, $\boldsymbol{s}^{(0)} = \boldsymbol{e}$ and $\boldsymbol{y}^{(0)} = \boldsymbol{0}$. We display our numerical results in Table 4, and visualize them graphically in Figure 4. The results from the CVX and MOSEK solvers are also presented in Table 4 and Figure 4 for comparison purposes.*

Table 3. The numerical results obtained for minimizing harmonic mean of affine functions in Example 1.

m	n	r	Iter.	CPU(s)	m	n	r	Iter.	CPU(s)	m	n	r	Iter.	CPU(s)
3	6	5	2	0.0156	9	6	5	8	0.1562	15	6	5	32	0.4062
3	6	10	4	0.0781	9	6	10	16	0.0781	15	6	10	42	0.3125
3	6	15	5	0.1094	9	6	15	20	0.4844	15	6	15	44	0.5781
3	6	20	5	0.1562	9	6	20	15	0.5938	15	6	20	57	0.4844
3	12	5	5	0.0312	9	12	5	24	0.1094	15	12	5	27	0.5406
3	12	10	7	0.1094	9	12	10	27	0.2188	15	12	10	32	0.5546
3	12	15	8	0.2031	9	12	15	27	0.4804	15	12	15	24	0.7679
3	12	20	11	0.2931	9	12	20	32	0.5000	15	12	20	31	0.5932
3	24	5	11	0.1719	9	24	5	22	0.1875	15	24	5	36	0.5381
3	24	10	15	0.2500	9	24	10	25	0.2969	15	24	10	31	0.8344
3	24	15	16	0.1562	9	24	15	29	0.3750	15	24	15	52	0.9306
3	24	20	19	0.3901	9	24	20	33	0.5156	15	24	20	49	0.8656

Table 3. *Cont.*

m	n	r	Iter.	CPU(s)	m	n	r	Iter.	CPU(s)	m	n	r	Iter.	CPU(s)
3	30	5	16	0.1904	9	30	5	23	0.1875	15	30	5	40	0.6265
3	30	10	20	0.1406	9	30	10	39	0.2969	15	30	10	44	0.9018
3	30	15	23	0.2812	9	30	15	23	0.2344	15	30	15	49	0.8031
3	30	20	26	0.3112	9	30	20	41	0.5469	15	30	20	54	0.8750
3	36	5	32	0.3938	9	36	5	28	0.1719	15	36	5	52	0.6925
3	36	10	19	0.2998	9	36	10	35	0.3906	15	36	10	54	0.7562
3	36	15	34	0.3438	9	36	15	39	0.3438	15	36	15	69	0.7906
3	36	20	26	0.5469	9	36	20	47	0.3125	15	36	20	72	0.9789

Table 4. The numerical results obtained for the randomly generated problems in Example 2.

Problem Size	Algorithm 1		CVX		MOSEK	
(m, n)	Iter.	CPU(s)	Iter.	CPU(s)	Iter.	CPU(s)
(5, 10)	3	0.0156	4	0.28	5	0.22
(10, 20)	5	0.0313	6	0.30	8	0.18
(20, 40)	5	0.0625	6	0.39	9	0.20
(30, 60)	6	0.1406	6	0.20	9	0.19
(40, 80)	6	0.1250	5	0.33	8	0.38
(50, 60)	9	0.0781	6	0.25	8	0.62
(60, 120)	12	0.2086	7	0.30	10	0.41
(70, 140)	11	0.2188	12	0.52	11	0.79
(80,160)	14	0.3100	21	0.38	17	0.60
(90, 180)	15	0.3343	13	0.49	15	0.79
(100, 200)	10	0.4531	14	0.66	19	0.74
(200, 400)	12	1.0900	15	0.44	8	1.85
(300,600)	7	4.0313	9	0.53	9	5.41
(400, 800)	10	9.804	14	1.02	19	11.97
(500, 1000)	22	18.6406	29	11.78	31	16.90
(600, 1200)	25	27.5938	30	41.03	37	31.67
(700, 1400)	39	48.2500	43	51.97	50	65.36
(800, 1600)	61	61.2813	73	72.77	86	94.64
(900, 1800)	53	65.4651	34	70.50	71	83.45
(1000, 2000)	66	74.7968	69	79.32	76	91.54

We can see overall that the computational results show that Algorithm 1 performs well in practice. We can see from the numerical results shown in Tables 3 and 4 and represented in Figures 3 and 4 that the number of iterations and CPU time required by Algorithm 1 increase as the dimension of the underlying problem increases, indicating that the dimension of the problem and the dimension of rotated quadratic cones have an impact on the number of iterations and the amount of time required by the proposed algorithm. Furthermore, when the randomly generated problems are solved using the CVX and Mosek solvers, we can see that most of the solutions exhibited a slight bias toward Algorithm 1 in terms of both the number of iterations and CPU time. This is most likely because these solvers begin at infeasible points or because the stopping condition of the algorithm used in these solvers differs from that of Algorithm 1.

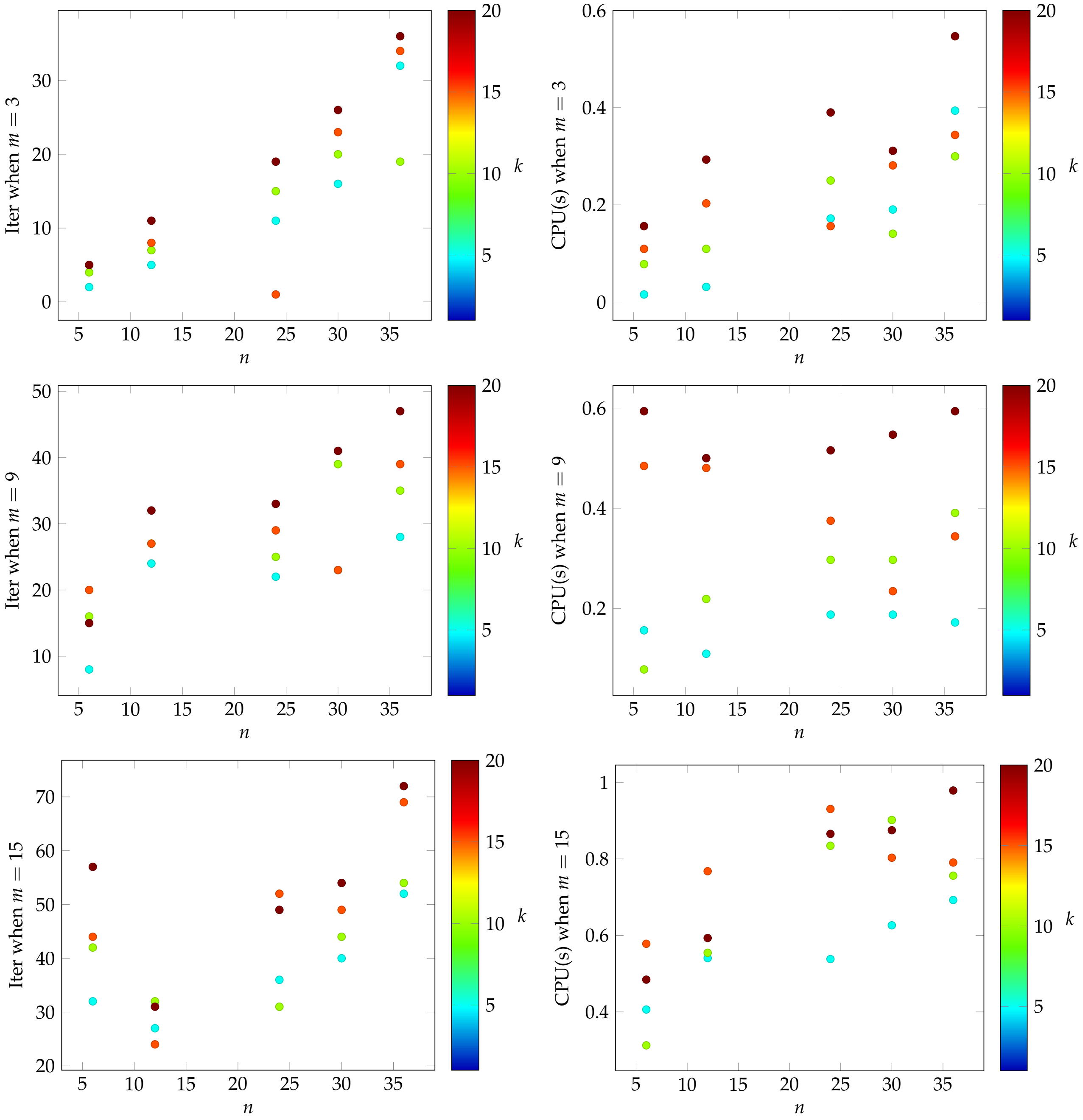

Figure 3. Dot plots of the numerical results obtained in Example 1.

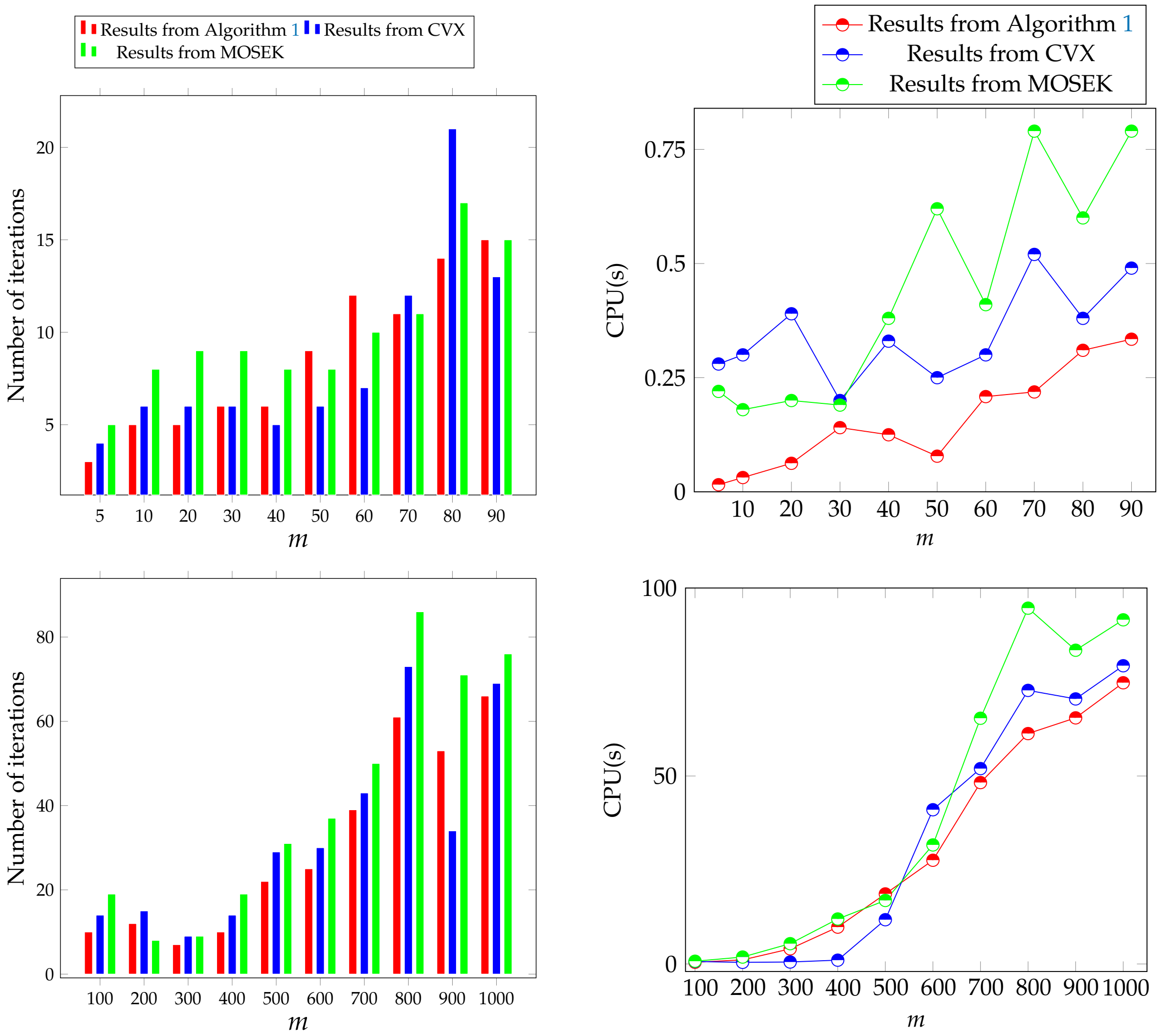

Figure 4. Two-dimensional plots for the numerical results in Example 2.

7. Concluding Remarks

All earlier work on optimization problems over the rotated quadratic cones has formulated these problems as second-order cone programming problems, and while doing this can be easier than developing special-purpose algorithms for solving this class of optimization problems, this approach may not always be the cheapest one in terms of computational cost. In this paper, we have introduced the rotated quadratic cone programming problems as a "self-made" class of optimization problems. We have proved that the barrier function associated with our cone is strongly self-concordant. We have discussed the duality theory associated with these problems, along with the development of the commutative class of search directions, and have developed a primal–dual interior-point algorithm rotated quadratic cone optimization problems based on our own Euclidean Jordan algebra. The efficiency of the proposed algorithm is shown by providing some numerical examples and comparing some of them with results from MOSEK and CVX solvers.

The proposed algorithm is attractive from an algebraic point of view. Most of this attractiveness comes from exploiting the algebraic structure of the quadratic rotated cone which allowed us to explicitly give expressions for the inverse operator, the linear representation, and the quadratic operator, and use these operators to compute the derivatives of the barrier function explicitly. In spite of its attractiveness, the algorithm has several limitations such as producing a good starting point, developing a practical step length selection procedure in the primal space, and reducing the barrier parameter with a practical strategy in our setting. These limitations, however, could be addressed in a future research paper developing practical implementations. Future work can be performed on developing algorithms for solving mixed-integer rotated quadratic cone optimization problems.

Author Contributions: K.T. and B.A. conceived the idea, set up the analysis, wrote the proofs, designed and performed the experiments, analyzed the results, drafted the initial manuscript, and revised the final manuscript. All authors have read and agreed to this version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors thank Blake Whitman from The Ohio State University for reading the manuscript and pointing out some printing errors. The authors also thank the two anonymous referees for their constructive comments and suggestions for improvements.

Conflicts of Interest: The authors have no competing interests to declare.

Abbreviations

The following abbreviations are used in this manuscript:

RQCP	Rotated quadratic cone programming
EJA	Euclidean Jordan algebra
IPM	Interior-point method
KKT	Karush–Kuhn–Tucker
Iter	Number of iterations
CPU	Central processing unit

References

1. Montoya, O.; Gil-González, W.; Garcés, A. On the conic convex approximation to locate and size fixed-step capacitor banks in distribution networks. *Computation* **2022**, *10*, 32. [CrossRef]
2. Alzalg, B. A primal-dual interior-point method based on various selections of displacement step for symmetric optimization. *Comput. Optim. Appl.* **2019**, *72*, 363–390. [CrossRef]
3. Nesterov, Y.; Nemirovskii, A. *Interior-Point Polynomial Algorithms in Convex Programming*; SIAM: Philadelphia, PA, USA, **1994**.
4. Nesterov, Y.E.; Todd, M.J. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Res.* **1997**, *22*, 1–42. [CrossRef]
5. Manshadi, S.D.; Liu, G.; Khodayar, M.E.; Wang, J.; Dai, R. A convex relaxation approach for power flow problem. *J. Mod. Power Syst. Clean Energy* **2019**, *7*, 1399–1410. [CrossRef]
6. Manshadi, S.D.; Liu, G.; Khodayar, M.E.; Wang, J.; Dai, R. A distributed convex relaxation approach to solve the power flow problem. *IEEE Syst. J.* **2019**, *14*, 803–812. [CrossRef]
7. Alizadeh, F.; Goldfarb, D. Second-order cone programming. *Math. Program.* **2003**, *95*, 3–51. [CrossRef]
8. Alzalg, B.; Pirhaji, M. Elliptic cone optimization and primal–dual path-following algorithms. *Optimization* **2017**, *66*, 2245–2274. [CrossRef]
9. Malakar, S.; Ghosh, M.; Bhowmik, S.; Sarkar, R.; Nasipuri, M. A GA based hierarchical feature selection approach for handwritten word recognition. *Neural. Comput. Appl.* **2020**, *32*, 2533–2552. [CrossRef]
10. Bacanin, N.; Stoean, R.; Zivkovic, M.; Petrovic, A.; Rashid, T.A.; Bezdan, T. Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization. *Mathematics* **2021**, *9*, 2705. [CrossRef]
11. Tuba, E.; Bacanin, N. An algorithm for handwritten digit recognition using projection histograms and SVM classifier. In Proceedings of the 2015 23rd Telecommunications Forum Telfor (TELFOR), Belgrade, Serbia, 24–26 November 2015; pp. 464–467.
12. Karmarkar, N. A new polynomial-time algorithm for linear programming. In Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing, Washington, DC, USA, 30 April–2 May 1984; pp. 302–311.
13. Güler, O. Barrier functions in interior point methods. *Math. Oper. Res.* **1996**, *21*, 860–885. [CrossRef]
14. Faraut, J. *Analysis on Symmetric Cones*; Oxford Mathematical Monographs: Oxford, UK, 1994.
15. Todd, M. Semidefinite optimization. *Acta Numer.* **2001**, *10*, 515–560. [CrossRef]
16. Alzalg, B. *Combinatorial and Algorithmic Mathematics: From Foundation to Optimization*; Kindle Direct Publishing: Seattle, WA, USA, 2022.

17. Alzalg, B. Homogeneous self-dual algorithms for stochastic second-order cone programming. *J. Optim. Theory Appl.* **2014**, *163*, 148–164. [CrossRef]
18. Alzalg, B. Volumetric barrier decomposition algorithms for stochastic quadratic second-order cone programming. *Appl. Math. Comput.* **2015**, *265*, 494–508. [CrossRef]
19. Alzalg, B. A logarithmic barrier interior-point method based on majorant functions for second-order cone programming. *Optim. Lett.* **2020**, *14*, 729–746. [CrossRef]
20. Alzalg, B.; Badarneh, K.; Ababneh, A. An infeasible interior-point algorithm for stochastic second-order cone optimization. *J. Optim. Theory Appl.* **2019**, *181*, 324–346. [CrossRef]
21. Alzalg, B.; Gafour, A.; Alzaleq, L. Volumetric barrier cutting plane algorithms for stochastic linear semi-infinite optimization. *IEEE Access* **2019**, *8*, 4995–5008. [CrossRef]
22. Alzalg, B.; Pirhaji, M. Primal-dual path-following algorithms for circular programming. *Commun. Comb. Optim.* **2017**, *2*, 65–85.
23. Kojima, M.; Mizuno, S.; Yoshise, A. A primal-dual interior point algorithm for linear programming. In *Progress in Math. Program*; Springer: New York, NY, USA, 1989; pp. 29–47.
24. Monteiro, R.D.; Adler, I. Interior Path Following Primal-Dual Algorithms. Part I: Linear Programming. *Math. Program.* **1989**, *44*, 27–41. [CrossRef]
25. Alzalg, B. Primal interior-point decomposition algorithms for two-stage stochastic extended second-order cone programming. *Optimization* **2018**, *67*, 2291–2323. [CrossRef]
26. Goldfarb, D.; Liu, S. An $O(n^3L)$ Primal interior point algorithm for convex quadratic programming. *Math. Program.* **1991**, *49*, 325–340. [CrossRef]
27. Tiande, G.; Shiquan, W. Properties of primal interior point methods for QP. *Optimization* **1996**, *37*, 227–238. [CrossRef]
28. Alzalg, B.; Tamsaouete, K.; Benakkouche, L.; Ababneh, A. The Jordan Algebraic Structure of the Rotated Quadratic Cone. Submitted for Publication. 2017. Available online: https://optimization-online.org/wp-content/uploads/2023/01/RQC.pdf (accessed on 31 January 2023).
29. Monteiro, R.D.C.; Zhang, Y. A unified analysis for a class of path-following primal-dual interior-point algorithms for semidefinite programming. *Math. Program.* **1998**, *81*, 281–299. [CrossRef]
30. Schmieta, S.H.; Alizadeh, F. Extension of primal-dual interior point algorithms to symmetric cones. *Math. Program.* **2003**, *96*, 409–438. [CrossRef]
31. Helmberg, C.; Rendl, F.; Vanderbei, R.J.; Wolkowicz, H. An interior-point method for semidefinite programming. *SIAM J. Optim.* **1996**, *6*, 342–361. [CrossRef]
32. Kojima, M.; Shindoh, S.; Hara, S. Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM J. Optim.* **1997**, *7*, 86–125. [CrossRef]
33. Grant, M.; Boyd, S.; Ye, Y. *CVX: Matlab Software for Disciplined Convex Programming (Webpage and Software)*; CVX Research, Inc.: Austin, TX, USA, 2009.
34. Mosek ApS. Mosek Optimization Toolbox for Matlab. In *User's Guide and Reference Manual*; Version 4; Mosek ApS: Copenhagen, Denmark, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.