

# A generalized alternating direction method of multipliers with semi-proximal terms for convex composite conic programming

Yunhai Xiao<sup>1</sup> · Liang Chen<sup>2</sup> · Donghui Li<sup>3</sup>

Received: 17 October 2016 / Accepted: 12 January 2018 / Published online: 23 January 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature and The Mathematical Programming Society 2018

**Abstract** In this paper, we propose a generalized alternating direction method of multipliers (ADMM) with semi-proximal terms for solving a class of convex composite conic optimization problems, of which some are high-dimensional, to moderate accuracy. Our primary motivation is that this method, together with properly chosen semi-proximal terms, such as those generated by the recent advance of block symmetric Gauss–Seidel technique, is capable of tackling these problems. Moreover, the proposed method, which relaxes both the primal and the dual variables in a natural way with a common relaxation factor in the interval of  $(0, 2)$ , has the potential of enhancing the performance of the classic ADMM. Extensive numerical experiments on various doubly non-negative semidefinite programming problems, with or without inequality constraints, are conducted. The corresponding results showed that all these multi-block problems can be successively solved, and the advantage of using the relaxation step is apparent.

---

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s12532-018-0134-9>) contains supplementary material, which is available to authorized users.

---

✉ Yunhai Xiao  
yhxiao@henu.edu.cn

Liang Chen  
chl@hnu.edu.cn

Donghui Li  
dhli@scnu.edu.cn

<sup>1</sup> Institute of Applied Mathematics, College of Mathematics and Statistics, Henan University, Kaifeng 475000, China

<sup>2</sup> College of Mathematics and Econometrics, Hunan University, Changsha 410082, China

<sup>3</sup> School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China

**Keywords** Convex composite conic programming · Alternating direction method of multipliers · Doubly non-negative semidefinite programming · Relaxation · Semi-proximal terms

**Mathematics Subject Classification** 90C22 · 90C25 · 90C06 · 65K05

### 1 Introduction

Let  $\mathbf{X}$  and  $\mathbf{Y}$  be two finite-dimensional real Euclidean spaces each equipped with an inner product  $\langle \cdot, \cdot \rangle$  and its induced norm  $\| \cdot \|$ . The primary motivation of this paper is to develop an efficient first-order method to solve the following convex composite conic programming problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \theta(\mathbf{x}) + \langle \mathbf{c}, \mathbf{x} \rangle \\ \text{s.t.} \quad & \mathcal{H}\mathbf{x} - \mathbf{b} \in \mathcal{C}, \mathbf{x} \in \mathcal{K}, \end{aligned} \tag{1}$$

where  $\theta : \mathbf{X} \rightarrow (-\infty, +\infty]$  is a closed proper convex function,  $\mathcal{H} : \mathbf{X} \rightarrow \mathbf{Y}$  is a linear map with its adjoint denoted by  $\mathcal{H}^*$ ,  $\mathbf{c} \in \mathbf{X}$  and  $\mathbf{b} \in \mathbf{Y}$  are given vectors,  $\mathcal{C} \subseteq \mathbf{Y}$  and  $\mathcal{K} \subseteq \mathbf{X}$  are two closed convex cones.

By simple calculations, one can observe that the dual of problem (1) can be recast as

$$\begin{aligned} \min_{\mathbf{s}, \mathbf{z}, \mathbf{y}} \quad & \theta^*(-\mathbf{s}) + \delta_{\mathcal{K}^*}(\mathbf{z}) + \delta_{\mathcal{C}^*}(\mathbf{y}) - \langle \mathbf{b}, \mathbf{y} \rangle \\ \text{s.t.} \quad & \mathbf{s} + \mathbf{z} + \mathcal{H}^*\mathbf{y} = \mathbf{c}, \end{aligned} \tag{2}$$

where  $\mathcal{C}^*$  and  $\mathcal{K}^*$  are the dual cones of  $\mathcal{C}$  and  $\mathcal{K}$ , respectively,  $\theta^*$  is the Fenchel conjugate of  $\theta$ ,  $\delta_{\mathcal{C}^*}(\cdot)$  and  $\delta_{\mathcal{K}^*}(\cdot)$  are the indicator functions of  $\mathcal{C}^*$  and  $\mathcal{K}^*$ , respectively. Let  $\mathcal{D} : \mathbf{X} \rightarrow \mathbf{X}$  be a given nonsingular linear operator and  $\mathcal{D}^*$  being its adjoint. Then, we can reformulate problem (2) equivalently as

$$\begin{aligned} \min_{\mathbf{s}, \mathbf{z}, \mathbf{y}, \mathbf{u}} \quad & \theta^*(-\mathbf{s}) + \delta_{\mathcal{K}^*}(\mathbf{z}) + \delta_{\mathcal{C}^*}(\mathbf{u}) - \langle \mathbf{b}, \mathbf{y} \rangle \\ \text{s.t.} \quad & \mathbf{s} + \mathbf{z} + \mathcal{H}^*\mathbf{y} = \mathbf{c}, \\ & \mathcal{D}^*(\mathbf{u} - \mathbf{y}) = 0. \end{aligned} \tag{3}$$

Obviously, problem (3) can be viewed (by setting  $(\mathbf{s}, \mathbf{u})$  as one block-variable) as an example of the following general multi-block convex composite problem

$$\begin{aligned} \min \quad & f_1(y_1) + f_2(y_1, y_2, \dots, y_p) + g_1(z_1) + g_2(z_1, z_2, \dots, z_q) \\ \text{s.t.} \quad & \mathcal{A}_1^*y_1 + \mathcal{A}_2^*y_2 + \dots + \mathcal{A}_p^*y_p + \mathcal{B}_1^*z_1 + \mathcal{B}_2^*z_2 + \dots + \mathcal{B}_q^*z_q = c, \end{aligned} \tag{4}$$

where  $p$  and  $q$  are two given nonnegative integers,  $\mathcal{X}, \mathcal{Y}_1, \dots, \mathcal{Y}_p, \mathcal{Z}_1, \dots, \mathcal{Z}_q$  are finite-dimensional real Euclidean spaces each endowed with an inner product  $\langle \cdot, \cdot \rangle$  and its induced norm  $\| \cdot \|$ ,  $c \in \mathcal{X}$  being the given data,  $\mathcal{A}_i, i = 1, \dots, p$ , and  $\mathcal{B}_j, j = 1, \dots, q$ , are linear maps from  $\mathcal{Y}_i$  and  $\mathcal{Z}_j$ , respectively, to  $\mathcal{X}$ ,  $f_1 : \mathcal{Y}_1 \rightarrow (-\infty, +\infty]$  and  $g_1 : \mathcal{Z}_1 \rightarrow (-\infty, +\infty]$  are simple closed proper convex functions,

$f_2 : \mathcal{Y} \rightarrow (-\infty, +\infty)$  and  $g_2 : \mathcal{Z} \rightarrow (-\infty, +\infty)$  are convex quadratic functions, in which  $\mathcal{Y}$  and  $\mathcal{Z}$  are defined by  $\mathcal{Y} := \mathcal{Y}_1 \times \mathcal{Y}_2 \times \dots \times \mathcal{Y}_p$  and  $\mathcal{Z} := \mathcal{Z}_1 \times \mathcal{Z}_2 \times \dots \times \mathcal{Z}_q$ .

For further simplicity, we define the linear maps  $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Y}$  and  $\mathcal{B} : \mathcal{X} \rightarrow \mathcal{Z}$  such that their adjoints are given by

$$\mathcal{A}^*y = \sum_{i=1}^p \mathcal{A}_i^*y_i \quad \forall y \in \mathcal{Y}, \quad \text{and} \quad \mathcal{B}^*z = \sum_{j=1}^q \mathcal{B}_j^*z_j \quad \forall z \in \mathcal{Z}.$$

Then, the model (4) falls into the following more general framework

$$\min_{y \in \mathcal{Y}, z \in \mathcal{Z}} \left\{ f(y) + g(z) \quad \text{s.t.} \quad \mathcal{A}^*y + \mathcal{B}^*z = c \right\}, \tag{5}$$

where  $f : \mathcal{Y} \rightarrow (-\infty, +\infty]$  and  $g : \mathcal{Z} \rightarrow (-\infty, +\infty]$  are two closed proper convex functions,  $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Y}$  and  $\mathcal{B} : \mathcal{X} \rightarrow \mathcal{Z}$  are two linear maps with their adjoints  $\mathcal{A}^*$  and  $\mathcal{B}^*$ , respectively, and  $c \in \mathcal{X}$  is a given vector.

Let  $\sigma > 0$  be the penalty parameter, the augmented Lagrangian function of problem (5) is defined by, for any  $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ ,

$$\mathcal{L}_\sigma(y, z; x) := f(y) + g(z) - \langle x, \mathcal{A}^*y + \mathcal{B}^*z - c \rangle + \frac{\sigma}{2} \|\mathcal{A}^*y + \mathcal{B}^*z - c\|^2. \tag{6}$$

Let  $\tau > 0$  be the step-length and choose an initial point  $(x^0, y^0, z^0) \in \mathcal{X} \times (\text{dom } f) \times (\text{dom } g)$ . The classic alternating direction method of multipliers (ADMM) scheme takes the following form, for  $k = 0, 1, \dots$ ,

$$\begin{cases} y^{k+1} \in \arg \min_y \mathcal{L}_\sigma(y, z^k; x^k), \\ z^{k+1} \in \arg \min_z \mathcal{L}_\sigma(y^{k+1}, z; x^k), \\ x^{k+1} := x^k - \tau \sigma (\mathcal{A}^*y^{k+1} + \mathcal{B}^*z^{k+1} - c). \end{cases} \tag{7}$$

Generally, the step-length  $\tau$  in (7) can be chosen in the interval of  $(0, (1 + \sqrt{5})/2)$ , and the thumb of the choice for  $\tau$  in numerical computations is the golden ratio of 1.618. Moreover, we should mention that the iteration scheme (7) may not be well-defined, and one may refer to [3] for details.

The classic ADMM, which is closely related to the method of multipliers by Hestenes [17], Powell [28] and Rockafellar [30] for solving constrained convex optimization problems, was originally proposed by Glowinski and Marroco [14] and Gabay and Mercier [12] in the mid-1970s. Early proofs for the convergence of the classic ADMM under certain conditions were given by Gabay and Mercier [12], Glowinski [13] and Fortin and Glowinski [10]. One may refer to [15] for a note on the historical development of the ADMM. In early-1980s, it was shown by Gabay [11] that the classic ADMM with the unit step-length is a special case of the Douglas–Rachford splitting method applied to finding the roots of the sum of two maximal monotone operators. In early-1990s, Eckstein and Bertsekas [7] showed that the Douglas–Rachford splitting

method itself is an instance of the proximal point algorithm [31] applied to a specially generated splitting operator. Based on this observation, and for the purpose of improving the performance of the classic ADMM with the unit step-length, Eckstein and Bertsekas [7] presented a generalized ADMM. For a recent survey, one may refer to [8].

Along a different line, Eckstien [6] introduced proximal terms to the ADMM to ensure that each of the subproblems involved admits a unique solution, without further assumptions on the objective functions and the constraints. Recently, a semi-proximal ADMM (sPADMM) with the step-length up to the golden ratio of 1.618 was proposed by Fazel et al. [9]. This extension has attracted much attention in the last couple of years in solving multi-block linearly constrained linear and quadratic convex semidefinite programming problems [2, 20–22, 33] to moderate accuracy. The successful applications of the two-block ADMM with semi-proximal terms to solving various multi-block problems for moderate accuracy inevitably inspire us to consider using generalized ADMM with semi-proximal terms to solve the convex composite conic programming (1). Moreover, the advantage of using the semi-proximal terms instead of the positive definite proximal terms is that the resulting algorithm not only reduces to the original generalized ADMM when the proximal terms are absent but also is more convenient in solving convex composite conic programming if the block symmetric Gauss–Seidel (sGS) iteration technique invented by Li, Sun and Toh [19] is adopted as in [2, 22, 33].

In this paper, we proposed a variant of generalized ADMM with semi-proximal terms, which can be used to solve problem (4) if the block sGS technique is involved. This variant is based on a crucial observation made by Chen [1] that the generalized ADMM of Eckstein and Bertsekas [7] can be reformulated equivalently as an ADMM with an extra relaxation step on both the primal and the dual variables with the relaxation factor lying in the open interval of  $(0, 2)$ . Furthermore, the semi-proximal terms used in this paper is in a pretty natural way due to the fact that only the most recent values of variables are always in it. To derive a more general theoretical convergence result, we particularly concentrate on the convergence analysis of the generalized ADMM with semi-proximal terms for solving the ‘two-block’ problem (5). Then, we illustrate how the proposed algorithm can be used to solve multi-block problems. At last, we conduct numerical experiments on a class linear doubly non-negative (DNN) semidefinite programming (SDP) problems with or without inequality constraints. The corresponding numerical results illustrate that the proposed method, together with the sGS technique, can solve these problems not only effectively but also efficiently.

The remaining parts of this paper are organized as follows. In Sect. 2, we provide some preliminary results, which will be used later, and give a quick review of some variants of the classic ADMM. In Sect. 3, we present a generalized ADMM with semi-proximal terms for solving the 2-block problem (5). Then, we discuss how to apply the proposed method, with properly chosen semi-proximal terms, to solving multi-block problems in Sect. 4. In Sect. 5, we focus on the convergence analysis of the proposed generalized ADMM. In Sect. 6, we are devoted to the implementation issues and numerical experiments of using the proposed method for solving DNN-SDP. The corresponding numerical results are provided. We conclude this paper in Sect. 7.

## 2 Preliminaries

We adopt the notation that  $w := (x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$  for convenience. Moreover, for any two finite-dimensional real Euclidean spaces  $\mathcal{X}'$  and  $\mathcal{X}''$  each endowed with an inner product  $\langle \cdot, \cdot \rangle$ , we take the convention that

$$\langle (x'_1, x''_1), (x'_2, x''_2) \rangle := \langle x'_1, x'_2 \rangle + \langle x''_1, x''_2 \rangle, \quad \forall x'_1, x'_2 \in \mathcal{X}', x''_1, x''_2 \in \mathcal{X}'',$$

and the corresponding norm in the product space induced by this inner product is also denoted by  $\| \cdot \|$ . We use  $\mathcal{I}$  to denote the identity linear operator, whose domain and range can be deduced from the context.

We denote the effective domains of  $f$  and  $g$  by  $\text{dom } f$  and  $\text{dom } g$ , respectively, and use  $\partial f$  and  $\partial g$  to denote their subdifferential mappings. It is well-known that  $\partial f$  and  $\partial g$  are maximal monotone operators [29] and there exist two self-adjoint positive semidefinite linear operators  $\Sigma_f : \mathcal{Y} \rightarrow \mathcal{Y}$  and  $\Sigma_g : \mathcal{Z} \rightarrow \mathcal{Z}$ , such that for any  $y, y' \in \mathcal{Y}$  and  $z, z' \in \mathcal{Z}$  with  $u \in \partial f(y), u' \in \partial f(y'), v \in \partial g(z),$  and  $v' \in \partial g(z'),$

$$\langle u - u', y - y' \rangle \geq \|y - y'\|_{\Sigma_f}^2 \quad \text{and} \quad \langle v - v', z - z' \rangle \geq \|z - z'\|_{\Sigma_g}^2. \quad (8)$$

The Lagrangian function of problem (5) is defined by

$$\mathcal{L}(y, z; x) := f(y) + g(z) - \langle x, \mathcal{A}^*y + \mathcal{B}^*z - c \rangle, \quad \forall (x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}.$$

A vector  $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$  is said to be a saddle point to the Lagrangian function if it is a solution to the following Karush-Kuhn-Tucker (KKT) system

$$\mathcal{A}x \in \partial f(y), \quad \mathcal{B}x \in \partial g(z) \quad \text{and} \quad \mathcal{A}^*y + \mathcal{B}^*z = c. \quad (9)$$

Throughout this paper, we make the following assumption which ensures  $(\bar{y}, \bar{z})$  is an optimal solution to problem (5) and  $\bar{x}$  is an optimal solution to the dual of problem (5).

**Assumption 1** There exists a vector  $(\bar{x}, \bar{y}, \bar{z}) \in \mathcal{X} \times (\text{dom } f) \times (\text{dom } g)$  satisfying the KKT system (9).

Next, we give a quick review of several ADMM-type methods. In order to improve the performance of the classic ADMM with the step-length  $\tau = 1$ , Eckstein and Bertsekas [7] proposed the following generalized ADMM<sup>1</sup> scheme:

$$\begin{cases} y^{k+1} := \arg \min_y \mathcal{L}_\sigma(y, z^k; x^k), \\ z^{k+1} := \arg \min_z \left\{ g(z) - \langle z, \mathcal{B}x^k \rangle + \frac{\sigma}{2} \left\| \rho \mathcal{A}^*y^{k+1} - (1 - \rho)\mathcal{B}^*z^k + \mathcal{B}^*z - \rho c \right\|^2 \right\}, \\ x^{k+1} := x^k - \sigma(\rho \mathcal{A}^*y^{k+1} - (1 - \rho)\mathcal{B}^*z^k + \mathcal{B}^*z^{k+1} - \rho c), \end{cases} \quad (10)$$

<sup>1</sup> In [7], the authors considered the case that  $\mathcal{B}^*$  is an identity operator and  $c = 0$ . However, it is easy to see that the scheme described in (10) is a direct application of their idea to problem (5).

where  $\rho \in (0, 2)$  is the relaxation factor. For  $\rho = 1$ , the above generalized ADMM scheme is exactly the classic ADMM scheme (10) with  $\tau = 1$ . In his Ph.D. thesis [1, Section 3.2], Chen has made the very interesting observation that, cyclically, the generalized ADMM scheme (10) is equivalent to the following ADMM scheme with initial point  $\tilde{w}^0 = (\tilde{x}^0, \tilde{y}^0, \tilde{z}^0) \in \mathcal{X} \times (\text{dom } f) \times (\text{dom } g)$ ,

$$\begin{cases} z^k := \arg \min \mathcal{L}_\sigma(\tilde{y}^k, z; \tilde{x}^k), \\ x^k := \tilde{x}^k - \sigma(\mathcal{A}^* \tilde{y}^k + \mathcal{B}^* z^k - c), \\ y^k := \arg \min \mathcal{L}_\sigma(y, z^k; x^k), \\ \tilde{w}^{k+1} := \tilde{w}^k + \rho(w^k - \tilde{w}^k), \end{cases} \tag{11}$$

where  $\rho \in (0, 2)$  is the uniform relaxation factor. For the subsequent discussion, we need to distinguish between the sequence  $\{(x^k, y^k, z^k)\}_{k \geq 0}$  generated by (10) and the sequence  $\{(x^k, y^k, z^k)\}_{k \geq 0}$  generated by (11). For this purpose, we use  $\{(\hat{x}^k, \hat{y}^k, \hat{z}^k)\}_{k \geq 0}$  as a surrogate for the sequence  $\{(x^k, y^k, z^k)\}_{k \geq 0}$  generated by (11). The equivalence between (10) and (11) is interpreted as follows. Its proof can be found in Chen [1, Section 3.2].

**Proposition 2.1** *If the initial point  $(x^0, y^0, z^0) \in \mathcal{X} \times (\text{dom } f) \times (\text{dom } g)$  for the generalized ADMM scheme (10) satisfies  $x^0 = \hat{x}^0$  and  $z^0 = \hat{z}^0 \in \text{dom } g$ , the sequence  $\{(x^{k+1}, y^{k+1}, z^{k+1})\}_{k \geq 0}$  generated by the generalized ADMM scheme (10) is exactly the sequence  $\{(\hat{x}^{k+1}, \hat{y}^k, \hat{z}^{k+1})\}_{k \geq 0}$  generated by the generalized ADMM scheme (11).*

The proximal method of multipliers of Rockafellar [30,32] for solving (5) takes the iterative scheme

$$(y^{k+1}, z^{k+1}) = \arg \min_{y \in \mathcal{Y}, z \in \mathcal{Z}} \mathcal{L}_\sigma(y, z; x^k) + \frac{1}{2}(\|y - y^k\|_{\mathcal{S}}^2 + \|z - z^k\|_{\mathcal{T}}^2) \tag{12}$$

where  $\mathcal{S}$  and  $\mathcal{T}$  are the identity operators multiplied by a positive scalar  $\mu$ . The proximal method of multipliers is globally convergent under very mild assumptions on the problem’s data, namely, convexity and existence of optimal solutions are enough to guarantee convergence. To preserve the good features of both proximal method of multipliers (12) and ADMM (7), Eckstein [6] firstly constructed a proximal ADMM, which iterates from the given point  $(y^k, z^k; x^k) \in \mathcal{Y} \times \mathcal{Z} \times \mathcal{X}$  in the form of

$$\begin{cases} y^{k+1} = \arg \min_{y \in \mathcal{Y}} \mathcal{L}_\sigma(y, z^k; x^k) + \frac{1}{2}\|y - y^k\|_{\mathcal{S}}^2, & (13a) \\ z^{k+1} = \arg \min_{z \in \mathcal{Z}} \mathcal{L}_\sigma(y^{k+1}, z; x^k) + \frac{1}{2}\|z - z^k\|_{\mathcal{T}}^2, & (13b) \\ x^{k+1} = x^k - \tau\sigma(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^{k+1} - c), & (13c) \end{cases}$$

where  $\tau = 1$ ,  $\mathcal{S} = \mu_1 \mathcal{I}$  and  $\mathcal{T} = \mu_2 \mathcal{I}$  with  $\mu_1 > 0$  and  $\mu_2 > 0$ .

It should be emphasised that the above mentioned approaches require uniformly the linear operators  $\mathcal{S}$  and  $\mathcal{T}$  to either be zero, or positive definite. However, the restriction on the positive definiteness will exclude some interesting situations such as the classic ADMM. Moreover, the guiding principle is that  $\mathcal{S}$  and  $\mathcal{T}$  should be as ‘small’ as possible while (13a) and (13b) are still relatively easy to compute. The exciting progress in designing convergent and efficient (proximal) ADMM in which  $\mathcal{S}$  and  $\mathcal{T}$  are required only to be self-adjoint positive semidefinite owing to Fazel et al. [9]. This algorithm framework, which was called smi-proximal ADMM (and was abbreviated as sPADMM) later, takes the form of (13a) to (13c) with  $\tau \in (0, \frac{1+\sqrt{5}}{2})$ . With properly ‘chosen’  $\mathcal{S}$  and  $\mathcal{T}$ , sPADMM has exhibited as a powerful algorithmic tool in solving multi-block linearly constrained linear and quadratic convex semidefinite programming problems [2, 20–22, 33] to moderate accuracy.

### 3 A generalized ADMM with semi-proximal terms

Observe that the subproblems in the generalized ADMM schemes (10) and (11) may not admit solutions because  $\mathcal{A}$  or  $\mathcal{B}$  is not assumed to be surjective, which is always the case for multi-block convex conic programming problems considered in [2, 20–22, 33]. One natural way to fix this problem is to add proximal terms to these subproblems. For this purpose and inspired by the sPADMM, we first choose  $\mathcal{S} : \mathcal{Y} \rightarrow \mathcal{Y}$  and  $\mathcal{T} : \mathcal{Z} \rightarrow \mathcal{Z}$  be two self-adjoint positive semidefinite linear operators. Note that although the generalized ADMM schemes (10) and (11) are equivalent in certain sense, one can get two different variants of the generalized ADMM with semi-proximal terms. On one hand, we may add the semi-proximal terms  $\frac{1}{2}\|y - y^k\|_{\mathcal{S}}$  and  $\frac{1}{2}\|z - z^k\|_{\mathcal{T}}^2$  to the subproblems for computing  $y^{k+1}$  and  $z^{k+1}$  in the generalized ADMM scheme (10). On the other hand, for the generalized ADMM scheme (11), the more natural choice for the semi-proximal terms to be added will be  $\frac{1}{2}\|y - \tilde{y}^k\|_{\mathcal{S}}$  and  $\frac{1}{2}\|z - \tilde{z}^k\|_{\mathcal{T}}^2$  to the subproblems for computing  $z^k$  and  $y^k$ . In fact, the former one can be reformulated, in light of (11), as

$$\begin{cases} z^k := \arg \min_z \mathcal{L}_\sigma(\tilde{y}^k, z; \tilde{x}^k) + \frac{1}{2}\|z - z^{k-1}\|_{\mathcal{T}}^2, \\ x^k := \tilde{x}^k - \sigma(\mathcal{A}^* \tilde{y}^k + \mathcal{B}^* z^k - c), \\ y^k := \arg \min_y \mathcal{L}_\sigma(y, z^k; x^k) + \frac{1}{2}\|y - y^{k-1}\|_{\mathcal{S}}^2, \\ \tilde{w}^{k+1} := \tilde{w}^k + \rho(w^k - \tilde{w}^k), \end{cases} \tag{14}$$

where  $z^{-1} := \tilde{z}^0$  and  $y^{-1} := \tilde{y}^0$ . In this paper, for the sake of generality and numerical convenience, we consider the latter variant with only semi-proximal terms. More precisely, we develop the following generalized ADMM with semi-proximal terms (abbr. GADMM) in which the most recent values of the variables are used in the proximal terms.

*Remark 3.1* The order of variable updating in Algorithm 1 is different from that used in (11). This swapping is just for the convenience of convergence analysis and essentially does not affect anything else.

**Algorithm 1 (GADMM):** *A generalized ADMM with semi-proximal terms*

**Initialization.** Set  $\rho \in (0, 2)$  and  $\sigma > 0$ . Choose  $\mathcal{S} : \mathcal{Y} \rightarrow \mathcal{Y}$  and  $\mathcal{T} : \mathcal{Z} \rightarrow \mathcal{Z}$  such that  $\Sigma_f + \mathcal{S} + \mathcal{A}\mathcal{A}^* \succ 0$  and  $\Sigma_g + \mathcal{T} + \mathcal{B}\mathcal{B}^* \succ 0$ . Choose an initial point  $(\tilde{x}^0, \tilde{y}^0, \tilde{z}^0) \in \mathcal{X} \times \text{dom}(f) \times \text{dom}(g)$ . For  $k = 0, 1, \dots$ ,

**Step 1 (main step).** *Compute*

$$y^k := \arg \min_y \mathcal{L}_\sigma(y, \tilde{z}^k; \tilde{x}^k) + \frac{1}{2} \|y - \tilde{y}^k\|_{\mathcal{S}}^2, \tag{15a}$$

$$x^k := \tilde{x}^k - \sigma(\mathcal{A}^*y^k + \mathcal{B}^*\tilde{z}^k - c), \tag{15b}$$

$$z^k := \arg \min_z \mathcal{L}_\sigma(y^k, z; x^k) + \frac{1}{2} \|z - \tilde{z}^k\|_{\mathcal{T}}^2. \tag{15c}$$

**Step 2 (relaxation step).** *Compute*

$$\tilde{w}^{k+1} := \tilde{w}^k + \rho(w^k - \tilde{w}^k). \tag{16}$$

*Remark 3.2* Compared with the scheme (14), the semi-proximal terms in (15a) and (15c) are more natural in the sense that the most recently updated values of variables are involved.

*Remark 3.3* The main objective of this paper is to solve problem (4), where  $f_2$  and  $g_2$  are convex quadratic functions. Therefore, the majorization technique of Li et al. [21], which can be used together with indefinite  $\mathcal{S}$  and  $\mathcal{T}$ , is not necessary in this paper. On the other hand, we are more interested in the techniques which can be used to solve multi-block problems. As a result, we only need  $\mathcal{S}$  and  $\mathcal{T}$  being positive semidefinite, despite the fact that they can be chosen as indefinite for certain cases. In fact, even  $f_2$  and  $g_2$  in (4) are not separable, a slight modification of our proposed algorithm is applicable by using the techniques in [4]. We keep these techniques away from this paper just for the ease of reading, although the convergence analysis for a unified algorithm with all of these techniques contained would not bring too much work.

### 4 Applications to multi-block problems

In this part, we discuss how to apply the GADMM, i.e. Algorithm 1 to solve multi-block separable convex minimization problems. For the further discussions, we use the notations that  $y_{<i} := (y_1, \dots, y_{i-1})$ ,  $y_{>i} := (y_{i+1}, \dots, y_p)$  for  $y \in \mathcal{Y}$  and similar notations for  $z \in \mathcal{Z}$ .

#### 4.1 The block symmetric Gauss–Sidel iteration case

Firstly, we consider the general convex composite quadratic optimization model (4). We denote the Hessians of the quadratic functions  $f_2$  and  $g_2$  by  $\mathcal{P}$  and  $\mathcal{Q}$ , respectively, which are self-adjoint positive semidefinite linear operators defined on  $\mathcal{Y}$  and  $\mathcal{Z}$ .



According the block structure of the variable  $y$  and  $z$ , we can further write  $\mathcal{P}$  and  $\mathcal{Q}$  as

$$\mathcal{P} = \begin{pmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} & \cdots & \mathcal{P}_{1p} \\ \mathcal{P}_{12}^* & \mathcal{P}_{22} & \cdots & \mathcal{P}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{P}_{1p}^* & \mathcal{P}_{2p}^* & \cdots & \mathcal{P}_{pp} \end{pmatrix} \quad \text{and} \quad \mathcal{Q} = \begin{pmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} & \cdots & \mathcal{Q}_{1q} \\ \mathcal{Q}_{12}^* & \mathcal{Q}_{22} & \cdots & \mathcal{Q}_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{Q}_{1q}^* & \mathcal{Q}_{2q}^* & \cdots & \mathcal{Q}_{qq} \end{pmatrix}.$$

Moreover, we choose  $\mathcal{E}_i$  and  $\mathcal{H}_j$  to be self-adjoint positive semidefinite linear operators on  $\mathcal{Y}_i$  and  $\mathcal{Z}_j$  for  $i = 1, \dots, p$  and  $j = 1, \dots, q$ , respectively, such that

$$\mathcal{E}_i + \sigma^{-1}\mathcal{P}_{ii} + \mathcal{A}_i\mathcal{A}_i^* \succ 0, \quad \text{and} \quad \mathcal{H}_j + \sigma^{-1}\mathcal{Q}_{jj} + \mathcal{B}_j\mathcal{B}_j^* \succ 0.$$

By adopting the smart symmetric Gauss–Seidel (sGS) technique invented by Li et al. [22], the main step (Step 1) of Algorithm 1 can be implemented step by step according to the following procedures.

---

**Algorithm 2** *The main step of Algorithm 1 based on block sGS*

---

**Step 1.1** *Compute*

$$\begin{cases} y_i^{k-\frac{1}{2}} := \arg \min_{y_i} \mathcal{L}_\sigma((\tilde{y}_{<i}^k, y_i, y_{>i}^{k-\frac{1}{2}}), \tilde{z}^k; \tilde{x}^k) + \frac{\sigma}{2} \|y_i - \tilde{y}_i^k\|_{\mathcal{E}_i}^2, \quad i = p, \dots, 2, \\ y_1^k := \arg \min_{y_1} \mathcal{L}_\sigma((y_1, y_{>1}^{k-\frac{1}{2}}), \tilde{z}^k; \tilde{x}^k) + \frac{\sigma}{2} \|y_1 - \tilde{y}_1^k\|_{\mathcal{E}_1}^2, \\ y_i^k := \arg \min_{y_i} \mathcal{L}_\sigma((y_{<i}^k, y_i, y_{>i}^{k-\frac{1}{2}}), \tilde{z}^k; \tilde{x}^k) + \frac{\sigma}{2} \|y_i - \tilde{y}_i^k\|_{\mathcal{E}_i}^2, \quad i = 2, \dots, p. \end{cases} \tag{17}$$

**Step 1.2** *Compute*

$$x^k := \tilde{x}^k - \sigma(\mathcal{A}^*y^k + \mathcal{B}^*\tilde{z}^k - c). \tag{18}$$

**Step 1.3** *Compute*

$$\begin{cases} z_j^{k-\frac{1}{2}} := \arg \min_{z_j} \mathcal{L}_\sigma(y^k, (\tilde{z}_{<j}^k, z_j, z_{>j}^{k-\frac{1}{2}}); x^k) + \frac{\sigma}{2} \|z_j - \tilde{z}_j^k\|_{\mathcal{H}_j}^2, \quad j = q, \dots, 2, \\ z_1^k := \arg \min_{z_1} \mathcal{L}_\sigma(y^k, (z_1, z_{>1}^{k-\frac{1}{2}}); x^k) + \frac{\sigma}{2} \|z_1 - \tilde{z}_1^k\|_{\mathcal{H}_1}^2, \\ z_j^k := \arg \min_{z_j} \mathcal{L}_\sigma(y^k, (z_{<j}^k, z_j, z_{>j}^{k-\frac{1}{2}}); x^k) + \frac{\sigma}{2} \|z_j - \tilde{z}_j^k\|_{\mathcal{H}_j}^2, \quad j = 2, \dots, q. \end{cases} \tag{19}$$

---

Next, denote  $\mathcal{D}_y := \text{Diag}(\mathcal{E}_1 + \sigma^{-1}\mathcal{P}_{11} + \mathcal{A}_1\mathcal{A}_1^*, \dots, \mathcal{E}_p + \sigma^{-1}\mathcal{P}_{pp} + \mathcal{A}_p\mathcal{A}_p^*)$  and  $\mathcal{D}_z := \text{Diag}(\mathcal{H}_1 + \sigma^{-1}\mathcal{Q}_{11} + \mathcal{B}_1\mathcal{B}_1^*, \dots, \mathcal{H}_q + \sigma^{-1}\mathcal{Q}_{qq} + \mathcal{B}_q\mathcal{B}_q^*)$ . Furthermore, define the following linear operators:

$$\mathcal{M}_y := \begin{pmatrix} 0 & \sigma^{-1}\mathcal{P}_{12} + \mathcal{A}_1\mathcal{A}_2^* & \cdots & \sigma^{-1}\mathcal{P}_{1p} + \mathcal{A}_1\mathcal{A}_p^* \\ & \ddots & \cdots & \vdots \\ & & 0 & \sigma^{-1}\mathcal{P}_{(p-1)p} + \mathcal{A}_{p-1}\mathcal{A}_p^* \\ & & & 0 \end{pmatrix}$$

and

$$\mathcal{M}_z := \begin{pmatrix} 0 & \sigma^{-1}\mathcal{Q}_{12} + \mathcal{B}_1\mathcal{B}_2^* & \cdots & \sigma^{-1}\mathcal{Q}_{1q} + \mathcal{B}_1\mathcal{B}_q^* \\ & \ddots & \cdots & \vdots \\ & & 0 & \sigma^{-1}\mathcal{Q}_{(q-1)q} + \mathcal{B}_{q-1}\mathcal{B}_q^* \\ & & & 0 \end{pmatrix}.$$

From Li et al. [22], it is not hard to see that the point  $(z^k, y^k, x^k)$  produced by the procedure (17), (18) and (19) is exactly that generated by (15a)–(15c) with the two positive semidefinite linear operators

$$S = \sigma(\mathcal{M}_y\mathcal{D}_y^{-1}\mathcal{M}_y^* + \mathcal{E}) \quad \text{and} \quad T = \sigma(\mathcal{M}_z\mathcal{D}_z^{-1}\mathcal{M}_z^* + \mathcal{H}),$$

where  $\mathcal{E} = \text{Diag}(\mathcal{E}_1, \dots, \mathcal{E}_p)$ , and  $\mathcal{H} = \text{Diag}(\mathcal{H}_1, \dots, \mathcal{H}_q)$ .

### 4.2 The full Jacobian iteration case

Now we turn our attention to a much general case that there are at least three nonsmooth blocks in the objective function of problem (5), i.e.,

$$f(y) = \sum_{i=1}^p f_i(y_i), \quad \text{and} \quad g(z) = \sum_{j=1}^q g_j(z_j),$$

where  $f_i : \mathcal{Y}_i \rightarrow (-\infty, +\infty]$  and  $g_j : \mathcal{Z}_j \rightarrow (-\infty, +\infty]$  are closed proper convex (probably nonsmooth) functions. The convergence of the multi-block ADMM in a Gauss–Seidel manner for solving such problems under certain further conditions has been considered, e.g. [23,24], but we are interested in removing these conditions, e.g., strong convexity, continuously differentiable with Lipschitz continuous gradients etc.. Let  $\tau_1$  and  $\tau_2$  be two given numbers such that  $\tau_1 \geq p - 1$  and  $\tau_2 \geq q - 1$ . One can choose  $\mathcal{E}_i$  and  $\mathcal{H}_j$  to be self-adjoint positive semidefinite linear operators on  $\mathcal{Y}_i$  and  $\mathcal{Z}_j$  for  $i = 1, \dots, p$  and  $j = 1, \dots, q$ , respectively, such that for any  $1 \leq i \leq p$  and  $1 \leq j \leq q$ ,

$$\mathcal{E}_i \geq \tau_1\mathcal{A}_i\mathcal{A}_i^*, \quad \Sigma_{f_i} + \mathcal{E}_i + \mathcal{A}_i\mathcal{A}_i^* > 0 \quad \text{and} \quad \mathcal{H}_j \geq \tau_2\mathcal{B}_j\mathcal{B}_j^*, \quad \Sigma_{g_j} + \mathcal{H}_j + \mathcal{B}_j\mathcal{B}_j^* > 0.$$

Define  $\mathcal{E} := \text{Diag}(\mathcal{E}_1, \dots, \mathcal{E}_p)$  and  $\mathcal{H} := \text{Diag}(\mathcal{H}_1, \dots, \mathcal{H}_q)$  being two block-diagonal operators and let

$$S := \sigma[(1 + \tau_1)\mathcal{E} - \mathcal{A}\mathcal{A}^*] \quad \text{and} \quad T := \sigma[(1 + \tau_2) - \mathcal{B}\mathcal{B}^*]. \tag{20}$$

It is easy to verify that  $\mathcal{S} \geq 0$  and  $\mathcal{T} \geq 0$ , while it holds that  $\Sigma_f + \mathcal{A}\mathcal{A}^* + \mathcal{E} > 0$  and  $\Sigma_g + \mathcal{B}\mathcal{B}^* + \mathcal{H} > 0$ .

Now we consider the following procedure that generates  $\omega^k = (y^k, z^k, x^k)$  and  $\tilde{\omega}^{k+1}$  in a parallel manner from the given point  $\tilde{\omega}^k = (\tilde{y}^k, \tilde{z}^k, \tilde{x}^k)$ , i.e.,

$$\begin{cases} y_i^k = \arg \min_{x_i \in \mathcal{X}_i} \mathcal{L}_\sigma((\tilde{y}_{<i}^k, y_i, \tilde{y}_{>i}^k), \tilde{z}^k; \tilde{x}^k) + \frac{\sigma}{2} \|y_i - \tilde{y}_i^k\|_{\mathcal{E}_i}^2, & 1 \leq i \leq p, \\ x^k = \tilde{x}^k - \sigma(\mathcal{A}^*y^k + \mathcal{B}^*z^k - c), \\ z_j^k = \arg \min_{z \in \mathcal{Z}_j} \mathcal{L}_\sigma(y^k, (\tilde{z}_{<j}^k, z_j, \tilde{z}_{>j}^k); x^k) + \frac{\sigma}{2} \|z_j - \tilde{z}_j^k\|_{\mathcal{H}_j}^2, & 1 \leq j \leq q, \\ \tilde{\omega}^{k+1} = \tilde{\omega}^k + \rho(\omega^k - \tilde{\omega}^k). \end{cases}$$

It is straightforward to deduce that the above iterative scheme is exactly an instance of the Algorithm 1 with  $\mathcal{S}$  and  $\mathcal{T}$  defined by (20).

### 5 Convergence analysis

In this section, we analyze the convergence of Algorithm 1 step by step. For this purpose, we need the following two basic equalities:

$$2\langle u_1 - u_2, \mathcal{G}(v_1 - v_2) \rangle = \|u_1 - v_2\|_{\mathcal{G}}^2 - \|u_1 - v_1\|_{\mathcal{G}}^2 + \|u_2 - v_1\|_{\mathcal{G}}^2 - \|u_2 - v_2\|_{\mathcal{G}}^2, \tag{21}$$

and

$$2\langle u_1, \mathcal{G}u_2 \rangle = \|u_1\|_{\mathcal{G}}^2 + \|u_2\|_{\mathcal{G}}^2 - \|u_1 - u_2\|_{\mathcal{G}}^2 = \|u_1 + u_2\|_{\mathcal{G}}^2 - \|u_1\|_{\mathcal{G}}^2 - \|u_2\|_{\mathcal{G}}^2, \tag{22}$$

where  $u_1, u_2, v_1$ , and  $v_2$  are vectors in the same finite dimensional real Euclidean space endowed with inner product  $\langle \cdot, \cdot \rangle$  and induced norm  $\| \cdot \|$ , and  $\mathcal{G}$  is an arbitrary self-adjoint positive semidefinite linear operator from that space to itself.

Next, let  $(\bar{x}, \bar{y}, \bar{z}) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$  be an arbitrary solution to the KKT system (9). For any  $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$  we denote  $x_e = x - \bar{x}$ ,  $y_e = y - \bar{y}$  and  $z_e = z - \bar{z}$ . We first have the following result.

**Lemma 5.1** *Let  $(\bar{x}, \bar{y}, \bar{z})$  be a solution to the KKT system (9) and  $\{(x^k, y^k, z^k)\}$  be the sequence generated by Algorithm 1. For any  $k \geq 0$  it holds that*

$$\begin{aligned} &\langle x^{k+1} - x^k, \mathcal{A}^*(y^{k+1} - y^k) \rangle - \frac{\rho}{2} \|y^{k+1} - \tilde{y}^{k+1}\|_{\mathcal{S}}^2 \\ &\quad + \frac{\rho}{2} \|y^k - \tilde{y}^k\|_{\mathcal{S}}^2 \geq \|y^{k+1} - y^k\|_{\Sigma_f}^2, \end{aligned} \tag{23}$$

$$\begin{aligned} &\frac{\sigma\rho}{2} \|\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k\|^2 + \langle x_e^{k+1} + \sigma(1 - \rho)\mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle \\ &= -\frac{1}{2\sigma\rho} \left[ \|x_e^{k+1} + \sigma(1 - \rho)\mathcal{A}^*y_e^{k+1}\|^2 - \|x_e^k + \sigma(1 - \rho)\mathcal{A}^*y_e^k\|^2 \right] \end{aligned} \tag{24}$$

and

$$\begin{aligned} & \sigma(1 - \rho)\langle \mathcal{B}^* z_e^k, \mathcal{A}^* y_e^{k+1} + \mathcal{B}^* z_e^k \rangle + \langle x^k - x^{k+1}, \mathcal{B}^* z_e^k \rangle \\ & - \sigma \langle \mathcal{A}^* y^k + \mathcal{B}^* z^k - c, \mathcal{B}^* z_e^k \rangle \\ & \leq \frac{2 - \rho}{\rho} \left[ -\frac{\rho}{2} \|y^{k+1} - \tilde{y}^{k+1}\|_S^2 + \frac{\rho}{2} \|y^k - \tilde{y}^k\|_S^2 - \|y^{k+1} - y^k\|_{\Sigma_f}^2 \right. \\ & \quad \left. + \frac{\sigma\rho}{2} \|\mathcal{A}^* y_e^k\|^2 - \frac{\sigma\rho}{2} \|\mathcal{A}^* y_e^{k+1}\|^2 - \frac{\sigma(2 - \rho)}{2} \|\mathcal{A}^* y_e^k - \mathcal{A}^* y_e^{k+1}\|^2 \right]. \end{aligned} \tag{25}$$

*Proof* By the first order optimality condition of ((15a)), we get

$$\mathcal{A}\tilde{x}^k - \sigma\mathcal{A}(\mathcal{A}^* y^k + \mathcal{B}^* z^k - c) - \mathcal{S}(y^k - \tilde{y}^k) \in \partial f(y^k). \tag{26}$$

It follows from (15b) that

$$\mathcal{A}x^k - \mathcal{S}(y^k - \tilde{y}^k) \in \partial f(y^k), \quad \text{and} \quad \mathcal{A}x^{k+1} - \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}) \in \partial f(y^{k+1}), \tag{27}$$

which, together with (8), implies

$$\begin{aligned} & \langle x^{k+1} - x^k, \mathcal{A}^*(y^{k+1} - y^k) \rangle - \langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}) - \mathcal{S}(y^k - \tilde{y}^k), y^{k+1} - y^k \rangle \\ & \geq \|y^{k+1} - y^k\|_{\Sigma_f}^2. \end{aligned} \tag{28}$$

From (16), we can easily deduce

$$\begin{aligned} & \langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}) - \mathcal{S}(y^k - \tilde{y}^k), y^{k+1} - y^k \rangle \\ & = \langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}) - \mathcal{S}(y^k - \tilde{y}^k), (y^{k+1} - \tilde{y}^{k+1}) + (\tilde{y}^{k+1} - y^k) \rangle \\ & = \langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}) - \mathcal{S}(y^k - \tilde{y}^k), (y^{k+1} - \tilde{y}^{k+1}) - (1 - \rho)(y^k - \tilde{y}^k) \rangle \\ & = \|y^{k+1} - \tilde{y}^{k+1}\|_S^2 - (\rho - 1)\|y^k - \tilde{y}^k\|_S^2 - (2 - \rho)\langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}), y^k - \tilde{y}^k \rangle \\ & \geq \|y^{k+1} - \tilde{y}^{k+1}\|_S^2 - (\rho - 1)\|y^k - \tilde{y}^k\|_S^2 - \frac{2-\rho}{2} (\|y^{k+1} - \tilde{y}^{k+1}\|_S^2 + \|y^k - \tilde{y}^k\|_S^2) \\ & = \frac{\rho}{2} \|y^{k+1} - \tilde{y}^{k+1}\|_S^2 - \frac{\rho}{2} \|y^k - \tilde{y}^k\|_S^2, \end{aligned} \tag{29}$$

which, together with the inequality (28), implies (23).

Secondly, we get from (15b) and (16) that

$$\tilde{x}^{k+1} = x^k + (\rho - 1)(x^k - \tilde{x}^k) = x^k - \sigma(\rho - 1)(\mathcal{A}^* y^k + \mathcal{B}^* z^k - c). \tag{30}$$

From (15b), we can get

$$\begin{aligned} x^{k+1} & = \tilde{x}^{k+1} - \sigma(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^{k+1} - c), \\ & = x^k - \sigma\rho(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^k - c) - \sigma(\rho - 1)(\mathcal{A}^* y^k - \mathcal{A}^* y^{k+1}). \end{aligned}$$

This implies

$$[x_e^{k+1} + \sigma(1 - \rho)\mathcal{A}^* y_e^{k+1}] - [x_e^k + \sigma(1 - \rho)\mathcal{A}^* y_e^k] = -\sigma\rho(\mathcal{A}^* y_e^{k+1} + \mathcal{B}^* z_e^k). \tag{31}$$

Subsequently, using the relation (31) and the elementary equality (22), we can get (24).

Finally, from (30) we can deduce

$$\begin{aligned} & \sigma(1 - \rho)(\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k) + (x^k - x^{k+1}) - \sigma(\mathcal{A}^*y^k + \mathcal{B}^*z^k - c) \\ &= \sigma(1 - \rho)(\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k) + \sigma\rho(\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k) \\ & \quad + \sigma(\rho - 1)(\mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}) - \sigma(\mathcal{A}^*y_e^k + \mathcal{B}^*z_e^k) \\ &= -\sigma(2 - \rho)(\mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}). \end{aligned} \tag{32}$$

Thus, we have

$$\begin{aligned} & \sigma(1 - \rho)\langle \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k, \mathcal{B}^*z_e^k \rangle + \langle x^k - x^{k+1}, \mathcal{B}^*z_e^k \rangle \\ & \quad - \sigma\langle \mathcal{A}^*y^k + \mathcal{B}^*z^k - c, \mathcal{B}^*z_e^k \rangle \\ &= -\sigma(2 - \rho)\langle \mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}, \mathcal{B}^*z_e^k \rangle \\ &= -(2 - \rho)\sigma\langle \mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle \\ & \quad + (2 - \rho)\sigma\langle \mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} \rangle. \end{aligned} \tag{33}$$

By (31) we have

$$\begin{aligned} & \sigma\langle \mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle \\ &= -\rho^{-1}\langle \mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}, x_e^{k+1} - x_e^k \rangle + \rho^{-1}\sigma(1 - \rho)\|\mathcal{A}^*y_e^{k+1} - \mathcal{A}^*y_e^k\|^2. \end{aligned} \tag{34}$$

On the other hand, by using the elementary equality (22), we can get

$$\sigma\langle \mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} \rangle = \frac{\sigma}{2} \left[ \|\mathcal{A}^*y_e^k\|^2 - \|\mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}\|^2 - \|\mathcal{A}^*y_e^{k+1}\|^2 \right]. \tag{35}$$

Then by substituting (34) and (35) into (33) and using inequality (23) we can get (25). □

In what follows, we will establish a key inequality for analyzing the convergence of Algorithm 1. We define the sequence  $\{\phi_k\}_{k \geq 0}$  by

$$\begin{aligned} \phi_k &:= \frac{1}{\sigma\rho}\|x_e^k + \sigma(1 - \rho)\mathcal{A}^*y_e^k\|^2 + \rho^{-1}[\|\tilde{y}_e^{k+1}\|_S^2 + \|\tilde{z}_e^k\|_T^2] \\ & \quad + (2 - \rho)[\|y^k - \tilde{y}^k\|_S^2 + \sigma\|\mathcal{A}^*y_e^k\|^2]. \end{aligned} \tag{36}$$

**Lemma 5.2** *Suppose Assumption 1 holds. Let the sequence  $\{(x^k, y^k, z^k)\}$  be generated by Algorithm 1. Then for any  $k \geq 0$ , it holds that*

$$\begin{aligned} \phi_k - \phi_{k+1} &\geq 2\|y_e^{k+1}\|_{\Sigma_f}^2 + 2\|z_e^k\|_{\Sigma_g}^2 + \sigma(2 - \rho)\|\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k\|^2 \\ &\quad + (2 - \rho)\|\tilde{y}^{k+1} - y^{k+1}\|_{\mathcal{S}}^2 + (2 - \rho)\|\tilde{z}^k - z^k\|_{\mathcal{T}}^2 \\ &\quad + 2(2 - \rho)\rho^{-1}\|y^{k+1} - y^k\|_{\Sigma_f}^2 + \sigma\rho^{-1}(2 - \rho)^2\|\mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}\|^2. \end{aligned} \tag{37}$$

*Proof* We get from (15c)

$$\mathcal{B}x^{k+1} + (\mathcal{B}x^k - \mathcal{B}x^{k+1}) - \sigma\mathcal{B}(\mathcal{A}^*y^k + \mathcal{B}^*z^k - c) - \mathcal{T}(z^k - \tilde{z}^k) \in \partial g(z^k). \tag{38}$$

By combining (38), (9) and (8) together, one obtains

$$\begin{aligned} \langle x_e^{k+1}, \mathcal{B}^*z_e^k \rangle + \langle x^k - x^{k+1}, \mathcal{B}^*z_e^k \rangle - \langle \mathcal{T}(z^k - \tilde{z}^k), z_e^k \rangle \\ - \sigma \langle \mathcal{A}^*y^k + \mathcal{B}^*z^k - c, \mathcal{B}^*z_e^k \rangle \geq \|z_e^k\|_{\Sigma_g}^2. \end{aligned} \tag{39}$$

On the other hand, by combining (27), (9) and (8) together, we obtain

$$\langle x_e^{k+1}, \mathcal{A}^*y_e^{k+1} \rangle - \langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}), y_e^{k+1} \rangle \geq \|y_e^{k+1}\|_{\Sigma_f}^2. \tag{40}$$

Then by adding (39) to (40) we get

$$\begin{aligned} \langle x_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle + \langle x^k - x^{k+1}, \mathcal{B}^*z_e^k \rangle - \sigma \langle \mathcal{A}^*y^k + \mathcal{B}^*z^k - c, \mathcal{B}^*z_e^k \rangle \\ - \langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}), y_e^{k+1} \rangle - \langle \mathcal{T}(z^k - \tilde{z}^k), z_e^k \rangle \geq \|y_e^{k+1}\|_{\Sigma_f}^2 + \|z_e^k\|_{\Sigma_g}^2. \end{aligned} \tag{41}$$

Note that

$$\begin{aligned} &\langle x_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle \\ &= \langle x_e^{k+1} + \sigma(1 - \rho)\mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle \\ &\quad - \sigma(1 - \rho)\langle \mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle \\ &= \langle x_e^{k+1} + \sigma(1 - \rho)\mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle \\ &\quad - \sigma(1 - \rho)\langle \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle \\ &\quad + \sigma(1 - \rho)\langle \mathcal{B}^*z_e^k, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle. \end{aligned} \tag{42}$$

Consequently, (41) can be reformulated as

$$\begin{aligned} \langle x_e^{k+1} + \sigma(1 - \rho)\mathcal{A}^*y_e^{k+1}, \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k \rangle - \sigma(1 - \rho)\|\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k\|^2 \\ + \sigma(1 - \rho)\langle \mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k, \mathcal{B}^*z_e^k \rangle + \langle x^k - x^{k+1}, \mathcal{B}^*z_e^k \rangle \\ - \sigma \langle \mathcal{A}^*y^k + \mathcal{B}^*z^k - c, \mathcal{B}^*z_e^k \rangle - \langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}), y_e^{k+1} \rangle \\ - \langle \mathcal{T}(z^k - \tilde{z}^k), z_e^k \rangle \geq \|y_e^{k+1}\|_{\Sigma_f}^2 + \|z_e^k\|_{\Sigma_g}^2. \end{aligned} \tag{43}$$

Next, we shall estimate the left-hand-side of (43). Using the relation (16) and the elementary equality (21), we get

$$\langle \mathcal{S}(y^{k+1} - \tilde{y}^{k+1}), y_e^{k+1} \rangle$$

$$= \frac{1}{2\rho} \left[ \|\tilde{y}^{k+2} - \bar{y}\|_{\mathcal{S}}^2 - \|\tilde{y}^{k+1} - \bar{y}\|_{\mathcal{S}}^2 + \rho(2 - \rho)\|\tilde{y}^{k+1} - y^{k+1}\|_{\mathcal{S}}^2 \right], \tag{44}$$

and

$$\langle \mathcal{T}(z^k - \bar{z}^k), z_e^k \rangle = \frac{1}{2\rho} \left[ \|\bar{z}^{k+1} - \bar{z}\|_{\mathcal{T}}^2 - \|\bar{z}^k - \bar{z}\|_{\mathcal{T}}^2 + \rho(2 - \rho)\|\bar{z}^k - z^k\|_{\mathcal{T}}^2 \right]. \tag{45}$$

Substituting (24), (25), (44), and (45) into (43), we get

$$\begin{aligned} & -\frac{1}{\sigma\rho} \left[ \|x_e^{k+1} + \sigma(1 - \rho)\mathcal{A}^*y_e^{k+1}\|^2 - \|x_e^k + \sigma(1 - \rho)\mathcal{A}^*y_e^k\|^2 \right] \\ & -\frac{1}{\rho} \left[ \|\tilde{y}^{k+2} - \bar{y}\|_{\mathcal{S}}^2 - \|\tilde{y}^{k+1} - \bar{y}\|_{\mathcal{S}}^2 \right] - \frac{1}{\rho} \left[ \|\bar{z}^{k+1} - \bar{z}\|_{\mathcal{T}}^2 - \|\bar{z}^k - \bar{z}\|_{\mathcal{T}}^2 \right] \\ & + (2 - \rho)\rho^{-1} \left[ \rho\|y^k - \tilde{y}^k\|_{\mathcal{S}}^2 - \rho\|y^{k+1} - \tilde{y}^{k+1}\|_{\mathcal{S}}^2 - \sigma\rho\|\mathcal{A}^*y_e^{k+1}\|^2 + \sigma\rho\|\mathcal{A}^*y_e^k\|^2 \right] \\ & \geq 2\|y_e^{k+1}\|_{\Sigma_f}^2 + 2\|z_e^k\|_{\Sigma_g}^2 + \sigma(2 - \rho)\|\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k\|^2 \\ & \quad + (2 - \rho)\|\tilde{y}^{k+1} - y^{k+1}\|_{\mathcal{S}}^2 + (2 - \rho)\|\bar{z}^k - z^k\|_{\mathcal{T}}^2 \\ & \quad + 2(2 - \rho)\rho^{-1}\|y^{k+1} - y^k\|_{\Sigma_f}^2 + \sigma\rho^{-1}(2 - \rho)^2\|\mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}\|^2. \end{aligned} \tag{46}$$

Hence, we get (37) by the definition of  $\phi_k$  and complete the proof. □

*Remark 5.1* The above proof looks like to those in [9] and [3] since the basic tools for proving all these theoretical results are closely based on an earlier work of Fortin and Glowinski[10]. However, the most important steps of this paper are quite different from the previous ones. An explicit example is the definition of  $\phi_k$  in (37), which originally comes from (42), that plays the key role to get (46). The motivation is that we want to decompose  $x_e^{k+1}$  so that there is not any inner product terms in (37). Therefore, we need some elementary but careful calculations, which are not previously conducted.

Now, we are ready to establish the global convergence of Algorithm 1.

**Theorem 5.1** *Suppose Assumption 1 holds. Let  $\{(x^k, y^k, z^k)\}$  be the sequence generated by Algorithm 1. Then the whole sequence  $\{(x^k, y^k, z^k)\}$  converges to a solution to the KKT system (9).*

*Proof* Note that  $\rho \in (0, 2)$ . We clearly see from (37) that  $\{\phi_k\}_{k \geq 0}$  is a nonnegative and monotonically non-increasing sequence. Hence,  $\{\phi_k\}$  is also bounded. As a result, the following sequences are bounded:

$$\{\|x_e^k + \sigma(1 - \rho)\mathcal{A}^*y_e^k\|\}, \{\|\tilde{y}^{k+1}\|_{\mathcal{S}}\}, \{\|\bar{z}^k\|_{\mathcal{T}}\}, \{\|y^k - \tilde{y}^k\|_{\mathcal{S}}\}, \text{ and } \{\|\mathcal{A}^*y_e^k\|\}. \tag{47}$$

Moreover, from the inequality (37) we have as  $k \rightarrow \infty$ ,

$$\begin{aligned} & \|y_e^{k+1}\|_{\Sigma_f} \rightarrow 0, \|z_e^k\|_{\Sigma_g} \rightarrow 0, \|\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k\| \rightarrow 0, \|\tilde{y}^{k+1} - y^{k+1}\|_{\mathcal{S}} \rightarrow 0, \\ & \|\bar{z}^k - z^k\|_{\mathcal{T}} \rightarrow 0, \|y^{k+1} - y^k\|_{\Sigma_f} \rightarrow 0, \text{ and } \|\mathcal{A}^*y_e^k - \mathcal{A}^*y_e^{k+1}\| \rightarrow 0. \end{aligned} \tag{48}$$

Thus, by the fact  $\|y^k\|_{\mathcal{S}} \leq \|y^k - \tilde{y}^k\|_{\mathcal{S}} + \|\tilde{y}^k\|_{\mathcal{S}}$  we can see that  $\{\|y^k\|_{\mathcal{S}}\}$  is also bounded. Consequently, the sequence  $\{\|y^k\|_{\Sigma_f + \mathcal{S} + \mathcal{A}^*}\}$  is bounded. Since  $\Sigma_f +$

$\mathcal{S} + \mathcal{A}\mathcal{A}^* > 0$ , the sequence  $\{\|y^k\|\}$  is bounded. Similarly, the sequences  $\{\|z^k\|_{\mathcal{T}}\}$  and  $\{\|\mathcal{B}^*z^k\|\}$  are both bounded and so is the sequence  $\{\|z^k\|_{\Sigma_g + \mathcal{T} + \mathcal{B}\mathcal{B}^*}\}$ . It implies that the sequence  $\{\|z^k\|\}$  is bounded from  $\Sigma_g + \mathcal{T} + \mathcal{B}\mathcal{B}^* > 0$ . The boundedness of  $\{\|x_e^k + \sigma(1 - \rho)\mathcal{A}^*y_e^k\|\}$  and  $\{\|y^k\|\}$  further indicate that the sequence  $\{\|x^k\|\}$  is bounded. The above arguments have shown that  $\{(x^k, y^k, z^k)\}$  is a bounded sequence.

Consequently, the sequence  $\{(x^k, y^k, z^k)\}$  admits at least one convergent subsequence. Suppose that  $\{(x^{k_i}, y^{k_i}, z^{k_i})\}$  is a subsequence of  $\{(x^k, y^k, z^k)\}$  converging to  $(x^\infty, y^\infty, z^\infty) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ . It follows from (27) and (38) that

$$\begin{cases} \mathcal{A}x^{k_i} - \mathcal{S}(y^{k_i} - \tilde{y}^{k_i}) \in \partial f(y^{k_i}), \\ \mathcal{B}x^{k_i} - \sigma\mathcal{B}(\mathcal{A}^*y^{k_i} + \mathcal{B}^*z^{k_i} - c) - \mathcal{T}(z^{k_i} - \tilde{z}^{k_i}) \in \partial g(z^{k_i}). \end{cases} \tag{49}$$

We also have from (48) that  $\lim_{k \rightarrow \infty} (\mathcal{A}^*y^k + \mathcal{B}^*z^k - c) = 0$ . Taking limits in (49) and using (48), one obtains

$$\mathcal{A}x^\infty \in \partial f(y^\infty), \quad \mathcal{B}x^\infty \in \partial g(z^\infty) \quad \text{and} \quad \mathcal{A}^*y^\infty + \mathcal{B}^*z^\infty - c = 0,$$

which indicates that  $(x^\infty, y^\infty, z^\infty)$  is a solution to the KKT system (9).

Next we need to show that  $(x^\infty, y^\infty, z^\infty)$  is the unique limit point of the sequence  $\{(x^k, y^k, z^k)\}$ . Without loss of generality we can let  $(\bar{x}, \bar{y}, \bar{z}) = (x^\infty, y^\infty, z^\infty)$ . Consequently, the sequence  $\{\phi_k\}$  defined by (36) has a subsequence converges to zero. Hence  $\{\phi_k\}$  itself converging to zero. By the definition of  $\phi_k$  we have  $\lim_{k \rightarrow \infty} x^k = \bar{x}$ . Moreover, from  $\|\tilde{y}^{k+1} - y^{k+1}\|_{\mathcal{S}} \rightarrow 0$  in (48), it is easy to show that  $\|y_e^k\|_{\mathcal{S}} \rightarrow 0$  as  $k \rightarrow \infty$ . Noting that  $\|\mathcal{A}^*y_e^k\| \rightarrow 0$  in (47) and  $\|y_e^k\|_{\Sigma_f} \rightarrow 0$  in (48), we have  $\{\|y_e^k\|_{\Sigma_f} + \|y_e^k\|_{\mathcal{S}} + \|\mathcal{A}^*y_e^k\|\} \rightarrow 0$  as  $k \rightarrow \infty$ . Hence, we have  $\lim_{k \rightarrow \infty} y^k = \bar{y}$  since  $\Sigma_f + \mathcal{S} + \mathcal{A}\mathcal{A}^* > 0$ . Finally, from the fact  $\|\mathcal{A}^*y_e^{k+1}\| \rightarrow 0$ ,  $\|\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k\| \rightarrow 0$  in (48), and

$$\|\mathcal{B}^*z_e^k\| \leq \|\mathcal{A}^*y_e^{k+1} + \mathcal{B}^*z_e^k\| + \|\mathcal{A}^*y_e^{k+1}\|,$$

we get  $\|\mathcal{B}^*z_e^k\| \rightarrow 0$ . Since  $\|z_e^k\|_{\mathcal{T}} \rightarrow 0$  and  $\|z_e^k\|_{\Sigma_g} \rightarrow 0$  by (48), we have  $\{\|z_e^k\|_{\Sigma_g} + \|z_e^k\|_{\mathcal{T}} + \|\mathcal{B}^*z_e^k\|\} \rightarrow 0$  as  $k \rightarrow \infty$ . Therefore, from the fact  $\Sigma_g + \mathcal{T} + \mathcal{B}\mathcal{B}^* > 0$  we know that  $\lim_{k \rightarrow \infty} z^k = \bar{z}$ . This completes the proof.  $\square$

### 6 Numerical experiments

In this section, we apply the generalized ADMM with semi-proximal terms (Algorithm 1) proposed in this paper together with the block sGS techniques (Algorithm 2) to solve convex DNN-SDP problem via its dual. Specifically, the DNN-SDP problem is represented by

$$\begin{aligned} & \min_X \langle C, X \rangle \\ & \text{s.t. } \mathcal{A}_E X = b_E, \mathcal{A}_I X \geq b_I, X \in \mathcal{S}_+^n \cap \mathcal{N}, \end{aligned} \tag{50}$$



where  $\mathcal{S}^n$  is the space of  $n \times n$  symmetric matrices,  $\mathcal{A}_E : \mathcal{S}^n \rightarrow \mathfrak{R}^{m_E}$  and  $\mathcal{A}_I : \mathcal{S}^n \rightarrow \mathfrak{R}^{m_I}$  are two linear maps,  $\mathcal{S}_+^n$  is the closed convex cone of  $n \times n$  symmetric and positive semidefinite matrices,  $\mathcal{N}$  is a nonempty simple closed convex set (e.g.,  $\mathcal{N} = \{X \in \mathcal{S}^n \mid L \leq X \leq U\}$  with  $L, U \in \mathcal{S}^n$  being given matrices),  $C \in \mathcal{S}^n$ ,  $b_E \in \mathfrak{R}^{m_E}$  and  $b_I \in \mathfrak{R}^{m_I}$  are given data.

The dual of problem (50) then takes the following form

$$\begin{aligned} \max_{S, Z, y_E, y_I} & -\delta_{\mathcal{N}}^*(-Z) + \langle b_E, y_E \rangle + \langle b_I, y_I \rangle \\ \text{s.t.} & Z + S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I = C, \quad S \in \mathcal{S}_+^n, \quad y_I \geq 0, \end{aligned} \tag{51}$$

which is an instance of multi-block composite conic optimization problems. In order to handle the inequality constraints in (51) so that it can be solved by the GADMM algorithm together with the sGS technique, we introduce a slack variable  $v$  and a positive definite nonsingular linear operator  $\mathcal{D}^* : \mathcal{Y} \rightarrow \mathcal{Y}$  to get the following equivalent optimization problem:

$$\begin{aligned} \min_{S, Z, y_E, y_I, v} & (\delta_{\mathcal{N}}^*(-Z) + \delta_{\mathfrak{R}^{m_I}}(v)) + \delta_{\mathcal{S}_+^n}(S) - \langle b_E, y_E \rangle - \langle b_I, y_I \rangle \\ \text{s.t.} & Z + S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I = C, \quad \mathcal{D}^*(v - y_I) = 0. \end{aligned} \tag{52}$$

Therefore, the DNN-SDP (50) can be solved by the GADMM algorithm via its dual form (52), together with the sGS techniques that were illustrated in Algorithm 2.

We use  $X \in \mathcal{S}^n$  and  $u \in \mathfrak{R}^{m_I}$  to denote the dual variables corresponding to the two groups of equality constraints in (52), respectively. Moreover, we denote the normal cone of  $\mathcal{N}$  at  $X$  by  $N_{\mathcal{N}}(X)$ . Then, the KKT system of problem (50) is given by

$$\begin{cases} \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I + S + Z - C = 0, \quad \mathcal{A}_E X - b_E = 0, \\ 0 \in N_{\mathcal{N}}(X) + Z, \quad X \in \mathcal{S}_+^n, \quad S \in \mathcal{S}_+^n, \quad \langle X, S \rangle = 0, \\ \mathcal{A}_I X - b_I \geq 0, \quad y_I \geq 0, \quad \langle \mathcal{A}_I X - b_I, y_I \rangle = 0. \end{cases} \tag{53}$$

Based on the optimality condition (53), we measure the accuracy of a computed candidate solution  $(X, Z, S, y_E, y_I)$  for (50) and its dual (51) via

$$\eta_{SDP} = \max\{\eta_D, \eta_X, \eta_Z, \eta_P, \eta_S, \eta_I\},$$

where

$$\begin{aligned} \eta_D &= \frac{\|\mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I + S + Z - C\|}{1 + \|C\|}, \quad \eta_X = \frac{\|X - \Pi_{\mathcal{N}}(X)\|}{1 + \|X\|}, \quad \eta_Z = \frac{\|X - \Pi_{\mathcal{N}}(X - Z)\|}{1 + \|X\| + \|Z\|}, \\ \eta_P &= \frac{\|\mathcal{A}_E X - b_E\|}{1 + \|b_E\|}, \quad \eta_S = \max\left\{ \frac{\|X - \Pi_{\mathcal{S}_+^n}(X)\|}{1 + \|X\|}, \frac{|(X, S)|}{1 + \|X\| + \|S\|} \right\}, \\ \eta_I &= \max\left\{ \frac{\|\min(0, y_I)\|}{1 + \|y_I\|}, \frac{\|\min(0, \mathcal{A}_I X - b_I)\|}{1 + \|b_I\|}, \frac{|(\mathcal{A}_I X - b_I, y_I)|}{1 + \|\mathcal{A}_I X - b_I\| + \|y_I\|} \right\}. \end{aligned} \tag{54}$$

Next, we would like to report our numerical results.

## 6.1 Numerical results on DNN-SDP with many inequality constraints

In our numerical experiments, we first test DNN-SDP problems from the relaxation of a binary integer quadratic (BIQ) programming, with a large number of valid inequalities introduced by Sun et. al. [33] to get tighter bounds, i.e.,

$$\begin{aligned} & \min \frac{1}{2} \langle Q, \bar{X} \rangle + \langle c, x \rangle \\ & \text{s.t. } \text{diag}(\bar{X}) - x = 0, \quad X = \begin{pmatrix} \bar{X} & x \\ x^T & 1 \end{pmatrix} \in \mathcal{S}_+^n, \quad X \geq 0, \\ & \quad x_i - \bar{X}_{ij} \geq 0, \quad x_j - \bar{X}_{ij} \geq 0, \quad \bar{X}_{ij} - x_i - x_j \geq -1 \quad \forall i < j, \quad j = 2, \dots, n-1. \end{aligned}$$

In the above problem,  $\text{diag}(\bar{X})$  denotes the vector consisted of the diagonal elements of the  $(n-1) \times (n-1)$  matrix  $\bar{X}$ . In our test, the data for  $Q$  and  $c$  are taken from the Biq Mac Library<sup>2</sup> maintained by Wiegele. The linear operator  $\mathcal{D}$  in (52) is set by  $\mathcal{D} := \alpha \mathcal{I}$ . By writing down the augmented Lagrangian function of problem (52), one can observe that the quadratic (penalty) terms in it takes the form of

$$\frac{\sigma}{2} \|Z + S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I - C\|^2 + \frac{\sigma \alpha^2}{2} \|v - y_I\|^2.$$

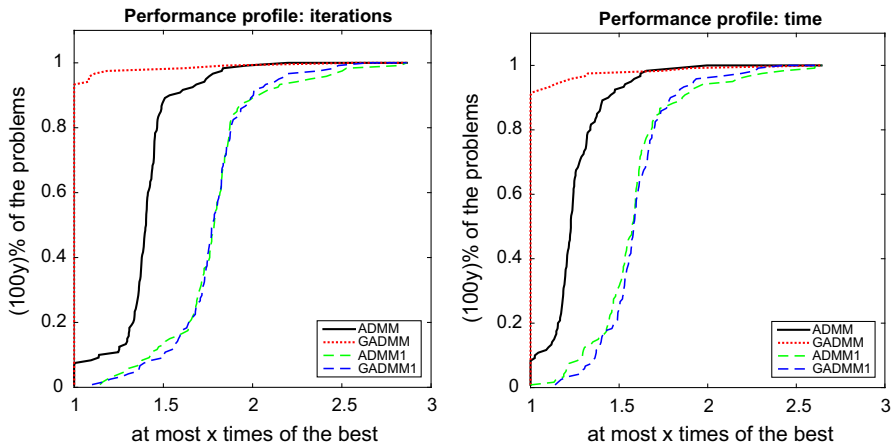
Therefore, we can set choose  $\alpha = \sqrt{\|\mathcal{A}_k\|}/2$  so that, numerically, the second term above is relatively stable with respect to the previous one. Of course, one can choose other numbers. By properly choosing the proximal terms via the techniques provided in [2, Sect. 7.1], the corresponding subproblems will have closed-form solutions. We mention that all these algorithms are tested by running MATLAB on a HP Elitedesk with one Intel Core i7-4770S Processor (4 Cores, 8 Threads, 8M Cache, 3.1 to 3.9 GHz) and 8 GB (DDR3-1600 MHz) RAM.

We compare the performance of GADMM with the relaxation parameter  $\rho$  ranging from 1 to 1.9. The reason is that, on the one hand, during a preliminary test on a few number of problems, we found that the case  $\rho < 1$  is not interesting in the sense that it is almost no better than  $\rho = 1$ , and, on the other hand, the design of the algorithm in this paper is intended to use a relatively large  $\rho$  to get better numerical performance. Besides, the performance of the sGS based semi-proximal ADMM from [19, 33] with step length of 1 and 1.618<sup>3</sup> is also reported to serve as the benchmark for comparison.

Table 1 presents the computational results of all the 12 algorithms, i.e. GADMM with  $\rho = 1, \rho = 1.1, \dots, \rho = 1.9$  and sPADMM with  $\tau = 1$  and  $\tau = 1.618$ . All the algorithms are terminated if  $\eta_{SDP} < 10^{-6}$ , or the maximum iteration number 500,000 is achieved. We have not presented  $\eta_{SDP}$  for each computed instance in this table since almost all the problems are solved to the required accuracy by each of the above mentioned algorithms. We should mention that the GADMM with  $\rho = 1$  (denoted by GADMM1) and the sPADMM with  $\tau = 1$  (denoted by sPADMM1) are essentially the

<sup>2</sup> <http://biqmac.uni-klu.ac.at/biqmaclib.html>.

<sup>3</sup> In fact, we have applied the technique discussed in [3, 33] for using a step-length which can sometimes larger than 1.618 to further improve the computational efficiency. However, we still call the step-length is 1.618, just for convenience.



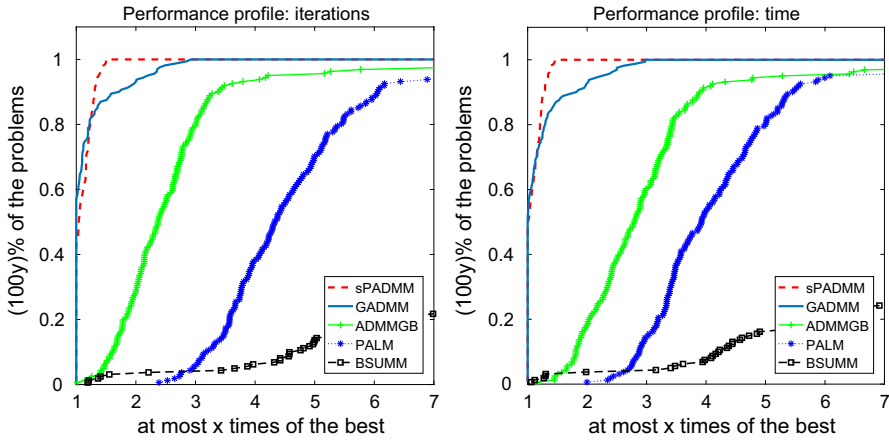
**Fig. 1** The performance profile of ADMM, GADMM, ADMM1 and GADMM1 on solving DNN-SDP problems with many inequality constraints

same algorithm, even if the implementations of them are slightly different. As one can observe from Table 1, the total computational time of GADMM1 and sPADMM1 are also similar, while GADMM with  $\rho = 1.8$  has the minimum computational time. One can also see that for a certain portion of problems, GADMM with  $\rho = 1.9$  performs the best, but there are some problems for which this relaxation parameter is too large to provide a satisfactory solution within the prescribed maximum iteration number. Therefore, generally speaking, the merit of large relaxation parameter  $\rho$  is apparent, but one should avoid of the risk that brought by too large  $\rho$ . Therefore, we would like to suggest that  $\rho = 1.8$  is a desirable choice.

Figure 1 presents the performance profile of four algorithms (GADMM with  $\rho = 1$  and  $\rho = 1.8$ , denoted by GADMM1 and GADMM, respectively, and sPADMM with  $\tau = 1$  and  $\tau = 1.618$ , denoted by sPADMM1 and sPADMM, respectively). From Fig. 1 one can see that GADMM1 and sPADMM1 have very similar performance when solving these problems and GADMM performs the best. As a consequence, in the subsequent section, we would like to test the performance of GADMM with  $\rho = 1.8$  only for DNN-SDP problems without linear inequality constraints, and compare it with other algorithms. Therefore, it is obvious that both the relaxation step of GADMM and large step-length of sPADMM can enhance the numerical performance of sPADMM with unit step-length and the GADMM has the probability to perform even better.

## 6.2 Numerical results on DNN-SDP without inequality constraints

Next, we report our numerical results on solving DNN-SDP problems without inequality constraints see the performance of the proposed generalized ADMM. In this part, we fix  $\rho$  for GADMM as  $\rho = 1.8$  since the numerical results reported in the previous subsection suggests that this relaxation parameter performs pretty good. The detailed



**Fig. 2** The performance profile of sPADMM, GADMM, ADMMGB, PALM and BSUMM on solving DNN-SDP problems without inequality constraints

description of the 281 instances of DNN-SDP problems can be found in Sun et al. [33, Section 5.1.1].

For comparison with other state-of-the-art algorithms, we also tested the sGS based sPADMM [19, 33] with  $\tau = 1.618$ , the ADMM with Gaussian back substitution [16] by He et al. (denoted by ADMMGB), the parallel multi-block ADMM by Deng et al. [5] (denoted by PALM), and the block successive upper bound minimization method of multipliers [18] by Hong et al. (denoted by BSUMM). We mention that all the solvers are tested by running MATLAB on an Apple Macbook Air Laptop with one Intel Core i7-5650U Processor (2 Cores, 4 Threads, 4M Cache, 2.2 to 3.1 GHz) and 8 GB (DDR3-1600 MHz) RAM.

For ADMMGB, we set the back substitution parameter as 0.99. For PALM, we used the proximal-linear type (c.f. [5, Theorem 2.1]) proximal terms while the adaptive parameter tuning strategy in [5, Sect. 3] was not used. The reason is that, during a preliminary test, we found that the improvement is very limited but sometimes it can dramatically increase the iteration numbers. For BSUMM, it is very hard to provide a uniform way to choose the step-length. Therefore, although we have tried our best to find a strategy of providing the step-length, the computational results are not promising. All the tested algorithms are terminated if  $\eta_{SDP} < 10^{-6}$ , where

$$\eta_{SDP} := \max\{\eta_D, \eta_X, \eta_Z, \eta_P, \eta_S\},$$

with  $\eta_X, \eta_Z, \eta_P, \eta_S$  being defined in (54) and

$$\eta_D := \frac{\|\mathcal{A}_E^* y_E + S + Z - C\|}{1 + \|C\|},$$

or the maximum iteration number 500,000 is achieved.

Table 2 presents the computational results of sPADMM, GADMM, ADMMGB, PALM and BSUMM. The corresponding performance profiles are presented in Fig. 2. As one can observe that, the sPADMM performs the best and the GADMM proposed in this paper has similar performances for about 80% of the tested problems. However, both sPADMM and GADMM are far superior than ADMMGB, PALM and BSUMM. The reason behind the performance profile can be described as follows. Both sPADMM and GADMM used certain over-relaxation techniques and the sGS techniques for generating relatively small semi-proximal terms. The ADMMGB does not rely on proximal terms, but it can take advantage of some over-relaxation techniques. For PALM, the proximal term is much larger than other algorithms so that its efficiency is even worse than ADMMGB. Finally, the requirements for sufficiently small step-length plays the key role of making BSUMM inefficient, especially for solving problems that can not be handled by first-order methods within a small number of iterations.

## 7 Conclusions

In a nutshell, this paper provides the convergence analysis for a variant of generalized ADMM with semi-proximal terms and applies it to a class of multi-block convex composite conic optimization problems. The method studied in this paper relaxes both the primal and the dual variables simultaneously and, more importantly, has the advantage of solving multi-block problems by using the techniques discussed in Sect. 4. The semi-proximal terms for the generalized ADMM considered in this paper is more natural than the other variant since the most recent values of variables are always used in the semi-proximal terms. Numerically, we conducted a comparison between the proposed algorithm and four state-of-the-art first order solvers. The computational results suggest that the proposed method performs similar to the sPADMM and superior than the rest. Therefore, this paper not only provided an efficient algorithm, but also can help to get clear the performance of several first-order methods, at least for DNN-SDP problems. We should mention that although there are many solvers [25–27, 33–35] for solving the standard linear SDP problems, we only choose the block sGS based semi-Proximal ADMM for comparison as it has been proven in [33] to be the most efficient ADMM-type method for solving DNN-SDP considered here. Moreover, solving sub-problems inexactly can also improve the performance of ADMM-type algorithms, as was observed in [2]. Therefore, it is of interest to combine the relaxation scheme proposed in this paper and the techniques of using approximate solutions of subproblems, and we leave it as our future work.

**Acknowledgements** We would like to thank the anonymous referees and the associate editor for their useful comments and suggestions which improved this paper greatly. We are very grateful to Professor Defeng Sun at the Hong Kong Polytechnic University for sharing his knowledge with us on topics covered in this paper and beyond. The research of Y. Xiao and L. Chen was supported by the China Scholarship Council while they were visiting the National University of Singapore. The research of Y. Xiao was supported by the Major State Basic Research Development Program of China (973 Program) (Grant No. 2015CB856003), and the National Natural Science Foundation of China (Grant No. 11471101). The research of L. Chen was supported by the Fundamental Research Funds for Central Universities and the National Natural Science

Foundation of China (Grant No. 11271117). The research of D. Li was supported by the National Natural Science Foundation of China (Grant No. 11371154 and 11771157).

## References

1. Chen, C.H.: Numerical algorithms for a class of matrix norm approximation problems. Ph.D. Thesis, Department of Mathematics, Nanjing University, Nanjing, China. [http://www.math.nus.edu.sg/~matsundf/Thesis\\_Caihua.pdf](http://www.math.nus.edu.sg/~matsundf/Thesis_Caihua.pdf) (2012)
2. Chen, L., Sun, D.F., Toh, K.-C.: An efficient inexact symmetric Gauss–Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Math. Program.* **161**(1), 237–270 (2017)
3. Chen, L., Sun, D.F., Toh, K.-C.: A note on the convergence of ADMM for linearly constrained convex optimization Problems. *Comput. Optim. Appl.* **66**(2), 327–343 (2017)
4. Cui, Y., Li, X.D., Sun, D.F., Toh, K.-C.: On the convergence properties of a majorized ADMM for linearly constrained convex optimization problems with coupled objective functions. *J. Optim. Theory Appl.* **169**(3), 1013–1041 (2016)
5. Deng, W., Lai, M.-J., Peng, Z., W, Yin: Parallel multi-block ADMM with  $o(1/k)$  convergence. *J. Sci. Comput.* **71**(2), 712–736 (2017)
6. Eckstein, J.: Some saddle-function splitting methods for convex programming. *Optim. Methods Softw.* **4**, 75–83 (1994)
7. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **55**, 293–318 (1992)
8. Eckstein, J., Yao, W.: Understanding the convergence of the alternating direction method of multipliers: theoretical and computational perspectives. *Pac. J. Optim.* **11**(4), 619–644 (2014)
9. Fazel, M., Pong, T.K., Sun, D.F., Tseng, P.: Hankel matrix rank minimization with applications in system identification and realization. *SIAM J. Matrix Anal. Appl.* **34**, 946–977 (2013)
10. Fortin, M., Glowinski, R.: Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems. *Studies in mathematics and its applications*, vol. 15. (translated from French by Hunt, B. and Spicer, D.C.) Elsevier Science Publishers B.V. (1983)
11. Gabay, D.: Studies in mathematics and its applications. In: Fortin, M., Glowinski, R. (eds.) *Applications of the method of multipliers to variational inequalities in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, vol. 15, pp. 299–331. Elsevier, Amsterdam (1983)
12. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.* **2**, 17–40 (1976)
13. Glowinski, R.: *Lectures on numerical methods for non-linear variational problems*. Published for the Tata Institute of Fundamental Research, Bombay [by] Springer (1980)
14. Glowinski, R. and Marroco, A.: Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *Revue française d’atomatique, Informatique Recherche Opérationnelle. Analyse Numérique*, **9**(2), 41–76 (1975)
15. Glowinski, R.: Modeling, simulation and optimization for science and technology. In: Fitzgibbon, W., Kuznetsov, Y.A., Neittaanmaki, P., Pironneau, O. (eds.) *On alternating direction methods of multipliers: A historical perspective*, pp. 59–82. Springer, Netherlands (2014)
16. He, B., Tao, M., Yuan, X.: Alternating direction method with Gaussian back substitution for separable convex programming. *SIAM J. Optim.* **22**(2), 313–340 (2012)
17. Hestenes, M.: Multiplier and gradient methods. *J. Optim. Theory Appl.* **4**(5), 303–320 (1969)
18. Hong, M., Chang, T.-H., Wang, X., Razaviyayn, M., Ma, S. and Luo, Z.-Q.: A block successive upper bound minimization method of multipliers for linearly constrained convex optimization. [arXiv:1401.7079](https://arxiv.org/abs/1401.7079) (2014)
19. Li, X.D., Sun, D.F., Toh, K.-C.: QSDPNAL: A two-phase Newton-CG proximal augmented Lagrangian method for convex quadratic semidefinite programming problems, [arXiv:1512.08872](https://arxiv.org/abs/1512.08872) (2015)
20. Li, X.D.: A two-phase augmented Lagrangian method for convex composite quadratic programming, PhD Thesis, Department of Mathematics, National University of Singapore (2015)
21. Li, M., Sun, D.F., Toh, K.-C.: A majorized ADMM with indefinite proximal terms for linearly constrained convex composite optimization. *SIAM J. Optim.* **26**, 922–950 (2016)
22. Li, X.D., Sun, D.F., Toh, K.-C.: A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions. *Math. Program.* **155**, 333–373 (2016)

23. Lin, M., Ma, S.Q., Zhang, S.Z.: On the global linear convergence of the ADMM with multi-block variables. *SIAM J. Optim.* **25**(3), 1478–1497 (2015)
24. Lin, M., Ma, S.Q., Zhang, S.Z.: Iteration complexity analysis of multi-block ADMM for a family of convex minimization without strong convexity. *J. Sci. Comput.* **69**, 52–81 (2016)
25. Monteiro, R.D.C., Ortiz, C., Svaiter, B.F.: A first-order block-decomposition method for solving two-easy-block structured semidefinite programs. *Math. Program. Comput.* **6**, 103–150 (2014)
26. 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.* **57**, 45–69 (2014)
27. Povh, J., Rendl, F., Wiegale, A.: A boundary point method to solve semidefinite programs. *Computing* **78**, 277–286 (2006)
28. Powell, M.J.D.: Optimization. In: Fletcher, R. (ed.) *A method for nonlinear constraints in minimization problems*, pp. 283–298. Academic Press, London (1969)
29. Rockafellar, R.T.: *Convex Analysis*. Princeton University Press, Princeton (1970)
30. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.* **1**(2), 97–116 (1976)
31. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* **14**(5), 877–898 (1976)
32. Rockafellar, R.T.: Monotone operators and augmented lagrangian methods in nonlinear programming. In: Mangasarian, O.L., Meyer, R.M., Robinson, S.M. (eds.) *Nonlinear Programming 3*, pp. 1–25. Academic Press, New York (1977)
33. Sun, D.F., Toh, K.-C., Yang, L.: A convergent 3-block semi-proximal alternating direction method of multipliers for conic programming with 4-type constraints. *SIAM J. Optim.* **25**, 882–915 (2015)
34. Wen, Z., Goldfarb, D., Yin, W.: Alternating direction augmented Lagrangian methods for semidefinite programming. *Math. Program. Comput.* **2**, 203–230 (2010)
35. Zhao, X.Y., Sun, D.F., Toh, K.-C.: A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM J. Optim.* **20**, 1737–1765 (2010)