Journal of Scientific Computing (2023) 97:3
https://doi.org/10.1007/s10915-023-02302-6

# Proximal Gradient/Semismooth Newton Methods for Projection onto a Polyhedron via the Duality-Gap-Active-Set Strategy

**Yunlong Wang[1] · Chungen Shen[2] · Lei-Hong Zhang[3] · Wei Hong Yang[4]**

Received: 9 November 2022 / Revised: 30 June 2023 / Accepted: 12 July 2023 /
Published online: 14 August 2023


**Abstract**
The polyhedral projection problem arises in various applications. To efficiently solve the dual problem, one of the crucial issues is to safely identify zero-elements as well as the signs of nonzero elements at the optimal solution. In this paper, relying on its nonsmooth dual problem and active set techniques, we first propose a Duality-Gap-Active-Set strategy (DGASS) to effectively identify the indices of zero-elements and the signs of nonzero entries of the optimal solution. Serving as an efficient acceleration strategy, DGASS can be embedded into certain iterative methods. In particular, by applying DGASS to both the proximal gradient algorithm (PGA) and the proximal semismooth Newton algorithm (PSNA), we propose the method of PGA-DGASS and PSNA-DGASS, respectively. Global convergence and local quadratic convergence rate are discussed. We report on numerical results over both synthetic and real data sets to demonstrate the high efficiency of the two DGASS-accelerated methods.

The work of the first author was supported in part by the National Natural Science Foundation of China NSFC-72025201.
The work of the third author was supported in part by the National Natural Science Foundation of China NSFC-12071332.
The work of the last author was supported by the National Natural Science Foundation of China NSFC-11971118.

✉ Chungen Shen
shenchungen@usst.edu.cn

Yunlong Wang
wylwork_sjtu@sjtu.edu.cn

Lei-Hong Zhang
longzlh@suda.edu.cn

Wei Hong Yang
whyang@fudan.edu.cn

1 Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China

2 College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China

3 School of Mathematical Sciences, Soochow University, Suzhou 215006, China

4 School of Mathematical Sciences, Fudan University, Shanghai 200433, China

**Keywords** Polyhedral projection problem · Active set · Duality gap · Proximal gradient algorithm · Proximal semismooth Newton method

## 1 Introduction

In this paper, we consider the polyhedral projection problem of computing the closest point $x$ in a polyhedron $\Omega$ to a given $y \in \mathbb{R}^n$ (see [23]):

$$\min \left\{ \frac{1}{2}\|y - x\|^2 : x \in \Omega \right\} \tag{1.1}$$

where $\Omega := \{x \in \mathbb{R}^n : l \le Ax \le u, x \in \mathcal{X}\}$, $A = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^{m\times n}$ with $0 \ne a_i \in \mathbb{R}^m$, $i \in [n] := \{1, 2, \ldots, n\}$, $l$ and $u \in \mathbb{R}^m$ ($l < u$) are finite lower and upper bounds for $Ax$, respectively, and $\mathcal{X}$:=$\{x \in \mathbb{R}^n : \underline{b} \le x \le \bar{b}\}$ with $\underline{b}$ and $\bar{b} \in \mathbb{R}^n$ ($\underline{b} < \bar{b}$). Throughout the paper, we assume $\Omega \ne \emptyset$. The polyhedral projection problem (1.1) is fundamental in many applications, and we mention [8, 9, 17, 22, 28, 47, 49, 50] for only a few of them, and refer to [23, 50] and the references therein for more others.

### 1.1 The Dual Problem and Previous Works

By introducing an additional variable $z \in \mathbb{R}^m$, we can rewrite (1.1) as

$$\min_{x,z} \left\{ \frac{1}{2}\|y - x\|^2 : Ax = z,\ l \le z \le u,\ x \in \mathcal{X},\ z \in \mathbb{R}^m \right\}. \tag{1.2}$$

Note that the Lagrange function associated with (1.2) is

$$\mathcal{L}(x, z; \lambda) = \frac{1}{2}\|y - x\|^2 + \lambda^T(z - Ax),$$

where $\lambda \in \mathbb{R}^n$ denotes the Lagrange multiplier corresponding to the equality constraints $Ax = z$. Since $x$ and $z$ are separable in the Lagrange function, minimizing $\mathcal{L}(x, z; \lambda)$ with respect to $x$ and $z$ leads to the dual problem

$$\max\{L(\lambda) : \lambda \in \mathbb{R}^m\}, \tag{1.3}$$

where

$$L(\lambda) = \min_{x,z}\{\mathcal{L}(x, z; \lambda) :\ l \le z \le u,\ x \in \mathcal{X},\ z \in \mathbb{R}^m\}.$$

To get an explicit formula of $L(\lambda)$, denote $\mathcal{Z}$:=$\{z \in \mathbb{R}^m : l \le z \le u\}$ and let $\delta_{\mathcal{C}}$ be the indicator function of any convex set $\mathcal{C}$; for a proper function $p(x)$, we denote by $p^* : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ its conjugate function [24, Chapter E] which is defined by

$$p^*(s) := \sup_x\{\langle s, x\rangle - p(x)\},$$

where the inner product $\langle s, x\rangle$ is defined by $\langle s, x\rangle := s^T y$; also for a convex function $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ at $x$, Prox is defined as

$$\mathrm{Prox}_h(x) = \operatorname*{argmin}_s \left\{ h(s) + \frac{1}{2}\|s - x\|^2 \right\}. \tag{1.4}$$

Thus, by direct calculations, it holds that (similar to [23])

$$\begin{aligned} L(\lambda) &= \min_{x\in\mathcal{X},z\in\mathcal{Z}} \mathcal{L}(x,z;\lambda) \\ &= \min_{x\in\mathbb{R}^n,z\in\mathbb{R}^m} \left\{ \frac{1}{2}\|y-x\|^2 + \lambda^T(z-Ax) + \delta_{\mathcal{X}}(x) + \delta_{\mathcal{Z}}(z) \right\} \\ &= \min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\|y-x\|^2 + (A^T\lambda)^T(y-x) - (A^T\lambda)^T y + \delta_{\mathcal{X}}(x) \right\} + \min_{z\in\mathbb{R}^m} \left\{ \lambda^T z + \delta_{\mathcal{Z}}(z) \right\} \\ &= \min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\left\| y + A^T\lambda - x \right\|^2 - \frac{1}{2}\|A^T\lambda\|^2 - (Ay)^T\lambda + \delta_{\mathcal{X}}(x) \right\} + \min_{z\in\mathbb{R}^m} \left\{ \lambda^T z + \delta_{\mathcal{Z}}(z) \right\} \\ &= \frac{1}{2}\left\| \mathrm{Prox}_{\delta^*_{\mathcal{X}}}(y + A^T\lambda) \right\|^2 - \frac{1}{2}\|A^T\lambda\|^2 - (Ay)^T\lambda - \delta^*_{\mathcal{Z}}(-\lambda), \end{aligned} \tag{1.5}$$

where the last equality follows from the Moreau identity

$$\mathrm{Prox}_{\delta^*_{\mathcal{X}}}(y + A^T\lambda) = y + A^T\lambda - \mathrm{Prox}_{\delta_{\mathcal{X}}}(y + A^T\lambda).$$

Note that the proximal operator of an indicator function of a convex set $\mathcal{C}$ is the same as the projection operator onto $\mathcal{C}$, and the conjugate function $\delta^*_{\mathcal{C}}(\cdot)$ of an indicator function of a convex set $\mathcal{C}$ is the support function of $\mathcal{C}$ (see [4]). Thus, for each $\lambda$, the minimum in (1.5) is achieved by

$$x(\lambda) = \mathrm{Prox}_{\delta_{\mathcal{X}}}(y + A^T\lambda) = \Pi_{\mathcal{X}}(y + A^T\lambda)$$

and

$$z(\lambda) = \arg\min_{z\in\mathbb{R}^m} \left\{ \lambda^T z + \delta_{\mathcal{Z}}(z) \right\},$$

where $\Pi_{\mathcal{X}}(y + A^T\lambda)$ denotes the unique orthogonal projection of $y + A^T\lambda$ onto $\mathcal{X}$. It can be seen that $x(\lambda)$ and $z(\lambda)$ have the following explicit formulae

$$[x(\lambda)]_i = \begin{cases} \underline{b}_i, & \text{if } [y + A^T\lambda]_i < \underline{b}_i, \\ \bar{b}_i, & \text{if } [y + A^T\lambda]_i > \bar{b}_i, \\ [y + A^T\lambda]_i, & \text{otherwise}, \end{cases} \quad \text{for } \forall i \in [n], \tag{1.6}$$

and

$$[z(\lambda)]_j = \begin{cases} l_j, & \text{if } \lambda_j > 0, \\ [l_j, u_j], & \text{if } \lambda_j = 0, \\ u_j, & \text{if } \lambda_j < 0, \end{cases} \quad \text{for } \forall j \in [m]. \tag{1.7}$$

The conjugate function $\delta^*_{\mathcal{Z}}$ further implies $\delta^*_{\mathcal{Z}}(-\lambda) = -z(\lambda)^T\lambda$.

Now, introduce

$$D(\lambda) := -L(\lambda) = f(\lambda) + \psi(\lambda) \tag{1.8}$$

with

$$f(\lambda) := \frac{1}{2}\|A^T\lambda\|^2 + (Ay)^T\lambda - \frac{1}{2}\left\| \mathrm{Prox}_{\delta^*_{\mathcal{X}}}(y + A^T\lambda) \right\|^2, \tag{1.9a}$$

$$\psi(\lambda) := \delta^*_{\mathcal{Z}}(-\lambda) = -z(\lambda)^T\lambda = -\sum_{\lambda_i>0} \lambda_i l_i - \sum_{\lambda_i<0} \lambda_i u_i \tag{1.9b}$$

to write the dual problem (1.3) as

$$\min\{D(\lambda) : \lambda \in \mathbb{R}^m\}. \tag{1.10}$$

Note from (1.6) and (1.9a) that $f(\lambda)$ is a piecewise quadratic function, whose gradient (by [4, Theorem 6.60]) is Lipschitz continuous (but not necessarily differentiable) and is given by

$$\nabla f(\lambda) = A(y + A^T\lambda) - A\mathrm{Prox}_{\delta^*_{\mathcal{X}}}(y + A^T\lambda) = A\mathrm{Prox}_{\delta_{\mathcal{X}}}(y + A^T\lambda) = Ax(\lambda). \tag{1.11}$$

Similarly, by (1.7) and (1.9b), $\psi(\lambda)$ is a piecewise linear function, and therefore, $D(\lambda)$ is a piecewise quadratic function (see also [23]). Using (1.7) and (1.9b), one can further know the subdifferential $\partial\psi(\lambda) = -z(\lambda)$. Consequently, the subdifferential of $D(\lambda)$ is

$$\partial D(\lambda) := \nabla f(\lambda) + \partial\psi(\lambda).$$

If we get a solution $\lambda^*$ of the dual problem (1.10), we know $x(\lambda^*)$ is the solution of (1.1), and $P(x(\lambda^*)) = -D(\lambda^*)$, where $P(x)$ denotes the primal objective function in (1.1), i.e., $P(x) = \frac{1}{2}\|x - y\|^2$. Therefore, the projection problem (1.1) can be handled by solving the dual (1.10).

In the literature, numerical methods have been proposed for solving the dual problem. In particular, efficient algorithms make use of the fact that the dual (1.10) is a sum of a smooth convex function $f(\lambda)$ and a nonsmooth convex function $\psi(\lambda)$ consisting of a simple proximal mapping, and first-order algorithms such as the forward-backward splitting (FBS) or proximal gradient methods [4] can be used. In [23], Hager and Zhang proposed a first-order method that combines the sparse reconstruction by separable approximation (SpaRSA) [54] with the dual active set algorithm (DASA) [21]; the Q-linear convergence result is achieved. Furthermore, relying on the work of Nesterov [36], accelerated proximal gradient methods [4], such as FISTA [53] and MFISTA [4], have also been proposed.

Besides first-order methods, second-order methods including proximal Newton/quasi-Newton methods [5, 6, 29, 32, 40, 46] are also efficient to solve composite optimization problems. For example, Lee et al. [29] present exact/inexact proximal Newton-type methods and prove its fast convergence; in [40], Patrinos and Bemporad proposed two proximal Newton methods based on a continuously differentiable exact penalty function (Composite Moreau Envelope). In [1, 5, 6, 32], various quasi-Newton methods (such as L-BFGS) are developed towards fast but inexpensive computations.

To handle large-scale problems, various strategies for reducing computational efforts have been proposed. Among them, the screening rules [33–35] are capable of safely identifying active structures and reducing the dimension of optimization problems in data science (such as sparsity and group sparsity). Originally proposed by Ghaoui et al. [19] for Lasso, various types of screening rules have been discussed in practical applications, including sparse multi-task and multi-class models [33] and sparse group Lasso [34]. For more screening rules and related applications, we refer to [15, 37, 38, 51, 52] and the references therein.

### 1.2 Our Contributions

Motivated by safe screening rules [33–35], in this paper, we propose a Duality-Gap-Active-Set strategy (DGASS) and apply it to two proximal-type algorithms, namely the proximal gradient algorithm (PGA) and the proximal semismooth Newton algorithm (PSNA). Specifically, the proposed DGASS exploits the duality gap to safely identify zero entries and the signs of nonzero elements in the dual optimal solution $\lambda^*$. The resulting safe identification

of zeros in $\lambda^*$ leads to a reduction of dimension in the dual problem; moreover, based on the detected index set for zeros and the signs of nonzero elements, we construct an equivalent but reduced dual problem, for which the first-order method PGA can be applied. This leads to an efficient version of PGA, namely PGA-DGASS. To accelerate the convergence of PGA, we bring information of the generalized Jacobian of the dual problem to PGA subproblem and employ PSNA, a second-order method. This leads to our PSNA-DGASS algorithm. Global convergence and local quadratic convergence rate will be established under certain assumptions. The proposed methods are evaluated by comparing the performance with two popular and efficient solvers, Gurobi[1] and IPOPT[2], and our numerical experiments on both synthetic and real data sets demonstrate the high efficiency of the two DGASS-accelerated methods.

### 1.3 Organization of the Paper

We organize the paper as follows. In Sect. 2, we provide preliminary results concerning the safe detection of the active/inactive sets, duality gap region, and the reduced dual problem. In Sect. 3, we present our DGASS, and embed it into the proximal gradient algorithm. In Sect. 4, we discuss the proximal semismooth Newton algorithm with DGASS embedded. Numerical evaluation and comparisons of the proposed methods are carried out in Sect. 5, and final remarks are drawn in Sect. 6.

## 2 Safe Active/Inactive Sets

Let $x^*$ be the solution of the primal problem (1.1), which is unique due to the strong convexity of the objective function $P(x)$. Associated with $x^*$ is the optimal solution set $\Lambda$ of the dual problem (1.10). According to the strong duality theorem, for all $\lambda^* \in \Lambda$, we have $P(x(\lambda^*)) = -D(\lambda^*)$.

### 2.1 Active/Inactive Sets

Denote $[m]$ by $\mathcal{I}$. Similarly as in [23], we partition the index set $\mathcal{I}$ into the following three subsets $\mathcal{I}_0$, $\mathcal{I}_+$ and $\mathcal{I}_-$:

$$\begin{cases} \mathcal{I}_0 = & \{j \in \mathcal{I} : l_j < [Ax^*]_j < u_j\}, \\ \mathcal{I}_+ = & \{j \in \mathcal{I} : [Ax^*]_j = l_j\}, \\ \mathcal{I}_- = & \{j \in \mathcal{I} : [Ax^*]_j = u_j\}. \end{cases} \tag{2.1}$$

Here $\mathcal{I}_0, \mathcal{I}_+$ and $\mathcal{I}_-$ represent the sets of inactive constraints, active constraints corresponding to lower bounds and upper bounds at $x^*$, respectively.

The first-order optimality conditions of the primal problem (1.2) can be written as

$$Ax - z = 0, \lambda - v_1 + v_2 = 0, \tag{2.2a}$$

$$x - y - A^T\lambda - v_3 + v_4 = 0, \tag{2.2b}$$

$$v_1 \geq 0, \quad v_1^T(z - l) = 0, \quad z - l \geq 0, \tag{2.2c}$$

$$v_2 \geq 0, \quad v_2^T(z - u) = 0, \quad z - u \leq 0, \tag{2.2d}$$

[1] https://www.gurobi.com/.

[2] https://coin-or.github.io/Ipopt/.

$$v_3 \geq 0, \quad v_3^T(x - \underline{b}) = 0, \quad x - \underline{b} \geq 0, \tag{2.2e}$$

$$v_4 \geq 0, \quad v_4^T(x - \bar{b}) = 0, \quad x - \bar{b} \leq 0, \tag{2.2f}$$

where $\lambda \in \mathbb{R}^m$, $v_1 \in \mathbb{R}^m$, $v_2 \in \mathbb{R}^m$, $v_3 \in \mathbb{R}^n$ and $v_4 \in \mathbb{R}^n$ are the multipliers corresponding to equality constraints $Ax = z$, inequality constraints $l \leq z \leq u$ and $\underline{b} \leq x \leq \bar{b}$, respectively. Based on (2.2a)-(2.2f), the signs of components of $\lambda^* \in \Lambda$ can be determined.

**Proposition 2.1** *For any $\lambda^* \in \Lambda$, the following relations hold with $x^* = x(\lambda^*)$:*

$$\begin{aligned} \lambda_j^* &= 0, \quad \text{for } j \in \mathcal{I}_0, \\ \lambda_j^* &\geq 0, \quad \text{for } j \in \mathcal{I}_+, \\ \lambda_j^* &\leq 0, \quad \text{for } j \in \mathcal{I}_-. \end{aligned}$$

If $\mathcal{I}_0$ can be identified, then we are able to safely remove the variables $\lambda_j$, $j \in \mathcal{I}_0$, and hence reduce the dimension of the dual problem. Similarly, if $\mathcal{I}_+$ and/or $\mathcal{I}_-$ can be detected, then imposing additional constraints $\lambda_j \geq 0$, $j \in \mathcal{I}_+$ and/or $\lambda_j \leq 0$, $j \in \mathcal{I}_-$ to the dual problem can potentially speed up the convergence of the relevant algorithms. With these information, the dual problem can be reformulated as

$$\min_{\lambda \in \mathbb{R}^m} \tilde{D}(\lambda) := f(\lambda) + \tilde{\psi}(\lambda), \tag{2.4}$$

where $\tilde{\psi}(\lambda) := \psi(\lambda) + \delta_{\mathcal{Q}}(\lambda)$, and

$$\mathcal{Q} := \{\lambda \in \mathbb{R}^m : \lambda_{\mathcal{I}_0} = 0, \lambda_{\mathcal{I}_+} \geq 0, \ \lambda_{\mathcal{I}_-} \leq 0\} \tag{2.5}$$

is the dual feasible region. In practice, however, $\mathcal{I}_0$, $\mathcal{I}_+$ and $\mathcal{I}_-$ might not be identified exactly before the optimal solution $x^*$ is found. Thus, our practical way of detecting $\mathcal{I}_0$, $\mathcal{I}_+$ and $\mathcal{I}_-$ is to dynamically generate their approximations, which will be described in detail in the next subsections.

### 2.2 Safe Detection of Active/Inactive Sets

Our practical strategy for active/inactive sets detection is motivated by the screening rules in [33–35]. Recall Proposition 2.1 and we construct alternative index sets $\mathcal{A}_0$, $\mathcal{A}_+$ and $\mathcal{A}_-$ corresponding to a closed domain $\mathcal{C}$ containing $x^*$.

$$\begin{aligned} \mathcal{A}_+(\mathcal{C}) &:= \{j \in \mathcal{I} : [Ax]_j < u_j, \forall x \in \mathcal{C}\}, \\ \mathcal{A}_-(\mathcal{C}) &:= \{j \in \mathcal{I} : [Ax]_j > l_j, \forall x \in \mathcal{C}\}, \end{aligned} \tag{2.6}$$

$$\mathcal{A}_0(\mathcal{C}) := \mathcal{A}_+(\mathcal{C}) \cap \mathcal{A}_-(\mathcal{C}) = \{j \in \mathcal{I} : l_j < [Ax]_j < u_j, \forall x \in \mathcal{C}\}. \tag{2.7}$$

We remark that $\mathcal{A}_+(\mathcal{C})$, $\mathcal{A}_-(\mathcal{C})$ and $\mathcal{A}_0(\mathcal{C})$ are defined to approximate $\mathcal{I}_+$, $\mathcal{I}_-$ and $\mathcal{I}_0$, respectively. In particular, due to $x^* \in \mathcal{C}$, $\mathcal{A}_0(\mathcal{C})$ can safely identify the zero components of $\lambda^* \in \Lambda$, and $\mathcal{A}_+$ and $\mathcal{A}_-$ can safely identify the positive and negative signs of components of $\lambda^* \in \Lambda$, respectively.

**Proposition 2.2** *Assume that $\mathcal{C}$ is a closed region containing $x^*$ and $\mathcal{A}_+(\mathcal{C})$, $\mathcal{A}_-(\mathcal{C})$, $\mathcal{A}_0(\mathcal{C})$ are defined by (2.6)-(2.7). If $\lambda^* \in \Lambda$, $\mathcal{A}_+(\mathcal{C}) \neq \emptyset$ and $\mathcal{A}_-(\mathcal{C}) \neq \emptyset$, we have*

$$\lambda_j^* \geq 0, \ \text{for } j \in \mathcal{A}_+(\mathcal{C}), \tag{2.8a}$$

$$\lambda_j^* \leq 0, \ \text{for } j \in \mathcal{A}_-(\mathcal{C}). \tag{2.8b}$$

*Moreover, if* $\mathcal{A}_+(\mathcal{C}) \cap \mathcal{A}_-(\mathcal{C}) \neq \emptyset$, *we have*

$$\lambda_j^* = 0, \text{ for } j \in \mathcal{A}_0(\mathcal{C}). \tag{2.8c}$$

***Proof*** By (2.6)–(2.7) and (2.1), it holds that

$$\mathcal{I}_+ \cup \mathcal{I}_0 \supseteq \mathcal{A}_+(\mathcal{C}), \quad \mathcal{I}_- \cup \mathcal{I}_0 \supseteq \mathcal{A}_-(\mathcal{C}), \text{ and } \mathcal{I}_0 \supseteq \mathcal{A}_0(\mathcal{C}).$$

According to Proposition 2.1, (2.8a)–(2.8c) are true. □

***Remark 2.1*** If

$$\max_{x\in\mathcal{C}}[Ax]_j - \min_{x\in\mathcal{C}}[Ax]_j < u_j - l_j, \forall j \in \mathcal{I},$$

then $\mathcal{A}_+(\mathcal{C}) \cup \mathcal{A}_-(\mathcal{C}) = \mathcal{I}$. Indeed, if $j \notin \mathcal{A}_-(\mathcal{C})$, then $\min_{x\in\mathcal{C}}[Ax]_j \leq l_j$. Using the hypothesis, we have

$$\max_{x\in\mathcal{C}}[Ax]_j - \min_{x\in\mathcal{C}}[Ax]_j < u_j - l_j \leq u_j - \min_{x\in\mathcal{C}}[Ax]_j,$$

yielding $\max_{x\in\mathcal{C}}[Ax]_j < u_j$, i.e., $j \in \mathcal{A}_+(\mathcal{C})$. Hence, $\mathcal{I} \subseteq \mathcal{A}_+(\mathcal{C}) \cup \mathcal{A}_-(\mathcal{C})$. On the other hand, it is obvious that $\mathcal{A}_+(\mathcal{C}) \cup \mathcal{A}_-(\mathcal{C}) \subseteq \mathcal{I}$. Therefore, the desired conclusion follows.

***Remark 2.2*** If $x^* \in \mathcal{C}_2 \subset \mathcal{C}_1$, then

$$\begin{aligned} &\mathcal{I}_+ \cup \mathcal{I}_0 \supseteq \mathcal{A}_+(\mathcal{C}_2) \supseteq \mathcal{A}_+(\mathcal{C}_1), \quad \mathcal{I}_- \cup \mathcal{I}_0 \supseteq \mathcal{A}_-(\mathcal{C}_2) \supseteq \mathcal{A}_-(\mathcal{C}_1), \text{ and} \\ &\mathcal{I}_0 \supseteq \mathcal{A}_0(\mathcal{C}_2) \supseteq \mathcal{A}_0(\mathcal{C}_1). \end{aligned} \tag{2.9}$$

This implies that the smaller the $\mathcal{C}$ is, the more accurate the approximations $\mathcal{A}_0(\mathcal{C})$, $\mathcal{A}_+(\mathcal{C})$, and $\mathcal{A}_-(\mathcal{C})$ are.

For notation simplicity, we will simply denote $\mathcal{A}_0(\mathcal{C})$, $\mathcal{A}_+(\mathcal{C})$, and $\mathcal{A}_-(\mathcal{C})$ by $\mathcal{A}_0$, $\mathcal{A}_+$, and $\mathcal{A}_-$, respectively. By Proposition 2.2, we can replace $\mathcal{Q}$ in (2.5) by

$$\mathcal{Q}_\mathcal{C} := \{\lambda \in \mathbb{R}^m : \lambda_{\mathcal{A}_0} = 0, \lambda_{\mathcal{A}_+\setminus\mathcal{A}_0} \geq 0, \ \lambda_{\mathcal{A}_-\setminus\mathcal{A}_0} \leq 0\}, \tag{2.10}$$

and define a reduced dual problem

$$\min_{\lambda\in\mathbb{R}^m} \tilde{D}(\lambda) := f(\lambda) + \tilde{\psi}_\mathcal{C}(\lambda), \tag{2.11}$$

where $\tilde{\psi}_\mathcal{C}(\lambda) := \psi(\lambda) + \delta_{\mathcal{Q}_\mathcal{C}}(\lambda)$. In contrast to (2.4) which is related to $x^*$, problem (2.11) depends on the region $\mathcal{C}$. By Proposition 2.2, the index sets $\mathcal{A}_0$, $\mathcal{A}_+$ and $\mathcal{A}_-$ safely identify the zero components, the positive and negative signs of components of $\lambda^* \in \Lambda$, respectively. This implies that (2.11) and (2.4) have the same set of solutions[3]. Also, $\tilde{D}(\lambda) = D(\lambda)$ for all $\lambda \in \mathcal{Q}_\mathcal{C}$. In the next subsection, we will state how to exploit the duality gap to construct $\mathcal{C}$ during iterations.

[3] To see (2.11) and (2.4) share the same set of solutions, let $\Lambda$ be the solution set of (1.10). By Proposition 2.1, all $\lambda^* \in \Lambda$ satisfy $\lambda^* \in \mathcal{Q}$, and therefore $\Lambda \subseteq \mathcal{Q}$; as $\lambda^*$ attains the minimum of $D(\lambda)$ over $\lambda \in \mathbb{R}^m$, $\Lambda$ is the solution set of (2.4). Analogously, by Proposition 2.2, all $\lambda^* \in \Lambda$ satisfy $\lambda^* \in \mathcal{Q}_\mathcal{C}$, yielding $\Lambda \subseteq \mathcal{Q}_\mathcal{C}$, i.e., $\Lambda$ is the solution set of (2.11). Thus, $\Lambda$ is the solution set for (1.10),(2.4) and (2.11).

### 2.3 Duality Gap Region

To define $\mathcal{C}$ properly, the duality gap plays an important role.

**Lemma 2.1** *Assume that $(x, \lambda)$ is a feasible pair for (1.1) and (1.10), i.e., $x$ and $\lambda$ are feasible for (1.1) and (1.10), respectively. If $x^*$ is the optimal solution of (1.1), then*

$$\|x - x^*\| \le \sqrt{2\mathcal{G}(x, \lambda)}, \tag{2.12}$$

*where $\mathcal{G}(x, \lambda) := P(x) + D(\lambda)$ denotes the duality gap at $(x, \lambda)$.*

***Proof*** Note that $P(x) = \frac{1}{2}\|x - y\|^2$ is a strongly convex function with parameter $\tilde{\sigma} = 1$. By [4, Theorem 5.25], we have $P(x) - P(x^*) \ge \frac{1}{2}\|x - x^*\|^2$. This with $D(\lambda) \ge -P(x^*)$ yields (2.12). □

**Theorem 2.1** *Assume that $(x, \lambda)$ is a feasible pair for (1.1) and (1.10). Then the Euclidean norm ball*

$$\mathcal{B}(x, r(x, \lambda)) := \{\hat{x} \in \mathbb{R}^n : \|\hat{x} - x\| \le r(x, \lambda)\} \tag{2.13}$$

*contains the optimal solution $x^*$, where*

$$r(x, \lambda) := \sqrt{2\mathcal{G}(x, \lambda)}. \tag{2.14}$$

***Proof*** The desired conclusion is a consequence of Lemma 2.1. □

Theorem 2.1 shows that $x^* \in \mathcal{B}(x, r(x, \lambda))$ whenever $(x, \lambda)$ is a feasible pair for (1.1) and (1.10); thus $\mathcal{B}(x, r(x, \lambda))$ can be a choice for $\mathcal{C}$ (see Proposition 2.2). Theorem 2.2 provides more information for $\mathcal{A}_0(\mathcal{C})$, $\mathcal{A}_+(\mathcal{C})$, and $\mathcal{A}_-(\mathcal{C})$ with $\mathcal{C} = \mathcal{B}(x, r(x, \lambda))$.

**Theorem 2.2** *Let $\mathcal{C} = \mathcal{B}(x, r(x, \lambda))$ with $(x, \lambda)$ being a feasible pair for (1.1) and (1.10), and let $A_j^T \in \mathbb{R}^n$ be the $j^{\text{th}}$ row of $A$. Then*

$$\mathcal{A}_+(\mathcal{C}) := \left\{ j \in \mathcal{I} : \mathcal{T}_j^+(x, \lambda) < u_i \right\}, \tag{2.15a}$$

$$\mathcal{A}_-(\mathcal{C}) := \left\{ j \in \mathcal{I} : \mathcal{T}_j^-(x, \lambda) > l_j \right\}, \tag{2.15b}$$

*and*

$$\begin{aligned} \mathcal{A}_0(\mathcal{C}) &:= \mathcal{A}_+(\mathcal{C}) \cap \mathcal{A}_-(\mathcal{C}) \\ &= \left\{ j \in \mathcal{I} : l_j < \mathcal{T}_j^-(x, \lambda) \le \mathcal{T}_j^+(x, \lambda) < u_j \right\}, \end{aligned} \tag{2.15c}$$

*where*

$$\mathcal{T}_j^+(x, \lambda) := A_j^T x + \|A_j\| r(x, \lambda), \tag{2.16a}$$

$$\mathcal{T}_j^-(x, \lambda) := A_j^T x - \|A_j\| r(x, \lambda), \tag{2.16b}$$

*and $r(x, \lambda)$ is defined in (2.14).*

***Proof*** By Theorem 2.1, $\mathcal{C} = \mathcal{B}(x, r(x, \lambda))$ contains $x^*$. From (2.16a),

$$\begin{aligned} \mathcal{T}_j^+(x, \lambda) &= A_j^T x + \|A_j\| r(x, \lambda) \\ &= \max_{y \in \mathcal{B}(x, r(x, \lambda))} A_j^T y \\ &= \max_{y \in \mathcal{C}} [Ay]_j, \end{aligned} \tag{2.17}$$

which together with (2.6) yields

$$\mathcal{A}_+(\mathcal{C}) = \left\{ j \in \mathcal{I} : \max_{y\in\mathcal{C}} [Ay]_j < u_j \right\} = \left\{ j \in \mathcal{I} : \mathcal{T}_j^+(x,\lambda) < u_j \right\}.$$

Similarly, by (2.16b) and (2.6), we have

$$\begin{aligned} \mathcal{T}_j^-(x,\lambda) &= A_j^T x - \|A_j\| r(x,\lambda) \\ &= \min_{y\in\mathcal{B}(x,r(x,\lambda))} A_j^T y \\ &= \min_{y\in\mathcal{C}} [Ay]_j, \end{aligned}$$

and

$$\mathcal{A}_-(\mathcal{C}) = \left\{ j \in \mathcal{I} : \min_{y\in\mathcal{C}} [Ay]_j > l_j \right\} = \left\{ j \in \mathcal{I} : \mathcal{T}_j^-(x,\lambda) > l_j \right\}.$$

Finally, (2.15c) follows from (2.15a) and (2.15b). □

For a given $\mathcal{C}$, we call the procedure of determining $\mathcal{A}_+$, $\mathcal{A}_-$, and $\mathcal{A}_0$ (2.15)-(2.16) as the Duality-Gap-Active-Set Strategy (DGASS). Let $\{(x^k, \lambda^k)\}$ be a feasible sequence for (1.1) and (1.10). Assume $\{(x^k, \lambda^k)\}$ converges to a pair $(x^*, \lambda^*)$ with some $\lambda^* \in \Lambda$. Then, as $k$ increases, $\mathcal{C}^k := \mathcal{B}(x^k, r(x^k, \lambda^k))$ gets smaller, and therefore, more zeros components, and positive and negative signs of $\lambda^*$ are detectable.

**Theorem 2.3** *Let $\{(x^k, \lambda^k)\}$ be a convergent sequence where $x^k$ and $\lambda^k$ are feasible for (1.1) and (1.10), respectively. Suppose $\{(x^k, \lambda^k)\}$ converges to $(x^*, \lambda^*)$ with some $\lambda^* \in \Lambda$. Then*

$$\mathcal{A}_+(\mathcal{C}^k) \to \mathcal{I}_+ \cup \mathcal{I}_0, \quad \mathcal{A}_-(\mathcal{C}^k) \to \mathcal{I}_- \cup \mathcal{I}_0, \quad \mathcal{A}_0(\mathcal{C}^k) \to \mathcal{I}_0,$$

*where $\mathcal{C}^k = \mathcal{B}(x^k, r(x^k, \lambda^k))$.*

***Proof*** By assumptions, we have $\mathcal{T}_j^+(x^k, \lambda^k) \to \mathcal{T}_j^+(x^*, \lambda^*) = A_j^T x^*$ and $\mathcal{T}_j^-(x^k, \lambda^k) \to \mathcal{T}_j^-(x^*, \lambda^*) = A_j^T x^*$. From (2.1), $j \in \mathcal{I}_+ \cup \mathcal{I}_0$ if and only if $A_j^T x^* < u_j$. Due to $(x^k, \lambda^k) \to (x^*, \lambda^*)$ and (2.17), it follows that, for any $j \in \mathcal{I}_+ \cup \mathcal{I}_0$,

$$\mathcal{T}_j^+(x^k, \lambda^k) = \max_{x\in\mathcal{C}^k} [Ax]_j < u_j$$

holds for all sufficiently large $k$, and thus $j \in \mathcal{A}_+(\mathcal{C}^k)$ for all sufficiently large $k$. Conversely, if $j \in \mathcal{A}_+(\mathcal{C}^k)$ then $\mathcal{T}_j^+(x^k, \lambda^k) < u_j$ by the definition of $\mathcal{A}_+(\mathcal{C}^k)$; as $x^* \in \mathcal{C}^k$, this gives

$$A_j^T x^* \le \max_{x\in\mathcal{C}^k} [Ax]_j = \mathcal{T}_j^+(x^k, \lambda^k) < u_j,$$

which implies $j \in \mathcal{I}_+ \cup \mathcal{I}_0$. Therefore, $\mathcal{A}_+(\mathcal{C}^k) \to \mathcal{I}_+ \cup \mathcal{I}_0$. Similarly, we can prove $\mathcal{A}_-(\mathcal{C}^k) \to \mathcal{I}_- \cup \mathcal{I}_0$. The last conclusion $\mathcal{A}_0(\mathcal{C}^k) \to \mathcal{I}_0$ follows by using

$$\mathcal{A}_0(\mathcal{C}^k) = \mathcal{A}_+(\mathcal{C}^k) \cap \mathcal{A}_-(\mathcal{C}^k) \to (\mathcal{I}_+ \cup \mathcal{I}_0) \cap (\mathcal{I}_- \cup \mathcal{I}_0) = \mathcal{I}_0,$$

and the proof is complete. □

This limit behavior suggests that $\mathcal{C}^k$ can be dynamically updated. As a practical strategy, when zero components of $\lambda^*$ corresponding to $\mathcal{A}_0$ are detected safely, we can then remove the corresponding variables to reduce the dimension of (1.10), while if positive and negative signs of $\lambda^*$ corresponding to $\mathcal{A}_+$ and $\mathcal{A}_-$, respectively, are determined, we then impose sign constraints to restrict the dual variable $\lambda$. Such a strategy is useful in accelerating the iterative methods to be described in Sect. 3.

## 3 Proximal Gradient Algorithm with DGASS

In this section, we propose a DGASS-accelerated proximal gradient algorithm (PGA-DGASS).

### 3.1 Determination of $\mathcal{A}_+$, $\mathcal{A}_-$ and $\mathcal{A}_0$

Starting from $\mathcal{A}_+ = \mathcal{A}_- = \emptyset$ and $\mathcal{Q}_{\mathcal{C}} = \mathbb{R}^m$, in PGA-DGASS, we will update $\mathcal{A}_+$, $\mathcal{A}_-$ and $\mathcal{A}_0$ dynamically using the new computed information.

Let $p$ be a fixed integer. Given the iterate $\lambda^k$ with $k = pi$, $i \in \mathbb{N}^+$, we first construct a feasible point $\bar{x}^k$ for (1.1) and then define

$$\bar{\mathcal{A}}_+(k) := \bar{\mathcal{A}}_+(\mathcal{B}(\bar{x}^k, r(\bar{x}^k, \lambda^k))) := \bigcup_{\ell=1}^{i} \mathcal{A}_+(\mathcal{B}(\bar{x}^{p\ell}, r(\bar{x}^{p\ell}, \lambda^{p\ell}))), \tag{3.1}$$

$$\bar{\mathcal{A}}_-(k) := \bar{\mathcal{A}}_-(\mathcal{B}(\bar{x}^k, r(\bar{x}^k, \lambda^k))) := \bigcup_{\ell=1}^{i} \mathcal{A}_-(\mathcal{B}(\bar{x}^{p\ell}, r(\bar{x}^{p\ell}, \lambda^{p\ell}))), \tag{3.2}$$

$$\bar{\mathcal{A}}_0(k) := \bar{\mathcal{A}}_+(k) \cap \bar{\mathcal{A}}_-(k), \tag{3.3}$$

where $\mathcal{B}(\bar{x}^k, r(\bar{x}^k, \lambda^k))$ and $r(\bar{x}^k, \lambda^k)$ are defined by (2.13) and (2.14), respectively. Note

$$\bar{\mathcal{A}}_+(pi) = \bigcup_{\ell=1}^{i} \mathcal{A}_+(\mathcal{B}(\bar{x}^{p\ell}, r(\bar{x}^{p\ell}, \lambda^{p\ell}))) \subseteq \bigcup_{\ell=1}^{i+1} \mathcal{A}_+(\mathcal{B}(\bar{x}^{p\ell}, r(\bar{x}^{p\ell}, \lambda^{p\ell}))) = \bar{\mathcal{A}}_+(p(i+1)),$$

and similarly $\bar{\mathcal{A}}_-(pi) \subseteq \bar{\mathcal{A}}_-(p(i+1))$, implying the monotonically increasing property

$$\bar{\mathcal{A}}_0(pi) = \big(\bar{\mathcal{A}}_+(pi) \bigcap \bar{\mathcal{A}}_-(pi)\big) \subseteq \big(\bar{\mathcal{A}}_+(p(i+1)) \bigcap \bar{\mathcal{A}}_-(p(i+1))\big) = \bar{\mathcal{A}}_0(p(i+1)).$$

Thus by (2.9), for all $\ell$, it holds $\mathcal{A}_+(\mathcal{B}(\bar{x}^{p\ell}, r(\bar{x}^{p\ell}, \lambda^{p\ell}))) \subseteq \mathcal{I}_+ \cup \mathcal{I}_0$ and hence

$$\lim_{i\to\infty} \bar{\mathcal{A}}_+(pi) = \lim_{i\to\infty} \bigcup_{\ell=1}^{i} \mathcal{A}_+(\mathcal{B}(\bar{x}^{p\ell}, r(\bar{x}^{p\ell}, \lambda^{p\ell}))) \subseteq \mathcal{I}_+ \cup \mathcal{I}_0, \quad \lim_{i\to\infty} \bar{\mathcal{A}}_-(pi) \subseteq \mathcal{I}_- \cup \mathcal{I}_0, \text{ and } \lim_{i\to\infty} \bar{\mathcal{A}}_0(pi) \subseteq \mathcal{I}_0.$$

Since $\mathcal{I}_+$, $\mathcal{I}_-$ and $\mathcal{I}_0$ have finite indices, $\bar{\mathcal{A}}_+(pi)$, $\bar{\mathcal{A}}_-(pi)$ and $\bar{\mathcal{A}}_0(pi)$ do not change for all sufficiently large $i$. This is crucial for the global convergence of our proposed algorithms and we summarize it for ease of reference.

**Lemma 3.1** *Assume that, for each $k = pi$ with $i \in \mathbb{N}^+$, $\lambda^k \in \mathbb{R}^m$ and $\bar{x}^k \in \mathbb{R}^n$ are feasible for (1.10) and (1.1), respectively. Let $\bar{\mathcal{A}}_+(pi)$, $\bar{\mathcal{A}}_-(pi)$ and $\bar{\mathcal{A}}_0(pi)$ be defined by (3.1)–(3.3). Then $\bar{\mathcal{A}}_+(pi)$, $\bar{\mathcal{A}}_-(pi)$ and $\bar{\mathcal{A}}_0(pi)$ do not change for all sufficiently large $i$.*

To construct a feasible point $\bar{x}^k$ for (1.1), where $k = pi$, we assume $x^0$ is feasible for (1.1). Initialize $\bar{x}^0$ as $\bar{x}^0 := x^0$. Supposing a feasible $\bar{x}^{p(i-1)}$ as well as its dual approximation $\lambda^{pi}$ have been computed, the following lemma gives a way to generate $\bar{x}^{pi}$ for (1.1).

**Lemma 3.2** *Let $\bar{x}^{p(i-1)}$ be feasible for (1.1), and $\lambda^{pi}$ be feasible for (1.10). Denote*

$$\mathcal{K}_1 = \{j \in \mathcal{I} : u_j = A_j^T \bar{x}^{p(i-1)}\},$$

$$\mathcal{K}_2 = \{j \in \mathcal{I} : l_j = A_j^T \bar{x}^{p(i-1)}\}.$$

*If the following two conditions*

$$A_j^T \Delta\bar{x}^{p(i-1)} \le 0, \forall j \in \mathcal{K}_1, \tag{3.4}$$

$$A_j^T \Delta\bar{x}^{p(i-1)} \ge 0, \forall j \in \mathcal{K}_2 \tag{3.5}$$

*hold with* $\Delta\bar{x}^{p(i-1)} := x(\lambda^{pi}) - \bar{x}^{p(i-1)}$*, then a new primal feasible point* $\bar{x}^{pi}$ *can be constructed as*

$$\bar{x}^{pi} = \bar{s}x(\lambda^{pi}) + (1-\bar{s})\bar{x}^{p(i-1)},$$

*where* $\bar{s} = \min(\bar{s}_1, \bar{s}_2, 1)$ *with*

$$\bar{s}_1 = \min_{j\in\mathcal{I}\setminus\mathcal{K}_1, A_j^T\Delta\bar{x}^{p(i-1)}>0} \frac{u_j - A_j^T\bar{x}^{p(i-1)}}{A_j^T\Delta\bar{x}^{p(i-1)}},$$

$$\bar{s}_2 = \min_{j\in\mathcal{I}\setminus\mathcal{K}_2, A_j^T\Delta\bar{x}^{p(i-1)}<0} \frac{l_j - A_j^T\bar{x}^{p(i-1)}}{A_j^T\Delta\bar{x}^{p(i-1)}}.$$

***Proof*** We first prove $l \le A^T\bar{x}^{pi} \le u$. By the feasibility of $\bar{x}^{p(i-1)}$, it holds

$$l - A\bar{x}^{p(i-1)} \le 0 \le u - A\bar{x}^{p(i-1)}. \tag{3.6}$$

For $j \in \mathcal{I}$ with $A_j^T\Delta\bar{x}^{p(i-1)} \le 0$,

$$\bar{s}A_j^T\Delta\bar{x}^{p(i-1)} \le 0 \le u_j - A_j^T\bar{x}^{p(i-1)},$$

where the second inequality follows from (3.6). This implies $A_j^T\bar{x}^{pi} \le u_j$ if $A_j^T\Delta\bar{x}^{p(i-1)} \le 0$. For $j \in \mathcal{I}$ with $A_j^T\Delta\bar{x}^{p(i-1)} > 0$, the index $j$ can not be in $\mathcal{K}_1$ due to (3.4). By the definitions of $\bar{s}$ and $\bar{s}_1$, we get

$$\bar{s}A_j^T\Delta\bar{x}^{p(i-1)} \le \bar{s}_1A_j^T\Delta\bar{x}^{p(i-1)} \le u_j - A_j^T\bar{x}^{p(i-1)},$$

which also implies $A_j^T\bar{x}^{pi} \le u_j$. Hence, $A^T\bar{x}^{pi} \le u$. We can similarly prove $l \le A^T\bar{x}^{pi}$.

Next, we show $\underline{b} \le \bar{x}^{pi} \le \bar{b}$. Note from (1.6) that $\underline{b} \le x(\lambda^{pi}) \le \bar{b}$. Since $\bar{x}^{p(i-1)}$ is feasible for (1.1), the convex combination $\bar{x}^{pi} = \bar{s}x(\lambda^{pi}) + (1-\bar{s})\bar{x}^{p(i-1)}$ is located on $[\underline{b}, \bar{b}]$. □

Application of Lemma 3.2 relies on two additional assumptions (3.4)-(3.5). For a more practical way, we relax it by defining a weakly strictly feasible point for (1.1).

**Definition 3.1** A feasible point $x^0 \in \mathbb{R}^m$ for (1.1) is said to be a weakly strictly feasible point if

$$l < Ax^0 < u,\ \underline{b} \le x^0 \le \bar{b}.$$

In our strategy, if one of (3.4)–(3.5) is violated for $\bar{x}^{p(i-1)}$, we then resort to a weakly strictly feasible point $\tilde{x}^{p(i-1)}$ constructed as a convex combination of $\bar{x}^{p(i-1)}$ and $x^0$ (here $x^0$ is assumed to be a weakly strictly feasible point), i.e.,

$$\tilde{x}^{p(i-1)} := \varsigma\bar{x}^{p(i-1)} + (1-\varsigma)x^0, \tag{3.7}$$

where $\varsigma \in (0, 1)$ is a fixed scalar. Replacing $\bar{x}^{p(i-1)}$ with $\tilde{x}^{p(i-1)}$ and applying Lemma 3.2, we obtain the following lemma.

**Lemma 3.3** *Let $\tilde{x}^{p(i-1)}$ be weakly strictly feasible for (1.1), and let $\lambda^{pi}$ be feasible for the dual problem (1.10). Denote $\Delta\tilde{x}^{p(i-1)} := x(\lambda^{pi}) - \tilde{x}^{p(i-1)}$. Then a new primal feasible point $\bar{x}^{pi}$ can be constructed as*

$$\bar{x}^{pi} = \bar{s}x(\lambda^{pi}) + (1-\bar{s})\tilde{x}^{p(i-1)},$$

*where $\bar{s} = \min(\bar{s}_1, \bar{s}_2, 1)$ with*

$$\bar{s}_1 = \min_{j\in\mathcal{I}, A_j^T\Delta\tilde{x}^{p(i-1)}>0} \frac{u_j - A_j^T\tilde{x}^{p(i-1)}}{A_j^T\Delta\tilde{x}^{p(i-1)}},$$

$$\bar{s}_2 = \min_{j\in\mathcal{I}, A_j^T\Delta\tilde{x}^{p(i-1)}<0} \frac{l_j - A_j^T\tilde{x}^{p(i-1)}}{A_j^T\Delta\tilde{x}^{p(i-1)}}.$$

### 3.2 Barzilai-Borwein Stepsize

First proposed by Barzlai and Borwein [3], the BB method has received many interests in the community of optimization (see e.g., [7, 16, 26, 27, 42]); generally, it is an effective method for the smooth minimization problem (see [16, 42])

$$\min\{\bar{f}(\lambda) : \lambda \in \mathbb{R}^m\},$$

where $\bar{f} : \mathbb{R}^m \to \mathbb{R}$ is continuously differentiable. At each iteration $k$, the BB gradient method generates iterates as $\lambda^{k+1} = \lambda^k - \alpha^k\nabla\bar{f}(\lambda^k)$, where the stepsize $\alpha^k > 0$ [3] can be computed by

$$\alpha_l^k = \frac{\|s^{k-1}\|_2^2}{(s^{k-1})^T w^{k-1}} \quad \text{and} \quad \alpha_s^k = \frac{(s^{k-1})^T w^{k-1}}{\|w^{k-1}\|_2^2},$$

with $s^{k-1} = \lambda^k - \lambda^{k-1}$, $w^{k-1} = \nabla\bar{f}(\lambda^k) - \nabla\bar{f}(\lambda^{k-1})$. The stepsize $\alpha_l^k$ and $\alpha_s^k$ represent the long BB stepsize and the short BB one, respectively. Extensive numerical results (see [7, 16, 42]) show that in general the long BB stepsize $\alpha_l^k$ outperforms the short one $\alpha_s^k$ in many cases. Furthermore, Zhou, Gao and Dai [58] present an adaptive stepsize selection strategy given by

$$\alpha_B^k = \begin{cases} \min\left(\alpha_{\max}, \max(\alpha_{\min}, \alpha_s^k)\right), & \alpha_s^k/\alpha_l^k \le \kappa, \\ \min\left(\alpha_{\max}, \max(\alpha_{\min}, \alpha_l^k)\right), & \text{otherwise}, \end{cases} \tag{3.8}$$

where $\kappa \in (0, 1)$, and $\alpha_{\max} > \alpha_{\min} > 0$. The rule (3.8) is a combination of the long and the short BB stepsize, determined by the trade-off parameter $\kappa$, which generally is set around 0.2. Other adaptive strategies are discussed in [11, 58].

The BB strategy is also effective for nonsmooth minimization problems (see [20, 44]). For our case, the first term $f(\lambda)$ in $\tilde{D}(\lambda)$ of (2.11) is continuously differentiable, while $\tilde{\psi}_{\mathcal{C}}(\lambda)$ is piecewise linear. We notice that the iterates generated by PGA may stay on some linear subspace and thus the piecewise linear terms reduce to linear functions; when this happens, then the nonsmooth terms will not effect the BB stepsize. For this reason, we particularly replace $w^{k-1} = \nabla\bar{f}(\lambda^k) - \nabla\bar{f}(\lambda^{k-1})$ with $w^{k-1} = \nabla f(\lambda^k) - \nabla f(\lambda^{k-1})$, and employ the long BB stepsize

$$\alpha_{BB}^k = \min\left(\alpha_{\max}, \max\left(\alpha_{\min}, \frac{\|s^{k-1}\|_2^2}{(s^{k-1})^T w^{k-1}}\right)\right). \tag{3.9}$$

### 3.3 PGA-DGASS for (2.11)

The proximal gradient algorithm (PGA) has been widely used to minimize composite nonsmooth functions (see [4]). Associated with PGA is the proximal operator Prox defined in (1.4). Using $\mathrm{Prox}_h(\lambda)$, an iteration of PGA for solving (2.11) takes the form:

$$\lambda^{k+1} = \lambda^k + \alpha^k d_g^k(\alpha^k),$$

where $\alpha^k > 0$ is the stepsize, and $d_g^k(\alpha^k) = (\mathrm{Prox}_{\alpha^k \tilde{\psi}_{\mathcal{C}}}(\lambda^k - \alpha^k \nabla f(\lambda^k)) - \lambda^k)/\alpha^k$ is a direction. Here, the proximal operator Prox is defined in (1.4). It is easy to verify that such a $\lambda^{k+1}$ is the minimizer of the sum of the linearization of $f$ around $\lambda^k$, the nonsmooth function $\tilde{\psi}_{\mathcal{C}}$, and a quadratic proximal term:

$$\min_{\lambda \in \mathbb{R}^m} \Phi_k(\lambda; \alpha^k) := f(\lambda^k) + \nabla f(\lambda^k)^T(\lambda - \lambda^k) + \frac{1}{2\alpha^k}\|\lambda - \lambda^k\|^2 + \tilde{\psi}_{\mathcal{C}}(\lambda).$$

Equivalently, $d_g^k(\alpha^k)$ is the minimizer of

$$\min_{d_g \in \mathbb{R}^m} \Phi_k(\lambda^k + \alpha^k d_g; \alpha^k) := f(\lambda^k) + \alpha^k \nabla f(\lambda^k)^T d_g + \frac{\alpha^k}{2}\|d_g\|^2 + \tilde{\psi}_{\mathcal{C}}(\lambda^k + \alpha^k d_g). \quad (3.10)$$

The special structure of $\tilde{\psi}_{\mathcal{C}}(\lambda)$ further leads to a closed form of the proximal operator of $\tilde{\psi}_{\mathcal{C}}(\lambda)$.

**Lemma 3.4** *Let $\tilde{\psi}_{\mathcal{C}}(\lambda) = \psi(\lambda) + \delta_{\mathcal{Q}_{\mathcal{C}}}(\lambda)$, where $\psi(\lambda)$ and $\mathcal{Q}_{\mathcal{C}}$ are defined in (1.9b) and (2.10), respectively. Then for any stepsize $\alpha^k > 0$, it holds that*

$$Prox_{\alpha^k \tilde{\psi}_{\mathcal{C}}}(\lambda^k - \alpha^k \nabla f(\lambda^k)) = \left([Prox_{\alpha^k \tilde{\psi}_{\mathcal{C}}}(\lambda^k - \alpha^k \nabla f(\lambda^k))]_j\right)_{j=1}^m$$

*with*

$$[Prox_{\alpha^k \tilde{\psi}_{\mathcal{C}}}(\lambda^k - \alpha^k \nabla f(\lambda^k))]_j = \begin{cases} \lambda_j^k(l), & \text{if } \lambda_j^k(l) > 0 \ \& \ j \notin \mathcal{A}_-, \\ \lambda_j^k(u), & \text{if } \lambda_j^k(u) < 0 \ \& \ j \notin \mathcal{A}_+, \\ 0, & \text{otherwise}, \end{cases} \quad (3.11)$$

*where*

$$\lambda_j^k(l) := [\lambda^k - \alpha^k(Ax(\lambda^k) - l)]_j \quad \text{and} \quad \lambda_j^k(u) := [\lambda^k - \alpha^k(Ax(\lambda^k) - u)]_j. \quad (3.12)$$

***Proof*** The consequences can be obtained from (1.11), and the definitions of $\tilde{\psi}_{\mathcal{C}}$ and the proximal operator. □

For the solution $d_g^k(\alpha^k)$ of (3.10), we have the optimality condition:

$$0 \in \nabla f(\lambda^k) + d_g^k(\alpha^k) + \partial \tilde{\psi}_{\mathcal{C}}(\lambda^k + \alpha^k d_g^k(\alpha^k)). \quad (3.13)$$

Here, for any $\lambda \in \mathcal{Q}_{\mathcal{C}}$, the subdifferential $\partial \tilde{\psi}_{\mathcal{C}}(\lambda) = ([\partial \tilde{\psi}_{\mathcal{C}}(\lambda)]_j)_{j=1}^m$ with

$$[\partial \tilde{\psi}_{\mathcal{C}}(\lambda)]_j = \begin{cases} -l_j, & \text{if } \lambda_j > 0, \\ -u_j, & \text{if } \lambda_j < 0, \\ [-u_j, -l_j], & \text{if } \lambda_j = 0 \ \& \ j \notin \mathcal{A}_+ \cup \mathcal{A}_-, \\ (-\infty, -l_j], & \text{if } \lambda_j = 0 \ \& \ j \in \mathcal{A}_+ \backslash \mathcal{A}_0, \\ [-u_j, +\infty), & \text{if } \lambda_j = 0 \ \& \ j \in \mathcal{A}_- \backslash \mathcal{A}_0, \\ (-\infty, +\infty), & \text{if } \lambda_j = 0 \ \& \ j \in \mathcal{A}_0 \end{cases} \quad (3.14)$$

can be derived from (1.9b), (2.10), and the definition of the subdifferential. Note that if $d_g^k(1) = 0$, then from (3.13), we have

$$0 \in \nabla f(\lambda^k) + \partial \tilde{\psi}_{\mathcal{C}}(\lambda^k),$$

which implies that $\lambda^k$ solves (2.11). Thus, $\|d_g^k(1)\|_\infty$ serves as a measure of optimality for $\lambda^k$.

To determine the stepsize $\alpha^k$, we generally desire a sufficient decrease in the dual objective function:

$$\tilde{D}(\lambda^k + \alpha^k d_g^k(\alpha^k)) \leq \tilde{D}^k(\lambda^k) - \xi \alpha^k \Delta_g^k(\alpha^k) \tag{3.15}$$

at each iteration, where $\xi \in (0, \frac{1}{2})$ and $\Delta_g^k(\alpha^k) := \|d_g^k(\alpha^k)\|^2$. By [4, Remark 10.20], it follows that (3.15) is fulfilled for all $\alpha^k \in (0, \frac{1}{L_f})$, where $L_f > 0$ is the Lipschitz constant of $\nabla f(\lambda)$. In our case, we choose a modified nonmonotone sufficient descent condition (see [55])

$$\tilde{D}(\lambda^k + \alpha^k d_g^k(\alpha^k)) \leq \tilde{D}_R^k - \xi \alpha^k \Delta_g^k(\alpha^k) \tag{3.16}$$

for determining $\alpha^k > 0$, where $\tilde{D}_R^k$ is recursively updated by

$$\tilde{D}_R^k = \frac{\zeta Q^{k-1} \tilde{D}_R^{k-1} + \tilde{D}(\lambda^k)}{Q^k} \tag{3.17}$$

with $Q^k = \zeta Q^{k-1} + 1$, $Q^0 = 0$, $\tilde{D}_R^0 = \tilde{D}(\lambda^0)$, and $\zeta \in (0, 1)$. We remark that such a nonmonotone technique (3.16) has been widely used in applications of semidefinite programming [45], manifold optimization [18, 25], multiobjective optimization [56] and others. Indeed $\tilde{D}_R^k$ is a weighted average of the previous objective function values $\tilde{D}(\lambda^j)$, $j = 1, \ldots, k$, where the weights depend on the decay parameter $\zeta$ (say, $\zeta = 0.9$). By [55, Lemma 1], it holds $\tilde{D}(\lambda^k) \leq \tilde{D}_R^k \leq \chi^k$, for all $k$, where $\chi^k = \frac{1}{k} \sum_{j=0}^k \tilde{D}(\lambda^j)$.

The overall PGA-DGASS then can be summarized in Algorithm 1.

We have several more comments for Algorithm 1 before the global convergence in Theorem 3.1.

***Remark 3.1*** The strategy DGASS is carried out in lines 15 and 16 in Algorithm 1.

***Remark 3.2*** In our implementation, all the variables $\lambda_\ell$, $\ell \in \mathcal{A}_0$ will be removed whenever $\mathcal{A}_0 \neq \emptyset$. This reduces the dimension of the dual (2.10).

***Remark 3.3*** In line 4 of Algorithm 1, we set $\alpha^k = \alpha_{BB}^k$ as a trial at each iteration. If the sufficient descent condition (3.16) is not met, then we shrink $\alpha^k$ until it is fulfilled.

***Remark 3.4*** In line 15 of Algorithm 1, we generate a feasible point $\bar{x}^k$ for (1.1) according to Lemmas 3.2 and 3.3.

**Theorem 3.1** *Let $\{\lambda^k\}$ be the sequence generated by Algorithm 1, starting from a feasible pair $(x^0, \lambda^0)$ for (1.1) and (1.10). Then any accumulation point $\lambda^*$ of $\{\lambda^k\}$ is a solution of (1.10), i.e., $\lambda^* \in \Lambda$.*

***Proof*** According to Lemma 3.1, the asymptotic behavior of Algorithm 1 is the same as that of PGA for the dual problem (2.10). Thus, the conclusion follows by a similar argument of [4, Theorem 10.24]. □

**Algorithm 1:** PGA-DGASS

Given $\xi \in (0, 1/2), \sigma \in (0, 1), \zeta \in (0, 1), p \in \mathbb{N}^+$;
Initialization: $\lambda^0 \in \mathbb{R}^m, \alpha_{BB}^0 = 1, \tilde{D}_R^0 = \tilde{D}(\lambda^0), \mathcal{A}_0 = \mathcal{A}_+ = \mathcal{A}_- = \emptyset, \ \mathcal{Q}_\mathcal{C} = \mathbb{R}^m, k = 0$;
**while** *not convergent* **do**
  Set $\alpha^k = \alpha_{BB}^k$;
  Compute $d_g^k(\alpha^k)$ by solving (3.10);
  **while** $\tilde{D}(\lambda^k + \alpha^k d_g^k(\alpha^k)) > \tilde{D}_R^k - \xi \alpha^k \Delta_g^k(\alpha^k)$ **do**
    Set $\alpha^k = \sigma \alpha^k$;
    Compute $d_g^k(\alpha^k)$ by solving (3.10);
  **end**
  Set $\lambda^{k+1} = \lambda^k + \alpha^k d_g^k(\alpha^k)$;
  Compute the BB step size $\alpha_{BB}^{k+1}$ by (3.9);
  Update $\tilde{D}_R^{k+1}$ via (3.17);
  Set $k = k + 1$;
  **if** $\mathrm{mod}(k, p) = 0$ **then**
    Compute $\mathcal{A}_0 = \bar{\mathcal{A}}_0(k)$, $\mathcal{A}_+ = \bar{\mathcal{A}}_+(k)$ and $\mathcal{A}_- = \bar{\mathcal{A}}_-(k)$ at the new iterate $\lambda^k$ via (3.1)-(3.3);
    Update $\mathcal{Q}_\mathcal{C}$ by (2.10);
  **end**
**end**

## 4 Proximal Semismooth Newton Algorithm with DGASS

We have demonstrated that the identification of $\mathcal{A}_+, \mathcal{A}_-$ and $\mathcal{A}_0$ facilitates the computation of the proximal operation of (3.11) in PGA-DGASS, and in this section, we shall further show that these index sets work as the active/inactive sets and, therefore improve the efficiency of the proximal semismooth Newton algorithm.

### 4.1 Semismoothness

**Definition 4.1** (Semismoothness [31, 41, 48]) Suppose $F : \mathcal{O} \subseteq \mathbb{R}^m \to \mathbb{R}^m$ is locally Lipschitz continuous on the open set $\mathcal{O}$. $F$ is said to be semismooth at $\lambda \in \mathcal{O}$ if

(a) $F$ is directionally differentiable, and
(b) for any direction $d \to 0$ and $V \in \partial F(\lambda + d)$, it holds that

$$F(\lambda + d) - F(\lambda) - Vd = o(\|d\|).$$

Also, $F$ is said to be strongly semismooth at $\lambda \in \mathcal{O}$ if the condition ((a)) is true, and

(b′) for any direction $d \to 0$ and $V \in \partial F(\lambda + d)$, it holds that

$$F(\lambda + d) - F(\lambda) - Vd = O(\|d\|^2).$$

Moreover, $F$ is said to be a semismooth (respectively, strongly semismooth) function on the open set $\mathcal{O}$ if it is semismooth (respectively, strongly semismooth) everywhere in $\mathcal{O}$.

Note that any continuous piecewise affine function is strongly semismooth [14, Proposition 7.4.7]. So the gradient function $F(\lambda) = \nabla f(\lambda)$ defined by (1.11) is strongly semismooth.

Let $\partial(\nabla f(\lambda))$ be the generalized Hessian of $f(\lambda)$ at $\lambda$ in the sense of Clarke [10]. Generally, explicit expression for $\partial(\nabla f(\lambda))$ is not easy to formulate, but the set $\hat{\partial}(\nabla f(\lambda))$ given

by

$$\hat{\partial}(\nabla f(\lambda)) := A\partial \mathrm{Prox}_{\delta_{\mathcal{X}}}(y + A^T\lambda)A^T, \tag{4.1}$$

contains $\partial(\nabla f(\lambda))$ (see [10, Proposition 2.3.3, Theorem 2.6.6] or [2, Theorems 3.16 and 3.20]), i.e., $\partial(\nabla f(\lambda)) \subseteq \hat{\partial}(\nabla f(\lambda))$, where $\partial \mathrm{Prox}_{\delta_{\mathcal{X}}}(\cdot)$ denotes the Clarke generalized Jacobian of $\mathrm{Prox}_{\delta_{\mathcal{X}}}(\cdot)$. The following lemma gives the explicit form of $\hat{\partial}(\nabla f(\lambda))$, which has the so-called second order sparsity [30].

**Lemma 4.1** *The set $\hat{\partial}(\nabla f(\lambda))$ defined in (4.1) is*

$$\hat{\partial}(\nabla f(\lambda)) = \{AUA^T | U \in \mathcal{U}\}, \tag{4.2}$$

*where $\mathcal{U} := \partial Prox_{\delta_{\mathcal{X}}}(y + A^T\lambda)$ with*

$$\partial \mathrm{Prox}_{\delta_{\mathcal{X}}}(w) := \left\{ \mathrm{Diag}(v) \,\middle|\, v_i = \begin{cases} 1, & \underline{b}_i < w_i < \bar{b}_i, \\ 0, & w_i < \underline{b}_i \text{or} w_i > \bar{b}_i, \\ [0,1] & w_i = \underline{b}_i \text{or} w_i = \bar{b}_i \end{cases} \quad i \in [n] \right\},$$

*and $Diag(v) \in \mathbb{R}^{|v|\times|v|}$ denotes a diagonal matrix with the elements of vector $v$ on the main diagonal.*

***Proof*** By (1.11), $\nabla f(\lambda) = A(\mathrm{Prox}_{\delta_{\mathcal{X}}}\left(y + A^T\lambda\right))$. It follows from (1.6) that

$$\partial \mathrm{Prox}_{\delta_{\mathcal{X}}}(w) = \left\{ \mathrm{Diag}(v) \,\middle|\, v_i = \begin{cases} 1, & \underline{b}_i < w_i < \bar{b}_i, \\ 0, & w_i < \underline{b}_i \text{ or } w_i > \bar{b}_i, \\ [0,1] & w_i = \underline{b}_i \text{ or } w_i = \bar{b}_i \end{cases} \quad i \in [n] \right\}; \tag{4.3}$$

this implies $\hat{\partial}(\nabla f(\lambda)) = \{AUA^T | U \in \mathcal{U}\}$. □

Note from (4.2) that all matrices in $\hat{\partial}(\nabla f(\lambda))$ are positive semidefinite, and could be singular; particularly, all matrices in $\hat{\partial}(\nabla f(\lambda))$ are singular if $m > n$.

### 4.2 PSNA-DGASS for (2.11)

As a second-order method, the proximal Newton-type method [29, 39] is used to accelerate first-order methods. In our PSNA-DGASS, similar to [29], we propose to sequentially solve quadratic minimizations:

$$\min_{\lambda\in\mathbb{R}^m} \Psi_k(\lambda; H^k) := f(\lambda^k) + \nabla f(\lambda^k)^T(\lambda - \lambda^k) + \frac{1}{2}(\lambda - \lambda^k)^T H^k(\lambda - \lambda^k) + \tilde{\psi}_{\mathcal{C}}(\lambda) \tag{4.4}$$

where $\Psi_k(\lambda; H^k)$ is an approximation of $\tilde{D}(\lambda) = f(\lambda) + \tilde{\psi}_{\mathcal{C}}(\lambda)$ at $\lambda^k$ and $H^k$ is a positive definite matrix that approximates elements in $\hat{\partial}(\nabla f(\lambda^k))$. To deal with the singularity in the elements of $\hat{\partial}(\nabla f(\lambda^k))$, we choose $H^k$ as the following regularized positive definite matrix

$$H^k = V^k + \bar{\vartheta}^k I \text{ with } V^k \in \hat{\partial}(\nabla f(\lambda^k)), \tag{4.5}$$

where $\bar{\vartheta}^k := \min(\varepsilon_0, \vartheta^k)$, $\varepsilon_0 \in (0, 1)$ and $\vartheta^k := \left\| d_g^k(1) \right\|$.

Define the scaled proximal mapping [29] as

$$\mathrm{Prox}_h^{H^k}(\lambda) := \arg\min_w \left\{ \frac{1}{2}\|w - \lambda\|_{H^k}^2 + h(w) \right\}.$$

Note the scaled proximal mapping is firmly nonexpansive [29, after (2.11)] and Lipschitz continuous (the Lipschitz constant is 1), i.e., for any $w, \bar{w} \in \mathbb{R}^m$,

$$\left(\text{Prox}_h^{H^k}(w) - \text{Prox}_h^{H^k}(\bar{w})\right)^T H^k(w - \bar{w}) \geq \left\|\text{Prox}_h^{H^k}(w) - \text{Prox}_h^{H^k}(\bar{w})\right\|_{H^k}^2, \tag{4.6a}$$

$$\left\|\text{Prox}_h^{H^k}(w) - \text{Prox}_h^{H^k}(\bar{w})\right\|_{H^k} \leq \|w - \bar{w}\|_{H^k}, \tag{4.6b}$$

where the $H^k$-norm $\|\cdot\|_{H^k}$ is defined as $\|w\|_{H^k} := \sqrt{w^T H^k w}$. By the positive definiteness of $H^k$, the optimal solution, denoted by $\lambda_N^{k+1}$, to (4.4) is unique, which can be given by

$$\lambda_N^{k+1} = \text{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}(\lambda^k - (H^k)^{-1}\nabla f(\lambda^k)).$$

Let $d_N = \lambda - \lambda^k$. The subproblem (4.4) can be rewritten as

$$\min_{d_N \in \mathbb{R}^m} \Psi_k(\lambda^k + d_N; H^k) = f(\lambda^k) + \nabla f(\lambda^k)^T d_N + \frac{1}{2} d_N^T H^k d_N + \tilde{\psi}_{\mathcal{C}}(\lambda^k + d_N), \tag{4.7}$$

and the generalized Newton search direction

$$d_N^k := \lambda_N^{k+1} - \lambda^k = \text{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}(\lambda^k - (H^k)^{-1}\nabla f(\lambda^k)) - \lambda^k$$

is obtained by solving the subproblem (4.7). Equivalently,

$$0 \in \nabla f(\lambda^k) + H^k d_N^k + \partial\tilde{\psi}_{\mathcal{C}}(\lambda^k + d_N^k). \tag{4.8}$$

Note that $\lambda^k$ is optimal if and only if $d_N^k = 0$.

The proximal gradient method (see [4]) can be used to solve (4.7) for $d_N^k$, and then we update the iterate as $\lambda^{k+1} = \lambda^k + \alpha^k d_N^k$, where $\alpha^k > 0$ is the stepsize to fulfill the sufficient descent condition in $\tilde{D}(\lambda)$

$$\tilde{D}(\lambda^{k+1}) \leq \tilde{D}_R^k - \xi\alpha^k \Delta_N^k, \tag{4.9}$$

where $\Delta_N^k := (d_N^k)^T H^k d_N^k$, and $\tilde{D}_R^k$ is defined by (3.17). The overall PSNA-DGASS is given in Algorithm 2.

***Remark 4.1*** In Algorithm 2, one can use PGA to solve (4.7). Also, an approximation of (4.7) with a certain level of accuracy is sufficient to ensure global convergence. In our implementation, we employ PGA to solve (4.7) and terminate it whenever the ratio of the residuals between (4.7) and (1.10) is less than a given constant. That is, we terminate PGA whenever

$$\|\text{Prox}_{\tilde{\psi}_{\mathcal{C}}}(\lambda^k - \nabla f(\lambda^k) - H^k d_N^{k,j}) - \lambda^k\| \leq \bar{\gamma} d_g^k(1),$$

where $\bar{\gamma} \in (0, 1)$ is a given constant, and $d_N^{k,j}$ is the $j$-th iterate from PGA.

### 4.3 Convergence Analysis of PSNA-DGASS

In this subsection, we establish the global convergence and the local convergence rate of Algorithm 2.

**Algorithm 2:** PSNA-DGASS

Given $\xi \in (0, 1/2)$, $\sigma \in (0, 1)$, $\zeta \in (0, 1)$, $p \in \mathbb{N}^+$;
Initialization: $\lambda^0 \in \mathbb{R}^m$, $\tilde{D}_R^0 = \tilde{D}(\lambda^0)$, $\mathcal{A}_0 = \mathcal{A}_+ = \mathcal{A}_- = \emptyset$, $\mathcal{Q}_\mathcal{C} = \mathbb{R}^m$, $k = 0$;
**while** *not convergent* **do**
  Set $\alpha^k = 1$;
  Compute $d_N^k$ by solving (4.7);
  **while** $\tilde{D}(\lambda^k + \alpha^k d_N^k) > \tilde{D}_R^k - \xi \alpha^k \Delta_N^k$ **do**
    Set $\alpha^k = \sigma \alpha^k$;
  **end**
  Set $\lambda^{k+1} = \lambda^k + \alpha^k d_N^k$;
  Update $\tilde{D}_R^{k+1}$ via (3.17);
  Set $k = k + 1$;
  **if** $\mathrm{mod}(k, p) = 0$ **then**
    Compute $\mathcal{A}_0 = \bar{\mathcal{A}}_0(k)$, $\mathcal{A}_+ = \bar{\mathcal{A}}_+(k)$ and $\mathcal{A}_- = \bar{\mathcal{A}}_-(k)$ at the new iterate $\lambda^k$ via (3.1)-(3.3);
    Update $\mathcal{Q}_\mathcal{C}$ by (2.10);
  **end**
**end**

**Lemma 4.2** *For* $0 < \alpha \le \min(1, \frac{2\vartheta^k(1-\xi)}{L_f})$, *it holds that*

$$\tilde{D}(\lambda^k + \alpha d_N^k) \le \tilde{D}_R^k - \xi \alpha \Delta_N^k, \tag{4.10}$$

*where* $L_f > 0$ *is the Lipschitz constant of* $\nabla f(\lambda)$.

***Proof*** By [29, Proposition 2.6], we know that for $0 < \alpha \le \min(1, \frac{2\vartheta^k(1-\xi)}{L_f})$,

$$\tilde{D}(\lambda^k + \alpha d_N^k) \le \tilde{D}(\lambda^k) - \xi \alpha \Delta_N^k. \tag{4.11}$$

Due to [55, Lemma 1], $\tilde{D}(\lambda^k) \le \tilde{D}_R^k$. Thus, for $0 < \alpha \le \min(1, \frac{2\vartheta^k(1-\xi)}{L_f})$, (4.10) is true. □

**Lemma 4.3** *Let* $\alpha^k$ *be the stepsize in line 9 of Algorithm 2. Then the nonmonotone descent condition* (4.9) *is satisfied, and* $\alpha^k \ge \sigma \min(1, \frac{2\vartheta^k(1-\xi)}{L_f})$.

***Proof*** The conclusion follows from the procedure of Algorithm 2 and Lemma 4.2. □

**Theorem 4.1** *Let* $\{\lambda^k\}$ *be the sequence generated by Algorithm 2, and starting from a feasible pair* $(x^0, \lambda^0)$ *for* (1.1) *and* (1.10). *Then any accumulation point of* $\{\lambda^k\}$ *is an optimal point of the dual problem* (2.11).

***Proof*** According to Lemma 3.1, Algorithm 2 asymptotically reduces to a proximal semismooth Newton algorithm. Without loss of generality, we assume that $\mathcal{A}_+$, $\mathcal{A}_-$ and $\mathcal{A}_0$ are constants for all iterations. The update rule of $Q^{k+1}$ given by Algorithm 2 yields

$$Q^{k+1} = 1 + \zeta Q^k = 1 + \sum_{j=1}^{k} \zeta^j \le \frac{1}{1-\zeta}. \tag{4.12}$$

By (4.9) and (3.17), we have

$$\tilde{D}_R^{k+1} = \frac{\zeta Q^k \tilde{D}_R^k + \tilde{D}(\lambda^{k+1})}{Q^{k+1}}$$

$$\leq \frac{(\zeta Q^k + 1)\tilde{D}_R^k - \xi\alpha^k \Delta_N^k}{Q^{k+1}}$$
$$\leq \tilde{D}_R^k - \xi(1-\zeta)\alpha^k \Delta_N^k, \tag{4.13}$$

where the last inequality follows from (4.12). Hence, $\{\tilde{D}_R^k\}$ is a monotonically decreasing sequence. Since $x^0$ is feasible for (1.1), it holds that $-P(x^0) \leq \tilde{D}(\lambda^k) \leq \tilde{D}_R^k$ for all $k$, which implies the convergence of $\{\tilde{D}_R^k\}$. Summing (4.13) for $k = 0, 1, \ldots,$ we obtain that

$$\xi(1-\zeta)\sum_{k=1}^{\infty} \alpha^k \Delta_N^k \leq \sum_{k=0}^{\infty} (\tilde{D}_R^k - \tilde{D}_R^{k+1}) < +\infty; \tag{4.14}$$

hence by Lemma 4.3 and the definition of $H^k$, it yields

$$\lim_{k\to\infty} \min(1, \tfrac{2\vartheta^k(1-\xi)}{L_f})\vartheta^k \|d_N^k\|^2 = 0.$$

Now, let $\lambda^*$ be an accumulation point of $\{\lambda^k\}$. Without loss of generality, we assume $\{\lambda^k\}$ itself converges to $\lambda^*$. If there exists an index set $\mathcal{K} \subseteq \mathbb{N}^+$ such that

$$\lim_{k\to\infty, k\in\mathcal{K}} \vartheta^k = 0,$$

then $\lambda^*$ is an optimal point for the dual (2.11). Otherwise, there exists a scalar $\hat{\epsilon} > 0$ such that

$$\vartheta^k \geq \hat{\epsilon} > 0, \quad \forall k.$$

This combining with (4.14) gives $\lim_{k\to\infty} d_N^k = 0$. On the other hand, by Lemma 4.1 and the definition of $\vartheta^k$, the sequence $\{H^k\}$ is bounded for all $k$. Thus, if we let $k \to \infty$ in (4.8), we have $0 \in \nabla f(\lambda^*) + \partial\tilde{\psi}_{\mathcal{C}}(\lambda^*)$ by the continuity of the subdifferential $\partial\tilde{\psi}_{\mathcal{C}}(\lambda)$ at $\lambda^*$ (see [43, Theorem 24.4]). Hence, $\lambda^*$ is a minimizer of (2.11). □

Next, we perform quadratic local convergence analysis for Algorithm 2. In view of [23, Corollary 3.2], the following error bound condition holds:

$$\text{dist}(0, \partial\tilde{D}(\lambda)) \geq \bar{c}\,\text{dist}(\lambda, \Lambda) \tag{4.15}$$

for all $\lambda \in \mathbb{R}^m$ with $\tilde{D}(\lambda) \leq \tilde{D}(\lambda^0)$, where $\text{dist}(\lambda, C) := \inf_{\lambda'\in C} \|\lambda - \lambda'\|$ is the distance of $\lambda \in \mathbb{R}^m$ to $C \subseteq \mathbb{R}^m$. This fact leads to an important property of $d_N^k$.

**Lemma 4.4** *There exist two scalars $c_1 > 0$ and $c_2 > 0$ such that*

$$c_1\,\text{dist}(\lambda^k, \Lambda) \leq \left\|d_N^k\right\| \leq c_2\,\text{dist}(\lambda^k, \Lambda), \tag{4.16}$$

*whenever* $\text{dist}(\lambda^k, \Lambda)$ *is sufficiently small.*

***Proof*** For ease of presentation, in the following proof, we assume that $\text{dist}(\lambda^k, \Lambda)$ is sufficiently small so that $\bar{\vartheta}^k = \min(\varepsilon_0, \vartheta^k) = \vartheta^k$. We first show the right inequality in (4.16). Let $\hat{\lambda}$ denote the projection of $\lambda^k$ onto $\Lambda$, i.e., $\left\|\lambda^k - \hat{\lambda}\right\| = \text{dist}(\lambda^k, \Lambda)$. By optimality of $\hat{\lambda}$,

$$\text{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\hat{\lambda} - (H^k)^{-1}\nabla f(\hat{\lambda})\right) = \hat{\lambda},$$

and by the definition of $d_N^k$,

$$\left\|d_N^k\right\|_2 = \left\|\mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\lambda^k - (H^k)^{-1}\nabla f(\lambda^k)\right) - \lambda^k\right\| \tag{4.17a}$$

$$\leq \left\|\mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\lambda^k - (H^k)^{-1}\nabla f(\lambda^k)\right) - \hat{\lambda}\right\| + \left\|\hat{\lambda} - \lambda^k\right\| \tag{4.17b}$$

$$= \left\|\mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\lambda^k - (H^k)^{-1}\nabla f(\lambda^k)\right) - \mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\hat{\lambda} - (H^k)^{-1}\nabla f(\hat{\lambda})\right)\right\| + \left\|\hat{\lambda} - \lambda^k\right\|. \tag{4.17c}$$

As $H^k - \vartheta^k I$ is positive semidefnite, $\|w\|_{H^k} \geq \sqrt{\vartheta^k}\|w\|$ for all $w \in \mathbb{R}^m$. From (4.17),

$$\left\|d_N^k\right\|_2 \leq \frac{1}{\sqrt{\vartheta^k}}\left\|\mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\lambda^k - (H^k)^{-1}\nabla f(\lambda^k)\right) - \mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\hat{\lambda} - (H^k)^{-1}\nabla f(\hat{\lambda})\right)\right\|_{H^k} + \left\|\hat{\lambda} - \lambda^k\right\| \tag{4.18a}$$

$$\leq \frac{1}{\sqrt{\vartheta^k}}\left\|\lambda^k - \hat{\lambda} - (H^k)^{-1}\left(\nabla f(\lambda^k) - \nabla f(\hat{\lambda})\right)\right\|_{H^k} + \left\|\hat{\lambda} - \lambda^k\right\| \quad \text{(by (4.6}b\text{))} \tag{4.18b}$$

$$\leq \frac{1}{\vartheta^k}\left\|\nabla f(\lambda^k) - \nabla f(\hat{\lambda}) - H^k(\lambda^k - \hat{\lambda})\right\| + \left\|\hat{\lambda} - \lambda^k\right\| \tag{4.18c}$$

$$\leq \frac{1}{\vartheta^k}\left\|\nabla f(\lambda^k) - \nabla f(\hat{\lambda}) - V^k(\lambda^k - \hat{\lambda})\right\| + 2\left\|\hat{\lambda} - \lambda^k\right\| \quad \text{(by (4.5))}. \tag{4.18d}$$

As $\nabla f(\lambda)$ is strongly smooth and $V^k \in \hat{\partial}(\nabla f(\lambda^k))$, following the proof of [57, Theorem 3.5], we obtain

$$\left\|\nabla f(\lambda^k) - \nabla f(\hat{\lambda}) - V^k(\lambda^k - \hat{\lambda})\right\| \leq \underline{c}\|\lambda^k - \hat{\lambda}\|^2, \tag{4.19}$$

where $\underline{c} > 0$ is a constant. By [13, Theorems 3.4 and 3.5] and (4.15), there exists a scalar $\tilde{c} > 0$ such that

$$\vartheta^k = \left\|d_g^k(1)\right\| \geq \tilde{c}\,\mathrm{dist}(\lambda^k, \Lambda) = \tilde{c}\left\|\lambda^k - \hat{\lambda}\right\|.$$

Plug this with (4.19) into (4.18) to have

$$\left\|d_N^k\right\| \leq \frac{\underline{c}}{\vartheta^k}\left\|\lambda^k - \hat{\lambda}\right\|^2 + 2\left\|\lambda^k - \hat{\lambda}\right\| \leq \frac{\underline{c} + 2\tilde{c}}{\tilde{c}}\left\|\lambda^k - \hat{\lambda}\right\|, \tag{4.20}$$

and then the right inequality in (4.16) follows.

For the left inequality in (4.16), let $\tilde{\eta}^{k+1} \in \partial\tilde{\psi}_{\mathcal{C}}(\lambda^k + d_N^k)$ satisfy $H^k d_N^k = -\left(\nabla f(\lambda^k) + \tilde{\eta}^{k+1}\right)$. Then, for sufficiently large $k$,

$$\left\|\nabla f(\lambda^k + d_N^k) + \tilde{\eta}^{k+1}\right\| \geq \min_{\eta \in \partial\tilde{\psi}_{\mathcal{C}}(\lambda^k + d_N^k)}\left\|\nabla f(\lambda^k + d_N^k) + \eta\right\| \tag{4.21a}$$

$$= \mathrm{dist}(0, \partial\tilde{D}(\lambda^k + d_N^k)) \tag{4.21b}$$

$$\geq \bar{c}\,\mathrm{dist}(\lambda^k + d_N^k, \Lambda), \tag{4.21c}$$

where the last inequality follows since, for sufficiently large $k$, $\tilde{D}(\lambda^k + d_N^k)$ does not exceed $\tilde{D}(\lambda^0)$[4] and the error bound condition (4.15) is true at $\lambda^k + d_N^k$. Let $\tilde{\lambda}$ be the projection of

[4] In the case that $\tilde{D}(\lambda^1) < \tilde{D}(\lambda^0)$, from the proof of Theorem 4.1, we know that $\tilde{D}(\lambda^k) \leq \tilde{D}_R^k$ for all $k$, and the sequence $\{\tilde{D}_R^k\}$ is monotonically decreasing. Thus, for all sufficiently large $k$, $\tilde{D}(\lambda^k + d_N^k)$ does not

$\lambda^k + d_N^k$ onto $\Lambda$, i.e., $\left\|\lambda^k + d_N^k - \tilde{\lambda}\right\| = \text{dist}(\lambda^k + d_N^k, \Lambda)$. Using (4.21), it holds

$$\left\|\nabla f(\lambda^k + d_N^k) + \tilde{\eta}^{k+1}\right\| \geq \bar{c}\left\|\lambda^k + d_N^k - \tilde{\lambda}\right\| \geq \bar{c}\left(\left\|\lambda^k - \tilde{\lambda}\right\| - \left\|d_N^k\right\|\right). \tag{4.22}$$

Lemma 4.1 and the definition of $\vartheta^k$ ensure the boundedness of $\{H^k\}$:

$$\|H^k\|_2 \leq \rho_{\max} \tag{4.23}$$

for some scalar $\rho_{\max} > 0$. Using $d_N^k = -(H^k)^{-1}\left(\nabla f(\lambda^k) + \tilde{\eta}^{k+1}\right)$, we obtain

$$\begin{aligned}
\left\|d_N^k\right\| &= \left\|(H^k)^{-1}\left(\nabla f(\lambda^k) + \tilde{\eta}^{k+1}\right)\right\| \\
&\geq \frac{1}{\rho_{\max}}\left\|\nabla f(\lambda^k) + \tilde{\eta}^{k+1}\right\| \\
&\geq \frac{1}{\rho_{\max}}\left(\left\|\nabla f(\lambda^k + d_N^k) + \tilde{\eta}^{k+1}\right\| - \left\|\nabla f(\lambda^k) - \nabla f(\lambda^k + d_N^k)\right\|\right) \\
&\geq \frac{\bar{c}}{\rho_{\max}}\left(\left\|\lambda^k - \tilde{\lambda}\right\| - \left\|d_N^k\right\|\right) - \frac{L_f}{\rho_{\max}}\left\|d_N^k\right\| \\
&= \frac{\bar{c}}{\rho_{\max}}\left\|\lambda^k - \tilde{\lambda}\right\| - \frac{\bar{c} + L_f}{\rho_{\max}}\left\|d_N^k\right\|,
\end{aligned}$$

where the first inequality follows from (4.23), and the third inequality from (4.22) and the Lipschitz continuity of $\nabla f(\lambda)$. Therefore,

$$\left\|d_N^k\right\| \geq \frac{\bar{c}}{\rho_{\max} + L_f + \bar{c}}\left\|\lambda^k - \tilde{\lambda}\right\| \geq \frac{\bar{c}}{\rho_{\max} + L_f + \bar{c}}\text{dist}(\lambda^k, \Lambda),$$

verifying the left inequality in (4.16). □

**Lemma 4.5** *Let $\lambda^*$ be a solution of (2.11). Assume all matrices in $\hat{\partial}(\nabla f(\lambda^*))$ are positive definite. Then $\lambda^*$ is a strong minimum of the dual problem (2.11), i.e., there exists a scalar $\iota > 0$ such that*

$$\tilde{D}(\lambda) - \tilde{D}(\lambda^*) \geq \iota\|\lambda - \lambda^*\|^2, \tag{4.25}$$

*for all $\lambda$ sufficiently close to $\lambda^*$.*

***Proof*** Since $\nabla f(\lambda)$ is strongly semismooth, it follows from [14, Proposition 7.4.10] that for any $V \in \partial(\nabla f(\lambda))$,

$$f(\lambda) = f(\lambda^*) + \nabla f(\lambda^*)^T(\lambda - \lambda^*) + \frac{1}{2}(\lambda - \lambda^*)^T V(\lambda - \lambda^*) + o(\|\lambda - \lambda^*\|^2), \tag{4.26}$$

for all $\lambda$ sufficiently close to $\lambda^*$. From (4.1) and [14, Proposition 7.1.4], $\partial(\nabla f(\lambda^*)) \subseteq \hat{\partial}(\nabla f(\lambda^*))$, and $\partial(\nabla f)$ is nonempty, convex, compact, and upper semi-continuous at $\lambda^*$. This with positive definiteness of all matrices in $\hat{\partial}(\nabla f(\lambda^*))$ implies that there exists a neighborhood $\mathcal{O}_{\lambda^*}$ of $\lambda^*$ and a scalar $\iota > 0$ such that $d_\lambda^T V d_\lambda \geq \iota\|d_\lambda\|^2$ for all $d_\lambda \in \mathbb{R}^m$, all $\lambda \in \mathcal{O}_{\lambda^*}$, and all $V \in \partial(\nabla f(\lambda))$. Thus, $\frac{1}{2}(\lambda - \lambda^*)^T V(\lambda - \lambda^*) \geq \iota\|\lambda - \lambda^*\|^2$ holds for all $\lambda$ close to $\lambda^*$. Substituting this into (4.26) yields

$$f(\lambda) \geq f(\lambda^*) + \nabla f(\lambda^*)^T(\lambda - \lambda^*) + \iota\|\lambda - \lambda^*\|^2. \tag{4.27}$$

exceed $\tilde{D}(\lambda^0)$. The trivial case $\tilde{D}(\lambda^1) = \tilde{D}(\lambda^0)$ leads to $\Delta_N^0 = 0$ due to $\tilde{D}(\lambda^0) = \tilde{D}_R^0$ and the sufficient descent condition (4.9); using the positive definiteness of $H^0$ and the definition of $\Delta_N^0$, we have $d_N^0 = 0$, and by (4.8), the initial point $\lambda^0$ is optimal.

As $\tilde{\psi}_{\mathcal{C}}(\lambda)$ is convex, from (4.27), for any $\lambda$ close to $\lambda^*$,

$$\begin{aligned}\tilde{D}(\lambda) - \tilde{D}(\lambda^*) &= f(\lambda) - f(\lambda^*) + \tilde{\psi}_{\mathcal{C}}(\lambda) - \tilde{\psi}_{\mathcal{C}}(\lambda^*) \\ &\geq \nabla f(\lambda^*)^T(\lambda - \lambda^*) + \iota\|\lambda - \lambda^*\|^2 + (\eta^*)^T(\lambda - \lambda^*) \\ &= \iota\|\lambda - \lambda^*\|^2,\end{aligned} \tag{4.28}$$

where $\eta^* \in \partial\tilde{\psi}_{\mathcal{C}}(\lambda^*)$ satisfies $\nabla f(\lambda^*) + \eta^* = 0$ due to the optimality of $\lambda^*$. □

**Lemma 4.6** *Let $\{\lambda^k\}$ be generated by Algorithm 2. Assume all matrices in $\hat{\partial}(\nabla f(\lambda^*))$ are positive definite, where $\lambda^*$ is a solution of (2.11). If $\lambda^0$ is sufficiently close to $\lambda^*$, then*

$$\|\lambda^k + d_N^k - \lambda^*\| = O(\|\lambda^k - \lambda^*\|^2).$$

***Proof*** By Theorem 4.1 and Lemma 4.5, the sequence $\{\lambda^k\}$ converges to the unique solution $\lambda^*$ of (2.11). By [14, Lemma 7.5.2], both $\{H^k\}$ and $\{(H^k)^{-1}\}$ are bounded for all $k$. Assume $\|(H^k)^{-1}\|_2 \leq \frac{1}{c_3}$ for a scalar $c_3 > 0$, which implies that $\|w\|_{H^k} \geq \sqrt{c_3}\|w\|$ for all $w \in \mathbb{R}^m$. Similar to the proof of Lemma 4.4, by the definition of $d_N^k$,

$$\|\lambda^k + d_N^k - \lambda^*\| \leq \frac{1}{\sqrt{c_3}}\|\lambda^k + d_N^k - \lambda^*\|_{H^k} \tag{4.29a}$$

$$= \frac{1}{\sqrt{c_3}}\left\|\mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\lambda^k - (H^k)^{-1}\nabla f(\lambda^k)\right) - \mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}^{H^k}\left(\lambda^* - (H^k)^{-1}\nabla f(\lambda^*)\right)\right\|_{H^k} \tag{4.29b}$$

$$\leq \frac{1}{\sqrt{c_3}}\left\|\lambda^k - \lambda^* - (H^k)^{-1}\left(\nabla f(\lambda^k) - \nabla f(\lambda^*)\right)\right\|_{H^k} \tag{4.29c}$$

$$\leq \frac{1}{c_3}\left\|\nabla f(\lambda^k) - \nabla f(\lambda^*) - V^k\left(\lambda^k - \lambda^*\right)\right\| + \frac{\vartheta^k}{c_3}\|\lambda^k - \lambda^*\|, \tag{4.29d}$$

where the second inequality follows from (4.6b), and the last one from the definition of $H^k$. As $\lambda^k \to \lambda^*$, and $\mathrm{Prox}_{\tilde{\psi}_{\mathcal{C}}}(\cdot)$ and $f$ are Lipschitz continuous, it holds

$$\vartheta^k = \left\|d_g^k(1)\right\| = O(\|\lambda^k - \lambda^*\|). \tag{4.30}$$

Putting (4.30) and (4.19) into (4.29) yields the desired result. □

Our next lemma establishes that the full proximal semismooth Newton step $d_N^k$ will be executed asymptotically, which provides a guarantee for the local quadratic convergence given in Theorem 4.2.

**Lemma 4.7** *Let $\{\lambda^k\}$ be generated by Algorithm 2, in which $\lambda^{k+1}$ is determined by $\lambda^{k+1} = \lambda^k + \alpha^k d_N^k$. Assume all matrices in $\hat{\partial}(\nabla f(\lambda^*))$ are positive definite, where $\lambda^*$ is a solution of (2.11). Then $\alpha^k = 1$ for all sufficiently large $k$.*

***Proof*** According to the procedure of Algorithm 2, it suffices to show that for all sufficiently large $k$,

$$\tilde{D}(\lambda^k + d_N^k) \leq \tilde{D}(\lambda^k) - \xi\Delta_N^k, \tag{4.31}$$

where $\xi \in (0, 1/2)$. For this, by [14, Proposition 7.4.10] and Lemma 4.6, we have

$$f(\lambda^k) = f(\lambda^*) + \nabla f(\lambda^*)^T(\lambda^k - \lambda^*) + \frac{1}{2}(\lambda^k - \lambda^*)^T V^k(\lambda^k - \lambda^*) + o(\|\lambda^k - \lambda^*\|^2) \tag{4.32}$$

and

$$f(\lambda^k + d_N^k) = f(\lambda^*) + \nabla f(\lambda^*)^T(\lambda^k + d_N^k - \lambda^*) + o(\|\lambda^k - \lambda^*\|^2), \tag{4.33}$$

where $V^k \in \hat{\partial}(\nabla f(\lambda^*))$ is defined in (4.5). Let $\eta^* = -\nabla f(\lambda^*)$. Due to the optimality of $\lambda^*$, $\eta^* \in \partial\tilde{\psi}_{\mathcal{C}}(\lambda^*)$, and the subgradients gives

$$\tilde{\psi}_{\mathcal{C}}(\lambda^*) - \tilde{\psi}_{\mathcal{C}}(\lambda^k) \leq (\eta^*)^T(\lambda^* - \lambda^k). \tag{4.34}$$

Similarly,

$$\tilde{\psi}_{\mathcal{C}}(\lambda^k + d_N^k) - \tilde{\psi}_{\mathcal{C}}(\lambda^*) \leq (\tilde{\eta}^{k+1})^T(\lambda^k + d_N^k - \lambda^*), \tag{4.35}$$

where $\tilde{\eta}^{k+1} \in \partial\tilde{\psi}_{\mathcal{C}}(\lambda^k + d_N^k)$ satisfies $H^k d_N^k = -\left(\nabla f(\lambda^k) + \tilde{\eta}^{k+1}\right)$. By (4.32)-(4.35),

$$\begin{aligned}
&\tilde{D}(\lambda^k + d_N^k) - \tilde{D}(\lambda^k) + \xi(d_N^k)^T H^k d_N^k \\
\leq&(\tilde{\eta}^{k+1} + \nabla f(\lambda^*))^T(\lambda^k + d_N^k - \lambda^*) - \frac{1}{2}(\lambda^k - \lambda^*)^T V^k(\lambda^k - \lambda^*) + \xi(d_N^k)^T H^k d_N^k \\
&+ o(\|\lambda^k - \lambda^*\|^2) \\
=&(\nabla f(\lambda^*) - \nabla f(\lambda^k) - H^k d_N^k)^T(\lambda^k + d_N^k - \lambda^*) - \frac{1}{2}(\lambda^k - \lambda^*)^T V^k(\lambda^k - \lambda^*) \\
&+ \xi(d_N^k)^T H^k d_N^k + o(\|\lambda^k - \lambda^*\|^2).
\end{aligned} \tag{4.36}$$

where the last equality follows from $\tilde{\eta}^{k+1} = -\left(\nabla f(\lambda^k) + H^k d_N^k\right)$. By Lemma 4.6 and Lipschitz continuity of $\nabla f(\lambda)$, we have

$$\begin{aligned}
&\left(\nabla f(\lambda^*) - \nabla f(\lambda^k)\right)^T\left(\lambda^k + d_N^k - \lambda^*\right) \\
&= o(\|\lambda^k - \lambda^*\|^2).
\end{aligned} \tag{4.37}$$

By Lemmas 4.4 and 4.6, and (4.23), it holds that

$$(H^k d_N^k)^T(\lambda^k + d_N^k - \lambda^*) = o(\|\lambda^k - \lambda^*\|^2). \tag{4.38}$$

Again, using Lemmas 4.4 and 4.6, we have that

$$\begin{aligned}
&-\frac{1}{2}\left(\lambda^k - \lambda^*\right)^T V^k\left(\lambda^k - \lambda^*\right) \\
&= -\frac{1}{2}\left(\lambda^k + d_N^k - \lambda^*\right)^T V^k\left(\lambda^k - \lambda^*\right) + \frac{1}{2}(d_N^k)^T V^k\left(\lambda^k - \lambda^*\right) \\
&= -\frac{1}{2}\left(\lambda^k + d_N^k - \lambda^*\right)^T V^k\left(\lambda^k - \lambda^*\right) + \frac{1}{2}(d_N^k)^T V^k(\lambda^k + d_N^k - \lambda^*) - \frac{1}{2}(d_N^k)^T V^k d_N^k \\
&= -\frac{1}{2}(d_N^k)^T V^k d_N^k + o\left(\|\lambda^k - \lambda^*\|^2\right).
\end{aligned} \tag{4.39}$$

Due to $\vartheta^k \to 0$, we assume, without loss of generality, that $\bar{\vartheta}^k = \min(\varepsilon_0, \vartheta^k) = \vartheta^k$ for all $k$. By (4.30), and Lemmas 4.4 and 4.6,

$$\begin{aligned}
-\frac{1}{2}(d_N^k)^T V^k d_N^k &= -\frac{1}{2}(d_N^k)^T H^k d_N^k + \frac{1}{2}\vartheta^k\|d_N^k\|^2 \\
&= -\frac{1}{2}(d_N^k)^T H^k d_N^k + o\left(\|\lambda^k - \lambda^*\|^2\right).
\end{aligned} \tag{4.40}$$

Plugging (4.37)–(4.40) into (4.36) gives

$$\tilde{D}(\lambda^{k+1}) - \tilde{D}(\lambda^k) + \xi(d_N^k)^T H^k d_N^k$$

$$\leq (\xi - \frac{1}{2})(d_N^k)^T H^k d_N^k + o(\|\lambda^k - \lambda^*\|^2).$$

Combine it with $\|(H^k)^{-1}\| \leq \frac{1}{c_3}$ and Lemma 4.4 to have (4.31). □

**Theorem 4.2** *Let $\{\lambda^k\}$ be generated by Algorithm 2, in which $\lambda^{k+1}$ is determined by $\lambda^{k+1} = \lambda^k + \alpha^k d_N^k$. Assume all matrices in $\hat{\partial}(\nabla f(\lambda^*))$ are positive definite, where $\lambda^*$ is a solution of (2.11). If $\lambda^0$ is sufficiently close to $\lambda^*$, then the full sequence $\{\lambda^k\}$ converges to $\lambda^*$ quadratically, i.e., for all sufficiently large $k$,*

$$\|\lambda^{k+1} - \lambda^*\| = O(\|\lambda^k - \lambda^*\|^2).$$

***Proof*** The desired conclusion follows from Lemmas 4.6 and 4.7. □

## 5 Numerical Experiments

In this section, we carry out numerical testings on both synthetic and real data sets for the following two tasks:

(a) verification of the effectiveness of DGASS in improving the basic PGA and PSNA, and
(b) demonstration of the efficiency of PGA-DGASS and PSNA-DGASS.

For the second task, we will compare PGA-DGASS and PSNA-DGASS with Gurobi[5], one of the most popular commercial solvers, and IPOPT[6], a general optimizer for large-scale nonlinear optimization using interior point methods. All our numerical testings are conducted in MATLAB 2022a on a PC with 16 RAM and 6-core Intel(R) Core(TM) i7-9750 @2.60 GHz processor.

For the stopping rule, we define the relative residual

$$R(\lambda^k) := \frac{\left\| d_g^k(1) \right\|_\infty}{1 + \left\| d_g^k(1) \right\|_\infty + \|l\|_\infty + \|u\|_\infty}, \tag{5.1}$$

which measures the optimality of $\lambda^k$. We terminate PGA-DGASS, PSNA-DGASS, PGA and PSNA whenever $R(\lambda^k) \leq \varepsilon := 10^{-8}$. Relevant parameters of PGA-DGASS, PSNA-DGASS, PGA and PSNA are set as follows:

$$\xi = 0.45, \qquad \sigma = 0.5, \qquad \zeta = 0.9, \qquad p = 5.$$

### 5.1 Effectiveness of DGASS

For our first task (a), we compare PGA-DGASS, PSNA-DGASS with the basic PGA and PSNA on synthetic problems randomly generated in the following way:

$A$ : `A=randi([-10,10], [m,n])`
$y$ : `y=2*rand(n,1)-1`
$l$ : `l=min(A*y)*rand(m,1)`
$u$ : `u=max(A*y)*rand(m,1)`

[5] https://www.gurobi.com/.

[6] https://coin-or.github.io/Ipopt/.

**Table 1** CPU time (in seconds) consumed by PGA, PSNA, PGA-DGASS and PSNA-DGASS

| (m, n) | PGA | PSNA | PGA-DGASS | PSNA-DGASS | (m, n) | PGA | PSNA | PGA-DGASS | PSNA-DGASS |
|---|---|---|---|---|---|---|---|---|---|
| (100, 100) | 2.2e-02 | 1.7e-02 | 1.5e-02 | **1.2e-02** | (1000, 100) | 5.1e-02 | 2.8e-02 | 2.2e-02 | **1.9e-02** |
| (100, 200) | 1.1e-02 | 7.0e-03 | 9.0e-03 | **6.0e-03** | (1000, 200) | 1.2e-01 | 4.0e-02 | **2.6e-02** | **3.2e-02** |
| (100, 500) | 7.0e-03 | 6.0e-03 | 6.0e-03 | **5.0e-03** | (1000, 500) | 2.5e-01 | 1.1e-01 | 5.3e-02 | **3.7e-02** |
| (100, 800) | 1.0e-02 | 8.0e-03 | 6.0e-03 | **6.0e-03** | (1000, 800) | 3.2e-01 | 1.4e-01 | 9.2e-02 | **5.5e-02** |
| (100, 1000) | 9.0e-03 | 5.0e-03 | 6.0e-03 | **4.0e-03** | (1000, 1000) | 3.5e-01 | 1.6e-01 | 8.7e-02 | **6.1e-02** |
| (100, 2000) | 2.2e-02 | 1.1e-02 | 1.1e-02 | **1.0e-02** | (1000, 2000) | 3.9e-01 | 2.9e-01 | 1.9e-01 | **1.3e-01** |
| (200, 100) | 1.9e-02 | 1.2e-02 | **8.0e-03** | **9.0e-03** | (2000, 100) | 2.4e-01 | 6.3e-02 | 5.2e-02 | **3.3e-02** |
| (200, 200) | 1.6e-02 | 1.0e-02 | 9.0e-03 | **8.0e-03** | (2000, 200) | 3.8e-01 | 1.0e-01 | 8.3e-02 | **3.8e-02** |
| (200, 500) | 1.4e-02 | 1.4e-02 | 8.0e-03 | **8.0e-03** | (2000, 500) | 4.9e-01 | 1.9e-01 | 9.4e-02 | **5.2e-02** |
| (200, 800) | 3.2e-02 | 2.0e-02 | 1.8e-02 | **1.5e-02** | (2000, 800) | 5.8e-01 | 2.3e-01 | 1.8e-01 | **9.5e-02** |
| (200, 1000) | 4.6e-02 | **1.6e-02** | 1.6e-02 | **1.7e-02** | (2000, 1000) | 8.2e-01 | 3.8e-01 | 2.6e-01 | **1.6e-01** |
| (200, 2000) | 6.7e-02 | **2.9e-02** | 3.2e-02 | **3.1e-02** | (2000, 2000) | 9.1e-01 | 4.7e-01 | 4.1e-01 | **2.4e-01** |
| (10000, 100) | 1.7e+00 | 3.5e-01 | 2.6e-01 | **1.5e-01** | (100000, 100) | 2.4e+01 | 5.1e+00 | 6.8e+00 | **1.8e+00** |
| (10000, 200) | 1.7e+00 | 7.3e-01 | 3.3e-01 | **2.3e-01** | (100000, 200) | 2.7e+01 | 8.3e+00 | 6.5e+00 | **3.0e+00** |
| (10000, 500) | 4.0e+00 | 1.7e+00 | 1.0e+00 | **6.9e-01** | (100000, 500) | 6.3e+01 | 2.1e+01 | 1.4e+01 | **1.0e+01** |
| (10000, 800) | 5.4e+00 | 2.4e+00 | 1.8e+00 | **1.2e+00** | (100000, 800) | 7.3e+01 | 2.9e+01 | 2.3e+01 | **1.6e+01** |
| (10000, 1000) | 5.5e+00 | 2.3e+00 | 2.2e+00 | **1.3e+00** | (100000, 1000) | 1.2e+02 | 4.1e+01 | 3.2e+01 | **2.9e+01** |
| (10000, 2000) | 7.3e+00 | 3.4e+00 | 2.9e+00 | **1.8e+00** | (10000, 2000) | 1.9e+02 | 7.1e+01 | 6.2e+01 | **4.4e+01** |
| (20000, 100) | 4.6e+00 | 9.9e-01 | 9.1e-01 | **3.8e-01** | (200000, 100) | 6.2e+01 | 1.2e+01 | 1.5e+01 | **5.0e+00** |
| (20000, 200) | 5.0e+00 | 1.4e+00 | 1.2e+00 | **6.5e-01** | (200000, 200) | 1.0e+02 | 1.7e+01 | 2.2e+01 | **7.0e+00** |
| (20000, 500) | 1.0e+01 | 3.9e+00 | 2.6e+00 | **1.6e+00** | (200000, 500) | 1.5e+02 | 4.5e+01 | 3.2e+01 | **2.0e+01** |
| (20000, 800) | 1.3e+01 | 5.0e+00 | 4.6e+00 | **3.2e+00** | (200000, 800) | 2.0e+02 | 7.8e+01 | 5.3e+01 | **2.8e+01** |
| (20000, 1000) | 1.5e+01 | 6.8e+00 | 5.2e+00 | **3.6e+00** | (200000, 1000) | 3.1e+02 | 9.7e+01 | 7.1e+01 | **6.3e+01** |
| (20000, 2000) | 1.8e+01 | 8.5e+00 | 7.9e+00 | **5.9e+00** | (200000, 2000) | 4.4e+02 | 1.8e+02 | 1.3e+02 | **9.6e+01** |

The one with the smallest value (CPU time) in each row is highlighted in bold

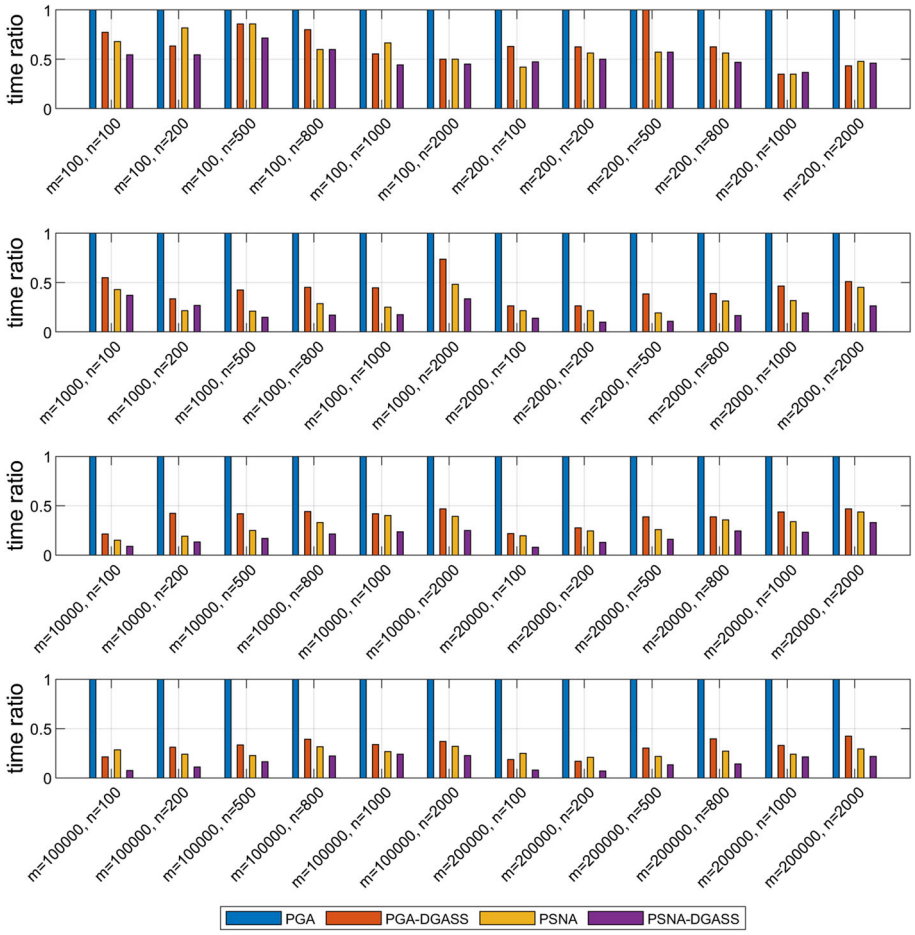



**Fig. 1** Scaled CPU time profiles of PGA, PGA-DGASS, PSNA and PSNA-DGASS

where `randi`, `rand`, `min`, `max` are MATLAB built-in functions.[7] Also, we choose $\underline{b}_i = 0, \bar{b}_i = +\infty, i = 1, 2, \ldots, n$ in our numerical experiments. It should be pointed out that the choice $\bar{b}_i = +\infty$ does not invalidate the convergence of PGA-DGASS and PSNA-DGASS. By varying $m$ from 100 to 200000 and $n$ from 100 to 2000, we report on detailed consumed CPU time from algorithms PGA-DGASS, PSNA-DGASS, PGA and PSNA in Table 1. Also, for a clear comparison, we scale CPU time in [0,1] and plot it in Fig. 1.

From Fig. 1, we can see that (1) PSNA-DGASS has the best performance, which is followed by PSNA, (2) DGASS is able to speed up both PGA and PSNA, and (3) the improvement from DGASS gets more manifest, especially when $m$ is much larger than $n$.

Besides the efficiency of the algorithms revealed by the consumed CPU time, we next investigate the convergence indicated in the relative residual $R(\lambda^k)$ with respect to iteration

[7] The functions `randi` and `rand` are used to generate uniformly distributed pseudo-random integers and uniformly distributed pseudo-random numbers, respectively, and `min(v)` and `max(v)` compute the minimum and maximum values of $v$, respectively. Also, in order to ensure $y = 0$ is a feasible solution for (1.1), the generated random $A$ and $y$ satisfy `min(A*y)<0` and `max(A*y)>0`.

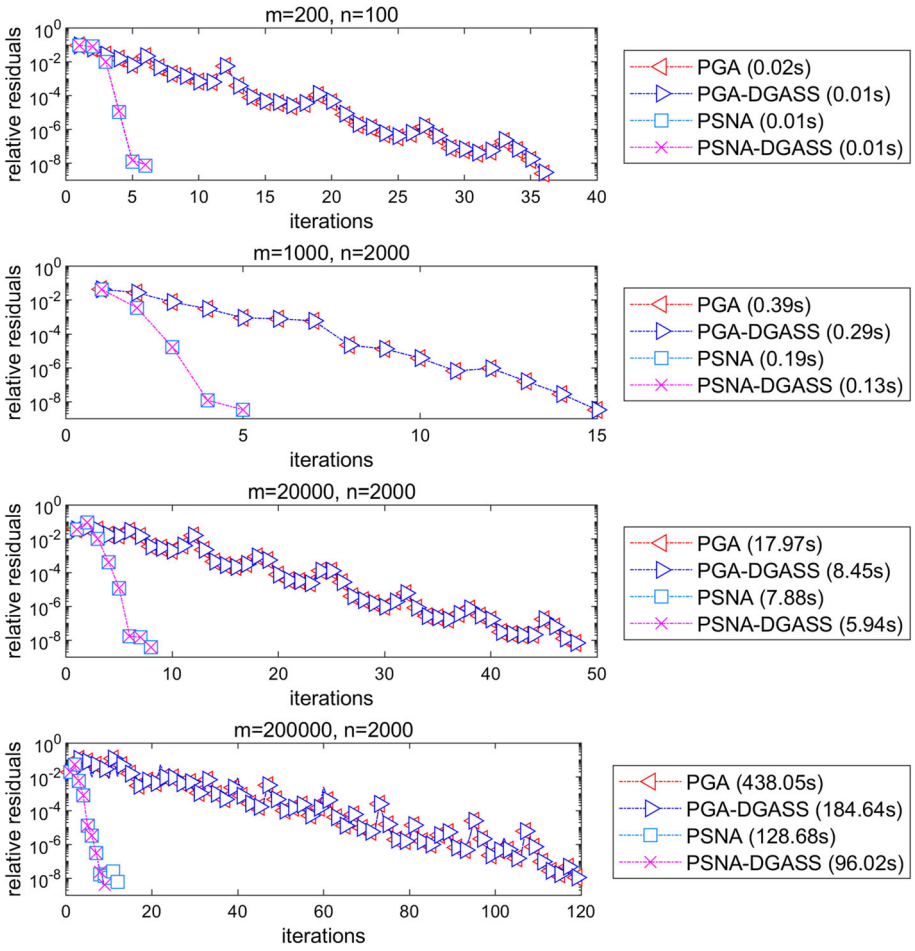



**Fig. 2** Relative residuals vs the number of iterations $k$

number $k$. For this purpose, we choose four instances with different $m$ and $n$, and then plot the relative residuals *vs* number of iterations in Fig. 2. It can be observed that $R(\lambda^k)$ from the second-order methods, PSNA and PSNA-DGASS, decrease faster than that from PGA and PGA-DGASS. Also, the convergence history curves of $R(\lambda^k)$ of PGA and PGA-DGASS are roughly the same, and so are PNSA and PNSA-DGASS. Taking the CPU times reported in Table 1 and Fig. 1 into account, it seems that the acceleration in PGA-DGASS and PNSA-DGASS is mainly brought by the utilization of DGASS. We will give more detailed information on how DGASS gradually screens zeros in the solution to accelerate PGA and PSNA in our following testing.

To see the effectiveness of DGASS in screening zeros for PGA and PSNA, we plot Fig. 3 and show how the *screening ratio* behaves against the number of iterations $k$, where the *screening ratio* is defined as the ratio between the number of the identified zeros and the total number of zeros in the solution. From Fig. 3, we can see that almost all zeros are identified and screened out before the optimal solution is finally computed. The iteration can

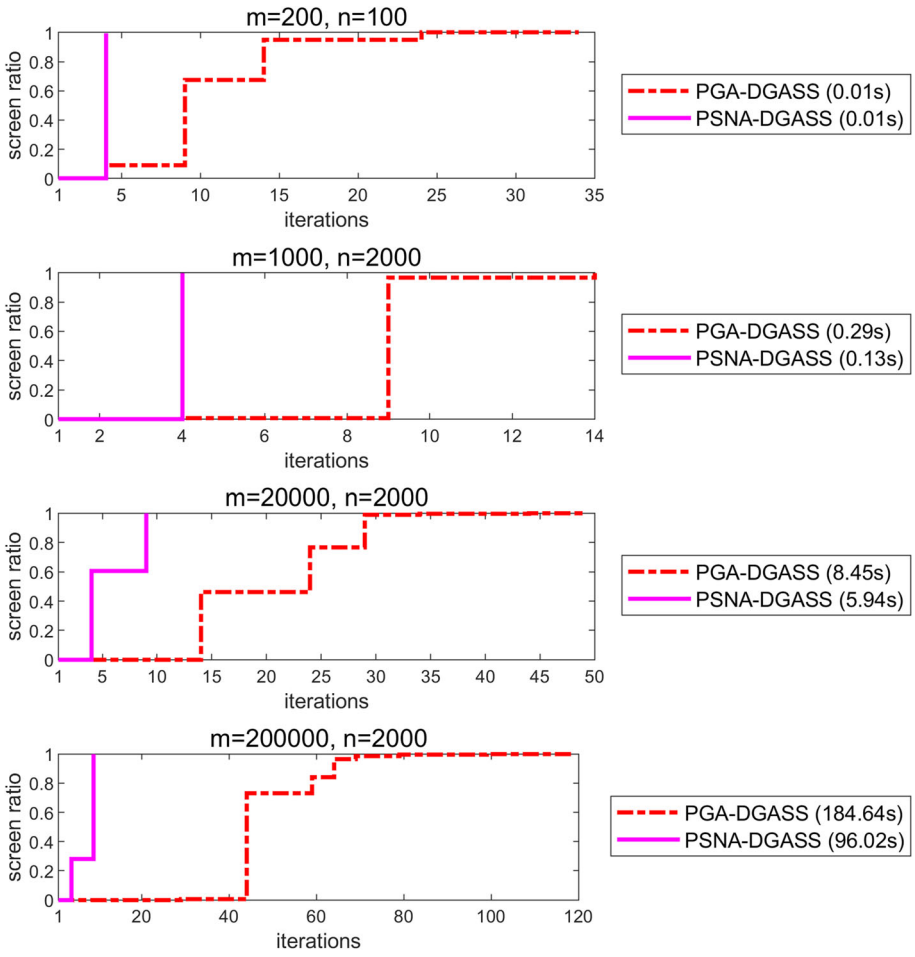



**Fig. 3** Screening ratios vs the number of iterations $k$

be significantly speeded up as long as the exact zeros in the optimal solution can be identified in the early stage.

### 5.2 Comparison with Gurobi and IPOPT

We now focus on our second task to verify the performance of PGA-DGASS and PNSA-DGASS by comparing it with Gurobi-default (the default method), Gurobi-primal (the primal simplex method), Gurobi-dual (the dual simplex method), and IPOPT. The numerical experiments are conducted on both synthetic and real data sets. To obtain solutions with comparable precisions, we set the tolerance $\varepsilon = 10^{-9}$ for PGA-DGASS and PNSA-DGASS. Default parameters are used for other solvers, which include the feasibility and optimality tolerances. Also, all algorithms will be terminated if the consumed CPU time reaches 1500 seconds.

For comparison, we use the performance profiles of Dolan and Moré [12]. Let $t_{p,c}$ denote CPU time that the solver $c$ consumes in solving the problem $p$. The performance ratio is

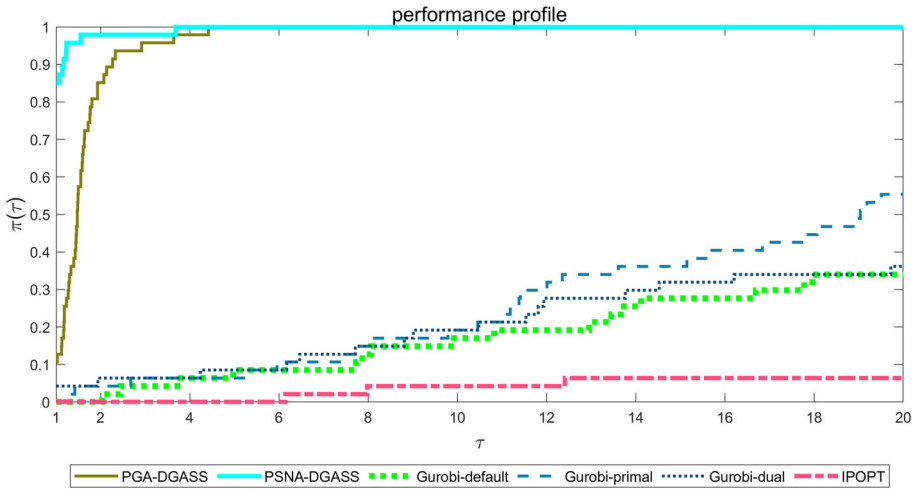



**Fig. 4** Performance profiles of PGA-DGASS, PSNA-DGASS, Gurobi-default, Gurobi-primal, Gurobi-dual, and IPOPT on synthetic data

defined as

$$r_{p,c} = \frac{t_{p,c}}{t_p^*},$$

where $t_p^*$ denotes the least CPU time among the six solvers. If an algorithm fails in a problem, the ratio $r_{p,c}$ is set to be a large number $M = 10^5$. Let $\mathcal{P}$ denote the set of $n_p$ test problems; then the distribution function $\pi_c(\tau)$ for each solver $c$ is defined as

$$\pi_c(\tau) = \frac{size\{p \in \mathcal{P} : r_{p,c} \leq \tau\}}{n_p}, \quad \tau \geq 1.$$

which serves as a performance metric for the solver $c$. The performance profile of each solver $c$ is generated by plotting the corresponding distribution function $\pi_c(\tau)$.

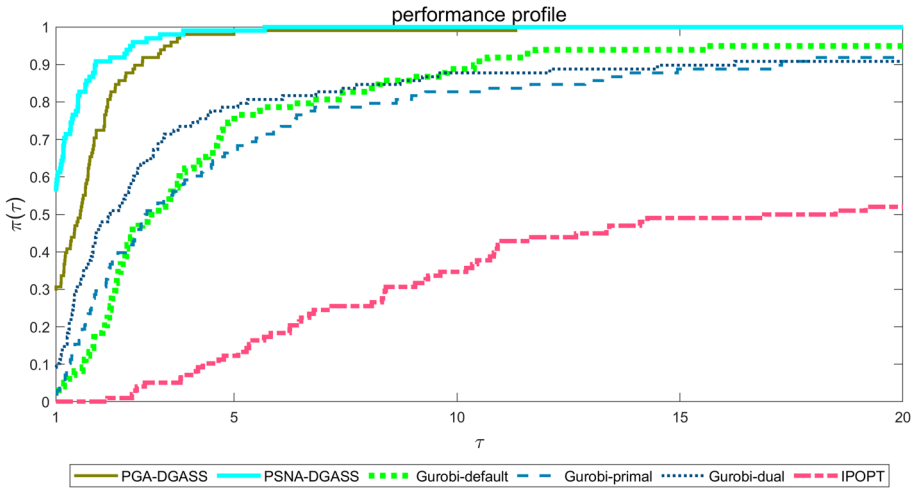
### 5.2.1 Results from a Synthetic Data Set

We first test PGA-DGASS, PNSA-DGASS, Gurobi-default, Gurobi-primal, Gurobi-dual, and IPOPT on synthetic data. Similar to Sect. 5.1, we generate a set of 240 test instances with $m$ ranging from 100 to 200000 and $n$ ranging from 100 to 2000. The performance profiles are given in Fig. 4.

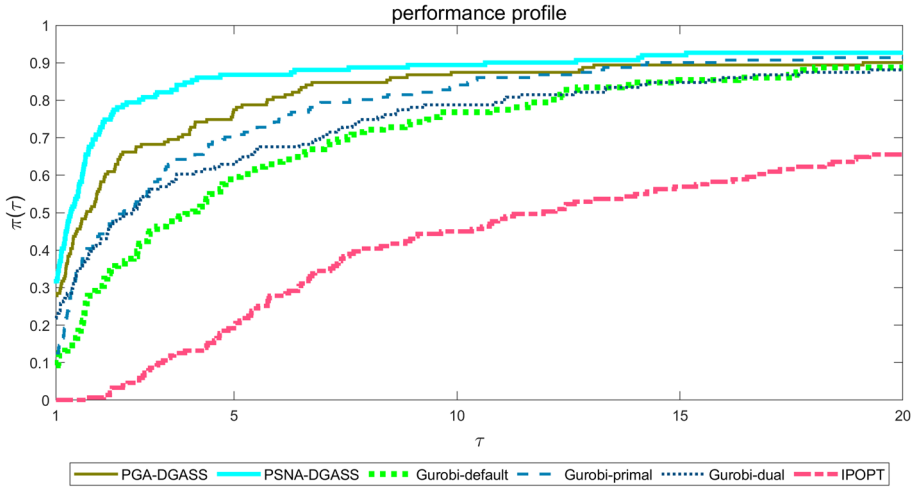
Among all test problems, we noticed that PGA-DGASS, PNSA-DGASS, Gurobi-primal, Gurobi-dual succeed in solving all instances, but Gurobi-default and IPOPT fail to solve two instances with ($m = 100000, n \geq 500$) and ($m = 200000, n \geq 100$) as they run out of memory or exceed the maximum allowed CPU time. Figure 4 illustrates that PSNA-DGASS has the best numerical performance in terms of CPU time, followed by PGA-DGASS. Precisely, PSNA-DGASS outperforms other five solvers in about 85% of test instances, while PGA-DGASS outperforms others in about 13% of test instances.

**Table 2** Information about test problems from NETLIB-LP and MIPLIB

| Data set | # of problems | min(m) | max(m) | min(n) | max(n) | min(NNZ) | max(NNZ) | min $\frac{nnz(A)}{numel(A)}$ | max $\frac{nnz(A)}{numel(A)}$ |
|---|---|---|---|---|---|---|---|---|---|
| NETLIB-LP | 98 | 24 | 105127 | 531 | 379350 | 1590 | 1567800 | 2.20e-05 | 0.53333 |
| MIPLIB | 151 | 32 | 2897380 | 214 | 1429098 | 1200 | 27329856 | 6.21e-06 | 0.32646 |




(a) Performance profiles on NETLIB-LP data set




(b) Performance profiles on MIPLIB data set

**Fig. 5** Performance profiles of PGA-DGASS, PSNA-DGASS, Gurobi-default, Gurobi-primal, Gurobi-dual, and IPOPT on NETLIB-LP and MIPLIB

### 5.2.2 Results from Real Data Sets

To extend our numerical testing, we next apply algorithms on two real data sets: NETLIP-LP[8] and MIPLIB 2017[9]. NETLIP-LP data set is a collection of real-life linear programmings from a variety of sources, and MIPLIB 2017 is a standard test set for mixed integer optimizers. In our testing, data matrices $A$ are selected from NETLIP-LP and MIPLIB 2017, and all other vectors $y, l, u, \underline{b}, \bar{b}$ are generated as in Sect. 5.1. Particularly, for $A$, we choose all

[8] https://www.cuter.rl.ac.uk/Problems/netlib.shtml.

[9] http://miplib.zib.de/.

matrices in NETLIP-LP and MIPLIB 2017 satisfying $m+n \geq 1000$, which results in 98 and 151 instances, respectively. Detailed information about the test problems is given in Table 2, where *NNZ* and $\frac{nnz(A)}{numel(A)}$ represent the number of non-zeros and the ratio of *NNZ* to the total number of the elements in $A$, respectively. Figure 5 provides the performance profiles of all solvers on these data. Again, it shows that PSNA-DGASS performs the best on both NETLIP-LP and MIPLIB data sets in terms of CPU time, while PGA-DGASS ranks the second.

## 6 Conclusion

As a fundamental problem in many applications, the polyhedral projection problem concerns computing the closest point in a polyhedron. In this paper, we have made efforts for numerically solving the polyhedral projection problem. Our proposed methods make use of the fact that the underlying inactive and active constraints at the optimal solution for the primal are related with the zero and nonzero elements of the solution of the dual, respectively. Based on this connection, we proposed a safe screening strategy, DGASS, working on the dual problem which detects zeros and signs of nonzeros of the optimal dual solution dynamically along iterations. By applying DGASS to PGA and PSNA, we proposed PGA-DGASS, PSNA-DGASS, respectively. Global convergence of both methods are established under some mild conditions, and moreover, local quadratic convergence of PSNA-DGASS is also analyzed. The numerical performance of PGA-DGASS and PSNA-DGASS is evaluated on various synthetic and real data, and the comparison with solvers of Gurobi and IPOPT demonstrates the efficiency in solving the polyhedral projection problem.

**Acknowledgements** The authors are grateful to the associate editor and the two anonymous referees for their valuable comments and suggestions.

**Data Availability** Enquiries about data availability should be directed to the authors.

## Declarations

**Conflict of interest** The authors have not disclosed any competing interests.

## References

1. Adler, I., Hu, Z.T. and Lin, T.: New proximal Newton-type methods for convex optimization. In 59th IEEE conference on decision and control, pp. 4828–4835. IEEE, (2020)
2. Bagirov, A., Karmitsa, N., Mäkelä, M.M.: Introduction to nonsmooth optimization: theory, practice and software, vol. 12. Springer, (2014)
3. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)
4. Beck, A.: First-order methods in optimization. SIAM, (2017)
5. Becker, S., Fadili, J.: A quasi-Newton proximal splitting method. In: Advances in neural information processing systems **25**, 2618–2626 (2012)
6. Becker, S., Fadili, J., Ochs, P.: On quasi-Newton forward-backward splitting: proximal calculus and convergence. SIAM J. Optim. **29**(4), 2445–2481 (2019)
7. Birgin, E.G., Martínez, J.M., Raydan, M.: Spectral projected gradient methods: review and perspectives. J. Stat. Softw. **60**, 1–21 (2014)
8. Boyd, S., Diaconis, P., Xiao, L.: Fastest mixing Markov chain on a graph. SIAM Rev. **46**(4), 667–689 (2004)

9. Censor, Y.: Computational acceleration of projection algorithms for the linear best approximation problem. Linear Algebra Appl. **416**(1), 111–123 (2006)
10. Clarke, F.H.: Optimization and Nonsmooth Analysis. SIAM, (1990)
11. Dai, Y.H.: Alternate step gradient method. Optimization **52**(4–5), 395–415 (2003)
12. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
13. Drusvyatskiy, D., Lewis, A.S.: Error bounds, quadratic growth, and linear convergence of proximal methods. Math. Oper. Res. **43**(3), 919–948 (2018)
14. Facchinei, F., Pang, J.-S.: Finite-dimensional Variational Inequalities and Complementarity Problems. Springer, (2003)
15. Fercoq, O., Gramfort, A., Salmon, J.: Mind the duality gap: safer rules for the lasso. In: International conference on machine learning, pp. 333–342. PMLR, (2015)
16. Fletcher, R.: On the Barzilai-Borwein method. In: Optimization and control with applications, pp. 235–256. Springer, (2005)
17. Gabidullina, Z.: A linear separability criterion for sets of Euclidean space. J. Optim. Theory Appl. **158**(1), 145–171 (2013)
18. Gao, B., Son, N.T., Absil, P.-A., Stykel, T.: Riemannian optimization on the symplectic Stiefel manifold. SIAM J. Optim. **31**(2), 1546–1575 (2021)
19. Ghaoui, L.E., Viallon, V., Rabbani, T.: Safe feature elimination for the lasso and sparse supervised learning problems. arXiv preprint arXiv:1009.4219, (2010)
20. Gotoh, J., Takeda, A., Tono, K.: DC formulations and algorithms for sparse optimization problems. Math. Program. **169**(1), 141–176 (2018)
21. Hager, W.W., Zhang, H.: A new active set algorithm for box constrained optimization. SIAM J. Optim. **17**(2), 526–557 (2006)
22. Hager, W.W., Zhang, H.: An active set algorithm for nonlinear optimization with polyhedral constraints. Sci. China Math. **59**(8), 1525–1542 (2016)
23. Hager, W.W., Zhang, H.: Projection onto a polyhedron that exploits sparsity. SIAM J. Optim. **26**(3), 1773–1798 (2016)
24. Hiriart-Urruty, J.-B., Lemaréchal, C.: Fundamentals of Convex Analysis. Springer Science and Business Media, (2004)
25. Hu, J., Liu, X., Wen, Z.W., Yuan, Y.X.: A brief introduction to manifold optimization. J. Oper. Res. Soc. China **8**(2), 199–248 (2020)
26. Huang, Y.-K., Dai, Y.-H., Liu, X.-W.: Equipping the Barzilai-Borwein method with the two dimensional quadratic termination property. SIAM J. Optim. **31**(4), 3068–3096 (2021)
27. Huang, Y.-K., Dai, Y.-H., Liu, X.-W., Zhang, H.: On the acceleration of the Barzilai-Borwein method. Comput. Optim. Appl. **81**(3), 717–740 (2022)
28. Laurent, C.: Fast projection onto the simplex and the l1 ball. Math. Program. **158**, 575–585 (2016)
29. Lee, J.D., Sun, Y., Saunders, M.A.: Proximal Newton-type methods for minimizing composite functions. SIAM J. Optim. **24**(3), 1420–1443 (2014)
30. Li, X., Sun, D., Toh, K.-C.: A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems. SIAM J. Optim. **28**(1), 433–458 (2018)
31. Mifflin, R.: Semismooth and semiconvex functions in constrained optimization. SIAM J. Control. Optim. **15**(6), 959–972 (1977)
32. Nakayama, S., Narushima, Y., Yabe, H.: Inexact proximal memoryless quasi-Newton methods based on the Broyden family for minimizing composite functions. Comput. Optim. Appl. **79**(1), 127–154 (2021)
33. Ndiaye, E., Fercoq, O., Gramfort, A., Salmon, J.: Gap safe screening rules for sparse multi-task and multi-class models. Adv. Neural Inf. Process. Syst. **28**, 811–819 (2015)
34. Ndiaye, E., Fercoq, O., Gramfort, A., Salmon, J.: Gap safe screening rules for Sparse-Group Lasso. Adv. Neural Inf. Process. Syst. **29**, 388–396 (2016)
35. Ndiaye, E., Fercoq, O., Gramfort, A., Salmon, J.: Gap safe screening rules for sparsity enforcing penalties. J. Mach. Learn. Res. **18**(1), 4671–4703 (2017)
36. Nesterov, Y.E.: A method for solving a convex programming problem with convergence rate $O(1/k^2)$. In Doklady A.N. (Eds.) Russian Academy of Sciences, vol. 269, pp. 543–547. (1983)
37. Ogawa, K., Suzuki, Y., Takeuchi, I.: Safe screening of non-support vectors in pathwise SVM computation. In: International conference on machine learning, pp. 1382–1390. PMLR, (2013)
38. Olbrich, J.: Screening rules for convex problems. Master's thesis, ETH-Zürich, (2015)
39. Patriksson, M.: Cost approximation: a unified framework of descent algorithms for nonlinear programs. SIAM J. Optim. **8**(2), 561–582 (1998)
40. Panagiotis, P. and Alberto B.: Proximal Newton methods for convex composite optimization. In: 52nd IEEE conference on decision and control, pp 2358–2363. IEEE, (2013)

41. Qi, L., Sun, J.: A nonsmooth version of Newton's method. Math. Program. **58**(1), 353–367 (1993)
42. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. **7**(1), 26–33 (1997)
43. Rockafellar, T.: Convex analysis. Princeton University Press, (1970)
44. Shen, C., Liu, X.: Solving nonnegative sparsity-constrained optimization via DC quadratic-piecewise-linear approximations. J. Global Optim. **81**(4), 1019–1055 (2021)
45. Shen, C., Wang, Y., Xue, W., Zhang, L.-H.: An accelerated active-set algorithm for a quadratic semidefinite program with general constraints. Comput. Optim. Appl. **78**(1), 1–42 (2021)
46. Shen, C., Xue, W., Zhang, L.-H., Wang, B.: An active-set proximal-Newton algorithm for $\ell_1$ regularized optimization problems with box constraints. J. Sci. Comput. **85**(3), 1–34 (2020)
47. Stetsyuk, P.I., Nurminski, E.A.: Nonsmooth penalty and subgradient algorithms to solve the problem of projection onto a polytope. Cybern. Syst. Anal. **46**(1), 51 (2010)
48. Sun, D., Sun, J.: Semismooth matrix-valued functions. Math. Oper. Res. **27**(1), 150–169 (2002)
49. Torrisi, G., Grammatico, S., Smith, R.S., Morari, M.: A projected gradient and constraint linearization method for nonlinear model predictive control. SIAM J. Control Optim. **56**(3), 1968–1999 (2018)
50. Wang, B., Lin, L., Liu, Y.-J.: Efficient projection onto the intersection of a half-space and a box-like set and its generalized Jacobian. Optimization **71**(4), 1073–1096 (2022)
51. Wang, J. and Ye, J.: Safe screening for multi-task feature learning with multiple data matrices. In: International conference on machine learning, pp. 1747–1756. PMLR, (2015)
52. Wang, J., Zhou, J., Liu, J., Wonka, P., Ye, J.: A safe screening rule for sparse logistic regression. Adv. Neural Inf. Process. Syst. **2**, 1053–1061 (2014)
53. Wen, Z., Yin, W., Goldfarb, D., Zhang, Y.: A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. SIAM J. Sci. Comput. **32**(4), 1832–1857 (2010)
54. Wright, S.J., Nowak, R.D., Figueiredo, M.A.T.: Sparse reconstruction by separable approximation. IEEE Trans. Signal Process. **57**(7), 2479–2493 (2009)
55. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**(4), 1043–1056 (2004)
56. Zhao, X., Yao, J.-C.: Linear convergence of a nonmonotone projected gradient method for multiobjective optimization. J. Global Optim. **82**(3), 577–594 (2022)
57. Zhao, X.-Y., Sun, D., Toh, K.-C.: A Newton-CG augmented lagrangian method for semidefinite programming. SIAM J. Optim. **20**(4), 1737–1765 (2010)
58. Zhou, B., Gao, L., Dai, Y.-H.: Gradient methods with adaptive step-sizes. Comput. Optim. Appl. **35**(1), 69–86 (2006)