Math Meth Oper Res (2017) 86:123–147
DOI 10.1007/s00186-017-0585-1

ORIGINAL ARTICLE

An optimal subgradient algorithm for large-scale bound-constrained convex optimization

Masoud Ahookhosh[1] · **Arnold Neumaier**[1]

Received: 19 March 2016 / Accepted: 23 March 2017 / Published online: 12 April 2017

Abstract This paper shows that the optimal subgradient algorithm (OSGA)—which uses first-order information to solve convex optimization problems with optimal complexity—can be used to efficiently solve arbitrary bound-constrained convex optimization problems. This is done by constructing an explicit method as well as an inexact scheme for solving the bound-constrained rational subproblem required by OSGA. This leads to an efficient implementation of OSGA on large-scale problems in applications arising from signal and image processing, machine learning and statistics. Numerical experiments demonstrate the promising performance of OSGA on such problems. A software package implementing OSGA for bound-constrained convex problems is available.

Keywords Bound-constrained convex optimization · Nonsmooth optimization · First-order black-box oracle · Subgradient methods · Optimal complexity · High-dimensional data

Mathematics Subject Classification 90C25 · 90C60 · 49M37 · 65K05 · 68Q25

1 Introduction

Let V be a finite-dimensional real vector space and V^* its dual space. In this paper we consider the bound-constrained convex minimization problem

✉ Masoud Ahookhosh
masoud.ahookhosh@univie.ac.at

Arnold Neumaier
Arnold.Neumaier@univie.ac.at

[1] Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria

$$\begin{aligned} &\min\ f(x)\\ &\text{s.t.}\ \ x \in \mathbf{x}, \end{aligned} \tag{1}$$

where $f : \mathbf{x} \to \mathbb{R}$ is a—smooth or nonsmooth—convex function, and $\mathbf{x} = [\underline{x}, \overline{x}]$ is an axis-parallel box in V in which $\underline{x}$ and $\overline{x}$ are the vectors of lower and upper bounds on the components of x, respectively. Lower bounds are allowed to take the value $-\infty$, and upper bounds the value $+\infty$.

Throughout the paper, $\langle g, x\rangle$ denotes the value of $g \in V^*$ at $x \in V$. A subgradient of the objective function f at x is a vector $g(x) \in V^*$ satisfying

$$f(z) \geq f(x) + \langle g(x), z - x\rangle$$

for all $z \in V$. It is assumed that the set of optimal solutions of (1) is nonempty and the first-order information about the objective function (i.e., for any $x \in \mathbf{x}$, the function value $f(x)$ and some subgradient $g(x)$ at x) are available by a first-order black-box oracle.

Motivation and history Bound-constrained optimization in general is an important problem appearing in many fields of science and engineering, where the parameters describing physical quantities are constrained to be in a given range. Furthermore, it plays a prominent role in the development of general constrained optimization methods since many methods reduce the solution of the general problem to the solution of a sequence of bound-constrained problems.

There are many algorithms for solving bound-constrained optimization; here, we mention only those related to our study. Lin and Moré (1999) and Kim et al. (2010) proposed Newton and quasi-Newton methods for solving bound-constrained optimization. In 1995, Byrd et al. (1995) proposed a limited memory algorithm called LBFGS-B for general smooth nonlinear bound-constrained optimization. Branch et al. (1999) proposed a trust-region method to solve this problem. Neumaier and Azmi (2016) solved this problem by a limited memory algorithm. The smooth bound-constrained optimization problem was also solved by Birgin et al. (2000) and Hager and Zhang (2006, 2013) using nonmonotone spectral projected gradient methods, active set strategy and affine scaling scheme, respectively. Some limited memory bundle methods for solving bound-constrained nonsmooth problems were proposed by Karmitsa and Mäkelä (2010a, b).

In recent years convex optimization has received much attention because it arises in many applications and is suitable for solving problems involving high-dimensional data. The particular case of bound-constrained convex optimization involving a smooth or nonsmooth objective function also appears in a variety of applications, of which we mention the following:

Example 1 (Bound-constrained linear inverse problems) Given $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $\lambda \in \mathbb{R}$, for $m \geq n$, the bound-constrained regularized least-squares problem is given by

$$\begin{aligned} &\min\quad f(x) := \frac{1}{2}\|Ax - b\|_2^2 + \lambda\varphi(x)\\ &\text{s.t.}\quad x \in \mathbf{x}, \end{aligned} \tag{2}$$

and the bound-constrained regularized l_1 problem is given by

$$\begin{array}{ll} \min & f(x) := \|Ax - b\|_1 + \lambda\varphi(x) \\ \text{s.t.} & x \in \mathbf{x}, \end{array} \tag{3}$$

where $\mathbf{x} = [\underline{x}, \overline{x}]$ is a box and φ is a smooth or nonsmooth regularizer, often a weighted power of a norm; see Sect. 4 for examples. The problems (2) and (3) are commonly arising in the context of control and inverse problems, especially for some imaging problems like denoising, deblurring and inpainting. Morini et al. (2010) formulated the bound-constrained least-squares problem (2) as a nonlinear system of equations and proposed an iterative method based on a reduced Newton's method. Recently, Zhang and Morini (2013) used alternating direction methods to solve these problems. More recently, Chan et al. (2013), Boţ et al. (2013), and Boţ and Hendrich (2013) proposed alternating direction methods, primal-dual splitting methods, and a Douglas–Rachford primal-dual method, respectively, to solve both (2) and (3) for some applications.

Content In this paper, we show that the optimal subgradient algorithm OSGA proposed by Neumaier (2016) can be used for solving bound-constrained problems of the form (1). In order to run OSGA, one needs to solve a rational auxiliary subproblem. We here investigate efficient schemes for solving this subproblem in the presence of bounds on its variables. To this end, we show that the solution of the subproblem has a one-dimensional piecewise linear representation and that it may be computed by solving a sequence of one-dimensional piecewise rational optimization problems. We also give an iterative scheme that can solve the OSGA subproblem approximately by solving a one-dimensional nonlinear equation. We give numerical results demonstrating the performance of OSGA on some problems from applications. More specifically, in Sect. 2, we give a brief review of the main idea of OSGA. In Sect. 3, we investigate properties of the solution of the subproblem (9) that lead to two algorithms for solving it efficiently. In Sect. 4, we report numerical results of OSGA for an one-dimensional signal recovery and a two-dimensional image deblurring problem. Finally, Sect. 5 delivers some conclusions.

2 A review of OSGA

In this section, we briefly review the main idea of the optimal subgradient algorithm (see Algorithm 1) proposed by Neumaier (2016) for solving the convex constrained minimization problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in C, \end{array} \tag{4}$$

where $f : C \to \mathbb{R}$ is a proper and convex function defined on a nonempty, closed, and convex subset C of a finite-dimensional vector space V, which we take without loss of generality to be $\mathbb{R}^n$, which is its own dual space.

OSGA is a subgradient algorithm for problem (4) that uses first-order information, i.e., function values and subgradients, to construct a sequence of iterations $\{x_j\} \in C$ whose sequence of function values $\{f(x_j)\}$ converge to the minimum $\widehat{f} = f(\widehat{x})$ with the optimal complexity. OSGA requires no information regarding global parameters such as Lipschitz constants of function values and gradients. It uses a so-called prox function which we take to be

$$Q(x) := Q_0 + \frac{1}{2}\|x - x^0\|_2^2 \tag{5}$$

where $Q_0 > 0$. Thus $Q(x) \geq Q_0 > 0$ for all $x \in \mathbb{R}^n$, and

$$Q(z) \geq Q(x) + \langle g_Q(x), z - x\rangle + \frac{1}{2}\|z - x\|_2^2, \tag{6}$$

where $g_Q(x) = x - x^0$ is the gradient of Q at x and $\|x\|_2$ is the Euclidean norm. At each iteration, OSGA satisfies the bound

$$0 \leq f(x_b) - \widehat{f} \leq \eta Q(u) \tag{7}$$

on the currently best function value $f(x_b)$ with a monotonically decreasing error factor η that is guaranteed to converge to zero by an appropriate steplength selection strategy (see Procedure PUS). Note that $\widehat{x}$ is not known a priori, thus the error bound is not fully constructive. But it is sufficient to guarantee the convergence of $f(x_b)$ to $\widehat{f}$ with a predictable worst case complexity. To maintain (7), OSGA considers linear relaxations of f at z,

$$f(z) \geq \gamma + \langle h, z\rangle \quad \text{for all } z \in C, \tag{8}$$

where $\gamma \in \mathbb{R}$ and $h \in V^*$, updated using linear underestimators available from the subgradients evaluated (see Algorithm 1). For each such linear relaxation, OSGA solves a maximization problem of the form

$$\begin{aligned} E(\gamma, h) := \max \quad & E_{\gamma,h}(x) \\ \text{s.t.} \quad & x \in C, \end{aligned} \tag{9}$$

where

$$E_{\gamma,h}(x) := -\frac{\gamma + \langle h, x\rangle}{Q(x)}. \tag{10}$$

Let $\gamma_b := \gamma - f(x_b)$ and $u := U(\gamma_b, h) \in C$ be the solution of (9). From (8) and (10), we obtain

$$E(\gamma_b, h) \geq -\frac{\gamma - f(x_b) + \langle h, u\rangle}{Q(u)} \geq \frac{f(x_b) - \widehat{f}}{Q(u)} \geq 0. \tag{11}$$

Setting $\eta := E(\gamma_b, h)$ in (11) implies that (7) is valid. If x_b is not optimal for (1), then the right inequality in (11) is strict, and since $Q(z) \geq Q_0 > 0$, we conclude that the maximum η is positive. In the remainder of the paper, we denote by g_{x_b} and f_{x_b} a subgradient of f at x_b and the function value $f(x_b)$, respectively.

In each step, OSGA uses the next scheme for updating the given parameters α, h, γ, η, and u, see Neumaier (2016) for more details.

Procedure PUS(parameters updating scheme)

Input: $\delta,\ \alpha_{\max} \in]0, 1[,\ 0 < \kappa' \leq \kappa, \alpha, \eta, \bar{h}, \bar{\gamma}, \bar{\eta}, \bar{u}$;
Output: $\alpha,\ h,\ \gamma,\ \eta,\ u$;

begin
  $R = (\eta - \overline{\eta})/(\delta\alpha\eta)$;
  if $R < 1$ **then**
    $\overline{\alpha} = \alpha e^{-\kappa}$;
  else
    $\overline{\alpha} = \min(\alpha e^{\kappa'(R-1)}, \alpha_{\max})$;
  end
  $\alpha = \overline{\alpha}$;
  if $\overline{\eta} < \eta$ **then**
    $h = \overline{h};\ \gamma = \overline{\gamma};\ \eta = \overline{\eta};\ u = \overline{u}$;
  end
end

Algorithm 1: OSGA (optimal subgradient algorithm)

Input: $\delta, \alpha_{\max} \in]0, 1[,\ 0 < \kappa' \leq \kappa$; local parameters: $x^0, \mu \geq 0$;
Output: $x_b,\ f_{x_b}$;

begin
  $x_b = x^0;\ h = g_{x_b} - \mu g_Q(x_b);\ \gamma = f_{x_b} - \mu Q(x_b) - \langle h, x_b\rangle$;
  $\gamma_b = \gamma - f_{x_b};\ u = U(\gamma_b, h);\ \eta = E(\gamma_b, h) - \mu;\ \alpha \leftarrow \alpha_{\max}$;
  while *stopping criteria do not hold* **do**
    $x = x_b + \alpha(u - x_b); g = g_x - \mu g_Q(x)$;
    $\overline{h} = h + \alpha(g - h); \overline{\gamma} = \gamma + \alpha(f_x - \mu Q(x) - \langle g, x\rangle - \gamma)$;
    $x'_b = \text{argmin}_{z\in\{x_b,x\}} f(z);\ f_{x'_b} = \min\{f_{x_b}, f_x\}$;
    $\gamma'_b = \overline{\gamma} - f_{x'_b};\ u' = U(\gamma'_b, \overline{h}); x' = x_b + \alpha(u' - x_b)$;
    choose $\overline{x}_b$ in such a way that $f_{\overline{x}_b} \leq \min\{f_{x'_b}, f_{x'}\}$;
    $\overline{\gamma}_b = \overline{\gamma} - f_{\overline{x}_b};\ \overline{u} = U(\overline{\gamma}_b, \overline{h});\ \overline{\eta} = E(\overline{\gamma}_b, \overline{h}) - \mu;\ x_b = \overline{x}_b;\ \ f_{x_b} = f_{\overline{x}_b}$;
    update the parameters α, h, γ, η and u using PUS;
  end
end

The original stopping criterion of OSGA is $\eta \leq \varepsilon$; however, we will use a more practical stopping criterion in Sect. 4. In Neumaier (2016), it is shown that the number of iterations to achieve an ε-optimum is of the optimal order $\mathcal{O}\left(\varepsilon^{-1/2}\right)$ for a smooth function f with Lipschitz continuous gradients and of the order $\mathcal{O}\left(\varepsilon^{-2}\right)$ for a Lipschitz continuous nonsmooth function f, cf. Nemirovsky and Yudin (1983) and Nesterov (2004, 2005). The algorithm has low memory requirements so that, if the subproblem (9) can be solved efficiently, OSGA is appropriate for solving large-scale problems. Numerical results reported by Ahookhosh (2016) for unconstrained problems, and by

Ahookhosh and Neumaier (2016a, b, 2013) for simply constrained problems show the good behavior of OSGA for solving practical problems.

In this paper, for the above choices of $Q(x)$ and an arbitrary box $\mathbf{x}$, we solve the subproblem (9) for both medium- and large-scale problems. It follows that OSGA is applicable to solve bound-constrained convex problems as well. Since the underlying problem (1) is a special case of the problem considered in Neumaier (2016), the complexity of OSGA remains valid for (1), which is summarized in the following theorem.

Theorem 2 *Suppose that $f - \mu Q$ is convex and $\mu \geq 0$. Then we have*

(i) (NONSMOOTH COMPLEXITY BOUND) *If the points generated by Algorithm 1 stay in a bounded region of the interior of $\mathbf{x}$, or if f is Lipschitz continuous on $\mathbf{x}$, the total number of iterations needed to reach a point with $f(x) \leq f(u) + \varepsilon$ is at most $\mathcal{O}((\varepsilon^2 + \mu\varepsilon)^{-1})$. Thus the asymptotic worst case complexity is $\mathcal{O}(\varepsilon^{-2})$ when $\mu = 0$ and $\mathcal{O}(\varepsilon^{-1})$ when $\mu > 0$.*

(ii) (SMOOTH COMPLEXITY BOUND) *If f has Lipschitz continuous gradients with Lipschitz constant L, the total number of iterations needed by Algorithm 1 to reach a point with $f(x) \leq f(u) + \varepsilon$ is at most $\mathcal{O}(\varepsilon^{-1/2})$ if $\mu = 0$, and at most $\mathcal{O}(|\log \varepsilon| \sqrt{L/\mu})$ if $\mu > 0$.*

Proof Since all assumptions of Theorems 4.1 and 4.2, Propositions 5.2 and 5.3, and Theorem 5.1 in Neumaier (2016) are satisfied, the results remain valid. □

3 Solution of the bound-constrained subproblem (9)

We here emphasize that the function $E_{\gamma,h}(\cdot)$ is quasi-concave. Hence finding a solution of this subproblem is the bottleneck of OSGA, which is both theoretically and practically interesting to be studied. Therefore, in this section we investigate the solution of the bound-constrained subproblem (9) and give two iterative schemes, where the first one solves (9) exactly whereas the second one solves it approximately.

3.1 Global solution of the OSGA rational subproblem (9)

In this subsection, we describe an explicit solution of the bound-constrained subproblem (9).

Without loss of generality, we here consider $V = \mathbb{R}^n$. It is not hard to adapt the results to $V = \mathbb{R}^{m\times n}$ and other finite-dimensional spaces. The method is related to one used in several earlier papers. In 1980, Helgason et al. (1980) characterized the solution of a singly constrained quadratic problem with bound constraints. Later, Pardalos and Kovoor (1990) developed an $\mathcal{O}(n)$ algorithm for this problem using binary search to solve the associated Kuhn–Tucker system. This problem was also solved by Dai and Fletcher (2006) using a projected gradient method. Zhang et al. (2011) solved the linear support vector machine problem by a cutting plane method employing a similar technique.

In the papers mentioned, the key is showing that the problem can be reduced to a piecewise linear problem in a single dimension. To apply this idea to the present

problem, we prove that (9) is equivalent to a one-dimensional minimization problem and then develop a procedure to calculate its minimizer. We write

$$u(\lambda) := \sup\{\underline{x}, \inf\{x^0 - \lambda h, \overline{x}\}\} \tag{12}$$

for the projection of $x^0 - \lambda h$ to the box $\mathbf{x}$.

Proposition 3 *For $h \neq 0$, the maximum of the subproblem (9) is attained at $u := u(\lambda)$, where $\lambda > 0$ or $\lambda = +\infty$ is the inverse of the value of the maximum.*

Proof The function $E_{\gamma,h} : V \to \mathbb{R}$ defined by (10) is continuously differentiable and $\eta := E(\gamma, h) > 0$. Since $Q(x) = \frac{1}{2}\|x - x^0\|^2, g_Q(x) = x - x^0$. By differentiating both sides of the equation $E_{\gamma,h}(x)Q(x) = -\gamma - \langle h, x\rangle$, we obtain $\frac{\partial E_{\gamma,h}}{\partial x} Q(x) + \eta(x - x^0) = -h,$, leading to

$$\frac{\partial E_{\gamma,h}}{\partial x} Q(x) = -\eta(x - x^0) - h.$$

At the maximizer u, we have $\eta Q(u) = -\gamma - \langle h, u\rangle$. Now the first-order optimality conditions imply that for $i = 1, 2, \ldots, n$,

$$-\eta(u_i - x_i^0) - h_i \begin{cases} \leq 0 & \text{if } u_i = \underline{x}_i, \\ \geq 0 & \text{if } u_i = \overline{x}_i, \\ = 0 & \text{if } \underline{x}_i < u_i < \overline{x}_i. \end{cases} \tag{13}$$

Since $\eta > 0$, we may define $\lambda := \eta^{-1}$ and find that, for $i = 1, 2, \ldots, n$,

$$u_i = \begin{cases} \underline{x}_i & \text{if } \underline{x}_i \geq x_i^0 - \lambda h_i, \\ \overline{x}_i & \text{if } \overline{x}_i \leq x_i^0 - \lambda h_i, \\ x_i^0 - \lambda h_i & \text{if } \underline{x}_i \leq x_i^0 - \lambda h_i \leq \overline{x}_i. \end{cases} \tag{14}$$

This implies that $u = u(\lambda)$. □

Proposition 3 gives the key feature of the solution of the subproblem (9) implying that it is enough to consider points of the form (12) which depend on only one variable λ. In the remainder of this section, we focus on deriving the optimal value for λ.

Example 4 Let us consider a very special case that $\mathbf{x}$ is the n-dimensional nonnegative orthant, i.e., $\underline{x}_i = 0$ and $\overline{x}_i = +\infty$, for $i = 1, \ldots, n$. Nonnegativity as a constraint is important in many applications, see Bardsley and Vogel (2003), Elfving et al. (2012), Esser et al. (2013) and Kaufman and Neumaier (1996, 1997). For the prox function (5) with $x^0 = 0$, (12) becomes

$$u(\lambda) = \sup\{\underline{x}, \inf\{-\lambda h, \overline{x}\}\} = \lambda h_-,$$

where $z_- := \max\{0, -z\}$. By Proposition 2.2 of Neumaier (2016), we have

$$\frac{1}{\lambda}\left(\frac{1}{2}\|u(\lambda)\|_2^2 + Q_0\right) + \gamma + \langle h, u(\lambda)\rangle = \left(\frac{1}{2}\|h_-\|_2^2 + \langle h, h_-\rangle\right)\lambda^2 + \gamma\lambda + Q_0 = 0,$$

leading to

$$\beta_1\lambda^2 + \beta_2\lambda + \beta_3 = 0,$$

where $\beta_1 = \frac{1}{2}\|h_-\|_2^2 + \langle h, h_-\rangle$, $\beta_2 = \gamma$, and $\beta_3 = Q_0$. Since we search for the maximum η, the solution is the largest root of this equation, i.e.,

$$\lambda = \frac{-\beta_2 + \sqrt{\beta_2^2 - 4\beta_1\beta_3}}{2\beta_1}.$$

This shows that for the nonnegativity constraint the subproblem (9) can be solved in a closed form.

However, for a general bound-constrained problem, solving (9) requires a much more sophisticated scheme. To derive the optimal $\lambda \geq 0$ in Proposition 3, we first determine its permissible range provided by the three conditions considered in (14) leading to the interval

$$\lambda \in [\underline{\lambda}_i, \overline{\lambda}_i], \tag{15}$$

for each component of x. In particular, if $h_i = 0$, since x^0 is a feasible point, $u_i = x_i^0 - \lambda h_i = x_i^0$ satisfies the third condition in (14). Thus there is no upper bound for λ, leading to

$$\underline{\lambda}_i = 0, \ \ \overline{\lambda}_i = +\infty \ \ \text{if } u_i = x_i^0, \ h_i = 0. \tag{16}$$

If $h_i \neq 0$, we consider the three cases (i) $\underline{x}_i \geq x_i^0 - \lambda h_i$, (ii) $\overline{x}_i \leq x_i^0 - \lambda h_i$, and (iii) $\underline{x}_i \leq x_i^0 - \lambda h_i \leq \overline{x}_i$ of (14). In Case (i), if $h_i < 0$, division by h_i implies that $\lambda \leq -(\underline{x}_i - x_i^0)/h_i \leq 0$, which is not in the acceptable range for λ. In this case, if $h_i > 0$, then $\lambda \geq -(\underline{x}_i - x_i^0)/h_i$ leading to

$$\underline{\lambda}_i = -\frac{\underline{x}_i - x_i^0}{h_i}, \ \overline{\lambda}_i = +\infty \ \ \text{if } u_i = \underline{x}_i, \ h_i > 0. \tag{17}$$

In Case (ii), if $h_i < 0$, then $\lambda \geq -(\overline{x}_i - x_i^0)/h_i$ implying

$$\underline{\lambda}_i = -\frac{\overline{x}_i - x_i^0}{h_i}, \ \overline{\lambda}_i = +\infty \ \ \text{if } u_i = \overline{x}_i, \ h_i < 0. \tag{18}$$

In Case (ii), if $h_i > 0$, then $\lambda \leq -(\overline{x}_i - x_i^0)/h_i \leq 0$, which is not in the acceptable range of λ. In Case (iii), if $h_i < 0$, division by h_i implies

$$-\frac{\underline{x}_i - x_i^0}{h_i} \leq \lambda \leq -\frac{\overline{x}_i - x_i^0}{h_i}.$$

The lower bound satisfies $-(\underline{x}_i - x_i^0)/h_i \leq 0$, so it is not acceptable, leading to

$$\underline{\lambda}_i = 0, \ \overline{\lambda}_i = -\frac{\overline{x}_i - x_i^0}{h_i} \ \ \text{if } u_i = x_i^0 - \lambda h_i \in [\underline{x}_i \overline{x}_i], \ h_i < 0. \tag{19}$$

In Case (iii), if $h_i > 0$, then

$$-\frac{\overline{x}_i - x_i^0}{h_i} \le \lambda \le -\frac{\underline{x}_i - x_i^0}{h_i}.$$

However, the lower bound $-(\overline{x}_i - x_i^0)/h_i \le 0$ is not acceptable, i.e.,

$$\underline{\lambda}_i = 0,\ \overline{\lambda}_i = -\frac{\underline{x}_i - x_i^0}{h_i} \quad \text{if } u_i = x_i^0 - \lambda h_i \in [\underline{x}_i \overline{x}_i],\ h_i > 0. \tag{20}$$

As a result, the following proposition is valid.

Proposition 5 *If $u(\lambda)$ is solution of the problem (9), then*

$$\lambda \in [\underline{\lambda}_i, \overline{\lambda}_i] \quad i = 1, \ldots, n,$$

where $\underline{\lambda}_i$ and $\overline{\lambda}_i$ are computed by

$$\underline{\lambda}_j = \begin{cases} -\dfrac{\underline{x}_i - x_i^0}{h_i} & \text{if } u_i = \underline{x}_i,\ h_i > 0, \\ -\dfrac{\overline{x}_i - x_i^0}{h_i} & \text{if } u_i = \overline{x}_i,\ h_i < 0, \\ 0 & \text{if } \widetilde{x}_i \in [\underline{x}_i \overline{x}_i],\ h_i < 0, \\ 0 & \text{if } \widetilde{x}_i \in [\underline{x}_i \overline{x}_i],\ h_i > 0, \\ 0 & \text{if } h_i = 0, \end{cases}$$

$$\overline{\lambda}_j = \begin{cases} +\infty & \text{if } u_i = \underline{x}_i,\ h_i > 0, \\ +\infty & \text{if } u_i = \overline{x}_i,\ h_i < 0, \\ -\dfrac{\overline{x}_i - x_i^0}{h_i} & \text{if } \widetilde{x}_i \in [\underline{x}_i \overline{x}_i],\ h_i < 0, \\ -\dfrac{\underline{x}_i - x_i^0}{h_i} & \text{if } \widetilde{x}_i \in [\underline{x}_i \overline{x}_i],\ h_i > 0, \\ +\infty & \text{if } h_i = 0, \end{cases} \tag{21}$$

in which $\widetilde{x}_i = x_i^0 - \lambda h_i$ for $i = 1, \ldots, n$.

From Proposition 5, only one of the conditions (16)–(20) is satisfied for each component of x. Thus, for each $i = 1, \ldots, n$ with $h_i \ne 0$, we have a single breakpoint

$$\widetilde{\lambda}_i := \begin{cases} -\dfrac{\overline{x}_i - x_i^0}{h_i} & \text{if } h_i < 0, \\ -\dfrac{\underline{x}_i - x_i^0}{h_i} & \text{if } h_i > 0, \\ +\infty & \text{if } h_i = 0. \end{cases} \tag{22}$$

Sorting the n bounds $\widetilde{\lambda}_i,\ i = 1, \ldots, n$, in increasing order, augmenting the resulting list by 0 and $+\infty$, and deleting possible duplicate points, we obtain a list of $m + 1$ different breakpoints ($m + 1 \le n + 2$), denoted by

$$0 = \lambda_1 < \lambda_2 < \cdots < \lambda_m < \lambda_{m+1} = +\infty. \tag{23}$$

By construction, $u(\lambda)$ is linear in each interval $[\lambda_k, \lambda_{k+1}]$, for $k = 1, \ldots, m$. The next proposition gives an explicit representation for $u(\lambda)$.

Proposition 6 *The solution $u(\lambda)$ of the auxiliary problem (9) defined by (12) has the form*

$$u(\lambda) = p^k + \lambda q^k \quad \text{for } \lambda \in [\lambda_k, \lambda_{k+1}] \quad (k = 1, 2, \ldots, m), \tag{24}$$

where

$$p_i^k = \begin{cases} x_i^0 & \text{if } h_i = 0, \\ x_i^0 & \text{if } \lambda_{k+1} \leqq \widetilde{\lambda}_i, \\ \underline{x}_i & \text{if } \lambda_k \geq \widetilde{\lambda}_i,\ h_i > 0, \\ \overline{x}_i & \text{if } \lambda_k \geq \widetilde{\lambda}_i,\ h_i < 0, \end{cases} \qquad q_i^k = \begin{cases} 0 & \text{if } h_i = 0, \\ -h_i & \text{if } \lambda_{k+1} \leqq \widetilde{\lambda}_i, \\ 0 & \text{if } \lambda_k \geq \widetilde{\lambda}_i,\ h_i > 0, \\ 0 & \text{if } \lambda_k \geq \widetilde{\lambda}_i,\ h_i < 0. \end{cases} \tag{25}$$

Proof Since $\lambda > 0$, there exists $k \in \{1, \ldots, m\}$ such that $\lambda \in [\lambda_k, \lambda_{k+1}]$. Let $i \in \{1, \ldots, n\}$. If $h_i = 0$, (16) implies $u_i = x_i^0$. If $h_i \neq 0$, the way of construction of λ_i for $i = 1, \ldots, m$ implies that $\widetilde{\lambda}_i \notin (\lambda_k, \lambda_{k+1})$, so two cases are distinguished: (i) $\lambda_{k+1} \leq \widetilde{\lambda}_i$; (ii) $\lambda_k \geq \widetilde{\lambda}_i$. In Case (i), Proposition 5 implies that $\widetilde{\lambda}_i = \overline{\lambda}_i$, while it is not possible $\widetilde{\lambda}_i \neq \underline{\lambda}_i$. Therefore, either (19) or (20) holds dependent on the sign of h_i, implying $x_i^0 - \lambda h_i \in [\underline{x}_i, \overline{x}_i]$, so that $p_i^k = x_i^0$ and $q_i^k = -h_i$. In Case (ii), Proposition 5 implies that $\widetilde{\lambda}_i = \underline{\lambda}_i$, while it is not possible $\widetilde{\lambda}_i \neq \overline{\lambda}_i$. Therefore, either (17) or (18) holds. If $h_i < 0$, then (18) holds, i.e., $p_i^k = \overline{x}_i$ and $q_i^k = 0$. Otherwise, (17) holds, implying $p_i^k = \underline{x}_i$ and $q_i^k = 0$. This proves the claim. □

Proposition 6 exhibits the solution $u(\lambda)$ of the auxiliary problem (9) as a piecewise linear function of λ. In the next result, we show that solving the problem (9) is equivalent to maximizing a one-dimensional piecewise rational function.

Proposition 7 *The maximal value of the subproblem (9) is the maximum of the piecewise rational function $\eta(\lambda)$ defined by*

$$\eta(\lambda) := \frac{a_k + b_k\lambda}{c_k + d_k\lambda + s_k\lambda^2} \quad \text{if } \lambda \in [\lambda_k, \lambda_{k+1}] \quad (k = 1, 2, \ldots, m), \tag{26}$$

where

$$\begin{aligned} &a_k := -\gamma - \langle h, p^k \rangle, \quad b_k := -\langle h, q^k \rangle, \\ &c_k := Q_0 + \frac{1}{2}\|p^k - x^0\|^2, \quad d_k := \langle p^k - x^0, q^k \rangle, \quad s_k := \frac{1}{2}\|q^k\|^2. \end{aligned}$$

Moreover, $c_k > 0$, $s_k > 0$ and $4s_kc_k > d_k^2$.

Proof By Proposition 3 and 6, the global minimizer of (9) has the form (24). We substitute (24) into the function (10), and obtain from

$$\gamma + \langle h, x^k(\lambda) \rangle = \gamma + \langle h, p^k + q^k\lambda \rangle = \gamma + \langle h, p^k \rangle + \langle h, q^k \rangle\lambda = -a_k - b_k\lambda$$

and

$$\begin{aligned} Q_0 &\le Q(x^k(\lambda)) \\ &= Q(p^k + q^k\lambda) \\ &= Q_0 + \frac{1}{2}\|p^k - x^0\|^2 + \langle p^k - x^0, q^k\rangle\lambda + \frac{1}{2}\|q^k\|^2\lambda^2 = c_k + d_k\lambda + s_k\lambda^2 \end{aligned}$$

the formula

$$E_{\gamma,h}(u(\lambda)) = -\frac{\gamma + \langle h, x^k(\lambda)\rangle}{Q(x^k(\lambda))} = \eta(\lambda). \tag{27}$$

Since $Q_0 > 0$, the denominator of (26) is bounded away from zero; in particular $c_k > 0$. This implies $4s_kc_k > d_k^2$. It is enough to verify $s_k > 0$ For $k = 1, 2, \ldots, m$ and $\lambda \in [\lambda_k, \lambda_{k+1}]$. Now the definition of q_k in (25) implies that $h_i \neq 0$ for $i \in I = \{i : \lambda_{k+1} \le \widetilde{\lambda}_i\}$, leading to $q^k \neq 0$, hence $s_k > 0$. □

The next result leads to a systematic way to maximize the one-dimensional rational problem (26).

Proposition 8 *Let a, b, c, d, and s be real constants with $c > 0$, $s > 0$, and $4sc > d^2$. Then*

$$\phi(\lambda) := \frac{a + b\lambda}{c + d\lambda + s\lambda^2} \tag{28}$$

defines a function $\phi : \mathbb{R} \to \mathbb{R}$ that has at least one stationary point. Moreover, the global maximizer of ϕ is determined by the following cases:

(i) If $b \neq 0$, then $a^2 - b(ad - bc)/s > 0$ and the global maximum

$$\phi(\widehat{\lambda}) = \frac{b}{2s\widehat{\lambda} + d} \tag{29}$$

is attained at

$$\widehat{\lambda} = \frac{-a + \sqrt{a^2 - b(ad - bc)/s}}{b}. \tag{30}$$

(ii) If $b = 0$ and $a > 0$, the global maximum is

$$\phi(\widehat{\lambda}) = \frac{4as}{4cs - d^2}, \tag{31}$$

attained at

$$\widehat{\lambda} = -\frac{d}{2s}. \tag{32}$$

(iii) If $b = 0$ and $a \le 0$, the maximum is $\phi(\widehat{\lambda}) = 0$, attained at $\widehat{\lambda} = +\infty$ for $a < 0$ and at all $\lambda \in \mathbb{R}$ for $a = 0$.

Proof The denominator of (28) is positive for all $\lambda \in \mathbb{R}$ if and only if the stated condition on the coefficients hold. By the differentiation of ϕ and using the first-order optimality condition, we obtain

$$\phi'(\lambda) = \frac{b(c + d\lambda + s\lambda^2) - (a + b\lambda)(d + 2s\lambda)}{(c + d\lambda + s\lambda^2)^2} = -\frac{bs\lambda^2 + 2as\lambda + ad - bc}{(c + d\lambda + s\lambda^2)^2}.$$

For solving $\phi'(\lambda) = 0$, we consider possible solutions of the quadratic equation $bs\lambda^2 + 2as\lambda + ad - bc = 0$. Using the assumption $4sc > d^2$, we obtain

$$\begin{aligned}(2as)^2 - 4bs(ad - bc) &= (2as)^2 - (4abds - 4b^2cs)\\ &= (2as)^2 - 4abds - b^2(d^2 - 4cs - d^2)\\ &= (2as)^2 - 4abds + (bd)^2 - b^2(d^2 - 4cs)\\ &\geq (2as - bd)^2 - b^2(d^2 - 4cs) \geq 0,\end{aligned}$$

leading to

$$a^2 - \frac{b(ad - bc)}{s} \geq 0,$$

implying that $\phi'(\lambda) = 0$ has at least one solution.

(i) If $b \neq 0$, then

$$a^2 - \frac{b(ad - bc)}{s} = a^2 - \frac{bd}{s}a - \frac{b^2c}{s} = \left(a - \frac{bd}{2s}\right)^2 + \frac{b^2}{4s^2}(4sc - d^2) > 0,$$

implying there exist two solutions. Solving $\phi'(\lambda) = 0$, the stationary points of the function are found to be

$$\lambda = \frac{-a \pm \sqrt{a^2 - b(ad - bc)/s}}{b}. \tag{33}$$

Therefore, $a + b\lambda = \pm w$ with

$$w := \sqrt{a^2 - b(ad - bc)/s} > 0,$$

and we have

$$\phi(\lambda) = \frac{\pm w}{c + d\lambda + s\lambda^2}. \tag{34}$$

Since the denominator of this fraction is positive and $w \geq 0$, the positive sign in Eq. (33) gives the maximizer, implying that (30) is satisfied. Finally, substituting this maximizer into (34) gives

$$\begin{aligned}\phi(\widehat{\lambda}) &= \frac{w}{c + d\widehat{\lambda} + s\widehat{\lambda}^2} = \frac{b^2 w}{b^2 c + bd(w - a) + s(w - a)^2}\\ &= \frac{b^2 w}{a^2 s - b(ad - bc) + sw^2 + (bd - 2as)w} = \frac{b^2 w}{2sw^2 + (bd - 2as)w}\\ &= \frac{b^2 w}{w(2s(w - a) + bd)} = \frac{b}{2s\widehat{\lambda} + d},\end{aligned}$$

hence (29) holds.

(ii) If $b = 0$, we obtain

$$\phi'(\lambda) = \frac{-a(d + 2s\lambda)}{(c + d\lambda + s\lambda^2)^2}.$$

Hence the condition $\phi'(\lambda) = 0$ implies that $a = 0$ or $d + 2s\lambda = 0$. The latter case implies

$$\widehat{\lambda} = -\frac{d}{2s}, \quad \phi(\widehat{\lambda}) = \frac{4as}{4cs - d^2},$$

whence $\widehat{\lambda}$ is a stationary point of ϕ. If $a > 0$, its maximizer is $\widehat{\lambda} = -\frac{d}{2s}$ and (31) is satisfied.

(iii) If $b = 0$ and $a < 0$, then

$$\lim_{\lambda \to -\infty} \phi(\lambda) = \lim_{\lambda \to +\infty} \phi(\lambda) = 0$$

implies $\phi(\widehat{\lambda}) = 0$ at $\widehat{\lambda} = \pm\infty$. In case $a = 0$, $\phi(\lambda) = 0$ for all $\lambda \in \mathbb{R}$. □

We summarize the results of Propositions 3–8 into the following algorithm for computing the global optimizer x_b and the optimum η_b of (9).

Algorithm 2: BCSS (bound-constrained subproblem solver)

Input: $Q_0, x^0, h, \underline{x}, \overline{x}$;
Output: $u_b = U(\gamma, h)$, $\eta_b = \eta(x_b)$;

1 **begin**
2 **for** $i = 1, 2, \ldots, n$ **do**
3 find $\widetilde{\lambda}_i$ by (22) using $\underline{x}$ and $\overline{x}$;
4 **end**
5 determine the breakpoints λ_k, $k = 1, \ldots, m + 1$, by (23); $\eta_b = 0$;
6 **for** $k = 1, 2, \ldots, m$ **do**
7 compute p^k and q^k using (25); construct $\eta(\lambda)$ using (26) for $[\lambda_k, \lambda_{k+1}]$;
8 find the maximizer $\widehat{\lambda}$ of $\eta(\lambda)$ using Proposition 8;
9 **if** $\widehat{\lambda} \in [\lambda_k, \lambda_{k+1}]$ **then**
10 compute $\eta^k = \eta(\widehat{\lambda})$ using Proposition 8; $\widehat{\lambda}^k = \widehat{\lambda}$;
11 **else**
12 $\eta^k = \max\{\eta(\lambda_k), \eta(\lambda_{k+1})\}$; $\widehat{\lambda}^k = \operatorname{argmax}_{i \in \{k, k+1\}}\{\eta(\lambda_i)\}$;
13 **end**
14 $E(k) = \eta^k$, $LAM(k) = \widehat{\lambda}^k$;
15 **end**
16 $j = \operatorname{argmax}\{E(i) \mid i = 1, \ldots, m\}$; $\eta_b = E(j)$, $\widehat{\lambda} = LAM(j)$, $u_b = x^0 - \widehat{\lambda}h$;
17 **end**

The first loop (lines 2–4) needs $\mathcal{O}(n)$ operations (including comparisons). Line 5 needs sorting and removing duplicates, requiring $\mathcal{O}(n\ log(n))$ operations. The

second loop (lines 6–15) needs $\mathcal{O}(m^2)$ operations. Line 16 require $\mathcal{O}(m)$ comparisons. Therefore, the computational complexity of, the algorithm BCSS is given by

$$\mathcal{N}(m,n) = \mathcal{O}(n\, log(n) + m^2). \tag{35}$$

The the cost of BCSS is negligible for small-scale and medium-scale problems, where m does not get too large.

3.2 Inexact solution of the OSGA rational subproblem (9)

In the BCSS algorithm, it is possible that the number m of different breakpoints is $\mathcal{O}(n)$. If m is large solving the subproblem (9) with BCSS is costly in a Matlab implementation, where branching is comparatively slow. If m has the same order as n the second term in (35) dominates and we have $\mathcal{N}(m,n) = \mathcal{O}(n^2)$. For the application to large-scale problems we need a cheaper alternative. We therefore looked for a theoretically less satisfactory (but in practice for large m superior) approximate technique for solving (9). For simplicity, we consider the quadratic prox-function (5) with $x^0 = 0$; the general case can be easily reduced to this one by shifting x appropriately.

In view of Proposition 3 and Theorem 3.1 in Ahookhosh and Neumaier (2017), the solution of the subproblem (9) is given by $u(\lambda)$ defined in (12), where λ can be computed by solving the one-dimensional nonlinear equation

$$\varphi(\lambda) = 0,$$

in which

$$\varphi(\lambda) := \frac{1}{\lambda}\left(\frac{1}{2}\|u(\lambda)\|_2^2 + Q_0\right) + \gamma + \langle h, u(\lambda)\rangle. \tag{36}$$

The solution of the OSGA subproblem can be found by Algorithm 3 (OSS) in Ahookhosh and Neumaier (2017). In Ahookhosh and Neumaier (2017), it is shown that in many convex domains the nonlinear Eq. (36) can be solved explicitly, however, for the bound-constrained problems it can be only solved approximately. The main advantages of the inexact approach is its simplicity and cheap cost for extremely large-scale problems.

As discussed in Ahookhosh and Neumaier (2017), the one-dimensional nonlinear equation can be solved by some zero-finder schemes such as the bisection method and the secant bisection scheme described in Chapter 5 of Neumaier (2001). One can also use the MATLAB `fzero` function combining the bisection scheme, the inverse quadratic interpolation, and the secant method. In the next section we will use this inexact solution of the OSGA rational subproblem (9) for solving large-scale imaging problems, which turned out to be much faster.

4 Numerical experiments and applications

In this section, we report numerical results for two inverse problems (one-dimensional signal recovery and two-dimensional image deblurring) to show the performance of OSGA compared with some state-of-the-art algorithms.

A software package implementing OSGA for solving unconstrained and bound-constrained convex optimization problems is publicly available at

http://www.mat.univie.ac.at/~neum/software/OSGA/.

The package is written in MATLAB, where the parameters

$$\delta = 0.9, \quad \alpha_{max} = 0.7, \quad \kappa = \kappa' = 0.5,$$

are used. We use the prox-function (5) with $Q_0 = \frac{1}{2}\|x^0\|_2 + \epsilon$, where $\epsilon > 0$ is the machine precision. The interface to each subprogram in the package is fully documented in the corresponding file. Some examples for each class of problems are available to show how the user can implement it. The OSGA user's manual (Ahookhosh 2014) describes the design of the package and how the user can solve his/her own problems.

The algorithms considered in the comparison use the default parameter values reported in the associated literature or packages. All implementations are executed on a Dell Precision Tower 7000 Series 7810 (Dual Intel Xeon Processor E5-2620 v4 with 32 GB RAM).

4.1 One-dimensional signal recovery

In this section, we deal with the linear inverse problem

$$Ax = b, \ x \in \mathbf{x}$$

that can be translated to a problem of the form (1) with the objective functions

$$\begin{array}{ll} f(x) = \frac{1}{2}\|Ax-b\|_2^2 + \frac{1}{2}\lambda\|x\|_2^2 & \text{(L22L22R)}, \\ f(x) = \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1 & \text{(L22L1R)}, \\ f(x) = \|Ax-b\|_1 + \frac{1}{2}\lambda\|x\|_2^2 & \text{(L1L22R)}, \\ f(x) = \|Ax-b\|_1 + \lambda\|x\|_1 & \text{(L1L1R)}, \end{array} \tag{37}$$

where λ is a regularization parameter.

We solve all of the above-mentioned problems with the dimensions $n = 1000$ and $m = 500$. The problem is generated by the same procedure given in the SpaRSA (Wright et al. 2009) package available at

http://www.lx.it.pt/~mtf/SpaRSA/

which is

$$\begin{aligned}
&\texttt{n_spikes} = \texttt{floor(spike_rate} * \texttt{n)};\\
&\texttt{p} = \texttt{zeros(n, 1)};\ \texttt{q} = \texttt{randperm(n)};\\
&\texttt{p(q(1 : n_spikes))} = \texttt{sign(randn(n_spikes, 1))};\\
&\texttt{B} = \texttt{randn(m, n)};\ \texttt{B} = \texttt{orth(B')'};\\
&\texttt{bf} = \texttt{B} * \texttt{p};\ \texttt{rk} = \texttt{randn(m, 1)};\\
&\texttt{b} = \texttt{bf} + \texttt{sigma} * \texttt{norm(bf)/norm(rk)} * \texttt{rk};
\end{aligned}$$

with $\texttt{spike\_rate} = 0.1$ and the levels of noise $\texttt{sigma} = 0.4, 0.6, 0.8$. The lower and upper bounds on the variables are generated by

$$\underline{\texttt{x}} = 0.05 * \texttt{ones(n)}, \quad \overline{\texttt{x}} = 0.95 * \texttt{ones(n)},$$

respectively. Since among the problems given in (37) only L22L22R is differentiable, we need some nonsmooth algorithms to be compared with OSGA. In our experiment, we consider two versions of OSGA, i.e., one version uses BCSS for solving the subproblem (9) (OSGA-1) and another version uses the inexact solution described in Sect. 3.2 for solving the subproblem (9) (OSGA-2), compared with PSGA-1 (a projected subgradient algorithm with nonsummable diminishing step-size), and PSGA-2 (a projected subgradient algorithm with nonsummable diminishing steplength), cf. Boyd et al. (2003).

The results for L22L22R, L22L1R, L1L22R, and L1L1R are illustrated in Table 1 and Fig. 1. We first run OSGA-2 and stop it after 100 iterations for each problem of (37) and set the best founded function value as f_b. Then we stop the other algorithms once they achieve a function value less or equal than f_b or after 2000 iterations. Figure 1 displays the relative error of function vales versus iterations

$$\delta_k := \frac{f_k - \widehat{f}}{f_0 - \widehat{f}}, \tag{38}$$

where $\widehat{f} = f_b - 0.01 f_b$ denotes an approximation of the minimum and f_0 shows the function value on an initial point x^0. In our experiments, PSGA-1 and PSGA-2 exploit the step-sizes $\alpha := 1/\sqrt{k}\|g_k\|$ and $\alpha := 0.1/\sqrt{k}$, respectively, in which k is the iteration counter and g_k is a subgradient of f at x_k.

In Table 1, N_i and T denote the total number of iterations and the running time, respectively. From this table, we can see that for the problems L22L1R and L1L1R, OSGA-1 and OSGA-2 outperform PSGA-1 and PSGA-2 significantly; however, for L22L22R and L1L22R PSGA-2 attains a comparable or better results than OSGA-1. In Fig. 1, we illustrate the relative error δ_k versus iterations for several levels of noise and regularization parameters. It is clear that the considered algorithms have a good behaviour by increasing the levels of noise. Subfigures (a)–(f) and (j)-(l) show that OSGA-1 and OSGA-2 outperform PSGA-1 and PSGA-2 substantially with respect to the relative error of function values δ_k (38); however, from subfigures (g)–(i), PSGA-2 attains the best results but rather comparable with OSGA-1 and OSGA-2. These results show that OSGA-1 and OSGA-2 are suitable for the sparse signal recovery with the

Table 1 Result summary for solving L22L22R, L22L1R, L1L22R, and L1L1R, where N_i and T denote the number of iterations and the running time, respectively

Problem	sigma	λ	PSGA-1		PSGA-2		OSGA-1		OSGA-2	
			N_i	T	N_i	T	N_i	T	N_i	T
L22L22	0.4	1.3	421	0.23	266	0.15	36	0.16	100	0.85
L22L22	0.4	1.4	430	0.29	222	0.20	52	0.23	100	0.79
L22L22	0.4	1.5	403	0.23	201	0.11	91	0.32	100	0.91
L22L22	0.6	1.3	438	0.24	253	0.14	77	0.26	100	0.82
L22L22	0.6	1.4	439	0.29	235	0.17	45	0.18	100	0.96
L22L22	0.6	1.5	417	0.30	191	0.16	54	0.22	100	1.03
L22L22	0.8	1.3	432	0.25	246	0.18	28	0.12	100	0.94
L22L22	0.8	1.4	436	0.28	229	0.15	37	0.19	100	0.85
L22L22	0.8	1.5	409	0.27	190	0.13	47	0.19	100	0.95
L22L1	0.4	0.3	2000	1.32	2000	1.15	12	0.07	100	0.90
L22L1	0.4	0.4	2000	1.26	1842	1.11	8	0.06	100	0.96
L22L1	0.4	0.5	2000	1.19	1545	0.93	9	0.07	100	0.87
L22L1	0.6	0.3	2000	1.07	2000	1.06	10	0.06	100	0.94
L22L1	0.6	0.4	2000	1.02	1865	1.05	9	0.09	100	0.83
L22L1	0.6	0.5	2000	1.14	1196	0.73	8	0.06	100	0.94
L22L1	0.8	0.3	2000	1.10	2000	1.06	8	0.06	100	0.93
L22L1	0.8	0.4	2000	1.16	2000	1.09	8	0.06	100	0.85
L22L1	0.8	0.5	2000	1.13	1363	1.11	9	0.05	100	0.98
L1L22	0.4	3.0	388	0.23	43	0.03	32	0.14	100	0.84
L1L22	0.4	3.1	381	0.31	37	0.05	43	0.17	100	0.74
L1L22	0.4	3.2	371	0.25	32	0.03	37	0.19	100	0.89
L1L22	0.6	3.0	395	0.28	37	0.03	38	0.18	100	0.79
L1L22	0.6	3.1	379	0.30	38	0.03	48	0.19	100	0.80
L1L22	0.6	3.2	371	0.25	33	0.05	43	0.19	100	0.79
L1L22	0.8	3.0	382	0.23	36	0.03	40	0.16	100	0.76
L1L22	0.8	3.1	375	0.26	32	0.04	47	0.22	100	0.84
L1L22	0.8	3.2	371	0.27	32	0.04	37	0.16	100	0.92
L1L1	0.4	0.8	2000	1.18	410	0.31	17	0.10	100	0.83
L1L1	0.4	0.9	2000	1.22	446	0.33	18	0.11	100	0.83
L1L1	0.4	1.0	2000	1.36	370	0.25	14	0.09	100	1.03
L1L1	0.6	0.8	2000	1.23	301	0.21	17	0.10	100	1.07
L1L1	0.6	0.9	2000	1.13	442	0.27	16	0.11	100	1.01
L1L1	0.6	1.0	2000	1.20	485	0.38	17	0.11	100	0.91
L1L1	0.8	0.8	2000	1.39	444	0.31	21	0.11	100	0.99
L1L1	0.8	0.9	2000	1.25	419	0.25	11	0.07	100	1.02
L1L1	0.8	1.0	2000	1.28	396	0.24	17	0.10	100	1.10

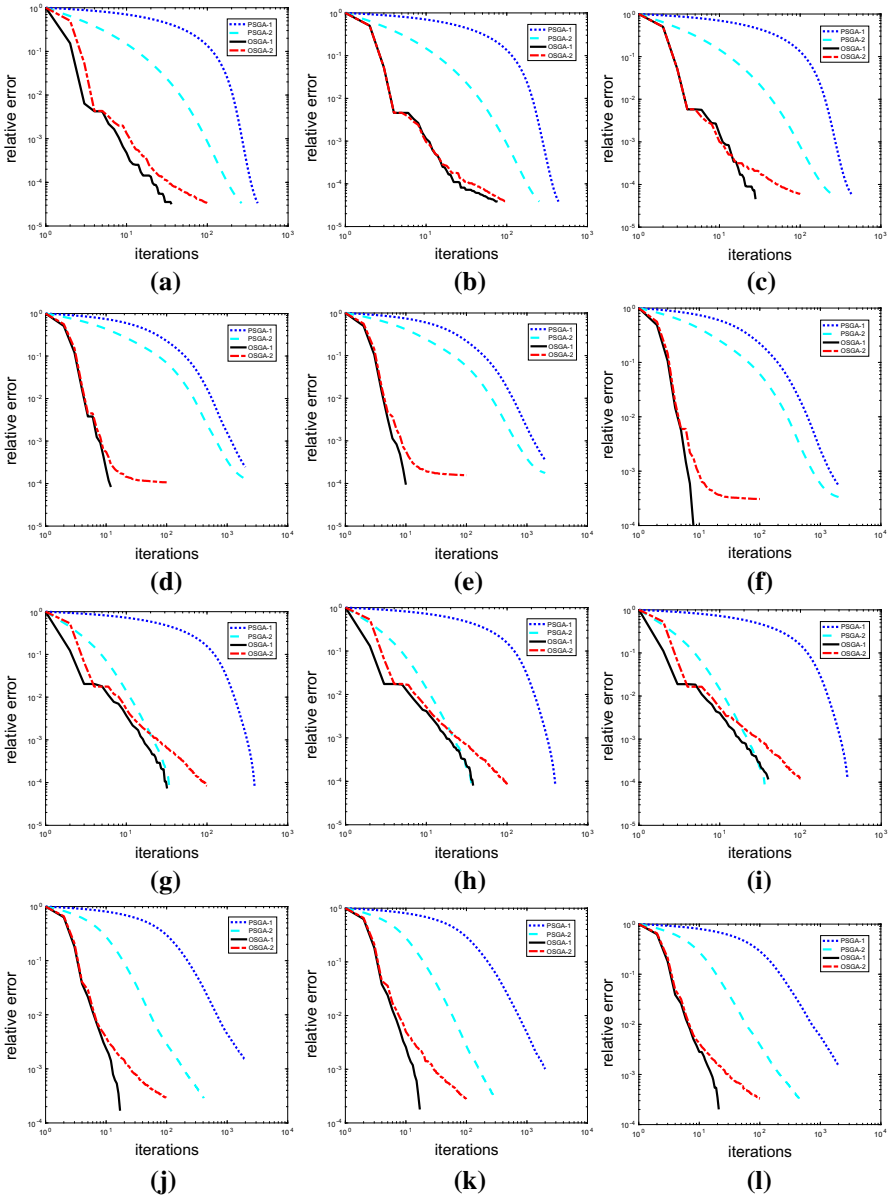

Fig. 1 The relative error δ_k of function values versus iterations of PSGA-1, PSGA-2, OSGA-1, and OSGA-2 for the problems L22L22R, L22L1R, L22L22R, and L22L1R with several levels of noise and regularization parameters. **a** L22L22R, $\sigma = 0.4$, $\lambda = 1.3$; **b** L22L22R, $\sigma = 0.6$, $\lambda = 1.3$; **c** L22L22R, $\sigma = 0.8$, $\lambda = 1.3$; **d** L22L1R, $\sigma = 0.4$, $\lambda = 0.3$; **e** L22L1R, $\sigma = 0.6$, $\lambda = 0.3$; **f** L22L1R, $\sigma = 0.8$, $\lambda = 0.3$; **g** L1L22R, $\sigma = 0.4$, $\lambda = 3.0$; **h** L1L22R, $\sigma = 0.6$, $\lambda = 3.0$; **i** L1L22R, $\sigma = 0.8$, $\lambda = 3.0$; **j** L1L1R, $\sigma = 0.4$, $\lambda = 0.8$; **k** L1L1R, $\sigma = 0.6$, $\lambda = 0.8$; **l** L1L1R, $\sigma = 0.8$, $\lambda = 0.8$

ℓ_1 regularizer. It can also be seen that OSGA-1 (using BCSS) performs much better than OSGA-2 (using inexact scheme) for this medium-scale problem.

4.2 Two-dimensional image deblurring

Image deblurring is one of the fundamental tasks in the context of digital imaging processing, aiming at recovering an image from a blurred/noisy observation. The problem is typically modeled as linear inverse problem

$$y = Ax + \omega, \quad x \in V, \tag{39}$$

where V is a finite-dimensional vector space, A is a blurring linear operator, x is a clean image, y is an observation, and ω is either Gaussian or impulsive noise.

The system of Eq. (39) is usually underdetermined and ill-conditioned, and ω is not commonly available, so it is not possible to solve it directly, see Neumaier (1998). Hence the solution is generally approximated by an optimization problem of the form

$$\min_{x \in V} \quad \frac{1}{2}\|Ax - b\|_2^2 + \lambda\varphi(x) \tag{40}$$

where φ is a smooth or nonsmooth regularizer such as $\varphi(x) = \frac{1}{2}\|x\|_2^2$, $\varphi(x) = \|x\|_1$, and $\varphi(x) = \|x\|_{ITV}$ in which ITV stands for the isotropic total variation. Among the various regularizers, the total variation is much more popular due to its strong edge preserving feature, see, e.g., Chambolle et al. (2010). Isotropic total variation is defined for $x \in \mathbb{R}^{m\times n}$ by

$$\begin{aligned}\|x\|_{ITV} = &\sum_i^{m-1}\sum_j^{n-1}\sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2} \\ &+ \sum_i^{m-1}|x_{i+1,n} - x_{i,n}| + \sum_j^{n-1}|x_{m,j+1} - x_{m,j}|.\end{aligned}$$

The common drawback of the unconstrained problem (40) is that it usually gives a solution outside of the dynamic range of the image, which is either [0, 1] or [0, 255] for 8-bit gray-scale images. Hence one has to project the unconstrained solution to the dynamic range of the image. However, the quality of the projected images is not always acceptable. As a result, it is worth to solve a bound-constrained problem of the form (2) in place of the unconstrained problem (40), where the bounds are defined by the dynamic range of the images, see Beck and Teboulle (2009), Chan et al. (2013) and Woo and Yun (2013).

The comparison concerning the quality of the recovered image is made via the so-called peak signal-to-noise ratio (PSNR) defined by

$$\text{PSNR} = 20\log_{10}\left(\frac{\sqrt{mn}}{\|x - x_t\|_F}\right) \tag{41}$$

and the improvement in signal-to-noise ratio (ISNR) defined by

$$\text{ISNR} = 20\log_{10}\left(\frac{\|y - x_t\|_F}{\|x - x_t\|_F}\right), \tag{42}$$

where $\|\cdot\|_F$ is the Frobenius norm, x_t denotes the $m \times n$ true image, y is the observed image, and pixel values are in $[0, 1]$.

We here consider the image restoration from a blurred/noisy observation using the model (2) equipped with the isotropic total variation regularizer. We employ OSGA, MFISTA (a monotone version of FISTA proposed by Beck and Teboulle (2009)), ADMM (an alternating direction method proposed by Chan et al. (2013)), and a projected subgradient algorithm PSGA (with nonsummable diminishing step-size, see Boyd et al. (2003)). In our implementation, we use the original code of MFISTA and ADMM provided by the authors, with minor adaptations about the stopping criterion.

We here restore the 512×512 blurred/noisy Barbara image. Let y be a blurred/noisy version of this image generated by a 9×9 uniform blur and adding a Gaussian noise with zero mean and the standard deviation set to $\sigma = 0.02,\ 0.04,\ 0.06,\ 0.08$. Our implementation shows that the algorithms are sensitive to the regularization parameter λ. Hence we consider three different regularization parameters $\lambda = 1 \times 10^{-2}$, $\lambda = 7 \times 10^{-3}$, and $\lambda = 4 \times 10^{-3}$. We run MFISTA for the deblurring problem, stop it after 25 iterations, and set f_b to the best function value found. Then we stop the other algorithms as soon as a function value less or equal than f_b is achieved or after 50 iterations. The results of our implementation are summarized in Table 2, Figs. 2, 3, and 4.

The results of Table 2, Figs. 2 and 3 show that the PSNR, and ISNR produced by the algorithms are sensitive to the regularization parameter λ; the function values are somewhat less sensitive. From Table 2, it can be seen that the running time of PSGA, ADMM, and OSGA are comparable and much better than MFISTA, and OSGA

Table 2 Result summary for the l_2^2 isotropic total variation, where PSNR and T denote the peak signal-to-noise (41) and the running time, respectively

Noise level	λ	PSGA		MFISTA		ADMM		OSGA	
		PSNR	T	PSNR	T	PSNR	T	PSNR	T
0.2	$4e{-}3$	23.65	1.31	23.86	5.01	23.76	1.11	23.89	1.44
0.2	$7e{-}3$	23.58	1.52	23.79	5.24	23.70	1.27	23.83	1.63
0.2	$1e{-}2$	23.56	1.30	23.73	4.91	23.64	1.05	23.79	1.62
0.4	$4e{-}3$	23.07	1.39	23.59	4.73	23.48	1.09	23.64	1.83
0.4	$7e{-}3$	23.10	1.26	23.61	4.71	23.53	1.14	23.70	2.00
0.4	$1e{-}2$	23.08	1.37	23.61	5.36	23.53	1.27	23.71	2.03
0.6	$4e{-}3$	21.99	1.25	23.13	5.00	23.01	0.97	23.18	2.15
0.6	$7e{-}3$	22.06	1.23	23.39	5.03	23.30	1.13	23.47	2.69
0.6	$1e{-}2$	22.10	1.31	23.04	5.27	23.35	1.07	23.50	2.21
0.8	$4e{-}3$	20.66	1.22	22.37	4.80	22.25	0.72	22.39	2.32
0.8	$7e{-}3$	20.80	1.28	23.97	4.97	22.89	1.11	23.01	2.77
0.8	$1e{-}2$	20.85	1.26	23.19	5.07	23.12	1.14	23.26	2.47

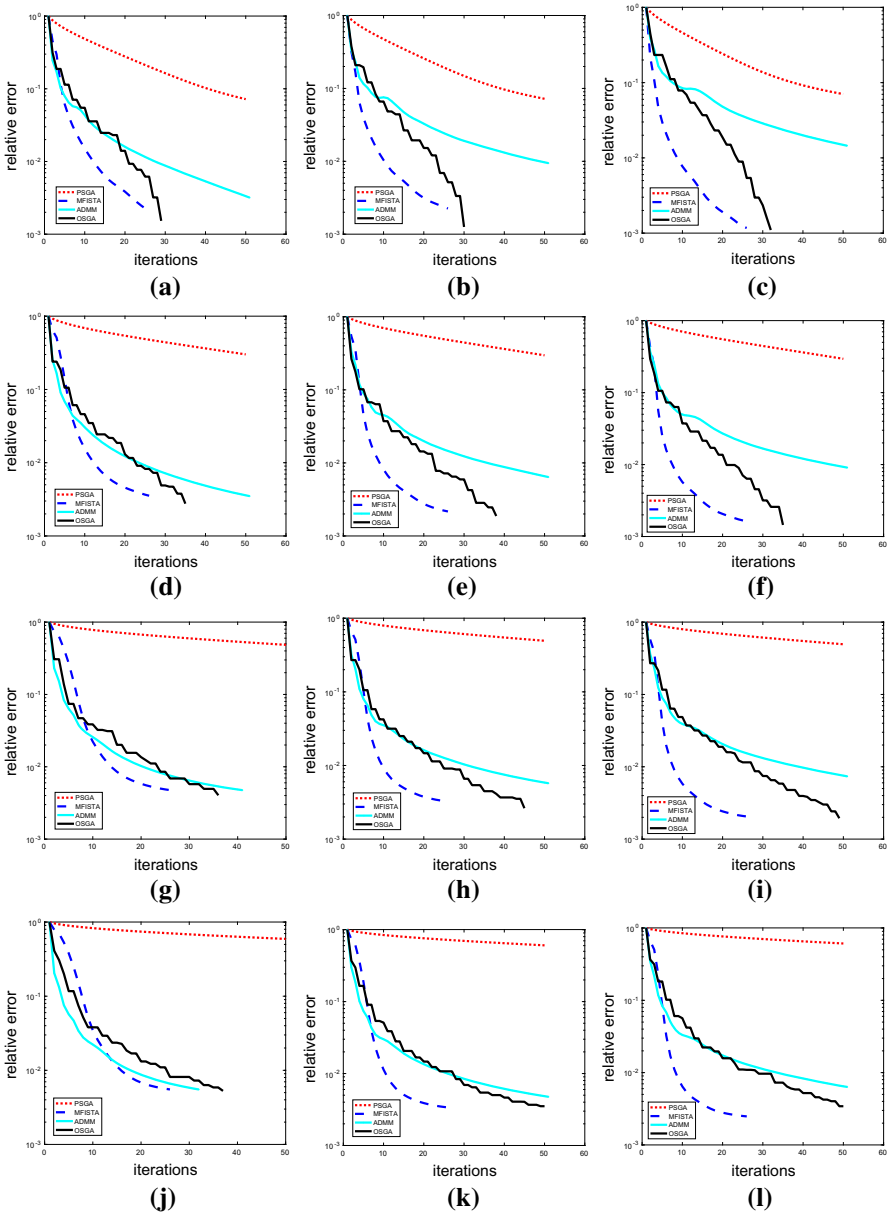

Fig. 2 The relative error δ_k of function values versus iterations of PSGA, MFISTA, ADMM, and OSGA for deblurring the 512×512 Barbara image with the 9×9 uniform blur and the Gaussian noise with deviation $\sigma = 0.02,\ 0.04,\ 0.06,\ 0.08$. **a** $\sigma = 0.02, \lambda = 4 \times 10^{-3}$; **b** $\sigma = 0.02, \lambda = 7 \times 10^{-3}$; **c** $\sigma = 0.02, \lambda = 1 \times 10^{-2}$; **d** $\sigma = 0.04, \lambda = 4 \times 10^{-3}$; **e** $\sigma = 0.04, \lambda = 7 \times 10^{-3}$; **f** $\sigma = 0.04, \lambda = 1 \times 10^{-2}$; **g** $\sigma = 0.06, \lambda = 4 \times 10^{-3}$; **h** $\sigma = 0.06, \lambda = 7 \times 10^{-2}$; **i** $\sigma = 0.06, \lambda = 1 \times 10^{-2}$; **j** $\sigma = 0.08, \lambda = 4 \times 10^{-3}$; **k** $\sigma = 0.08, \lambda = 7 \times 10^{-3}$; **l** $\sigma = 0.08, \lambda = 1 \times 10^{-2}$

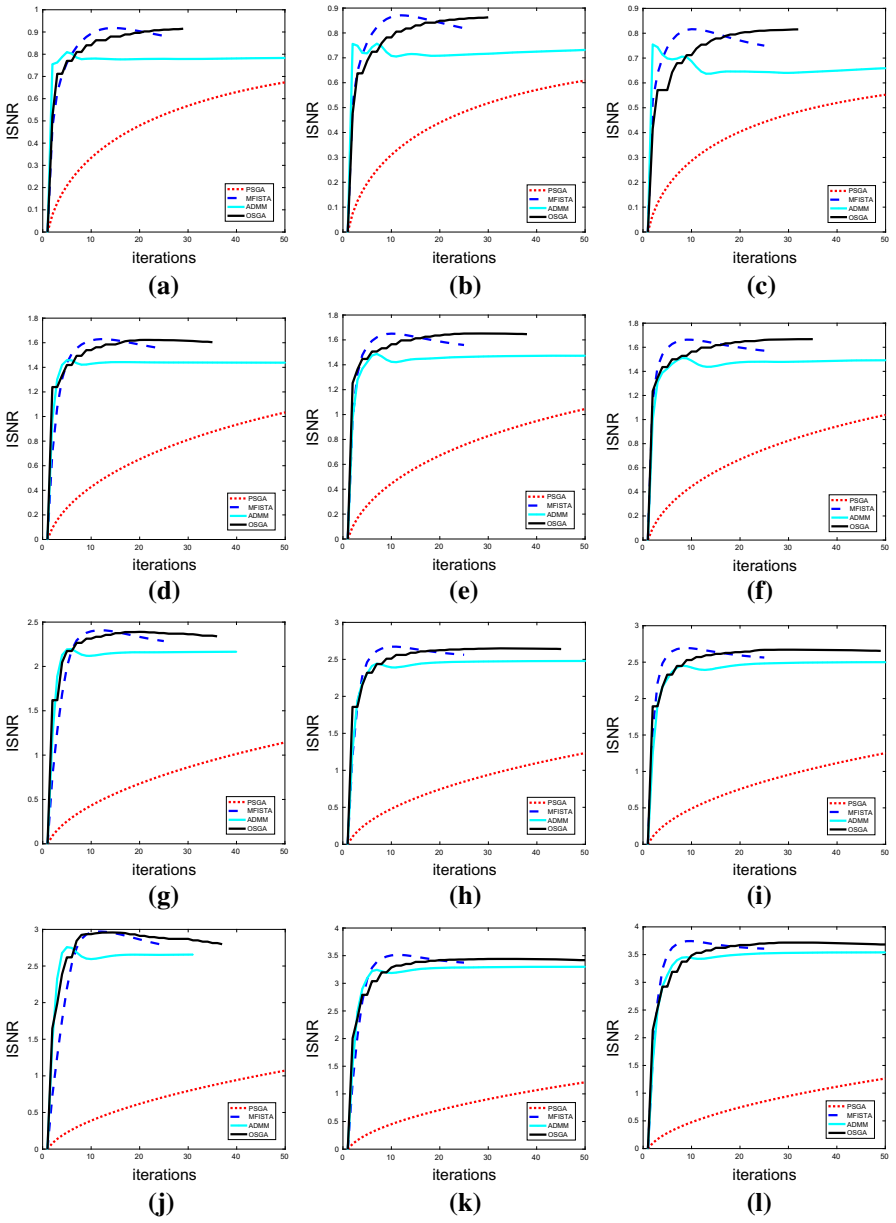

Fig. 3 ISNR versus iterations of PSGA, MFISTA, ADMM, and OSGA for deblurring the 512×512 Barbara image with the 9×9 uniform blur and the Gaussian noise with deviations $\sigma = 0.02,\ 0.04,\ 0.06,\ 0.08$. **a** $\sigma = 0.02$, $\lambda = 4 \times 10^{-3}$; **b** $\sigma = 0.02$, $\lambda = 7 \times 10^{-3}$; **c** $\sigma = 0.02$, $\lambda = 1 \times 10^{-2}$; **d** $\sigma = 0.04$, $\lambda = 4 \times 10^{-3}$; **e** $\sigma = 0.04$, $\lambda = 7 \times 10^{-3}$; **f** $\sigma = 0.04$, $\lambda = 1 \times 10^{-2}$; **g** $\sigma = 0.06$, $\lambda = 4 \times 10^{-3}$; **h** $\sigma = 0.06$, $\lambda = 7 \times 10^{-2}$; **i** $\sigma = 0.06$, $\lambda = 1 \times 10^{-2}$; **j** $\sigma = 0.08$, $\lambda = 4 \times 10^{-3}$; **k** $\sigma = 0.08$, $\lambda = 7 \times 10^{-3}$; **l** $\sigma = 0.08$, $\lambda = 1 \times 10^{-2}$

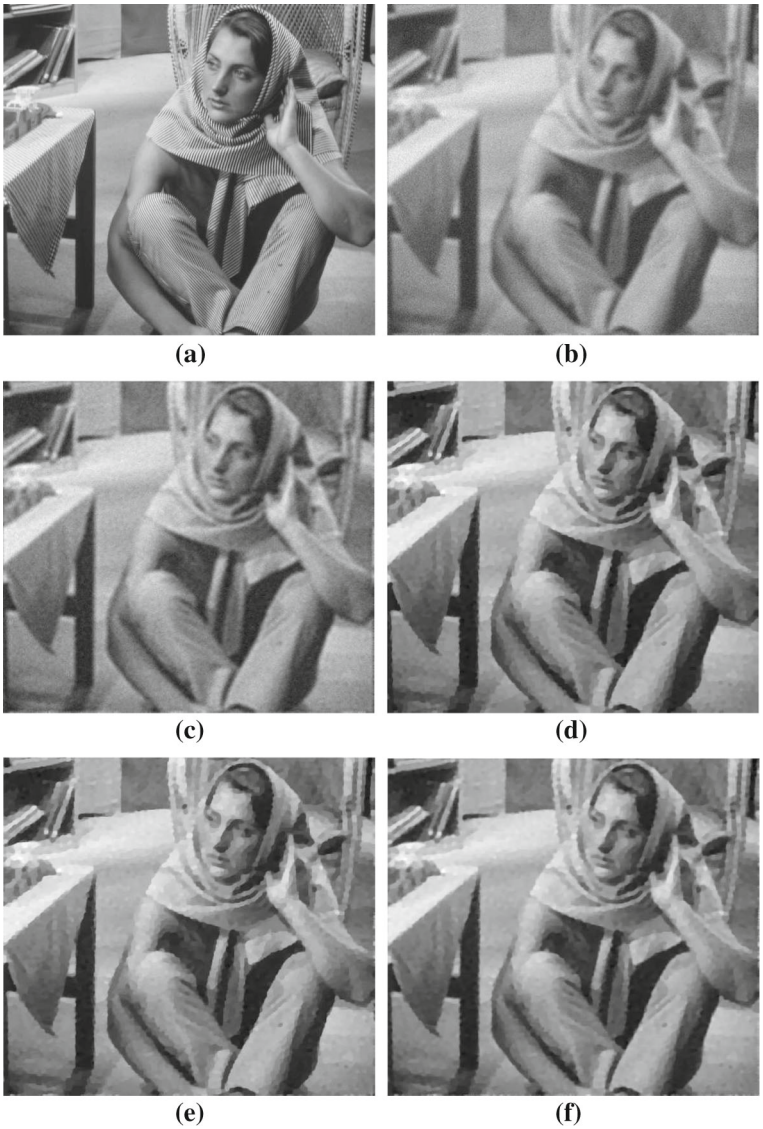

Fig. 4 A comparison among PSGA, MFISTA, ADMM, and OSGA for deblurring the 512×512 Barbara image with the 9×9 uniform blur and the Gaussian noise with the deviation 0.04 and the regularization parameter $\lambda = 4 \times 10^{-3}$. **a** Original image. **b** Blurred/noisy image. **c** PSGA: PSNR $= 23.07$ and $T = 1.39$. **d** MFISTA: PSNR $= 23.59$ and $T = 4.73$. **e** ADMM: PSNR $= 23.48$ and $T = 1.09$. **f** OSGA: PSNR $= 23.64$ and $T = 1.83$

attains the best PSNR. Figure 2 shows that MFISTA and then OSGA attains the better function value; however, MFISTA needs much more time. Figure 3 shows that OSGA outperforms the other methods with respect to ISNR. Figure 4 displays the original Barbara image, the blurred/noisy image, and the recovered images by PSGA, MFISTA, ADMM, and OSGA for the regularization parameter $\lambda = 4 \times 10^{-3}$.

5 Concluding remarks

This paper discussed how to apply the optimal subgradient algorithm OSGA to the task of solving bound-constrained convex optimization problems. It is shown that the solution of the auxiliary OSGA subproblem needed in each iteration has a piecewise linear form in a single variable.

We give two iterative schemes to solve this one-dimensional problem; one solves the OSGA subproblem exactly in polynomial time, the other inexactly but for very large problems significantly faster. The first scheme translates the subproblem into a one-dimensional piecewise rational problem, which allows the global optimizer of the subproblem to be found in $\mathcal{O}(n^2)$ operations. The second scheme solves a one-dimensional nonlinear equation with a standard zero finders and gives only an approximate, local optimizer. The exact scheme BCSS is suitable for small- and medium-scale problems, while the inexact version can be successfully applied even to very large-scale problems.

Numerical results are reported showing the efficiency of OSGA compared with some state-of-the-art algorithms.

Acknowledgements Open access funding provided by University of Vienna. We would like to thank the associate editor and the referees for a number of suggestions that improved the paper.

References

Ahookhosh M (2014) User's manual for OSGA (optimal subgradient algorithm). http://homepage.univie.ac.at/masoud.ahookhosh/uploads/User's_manual_for_OSGA.pdf

Ahookhosh M (2016) Optimal subgradient algorithms with application to large-scale linear inverse problems. Submitted. arXiv:1402.7291

Ahookhosh M, Neumaier A (2013) High-dimensional convex optimization via optimal affine subgradient algorithms. In: ROKS workshop, pp 83–84

Ahookhosh M, Neumaier A (2016) Solving structured nonsmooth convex optimization with complexity $\mathcal{O}(\varepsilon^{-1/2})$ (submitted)

Ahookhosh M, Neumaier A (2016) An optimal subgradient algorithm with subspace search for costly convex optimization problems (submitted)

Ahookhosh M, Neumaier A (2017) Optimal subgradient algorithms for large-scale convex optimization in simple domains. Numerical Algorithms

Bardsley J, Vogel CR (2003) A nonnegatively constrained convex programming method for image reconstruction. SIAM J Sci Comput 25:1326–1343

Beck A, Teboulle M (2009) Fast gradient-based algorithms for constrained total variation image denoising and deblurring. IEEE Trans Image Process 18(11):2419–2434

Birgin EG, Martinez JM, Raydan M (2000) Nonmonotone spectral projected gradient methods on convex sets. SIAM J Optim 10:1196–1211

Boţ RI, Hendrich C (2013) A Douglas–Rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators. SIAM J Optim 23(4):2541–2565

Boţ RI, Csetnek ER, Hendrich C (2013) A primal-dual splitting algorithm for finding zeros of sums of maximally monotone operators. SIAM J Optim 23:2011–2036

Boyd S, Xiao L, Mutapcic A (2003) Subgradient methods. http://www.stanford.edu/class/ee392o/subgrad_method.pdf
Branch MA, Coleman TF, Li Y (1999) A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. SIAM J Sci Comput 21:1–23
Byrd RH, Lu P, Nocedal J, Zhu C (1995) A limited memory algorithm for bound constrained optimization. SIAM J Sci Comput 16:1190–1208
Chambolle A, Caselles V, Cremers D, Novaga M, Pock T (2010) An introduction to total variation for image analysis, In: Theoretical foundations and numerical methods for sparse recovery, vol. 9, De Gruyter, Radon Series Comp. Appl. Math, pp 263–340
Chan RH, Tao M, Yuan X (2013) Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers. SIAM J Imaging Sci 6(1):680–697
Dai YH, Fletcher R (2006) New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. Math Program 106:403–421
Elfving T, Hansen PC, Nikazad T (2012) Semiconvergence and relaxation parameters for projected SIRT algorithms. SIAM J Sci Comput 34(4):2000–2017
Esser E, Lou Y, Xin J (2013) A method for finding structured sparse solutions to nonnegative least squares problems with applications. SIAM J Imaging Sci 6(4):2010–2046
Karmitsa N, Mäkelä MM (2010) Limited memory bundle method for large bound constrained nonsmooth optimization: convergence analysis. Optim Methods Softw 25(6):895–916
Karmitsa N, Mäkelä MM (2010) Adaptive limited memory bundle method for bound constrained large-scale nonsmooth optimization. Optimization 59(6):945–962
Kaufman L, Neumaier A (1996) PET regularization by envelope guided conjugate gradients. IEEE Trans Med Imaging 15:385–389
Kaufman L, Neumaier A (1997) Regularization of ill-posed problems by envelope guided conjugate gradients. J Comput Graph Stat 6(4):451–463
Kim D, Sra S, Dhillon IS (2010) Tackling box-constrained optimization via a new projected quasi-Newton approach. SIAM J Sci Comput 32:3548–3563
Hager WW, Zhang H (2006) A new active set algorithm for box constrained optimization. SIAM J Optim 17:526–557
Hager WW, Zhang H (2013) The limited memory conjugate gradient method. SIAM J Optim 23:2150–2168
Helgason R, Kennington J, Lall H (1980) A polynomially bound algorithms for a singly constrained quadratic program. Math Program 18:338–343
Lin CJ, Moré JJ (1999) Newton's method for large bound-constrained optimization problems. SIAM J Optim 9:1100–1127
Morini S, Porcelli M, Chan RH (2010) A reduced Newton method for constrained linear least squares problems. J Comput Appl Math 233:2200–2212
Nemirovsky AS, Yudin DB (1983) Problem complexity and method efficiency in optimization. Wiley, New York
Nesterov Y (2004) Introductory lectures on convex optimization: a basic course. Kluwer, Dordrecht
Nesterov Y (2005) Smooth minimization of non-smooth functions. Math Program 103:127–152
Neumaier A (1998) Solving ill-conditioned and singular linear systems: a tutorial on regularization. SIAM Rev 40(3):636–666
Neumaier A (2001) Introduction to numerical analysis. Cambridge University Press, Cambridge
Neumaier A (2016) OSGA: a fast subgradient algorithm with optimal complexity. Mathem Program 158:1–21
Neumaier A, Azmi B (2016) LMBOPT: a limited memory method for bound-constrained optimization, manuscript, University of Vienna
Pardalos PM, Kovoor N (1990) An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. Math Program 46:321–328
Woo H, Yun S (2013) Proximal linearized alternating direction method for multiplicative denoising. SIAM J Sci Comput 35:336–358
Wright SJ, Nowak RD, Figueiredo MAT (2009) Sparse reconstruction by separable approximation. IEEE Trans Signal Process 57(7):2479–2493
Zhang J, Morini B (2013) Solving regularized linear least-squares problems by the alternating direction method with applications to image restoration. Electron Trans Numer Anal 40:356–372
Zhang X, Saha A, Vishwanathan SVN (2011) Lower bounds on rate of convergence of cutting plane methods. In: Advances in neural information processing systems 23