J Optim Theory Appl (2012) 155:1073–1083
DOI 10.1007/s10957-012-0110-x

Homogeneous Self-dual Algorithms for Stochastic Semidefinite Programming

S. Jin · K.A. Ariyawansa · Y. Zhu

Received: 19 July 2011 / Accepted: 15 June 2012 / Published online: 29 June 2012

Abstract Ariyawansa and Zhu have proposed a new class of optimization problems termed stochastic semidefinite programs to handle data uncertainty in applications leading to (deterministic) semidefinite programs. For stochastic semidefinite programs with finite event space, they have also derived a class of volumetric barrier decomposition algorithms, and proved polynomial complexity of certain members of the class. In this paper, we consider homogeneous self-dual algorithms for stochastic semidefinite programs with finite event space. We show how the structure in such problems may be exploited so that the algorithms developed in this paper have complexity similar to those of the decomposition algorithms mentioned above.

Keywords Semidefinite programming · Homogeneous self-dual algorithms · Computational complexity · Stochastic semidefinite programming

1 Introduction

While linear programs have enjoyed wide applicability, they are based on deterministic data. Stochastic Linear Programs [1] (SLPs) were defined to handle uncertainty in

S. Jin
Department of Statistics, Wuhan University of Technology, Wuhan, 430063, China
e-mail: spjin@mail.whut.edu.cn

K.A. Ariyawansa (✉)
Department of Mathematics, Washington State University, Pullman, WA 99164-3113, USA
e-mail: ari@wsu.edu

Y. Zhu
Division of Mathematical and Natural Sciences, Arizona State University, Phoenix, AZ 85069-7100, USA
e-mail: yuntao.zhu@asu.edu

data defining linear programs leading to dramatic enhancement of areas of applicability. Deterministic semidefinite programs (DSDPs) have been the focus of intense research during the past 20 years, especially in the context of interior point methods for optimization [2]. DSDPs generalize linear programs and have a wide variety of applications, especially beyond those covered by linear programs. It then becomes natural to seek a generalization of DSDPs to handle uncertainty in data similar to the way SLPs generalize linear programs. In [3], Ariyawansa and Zhu have defined such a class of problems termed stochastic semidefinite programs (SSDPs).

Ariyawansa and Zhu [4] have presented a class of volumetric barrier decomposition algorithms for SSDPs and proved polynomial complexity of certain members of the class. Their derivation is for the case where the event space of the random variables in the SSDP is finite with K realizations. Another important feature of the algorithms in [4] is that the most expensive part of the algorithm naturally separates into K subproblems, which allows efficient parallel implementations.

Decomposition can be viewed as a way of exploiting the special structure in SSDPs. In this paper, we propose homogeneous self-dual algorithms for SSDPs (with finite event spaces of their random variables) as an alternative to decomposition for handling the special structure. We show that the complexity of these algorithms is similar to those in [4], and that the most expensive part of the algorithms separates into K independent subproblems.

2 Notation and Terminology

The notation and terminology we use in the rest of this paper follow that in [2]. We let $\mathbb{R}^{m\times n}$ and $\mathbb{R}^{n\vee n}$ denote the vector spaces of real $m \times n$ matrices and real symmetric $n \times n$ matrices, respectively. We write $X \succeq 0$ ($X \succ 0$) to mean that X is positive semidefinite (positive definite) and we use $U \succeq V$ or $V \preceq U$ to mean that $U - V \succeq 0$. For $U, V \in \mathbb{R}^{m\times n}$ we write $U \bullet V := \text{trace}\,(U^{\mathsf{T}}V)$ to denote the Frobenius inner product of U and V.

Given $A_i \in \mathbb{R}^{n\vee n}$ for $i = 1, 2, \ldots, m$, we define the operator $\mathcal{A} : \mathbb{R}^{n\vee n} \to \mathbb{R}^m$ by

$$\mathcal{A}X := [A_1 \bullet X, A_2 \bullet X, \ldots, A_m \bullet X]^{\mathsf{T}}, \tag{1}$$

for $X \in \mathbb{R}^{n\vee n}$. The adjoint operator of $\mathcal{A}$ with respect to the standard inner products in $\mathbb{R}^{n\vee n}$ and $\mathbb{R}^m$ is the operator $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{R}^{n\vee n}$ defined by $\mathcal{A}^*y := \sum_{i=1}^m y_i A_i$, for $y \in \mathbb{R}^m$.

The Kronecker product $P \circledast Q$, where $P, Q \in \mathbb{R}^{n\vee n}$ is the operator from $\mathbb{R}^{n\vee n}$ to itself defined by $(P \circledast Q)U := \frac{1}{2}(PUQ^{\mathsf{T}} + QUP^{\mathsf{T}})$, for $U \in \mathbb{R}^{n\vee n}$.

Finally, we recall that the operator $svec : \mathbb{R}^{n\vee n} \to \mathbb{R}^{\bar{n}}$ (see [5]) where $\bar{n} := (1/2)n(n+1)$ is defined by $svec(U) := [U_{11}, \sqrt{2}U_{21}, \ldots, \sqrt{2}U_{n1}, U_{22}, \sqrt{2}U_{32}, \ldots, \sqrt{2}U_{n2}, \ldots, U_{nn}]^{\mathsf{T}}$ for $U \in \mathbb{R}^{n\vee n}$.

3 Homogeneous Self-dual Methods for DSDPs

Given data $A_i \in \mathbb{R}^{n\vee n}$ for $i = 1, 2, \ldots, m$, $b \in \mathbb{R}^m$ and $C \in \mathbb{R}^{n\vee n}$, a DSDP in *primal standard form* is defined as

$$\begin{array}{ll} \text{minimize} & C \bullet X \\ \text{subject to} & \mathcal{A}X = b \\ & X \succeq 0, \end{array} \tag{2}$$

where $X \in \mathbb{R}^{n\vee n}$ is the variable. The dual of (2) is

$$\begin{array}{ll} \text{maximize} & b^{\mathsf{T}} y \\ \text{subject to} & \mathcal{A}^* y + S = C \\ & S \succeq 0, \end{array} \tag{3}$$

where $y \in \mathbb{R}^m$ and $S \in \mathbb{R}^{n\vee n}$ are the variables.

We briefly review the homogeneous interior point algorithm for DSDPs as in [6]. The homogeneous model for (2)–(3) is as follows:

$$\begin{array}{rrrl} \mathcal{A}X & -b\tau & & = 0 \\ -\mathcal{A}^* y & -S & +\tau C & = 0 \\ -C \bullet X & +b^{\mathsf{T}} y & -\kappa & = 0 \\ X \succeq 0, & S \succeq 0, & \tau \geq 0, & \kappa \geq 0. \end{array} \tag{4}$$

It is easy to show from (4) that $X \bullet S + \tau\kappa = 0$. The main step at each iteration of the homogeneous interior point algorithm (shown below in Algorithm 1) is the computation of the search direction $(\Delta X, \Delta y, \Delta S)$ from the symmetrized Newton equations with respect to an invertible matrix P (which is chosen as a function of (X, y, S)) defined by:

$$\begin{array}{rrrrl} \mathcal{A}\Delta X & & -b\Delta\tau & & = \eta r_p \\ & -\mathcal{A}^*\Delta y & -\Delta S & +\Delta\tau C & = \eta R_d \\ -C \bullet \Delta X & +b^{\mathsf{T}}\Delta y & & -\Delta\kappa & = \eta r_g \\ & & \kappa\Delta\tau & +\tau\Delta\kappa & = \gamma\mu - \tau\kappa \\ \mathcal{E}\Delta X & & +\mathcal{F}\Delta S & & = \gamma\mu I - H_P(XS), \end{array} \tag{5}$$

where $r_p := b\tau - \mathcal{A}X$, $R_d := \mathcal{A}^* y + S - \tau C$, $r_g := C \bullet X - b^{\mathsf{T}} y + \kappa$, $\mu := [1/(n+1)](X \bullet S + \tau\kappa)$, η and γ are two parameters, $H_P : \mathbb{R}^{n\vee n} \longrightarrow \mathbb{R}^{n\vee n}$ is the symmetrization operator defined by $H_P(U) := \frac{1}{2}(PUP^{-1} + (P^{-1})^{\mathsf{T}} U^{\mathsf{T}} P^{\mathsf{T}})$, and $\mathcal{E} : \mathbb{R}^{n\vee n} \longrightarrow \mathbb{R}^{n\vee n}$ and $\mathcal{F} : \mathbb{R}^{n\vee n} \longrightarrow \mathbb{R}^{n\vee n}$ are the linear operators defined by $\mathcal{E} := P \circledast (P^{-1})^{\mathsf{T}} S$ and $\mathcal{F} := PX \circledast (P^{-1})^{\mathsf{T}}$, respectively.

Currently, the most common choices of symmetrization for search directions in practice are as follows [7]:

- Helmberg–Rendel–Vanderbei–Wolkowicz/Kojima–Shindoh–Hara/Monteiro (HKM) direction, corresponding to $P := S^{1/2}$. In this case, we have that $\mathcal{E} = \mathcal{I}$ and $\mathcal{F} = X \circledast S^{-1}$.
- Kojima, Shindoh, and Hara (KSH) direction (rediscovered by Monteiro), corresponding to $P := X^{-1/2}$. In this case, we have that $\mathcal{E} = S \circledast X^{-1}$ and $\mathcal{F} = \mathcal{I}$.

– Nesterov–Todd (NT) direction, corresponding to $P := H^{-1/2}$, here H is the unique symmetric positive definite matrix satisfying $HXH = X$, which can be calculated by $H = X^{1/2}(X^{1/2}SX^{1/2})^{-1/2}X^{1/2}$. In this case, we have $\mathcal{E} = \mathcal{I}$ and $\mathcal{F} = H \circledast H$.

Lemma 3.1 *Suppose that $X \succ 0$, $S \succ 0$, and $A_1, A_2, \ldots, A_m$ are linearly independent. Then for each of the above three choices of P, $\mathcal{E}^{-1}$ and $\mathcal{F}^{-1}$ exist, and $\mathcal{E}^{-1}\mathcal{F}$ and $\mathcal{F}^{-1}\mathcal{E}$ are positive definite and self-adjoint.*

Proof See [2]. □

We state the generic homogeneous algorithm as in [6].

Algorithm 1 Generic homogeneous self-dual algorithm for solving (2)–(3)

$(X, y, S, \tau, \kappa) := (I, 0, I, 1, 1)$
while a stopping criterion is not satisfied **do**
  choose η, γ
  compute the solution $(\Delta X, \Delta y, \Delta S, \Delta\tau, \Delta\kappa)$ of the linear system (5)
  compute a step length $\bar{\theta}$ so that
  $X + \bar{\theta}\Delta X \succ 0$,
  $S + \bar{\theta}\Delta S \succ 0$,
  $\tau + \bar{\theta}\Delta\tau > 0$, and
  $\kappa + \bar{\theta}\Delta\kappa > 0$
  $(X, y, S, \tau, \kappa) := (X, y, S, \tau, \kappa) + \bar{\theta}(\Delta X, \Delta y, \Delta S, \Delta\tau, \Delta\kappa)$
end while

4 Homogeneous Self-dual Algorithms for SSDPs

4.1 SSDPs with Finite Event Space

The general definition of a SSDP in primal standard form is stated in [3]. We consider the special case in which the event space is finite with K realizations. Let (deterministic) data $W_0^{(i)} \in \mathbb{R}^{n_0 \vee n_0}$ for $i = 1, 2, \ldots, m_0$, $h_0 \in \mathbb{R}^{m_0}$ and $C_0 \in \mathbb{R}^{n_0 \vee n_0}$; and realizations of random data $B_k^{(i)} \in \mathbb{R}^{n_0 \vee n_0}$, $W_k^{(i)} \in \mathbb{R}^{n_1 \vee n_1}$ for $i = 1, 2, \ldots, m_1$, $h_k \in \mathbb{R}^{m_1}$ and $C_k \in \mathbb{R}^{n_1 \vee n_1}$ for $k = 1, 2, \ldots, K$ be given. Then a SSDP with finite event space in primal standard form is the problem

$$\begin{array}{lrcccl} \text{minimize} & C_0 \bullet X_0 + C_1 \bullet X_1 + \cdots + C_K \bullet X_K & & & & \\ \text{subject to} & \mathcal{W}_0 X_0 & & & & = h_0 \\ & \mathcal{B}_1 X_0 + & \mathcal{W}_1 X_1 & & & = h_1 \\ & \vdots & & \ddots & & \vdots \\ & \mathcal{B}_K X_0 & & & + \mathcal{W}_K X_K & = h_K \\ & X_0, & X_1, & \cdots & X_K & \succeq 0, \end{array} \tag{6}$$

where $X_0 \in \mathbb{R}^{n_0 \vee n_0}$ and $X_k \in \mathbb{R}^{n_1 \vee n_1}$ for $k = 1, 2, \ldots, K$ are the first-stage and second-stage variables, respectively.

Problem (6) is a DSDP in primal standard form with block diagonal structure. Algorithms that exploit this special structure are especially important when K is large as is the case in typical applications.

The dual of (6) is

$$\begin{array}{lrcccccl} \text{maximize} & h_0^{\mathsf{T}} y_0 + & h_1^{\mathsf{T}} y_1 & + \cdots + & h_K^{\mathsf{T}} y_K & & & \\ \text{subject to} & \mathcal{W}_0^* y_0 + & \mathcal{B}_1^* y_1 & + \cdots + & \mathcal{B}_K^* y_K & + & S_0 = C_0 \\ & & \mathcal{W}_1^* y_1 & & & + & S_1 = C_1 \\ & & & \ddots & & & \vdots & \\ & & & & \mathcal{W}_K^* y_K & + & S_K = C_K \\ & S_0, & S_1, & \cdots & S_K & \succeq & 0, \end{array} \tag{7}$$

where $y \in \mathbb{R}^{(m_0 + K m_1)}$ and $S_k \in \mathbb{R}^{(n_0 + K n_1) \vee (n_0 + K n_1)}$ for $k = 1, 2, \ldots, K$ are the variables.

Now Algorithm 1 can be applied to problem (6)–(7). In practice, K is very large. Then problem (6)–(7) is large, and in particular, the computation of the search direction in Algorithm 1 (i.e., the solution of the system (5)) is very expensive. As we shall see in the next section, this computational work can be reduced significantly by exploiting the special structure of problem (6, 7). In addition, the method we describe in next section for the computation of the search direction decomposes into K smaller computations that can be performed in parallel.

4.2 An Efficient Method for Computing Search Directions

We now describe a method for computing the search direction in Algorithm 1 that exploits the special structure in (6), (7). The homogeneous model (4) for problem (6)–(7) is

$$\begin{aligned} & \mathcal{W}_0 X_0 - h_0 \tau = 0 \\ & \mathcal{B}_k X_0 + \mathcal{W}_k X_k - h_k \tau = 0 \\ & -\mathcal{W}_0^* y_0 - \sum_{k=1}^{K} \mathcal{B}_k^* y_k + \tau C_0 - S_0 = 0 \\ & -\mathcal{W}_k^* y_k + \tau C_k - S_k = 0 \\ & \sum_{k=0}^{K} h_k^{\mathsf{T}} y_k - \sum_{k=0}^{K} C_k \bullet X_k - \kappa = 0 \\ & X_k \succeq 0, \quad S_k \succeq 0, \quad k = 0, 1, 2, \ldots, K, \quad \tau \geq 0, \quad \kappa \geq 0 \end{aligned} \tag{8}$$

where $\mathcal{W}_k^*$ and $\mathcal{B}_k^*$ are adjoint operators with appropriate dimensions for $k = 1, 2, \ldots, K$.

The search direction system corresponding to (8) can be derived via (5) as

$$\begin{aligned}
&\mathcal{W}_0\Delta X_0 - h_0\Delta\tau = \eta r_{p_0}\\
&\mathcal{B}_k\Delta X_0 + \mathcal{W}_k\Delta X_k - h_k\Delta\tau = \eta r_{p_k}\\
&-\mathcal{W}_0^*\Delta y_0 - \sum_{k=1}^{K}\mathcal{B}_k^*\Delta y_k + \Delta\tau C_0 - \Delta S_0 = \eta R_{d_0}\\
&-\mathcal{W}_k^*\Delta y_k + \Delta\tau C_k - \Delta S_k = \eta R_{d_k}\\
&\mathcal{E}_0\Delta X_0 + \mathcal{F}_0\Delta S_0 = \gamma\mu I_0 - H_{P_0}(X_0S_0)\\
&\mathcal{E}_k\Delta X_k + \mathcal{F}_k\Delta S_k = \gamma\mu I_k - H_{P_k}(X_kS_k)\\
&\kappa\Delta\tau + \tau\Delta\kappa = \gamma\mu - \tau\kappa\\
&\sum_{k=0}^{K}h_k^{\mathsf{T}}\Delta y_k - \sum_{k=0}^{K}C_k\bullet\Delta X_k - \Delta\kappa = \eta r_g,
\end{aligned} \tag{9}$$

where $r_{p_0} := h_0\tau - \mathcal{W}_0X_0$; $r_{p_k} := h_k\tau - \mathcal{B}_kX_0 - \mathcal{W}_kX_k$; $R_{d_0} := \mathcal{W}_0^*y_0 + \sum_{k=1}^{K}\mathcal{B}_k^*y_k + S_0 - \tau C_0$; $R_{d_k} := \mathcal{W}_k^*y_k + S_k - \tau C_k$; $r_g := \kappa - \sum_{k=0}^{K}h_k^{\mathsf{T}}y_k + \sum_{k=0}^{K}C_k\bullet X_k$; $\mu := (\sum_{k=0}^{K}X_k\bullet S_k + \tau\kappa)/(n_0 + Kn_1 + 1)$; η and γ are two parameters; and $\mathcal{E}_k$, $\mathcal{F}_k$, and H_{P_k} are linear operators which depend only on X_k and S_k; for $1, 2, \ldots, K$.

Now we present the crux of our method for finding the search direction as a solution to (9). By the sixth equation of (9) and the fact that $\mathcal{F}_k^{-1}$ for $k = 1, 2, \ldots, K$ exist, we have $\Delta S_k = -\mathcal{F}_k^{-1}\mathcal{E}_k\Delta X_k + \mathcal{F}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_kS_k))$ for $k = 1, 2, \ldots, K$. Substituting this into the fourth equation of (9), we get $-\mathcal{W}_k^*\Delta y_k + \Delta\tau C_k + \mathcal{F}_k^{-1}\mathcal{E}_k\Delta X_k = \eta R_{d_k} + \mathcal{F}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_kS_k))$ for $k = 1, 2, \ldots, K$. By this equation and by the fact that $(\mathcal{F}_k^{-1}\mathcal{E}_k)^{-1} = \mathcal{E}_k^{-1}\mathcal{F}_k$, because $\mathcal{E}_k^{-1}$ exist, for $k = 1, 2, \ldots, K$, we can express ΔX_k as

$$\begin{aligned}
\Delta X_k = {}&\mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*\Delta y_k - \mathcal{E}_k^{-1}\mathcal{F}_k(\Delta\tau C_k + \eta R_{d_k})\\
&+ \mathcal{E}_k^{-1}\big(\gamma\mu I_k - H_{P_k}(X_kS_k)\big).
\end{aligned} \tag{10}$$

Substituting (10) into the second equation of (9), we have $\mathcal{B}_k\Delta X_0 + \mathcal{W}_k(\mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*\Delta y_k - \mathcal{E}_k^{-1}\mathcal{F}_k(\Delta\tau C_k + \eta R_{d_k}) + \mathcal{E}_k^{-1}(\gamma\mu I_k - H_{P_k}(X_kS_k))) - h_k\Delta\tau = \eta r_{p_k}$ for $k = 1, 2, \ldots, K$. Thus,

$$\Delta y_k = -M_k^{-1}\mathcal{B}_k\Delta X_0 + q_k\Delta\tau + \nu_k, \tag{11}$$

where

$$\begin{aligned}
&M_k := \mathcal{W}_k\mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*, \qquad q_k := M_k^{-1}\big(\mathcal{W}_k\mathcal{E}_k^{-1}\mathcal{F}_kC_k + h_k\big),\\
&\nu_k := M_k^{-1}\big(\eta r_{p_k} - \eta\mathcal{W}_k\mathcal{E}_k^{-1}\mathcal{F}_kR_{d_k} - \mathcal{W}_k\mathcal{E}_k^{-1}\big(\gamma\mu I_k - H_{P_k}(X_kS_k)\big)\big),
\end{aligned} \tag{12}$$

for $k = 1, 2, \ldots, K$. By the fifth equation of (9) and the fact that $\mathcal{F}_0^{-1}$ exists, we get $\Delta S_0 = -\mathcal{F}_0^{-1}\mathcal{E}_0\Delta X_0 + \mathcal{F}_0^{-1}(\gamma\mu I_0 - H_{P_0}(X_0S_0))$. We use this equality and (11) in the third equation of (9) and get $-\mathcal{W}_0^*\Delta y_0 - \sum_{k=1}^{K}\mathcal{B}_k^*(-M_k^{-1}\mathcal{B}_k\Delta X_0 + q_k\Delta\tau +$

$v_k) + \Delta\tau C_0 + \mathcal{F}_0^{-1}\mathcal{E}_0\Delta X_0 - \mathcal{F}_0^{-1}(\gamma\mu I_0 - H_{P_0}(X_0S_0)) = \eta R_{d_0}$. From this relation, we have

$$\begin{aligned}\Delta X_0 &= \mathcal{M}_0^{-1}\left(\mathcal{W}_0^*\Delta y_0 + \left(\sum_{k=1}^{K}\mathcal{B}_k^* q_k - C_0\right)\Delta\tau\right)\\ &\quad + \mathcal{M}_0^{-1}\left(\sum_{k=1}^{K}\mathcal{B}_k^* v_k + \eta R_{d_0} + \mathcal{F}_0^{-1}\big(\gamma\mu I_0 - H_{P_0}(X_0S_0)\big)\right)\\ &= \mathcal{M}_0^{-1}\mathcal{W}_0^*\Delta y_0 - T_0\Delta\tau + U_0,\end{aligned} \tag{13}$$

where

$$\begin{aligned}&\mathcal{M}_0 := \mathcal{F}_0^{-1}\mathcal{E}_0 + \sum_{k=1}^{K}\mathcal{B}_k^* M_k^{-1}\mathcal{B}_k, \qquad T_0 := \mathcal{M}_0^{-1}\left(C_0 - \sum_{k=1}^{K}\mathcal{B}_k^* q_k\right),\\ &U_0 := \mathcal{M}_0^{-1}\left(\sum_{k=1}^{K}\mathcal{B}_k^* v_k + \eta R_{d_0} + \mathcal{F}_0^{-1}\big(\gamma\mu I_0 - H_{P_0}(X_0S_0)\big)\right).\end{aligned} \tag{14}$$

Now, substituting (13) into the first equation of (9), we have $\mathcal{W}_0(\mathcal{M}_0^{-1}\mathcal{W}_0^*\Delta y_0 - T_0\Delta\tau + U_0) - h_0\Delta\tau = \eta r_{p_0}$. Because $\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*$ is nonsingular (a fact to be discussed in Sect. 5), this equation leads us to

$$\Delta y_0 = \big(\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*\big)^{-1}\big((\mathcal{W}_0T_0 + h_0)\Delta\tau + \eta r_{p_0} - \mathcal{W}_0U_0\big) = \alpha_0\Delta\tau + \beta_0, \tag{15}$$

where

$$\begin{aligned}\alpha_0 &:= \big(\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*\big)^{-1}(\mathcal{W}_0T_0 + h_0),\\ \beta_0 &:= \big(\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*\big)^{-1}(\eta r_{p_0} - \mathcal{W}_0U_0).\end{aligned} \tag{16}$$

Now we substitute backward. First, we substitute (15) in (13) to get

$$\Delta X_0 = \mathcal{M}_0^{-1}\mathcal{W}_0^*(\alpha_0\Delta\tau + \beta_0) - T_0\Delta\tau + U_0 = \Psi_0\Delta\tau + \Phi_0, \tag{17}$$

where

$$\Psi_0 := \mathcal{M}_0^{-1}\mathcal{W}_0^*\alpha_0 - T_0, \qquad \Phi_0 := \mathcal{M}_0^{-1}\mathcal{W}_0^*\beta_0 + U_0. \tag{18}$$

Substituting (17) in (11), we obtain

$$\Delta y_k = -\big(\mathcal{W}_k\mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*\big)^{-1}\mathcal{B}_k(\Psi_0\Delta\tau + \Phi_0) + q_k\Delta\tau + v_k = \alpha_k\Delta\tau + \beta_k, \tag{19}$$

where

$$\alpha_k := -\big(\mathcal{W}_k\mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*\big)^{-1}\mathcal{B}_k\Psi_0 + q_k, \qquad \beta_k := \alpha_k - q_k + v_k \tag{20}$$

for $k = 1, 2, \ldots, K$. Furthermore, we substitute (19) in (10) to get

$$\begin{aligned}\Delta X_k &= \mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*(\alpha_k\Delta\tau + \beta_k) - \mathcal{E}_k^{-1}\mathcal{F}_k(\Delta\tau C_k + \eta R_{d_k})\\ &\quad + \mathcal{E}_k^{-1}\big(\gamma\mu I_k - H_{P_k}(X_kS_k)\big) = \Psi_k\Delta\tau + \Phi_k,\end{aligned} \tag{21}$$

where

$$\begin{aligned}\Psi_k &:= \mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*\alpha_k - \mathcal{E}_k^{-1}\mathcal{F}_kC_k\\ \Phi_k &:= \mathcal{E}_k^{-1}\mathcal{F}_k\mathcal{W}_k^*\beta_k - \mathcal{E}_k^{-1}\mathcal{F}_k\eta R_{d_k} + \mathcal{E}_k^{-1}\big(\gamma\mu I_k - H_{P_k}(X_kS_k)\big),\end{aligned} \tag{22}$$

for $k = 1, 2, \ldots, K$. Now, we substitute (15), (17), (19), and (21) in the last equation of (9). By the seventh equation of (9), this yields

$$\sum_{k=0}^{K} h_k^{\mathsf{T}}(\alpha_k \Delta\tau + \beta_k) - \sum_{k=0}^{K} C_k \bullet (\Psi_k \Delta\tau + \Phi_k) - \frac{1}{\tau}(-\kappa\Delta\tau + \gamma\mu - \tau\kappa) = \eta r_g.$$

Finally, $\Delta\tau$ is given by

$$\Delta\tau = \frac{\tau\eta r_g + \tau\sum_{k=0}^{K}(C_k \bullet \Phi_k - h_k^{\mathsf{T}}\beta_k) + (\gamma\mu - \tau\kappa)}{\tau\sum_{k=0}^{K}(h_k^{\mathsf{T}}\alpha_k - C_k \bullet \Psi_k) + \kappa}. \tag{23}$$

We get other directions by (23), and move to the next iteration in Algorithm 1.

5 Complexity Analysis

In this section, we first show that under reasonable conditions the operations described above are valid. Then we estimate the computational complexity of Algorithm 1 with the method described in Sect. 4.2 applied on problem (6)–(7). Finally, we compare that complexity to the complexity of Algorithm 1 applied on problem (6)–(7) treating it as a generic primal-dual DSDP pair.

5.1 Validation of Computations

We assume that $W_0^{(1)}, W_0^{(2)}, \ldots, W_0^{(m_0)}$, and $W_k^{(1)}, W_k^{(2)} \ldots, W_k^{(m_1)}$ for $k = 1, 2, \ldots, K$ are linearly independent. Then M_k for $k = 1, 2, \ldots, K$ in (12) are nonsingular and positive definite by Lemma 3.1.

Now we will show that $\mathcal{M}_0$ in (14) is nonsingular and that $\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*$ is also nonsingular.

Lemma 5.1 *Suppose that $W_0^{(1)}, W_0^{(2)}, \ldots, W_0^{(m_0)}$ and that $W_k^{(1)}, W_k^{(2)} \ldots, W_k^{(m_1)}$ are linearly independent for $k = 1, 2, \ldots, K$. Then $\mathcal{M}_0$ in (14) and $\mathcal{W}_0\mathcal{M}_0^{-1}\mathcal{W}_0^*$ are positive definite.*

Proof From (14), we have $\mathcal{M}_0 = \mathcal{F}_0^{-1}\mathcal{E}_0 + \sum_{k=1}^{K}\mathcal{B}_k^* M_k^{-1}\mathcal{B}_k$. We have that $\mathcal{F}_0^{-1}\mathcal{E}_0$ is positive definite by Lemma 3.1, so it suffices to show that $\mathcal{B}_k^* M_k^{-1}\mathcal{B}_k$ is positive semidefinite for $k = 1, 2, \ldots, K$. Letting $\mathcal{B}_k U = [B_k^{(1)} \bullet U, B_k^{(2)} \bullet U, \ldots, B_k^{(m_1)} \bullet U]^{\mathsf{T}}$, for each $U \in \mathbb{R}^{n_0 \vee n_0}$ and $M_k^{-1} = [\phi_{ij}^{(k)}]_{m_1 \times m_1}$, we have

$$\begin{aligned}\mathcal{B}_k^* M_k^{-1}\mathcal{B}_k U &= \mathcal{B}_k^*\big[\phi_{ij}^{(k)}\big]_{m_1\times m_1}\big[B_k^{(1)} \bullet U, \ldots, B_k^{(m_1)} \bullet U\big]^{\mathsf{T}} \\ &= \sum_i \sum_j \big(\phi_{ij}^{(k)} B_k^{(j)} \bullet U\big) B_k^{(i)},\end{aligned}$$

for $k = 1, 2, \ldots, K$. So

$$\mathcal{B}_k^* M_k^{-1}\mathcal{B}_k U \bullet U = \sum_i \sum_j \big(\phi_{ij}^{(k)} B_k^{(j)} \bullet U\big) B_k^{(i)} \bullet U = (\mathcal{B}_k U)^{\mathsf{T}} M_k^{-1}(\mathcal{B}_k U) \geq 0.$$

Table 1 Complexity estimates for dominant steps in method of Sect. 4.2

Equation number of computation	Estimate of the number of arithmetic operations
(12)	$O(K(m_0^3 + m_0 n_0^3))$
(14)	$O(K m_0^2 n_0^4 + n_0^6)$
(16)	$O(m_0^2 n_0^2 + m_0^3)$
(18)	$O(m_0 n_0^2)$
(20)	$O(m_0^2 n_0^2 + m_0^3)$
(22)	$O(m_0 n_0^2)$

The last inequality is due to the fact that M_k^{-1} is positive definite. As in [2], we can then show that $\mathcal{W}_0 \mathcal{M}_0^{-1} \mathcal{W}_0^*$ is positive definite. □

5.2 Complexity Estimates

Theorem 5.1 *Suppose that $m_1 = m_0$, $n_1 = n_0$ and that $W_k^{(1)}, W_k^{(2)}, \ldots, W_k^{(m_0)}$ are linearly independent for $k = 0, 1, \ldots, K$. By utilizing method described in Sect. 4.2 for computing the search directions in Algorithm 1, we have that the number of arithmetic operations in each iteration of Algorithm 1 is $O(K(m_0^3 + m_0^2 n_0^4) + n_0^6)$.*

Proof The dominant computations of the method in Sect. 4.2 occur at (12), (14), (16), (18), (20), and (22). The corresponding numbers of arithmetic operations for these computations are listed in Table 1. In particular, the computation in (14) will be analyzed in detail. The total number of arithmetic operations is dominated by $O(K(m_0^3 + m_0^2 n_0^4) + n_0^6)$ for choices of P stated in Sect. 3.

We analyze the work of computation in (14). Now $\mathcal{B}_k^* M_k^{-1} \mathcal{B}_k$ is a mapping from $\mathbb{R}^{n_0 \vee n_0}$ to itself. We use the operator $svec : \mathbb{R}^{n_0 \vee n_0} \to \mathbb{R}^{\bar{n}_0}$ where $\bar{n}_0 := \frac{1}{2} n_0 (n_0 + 1)$, to get the matrix of $\mathcal{B}_k^* M_k^{-1} \mathcal{B}_k$. For any $U \in \mathbb{R}^{n_0 \vee n_0}$,

$$
\begin{aligned}
svec\big(\mathcal{B}_k^* M_k^{-1} \mathcal{B}_k U\big) &= svec\Bigg(\sum_i \sum_j \big(\phi_{ij}^{(k)} B_k^{(j)} \bullet U\big) B_k^{(i)}\Bigg) \\
&= \sum_i \sum_j \big(\phi_{ij}^{(k)} B_k^{(j)} \bullet U\big)\, svec\big(B_k^{(i)}\big) \\
&= \sum_i \sum_j \phi_{ij}^{(k)} \big(svec\big(B_k^{(j)}\big)^{\mathsf{T}}\, svec(U)\big)\, svec\big(B_k^{(i)}\big) \\
&= \sum_i \sum_j \phi_{ij}^{(k)} \big(svec\big(B_k^{(i)}\big)\, svec\big(B_k^{(j)}\big)^{\mathsf{T}}\big)\, svec(U).
\end{aligned}
$$

The third equality above is from the relationship $U \bullet V = svec(U)^{\mathsf{T}}\, svec(V)$ for $U, V \in \mathbb{R}^{n_0 \vee n_0}$ (see [5]). So the matrix of $\mathcal{M}_0 = \mathcal{F}_0^{-1}\mathcal{E}_0 + \sum_{k=1}^{K} \mathcal{B}_k^* M_k^{-1} \mathcal{B}_k$ in $\mathbb{R}^{n_0 \vee n_0}$ is given by

$$H_0 + \sum_{k=1}^{K} \sum_{i=1}^{m_0} \sum_{j=1}^{m_0} \phi_{ij}^{(k)} \left(svec\left(B_k^{(i)}\right) svec\left(B_k^{(j)}\right)^{\mathsf{T}} \right), \tag{24}$$

where H_0 is the matrix of $\mathcal{F}_0^{-1}\mathcal{E}_0$, which depends on different choices of symmetrization for search directions.

The number of arithmetic operations needed to compute $svec(B_k^{(i)})\, svec(B_k^{(j)})^{\mathsf{T}}$ is $O(\bar{n}_0^2) = O(n_0^4)$, so the number of arithmetic operations needed for the second term in (24) is $O(Km_0^2 n_0^4)$. The inverse of (24) needs $O(\bar{n}_0^3) = O(n_0^6)$ arithmetic operations. Finally, the arithmetic operations needed for $\mathcal{M}_0^{-1}$ is $O(Km_0^2 n_0^4 + n_0^6)$. Thus, the arithmetic operations for (14) is dominated by $O(K(m_0^3 + m_0^2 n_0^4) + n_0^6)$. □

If we use a generic homogeneous algorithm such as the one in [5], then the number of arithmetic operations required to compute the search directions for (6), (7) is $O(mn^3 + m^2n^2 + m^3)$, where $n := (n_0 + Kn_1)$ and $m := (m_0 + Km_1)$. Setting $m_1 = m_0$ and $n_1 = n_0$ and substituting $n = (1 + K)n_0$ and $m = (1 + K)m_0$ in $O(mn^3 + m^2n^2 + m^3)$, we have that the complexity of such a generic method of computing the search directions is $O(K^4(m_0 n_0^3 + m_0^2 n_0^2) + K^3 m_0^3)$. This is much larger than the complexity $O(K(m_0^3 + m_0^2 n_0^4) + n_0^6)$ obtained for the method in Sect. 4.2 when $K \gg m_0$ and $K \gg n_0$.

If problem (6)–(7) has a solution, then the KSH method is globally convergent [6]. The algorithm finds an optimal solution or determines that the primal-dual pair has no solution of norm less than a given number in at most $O(n^{1/2}L)$ iterations. Here, n is the size of the problem and L is the logarithm of the ratio of the initial error and the tolerance. So, by using the method in Sect. 4.2 for computing the search direction with KSH symmetrization, the complexity of Algorithm 1 in terms of the total number of arithmetic operations is $O(K^{1.5}(m_0^3 n_0^{0.5} + m_0^2 n_0^{4.5}) + K^{0.5} n_0^{6.5})$ to find a solution or ascertain that the (8) has no solution. In comparison, the short- and long-step decomposition algorithms of Ariyawansa and Zhu [4] have complexities of $O(K^{1.5})$ and $O(K^2)$, respectively, in terms of the total number of arithmetic operations.

We note that the efficient computation of the Schur computation matrix M_k in (12) is crucial as this is the most expensive step in each iteration where usually 80 % of the total CPU time is spent if the algorithm in [7] is used. However, in each iteration, each M_k can be computed independently, and so distributed processing may be used to achieve substantial reductions in computation time.

6 Concluding Remarks

In this paper, we have proposed homogeneous self-dual algorithms for Stochastic Semidefinite Programming whose complexity is comparable to those of algorithms in [4]. The efficient method for calculating the search directions we developed can take advantage of parallel and distributed processing.

Acknowledgements The work of S.J. was performed while he was visiting Washington State University. Research supported in part by the Chinese National Foundation under Grants No. 51139005 and 51179147. K.A.A. research supported in part by the US Army Research Office under Grant DAAD 19-00-1-0465 and by Award W11NF-08-1-0530. Y.Z. research supported in part by ASU West MGIA Grant 2007.

References

1. Kall, P., Wallace, S.: Stochastic Programming. Wiley, New York (1994)
2. Todd, M.J.: Semidefinite optimization. Acta Numer. **10**, 515–560 (2001)
3. Ariyawansa, K.A., Zhu, Y.: Stochastic semidefinite programming: a new paradigm for stochastic optimization. 4OR **4**(3), 239–253 (2006). An earlier version of this paper appeared as Technical Report 2004-10 of the Department of Mathematics, Washington State University, Pullman, WA 99164-3113, in October 2004
4. Ariyawansa, K.A., Zhu, Y.: A class of polynomial volumetric barrier decomposition algorithms for stochastic semidefinite programming. Math. Comput. **80**, 1639–1661 (2011)
5. Todd, M.J., Toh, K.C., Tütüncü, R.H.: On the Nesterov-Todd direction in semidefinite programming. SIAM J. Optim. **8**, 769–796 (1998)
6. Potra, F., Sheng, R.: On homogeneous interior-point algorithms for semidefinite programming. Optim. Methods Softw. **9**, 161–184 (1998)
7. Toh, K.C., Todd, M.J., Tütüncü, R.H.: SDPT3—a MATLAB software package for semidefinite programming. Optim. Methods Softw. **11/12**, 545–581 (1999)