



An Infeasible Interior-Point Algorithm for Stochastic Second-Order Cone Optimization

Baha Alzalg^{1,2} · Khaled Badarneh¹ · Ayat Ababneh^{1,3}

Received: 24 March 2017 / Accepted: 13 November 2018 / Published online: 22 November 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Alzalg (J Optim Theory Appl 163(1):148–164, 2014) derived a homogeneous self-dual algorithm for stochastic second-order cone programs with finite event space. In this paper, we derive an infeasible interior-point algorithm for the same stochastic optimization problem by utilizing the work of Rangarajan (SIAM J Optim 16(4), 1211–1229, 2006) for deterministic symmetric cone programs. We show that the infeasible interior-point algorithm developed in this paper has complexity less than that of the homogeneous self-dual algorithm mentioned above. We implement the proposed algorithm to show that they are efficient.

Keywords Second-order cone programming · Stochastic programming · Infeasible interior-point algorithms · Euclidean Jordan algebra

Mathematics Subject Classification 90C25 · 90C06 · 90C15 · 90C30 · 90C51 · 90C60

1 Introduction

Two-stage stochastic second-order cone programs [1] have been defined to formulate many applications of second-order cone programs [2] that involve uncertainty in

Communicated by Florian Potra.

✉ Baha Alzalg
b.alzalg@ju.edu.jo
Khaled Badarneh
firso_ojo@yahoo.com
Ayat Ababneh
ababneh.3@osu.edu

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

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

³ Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA

data. General and different classes of optimization problems, such as stochastic linear programs, stochastic quadratic programs, and quadratically constrained stochastic quadratic programs, can be recast as stochastic second-order cone programs [1,2]. In [1], the reader can find various application models of stochastic second-order cone programs, namely the stochastic Euclidean facility location problem, the portfolio optimization problem with loss risk constraints, the optimal covering random ellipsoid problem, and an application in structural optimization.

Alzalg and Ariyawansa [3] (see also Alzalg [4,5]) proposed polynomial-time decomposition-based interior-point algorithms for stochastic programs over all symmetric cones including second-order cones. While the decomposition-based interior-point algorithms exploit the structure in a problem, there are other classes of algorithms such as infeasible interior-point algorithms and homogeneous self-dual algorithms, which are particularly designed for general problems. However, when applied to problems with structure, in many cases homogeneous and infeasible interior-point algorithms can be designed to exploit the structure by carefully tailoring the operations of the algorithm. In [6], Alzalg has derived a homogeneous self-dual algorithm for stochastic second-order cone programs with finite event space of their random variables. In this paper, the focus is on developing an infeasible interior-point algorithm for stochastic second-order cone programs.

For *deterministic* nonlinear programming problems, several infeasible interior-point methods have been proposed. See, for examples, the work of Nesterov [7] in nonlinear programming, the work of Rangarajan [8] over symmetric cones, the work of Potra and Sheng [9] over the semidefinite cone, and the work of Xiaoni and Sanyang [10] over the second-order cone. See also the work of Castro [11] and that of Castro and Cuesta [12] which present interior-point methods for large-scale deterministic optimization problems by exploiting block-angular structures. As mentioned earlier, we are concerned to derive an infeasible interior-point algorithm for stochastic second-order cone programs with finite event space of their random variables by utilizing the work of Rangarajan [8]. We will see that the infeasible algorithm developed for these problems has complexity less than that of the homogeneous algorithm developed in [6] for the same problems.

We organize this paper as follows. In Sect. 2, we briefly review some preliminary definitions, symbols, and notations from the theory of the Jordan algebra associated with the second-order cone. Section 3 presents the definition of a stochastic second-order cone programming with finite event space in primal standard form as defined in [1]. In Sect. 4, we exploit the special structure in our stochastic second-order cone programming problem to present a method for computing the search direction. In Sect. 5, we state the infeasible interior-point algorithm for solving the stochastic second-order cone programming problem and estimate its computational complexity with the method presented and analyzed in Sect. 4 and applied on our problem. In Sect. 6, we test the generic infeasible interior-point algorithm for stochastic second-order cone programs with some numerical examples. Section 7 gives some concluding remarks.

2 The Algebra of the Second-Order Cone: Definitions and Notations

The definitions and notations presented in this section are necessary to proceed with our subsequent development. Those who are not familiar with the basics of the theory of algebra of the second-order cone are highly encouraged to read [2, Section 4].

We use “;” for adjoining vectors and matrices in a row and use “;” for adjoining them in a column. For each vector $x \in \mathbb{R}^n$ whose first entry is indexed with 0, we write \tilde{x} for the subvector consisting of entries 1 through $n - 1$ (therefore $x = (x_0, \tilde{x}^\top)^\top \in \mathbb{R} \times \mathbb{R}^{n-1}$). We let \mathcal{E}^n denote the n^{th} -dimensional real vector space $\mathbb{R} \times \mathbb{R}^{n-1}$ whose elements x are indexed from 0.

In Table 1, we list some symbols and notations of the theory of the Jordan algebra associated with the second-order cone. This table presents these notations and definitions in both single and block-settings. The n^{th} -dimensional second-order cone, which is denoted by \mathcal{E}_+^n and defined in Table 1, is the norm cone for the Euclidean norm.

We point out that the direct sum (of two square matrices A and B) used in Table 1 is the block diagonal matrix

$$A \oplus B := \begin{bmatrix} A & O \\ \hat{O} & B \end{bmatrix}.$$

where O and \hat{O} are zero matrices with appropriate dimensions.

Note that the idempotents $c_1(x)$ and $c_2(x)$ defined in Table 1 satisfy the properties

$$\begin{aligned} c_1(x) \circ c_2(x) &= \mathbf{0}, \\ c_1(x) \circ c_1(x) &= c_1(x), \\ c_2(x) \circ c_2(x) &= c_2(x), \\ c_1(x) + c_2(x) &= e. \end{aligned}$$

Any pair of vectors $\{c_1, c_2\}$ that satisfies the above properties is called a *Jordan frame*. We say that elements x and y *operator commute* iff they share a Jordan frame, i.e., $x = \lambda_1 c_1 + \lambda_2 c_2$ and $y = \omega_1 c_1 + \omega_2 c_2$ for a Jordan frame $\{c_1, c_2\}$. In block-setting, the vectors $x = (x_1; x_2; \dots; x_r)$ and $y = (y_1; y_2; \dots; y_r)$ operator commute iff x_i and y_i operator commute for each $i = 1, 2, \dots, r$.

3 Stochastic Second-Order Cone Programs with Finite Event Space

We first write the stochastic second-order cone programming (SSOCP) problem in its general definition as defined in [1, Section 2], and then we consider the special case in which the event space is finite with K realizations.

Let $r \geq 1$ be an integer. For $i = 1, 2, \dots, r$, let $m_0, m_1, n_0, n_{0i}, \bar{n}, \bar{n}_i$ be positive integers such that $n_0 = \sum_{i=1}^r n_{0i}$ and $\bar{n} = \sum_{i=1}^r \bar{n}_i$. The two-stage SSOCP with recourse is the problem:

Table 1 Notations of the theory of the Jordan algebra associated with the second-order cone

Symbol	Definition and description
<i>In single-setting</i>	
$\ \cdot\ $	The <i>Euclidean norm</i> which is defined as $\ u\ = \left(\sum_{i=1}^n u_i ^2\right)^{1/2}$ for $u \in \mathbb{R}^n$
\mathcal{E}^n	The n -dimensional real vector space $\mathbb{R} \times \mathbb{R}^{n-1}$ whose elements are indexed from 0
\mathcal{E}_+^n	The <i>second-order cone</i> of dimension n defined as $\mathcal{E}_+^n := \{x \in \mathcal{E}^n : x_0 \geq \ \tilde{x}\ \}$
e	The <i>identity</i> of \mathcal{E}^n which is defined as $e := (1; \mathbf{0})$
$x \succeq \mathbf{0}$ ($x \succ \mathbf{0}$)	means that $x \in \mathcal{E}_+^n$ ($x \in \text{int}(\mathcal{E}_+^n)$)
$x \succeq y$ or $y \preceq x$	$x - y \succeq \mathbf{0}$. (We also write $x \succ y$ or $y \prec x$ to mean that $x - y \succ \mathbf{0}$)
$\text{Arw}(x)$	The <i>arrow-shaped matrix</i> associated with $x \in \mathcal{E}^n$; $\text{Arw}(x) := \begin{bmatrix} x_0 & \tilde{x}^\top \\ \tilde{x} & x_0 I_{n-1} \end{bmatrix}$
Q_x	The <i>quadratic representation</i> of $x \in \mathcal{E}^n$ which is defined as $Q_x := 2\text{Arw}^2(x) - \text{Arw}(x^2) = \begin{bmatrix} \ x\ ^2 & 2x_0\tilde{x}^\top \\ 2x_0\tilde{x} & \det(x)I + 2\tilde{x}\tilde{x}^\top \end{bmatrix}$
$x \circ y$	The <i>bilinear map</i> $\circ : \mathcal{E}^n \times \mathcal{E}^n \rightarrow \mathcal{E}^n$; $x \circ y := \begin{bmatrix} x^\top y \\ x_0\tilde{y} + y_0\tilde{x} \end{bmatrix} = \text{Arw}(x)y$
$\lambda_1(x), \lambda_2(x)$	The <i>eigenvalues</i> of x which are defined as $\lambda_{1,2}(x) := x_0 \pm \ \tilde{x}\ $
$c_1(x), c_2(x)$	The <i>eigenvectors</i> of x which are defined as $c_{1,2}(x) := \frac{1}{2} \left(1; \pm \frac{\tilde{x}}{\ \tilde{x}\ }\right)$
x^{-1}	The <i>inverse</i> of x ; $x^{-1} := \frac{1}{\lambda_1(x)}c_1(x) + \frac{1}{\lambda_2(x)}c_2(x)$ (provided that $\lambda_1, \lambda_2 \neq 0$)
x^n	Same as $x^{n-1} \circ x$ for $n \geq 2$; $x^n := \lambda_1^n(x)c_1(x) + \lambda_2^n(x)c_2(x)$
$x^{1/2}$	The <i>square root</i> of x ; $x^{1/2} := \sqrt{\lambda_1(x)}c_1(x) + \sqrt{\lambda_2(x)}c_2(x)$ (provided that $\lambda_1, \lambda_2 > 0$)
$\text{tr}(x)$	The <i>trace</i> of x which is defined as $\text{tr}(x) := \lambda_1(x) + \lambda_2(x) = 2x_0$
$\det(x)$	The <i>determinant</i> of x which is defined as $\det(x) := \lambda_1(x)\lambda_2(x) = x_0^2 - \ \tilde{x}\ ^2$
$x^\top y$	The <i>Frobenius inner product</i> of x and y defined as $x^\top y := \frac{1}{2}\text{tr}(x \circ y)$
<i>In block-setting</i>	
\mathcal{E}	$\mathcal{E} := \mathcal{E}^{n_1} \times \mathcal{E}^{n_2} \times \dots \times \mathcal{E}^{n_r}$
\mathcal{E}_+	$\mathcal{E}_+ := \mathcal{E}_+^{n_1} \times \mathcal{E}_+^{n_2} \times \dots \times \mathcal{E}_+^{n_r}$
$x \in \mathcal{E}$	$x := (x_1; x_2; \dots; x_r)$ and each $x_i \in \mathcal{E}^{n_i}$ for $i = 1, 2, \dots, r$
e	The identity of \mathcal{E} ; $e := (e_1; e_2; \dots; e_r)$
$x \succeq \mathbf{0}$ ($x \succ \mathbf{0}$)	$x_i \succeq \mathbf{0}$ ($x_i \succ \mathbf{0}$) for $i = 1, 2, \dots, r$
$\text{Arw}(x)$	$\text{Arw}(x) := \text{Arw}(x_1) \oplus \text{Arw}(x_2) \oplus \dots \oplus \text{Arw}(x_r)$
Q_x	$Q_x := Q_{x_1} \oplus Q_{x_2} \oplus \dots \oplus Q_{x_r}$
$x \circ y$	$x \circ y := (x_1 \circ y_1; x_2 \circ y_2; \dots; x_r \circ y_r)$
x^{-1}	$x^{-1} := (x_1^{-1}; x_2^{-1}; \dots; x_r^{-1})$ (provided that each x_i is invertible)
x^n	$x^n := (x_1^n; x_2^n; \dots; x_r^n)$ for $n \geq 2$
$x^{1/2}$	$x^{1/2} := (x_1^{1/2}; x_2^{1/2}; \dots; x_r^{1/2})$
$\text{tr}(x)$	$\text{tr}(x) := \sum_{i=1}^r \text{tr}(x_i)$
$\det(x)$	$\det(x) := \prod_{i=1}^r \det(x_i)$
$x^\top y$	$x^\top y := x_1^\top y_1 + x_2^\top y_2 + \dots + x_r^\top y_r$

$$\begin{aligned} \min \quad & \eta(\mathbf{x}_0) := \mathbf{c}_0^\top \mathbf{x}_0 + \mathbb{E} [Q(\mathbf{x}_0, \omega)] \\ \text{s.t.} \quad & W_0 \mathbf{x}_0 = \mathbf{h}_0, \\ & \mathbf{x}_0 \geq \mathbf{0}, \end{aligned} \quad (1)$$

where $\mathbb{E}[Q(\mathbf{x}_0, \omega)] := \int_{\Omega} Q(\mathbf{x}_0, \omega) P(d\omega)$ and $Q(\mathbf{x}_0, \omega)$ is the minimum value of the problem

$$\begin{aligned} \min \quad & \mathbf{c}(\omega)^\top \mathbf{x}(\omega) \\ \text{s.t.} \quad & W(\omega) \mathbf{x}(\omega) = \mathbf{h}(\omega) - B(\omega) \mathbf{x}_0, \\ & \mathbf{x}(\omega) \geq \mathbf{0}, \end{aligned} \quad (2)$$

where $\mathbf{x}_0 := (\mathbf{x}_{01}; \mathbf{x}_{02}; \dots; \mathbf{x}_{0r}) \in \mathcal{E}^{n_{01}} \times \mathcal{E}^{n_{02}} \times \dots \times \mathcal{E}^{n_{0r}}$ is the first-stage decision variable, $\mathbf{x}(\omega) := (\mathbf{x}_1(\omega); \mathbf{x}_2(\omega); \dots; \mathbf{x}_r(\omega)) \in \mathcal{E}^{\bar{n}_1} \times \mathcal{E}^{\bar{n}_2} \times \dots \times \mathcal{E}^{\bar{n}_r}$ is the second-stage variable, the matrix $W_0 := (W_{01}, W_{02}, \dots, W_{0r}) \in \mathbb{R}^{m_0 \times n_0}$ (with $W_{0i} \in \mathbb{R}^{m_0 \times n_{0i}}$) and the vectors $\mathbf{h}_0 \in \mathbb{R}^{m_0}$ and $\mathbf{c}_0 := (\mathbf{c}_{01}; \mathbf{c}_{02}; \dots; \mathbf{c}_{0r}) \in \mathbb{R}^{n_0}$ (with $\mathbf{c}_{0i} \in \mathcal{E}^{n_{0i}}$) are deterministic data, and the matrices $B(\omega) := (B_1(\omega), B_2(\omega), \dots, B_r(\omega)) \in \mathbb{R}^{m_1 \times \bar{n}}$ (with $B_i(\omega) \in \mathbb{R}^{m_1 \times \bar{n}_i}$) and $W(\omega) := (W_1(\omega), W_2(\omega), \dots, W_r(\omega)) \in \mathbb{R}^{m_1 \times \bar{n}}$ (with $W_i(\omega) \in \mathbb{R}^{m_1 \times \bar{n}_i}$) and the vectors $\mathbf{h}(\omega) \in \mathbb{R}^{m_1}$ and $\mathbf{c}(\omega) := (\mathbf{c}_1(\omega); \mathbf{c}_2(\omega); \dots; \mathbf{c}_r(\omega)) \in \mathbb{R}^{\bar{n}}$ (with $\mathbf{c}_i(\omega) \in \mathcal{E}^{\bar{n}_i}$) are random data whose realizations depend on an underlying outcome ω in an event space Ω with a known probability function P .

To write the extensive formulation of the SQSOCP problems (1) and (2), we examine problems (1) and (2) when the event space Ω is finite.

Let $\{(B_l, W_l, \mathbf{h}_l, \mathbf{c}_l) : l = 1, \dots, K\}$ be the set of the possible values of the random variables $(B(\omega), W(\omega), \mathbf{h}(\omega), \mathbf{c}(\omega))$ and let $p_l := P(B(\omega), W(\omega), \mathbf{h}(\omega), \mathbf{c}(\omega)) = (B_l, W_l, \mathbf{h}_l, \mathbf{c}_l)$ be the associated probability for $l = 1, 2, \dots, K$. For $i = 1, 2, \dots, r$ and $l = 1, 2, \dots, K$, let n_l and n_{li} be positive integers such that $n_l = \sum_{i=1}^r n_{li}$.

Thus, in analogue to the above notations, the matrices $B_{li} \in \mathbb{R}^{m_1 \times n_{li}}$ and $W_{li} \in \mathbb{R}^{m_1 \times n_{li}}$ and the vectors $\mathbf{c}_{li} \in \mathcal{E}^{n_{li}}$ are realizations of random data such that $B_l := (B_{l1}, B_{l2}, \dots, B_{lr})$, $W_l := (W_{l1}, W_{l2}, \dots, W_{lr})$, and $\mathbf{c}_l := (\mathbf{c}_{l1}; \mathbf{c}_{l2}; \dots; \mathbf{c}_{lr})$. Then, problems (1) and (2) become

$$\begin{aligned} \min \quad & \eta(\mathbf{x}_0) := \mathbf{c}_0^\top \mathbf{x}_0 + \sum_{k=1}^K p_k Q_k(\mathbf{x}_0) \\ \text{s.t.} \quad & W_0 \mathbf{x}_0 = \mathbf{h}_0, \\ & \mathbf{x}_0 \geq \mathbf{0}, \end{aligned} \quad (3)$$

where, for $l = 1, 2, \dots, K$, $Q_l(\mathbf{x}_0)$ is the minimum value of the problem

$$\begin{aligned} \min \quad & \mathbf{c}_l^\top \mathbf{x}_l \\ \text{s.t.} \quad & W_l \mathbf{x}_l = \mathbf{h}_l - B_l \mathbf{x}_0, \\ & \mathbf{x}_l \geq \mathbf{0}. \end{aligned} \quad (4)$$

For convenience, we redefine \mathbf{c}_l as $\mathbf{c}_l := p_l \mathbf{c}_l$ for each $l = 1, 2, \dots, K$ and consequently rewrite the objective function of Problem (3) as $\eta(\mathbf{x}_0) := \mathbf{c}_0^\top \mathbf{x}_0 + \sum_{l=1}^K Q_l(\mathbf{x}_0)$.

We write the SSOCP problems (3) and (4) as an SSOCP problem with finite event space in primal standard form and get

$$\begin{aligned}
 \min \quad & \mathbf{c}_0^\top \mathbf{x}_0 + \mathbf{c}_1^\top \mathbf{x}_1 + \dots + \mathbf{c}_K^\top \mathbf{x}_K \\
 \text{s.t.} \quad & W_0 \mathbf{x}_0 = \mathbf{h}_0, \\
 & B_1 \mathbf{x}_0 + W_1 \mathbf{x}_1 = \mathbf{h}_1, \\
 & \vdots \\
 & B_K \mathbf{x}_0 + \dots + W_K \mathbf{x}_K = \mathbf{h}_K, \\
 & \mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_K \geq \mathbf{0},
 \end{aligned} \tag{5}$$

where the variable $\mathbf{x}_0 := (\mathbf{x}_{01}; \mathbf{x}_{02}; \dots; \mathbf{x}_{0r}) \in \mathcal{E}^{n_{01}} \times \mathcal{E}^{n_{02}} \times \dots \times \mathcal{E}^{n_{0r}}$ is the first-stage decision variable, and $\mathbf{x}_l := (\mathbf{x}_{l1}; \mathbf{x}_{l2}; \dots; \mathbf{x}_{lr}) \in \mathcal{E}^{n_{l1}} \times \mathcal{E}^{n_{l2}} \times \dots \times \mathcal{E}^{n_{lr}}$, for $l = 1, 2, \dots, K$, are the second-stage decision variables.

The dual of (5) is the problem

$$\begin{aligned}
 \max \quad & \mathbf{h}_0^\top \mathbf{y}_0 + \mathbf{h}_1^\top \mathbf{y}_1 + \dots + \mathbf{h}_K^\top \mathbf{y}_K \\
 \text{s.t.} \quad & W_0^\top \mathbf{y}_0 + B_1^\top \mathbf{y}_1 + \dots + B_K^\top \mathbf{y}_K + \mathbf{s}_0 = \mathbf{c}_0, \\
 & W_1^\top \mathbf{y}_1 + \dots + \mathbf{s}_1 = \mathbf{c}_1, \\
 & \vdots \\
 & W_K^\top \mathbf{y}_K + \mathbf{s}_K = \mathbf{c}_K, \\
 & \mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_K \geq \mathbf{0},
 \end{aligned} \tag{6}$$

where $\mathbf{y} := (\mathbf{y}_0; \mathbf{y}_1; \dots; \mathbf{y}_K) \in \mathbb{R}^{m_0 + Km_1}$ is a dual variable and $\mathbf{s}_k := (\mathbf{s}_{k1}; \mathbf{s}_{k2}; \dots; \mathbf{s}_{kr}) \in \mathcal{E}^{n_{k1}} \times \mathcal{E}^{n_{k2}} \times \dots \times \mathcal{E}^{n_{kr}}$ are dual slacks for $k = 0, 1, \dots, K$.

Note that the programs in (5) and (6) are SSOCPs in primal and dual standard forms, respectively, with block diagonal structures [2, Definition 2]. It will become clear in our subsequent development (see, in particular, Theorems 5.1, 5.2) that the size of problems (5) and (6) increases with K , and thus is huge in practice since K is typically very large.

The infeasible interior-point algorithm in [8] can be directly applied to the primal-dual pair (5) and (6), but we emphasize that doing so is not good from a computational point of view because the computation of the search direction is very expensive. In fact, algorithms that exploit the special structures in problems (5) and (6) are very important specially when K is large, which is the case in practice. In the remaining part of this paper, we will see that this computational work can be significantly reduced by exploiting the special structure of problems (5) and (6).

4 Computation of Search Directions

In this section, we present a method for computing the search direction for the infeasible algorithm that exploits the special structures in (5) and (6). This method decomposes the problem into K smaller computations that can be performed in parallel. We start by making some assumptions about the primal-dual pair (5) and (6):

Assumption 4.1 The m_0 rows of the matrix $W_0 = (W_{01}, W_{02}, \dots, W_{0r})$, the m_1 rows of $W_l = (W_{l1}, W_{l2}, \dots, W_{lr})$, and the m_1 rows of $B_l = (B_{l1}, B_{l2}, \dots, B_{lr})$ are linearly independent for each $l = 1, 2, \dots, K$.

Assumption 4.2 Both primal and dual problems are strictly feasible, that is, there exist primal feasible vectors $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_K$ such that $\mathbf{x}_k > \mathbf{0}$ for $k = 0, 1, \dots, K$, and there exist dual-feasible vectors \mathbf{y} and s_0, s_1, \dots, s_K such that $s_k > \mathbf{0}$ for $k = 0, 1, \dots, K$.

Assumption 4.1 is of fundamental importance to ensure nonsingularity. Assumption 4.2 guarantees strong duality for the primal-dual pair (5) and (6).

Given Assumptions 4.1 and 4.2, the optimality conditions for the pair (5) and (6) are as follows:

$$\begin{aligned} W_0\mathbf{x}_0 &= \mathbf{h}_0; & (7a) \\ B_l\mathbf{x}_0 + W_l\mathbf{x}_l &= \mathbf{h}_l, \quad l = 1, 2, \dots, K; & (7b) \\ W_0^\top\mathbf{y}_0 + \sum_{l=1}^K B_l^\top\mathbf{y}_l + \mathbf{s}_0 &= \mathbf{c}_0; & (7c) \\ W_l^\top\mathbf{y}_l + \mathbf{s}_l &= \mathbf{c}_l, \quad l = 1, 2, \dots, K; & (7d) \\ \mathbf{s}_k \circ \mathbf{x}_k &= \mathbf{0}, \quad k = 0, 1, \dots, K; & (7e) \\ \mathbf{x}_k, \mathbf{s}_k &\geq \mathbf{0}, \quad k = 0, 1, \dots, K. & (7f) \end{aligned} \tag{7}$$

Note that using (7a–d) we have that

$$\begin{aligned} \sum_{k=0}^K \mathbf{c}_k^\top \mathbf{x}_k - \sum_{k=0}^K \mathbf{h}_k^\top \mathbf{y}_k &= \left(W_0^\top\mathbf{y}_0 + \sum_{l=1}^K B_l^\top\mathbf{y}_l + \mathbf{s}_0 \right)^\top \mathbf{x}_0 + \sum_{l=1}^K (W_l^\top\mathbf{y}_l + \mathbf{s}_l)^\top \mathbf{x}_l \\ &\quad - (W_0\mathbf{x}_0)^\top \mathbf{y}_0 - \sum_{l=1}^K (B_l\mathbf{x}_0 + W_l\mathbf{x}_l)^\top \mathbf{y}_l \\ &= \sum_{k=0}^K \mathbf{x}_k^\top \mathbf{s}_k \geq 0. \end{aligned}$$

Hence, for given primal feasible vectors $\mathbf{x}_0, \dots, \mathbf{x}_K$ and dual-feasible vectors \mathbf{y} and s_0, \dots, s_K , $\sum_{k=0}^K \mathbf{x}_k^\top \mathbf{s}_k = 0$ is sufficient for optimality, and therefore $\sum_{k=0}^K \mathbf{x}_k^\top \mathbf{s}_k$ is called the duality gap. By using [2, Lemma 15], for $\mathbf{x}_k, \mathbf{s}_k \geq \mathbf{0}, k = 0, 1, \dots, K$, $\sum_{k=0}^K \mathbf{x}_k^\top \mathbf{s}_k = 0$ is equivalent to (7e).

The key idea of the path-following interior-point algorithms is to replace the complementarity equation (7e) by the parameterized equation $\mathbf{x}_k \circ \mathbf{s}_k = \mu \mathbf{e}_k$, for $\mu > 0$ where \mathbf{e}_k be the identity element of $\mathcal{E}^{n_{k1}} \times \mathcal{E}^{n_{k2}} \times \dots \times \mathcal{E}^{n_{kr}}$ for $k = 0, 1, \dots, K$, we then obtain the so-called perturbed optimality conditions. We accordingly consider the system

$$\begin{aligned}
 W_0 \mathbf{x}_0 &= \mathbf{h}_0; \\
 B_l \mathbf{x}_0 + W_l \mathbf{x}_l &= \mathbf{h}_l, \quad l = 1, 2, \dots, K; \\
 W_0^\top \mathbf{y}_0 + \sum_{l=1}^K B_l^\top \mathbf{y}_l + s_0 &= \mathbf{c}_0; \\
 W_l^\top \mathbf{y}_l + s_l &= \mathbf{c}_l, \quad l = 1, 2, \dots, K; \\
 \mathbf{x}_k \circ \mathbf{s}_k &= \mu \mathbf{e}_k, \quad k = 0, 1, \dots, K; \\
 \mathbf{x}_k, \mathbf{s}_k &> \mathbf{0}, \quad k = 0, 1, \dots, K.
 \end{aligned} \tag{8}$$

Under Assumptions 4.1 and 4.2, system (8) has a unique solution denoted by $((\mathbf{x}_0(\mu), \mathbf{y}_0(\mu), s_0(\mu)), \dots, (\mathbf{x}_K(\mu), \mathbf{y}_K(\mu), s_K(\mu)))$ for any $\mu > 0$. We call $\mathbf{x}_0(\mu), \dots, \mathbf{x}_K(\mu)$ and $(y_0(\mu), s_0(\mu)), \dots, (y_K(\mu), s_K(\mu))$ as the μ -centers of primal and dual problems, respectively. The set of all μ -centers is called the central path of primal and dual. If μ approaches 0, then the limit of the central path exists, and since the limit points do satisfy the complementarity condition, they produce an ε -approximate optimal solution of primal and dual problems.

In order to solve system (8), we apply Newton’s method to find an ε -approximate optimal solution of our problem. That is, we compute the Newton search direction $((\Delta \mathbf{x}_0, \Delta s_0, \Delta \mathbf{y}_0), \dots, (\Delta \mathbf{x}_K, \Delta s_K, \Delta \mathbf{y}_K))$ and make a step in this direction to obtain the new iterate $((\mathbf{x}_0(\alpha), \mathbf{y}_0(\alpha), s_0(\alpha)), \dots, (\mathbf{x}_K(\alpha), \mathbf{y}_K(\alpha), s_K(\alpha)))$ belong to the sufficiently small neighborhood of the central path. The corresponding Newton equations are:

$$\begin{aligned}
 W_0 \Delta \mathbf{x}_0 &= \mathbf{r}_{p0}; & \text{where} & & \mathbf{r}_{p0} &:= \mathbf{h}_0 - W_0 \mathbf{x}_0; \\
 B_l \Delta \mathbf{x}_0 + W_l \Delta \mathbf{x}_l &= \mathbf{r}_{pl}; & & & \mathbf{r}_{pl} &:= \mathbf{h}_l - B_l \mathbf{x}_0 - W_l \mathbf{x}_l; \\
 W_0^\top \Delta \mathbf{y}_0 + \sum_{l=1}^K B_l^\top \Delta \mathbf{y}_l + \Delta s_0 &= \mathbf{r}_{d0}; & \mathbf{r}_{d0} &:= \mathbf{c}_0 - W_0^\top \mathbf{y}_0 - \sum_{l=1}^K B_l^\top \mathbf{y}_l - s_0; \\
 W_l^\top \Delta \mathbf{y}_l + \Delta s_l &= \mathbf{r}_{dl}; & \mathbf{r}_{dl} &:= \mathbf{c}_l - W_l^\top \mathbf{y}_l - s_l; \\
 \mathbf{s}_k \circ \Delta \mathbf{x}_k + \Delta \mathbf{s}_k \circ \mathbf{x}_k &= \mathbf{r}_{ck}, & \mathbf{r}_{ck} &:= \sigma \mu \mathbf{e}_k - \mathbf{s}_k \circ \mathbf{x}_k.
 \end{aligned} \tag{9}$$

Here, l varies from 1 to K and k varies from 0 to K , and $\sigma \in]0, 1[$ is a parameter.

Short-, semi-long-, and long-step algorithms are associated with the following centrality measures defined for each pair $(\mathbf{x}_k, \mathbf{s}_k)$:

$$\begin{aligned}
 d_F(\mathbf{x}_k, \mathbf{s}_k) &:= \sqrt{\sum_{i=1}^r \left((\lambda_1 (Q_{\mathbf{x}_{ki}^{1/2} \mathbf{s}_{ki}}) - \mu)^2 + (\lambda_2 (Q_{\mathbf{x}_{ki}^{1/2} \mathbf{s}_{ki}}) - \mu)^2 \right)}, \quad k = 0, 1, \dots, K, \\
 d_2(\mathbf{x}_k, \mathbf{s}_k) &:= \max_{1 \leq i \leq r} \left\{ \left| \lambda_1 (Q_{\mathbf{x}_{ki}^{1/2} \mathbf{s}_{ki}}) - \mu \right|, \left| \lambda_2 (Q_{\mathbf{x}_{ki}^{1/2} \mathbf{s}_{ki}}) - \mu \right| \right\}, \quad k = 0, 1, \dots, K, \\
 d_{-\infty}(\mathbf{x}_k, \mathbf{s}_k) &:= \mu - \min_{1 \leq i \leq r} \left\{ \lambda_1 (Q_{\mathbf{x}_{ki}^{1/2} \mathbf{s}_{ki}}), \lambda_2 (Q_{\mathbf{x}_{ki}^{1/2} \mathbf{s}_{ki}}) \right\}, \quad k = 0, 1, \dots, K.
 \end{aligned}$$

Let $\gamma \in]0, 1[$ be a given constant. We define the following neighborhoods of the central path for our problem:

$$\begin{aligned}
 \mathcal{N}_F(\gamma) &:= \{(\mathbf{x}_0, \mathbf{y}_0, s_0, \dots, \mathbf{x}_K, \mathbf{y}_K, s_K) : d_F(\mathbf{x}_k, \mathbf{s}_k) \leq \gamma \mu, \quad \mathbf{x}_k > \mathbf{0}, \mathbf{s}_k > \mathbf{0}, k = 0, 1, \dots, K\}, \\
 \mathcal{N}_2(\gamma) &:= \{(\mathbf{x}_0, \mathbf{y}_0, s_0, \dots, \mathbf{x}_K, \mathbf{y}_K, s_K) : d_2(\mathbf{x}_k, \mathbf{s}_k) \leq \gamma \mu, \quad \mathbf{x}_k > \mathbf{0}, \mathbf{s}_k > \mathbf{0}, k = 0, 1, \dots, K\}, \\
 \mathcal{N}_{-\infty}(\gamma) &:= \{(\mathbf{x}_0, \mathbf{y}_0, s_0, \dots, \mathbf{x}_K, \mathbf{y}_K, s_K) : d_{-\infty}(\mathbf{x}_k, \mathbf{s}_k) \leq \gamma \mu, \quad \mathbf{x}_k > \mathbf{0}, \mathbf{s}_k > \mathbf{0}, k = 0, 1, \dots, K\}.
 \end{aligned} \tag{10}$$

Here, $\mu = \frac{1}{2(K+1)} \sum_{k=0}^K \mathbf{x}_k^\top \mathbf{s}_k$. The iterates of our algorithm will be restricted to a specific neighborhood, namely the minus-infinity neighborhood, $\mathcal{N}_{-\infty}(\cdot)$, of the central path, which is defined above based on the minus-infinity centrality measure $d_{-\infty}(\mathbf{x}_k, \mathbf{s}_k)$.

Note that for each $k = 0, 1, \dots, K$, the elements \mathbf{x}_k and \mathbf{s}_k of the Newton equations given in (9) may not operator commute. For this reason, system (11) may not have unique solution. In order to overcome this difficulty, we scale the variables used in the Newton system (11) in a way that the scaled variables operator commute and the perturbed complementarity equation is preserved (see Lemma 4.2). In other words, we form different, but fully compatible, perturbed optimality conditions, and consequently lead ourselves to different Newton equations shown in (11) where their scaled elements, $\underline{\mathbf{s}}_k$ and $\overline{\mathbf{x}}_k$, operator commute as will be seen in Lemma 4.1. The way of scaling that we use is well known and proposed originally by Monteiro [13] and Zhang [14] (called *Monteiro-Zhang* family of directions) for semidefinite programming and generalized later by Schmieta and Alizadeh to symmetric programming [15].

Let $\mathbf{p} \in \mathcal{E}$ be such that $\mathbf{p} \succ \mathbf{0}$. With the quadratic form of \mathbf{p} , we define $\overline{\mathbf{w}} := Q_{\mathbf{p}} \mathbf{w}$, and $\underline{\mathbf{w}} := Q_{\mathbf{p}^{-1}} \mathbf{w}$. From [15, Lemma 8], we have that $Q_{\mathbf{p}} Q_{\mathbf{p}^{-1}} = I$, so the operators $\overline{\cdot}$ and $\underline{\cdot}$ are inverse of each other. Furthermore, from [2, Theorem 8], the existence of \mathbf{w}^{-1} indicates the existence of $\overline{\mathbf{w}}^{-1}$ and $\underline{\mathbf{w}}^{-1}$.

The above definitions are also applied to multiple block cases as follows:

$$\overline{\mathbf{w}} := (\overline{\mathbf{w}}_1; \overline{\mathbf{w}}_2; \dots; \overline{\mathbf{w}}_r), \quad \underline{\mathbf{w}} := (\underline{\mathbf{w}}_1; \underline{\mathbf{w}}_2; \dots; \underline{\mathbf{w}}_r) \quad \text{and} \quad \underline{\mathbf{A}} := (\underline{\mathbf{A}}_1, \underline{\mathbf{A}}_2, \dots, \underline{\mathbf{A}}_r),$$

where $\underline{\mathbf{A}}_i := \mathbf{A}_i Q_{\mathbf{p}^{-1}}$ for $i = 1, 2, \dots, r$.

For each $k = 0, 1, \dots, K$, an invertible vector \mathbf{p}_k is chosen as a function of \mathbf{x}_k and \mathbf{s}_k , and for each choice of \mathbf{p}_k we get a different search direction. The following three choices of \mathbf{p}_k are the most common in practice [15]:

- *Choice I (The HRVW/KSH/M direction)* We may choose $\mathbf{p}_k = \mathbf{s}_k^{1/2}$ and obtain $\underline{\mathbf{s}}_k = \mathbf{e}_k$. This choice is analogue of the *XS* direction in semidefinite programming and is known as the HRVW/KSH/M direction (it was introduced by Helmberg, Rendl, Vanderbei, and Wolkowicz [16], and Kojima, Shindoh, and Hara [17] independently, and then rediscovered by Monteiro [13]).
- *Choice II (The dual HRVW/KSH/M direction)* We may choose $\mathbf{p}_k = \mathbf{x}_k^{-1/2}$ and obtain $\overline{\mathbf{x}}_k = \mathbf{e}_k$. This choice of directions arises by switching the roles of *X* and *S*; it is analogue of the *SX* direction in semidefinite programming and is known as the dual HRVW/KSH/M direction.
- *Choice III (The NT direction)* We choose \mathbf{p}_k in such a way that $\overline{\mathbf{x}}_k = \underline{\mathbf{s}}_k$. In this case, we choose

$$\mathbf{p}_k = \left[Q_{\mathbf{x}_k^{1/2}} \left(Q_{\mathbf{x}_k^{1/2}} \mathbf{s}_k \right)^{-1/2} \right]^{-1/2} = \left[Q_{\mathbf{s}_k^{-1/2}} \left(Q_{\mathbf{s}_k^{-1/2}} \mathbf{x}_k \right)^{1/2} \right]^{-1/2}.$$

This choice of directions was introduced by Nesterov and Todd [18] and is known as the NT direction.

The following lemma is due to the discussion after Definition 32 in [2].

Lemma 4.1 *Suppose that $\mathbf{x}_k, \mathbf{s}_k \succ \mathbf{0}, k = 0, 1, \dots, K$. Then, for each of the choices I, II and III of $\mathbf{p}_k, \overline{\mathbf{x}}_k$ and $\underline{\mathbf{s}}_k$ operator commute.*

We have the following proposition, where the first item is based on the fact that $Q_{\mathbf{x}_{ki}^{1/2}} \mathbf{s}_{ki}$ and $Q_{\mathbf{s}_{ki}^{1/2}} \mathbf{x}_{ki}$ have the same spectrum [2, Proposition 2.11] and the second item follows from [8, Proposition 3.2].

Proposition 4.1 *Let $\gamma \in [0, 1]$ be a given constant, then*

1. $d_{-\infty}(\mathbf{x}_k, \mathbf{s}_k)$ is symmetric with respect to \mathbf{x}_k and \mathbf{s}_k , that is $d_{-\infty}(\mathbf{x}_k, \mathbf{s}_k) = d_{-\infty}(\mathbf{s}_k, \mathbf{x}_k), k = 0, \dots, K$, and so is $\mathcal{N}_{-\infty}(\gamma)$.
2. $\mathcal{N}_{-\infty}(\gamma)$ is scaling invariant, that is $(\mathbf{x}_0, \mathbf{y}_0, \mathbf{s}_0, \dots, \mathbf{x}_K, \mathbf{y}_K, \mathbf{s}_K) \in \mathcal{N}_{-\infty}(\gamma)$ iff $(\overline{\mathbf{x}}_0, \mathbf{y}_0, \underline{\mathbf{s}}_0, \dots, \overline{\mathbf{x}}_K, \mathbf{y}_K, \underline{\mathbf{s}}_K) \in \mathcal{N}_{-\infty}(\gamma)$.

We also have the following lemma, where item 1 follows from $Q_p Q_{p^{-1}} = I$ and item 2 follows from [15, Lemma 28].

Lemma 4.2 *Let $\mathbf{x}_k, \mathbf{s}_k \succ \mathbf{0}, k = 0, 1, \dots, K$, and \mathbf{p}_k be invertible. Then, for each $k = 0, 1, \dots, K$, we have $\mathbf{x}_k^T \mathbf{s}_k = \overline{\mathbf{x}}_k^T \underline{\mathbf{s}}_k$. Moreover, $\mathbf{s}_k \circ \mathbf{x}_k = \mu \mathbf{e}_k$ iff $\underline{\mathbf{s}}_k \circ \overline{\mathbf{x}}_k = \mu \mathbf{e}_k$.*

As a result, we have the following lemma.

Lemma 4.3 (Lemma 31, [2]) *The search directions $(\Delta \mathbf{x}_k, \Delta \mathbf{y}_k, \Delta \mathbf{s}_k), k = 0, 1, \dots, K$, solve the Newton equations (9) iff the search directions $(\Delta \overline{\mathbf{x}}_k, \Delta \mathbf{y}_k, \Delta \underline{\mathbf{s}}_k), k = 0, 1, \dots, K$, solves the scaled Newton equations:*

$$\underline{W}_0 \Delta \overline{\mathbf{x}}_0 = \hat{\mathbf{r}}_{p0}; \quad \text{where} \quad \hat{\mathbf{r}}_{p0} := \mathbf{h}_0 - \underline{W}_0 \overline{\mathbf{x}}_0; \tag{11a}$$

$$\frac{\underline{B}_l \Delta \overline{\mathbf{x}}_0 + \underline{W}_l \Delta \overline{\mathbf{x}}_l}{K} = \hat{\mathbf{r}}_{pl}; \quad \hat{\mathbf{r}}_{pl} := \mathbf{h}_l - \frac{\underline{B}_l \overline{\mathbf{x}}_0 - \underline{W}_l \overline{\mathbf{x}}_l}{K}; \tag{11b}$$

$$\underline{W}_0^T \Delta \mathbf{y}_0 + \sum_{l=1}^K \frac{\underline{B}_l^T \Delta \mathbf{y}_l}{K} + \Delta \underline{\mathbf{s}}_0 = \hat{\mathbf{r}}_{d0}; \quad \hat{\mathbf{r}}_{d0} := \mathbf{c}_0 - \underline{W}_0^T \mathbf{y}_0 - \sum_{l=1}^K \frac{\underline{B}_l^T \mathbf{y}_l - \underline{\mathbf{s}}_0}{K}; \tag{11c}$$

$$\frac{\underline{W}_l^T \Delta \mathbf{y}_l + \Delta \underline{\mathbf{s}}_l}{K} = \hat{\mathbf{r}}_{dl}; \quad \hat{\mathbf{r}}_{dl} := \mathbf{c}_l - \frac{\underline{W}_l^T \mathbf{y}_l - \underline{\mathbf{s}}_l}{K}; \tag{11d}$$

$$\frac{\underline{\mathbf{s}}_k \circ \Delta \overline{\mathbf{x}}_k + \Delta \underline{\mathbf{s}}_k \circ \overline{\mathbf{x}}_k}{K} = \hat{\mathbf{r}}_{ck}, \quad \hat{\mathbf{r}}_{ck} := \sigma \mu \mathbf{e}_k - \underline{\mathbf{s}}_k \circ \overline{\mathbf{x}}_k. \tag{11e}$$

Using Assumption 4.2 and using part 5 of Theorem 8 in [2], we conclude that $\overline{\mathbf{x}}_k, \underline{\mathbf{s}}_k \succ \mathbf{0}$ for $k = 0, 1, \dots, K$. Since $\overline{\mathbf{x}}_k$ and $\underline{\mathbf{s}}_k$ operator commute, it follows that $\overline{\mathbf{x}}_k \circ \underline{\mathbf{s}}_k \succ \mathbf{0}$. This under Assumption 4.1 validates the operations described in our upcoming computations. In particular, the matrix M_l defined in (13a) is nonsingular and positive definite for each $l = 1, 2, \dots, K$, and the matrix M_0 defined in (13c) and the matrix $W_0 M_0^{-1} W_0^T$ in (13e) are also nonsingular and positive definite.

Our main result in this section is the following proposition.

Proposition 4.2 *The solution of the scaled Newton equations in (11) at $(\overline{\mathbf{x}}_k, \mathbf{y}_k, \underline{\mathbf{s}}_k)$ is*

$$(\Delta \overline{\mathbf{x}}_k, \Delta \mathbf{y}_k, \Delta \underline{\mathbf{s}}_k) = (\boldsymbol{\beta}_k, \boldsymbol{\alpha}_k, -\overline{\mathbf{x}}_k^{-1} \circ (\underline{\mathbf{s}}_k \circ \boldsymbol{\beta}_k) + \overline{\mathbf{x}}_k^{-1} \circ \hat{\mathbf{r}}_{ck}), k = 0, 1, \dots, K, \tag{12}$$

where

$$M_l := \underline{W}_l \text{Arw}(\underline{s}_l)^{-1} \text{Arw}(\overline{x}_l^{-1})^{-1} \underline{W}_l^T; \tag{13a}$$

$$\mathbf{v}_l := M_l^{-1}(\hat{\mathbf{r}}_{pl} + \underline{W}_l(\underline{s}_l^{-1} \circ (\overline{x}_l \circ \hat{\mathbf{r}}_{dl})) - \underline{W}_l(\underline{s}_l^{-1} \circ \hat{\mathbf{r}}_{cl})); \tag{13b}$$

$$M_0 := \text{Arw}(\overline{x}_0^{-1}) \text{Arw}(\underline{s}_0) + \sum_{l=1}^K \underline{B}_l^T M_l^{-1} \underline{B}_l; \tag{13c}$$

$$\mathbf{v}_0 := M_0^{-1} \left(\sum_{l=1}^K \underline{B}_l^T \mathbf{v}_l - \hat{\mathbf{r}}_{d0} + \overline{x}_0^{-1} \circ \hat{\mathbf{r}}_{c0} \right); \tag{13d}$$

$$\boldsymbol{\alpha}_0 := (\underline{W}_0 M_0^{-1} \underline{W}_0^T)^{-1} (\hat{\mathbf{r}}_{p0} - \underline{W}_0 \mathbf{v}_0); \tag{13e}$$

$$\boldsymbol{\beta}_0 := M_0^{-1} \underline{W}_0^T \boldsymbol{\alpha}_0 + \mathbf{v}_0; \tag{13f}$$

$$\boldsymbol{\alpha}_l := -M_l^{-1} \underline{B}_l^T \boldsymbol{\beta}_0 + \mathbf{v}_l; \tag{13g}$$

$$\boldsymbol{\beta}_l := \underline{s}_l^{-1} \circ (\overline{x}_l \circ (\underline{W}_l^T \boldsymbol{\alpha}_l)) - \underline{s}_l^{-1} \circ (\overline{x}_l \circ \hat{\mathbf{r}}_{dl}) + \underline{s}_l^{-1} \circ \hat{\mathbf{r}}_{cl}. \tag{13h}$$

Here, l varies from 1 to K .

Proof From equation (11e), we obtain

$$\Delta \underline{s}_k = -\overline{x}_k^{-1} \circ (\underline{s}_k \circ \Delta \overline{x}_k) + \overline{x}_k^{-1} \circ \hat{\mathbf{r}}_{ck}, \tag{14}$$

for $k = 0, 1, \dots, K$. Substituting (14) into (11d), we get $-\underline{W}_l^T \Delta \mathbf{y}_l + \overline{x}_l^{-1} \circ (\underline{s}_l \circ \Delta \overline{x}_l) = -\hat{\mathbf{r}}_{dl} + \overline{x}_l^{-1} \circ \hat{\mathbf{r}}_{cl}$. Using the above equality, we can write $\Delta \overline{x}_l$ as

$$\Delta \overline{x}_l = \underline{s}_l^{-1} \circ (\overline{x}_l \circ (\underline{W}_l^T \Delta \mathbf{y}_l)) - \underline{s}_l^{-1} \circ (\overline{x}_l \circ \hat{\mathbf{r}}_{dl}) + \underline{s}_l^{-1} \circ \hat{\mathbf{r}}_{cl} \tag{15}$$

for $l = 1, 2, \dots, K$. Now, we substitute (15) into (11b) to obtain

$$\underline{B}_l \Delta \overline{x}_0 + \underline{W}_l(\underline{s}_l^{-1} \circ (\overline{x}_l \circ (\underline{W}_l^T \Delta \mathbf{y}_l)) - \underline{s}_l^{-1} \circ (\overline{x}_l \circ \hat{\mathbf{r}}_{dl}) + \underline{s}_l^{-1} \circ \hat{\mathbf{r}}_{cl}) = \hat{\mathbf{r}}_{pl}$$

for $l = 1, 2, \dots, K$. It follows that

$$\Delta \mathbf{y}_l = -M_l^{-1} \underline{B}_l \Delta \overline{x}_0 + \mathbf{v}_l, \tag{16}$$

where M_l and \mathbf{v}_l are defined in (13a, b) for $l = 1, 2, \dots, K$. By using (14) and (16) in (11c), we obtain

$$\begin{aligned} \hat{\mathbf{r}}_{d0} &= \underline{W}_0^T \Delta \mathbf{y}_0 + \sum_{l=1}^K \underline{B}_l^T (-M_l^{-1} \underline{B}_l \Delta \overline{x}_0 + \mathbf{v}_l) \\ &\quad - \overline{x}_0^{-1} \circ (\underline{s}_0 \circ \Delta \overline{x}_0) + \overline{x}_0^{-1} \circ \hat{\mathbf{r}}_{c0}. \end{aligned}$$

Thus,

$$\Delta \overline{x}_0 = M_0^{-1} \underline{W}_0^T \Delta \mathbf{y}_0 + M_0^{-1} \left(\sum_{l=1}^K \underline{B}_k^T \mathbf{v}_k - \hat{\mathbf{r}}_{d0} + \overline{x}_0^{-1} \circ \hat{\mathbf{r}}_{c0} \right).$$

So, $\Delta \bar{x}_0$ can be written as

$$\Delta \bar{x}_0 = M_0^{-1} \underline{W}_0^\top \Delta y_0 + v_0, \tag{17}$$

where M_0 and v_0 are defined in (13c, d).

Substituting (17) into (11a) we get $\hat{r}_{p0} = \underline{W}_0(M_0^{-1} \underline{W}_0^\top \Delta y_0 + v_0)$, which then is used to express Δy_0 as

$$\Delta y_0 = (\underline{W}_0 M_0^{-1} \underline{W}_0^\top)^{-1} (\hat{r}_{p0} - \underline{W}_0 v_0). \tag{18}$$

By substituting (17) in (16), Δy_l can be expressed as

$$\Delta y_l = -M_l^{-1} \underline{B}_l (M_0^{-1} \underline{W}_0^\top \Delta y_0 + v_0) + v_l, \tag{19}$$

for $l = 1, 2, \dots, K$. Hence, we can denote $\Delta y_k = \alpha_k$, $\Delta \bar{x}_k = \beta_k$, and $\Delta s_k = -\bar{x}_k^{-1} \circ (s_k \circ \beta_k) + \bar{x}_k^{-1} \circ \hat{r}_{ck}$, where α_k and β_k are defined in (13e–h) for $k = 0, 1, \dots, K$. The proof is complete. \square

5 Infeasible Interior-Point Algorithm for SSOCPs and Its Complexity Estimates

The generic infeasible algorithm for the primal-dual pair (5) and (6) is stated formally in Algorithm 5.1.

Note that we can provide different values for the primal step lengths $\alpha_{p0}^{(j)}, \alpha_{p1}^{(j)}, \dots, \alpha_{pK}^{(j)}$, the dual step lengths $\alpha_{d0}^{(j)}, \alpha_{d1}^{(j)}, \dots, \alpha_{dK}^{(j)}$, and the largest step length $\hat{\alpha}^{(j)}$ as long as we can satisfy all the required conditions. However, if we choose $\alpha_{p0}^{(j)} = \alpha_{p1}^{(j)} = \dots = \alpha_{pK}^{(j)} = \alpha_{d0}^{(j)} = \alpha_{d1}^{(j)} = \dots = \alpha_{dK}^{(j)} = \hat{\alpha}^{(j)}$, we can satisfy all the required conditions and remain inside the neighborhood while making a comparable progress toward feasibility and complementarity.

Note that the Rangarajan’s results in [8] apply to all types of symmetric cones. Because it is well known that the second-order cone is a particular symmetric cone with the standard inner product, we can apply the Rangarajan’s result in [8, Theorem 3.11] for iteration bound to our problem setting and obtain a bound for the number of iterations required to obtain ϵ -approximate optimal solution as we will see below. For this reason, Algorithm 5.1 is well defined and it converges.

Algorithm 5.1 Generic infeasible interior-point algorithm for solving the SSOCP (5) and (6).

Input $1 > \beta > \sigma > 0, \epsilon^* > 0$, and $\gamma \in]0, 1[$

$$(x_0^{(0)}, y_0^{(0)}, s_0^{(0)}, \dots, x_K^{(0)}, y_K^{(0)}, s_K^{(0)}) \in \mathcal{N}_{-\infty}(\gamma),$$

$$\phi_{p0}^{(0)} = \phi_{p1}^{(0)} = \dots = \phi_{pK}^{(0)} = 1, \phi_{d0}^{(0)} = \phi_{d1}^{(0)} = \dots = \phi_{dK}^{(0)} = 1 \text{ and } j = 0$$

while $\sum_{k=0}^K x_k^{(j)\top} s_k^{(j)} \geq \epsilon^* \sum_{k=0}^K x_k^{(0)\top} s_k^{(0)}$ **do**

choose scaling elements $p_0^{(j)}, p_1^{(j)}, \dots, p_K^{(j)}$ form the corresponding scaled

iterate

and compute $(\Delta \bar{\mathbf{x}}_0^{(j)}, \Delta \mathbf{y}_0^{(j)}, \Delta \underline{\mathbf{s}}_0^{(j)}, \dots, \Delta \bar{\mathbf{x}}_K^{(j)}, \Delta \mathbf{y}_K^{(j)}, \Delta \underline{\mathbf{s}}_K^{(j)})$ using (12) and (13)

for $k = 0, 1, \dots, K$ **do**

$$\Delta \mathbf{x}_k^{(j)} := \mathcal{Q}_{\mathbf{p}_k^{(j)-1}} \Delta \bar{\mathbf{x}}_k^{(j)}, \Delta \mathbf{y}_k^{(j)} := \Delta \mathbf{y}_k^{(j)}, \Delta \mathbf{s}_k^{(j)} := \mathcal{Q}_{\mathbf{p}_k^{(j)}} \Delta \underline{\mathbf{s}}_k^{(j)},$$

$$\mathbf{x}_k(\alpha) := \mathbf{x}_k^{(j)} + \alpha \Delta \mathbf{x}_k^{(j)}, \mathbf{y}_k(\alpha) := \mathbf{y}_k^{(j)} + \alpha \Delta \mathbf{y}_k^{(j)}, \mathbf{s}_k(\alpha) := \mathbf{s}_k^{(j)} + \alpha \Delta \mathbf{s}_k^{(j)},$$

end for

compute the largest step length $\hat{\alpha}^{(j)} \in]0, 1]$ such that for all $\alpha \in [0, \hat{\alpha}^{(j)}]$ satisfying that

$$(\mathbf{x}_0(\alpha), \mathbf{y}_0(\alpha), \mathbf{s}_0(\alpha), \dots, \mathbf{x}_K(\alpha), \mathbf{y}_K(\alpha), \mathbf{s}_K(\alpha)) \in \mathcal{N}_{-\infty}(\gamma),$$

$$\sum_{k=0}^K \mathbf{x}_k(\alpha)^\top \mathbf{s}_k(\alpha) \geq \max_{0 \leq k \leq K} (\phi_{pk}^{(j)}, \phi_{dk}^{(j)}) (1 - \alpha) \sum_{k=0}^K \mathbf{x}_k^{(0)\top} \mathbf{s}_k^{(0)} \text{ and}$$

$$\sum_{k=0}^K \mathbf{x}_k(\alpha)^\top \mathbf{s}_k(\alpha) \leq (1 - (1 - \beta)\alpha) \sum_{k=0}^K \mathbf{x}_k^{(j)\top} \mathbf{s}_k^{(j)}.$$

choose primal step lengths $\alpha_{p0}^{(j)}, \alpha_{p1}^{(j)}, \dots, \alpha_{pK}^{(j)} > 0$ and dual step lengths

$$\alpha_{d0}^{(j)}, \dots, \alpha_{dK}^{(j)} > 0$$

such that

$$(\mathbf{x}_0(\alpha_{p0}^{(j)}), \mathbf{y}_0(\alpha_{d0}^{(j)}), \mathbf{s}_0(\alpha_{d0}^{(j)}), \dots, \mathbf{x}_K(\alpha_{pK}^{(j)}), \mathbf{y}_K(\alpha_{dK}^{(j)}), \mathbf{s}_K(\alpha_{dK}^{(j)})) \in \mathcal{N}_{-\infty}(\gamma),$$

$$\sum_{k=0}^K \mathbf{x}_k(\alpha_{pk}^{(j)})^\top \mathbf{s}_k(\alpha_{dk}^{(j)}) \geq \max_{0 \leq k \leq K} (\phi_{pk}^{(j)} (1 - \alpha_{pk}^{(j)}), \phi_{dk}^{(j)})$$

$$(1 - \alpha_{dk}^{(j)}) \sum_{k=0}^K \mathbf{x}_k^{(0)\top} \mathbf{s}_k^{(0)} \text{ and}$$

$$\sum_{k=0}^K \mathbf{x}_k(\alpha_{pk}^{(j)})^\top \mathbf{s}_k(\alpha_{dk}^{(j)}) \leq (1 - (1 - \beta)\hat{\alpha}^{(j)}) \sum_{k=0}^K \mathbf{x}_k^{(j)\top} \mathbf{s}_k^{(j)}.$$

for $k = 0, 1, \dots, K$ **do**

$$\mathbf{x}_k(\alpha_{pk}^{(j)}) := \mathbf{x}_k^{(j)} + \alpha_{pk}^{(j)} \Delta \mathbf{x}_k^{(j)}, \mathbf{y}_k(\alpha_{dk}^{(j)}) := \mathbf{y}_k^{(j)} + \alpha_{dk}^{(j)} \Delta \mathbf{y}_k^{(j)},$$

$$\mathbf{s}_k(\alpha_{dk}^{(j)}) := \mathbf{s}_k^{(j)} + \alpha_{dk}^{(j)} \Delta \mathbf{s}_k^{(j)},$$

$$\mathbf{x}_k^{(j+1)} := \mathbf{x}_k(\alpha_{pk}^{(j)}), \mathbf{y}_k^{(j+1)} := \mathbf{y}_k(\alpha_{dk}^{(j)}), \mathbf{s}_k^{(j+1)} := \mathbf{s}_k(\alpha_{dk}^{(j)}),$$

$$\phi_{pk}^{(j+1)} := \phi_{pk}^{(j)} (1 - \alpha_{pk}^{(j)}), \phi_{dk}^{(j+1)} := \phi_{dk}^{(j)} (1 - \alpha_{dk}^{(j)})$$

end for

$$j := j + 1,$$

end while

An immediate result motivated by [8, Theorem 3.3] is that if $\hat{\alpha}^{(j)} \geq \alpha^*$ for all j and for some $\alpha^* > 0$, then Algorithm 5.1 will terminate with $(\mathbf{x}_0^{(j)}, \mathbf{y}_0^{(j)}, \mathbf{s}_0^{(j)}, \dots, \mathbf{x}_K^{(j)}, \mathbf{y}_K^{(j)}, \mathbf{s}_K^{(j)})$ such that $\|\hat{\mathbf{r}}_{pk}^{(j)}\| \leq \epsilon^* \|\hat{\mathbf{r}}_{pk}^{(0)}\|$ and $\|\hat{\mathbf{r}}_{dk}^{(j)}\| \leq \epsilon^* \|\hat{\mathbf{r}}_{dk}^{(0)}\|$, for $k = 0, 1, \dots, K$,

$$\text{and } \sum_{k=0}^K \mathbf{x}_k^{(j)\top} \mathbf{s}_k^{(j)} \leq \epsilon^* \sum_{k=0}^K \mathbf{x}_k^{(0)\top} \mathbf{s}_k^{(0)} \text{ in } \mathcal{O}\left(\frac{1}{\alpha^*} \ln\left(\frac{1}{\epsilon^*}\right)\right) \text{ iterations.}$$

Rangarajan [8] proved that such a lower bound on α^* exists and established an estimate of the lower bound. Taking into account such bounds in [8] and applying this to our problem setting, we define

$$\alpha^* := \min\left(1, \frac{\gamma\sigma}{((K+1)r+1-\gamma)\omega\sqrt{K}}, \frac{2\sigma}{\omega\sqrt{K}}, \frac{2(\beta-\sigma)}{\omega\sqrt{K}}\right),$$

where $\kappa := \frac{(K+1)r}{1-\gamma}$, $\omega := (\xi + \sqrt{\xi + \zeta})^2$, $\zeta \leq 9 + \frac{1}{1-\gamma}$, and $\xi \leq \sqrt{\kappa}(5 + 4\Psi)$ for some constant $\Psi > 0$. This leads to the following theorem which gives polynomial convergence results for Algorithm 5.1.

Theorem 5.1 *Algorithm 5.1 will terminate in $\mathcal{O}(((K + 1)r)^2 \ln(\frac{1}{\epsilon^*}))$ iterations if the NT direction is used at every iteration, and will terminate in $\mathcal{O}(((K + 1)r)^{2.5} \ln(\frac{1}{\epsilon^*}))$ iterations if the HRVW/KSH/M direction or the dual HRVW/KSH/M direction is used at every iteration.*

The above theorem is a consequence of [8, Theorem 3.11] where the underlying symmetric cone is replaced by the product $\prod_{k=0}^K \mathcal{E}_+^{n_{k1}} \times \mathcal{E}_+^{n_{k2}} \times \dots \times \mathcal{E}_+^{n_{kr}}$.

We estimate the computational complexity of Algorithm 5.1 for computing the search directions with the method described in Sect. 4 applied to problems (5) and (6). The dominant computations occur at (13) in Proposition 4.2. In Table 2, we list the corresponding numbers of arithmetic operations for these computations where $m_1 = m_0$ and $n_0 = n_1 = \dots = n_K$. Note that the most expensive steps in each iteration are the computations of M_0^{-1} in (13c, d). In particular, we analyze the computational work in (13c, d) and also (13a, b) in detail. For any $\mathbf{q} := (\mathbf{q}_1; \mathbf{q}_2; \dots; \mathbf{q}_r)$ with $\mathbf{q}_i \in \mathcal{E}^{n_i}$, the j^{th} block of $B_l^T M_l^{-1} B_l \mathbf{q}$ is written as

$$(B_l^T M_l^{-1} B_l \mathbf{q})_j = B_{lj}^T (M_l^{-1} B_l \mathbf{q}) = B_{lj}^T M_l^{-1} \left(\sum_{i=1}^r B_{li} \mathbf{q}_i \right) = \sum_{i=1}^r B_{lj}^T M_l^{-1} B_{li} \mathbf{q}_i.$$

The number of arithmetic operations required to compute $B_{lj}^T M_l^{-1} B_{li}$ is $\mathcal{O}(m_0^2 n_{0i} n_{0j})$ for all $i, j = 1, 2, \dots, r$. Accordingly, the number of arithmetic operations required to compute $B_l^T M_l^{-1} B_l$ is $\mathcal{O}(m_0^2 n_{0i} n_{0j} r^2) = \mathcal{O}(m_0^2 n_0^2)$ (noting that $\sum_{i=1}^r n_{0i} = n_0$). The inverse of M_0 requires $\mathcal{O}(n_0^3)$ arithmetic operations. Thus, the number of arithmetic operations required for M_0^{-1} is $\mathcal{O}(K m_0^2 n_0^2 + n_0^3)$ as listed in Table 2. Similarly, the number of arithmetic operations required to compute $M_l = \underline{W}_l \text{Arw}(\underline{s}_l)^{-1} \text{Arw}(\underline{x}_l) \underline{W}_l^T$ is $\mathcal{O}(m_0^2 n_0)$, since when we make a column-wise decomposition of \underline{W}_l , we can compute $\text{Arw}(\underline{x}_l) \underline{W}_l^T$ in $\mathcal{O}(m_0 n_0)$ by $\text{Arw}(\underline{x}_l) \underline{W}_l^T = [\underline{x}_l \circ \mathbf{w}_1 \dots \underline{x}_l \circ \mathbf{w}_{m_0}]$.

In addition, the number of arithmetic operations required to compute $\text{Arw}(\underline{s}_l)^{-1} \underline{W}_l^T$ is $\mathcal{O}(m_0 n_0)$. The inverse of M_l requires $\mathcal{O}(m_0^3)$ arithmetic operations. Thus, the number of arithmetic operations required for M_l^{-1} is $\mathcal{O}(m_0^2 n_0 + m_0^3)$ for each $l = 1, 2, \dots, K$. Accordingly, the number of arithmetic operations for (13c, d) in each iteration is dominated by $\mathcal{O}(K(m_0^2 n_0 + n_0^3))$. From the data presented in Table 2, it follows that the total number of arithmetic operations in each iteration is dominated by $\mathcal{O}(K(m_0^2 n_0^2 + n_0^3))$ for any choice of \mathbf{p} stated in Sect. 4. We then have the following theorem.

Theorem 5.2 *Suppose that $m_1 = m_0$ and $n_0 = n_1 = \dots = n_K$. By utilizing Proposition 4.2 for computing the search directions in Algorithm 5.1, we have that the number of arithmetic operations in each iteration of Algorithm 5.1 is $\mathcal{O}(K(m_0^2 n_0^2 + n_0^3))$.*

We investigate the comparison between Algorithm 5.1 and the homogeneous self-dual algorithm presented in [6].

Table 2 Complexity estimates for dominant steps in method of Sect. 4

Equation	Arithmetic operations involved in the equation	Estimate of the number of arithmetic operations
(13a, b)	$\underline{W}_l \text{Arw}(\underline{s}_l)^{-1} \text{Arw}(\underline{x}_l^{-1})^{-1} \underline{W}_l^T; 1 \leq l \leq K$ $M_l^{-1}(\hat{r}_{pl} + \underline{W}_l(\underline{s}_l^{-1} \circ (\underline{x}_l \circ \hat{r}_{dl})) - \underline{W}_l(\underline{s}_l^{-1} \circ \hat{r}_{cl})); 1 \leq l \leq K$	$\mathcal{O}(K m_0^2 n_0 + m_0^3)$
(13c, d)	$\text{Arw}(\underline{x}_0^{-1}) \text{Arw}(\underline{s}_0) + \sum_{l=1}^K \underline{B}_l^T M_l^{-1} \underline{B}_l$ $M_0^{-1} \left(\sum_{l=1}^K \underline{B}_l^T \underline{v}_l - \hat{r}_{d0} + \underline{x}_0^{-1} \circ \hat{r}_{c0} \right)$	$\mathcal{O}(K(m_0^2 n_0 + n_0^3))$
(13e)	$(\underline{W}_0 M_0^{-1} \underline{W}_0^T)^{-1} (\hat{r}_{p0} - \underline{W}_0 \underline{v}_0)$	$\mathcal{O}(m_0^2 n_0 + m_0 n_0^2 + m_0^3)$
(13f)	$M_0^{-1} \underline{W}_0^T \underline{\alpha}_0 + \underline{v}_0$	$\mathcal{O}(m_0 n_0^2)$
(13g)	$-M_l^{-1} \underline{B}_l \underline{\beta}_0 + \underline{v}_l; 1 \leq l \leq K$	$\mathcal{O}(K m_0^2 n_0)$
(13h)	$\underline{s}_l^{-1} \circ (\underline{x}_l \circ (\underline{W}_l^T \underline{\alpha}_l)) - \underline{s}_l^{-1} \circ (\underline{x}_l \circ \hat{r}_{dl}) + \underline{s}_l^{-1} \circ \hat{r}_{cl}; 1 \leq l \leq K$	$\mathcal{O}(K m_0 n_0^3)$

First, we provide an evidence of a clear superiority of Algorithm 5.1 over the homogeneous self-dual algorithm in terms of complexity estimates of the number of arithmetic operations needed per iteration. Theorem 5.2 is the counterpart of Theorem 5.1 in [6]. In [6, Theorem 5.1], the number of arithmetic operations in each iteration of the homogeneous self-dual algorithm is $\mathcal{O}(K(m_0^2 n_0^3 + m_0^3))$. Therefore, the Theorem 5.2 in this paper demonstrates that Algorithm 5.1 for our problem has complexity less than that of the homogeneous algorithm developed for the same problem in [6]. The reason of this superiority of Algorithm 5.1 is probably due to the fact that the homogeneous algorithm is based on embedding the original problem into the larger problem that is always feasible. From this point of comparison stems the importance of the results of this paper.

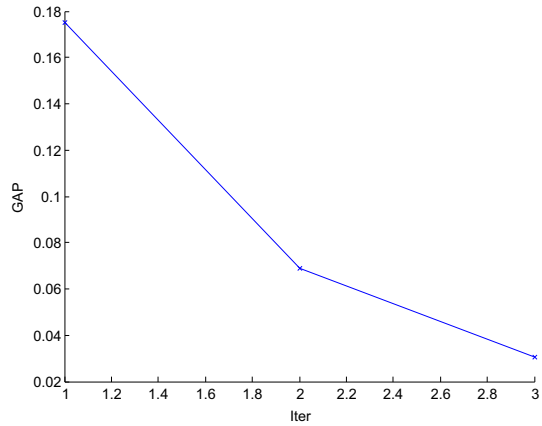
Another point of comparison is to look at the number of iterations needed to obtain ϵ -approximate optimal solution. As far as the homogeneous self-dual algorithm is concerned for comparison purposes, there is no confirmed evidence of a clear preference of Algorithm 5.1 over the homogeneous self-dual algorithm in terms of the number of iterations needed to obtain ϵ -approximate optimal solution because the latter has also a polynomial convergence [19,20]. However, numerically speaking, in the next section we will demonstrate through a simple numerical example (Example 6.1) such a clear preference of Algorithm 5.1 over the homogeneous self-dual algorithm in terms of both: the number of iterations needed to obtain ϵ -approximate optimal solution and the number of arithmetic operations needed per iteration.

It is known that the homogeneous self-dual algorithm begins with a solution that is feasible with respect to both the linear equalities and the second-order cone constraints, but Algorithm 5.1 begins with a solution that may be infeasible with respect to the linear equalities but is feasible with respect to the second-order cone constraints. In fact, on the other side of the comparison mirror, the main advantage of the homogeneous algorithm is to be able to detect infeasibility since the homogeneous self-dual model produces a problem with a strictly feasible starting point and has a relatively simple certificate of infeasibility. In fact, Algorithm 5.1 does not have this advantage, but it is worth mentioning that Algorithm 5.1 can also be modified to detect infeasibility by extending the constraint set and adding extra variables to make the so-called self-dual embedding (see for example [21]), where the original primal and dual problems are embedded in a larger problem with a known strictly feasible starting point on the central path.

6 Implementation of the Algorithm

In this section, we implement the generic infeasible interior-point algorithm to solve some two-stage SSOCP problems. To obtain our numerical experiments, we used MATLAB version R2013b on Windows 7 Ultimate, which were carried out on a PC with Intel(R) Core(TM) i5-4210U CPU at 2.40 GHz and 6 GB of physical memory. Our numerical results are presented and discussed in the following three examples. In these examples, “Iter” denotes the number of iterations, “CPU(s)” denotes the CPU time (in seconds) required to get an approximate optimal solution of the underlying problem, and “Gap” denotes the value of duality gap with a step length α^* .

Fig. 1 The relationship between the duality gap and the number of iterations in Example 6.1



Example 6.1 In this example, for simplicity, we consider the simple case of $K = 5$ scenarios of SSOCP problems where the matrices $W_0, W_l, B_0,$ and B_l and the vectors $c_0, c_l, h_0,$ and h_l , with $p_l = \frac{1}{K} = \frac{1}{5}$ as the associated probability for each $l = 1, 2, \dots, 5$, are as follows:

W_0	W_1	W_2	W_3	W_4	W_5
$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
B_1	B_2	B_3	B_4	B_5	
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
c_0	c_1	c_2	c_3	c_4	c_5
$\begin{pmatrix} 0.2 \\ 0.5 \\ 1.5 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$
h_0	h_1	h_2	h_3	h_4	h_5
$\begin{pmatrix} 0.020 \\ 0.005 \\ 0.005 \\ 0.005 \end{pmatrix}$	$\begin{pmatrix} 0.05 \\ 0.01 \\ 0.01 \end{pmatrix}$	$\begin{pmatrix} 0.06 \\ 0.01 \\ 0.01 \end{pmatrix}$	$\begin{pmatrix} 0.07 \\ 0.01 \\ 0.01 \end{pmatrix}$	$\begin{pmatrix} 0.08 \\ 0.01 \\ 0.01 \end{pmatrix}$	$\begin{pmatrix} 0.09 \\ 0.01 \\ 0.01 \end{pmatrix}$

The parameters of problems (1) and (2) 5.1 are given by $\epsilon = 0.1, \beta = 0.3, \gamma = 0.9$ and $\sigma = 0.25$. With the initial values $x_k^{(0)} = e, s_k^{(0)} = e, y_k^{(0)} = \mathbf{0}$, for $k = 0, 1, \dots, 5$, and with optimal $\alpha = 0.7$, the solution is reached after 3 iterations with $Gap = 0.0147$

and $CPU = 0.0936$. The curve in Fig. 1 illustrates the relationship between the duality gap and the number of iterations for this example. After running the algorithm, we obtain the following results:

x_0	x_1	x_2	x_3	x_4	x_5
$\begin{pmatrix} 0.0279 \\ 0.0050 \\ 0.0050 \\ 0.0050 \end{pmatrix}$	$\begin{pmatrix} 0.0361 \\ 0.0043 \\ 0.0043 \end{pmatrix}$	$\begin{pmatrix} 0.0433 \\ 0.0035 \\ 0.0035 \end{pmatrix}$	$\begin{pmatrix} 0.0500 \\ 0.0028 \\ 0.0028 \end{pmatrix}$	$\begin{pmatrix} 0.0562 \\ 0.0023 \\ 0.0023 \end{pmatrix}$	$\begin{pmatrix} 0.0622 \\ 0.0018 \\ 0.0018 \end{pmatrix}$
s_0	s_1	s_2	s_3	s_4	s_5
$\begin{pmatrix} 0.9070 \\ -0.0970 \\ -0.0970 \\ -0.0970 \end{pmatrix}$	$\begin{pmatrix} 0.7584 \\ -0.0419 \\ -0.0419 \end{pmatrix}$	$\begin{pmatrix} 0.6751 \\ -0.0231 \\ -0.0231 \end{pmatrix}$	$\begin{pmatrix} 0.6165 \\ -0.0132 \\ -0.0132 \end{pmatrix}$	$\begin{pmatrix} 0.5717 \\ -0.0075 \\ -0.0075 \end{pmatrix}$	$\begin{pmatrix} 0.5356 \\ -0.0040 \\ -0.0040 \end{pmatrix}$
y_0	y_1	y_2	y_3	y_4	y_5
$\begin{pmatrix} -7.3966 \\ -11.4556 \\ -10.4637 \\ 1.4251 \end{pmatrix}$	$\begin{pmatrix} 1.2369 \\ 2.0375 \\ 2.0375 \end{pmatrix}$	$\begin{pmatrix} 1.3403 \\ 2.0202 \\ 2.0202 \end{pmatrix}$	$\begin{pmatrix} 1.4190 \\ 2.0107 \\ 2.0107 \end{pmatrix}$	$\begin{pmatrix} 1.4819 \\ 2.0052 \\ 2.0052 \end{pmatrix}$	$\begin{pmatrix} 1.5338 \\ 2.0021 \\ 2.0021 \end{pmatrix}$

It is important to point out that a very similar numerical example has been reported in [20, Subsection 5.1] to test the performance of the algorithm proposed in [6]. The following remark demonstrates that Algorithm 5.1 is more efficient than the homogeneous algorithm proposed in [6].

Remark 6.1 The following observations compare the performance of Algorithm 5.1 and the algorithm proposed in [6].

1. The optimal solution in Example 6.1 obtained by Algorithm 5.1 has been reached after 3 iterations, while the optimal solution in [20, Subsection 5.1] obtained by the homogeneous self-dual algorithm proposed in [6] has been reached after 6 iterations.
2. Solving Example 6.1 using Algorithm 5.1, we have

$$\sum_{k=0}^K x_k^T s_k = 0.1762, \text{ and } \sum_{k=0}^K c_k^T x_k - h_k^T y_k = 0.1755, \text{ and hence}$$

$$\left| \sum_{k=0}^K x_k^T s_k - (c_k^T x_k - h_k^T y_k) \right| = 7.0000e - 04,$$

while solving the same example using the homogeneous self-dual algorithm proposed in [6], we get

$$\sum_{k=0}^K x_k^T s_k = 0.03900232, \text{ and } \sum_{k=0}^K c_k^T x_k - h_k^T y_k = 0.049936, \text{ and hence}$$

$$\left| \sum_{k=0}^K \mathbf{x}_k^\top \mathbf{s}_k - (\mathbf{c}_k^\top \mathbf{x}_k - \mathbf{h}_k^\top \mathbf{y}_k) \right| = 0.01093368.$$

3. According to Theorem 5.2, for any choice of \mathbf{p} , the number of arithmetic operations in each iteration is dominated by $\mathcal{O}(K(m_0^2 n_0^2 + n_0^3))$, which in our case is $K(m_0^2 n_0^2 + n_0^3) = 1600$. Solving this problem using the homogeneous self-dual algorithm, the number of arithmetic operations in each iteration is dominated by $\mathcal{O}(K(m_0^2 n_0^3 + n_0^3))$, which in that case is $K(m_0^2 n_0^3 + n_0^3) = 5440$.

In the following example, we consider K scenarios of our problem where K varies from 5 to 50.

Remark 6.2 Consider K scenarios of SSOCP problems where $K = 10, 15, \dots, 50$ and $p_l = \frac{1}{K}$ as the associated probability for $l = 1, 2, \dots, K$. We let $W_0 = I_{m_0 \times n_0}$, $W_l = I_{m_1 \times n_l}$ and $B_l = [I_{m_1 \times m_1} \ O_{m_1 \times (n_l - m_1)}]$. We also let

$$\mathbf{c}_0 = \begin{pmatrix} 0.2 \\ 0.5 \\ \vdots \\ 0.5 \\ 1.5 \\ 1 \end{pmatrix} \in \mathbb{R}^{n_0}, \quad \mathbf{c}_l = \begin{pmatrix} 2 \\ 2 \\ \vdots \\ 2 \end{pmatrix} \in \mathbb{R}^{n_l}, \quad \mathbf{h}_0 = \begin{pmatrix} 0.020 \\ 0.005 \\ 0.005 \\ \vdots \\ 0.005 \end{pmatrix} \in \mathbb{R}^{m_0} \text{ and}$$

$$\mathbf{h}_l = \begin{pmatrix} 0.04 + \frac{l}{100} \\ 0.01 \\ 0.01 \\ \vdots \\ 0.01 \end{pmatrix} \in \mathbb{R}^{m_l}, \quad l = 1, 2, \dots, K.$$

As in Example 6.1, the parameters of Algorithm 5.1 are given as $\epsilon = 0.1$, $\beta = 0.3$, $\gamma = 0.9$ and $\sigma = 0.25$. With the initial values $\mathbf{x}_k^{(0)} = \mathbf{e}$, $\mathbf{s}_k^{(0)} = \mathbf{e}$, $\mathbf{y}_k^{(0)} = \mathbf{0}$, for $k = 0, 1, \dots, K$, the solutions are reached and the numerical results of Algorithm 5.1 are summarized in Table 3.

Finally, Fig. 2 illustrates the relationship between the duality gap and the number of iterations for some numerical results with fixed $\alpha = 0.0003$ in Table 3. It is interesting to note that the duality gaps decrease in a linear pace as the number of iterations increases.

Remark 6.3 The algorithm is performed with the accuracy $\epsilon = 10^{-1}$ for a number of SSOCPs in which their data are generated randomly. We input (m_0, n_0) and (m_1, n_1) as the dimensions of coefficient matrices W_0 and W_l , respectively, $l = 0, 1, \dots, K$. For suitability, we assume $(m_0, n_0) = (m_1, n_1)$. In each problem, we also consider a finite event space with K scenarios. The algorithm proceeds to generate the deterministic data W_0, \mathbf{h}_0 and \mathbf{c}_0 and random data B_l, W_l, \mathbf{h}_l and \mathbf{c}_l for $l = 1, 2, \dots, K$, with $p_l = \frac{1}{K}$ as the associated probability. We summarize the numerical results of Algorithm 5.1 for this example in Table 4.

Table 3 The numerical results of Example 6.2

Problem size and number of realizations			Numerical results for optimal values for α				Numerical results with fixed $\alpha = 0.0003$			
(l, m) & (s, n)	K		Iter	CPU (s)	Gap	α	Iter	CPU (s)	Gap	
(5, 10)	5		8	0.0624	0.47292	0.007	211	1.8564	0.47291	
(10, 20)	10		9	0.2808	0.48689	0.003	94	2.3868	0.48766	
(15, 30)	15		9	11.4352	0.49119	0.002	60	9.2509	0.49204	
(20, 40)	20		13	5.5848	0.49386	0.001	44	17.0821	0.49411	
(25, 50)	25		9	9.7345	0.49517	0.001	35	29.6714	0.49528	
(30, 60)	30		8	14.4769	0.49605	0.001	28	47.4087	0.49619	
(35, 70)	35		8	25.5374	0.49642	0.0009	24	70.6841	0.49671	
(40, 80)	40		9	47.6115	0.49691	0.0007	21	104.5051	0.49711	
(45, 90)	45		8	66.6904	0.49721	0.0007	19	150.2914	0.49736	
(50, 100)	50		8	100.995	0.49762	0.0006	17	200.6485	0.49762	

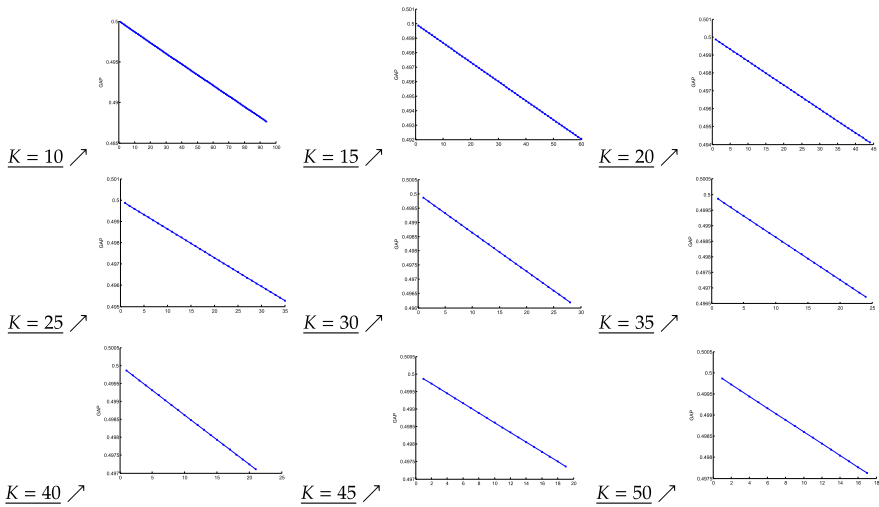


Fig. 2 The duality gap versus the number of iterations [Table 3 for which α is fixed ($\alpha = 0.0003$)]

Table 4 The numerical results of Example 6.3 with optimal values for α

Problem size		K	Iter	CPU (s)	Gap	α
(l, m)	(s, n)					
(5, 10)		5	11	0.14040090	0.113369732	0.1
(10, 20)		10	13	0.43680279	0.101689883	0.1
(15, 30)		15	9	1.66921070	0.177520305	0.1
(20, 40)		20	10	4.33682780	0.177053070	0.09
(25, 50)		25	9	9.21965910	0.218042132	0.08
(30, 60)		30	8	16.0681029	0.236607181	0.08
(35, 70)		35	9	31.1845998	0.243461841	0.07
(40, 80)		40	9	52.3539355	0.269078010	0.06
(45, 90)		45	9	81.1829204	0.258468730	0.06
(50, 100)		50	11	148.278950	0.251315772	0.04

7 Conclusions

Interior-point methods are one of the most effective classes of algorithms for solving SSOCPs. In this paper, we have developed an infeasible interior-point algorithm for solving SSOCPs as an alternative to the homogeneous self-dual algorithm proposed in [6]. We have presented a method for computing the search direction by exploiting the special structure in the SSOCP problem. The polynomial convergence of the proposed algorithm has been demonstrated for three important directions: the NT direction, the HRVW/KSH/M direction, and the dual HRVW/KSH/M direction. More specifically, we have seen that given a Cartesian product of r second-order cones with appropriate dimensions in the first-stage problem, a Cartesian product of r second-order cones

with appropriate dimensions in the second-stage problem, K number of realizations, and a stopping criterion $\epsilon^* > 0$, we need at most $\mathcal{O}(((K+1)r)^2 \ln(\frac{1}{\epsilon^*}))$ iterations to terminate the algorithm if the NT direction is used, and at most $\mathcal{O}(((K+1)r)^{2.5} \ln(\frac{1}{\epsilon^*}))$ iterations to terminate the algorithm if any of the other two directions is used.

We have also estimated the number of arithmetic operations in each iteration of the proposed algorithm. More specifically, by assuming that the dimensions of the underlying second-order cones in each stage have the same sum which equals n_0 and that the number of linear equalities in each stage equals m_0 , and by utilizing our method for computing the search directions, we need at most $\mathcal{O}(K(m_0^2 n_0^2 + n_0^3))$ arithmetic operations in each iteration of the algorithm. Given this result, we have concluded that the algorithm developed in this paper for SSOCs has complexity less than that of the homogeneous self-dual algorithm developed in [6] for the same problem. Finally, we have also presented computational results on three simple problems for the algorithm and have shown that the proposed algorithm is efficient. Our numerical results show also a clear superiority of the infeasible interior-point algorithm developed in this paper over the homogeneous self-dual algorithm developed in [6].

Future work is devoted to develop an infeasible interior-point algorithm for solving stochastic convex optimization problems over nonsymmetric cones (non-self-scaled cones). Despite widespread applications of nonsymmetric programming problems, they have been studied narrowly in comparison with symmetric programming problems such as linear programming, second-order cone programming, and semidefinite programming. This probably is due to the fact that there is a unifying theory based on Euclidean Jordan algebras that connects all symmetric cones. This makes developing an infeasible interior-point algorithm for nonsymmetric optimization problems more challenging. However, this could be achieved for instance by adopting local Hessian norms on the underlying nonsymmetric cone and its dual cone and by using properties based on skew-symmetry of the constraint system of the resulting model.

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

References

1. Alzalg, B.: Stochastic second-order cone programming: application models. *Appl. Math. Model.* **36**, 5122–5134 (2012)
2. Alizadeh, F., Goldfarb, D.: Second-order cone programming. *Math. Program. Ser. B* **95**, 3–51 (2003)
3. Alzalg, B., Ariyawansa, K.A.: Logarithmic barrier decomposition-based interior-point methods for stochastic symmetric programming. *J. Math. Anal. Appl.* **409**, 973–995 (2014)
4. Alzalg, B.: Decomposition-based interior point methods for stochastic quadratic second-order cone programming. *Appl. Math. Comput.* **249**, 1–18 (2014)
5. Alzalg, B.: Volumetric barrier decomposition algorithms for stochastic quadratic second-order cone programming. *Appl. Math. Comput.* **256**, 494–508 (2015)
6. Alzalg, B.: Homogeneous self-dual algorithms for stochastic second-order cone programming. *J. Optim. Theory Appl.* **163**(1), 148–164 (2014)

7. Nesterov, Y.E.: Infeasible start interior point primal-dual methods in nonlinear programming. Working paper, Center for Operations Research and Econometrics, 1348 Louvain-la-Neuve, Belgium (1994)
8. Rangarajan, B.K.: Polynomial convergence of infeasible-interior-point methods over symmetric cones. *SIAM J. Optim.* **16**(4), 1211–1229 (2006)
9. Potra, F., Sheng, R.: A superlinearly convergent primal-dual infeasible interior point algorithm for semidefinite programming. *SIAM J. Optim.* **8**(4), 1007–1028 (1998)
10. Xiaoni, C., Sanyang, L.: An infeasible-interior-point predictor-corrector algorithm for the second-order cone program. *Acta Math. Sci.* **28**(3), 551–559 (2008)
11. Castro, J.: An interior-point approach for primal block-angular problems. *Comput. Optim. Appl.* **36**, 195–219 (2007)
12. Castro, J., Cuesta, J.: Quadratic regularization in an interior-point method for primal block-angular problems. *Math. Program.* **130**, 415–445 (2011)
13. Monteiro, R.D.: Primal-dual path-following algorithms for semidefinite programming. *SIAM J. Optim.* **7**, 663–678 (1997)
14. Zhang, Y.: On extending primal-dual interior-point algorithms from linear programming to semidefinite programming. *SIAM J. Optim.* **8**, 356–386 (1998)
15. Schmieta, S.H., Alizadeh, F.: Extension of primal-dual interior point methods to symmetric cones. *Math. Program. Ser. A* **96**, 409–438 (2003)
16. Helmberg, C., Rendl, F., Vanderbei, R., Wolkowicz, H.: An interior-point method for semidefinite programming. *SIAM J. Optim.* **6**, 342–361 (1996)
17. Kojima, M., Shindoh, S., Hara, S.: Interior-point methods for the monotone semidefinite linear complementary problem in symmetric matrices. *SIAM J. Optim.* **7**, 86–125 (1997)
18. Nesterov, Y.E., Todd, M.J.: Primal-dual interior-point methods for self-scaled cones. *SIAM J. Optim.* **8**, 324–364 (1998)
19. Potra, F., Sheng, R.: On homogeneous interior-point algorithms for semidefinite programming. *Optim. Methods Softw.* **9**, 161–184 (1998)
20. 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)
21. De Klerk, E., Roos, C., Terlaky, T.: Infeasible-start semidefinite programming algorithms via self-dual embeddings. *Fields Inst. Commun.* **18**, 215–236 (1998)