J. Appl. Math. Comput. (2016) 51:569–591
DOI 10.1007/s12190-015-0920-8

ORIGINAL RESEARCH

# Douglas–Rachford splitting method for semidefinite programming

**Yunda Dong**[1]

Received: 8 February 2015 / Published online: 13 August 2015

**Abstract** In this paper, we study the Douglas–Rachford (DR) splitting method when applied to the standard semidefinite programming, and we introduce a new variant of this method and under weak assumptions prove its global convergence. At each iteration, it needs to solve two subproblems. First, project once onto the set of all symmetric positive semidefinite matrices. Then, solve one system of linear equations whose co-efficient matrix is symmetric positive definite. This is in sharp contrast to symmetric positive semi-definiteness of the corresponding co-efficient matrix in the boundary point method recently proposed by Povh et al. More importantly, we analyze the iteration complexity of the DR splitting method and derive the currently best result. Based on rigorous analysis in theory, we suggest an implementable version. Rudimentary numerical experiments confirm its efficiency and robustness on certain test problems with more than one million equality constraints.

**Keywords** Semidefinite programming · Douglas–Rachford splitting method · Convergence · Iteration complexity

**Mathematics Subject Classification** 90C06 · 90C22 · 90C30 · 90C35

## 1 Introduction

Let $S^n$ be the set of all $n \times n$ symmetric matrices, and $S^n_+$ be the set of all $n \times n$ symmetric positive semidefinite matrices. In this paper, we are mainly concerned with the semidefinite programming (SDP for short) in the primal form

✉ Yunda Dong
ydong@zzu.edu.cn

1 School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, People's Republic of China

$$\text{Minimize } \langle C,\ X\rangle \quad \text{subject to} \quad \mathcal{A}(X) = b, \quad X \succeq 0, \tag{1}$$

where $X \succeq 0$ means that $X$ belongs to $S^n_+$, and $\mathcal{A}(\cdot)\colon S^n \to R^m$ is a linear map, defined by

$$\mathcal{A}(X) = (\langle A_1,\ X\rangle, \ldots, \langle A_m,\ X\rangle)^T ,$$

with $C,\ A_1, \ldots, A_m$ being given $n \times n$ matrices, $b = (b_1, \ldots, b_m)^T \in R^m$. The dual form of this SDP is as follows:

$$\text{Maximize } b^T y \quad \text{subject to} \quad Z = C - \mathcal{A}^*(y), \quad Z \succeq 0, \tag{2}$$

where $\mathcal{A}^*(y) := \sum_{i=1}^m y_i A_i$, and $y = (y_1, \ldots, y_m)^T$ is called Lagrange multiplier variable. The SDP is an extension of linear programming and has a wide variety of applications in the fields such as optimization and control [1–4].

Our focus of this paper is on recently proposed methods for solving some large-scale SDPs [5–9]. In [5], Burer and Vandenbussche used the primal augmented Lagrangian function

$$\tilde{\mathcal{L}}_\mu(X,\ y) := \langle C,\ X\rangle - y^T(\mathcal{A}(X) - b) + \frac{1}{2\mu}\|\mathcal{A}(X) - b\|^2,$$

where $\mu > 0$ is the penalty parameter, to suggest an early method for certain classes of SDPs: for known $y^k$, first find the solution of minimizing $\tilde{\mathcal{L}}_\mu(X,\ y^k),\ X \succeq 0$ and take it to be $X^k$. Then the new multiplier is generated by

$$y^{k+1} = y^k - \mu^{-1}\left(\mathcal{A}\left(X^k\right) - b\right). \tag{3}$$

Meanwhile, Malick [10] investigated a similar approach to solve semidefinite least squares problems. From a viewpoint dual to [5,10], Povh et al. [6] applied the following dual augmented Lagrangian function

$$\mathcal{L}_\mu(X,\ y,\ Z) := -b^T y + \langle X,\ Z - C + \mathcal{A}^*(y)\rangle + \frac{1}{2\mu}\|Z - C + \mathcal{A}^*(y)\|_F^2,$$

where $\mu > 0$ is the penalty parameter, to proposing the boundary point method: for known $X^k,\ Z^k$, first solve the subproblem of minimizing $\mathcal{L}_\mu(X^k,\ y,\ Z^k),\ y \in R^m$ for $y^{k+1}$, i.e.,

$$AA^T y^{k+1} = \mathcal{A}\left(C - Z^k\right) - \mu\left(\mathcal{A}\left(X^k\right) - b\right), \tag{4}$$

where $AA^T$ is an $m \times m$ matrix whose $(i,\ j)$th entry is $\langle A_i,\ A_j\rangle$. Next, solve the subproblem of minimizing $\mathcal{L}_\mu(X^k,\ y^{k+1},\ Z),\ Z \succeq 0$ to get

$$Z^{k+1} = \left[C - \mathcal{A}^*\left(y^{k+1}\right) - \mu X^k\right]_+, \tag{5}$$

where $[\cdot]_+$ denotes the usual projection onto the cone $S^n_+$. Finally, $X^{k+1}$ is given by (the equivalence can be checked by Lemma 1)

$$\begin{aligned} X^{k+1} &= \left[X^k - \mu^{-1}\left(C - \mathcal{A}^*\left(y^{k+1}\right)\right)\right]_+ \\ &= X^k + \mu^{-1}\left(Z^{k+1} - C + \mathcal{A}^*\left(y^{k+1}\right)\right). \end{aligned} \tag{6}$$

A nice property of this method is that the generated sequence $\{(X^k,\ Z^k)\}$ satisfies the complementarity condition. Shortly after, Malick et al. [7] well studied the boundary point method and related ones and soon gave a Matlab code that they called mprw2, which is downloadable from Rendl's web page: www.math.uni-klu.ac.at/or/Software/. Meanwhile, Wen et al. [8] noticed that computational results can be improved once $\mu^{-1}$ in (6) is replaced by $\rho\mu^{-1}$ in some cases, where $\rho \in [1,\ 1.6]$. Furthermore, they developed an algorithm SPDAD for extended SDPs.

From a different standpoint, Zhao et al. [9] made use of a variant of the above dual augmented Lagrangian function, namely

$$\mathcal{L}_\mu(X,\ y) = \min\left\{\mathcal{L}_\mu(X,\ y,\ Z),\ Z \succeq 0\right\},$$

and suggested another method: for given $X^k,\ y^k$, first find an approximate solution of minimizing convex differentiable function $\mathcal{L}_\mu(X^k,\ y),\ \ y \in R^m$ and take it to be $y^{k+1}$. Then, $X^{k+1}$ is given by

$$X^{k+1} = \left[X^k - \mu^{-1}\left(C - \mathcal{A}^*\left(y^{k+1}\right)\right)\right]_+.$$

This method can be viewed as a refined instantiation of the methods [7], and its implementable version is called SDPNAL.

Very recently, Monteiro et al. [11] suggested the following (unscaled) adaptive block-decomposition method: for known $X^k$ and $y^k$, choose $\mu > 0,\ 0 < \sigma < 1$, compute

$$\bar{y}^k = y^k - \mu\left(\mathcal{A}\left(X^k\right) - b\right), \quad \bar{X}^k = \left[X^k - \mu\left(C - \mathcal{A}^*\left(\bar{y}^k\right)\right)\right]_+.$$

Then choose $\lambda_k$ to be the positive number $\lambda$ such that

$$\begin{aligned} &\left\|\left(\mu^{-1}\lambda - 1\right)\left(\bar{X}^k - X^k\right)\right\|_F^2 + \left\|\lambda\left(\mathcal{A}\left(\bar{X}^k\right) - b\right) + \bar{y}^k - y^k\right\|^2 \\ &\quad \le \sigma\left(\left\|\bar{X}^k - X^k\right\|_F^2 + \left\|\bar{y}^k - y^k\right\|^2\right). \end{aligned}$$

Finally, set

$$X^{k+1} = X^k + \mu^{-1}\lambda_k\left(\bar{X}^k - X^k\right), \quad y^{k+1} = y^k - \lambda_k\left(\mathcal{A}\left(\bar{X}^k\right) - b\right),$$

to get the new primal variable and the new Lagrange multiplier.

Today, the algorithms such as mprw2, SDPNAL, SDPAD and [11, Algorithm 1] have received much attention within the SDP community. But in theory only the latter's iteration complexity is known [11, Theorem 3.3]: there exists some $i \le k$ such that

$$f\left(X^i, y^i, 1\right) \le O(1/k),$$

where $f$ is the merit function defined by (22). Although the index $i$ on the left-hand side is not larger than $k$, this corresponding iteration complexity appears to be still $O(1/\epsilon)$ in general, where $\epsilon$ is the accuracy measure, whenever the stopping criterion $f \le \epsilon$ is used.

Then an important question is to ask: whether or not there exists an algorithm used for solving large-scale SDPs whose iteration complexity is better than $O(1/\epsilon)$?

In this paper, we give a positive answer of the question above. Specifically speaking, we will discuss the following iterative procedure for solving the standard SDP: at each iteration, for known $X^k$ and $y^k$, choose $\mu_k \ge 0$, compute

$$\bar{y}^k = y^k - \mu_k\left(\mathcal{A}\left(X^k\right) - b\right), \quad \bar{X}^k = \left[X^k - \alpha\left(C - \mathcal{A}^*\left(\bar{y}^k\right)\right)\right]_+, \tag{7}$$

where $\alpha$ is prescribed positive number. Next, choose $\gamma_k > 0$, solve one system of linear equations

$$\left(\alpha^{-2}I + AA^T\right)\Delta y^k = -\gamma_k\alpha^{-1}\left(A\bar{x}^k - b\right),$$

for $\Delta y^k$. Then make use of (32) and (33)below to get the new primal variable and the new Lagrange multiplier. Importantly, we can prove that, if $\mu_k \equiv 0$ and the stopping criterion $f \le \epsilon$ is used, then it has $o(1/\epsilon)$ iteration complexity, which is considerably better than $O(1/\epsilon)$.

Note that the case of $\mu_k \equiv 0$ corresponds to the Douglas–Rachford splitting method (DR method for short) of Lions and Mercier [12] (see also [13–15]) when applied to solving the SDPs. In this sense, our proposed iterative procedure can be viewed as a new variant of the DR method.

The main contributions of this paper are twofold. Firstly, we confirm that, as an algorithm used for solving large-scale SDPs, the DR method has $o(1/\epsilon)$ iteration complexity. This appears to be the best iteration complexity for all splitting methods and alternating directions methods used for solving large-scale SDPs. In contrast, the papers [5–9] either did not deal with the issue of iteration complexity or gave the result which is not attainable at the level of $o(1/\epsilon)$. Secondly, we develop our proposed method into an implementable one used for solving some large-scale SDPs. In particular, we justify our way of adaptively choosing $\alpha$ in (7) by introducing Lemma 4. This relies on a new idea, which occurred to the author in August of 2012.

The rest of this paper is organized as follows. In Sect. 2, we specify the notation used and review some useful results concerning the usual projection onto the cone $S^n_+$. Moreover, we introduce Lemmas 4 and 5 for the first time in the setting of the SDPs.

In Sect. 3, we state in details our proposed method. In Sect. 4, we prove global convergence of our proposed method, and the only assumption is that there exist primal and dual solutions with zero primal–dual gap. Furthermore, we confirm $o(1/\epsilon)$ iteration complexity of our proposed method when it reduces to the DR method as a limiting case. In Sect. 5, we explain the reason why our proposed method can be viewed as a variant of the DR method. In addition, we also show the equivalence of [16, Algorithm 1] to the DR method. In Sect. 6, we give an implementable version SDPsplit of our proposed method, and we cope with some most important issues that arise in practical implementations. In Sect. 7, we implement the SDPsplit to solve three test problems with more than one million equality constraints and report promising numerical results (no seemingly right but actually unwarranted arguments are involved). In Sect. 8, we close this paper by some concluding remarks.

## 2 Preliminaries

In this section, we first give the notation used throughout this paper. Then, we give some useful results concerning the usual projection onto the cone $S^n_+$.

For any given $p, q \in R^m$, $p^T q$ denotes the usual inner product and $\|q\| = \sqrt{q^T q}$ denotes its induced norm, where $T$ is the transpose. For any given $n \times n$ matrices, $\langle P, Q\rangle$ denotes the inner product between two matrices and is equal to $\sum_{i,j=1}^n p_{ij}q_{ij}$. Its induced Frobenius norm is defined by $\|Q\|_F = \sqrt{\langle Q, Q\rangle}$. We use a capital letter to write a matrix $Q = (q_{ij})_{m\times n}$, and use its small form $q$ to express the associated $mn$-dimensional column vector generated in the order $q = [q_{11}, \ldots, q_{n1}, ldots, q_{1n}, \ldots, q_{nn}]^T$. Let $A = (a_1, \ldots, a_m)^T$, where each $a_i \in R^{n^2}$ is generated by $A_i \in S^n$ that corresponds to the equality constraints of the SDP, and $\|AA^T\| = \max\{x^T AA^T x : x \in R^m, \|x\| = 1\}$. For any given $Q \in S^n$, $Q_+$ denotes the usual projection onto $S^n_+$ that uniquely minimizes $\|Q - X\|_F$, $X \succeq 0$. From now on, keep in mind that $\mathcal{A}(X)$ and $Ax$ have the same meaning, so do $\mathcal{A}\mathcal{A}^*(y)$ and $AA^T y$ for any given $y \in R^m$.

Below we list some fundamental properties of the usual projection onto the cone $S^n_+$, and the first one is the following

**Lemma 1** *If $Q = P^T \Lambda P$ is the spectral decomposition of $Q \in S^n$, where $P$ satisfies $P^T P = PP^T = I$ and $I$ stands for the unit matrix. Then $Q_+ = P^T \Lambda_+ P$, where $\Lambda_+$ is derived from $\Lambda$ by replacing all negative diagonal entries with zeros. Furthermore, $Q_+ = Q + [-Q]_+$.*

We refer the reader to [17–19] for further discussions. By using Lemma 1 and a known result that $Q \succeq 0$ if and only if $\langle Q, P\rangle \geq 0$ whenever $P \succeq 0$ (cf. [20, Corollary 1.2.7]), we can arrive at some fundamental properties of the usual projection onto the cone $S^n_+$.

**Lemma 2** *For all $V \in S^n$, the following statements hold*

(a) $\langle V_+, V - V_+\rangle = 0$, $\langle U - V_+, V - V_+\rangle \leq 0$, for all $U \in S^n_+$;
(b) $\|W_+ - V_+\|_F \leq \|W - V\|_F$, for all $W \in S^n$.

**Lemma 3** *If there exist $X^*$ and $y^*$ such that*

$$\mathcal{A}(X) = b, \quad X = [X - (C - \mathcal{A}^*(y))]_+, \tag{8}$$

*then $X^*$ solves the SDP.*

*Proof* For any given feasible $X'$, we can apply Lemma 2 with $U := X'$ and $V := X^* - (C - \mathcal{A}^*(y^*))$ to achieve the following inequality

$$\left\langle X' - [X^* - (C - \mathcal{A}^*(y^*))]_+,\ X^* - (C - \mathcal{A}^*(y^*)) - [X^* - (C - \mathcal{A}^*(y^*))]_+ \right\rangle \le 0.$$

Thus, by virtue of the last relation in (8), replacing $[X^* - (C - \mathcal{A}^*(y^*))]_+$ with $X^*$ in the inequality above yields

$$\langle X' - X^*,\ X^* - (C - \mathcal{A}^*(y^*)) - X^* \rangle \le 0 \Rightarrow \langle X' - X^*,\ C - \mathcal{A}^*(y^*) \rangle \ge 0. \tag{9}$$

Since $X'$ is feasible (thus $\langle A_i,\ X' \rangle - b_i = 0$ for all $i = 1, \ldots, n$), it follows from (9) that

$$\begin{aligned} \langle C,\ X' - X^* \rangle \ge \langle X' - X^*,\ \mathcal{A}^*(y^*) \rangle &= \sum_{i=1}^m y_i^* \left\langle A_i,\ X' - X^* \right\rangle \\ &= \sum_{i=1}^m y_i^* \left( \left\langle A_i,\ X' \right\rangle - b_i \right) = 0. \end{aligned}$$

The proof is complete. □

As a direct consequence, we give an equivalent form of the Lemma 3 above.

**Lemma 3′** *If there exist $X^*$ and $y^*$ such that the following*

$$\mathcal{A}(X) = b, \quad X = [X - \alpha(C - \mathcal{A}^*(y))]_+, \tag{10}$$

*holds for any given positive number $\alpha$, then $X^*$ solves the SDP.*

**Lemma 4** *Let $X,\ W \in S^n$. Then for any given two positive numbers $\alpha \le \alpha'$*

$$\begin{aligned} &\|X - [X - \alpha W]_+\|_F \le \|X - [X - \alpha' W]_+\|_F, \\ &\frac{\|X - [X - \alpha W]_+\|_F}{\alpha} \ge \frac{\|X - [X - \alpha' W]_+\|_F}{\alpha'}. \end{aligned}$$

Notice that these two statements are direct extensions of projection properties [21,22] to the SDPs. And we can follow the proof techniques in [23] to get it; see [24, Lemma 1] for an extension and a simple proof.

**Lemma 5** *In the setting of SDP, for all $X, \tilde{X}$ in $S^n$ and for all $y, \tilde{y}$ in $R^m$, if we set*

$$E := X - [X - \alpha(C - \mathcal{A}^*(y))]_+, \quad \tilde{E} := \tilde{X} - [\tilde{X} - \alpha(C - \mathcal{A}^*(\tilde{y}))]_+,$$

*then the following inequality holds*

$$0 \geq \langle \tilde{E} - E,\ X - \tilde{X} + \alpha\mathcal{A}^*(\tilde{y} - y)\rangle + \|\tilde{E} - E\|_F^2 + \langle \alpha\mathcal{A}(X) - \alpha\mathcal{A}(\tilde{X}),\ \tilde{y} - y\rangle.$$

*Proof* By Lemma 2, we know that for all $V \in S^n$, the following statement holds

$$\langle U - V_+,\ V - V_+\rangle \leq 0, \quad \text{for all} \quad U \in S^n_+.$$

Thus, we can get

$$\begin{aligned}
&\Big\langle [X - \alpha(C - \mathcal{A}^*(y))]_+ - [\tilde{X} - \alpha(C - \mathcal{A}^*(\tilde{y}))]_+,\\
&\tilde{X} - \alpha(C - \mathcal{A}^*(\tilde{y})) - [\tilde{X} - \alpha(C - \mathcal{A}^*(\tilde{y}))]_+\Big\rangle \leq 0,\\
&\Big\langle [\tilde{X} - \alpha(C - \mathcal{A}^*(\tilde{y}))]_+ - [X - \alpha(C - \mathcal{A}^*(y))]_+,\\
&X - \alpha(C - \mathcal{A}^*(y)) - [X - \alpha(C - \mathcal{A}^*(y))]_+\Big\rangle \leq 0.
\end{aligned}$$

That is,

$$\begin{aligned}
&\langle \tilde{E} - E + X - \tilde{X},\ \tilde{E} - \alpha(C - \mathcal{A}^*(\tilde{y}))\rangle \leq 0,\\
&\langle \tilde{E} - E + X - \tilde{X},\ \alpha(C - \mathcal{A}^*(y)) - E\rangle \leq 0.
\end{aligned}$$

Adding them yields

$$\begin{aligned}
0 &\geq \langle \tilde{E} - E + X - \tilde{X},\ \tilde{E} - E + \alpha\mathcal{A}^*(\tilde{y} - y)\rangle\\
&= \langle \tilde{E} - E,\ X - \tilde{X} + \alpha\mathcal{A}^*(\tilde{y} - y)\rangle + \|\tilde{E} - E\|_F^2 + \langle X - \tilde{X},\ \alpha\mathcal{A}^*(\tilde{y} - y)\rangle\\
&= \langle \tilde{E} - E,\ X - \tilde{X} + \alpha\mathcal{A}^*(\tilde{y} - y)\rangle + \|\tilde{E} - E\|_F^2 + \langle \alpha\mathcal{A}(X) - \alpha\mathcal{A}(\tilde{X}),\ \tilde{y} - y\rangle.
\end{aligned}$$

The proof is complete. □

## 3 Method

In this section, we state, step by step, our proposed method for solving the SDPs that we call *Method 1*.

Step 0. Choose $X^0 \in S^n,\ y^0 \in R^m$. Choose $\alpha > 0,\ 0 < \sigma < 1,\ 0 < \gamma < 2\sigma$. Set $k := 0$.

Step 1. Choose $0 \leq \mu_k \leq 2(1 - \sigma)/\max\{1,\ \|AA^T\|\}$. Compute

$$\bar{y}^k = y^k - \mu_k\left(\mathcal{A}\left(X^k\right) - b\right), \quad \bar{X}^k = \left[X^k - \alpha\left(C - \mathcal{A}^*\left(\bar{y}^k\right)\right)\right]_+.$$

Step 2. The new iterates $X^{k+1},\ y^{k+1}$ are given by

$$\begin{pmatrix} I & -\alpha A^T \\ \alpha A & I \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix} = -\gamma\alpha \begin{pmatrix} (x^k - \bar{x}^k)/\alpha \\ \mathcal{A}(X^k) - b \end{pmatrix}. \tag{11}$$

Set $k := k + 1$.

The method above has a close relation to the DR method when applied to solving the SDPs, and we will clarify it in Sect. 5. For an implementable version of this method, we will discuss it in more details in Sect. 6.

## 4 Convergence properties

In this section, we prove global convergence of the Method 1 and $o(1/\epsilon)$ iteration complexity of the DR method for solving the SDPs. We begin with the following assumption, which means that there exist primal and dual solutions with zero primal–dual gap.

**Assumption 1** There exist $X^*$ and $y^*$ such that

$$\mathcal{A}(X) = b, \quad X \succeq 0, \quad C - \mathcal{A}^*(y) \succeq 0, \quad \langle X,\ C - \mathcal{A}^*(y)\rangle = 0.$$

To further simplify the proof details of convergence analysis of the Method 1, we introduce the following lemma.

**Lemma 6** *Denote*

$$R := X - [X - \alpha(C - \mathcal{A}^*(\bar{y}))]_+, \quad s := \alpha(\mathcal{A}(X) - b). \tag{12}$$

*If Assumption 1 holds then*

$$\begin{pmatrix} r \\ s \end{pmatrix}^T \begin{pmatrix} I & -\alpha A^T \\ \alpha A & I \end{pmatrix} \begin{pmatrix} x - x^* \\ \bar{y} - y^* \end{pmatrix} \geq \left\| \begin{pmatrix} r \\ s \end{pmatrix} \right\|^2.$$

*Proof* Applying Lemma 2 with $U := X - R$ and $V := X^* - \alpha(C - \mathcal{A}^*(y^*))$ yields

$$\left\langle X - R - [X^* - \alpha(C - \mathcal{A}^*(y^*))]_+,\ X^* - \alpha(C - \mathcal{A}^*(y^*)) - [X^* - \alpha(C - \mathcal{A}^*(y^*))]_+ \right\rangle \leq 0, \tag{13}$$

where $X - R = [X - \alpha(C - \mathcal{A}^*(\bar{y}))]_+ \in S^n_+$. Since Assumption 1 holds, it follows from [25, Lemma 2.1] that there exist $X^*$ and $y^*$ such that

$$\mathcal{A}(X^*) = b, \quad X^* = [X^* - \alpha(C - \mathcal{A}^*(y^*))]_+,$$

which, together with (13), implies

$$\langle X - X^* - R,\ \alpha(C - \mathcal{A}^*(y^*))\rangle \geq 0. \tag{14}$$

Likewise, applying Lemma 2 with $U := X^* \in S^n_+$ and $V := X - \alpha(C - \mathcal{A}^*(\bar{y}))$ yields

$$\left\langle X^* - [X - \alpha(C - \mathcal{A}^*(\bar{y}))]_+,\ X - \alpha(C - \mathcal{A}^*(\bar{y})) - [X - \alpha(C - \mathcal{A}^*(\bar{y}))]_+ \right\rangle \le 0.$$

Therefore, by replacing $[X - \alpha(C - \mathcal{A}^*(\bar{y}))]_+$ with $X - R$ and rearranging the terms, we can further get

$$\langle X - X^* - R,\ R - \alpha(C - \mathcal{A}^*(\bar{y}))\rangle \ge 0. \tag{15}$$

Summing up (14) and (15) implies

$$\langle X - X^* - R,\ R + \alpha\mathcal{A}^*(\bar{y} - y^*)\rangle \ge 0\ \Rightarrow\ (x - x^* - r)^T\left(r + \alpha A^T(\bar{y} - y^*)\right) \ge 0. \tag{16}$$

Thus

$$\left[\begin{pmatrix} x - x^* \\ \bar{y} - y^* \end{pmatrix} - \begin{pmatrix} r \\ s \end{pmatrix}\right]^T \left[\begin{pmatrix} r \\ s \end{pmatrix} + \begin{pmatrix} 0 & \alpha A^T \\ -\alpha A & 0 \end{pmatrix}\begin{pmatrix} x - x^* \\ \bar{y} - y^* \end{pmatrix}\right] \ge 0, \tag{17}$$

where the upper part of this inequality exactly corresponds to (16), and the lower part comes from the notation

$$s := \alpha(\mathcal{A}(X) - b) = \alpha(Ax - b),$$

which implies $s - \alpha A(x - x^*) = s - \alpha(Ax - b) = 0$. By rearranging the terms in (17), we can achieve the desired result. □

We remark that our discussion lines mainly follow from proof techniques in [26] (also see [15] an extension to an infinite-dimensional Hilbert space), and the use of skew-symmetric matrix in (17) is inspired by [16]. Also, keep in mind that the $X'$s feasibility is not assumed in the whole proof.

**Theorem 1** *If Assumption 1 holds, then any sequence generated by the Method 1 converges to a solution of (8).*

*Proof* For all $r \in R^{n^2}$ and all $s \in R^m$, we always have

$$2r^T A^T s \le \|r\|^2 + \left\|A^T s\right\|^2 \le \|r\|^2 + \left\|AA^T\right\| \|s\|^2 \le \max\left\{1,\ \left\|AA^T\right\|\right\}\left(\|r\|^2 + \|s\|^2\right).$$

Dividing by $2(1 - \sigma)(\|r\|^2 + \|s\|^2)$ on both sides yields

$$\frac{r^T A^T s}{(1 - \sigma)(\|r\|^2 + \|s\|^2)} \le \frac{\max\{1,\ \|AA^T\|\}}{2(1 - \sigma)} := \mu_{\max}^{-1},$$

which implies that $(1 - \sigma)(\|r\|^2 + \|s\|^2) - \mu r^T A^T s \ge 0$ whenever $\mu \le \mu_{\max}$. Of course, the latter remains valid when $r,\ s$ here are specialized to the ones given in (12), respectively. Hence, it follows from this fact and Lemma 6 that

$$\begin{aligned}
&\begin{pmatrix} r^k \\ s^k \end{pmatrix}^T \begin{pmatrix} I & -\alpha A^T \\ \alpha A & I \end{pmatrix} \begin{pmatrix} x^k - x^* \\ y^k - y^* \end{pmatrix} \\
&\quad \geq \left\| \begin{pmatrix} r^k \\ s^k \end{pmatrix} \right\|^2 - \begin{pmatrix} r^k \\ s^k \end{pmatrix}^T \begin{pmatrix} I & -\alpha A^T \\ \alpha A & I \end{pmatrix} \begin{pmatrix} 0 \\ \bar{y}^k - y^k \end{pmatrix} \\
&\quad = \left\| r^k \right\|^2 + \left\| s^k \right\|^2 - \mu_k \left( \left( r^k \right)^T A^T s^k - \alpha^{-1} \left\| s^k \right\|^2 \right) \\
&\quad \geq \sigma \left( \left\| r^k \right\|^2 + \left\| s^k \right\|^2 \right) + (1-\sigma) \left( \left\| r^k \right\|^2 + \left\| s^k \right\|^2 \right) - \mu_k \left( r^k \right)^T A^T s^k \\
&\quad \geq \sigma \left( \left\| r^k \right\|^2 + \left\| s^k \right\|^2 \right), \qquad (18)
\end{aligned}$$

where the first equality follows from $\alpha(\bar{y}^k - y^k) = -\mu_k s^k$ (see Step 1 in the Method 1 and the notation $s^k = \alpha(Ax^k - b)$ ) and the last inequality from $\mu_k \leq \mu_{\max}$ and the discussion above.

Denote

$$d := \begin{pmatrix} r \\ s \end{pmatrix}, \quad M := \begin{pmatrix} 0 & -\alpha A^T \\ \alpha A & 0 \end{pmatrix}, \quad w := \begin{pmatrix} x \\ y \end{pmatrix}.$$

By using the notation above, we can rewrite (11) into the following form:

$$(I + M)w^{k+1} = (I + M)w^k - \gamma_k d^k. \qquad (19)$$

On the other hand, it follows from (18) that

$$\left( d^k \right)^T (I + M) \left( w^k - w^* \right) \geq \sigma \left\| d^k \right\|^2.$$

Thus, we further get

$$\begin{aligned}
&\left\| (I + M) \left( w^k - w^* \right) - \gamma d^k \right\|^2 \\
&\quad = \left\| (I + M) \left( w^k - w^* \right) \right\|^2 - 2\gamma \left( d^k \right)^T (I + M) \left( w^k - w^* \right) + \gamma^2 \left\| d^k \right\|^2 \\
&\quad \leq \left\| (I + M) \left( w^k - w^* \right) \right\|^2 - 2\sigma\gamma \left\| d^k \right\|^2 + \gamma^2 \left\| d^k \right\|^2 \\
&\quad = \left\| (I + M) \left( w^k - w^* \right) \right\|^2 - \gamma(2\sigma - \gamma) \left\| d^k \right\|^2.
\end{aligned}$$

Combining this with (19) yields

$$\left\| (I + M) \left( w^{k+1} - w^* \right) \right\|^2 \leq \left\| (I + M) \left( w^k - w^* \right) \right\|^2 - \gamma(2\sigma - \gamma) \left\| d^k \right\|^2,$$

which implies

(a) the sequence $\{\|(I+M)(w^k - w^*)\|\}$ converges;
(b) the sequence $\{\|d^k\|\}$ converges to zero.

It follows from the assertion (b) that, as $k \to +\infty$, we have

$$\left\|\mathcal{A}\left(X^k\right) - b\right\|^2 + \left\|X^k - \left[X^k - \alpha\left(C - \mathcal{A}^*\left(\bar{y}^k\right)\right)\right]_+\right\|_F^2 \Big/ \alpha^2 \to 0. \tag{20}$$

Meanwhile, it follows from (a) that the sequence $\{\|(I+M)(w^k - w^*)\|\}$ converges, so does the sequence $\{\|w^k - w^*\|\}$ because the skew-symmetry of $M$ tells us

$$\left\|(I+M)\left(w^k - w^*\right)\right\|^2 = \left\|w^k - w^*\right\|^2 + \left\|M\left(w^k - w^*\right)\right\|^2 \geq \left\|w^k - w^*\right\|^2.$$

Therefore, the sequence $\{w^k\}$ must be bounded, and we can choose a convergent subsequence $\{w^{k_j}\}$ such that $w^{k_j} \to w^\infty$ (as $k_j \to +\infty$), which implies

$$\left(X^{k_j}, y^{k_j}\right) \to \left(X^\infty, y^\infty\right), \quad \text{as} \quad k_j \to +\infty. \tag{21}$$

Since Lemma 2 tells us that the projection operator is continuous, taking the limits in (20) along $k_j$ yields

$$\mathcal{A}\left(X^\infty\right) - b = 0, \quad X^\infty - \left[X^\infty - \alpha\left(C - \mathcal{A}^*\left(y^\infty\right)\right)\right]_+ = 0,$$

where $\bar{y}^{k_j} \to y^\infty$ follows from the fact that

$$\bar{y}^{k_j} = y^{k_j} - \mu_{k_j}\left(\mathcal{A}\left(X^{k_j}\right) - b\right) \to y^\infty, \quad k_j \to +\infty.$$

This shows that the cluster point $(X^\infty, y^\infty)$ is a solution of (8). Below we prove that the whole sequence $(X^k, y^k)$ converges to $(X^\infty, y^\infty)$. In fact, in view of the discussions above, we have known the convergence of $\{\|w^k - w^*\|\}$, thus that of $\{\|w^k - w^\infty\|\}$. So, both $\{\|X^k - X^\infty\|_F\}$ and $\{\|y^k - y^\infty\|\}$ converge. This fact, together with (21), ensures that the whole sequence $\{(X^k, y^k)\}$ converges to $(X^\infty, y^\infty)$ as well.

So, the desired conclusions follow from Lemma 3′ and the discussions above. □

Note that convergence analysis of the Method 1 does not require $X^k$ to be feasible, i.e., $X^k$ can neither be positive semidefinite nor satisfy the equality constraints. Yet, as the iterations proceed, the sequence $\{X^k\}$ becomes asymptotically feasible.

It should be specially stressed that Theorem 1 does not tell us whether or not the sequence $\{\langle C, X^k\rangle\}$ decreases monotonically. See Table 1 below for a related numerical example.

In view of the discussion above (cf. (20)), we have known that, if we denote the merit function

$$f(X, y, \alpha) := \|\mathcal{A}(X) - b\|^2 + \|X - [X - \alpha(C - \mathcal{A}^*(y))]_+\|_F^2/\alpha^2, \tag{22}$$

**Table 1** Numerical results on the first problem with $n = 1500,\ m = 1, 125, 750$

| $k$ | 1 | 3 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|
| CPUtime | 3.46 | 8.81 | 14.68 | 17.92 | 20.10 | 23.01 | 25.08 |
| $c^T x^k$ | 46.31 | 57.63 | 56.39 | 56.38 | 56.38 | 56.38 | 56.38 |
| $\|x^k - x^*\|$ | 81.76 | 5.00 | 0.04 | 3.3e−3 | 2.9e−4 | 2.7e−5 | 5.8e−6 |
| $\epsilon_{1,k}$ | 93.8 | 6.12 | 0.05 | 4.2e−3 | 3.8e−4 | 3.6e−5 | 7.6e−6 |
| $\epsilon_{2,k}$ | 1100.9 | 1108.6 | 1108.1 | 1108.8 | 1108.2 | 1108.5 | 1109.0 |

then for any given positive number $\alpha$ the sequence $\{f(X^k,\ y^k,\ \alpha)\}$ must tend to zero. At the end of this section, we discuss the speed of its tendency to zero.

**Theorem 2** *In the Method 1, let $\sigma \to 1$ so that $\mu_k \equiv 0$ for $k = 0, 1, \ldots,$ and $0 < \gamma < 2$. Then for any given positive number $\alpha$ the sequence $\{f(X^k,\ y^k,\ \alpha)\}$ is decreasing and $f(X^k,\ y^k,\ \alpha) = o(1/k)$, where $f$ is given by (22).*

*Proof* In Lemma 5, if we let $(X^k,\ y^k) := (X,\ y),\ (X^{k+1},\ y^{k+1}) := (\tilde{X},\ \tilde{y})$, then we have

$$\begin{aligned}0 \geq \left\langle E^{k+1} - E^k,\ X^k - X^{k+1} + \alpha\mathcal{A}^*\left(y^{k+1} - y^k\right)\right\rangle + \left\|E^{k+1} - E^k\right\|_F^2 \\ + \left\langle \alpha\mathcal{A}\left(X^k\right) - \alpha\mathcal{A}\left(X^{k+1}\right),\ y^{k+1} - y^k\right\rangle.\end{aligned}$$

Combining this with the iterative formula $X^k - X^{k+1} + \alpha\mathcal{A}^*(y^{k+1} - y^k) = \gamma E^k$ yields

$$0 \geq \gamma\left\langle E^{k+1} - E^k,\ E^k\right\rangle + \left\|E^{k+1} - E^k\right\|_F^2 + \left\langle \alpha\mathcal{A}\left(X^k\right) - \alpha\mathcal{A}\left(X^{k+1}\right),\ y^{k+1} - y^k\right\rangle. \tag{23}$$

For the matrix inner product $\langle E^{k+1} - E^k,\ E^k\rangle$, we can make use of the identity relation

$$2\langle b - a,\ a\rangle = \|b\|^2 - \|a\|^2 + \|b - a\|^2, \quad \text{for all} \quad a,\ b \in R,$$

to get

$$\begin{aligned}&\gamma\left\langle E^{k+1} - E^k,\ E^k\right\rangle + \left\|E^{k+1} - E^k\right\|_F^2 \\ &\quad = \frac{\gamma}{2}\left\|E^{k+1}\right\|_F^2 - \frac{\gamma}{2}\left\|E^k\right\|_F^2 + \left(1 - \frac{\gamma}{2}\right)\left\|E^{k+1} - E^k\right\|_F^2.\end{aligned} \tag{24}$$

On the other hand, the lower part of the relation (11) corresponds to

$$\alpha\mathcal{A}\left(X^{k+1} - X^k\right) + y^{k+1} - y^k = -\gamma\alpha\left(\mathcal{A}\left(X^k\right) - b\right).$$

So, we have

$$\begin{aligned}
&\left\langle \alpha\mathcal{A}\left(X^k\right) - \alpha\mathcal{A}\left(X^{k+1}\right),\ y^{k+1} - y^k \right\rangle \\
&= \left\langle \alpha\mathcal{A}\left(X^k\right) - \alpha\mathcal{A}\left(X^{k+1}\right),\ \alpha\mathcal{A}\left(X^k\right) - \alpha\mathcal{A}\left(X^{k+1}\right) - \gamma\alpha\left(\mathcal{A}\left(X^k\right) - b\right) \right\rangle \\
&= \frac{\gamma}{2}\left\|\alpha\left(\mathcal{A}\left(X^{k+1}\right) - b\right)\right\|^2 - \frac{\gamma}{2}\left\|\alpha\left(\mathcal{A}\left(X^k\right) - b\right)\right\|^2 \\
&\quad + \left(1 - \frac{\gamma}{2}\right)\left\|\alpha\mathcal{A}\left(X^k\right) - \alpha\mathcal{A}\left(X^{k+1}\right)\right\|^2. \qquad (25)
\end{aligned}$$

Therefore, it follows from $0 < \gamma < 2$ and the inequalities (23)–(25) that

$$\left\|E^{k+1}\right\|_F^2 + \left\|\alpha\left(\mathcal{A}\left(X^{k+1}\right) - b\right)\right\|^2 \le \left\|E^k\right\|_F^2 + \left\|\alpha\left(\mathcal{A}\left(X^k\right) - b\right)\right\|^2.$$

Thus, the proof of the first conclusion is complete.

Next, we will prove the second conclusion. The proof needs a very recent proposition [27, Lemma 2.2]: let $\{\rho_k\}$, $\{\beta_k\}$, $\{\gamma_k\}$ be sequences of positive numbers. Assume that they satisfy

$$\rho_{k+1}^2 \le \rho_k^2 - \beta_k\gamma_k, \quad k = 0,\ 1, \ldots,$$

the sequence $\{\beta_k\}$ is nonsummable and the sequence $\{\gamma_k\}$ is decreasing. Then

$$\lim_{k\to+\infty} \sup \gamma_k \sum_{i=0}^{k} \beta_i \le 2\rho_0 \lim_{k\to+\infty} \rho_k. \qquad (26)$$

On the other hand, since we have assumed that $\sigma \to 1$ and $\gamma = 1$, it follows from the proof of Theorem 1 that

$$\begin{aligned}
\left\|(I+M)\left(w^{k+1} - w^\infty\right)\right\|^2 &\le \left\|(I+M)\left(w^k - w^\infty\right)\right\|^2 - \left\|d^k\right\|^2 \\
&= \left\|(I+M)\left(w^k - w^\infty\right)\right\|^2 - \alpha^2 f\left(X^k,\ y^k,\ \alpha\right), \quad k = 0,\ 1, \ldots,
\end{aligned}$$

where $w^\infty$ is the limit of the sequence $\{w^k\}$. Moreover, we have proved that

$$\lim_{k\to+\infty} \left\|(I+M)\left(w^k - w^\infty\right)\right\| = 0,$$

and the sequence $\{f(X^k,\ y^k,\ \alpha)\}$ is decreasing. Consequently, it follows from (26) that

$$\lim_{k\to+\infty} f\left(X^k,\ y^k,\ \alpha\right)(k+1)\alpha^2 = 0 \Rightarrow f\left(X^k,\ y^k,\ \alpha\right) = o(1/k).$$

The whole proof is complete. □

From Theorem 2, we can see that

$$f\left(X^k, y^k, 1\right) := \left\|\mathcal{A}\left(X^k\right) - b\right\|^2 + \left\|X^k - \left[X^k\left(C - \mathcal{A}^*\left(y^k\right)\right)\right]_+\right\|_F^2 = o(1/k).$$

This means that, if the stopping criterion $f(X^k, y^k, 1) \le \epsilon$ is used, then the DR method has $o(1/\epsilon)$ iteration complexity, which is considerably better than $O(1/\epsilon)$. To the best of our knowledge, this is the currently known best result concerning all splitting methods and alternating directions methods. Then, does the SDPsplit, as a variant of the DR method, also have such a nice property? This issue was posed by one of the referees and deserves further investigation.

Note that the idea of the first part's proof of Theorem 2 is due to Sun [28]. Yet, her paper was written in Chinese and, more importantly, it needs some effort to adapt the proof technique to our case, so we give a detailed proof here.

## 5 Relations to existing methods

In this section, we give an equivalent version of the DR method of Lions and Mercier when applied to solving the SDPs. Then, we demonstrate that [16, Algorithm 1] and our proposed SDPsplit can be viewed as a simple form and a new variant of this equivalent version, respectively.

First of all, we describe, step by step, the DR method for solving the SDPs. Roughly speaking, it corresponds to the Method 1 with $\sigma \to 1$.

Step 0. Choose $(X^0, y^0) \in S^n \times R^m$. Choose $\alpha > 0$, $0 < \gamma < 2$. Set $k := 0$.
Step 1. Compute

$$\tilde{X}^k = \left[X^k - \alpha\left(C - \mathcal{A}^*\left(y^k\right)\right)\right]_+.$$

Step 2. The new iterates $X^{k+1}$, $y^{k+1}$ are given by

$$\begin{pmatrix} I & -\alpha A^T \\ \alpha A & I \end{pmatrix}\begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix} = -\gamma\begin{pmatrix} x^k - \tilde{x}^k \\ \alpha(\mathcal{A}(X^k) - b) \end{pmatrix}. \tag{27}$$

Set $k := k + 1$.

Next, we explain the reasons why we call the method just described the DR method for solving the SDPs. To this end, we first consider the following problem of finding $w \in R^l$ such that

$$0 \in \partial\delta_\Omega(w) + Gw + q,$$

where $G$ is an $l \times l$ matrix, $q \in R^l$ and $\partial\delta_\Omega$ is the differential of the indicator function $\delta_\Omega$ defined by

$$\delta_\Omega(w) = \begin{cases} 0, & \text{if } w \in \Omega, \\ +\infty, & \text{otherwise,} \end{cases}$$

of some nonempty closed convex set $\Omega$ in $R^l$. When specialized to this case, the iterative formula of the DR method [15,26] is as follows. Choose $\alpha > 0$, for the current iterate $w^k \in R^l$, solve the system of linear equations

$$w + \alpha(Gw + q) = w^k + \alpha\left(Gw^k + q\right) - \gamma\left(w^k - (I + \alpha\partial\delta_\Omega)^{-1}\left(w^k - \alpha\left(Gw^k + q\right)\right)\right),$$

to get the new iterate $w^{k+1}$, i.e., this new iterate satisfies the following recursive formula

$$(I + \alpha G)w^{k+1} = (I + \alpha G)w^k - \gamma\left(w^k - (I + \alpha\partial\delta_\Omega)^{-1}\left(w^k - \alpha\left(Gw^k + q\right)\right)\right). \tag{28}$$

On the other hand, we know that $(I + \alpha\partial\delta_\Omega)^{-1}$ corresponds to the usual projection onto the set $\Omega$. So, if we set

$$w^k := \begin{pmatrix} x^k \\ y^k \end{pmatrix}, \quad G := \begin{pmatrix} 0 & -A^T \\ A & 0 \end{pmatrix}, \quad q := \begin{pmatrix} c \\ -b \end{pmatrix}, \quad \Omega = S^n_+ \times R^m,$$

and regard $\alpha(\mathcal{A}(X^k) - b)$ in (27), i.e., $\alpha(Ax^k - b)$, as the $y^k$ minus the usual projection of $y^k - \alpha(Ax^k - b)$ onto the $R^m$ space, then (27) has been of the form (28). Therefore, we can conclude that the method just described is nothing but the DR method for solving the SDPs.

Now let's check whether or not [16, Algorithm 1] is a simple form of the DR method above. First of all, it follows from Lemma 1 that

$$\left[C - \mathcal{A}^*\left(y^k\right) - X^k\right]_+ = C - \mathcal{A}^*\left(y^k\right) - X^k + \left[X^k - \left(C - \mathcal{A}^*\left(y^k\right)\right)\right]_+.$$

Thus, through the elimination of the dual variable $Z^k$, which is $[C - \mathcal{A}^*(y^k) - X^k]_+$, the basic iterative formula in [16, Algorithm 1] becomes

$$\begin{pmatrix} I & -A^T \\ A & I \end{pmatrix}\begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix} = -\gamma\begin{pmatrix} c - A^T y^k - z^k \\ \mathcal{A}(X^k) - b \end{pmatrix} = -\gamma\begin{pmatrix} x^k - \tilde{x}^k \\ \mathcal{A}(X^k) - b \end{pmatrix}. \tag{29}$$

That is to say, [16, Algorithm 1] is entirely equivalent to the DR method with $\alpha = 1$ when applied to solving the SDPs. By the way, we no longer follow to call [16, Algorithm 1] an alternating directions method because we do not know how to construct an appropriate augmented Lagrange function or merit function to justify such a name.

From the discussions above, we can see that our proposed Method 1 can be viewed as a new variant of the DR method by introducing an intermediate Lagrange multiplier at each iteration.

## 6 An implementable version

In this section, we give an implementable version of our proposed Method 1 and discuss related details.

*SDPsplit*: *an implementable version of the Method 1*

Step 0. Choose $X^0 \in S^n$ and compute $y^0$ by (35). Choose $\alpha_0 > 0$, $0.6 < \sigma < 1$. Calculate

$$h = 1/\max\left\{1, \left\|AA^T\right\|\right\}, \quad \mu_{\max} := 2(1-\sigma)h.$$

Choose $0 < \gamma_{\min} < \gamma_{\max} < 2\sigma$. Set $k := 0$.

Step 1. Check whether or not $X^k$, $y^k$ satisfy some stopping criterion. If so, stop. Otherwise, choose $\mu_k \le \mu_{\max}$. Compute

$$\bar{y}^k = y^k - \mu_k\left(\mathcal{A}\left(X^k\right) - b\right), \quad \bar{X}^k = \left[X^k - \alpha_k\left(C - \mathcal{A}^*\left(\bar{y}^k\right)\right)\right]_+. \tag{30}$$

Step 2. Choose $\gamma_k \in [\gamma_{\min}, \gamma_{\max}]$. Find $\Delta y^k$ such that

$$\left(\alpha_k^{-2} I + AA^T\right)\Delta y^k = -\gamma_k \alpha_k^{-1}\left(A\bar{x}^k - b\right). \tag{31}$$

Then compute

$$\Delta X^k = -\gamma_k\left(X^k - \bar{X}^k\right) + \alpha_k \mathcal{A}^*\left(\Delta y^k\right). \tag{32}$$

Step 3. The new iterates are given by

$$X^{k+1} = X^k + \Delta X^k, \quad y^{k+1} = y^k + \Delta y^k. \tag{33}$$

Step 4. Calculate

$$\phi_k := \left\|\mathcal{A}\left(X^k\right) - b\right\|^2, \quad \varphi_k := \left\|X^k - \left[X^k - \alpha_k\left(C - \mathcal{A}^*\left(\bar{y}^k\right)\right)\right]_+\right\|_F^2 / \alpha_k^2.$$

If $\phi_k \le 0.1\varphi_k$, we set $\alpha_{k+1} = 10\alpha_k$. Otherwise $\alpha_{k+1} = \alpha_k$. Set $k := k+1$.

From description of the SDPsplit, we can see that it has made use of prediction–correction strategy, namely first implement a prediction step to compute $(\bar{X}^k, \bar{y}^k)$ from known $(X^k, y^k)$ and then do a correction step to get $(X^{k+1}, y^{k+1})$ from $(\bar{X}^k, \bar{y}^k)$.

Below we discuss an equivalent expression of the iterative formulae (31)–(33) in the case of $\alpha_k \equiv \alpha$ and $\gamma_k \equiv \gamma$. Let

$$r^k = x^k - \bar{x}^k, \quad s^k = \alpha\left(Ax^k - b\right). \tag{34}$$

Thus, it follows from (31) that

$$\left(I + \alpha^2 AA^T\right) \Delta y^k = -\gamma_k s^k + \alpha \gamma_k A r^k.$$

Combining this with (32) yields

$$\begin{pmatrix} I & 0 \\ \alpha A & I \end{pmatrix} \begin{pmatrix} I & -\alpha A^T \\ 0 & I + \alpha^2 AA^T \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \end{pmatrix} = -\gamma \begin{pmatrix} r^k \\ s^k \end{pmatrix}.$$

So, it follows from (33) that

$$\begin{pmatrix} I & -\alpha A^T \\ \alpha A & I \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ y^{k+1} - y^k \end{pmatrix} = -\gamma \begin{pmatrix} r^k \\ s^k \end{pmatrix} = -\gamma \alpha \begin{pmatrix} (x^k - \bar{x}^k)/\alpha \\ \mathcal{A}(X^k) - b \end{pmatrix}.$$

This just coincides with the basic iterative formula described in (11). Such an interrelation, together with the discussions above, explains the reason why the SDPsplit is an implementable version of the Method 1.

## 6.1 How to solve the subproblem in Step 1

Clearly, at each iteration of the SDPsplit, one major computational task is to project the matrix $X^k - \alpha_k(C - \mathcal{A}^*(\bar{y}^k))$ onto the cone $S_+^n$. In practice, such a matrix is usually asymmetric and we can follow [5, Sect. 4.1] to alternatively project the matrix's symmetric part. Then, in view of Lemma 1, this can be done by performing the corresponding eigenvalue decomposition of $\mathcal{O}(n^3)$ operations, so is the projection of this matrix.

## 6.2 How to choose/update $\alpha$ and how to solve the subproblem in Step 2

By our numerical experience, how to choose/update $\alpha$ can affect the algorithm's performance significantly. The actual observations motivate us to do this carefully. First of all, we analyze how to choose an initial $\alpha_0$. By comparing (30) with (6), we can find out the correspondence between $\alpha$ and $1/\mu$, where $\mu$ is the penalty parameter and thus must be small. This means that $\alpha_0$ shall be not small. As to the way of updating $\alpha$ in Step 4, the idea is as follows. If $\varphi_k$ is larger, for example $\varphi_k \geq 10\phi_k$ as described in Step 4, we can expect a reduction in $\varphi_k$ by increasing $\alpha$ in the next step. This can be justified by (20) and the second statement of Lemma 4. Thus, $\phi_k$ and $\varphi_k$ will tend to zero in a desirably balanced way. Yet, for convergence of the SDPsplit, we update $\alpha$ dynamically only for the first 30 or 50 iterations. During this process, we may either choose the conjugate gradient method with two or three (not too much!) iterations to return an approximate solution of (31) or solve it by Matlab solver. After which, the $\alpha$ becomes a constant. In such case, we may solve (31) via the Cholesky factorization of $\alpha^{-2}I + AA^T$. Generally speaking, this is enough to retain convergence and keep better performance of the SDPsplit. For the reader interested in applying the conjugate gradient method to (31), we refer [29, Sect. 4] for a pertinent discussion.

Here we would like to point out that the co-efficient matrix $\alpha^{-2}I + AA^T$ must be symmetric positive definite. In contrast, those algorithms [6–9] have roots in augmented Lagrange functions and thus have to solve the systems of linear equations, and the corresponding co-efficient matrices $AA^T$ can be (asymptotically) symmetric positive semidefinite provided that the $\mathcal{A}$'s surjectivity is not assumed. This may break down their practical implementations at times, see the third test problem below for an instance.

## 6.3 How to choose an initial value of Lagrange multiplier

For the choice of an initial value of Lagrange multiplier, we can make use of the iterative formula (4) of the boundary point method. Set $Z^0 = 0$, and solve

$$AA^T y = \mathcal{A}\left(C - Z^0\right) - \mu\left(\mathcal{A}\left(X^0\right) - b\right),$$

to get our $y^0$ for some small $\mu > 0$. Alternatively, to enhance the robustness, we solve

$$\left(\delta I + AA^T\right) y = \mathcal{A}(C) - \mu\left(\mathcal{A}\left(X^0\right) - b\right), \quad \text{with} \quad \mu := \delta / \max\left\{1,\ \left\|\mathcal{A}\left(X^0\right) - b\right\|\right\}, \tag{35}$$

to get our $y^0$ for some prescribed $\delta := 10^{-4}$. In practice, we use Matlab solver to get an approximate solution of this system of linear equations and take it to be our $y^0$. Thus, once $X^0$ is chosen, an initial value of Lagrange multiplier is located correspondingly.

## 6.4 How to choose $\sigma$, $\mu$ and $\gamma$

How to choose $\sigma$ and $\gamma$ is not a difficult thing. By our numerical experience, when $\sigma$ is in the interval $(0.6,\ 0.9)$ and $\gamma_k$ is around one, the performance of the SDPsplit seems good enough and robust. So we generally set $\sigma = 0.7$ and $\gamma_k \equiv 1$ in practice. About the choice of $\mu_k$, its similarity to the $\mu^{-1}$ in (3) tells us that the value of $\mu_k$ shall be large because the corresponding penalty parameter $\mu$ shall be always small. This motivates us to choose $\mu_k \equiv \mu_{\max}$ invariantly and its usefulness has been supported by our numerical experience.

## 6.5 How to choose an appropriate stopping criterion

Today, as far as the algorithms for the SDPs are concerned, how to choose an appropriate stopping criterion looks unrealistic, when applied to solving some large-scale real-life problems. In fact, if

$$f\left(X^k,\ y^k,\ 1\right) \le \epsilon,$$

or something like is chosen, then one may raise a serious question about possibility of justifying that the closeness of $X^k$ to the solution set is really implied by a small positive number $\epsilon$ in general. In contrast, when one evaluates the algorithms for unconstrained minimization problems, an appropriate choice is to take the closeness of the iterates to the solution as a stopping criterion because the solution of each test problem [30] is known. Based on these considerations, in this paper we prefer to construct several test problems, with the unique solution being available, to evaluate our proposed SDPsplit. In this sense, this is indeed an urgent task to construct more and more well-devised test problems used for scientific evaluation about the algorithms for the SDPs.

## 7 Rudimentary experiments

In this section, we implemented the SDPsplit to solve three test examples to confirm its efficiency and robustness. In our writing style, rather than striving for maximal test problems, we tried to make the basic ideas and techniques as clear as possible. And we adhered to the following principles:

- We did not sacrifice global convergence of the SDPsplit and its $o(1/\epsilon)$ iteration complexity in order to improve numerical results;
- For all chosen test examples, their individual solution is known in advance. So, we took the closeness of the iterates to the solution as the stopping criterion, with the aim at seeking a rigorous (not seemingly right but actually unwarranted) evaluation about the algorithm's numerical performance.

All numerical experiments were run in MATLAB R2014a (8.3.0.532) with 32-bit (win32) on a desktop computer with an Intel(R) Core(TM) i3-2120 CPU 3.30 GHz and 2 GB of RAM. The operating system is Windows XP Professional.

Before implementing the SDPsplit, we followed mprw2 to normalize the problem date $C, A, b$. In its practical implementations, we always adopted $\mu_k \equiv \mu_{\max}$ and chose all other parameters as follows: $\alpha_0 = 10^4$ (or $10^5$), $\sigma = 0.7$, $\gamma_k \equiv 1$. After we had determined the parameters $\mu_k, \sigma, \gamma_k$, we ever chose $\alpha_0 = 10^j$, $j = 0, 1, 2, 3, 4, 5$, respectively, and we observed that the $\alpha_0 = 10^4$ yields better numerical results on the first two problems and the $\alpha_0 = 10^5$ yields better numerical results on the third problem.

For the first $K$ (say, $K = 10$) iterations, we updated $\alpha$. And we solved the subproblem (31) by the conjugate gradient method *via three iterations* to return an approximate solution. After which, we kept $\alpha$ unchanged, i.e., we let $\alpha_k \equiv \alpha_K$ whenever $k > K$ so as to retain convergence of the SDPsplit. Then, we solved (31) via the Cholesky factorization.

Our first test problem was constructed in the following way so that the uniquely determined solution is an interior point and the corresponding $AA^T$ is positive definite:

$$\begin{aligned}&\text{Minimize } \langle C, X\rangle\\ &\text{Subject to } \left\langle A_{ij}, X\right\rangle = 1, \quad i = 1, \ldots, n, \quad j \geq i, \quad X \succeq 0,\end{aligned}$$

where $A_{ii}$ $(i = 1, \ldots, n)$ denotes an $n \times n$ diagonal matrix whose $(i, i)$-th entry is one and all other entries are zeros and $A_{ij}$ $(j > i)$ denotes an $n \times n$ symmetric matrix whose $(i, j)$-th entry and $(j, i)$-th entry are ones and all other entries are zeros. Note that such $n(n+1)/2$ matrices define the unique feasible point $X^* = (I + ee^T)/2$, which is an interior point. Of course, it is also the unique and full-rank solution of this SDP.

Our second test problem was constructed in the following way so that the uniquely determined solution is a boundary point and the corresponding $AA^T$ is positive definite:

$$\begin{aligned}&\text{Minimize } \langle C, X\rangle\\ &\text{Subject to } \langle A_{ii}, X\rangle = 1, \quad \left\langle A_{ij}, X\right\rangle = 2, \quad i = 1, \ldots, n, \quad j > i, \quad X \succeq 0,\end{aligned}$$

where $A_{ii}$ ($i = 1, \ldots, n$) denotes an $n \times n$ diagonal matrix whose $(i, i)$-th entry is one and all other entries are zeros and $A_{ij}$ $(j > i)$ denotes an $n \times n$ symmetric matrix whose $(i, j)$-th entry and $(j, i)$-th entry are ones and all other entries are zeros. Note that such $n(n+1)/2$ matrices define the unique feasible point $X^* = ee^T$, which is a boundary point. Of course, it is also the unique and low-rank solution of this SDP.

Our third test problem was constructed by modifying the first one in the following way so that the corresponding $AA^T$ is singular:

$$\begin{aligned}&\text{Minimize } \langle C, X\rangle\\ &\text{Subject to } \left\langle A_{ij}, X\right\rangle = 1, \quad i = 1, \ldots, n, \quad j \geq i, \quad \langle I, X\rangle = n, \quad X \succeq 0,\end{aligned}$$

where each $A_{ij}$ is the matrix defined in the first example. Note that the first $n(n+1)/2$ matrices define the unique feasible point $X^* = (I+ee^T)/2$, which is an interior point, and the last equality constraint is superfluous. Of course, its unique solution remains unchanged.

We run the SPDsplit for these three test problems, and because of out-of-memory issue we had to set the largest matrix dimension $n = 1500$ and set

$$C = \text{sprandsym}(n, 0.1); \quad c = C(:); \quad X0 = 10 * \text{eye}(n, n).$$

Be aware of that the randomly generated matrix $C$ is commonly different for each single test problem. The corresponding numerical results were reported in the following tables, where "CPUtime" stands for the solution time (in seconds) of the SDPsplit and

$$\epsilon_{1,k} := \left\|Ax^k - b\right\|, \quad \epsilon_{2,k} := \left\|X^k - \left[X^k - \left(C - \mathcal{A}^*\left(y^k\right)\right)\right]_+\right\|_F.$$

From the tables above, we can see that the SDPsplit solved all these test problems efficiently and robustly. In each case, the speed of reduction in $\|x^j - x^*\|$ $(j = 1, \ldots, 5)$ is fast. Motivated by a suggestion from the above-mentioned referee, we also run the state-of-art algorithm mprw2 to solve these three test problems. Like the SDPsplit, it solved the first two efficiently. But for the third—a rather hard problem with $AA^T$ being singular and non-diagonal, it failed. This situation outstands robustness of the SDPsplit discussed in this paper (Tables 2, 3).

**Table 2** Numerical results on the second problem with $n = 1500,\ m = 1{,}125{,}750$

| $k$ | 1 | 3 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|
| CPUtime | 3.31 | 8.72 | 14.30 | 16.73 | 19.61 | 23.26 | 24.98 |
| $c^T x^k$ | 112.25 | 105.52 | 108.27 | 108.22 | 108.22 | 108.22 | 108.22 |
| $\|x^k - x^*\|$ | 164.17 | 9.97 | 0.07 | 6.5e−3 | 5.9e−4 | 5.4e−5 | 1.7e−5 |
| $\epsilon_{1,k}$ | 118.19 | 12.20 | 0.09 | 8.3e−3 | 7.6e−4 | 7.1e−5 | 2.0e−5 |
| $\epsilon_{2,k}$ | 2125.5 | 2143.1 | 2140.2 | 2144.1 | 2143.7 | 2143.2 | 2143.8 |

**Table 3** Numerical results on the third problem with $n = 1500,\ m = 1{,}125{,}751$

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| CPUtime | 3.63 | 6.43 | 8.93 | 11.85 | 14.92 | 17.14 |
| $c^T x^k$ | −65.44 | −66.80 | −64.60 | −64.26 | −64.29 | −64.29 |
| $\|x^k - x^*\|$ | 7.01 | 6.59 | 0.56 | 0.05 | 4.3e−3 | 1.1e−3 |
| $\epsilon_{1,k}$ | 9.91 | 9.31 | 0.79 | 0.07 | 0.02 | 0.04 |
| $\epsilon_{2,k}$ | 1109.0 | 1109.1 | 1108.4 | 1108.8 | 1108.7 | 1108.7 |

At the end of this section, we would like to made remarks on our test examples. Although one may argue that they are not real SDPs arising in practice. However, each has a nice property: we can numerically confirm that, for the first few iterations of the SDPsplit, the value of $\max\{\epsilon_{1,k},\ \epsilon_{2,k}\}$ or something like can not reflect the closeness of the iterates to the solution in a desirable way. Furthermore, it is not shared by practical test problems since one typically does not know their solution(s) in advance.

## 8 Conclusions

In this paper, we have introduced and studied a variant of the DR method when applied to the standard SDP, and under weak assumptions we have proved its global convergence, and we have derived the currently best iteration complexity of the DR method among all splitting methods and alternating directions methods. Furthermore, we have suggested an implementable version of this variant, with implementation details being analyzed by rigorous theoretical analysis instead of pure numerical experience. Also, we have done rudimentary numerical experiments to confirm its efficiency and robustness on certain large-scale test problems.

Recently, Davi and Jarr [31] proposed a first-order approach for solving SDP, with the aim of finding a high-accuracy solution in spite of only using partial second-order information. It appears to be also possible to use our approach to find an approximate solution and switch to their suggested algorithm to improve this approximation. How to realize it is one of our on-going research topics.

**Acknowledgments** An original draft (Lemma 6 and Theorem 1 included) of this paper was presented in the Eighth National Conference of China's Mathematical Programming Society, held at Shanghai University in May 2010. The $o(1/k)$ iteration complexity of the DR method was analyzed in August of 2012 after the proposal of Lemma 2.2 in [27]. The author is very grateful to the AE and the referees for their careful reading and useful suggestions. Special thanks go to Dr. Y.Y. Huang, Dr. X.H. Lu, Dr. C.Z. Zhang and Dr. X. Zhang for writing Matlab codes.

## References

1. Alizadeh, F.: Interior point methods in semi-definite programming with applications to combinatorial optimization. SIAM J. Optim. **5**, 13–51 (1995)
2. Nesterov, Y.E., Nemirovsky, A.S.: Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms. SIAM Publications, SIAM, Philadelphia (1994)
3. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Rev. **38**(1), 49–95 (1996)
4. Todd, M.J.: Semi-definite optimization. Acta Numer. **10**, 515–560 (2001)
5. Burer, S., Vandenbussche, D.: Solving lift-and-project relaxations of binary integer programs. SIAM J. Optim. **16**(3), 726–750 (2006)
6. Povh, J., Rendl, F., Wiegele, A.: A boundary point method to solve semi-definite programs. Computing **78**, 277–286 (2006)
7. Malick, J., Povh, J., Rendl, F., Wiegele, A.: Regularization methods for semi-definite programming. SIAM J. Optim. **20**(1), 336–356 (2009)
8. Wen, Z.W., Goldfarb, D., Yin, W.T.: Alternating direction augmented Lagrangian methods for semi-definite programming. Math. Program. Comput. **2**, 203–230 (2010)
9. Zhao, X.Y., Sun, D.F., Toh, K.: A Newton-CG augmented Lagrangian method for semidefinite programming. SIAM J. Optim. **20**(4), 1737–1765 (2010)
10. Malick, J.: A dual approach to semidefinite least-squares problems. SIAM J. Matrix Anal. Appl. **26**, 272–284 (2004)
11. Monteiro, R.D.C., Ortiz, C., Svaiter, B.F.: Implementation of a block-decomposition algorithm for solving large-scale conic semidefinite programming problems. Comput. Optim. Appl. **6**(2), 103–150 (2014)
12. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. **16**, 964–979 (1979)
13. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**(3), 293–318 (1992)
14. Combettes, P.L.: Solving monotone inclusions via compositions of nonexpansive averaged operators. Optimization **53**, 475–504 (2004)
15. Dong, Y.D., Fischer, A.: A family of operator splitting methods revisited. Nonlinear Anal. **72**, 4307–4315 (2010)
16. Yu, Z.S.: Solving semidefinite programming problems via alternating direction methods. J. Comput. Appl. Math. **193**, 437–445 (2006)
17. Schwertman, N.C., Allen, D.M.: Smoothing an indefinite variance–covariance matrix. J. Stat. Comput. Simul. **9**, 183–194 (1979)
18. Higham, N.: Computing a nearest symmetric positive semidefinite matrix. Linear Algebra Appl. **103**, 103–118 (1988)
19. Malick, J., Sendov, H.S.: Clarke generalized Jacobian of the projection onto the cone of positive semidefinite matrices. Set Valued Anal. **14**, 273–293 (2006)
20. Helmberg, C.: Semidefinite Programming for Combinatorial Optimization. Habilitationsschrift TU, Berlin (2000)
21. Gafni, E.M., Bertsekas, D.P.: Two-metric projection methods for constrained optimization. SIAM J. Control Optim. **22**(6), 936–964 (1984)
22. Toint, P.H.L.: Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space. IMA J. Numer. Anal. **8**(2), 231–252 (1988)
23. Zhu, T., Yu, Z.G.: A simple proof for some important properties of the projection mapping. Math. Inequal. Appl. **7**, 453–456 (2004)
24. Huang, Y.Y., Dong, Y.D.: New properties of forward–backward splitting and a practical proximal-descent algorithm. Appl. Math. Comput. **237**, 60–68 (2014)
25. Tseng, P.: Merit function for semidefinite complementarity problems. Math. Program. **83**, 159–185 (1998)

26. He, B.S.: Inexact implicit methods for monotone general variational inequalities. Math. Program. **86**, 199–217 (1999)
27. Dong, Y.D.: The proximal point algorithm revisited. J. Optim. Theory Appl. **161**, 478–489 (2014)
28. Sun, X.Z.: A descent property of implicit method for monotone variational inequality. Numer. Math. J. Chin. Univ. **32**(1), 75–80 (2002)
29. Fountoulakis, K., Gondzio, J., Zhlobich, P.: Matrix-free interior point method for compressed sensing problems. Math. Program. Comput. **6**(1), 1–31 (2014)
30. Moré, B.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization. ACM Trans. Math. Softw. **7**, 17–41 (1981)
31. Davi, T., Jarre, F.: High-accuracy solution of large-scale semidefinite programs. Optim. Methods Softw. **27**(4–5), 655–666 (2012)

**Yunda Dong** received his MSc Degree from the Chinese Academy of Sciences. He completed his PhD Program in computational mathematics at Nanjing University in June of 2003, and then was at Technische Universität Dresden from September of 2003 to July of 2005. Then he moved to the School of Mathematics and Statistics, Zhengzhou University. His research interest is mainly in the area of (monotone) inclusions and their numerical methods.