

AN OPTIMAL SUBGRADIENT ALGORITHM FOR LARGE-SCALE BOUND-CONSTRAINED CONVEX OPTIMIZATION

MASOUD AHOOKHOSH* AND ARNOLD NEUMAIER†

Abstract. This paper shows that the OSGA algorithm – 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 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.

Key words. bound-constrained convex optimization, nonsmooth optimization, first-order black-box oracle, subgradient methods, optimal complexity, high-dimensional data

AMS subject classifications. 90C25 90C60 90C06 65K05

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

$$(1.1) \quad \begin{array}{ll} \min & f(\mathcal{A}x) \\ \text{s.t.} & x \in \mathbf{x}, \end{array}$$

where $f : \mathbf{x} \rightarrow \mathbb{R}$ is a – smooth or nonsmooth – convex function, $\mathcal{A} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a linear operator, and $\mathbf{x} = [\underline{x}, \bar{x}]$ is an axiparallel box in \mathcal{V} in which \underline{x} and \bar{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 $g \in \mathcal{V}^*$ at $x \in \mathcal{V}$. A subgradient of the objective function f at x is a vector $g(x) \in \mathcal{V}^*$ satisfying

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

for all $z \in \mathcal{V}$. It is assumed that the set of optimal solutions of (1.1) 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 & 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 lots of bound-constrained algorithms and solvers for smooth and nonsmooth optimization; here, we mention only those related to our study. LIN & MORE in [28] and KIM et al. in [24] proposed Newton and quasi-Newton methods for solving bound-constrained optimization. In 1995, BYRD et al. [14] proposed a limited

*Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria. masoud.ahookhosh@univie.ac.at

†Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria. Arnold.Neumaier@univie.ac.at

memory algorithm called LBFGS-B for general smooth nonlinear bound-constrained optimization. BRANCH et al. in [13] proposed a trust-region method to solve this problem. NEUMAIER & AZMI [36] solved this problem by a limited memory algorithm. The smooth bound-constrained optimization problem was also solved by BIRGIN et al. in [9] and HAGER & ZHANG in [25, 26] 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 & MÄKELÄ [20, 21].

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:

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

$$(1.2) \quad \begin{aligned} \min \quad & f(x) := \frac{1}{2} \|Ax - b\|_2^2 + \lambda \varphi(x) \\ \text{s.t.} \quad & x \in \mathbf{x}, \end{aligned}$$

and the bound-constrained l_1 problem is given by

$$(1.3) \quad \begin{aligned} \min \quad & f(x) := \|Ax - b\|_1 + \lambda \varphi(x) \\ \text{s.t.} \quad & x \in \mathbf{x}, \end{aligned}$$

where $\mathbf{x} = [\underline{x}, \bar{x}]$ is a box and φ is a smooth or nonsmooth regularizer, often a weighted power of a norm; see Section 4 for examples.

The problems (1.2) and (1.3) are commonly arising in the context of control and inverse problems, especially for some imaging problems like denoising, deblurring and inpainting. MORINI et al. [29] formulated the bound-constrained least-squares problem (1.2) as a nonlinear system of equations and proposed an iterative method based on a reduced Newton's method. Recently, ZHANG & MORINI [39] used alternating direction methods to solve these problems. More recently, CHAN et al. in [16], BOŦ et al. in [11], and BOŦ & HENDRICH in [10] proposed alternating direction methods, primal-dual splitting methods, and a Douglas-Rachford primal-dual method, respectively, to solve both (1.2) and (1.3) for some applications.

Content. In this paper we show that the optimal subgradient algorithm OSGA proposed by NEUMAIER in [33] can be used for solving bound-constrained problems of the form (1.1). 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 OSGA's 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 the next section, we investigate properties of the solution of the subproblem (2.1) that lead to two algorithms for solving it efficiently. In Section 3, we report numerical results with artificial and real data, where to solve large-scale imaging problems we use the inexact scheme to find the solution of OSGA's subproblem. Finally, Section 4 gives some conclusions.

2. Overview of OSGA. OSGA (see Algorithm 1) is an optimal subgradient algorithm proposed by NEUMAIER in [33] that constructs – for a problem (1.1) with arbitrary nonempty, closed and convex domain, not necessarily a box – a sequence of iterates whose function values converge with the optimal complexity to the minimum of (1.1). Furthermore, OSGA requires no information regarding global parameters like Lipschitz constants of function values and gradients.

Apart from first-order information at the iterates, OSGA requires an efficient routine for finding a maximizer $\hat{x} = U(\gamma, h)$ and the optimal objective value $E(\gamma, h)$ of an auxiliary problem of the form

$$(2.1) \quad \begin{aligned} & \sup && E_{\gamma, h}(x) \\ & \text{s.t.} && x \in \mathbf{x}, \end{aligned}$$

where it is known that the supremum is positive. The function $E_{\gamma, h} : \mathbf{x} \rightarrow \mathbb{R}$ is defined by

$$(2.2) \quad E_{\gamma, h}(x) := -\frac{\gamma + \langle h, x \rangle}{Q(x)}$$

with $\gamma \in \mathbb{R}$, $h \in \mathcal{V}^*$, and $Q : \mathbf{x} \rightarrow \mathbb{R}$ is a prox-function, i.e., a strongly convex function satisfying $\inf_{x \in \mathbf{x}} Q(x) > 0$ and

$$(2.3) \quad Q(z) \geq Q(x) + \langle g_Q(x), z - x \rangle + \frac{\sigma}{2} \|z - x\|^2,$$

where the convexity parameter is $\sigma = 1$. In particular, with $Q_0 > 0$, if $x \in \mathbb{R}^n$, we may take

$$(2.4) \quad Q(x) := Q_0 + \frac{1}{2} \|x - x^0\|_2^2 \quad \text{if } x \in \mathbb{R}^n,$$

where $\|x\|_2 = \sqrt{\sum_i x_i^2}$ is the Euclidean norm, and

$$(2.5) \quad Q(x) := Q_0 + \frac{1}{2} \|x - x^0\|_F^2 \quad \text{if } x \in \mathbb{R}^{m \times n},$$

where $\|x\|_F = \sqrt{\sum_{i,k} x_{ik}^2}$ is the Frobenius norm.

In [1, 33], the unconstrained version (2.1) with the prox-function (2.4) or (2.5) is solved in a closed form. Numerical experiments and comparisons with some state-of-the-art solvers [1, 5, 6] show the promising behaviour of OSGA for solving practical unconstrained problems. A version of OSGA that can solve structured nonsmooth problems with the complexity $O(\varepsilon^{-1/2})$ discussed in [4]. In [3], it is shown that the OSGA subproblem can be solved for many simple domains. In this paper we show that for the above choices of $Q(x)$ and an arbitrary box \mathbf{x} , the subproblem (2.1) can be solved efficiently. It follows that OSGA is applicable to solve bound-constrained convex problems as well.

It is shown in NEUMAIER [33] that OSGA satisfies the optimal complexity bounds $O(\varepsilon^{-2})$ for Lipschitz continuous nonsmooth convex functions and $O(\varepsilon^{-1/2})$ for smooth convex functions with Lipschitz continuous gradients; cf. NEMIROVSKY & YUDIN [30] and NESTEROV [31, 32]. Since the underlying problem (1.1) is a special case of the problem considered in [33], the complexity of OSGA remains valid for (1.1).

As discussed in [33], OSGA uses the following scheme for updating the given parameters α , h , γ , η , and u :

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

Algorithm 2: OSGA (optimal subgradient algorithm)

Input: global parameters: $\delta, \alpha_{\max} \in]0, 1[, 0 < \kappa' \leq \kappa;$ local parameters: $x_0, \mu \geq 0, f_{\text{target}};$ **Output:** $x_b, f_{x_b};$ **begin** choose an initial best point $x_b;$ compute f_{x_b} and $g_{x_b};$ **if** $f_{x_b} \leq f_{\text{target}}$ **then**

stop;

else $h = g_{x_b} - \mu g_Q(x_b); \gamma = f_{x_b} - \mu Q(x_b) - \langle h, x_b \rangle;$ $\gamma_b = \gamma - f_{x_b}; u = U(\gamma_b, h); \eta = E(\gamma_b, h) - \mu;$ **end** $\alpha \leftarrow \alpha_{\max};$ **while** *stopping criteria do not hold* **do** $x = x_b + \alpha(u - x_b);$ compute f_x and $g_x;$ $g = g_x - \mu g_Q(x); \bar{h} = h + \alpha(g - h);$ $\bar{\gamma} = \gamma + \alpha(f_x - \mu Q(x) - \langle g, x \rangle - \gamma);$ $x'_b = \operatorname{argmin}_{z \in \{x_b, x\}} f(z, v_z); f_{x'_b} = \min\{f_{x_b}, f_x\};$ $\gamma'_b = \bar{\gamma} - f_{x'_b}; u' = U(\gamma'_b, \bar{h});$ $x' = x_b + \alpha(u' - x_b);$ compute $f_{x'};$ choose \bar{x}_b in such a way that $f_{\bar{x}_b} \leq \min\{f_{x'_b}, f_{x'}\};$ $\bar{\gamma}_b = \bar{\gamma} - f_{\bar{x}_b}; \bar{u} = U(\bar{\gamma}_b, \bar{h}); \bar{\eta} = E(\bar{\gamma}_b, \bar{h}) - \mu; x_b = \bar{x}_b; f_{x_b} = f_{\bar{x}_b};$ **if** $f_{x_b} \leq f_{\text{target}}$ **then**

stop;

else update the parameters α, h, γ, η and $u;$ **end** **end****end**

3. Solving OSGA's rational subproblem (2.1). In this section we investigate the solution of the bound-constrained subproblem (2.1) and give two iterative schemes, where the first one solves (2.1) exactly and the second one solves it approximately.

3.1. Explicit solution of OSGA's rational subproblem (2.1). In this subsection we describe an explicit solution of the bound-constrained subproblem (2.1).

Without loss of generality, we here consider $\mathcal{V} = \mathbb{R}^n$. It is not hard to adapt the results to $\mathcal{V} = \mathbb{R}^{n \times n}$ and other spaces. The method is related to one used in several earlier papers. In 1980, HELGASON et al. [27] characterized the solution of singly constrained quadratic problem with bound constraints. Later, PARADOLAS & KOVOOR [37] developed an $O(n)$ algorithm for this problem using binary search to solve the associated Kuhn-Tucker system. This problem was also solved by DAI & FLETCHER [17] using a projected gradient method. ZHANG et al. [40] 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 (2.1) is equivalent to an one-dimensional minimization problem and then develop a procedure to calculate its minimizer. We write

$$(3.1) \quad x(\lambda) := \sup\{\underline{x}, \inf\{x^0 - \lambda h, \bar{x}\}\}$$

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

PROPOSITION 3.1. *For $h \neq 0$, the maximum of the subproblem (2.1) is attained at $\hat{x} := x(\lambda)$, where $\lambda > 0$ or $\lambda = +\infty$ is the inverse of the value of the maximum.*

Proof. The function $E_{\gamma,h} : \mathcal{V} \rightarrow \mathbb{R}$ defined by (2.2) is continuously differentiable and from Proposition 2.1 in [33] we have $e := E_{\gamma,h} > 0$. By differentiating both side of the equation $E_{\gamma,h}(x)Q(x) = -\gamma - \langle h, x \rangle$, we obtain

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

At the maximizer \hat{x} , we have $eQ(\hat{x}) = -\gamma - \langle h, \hat{x} \rangle$. Now the first-order optimality conditions imply that for $i = 1, 2, \dots, n$,

$$(3.2) \quad -e(\hat{x}_i - x_i^0) - h_i \begin{cases} \leq 0 & \text{if } \hat{x}_i = \underline{x}_i, \\ \geq 0 & \text{if } \hat{x}_i = \bar{x}_i, \\ = 0 & \text{if } \underline{x}_i < \hat{x}_i < \bar{x}_i. \end{cases}$$

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

$$(3.3) \quad \hat{x}_i = \begin{cases} \underline{x}_i & \text{if } \underline{x}_i \geq x_i^0 - \lambda h_i, \\ \bar{x}_i & \text{if } \bar{x}_i \leq x_i^0 - \lambda h_i, \\ x_i^0 - \lambda h_i & \text{if } \underline{x}_i \leq x_i^0 - \lambda h_i \leq \bar{x}_i. \end{cases}$$

This implies that $\hat{x} = x(\lambda)$. \square

Proposition 3.1 gives the key feature of the solution of the subproblem (2.1) implying that it can be considered in the form of (3.1) with only one variable λ . In the remainder of this section, we focus on deriving the optimal λ .

EXAMPLE. 3.1. *Let first consider a very special case that \mathbf{x} is nonnegative orthant, i.e., $\underline{x}_i = 0$ and $\bar{x}_i = +\infty$, for $i = 1, \dots, n$. Nonnegativity constraint is important*

in many applications, see [7, 18, 19, 22, 23]. In this case we consider the quadratic function

$$(3.4) \quad Q(z) := \frac{1}{2}\|z\|_2^2 + Q_0,$$

where $Q_0 > 0$. In [1], it is proved that (3.4) is a prox-function. By using this prox-function and (3.1), we obtain

$$x(\lambda) = \sup\{\underline{x}, \inf\{-\lambda h, \bar{x}\} = -\lambda(h)_-\},$$

where $(z)_- = \min\{0, z\}$. By Proposition 2.2 of [33], we have

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

By substituting $\lambda = 1/e$ into this equation, we get

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

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

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

This shows that for the nonnegativity constraint the subproblem (2.1) can be solved in a closed form, however, for a general bound-constrained problem, we need a much more sophisticated scheme to solve (2.1).

To derive the optimal $\lambda \geq 0$ in Proposition 3.1, we first determine its permissible range provided by the three conditions considered in (3.3) leading to the interval

$$(3.5) \quad \lambda \in [\underline{\Delta}_i, \bar{\Delta}_i],$$

for each component of x . In particular, if $h_i = 0$, since x^0 is a feasible point, $\hat{x}_i = x_i^0 - \lambda h_i = x_i^0$ satisfies the third condition in (3.3). Thus there is no upper bound for λ , leading to

$$(3.6) \quad \underline{\Delta}_i = 0, \quad \bar{\Delta}_i = +\infty \quad \text{if } \hat{x}_i = x_i^0, \quad h_i = 0.$$

If $h_i \neq 0$, we consider the three cases (i) $\underline{x}_i \geq x_i^0 - \lambda h_i$, (ii) $\bar{x}_i \leq x_i^0 - \lambda h_i$, and (iii) $\underline{x}_i \leq x_i^0 - \lambda h_i \leq \bar{x}_i$ of (3.3). 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

$$(3.7) \quad \underline{\Delta}_i = -\frac{\underline{x}_i - x_i^0}{h_i}, \quad \bar{\Delta}_i = +\infty \quad \text{if } \hat{x}_i = \underline{x}_i, \quad h_i > 0.$$

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

$$(3.8) \quad \underline{\Delta}_i = -\frac{\bar{x}_i - x_i^0}{h_i}, \quad \bar{\Delta}_i = +\infty \quad \text{if } \hat{x}_i = \bar{x}_i, \quad h_i < 0.$$

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

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

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

$$(3.9) \quad \underline{\lambda}_i = 0, \quad \bar{\lambda}_i = -\frac{\bar{x}_i - x_i^0}{h_i} \quad \text{if } \hat{x}_i = x_i^0 - \lambda h_i \in [\underline{x}_i, \bar{x}_i], \quad h_i < 0.$$

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

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

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

$$(3.10) \quad \underline{\lambda}_i = 0, \quad \bar{\lambda}_i = -\frac{\underline{x}_i - x_i^0}{h_i} \quad \text{if } \hat{x}_i = x_i^0 - \lambda h_i \in [\underline{x}_i, \bar{x}_i], \quad h_i > 0.$$

As a result, the following proposition is valid.

PROPOSITION 3.2. *If $x(\lambda)$ is solution of the problem (2.1), then*

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

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

$$(3.11) \quad \underline{\lambda}_j = \begin{cases} -\frac{\underline{x}_j - x_j^0}{h_j} & \text{if } \hat{x}_j = \underline{x}_j, \quad h_j > 0, \\ -\frac{\bar{x}_j - x_j^0}{h_j} & \text{if } \hat{x}_j = \bar{x}_j, \quad h_j < 0, \\ 0 & \text{if } \tilde{x}_j \in [\underline{x}_j, \bar{x}_j], \quad h_j < 0, \\ 0 & \text{if } \tilde{x}_j \in [\underline{x}_j, \bar{x}_j], \quad h_j > 0, \\ 0 & \text{if } h_j = 0, \end{cases} \quad \bar{\lambda}_j = \begin{cases} +\infty & \text{if } \hat{x}_j = \underline{x}_j, \quad h_j > 0, \\ +\infty & \text{if } \hat{x}_j = \bar{x}_j, \quad h_j < 0, \\ -\frac{\bar{x}_j - x_j^0}{h_j} & \text{if } \tilde{x}_j \in [\underline{x}_j, \bar{x}_j], \quad h_j < 0, \\ -\frac{\underline{x}_j - x_j^0}{h_j} & \text{if } \tilde{x}_j \in [\underline{x}_j, \bar{x}_j], \quad h_j > 0, \\ +\infty & \text{if } h_j = 0, \end{cases}$$

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

Proposition 3.2 implies that each element of x satisfies in only one of the conditions (3.6)–(3.10). Thus, for each $i = 1, \dots, n$ with $h_i \neq 0$, we have a single breakpoint

$$(3.12) \quad \tilde{\lambda}_i := \begin{cases} -\frac{\bar{x}_i - x_i^0}{h_i} & \text{if } h_i < 0, \\ -\frac{\underline{x}_i - x_i^0}{h_i} & \text{if } h_i > 0, \\ +\infty & \text{if } h_i = 0. \end{cases}$$

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

$$(3.13) \quad 0 = \lambda_1 < \lambda_2 < \dots < \lambda_m < \lambda_{m+1} = +\infty.$$

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

PROPOSITION 3.3. *The solution $x(\lambda)$ of the auxiliary problem (2.1) defined by (3.1) has the form*

$$(3.14) \quad x(\lambda) = p^k + \lambda q^k \quad \text{for } \lambda \in [\lambda_k, \lambda_{k+1}] \quad (k = 1, 2, \dots, m),$$

where

$$(3.15) \quad p_i^k = \begin{cases} x_i^0 & \text{if } h_i = 0, \\ x_i^0 & \text{if } \lambda_{k+1} \leq \tilde{\lambda}_i, \\ \underline{x}_i & \text{if } \lambda_k \geq \tilde{\lambda}_i, h_i > 0, \\ \bar{x}_i & \text{if } \lambda_k \geq \tilde{\lambda}_i, h_i < 0, \end{cases} \quad q_i^k = \begin{cases} 0 & \text{if } h_i = 0, \\ -h_i & \text{if } \lambda_{k+1} \leq \tilde{\lambda}_i, \\ 0 & \text{if } \lambda_k \geq \tilde{\lambda}_i, h_i > 0, \\ 0 & \text{if } \lambda_k \geq \tilde{\lambda}_i, h_i < 0. \end{cases}$$

Proof. Since $\lambda > 0$, there exists $k \in \{1, \dots, m\}$ such that $\lambda \in [\lambda_k, \lambda_{k+1}]$. Let $i \in \{1, \dots, n\}$. If $h_i = 0$, (3.6) implies $\hat{x}_i = x_i^0$. If $h_i \neq 0$, the way of construction of λ_i for $i = 1, \dots, m$ implies that $\tilde{\lambda}_i \notin (\lambda_k, \lambda_{k+1})$, so two cases are distinguished: (i) $\lambda_{k+1} \leq \tilde{\lambda}_i$; (ii) $\lambda_k \geq \tilde{\lambda}_i$. In Case (i), Proposition 3.2 implies that $\tilde{\lambda}_i = \bar{\lambda}_i$, while it is not possible $\tilde{\lambda}_i \neq \underline{\lambda}_i$. Therefore, either (3.9) or (3.10) holds dependent on the sign of h_i , implying $x_i^0 - \lambda h_i \in [\underline{x}_i, \bar{x}_i]$, so that $p_i^k = x_i^0$ and $q_i^k = -h_i$. In Case (ii), Proposition 3.2 implies that $\tilde{\lambda}_i = \underline{\lambda}_i$, while it is not possible $\tilde{\lambda}_i \neq \bar{\lambda}_i$. Therefore, either (3.7) or (3.8) holds. If $h_i < 0$, then (3.8) holds, i.e., $p_i^k = \bar{x}_i$ and $q_i^k = 0$. Otherwise, (3.7) holds, implying $p_i^k = \underline{x}_i$ and $q_i^k = 0$. This proves the claim. \square

Proposition 3.3 exhibits the solution $x(\lambda)$ of the auxiliary problem (2.1) as a piecewise linear function of λ . In the next result, we show that solving the problem (2.1) is equivalent to maximizing a one-dimensional piecewise rational function.

PROPOSITION 3.4. *The maximal value of the subproblem (2.1) is the maximum of the piecewise rational function $e(\lambda)$ defined by*

$$(3.16) \quad e(\lambda) := \frac{a_k + b_k \lambda}{c_k + d_k \lambda + s_k \lambda^2} \quad \text{if } \lambda \in [\lambda_k, \lambda_{k+1}] \quad (k = 1, 2, \dots, m),$$

where

$$a_k := -\gamma - \langle h, p^k \rangle, \quad b_k := -\langle h, q^k \rangle,$$

$$c_k := Q_0 + \frac{1}{2} \|p^k - x^0\|^2, \quad d_k := \langle p^k - x^0, q^k \rangle, \quad s_k := \frac{1}{2} \|q^k\|^2.$$

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

Proof. By Proposition 3.1 and 3.3, the global minimizer of (2.1) has the form (3.14). For $k = 1, 2, \dots, m$ and $\lambda \in [\lambda_k, \lambda_{k+1}]$, we substitute (3.14) into the function (2.2), and obtain

$$(3.17) \quad \begin{aligned} E_{\gamma, h}(x(\lambda)) &= -\frac{\gamma + \langle h, x^k(\lambda) \rangle}{Q(x^k(\lambda))} = -\frac{\gamma + \langle h, p^k + q^k \lambda \rangle}{Q(p^k + q^k \lambda)} \\ &= -\frac{\gamma + \langle h, p^k \rangle + \langle h, q^k \rangle \lambda}{Q_0 + \frac{1}{2} \|p^k - x^0\|^2 + \langle p^k - x^0, q^k \rangle \lambda + \frac{1}{2} \|q^k\|^2 \lambda^2} = e(\lambda), \end{aligned}$$

as defined in the proposition.

Since $Q_0 > 0$, we have $c_k > 0$ and the denominator of (3.16) is bounded away from zero, implying $4s_k c_k > d_k^2$. It is enough to verify $s_k > 0$. The definition of q_k in (3.15) implies that $h_i \neq 0$ for $i \in I = \{i : \lambda_{k+1} \leq \tilde{\lambda}_i\}$ leading to $q^k \neq 0$ implying $s_k > 0$. \square

The next result leads to a systematic way to maximize the one-dimensional rational problem (3.16).

PROPOSITION 3.5. *Let a, b, c, d , and s be real constants with $c > 0$, $s > 0$, and $4sc > d^2$. Then*

$$(3.18) \quad \phi(\lambda) := \frac{a + b\lambda}{c + d\lambda + s\lambda^2}$$

defines a function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ 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*

$$(3.19) \quad \phi(\hat{\lambda}) = \frac{b}{2s\hat{\lambda} + d}$$

is attained at

$$(3.20) \quad \hat{\lambda} = \frac{-a + \sqrt{a^2 - b(ad - bc)/s}}{b}.$$

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

$$(3.21) \quad \phi(\hat{\lambda}) = \frac{4as}{4cs - d^2},$$

attained at

$$(3.22) \quad \hat{\lambda} = -\frac{d}{2s}.$$

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

Proof. The denominator of (3.18) is positive for all $\lambda \in \mathbb{R}^n$ iff 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

$$(3.23) \quad \lambda = \frac{-a \pm \sqrt{a^2 - b(ad - bc)/s}}{b}.$$

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

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

and we have

$$(3.24) \quad \phi(\lambda) = \frac{\pm w}{c + d\lambda + s\lambda^2}.$$

Since the denominator of this fraction is positive and $w \geq 0$, the positive sign in equation (3.23) gives the maximizer, implying that (3.20) is satisfied. Finally, substituting this maximizer into (3.24) gives

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

so that (3.19) 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 (3.21) is satisfied.

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

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

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

We summarize the results of Propositions 3.1–3.5 into the following algorithm for computing the global optimizer x_b and the optimum e_b of (2.1).

Algorithm 3: BCSS (bound-constrained subproblem solver)

Input: $Q_0, x^0, h, \underline{x}, \bar{x}$;
Output: $u_b = U(\gamma, h), e_b = e(x_b)$;
begin
 for $i = 1, 2, \dots, n$ **do**
 | find $\tilde{\lambda}_i$ by (3.12) using \underline{x} and \bar{x} ;
 end
 determine the breakpoints $\lambda_k, k = 1, \dots, m + 1$, by (3.13);
 $e_b = 0$;
 for $k = 1, 2, \dots, m$ **do**
 | compute p^k and q^k using (3.15);
 | construct $e(\lambda)$ using (3.16) for $[\lambda_k, \lambda_{k+1}]$;
 | find the maximizer $\hat{\lambda}$ of $e(\lambda)$ using Proposition 3.5;
 if $\hat{\lambda} \in [\lambda_k, \lambda_{k+1}]$ **then**
 | compute $e^k = e(\hat{\lambda})$ using Proposition 3.5;
 | $\hat{\lambda}^k = \hat{\lambda}$;
 else
 | $e^k = \max\{e(\lambda_k), e(\lambda_{k+1})\}$;
 | $\hat{\lambda}^k = \operatorname{argmax}_{i \in \{k, k+1\}} \{e(\lambda_i)\}$;
 end
 $E(k) = e^k, LAM(k) = \hat{\lambda}^k$;
 end
 $j = \operatorname{argmax}\{E(i) \mid i = 1, \dots, m\}$;
 $e_b = E(j), \hat{\lambda} = LAM(j), u_b = x^0 - \hat{\lambda}h$;
end

3.2. Inexact solution of OSGA's rational subproblem (2.1). In this subsection we give a scheme to compute an inexact solution for the subproblem (2.1).

We here use the quadratic prox-function (3.4). In view of Proposition 3.1 and Theorem 3.1 in [3], the solution of the subproblem (2.1) is given by $x(\lambda)$ defined in (3.1), where λ can be computed by solving the one-dimensional nonlinear equation

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

in which

$$(3.25) \quad \varphi(\lambda) := \frac{1}{\lambda} \left(\frac{1}{2} \|x(\lambda)\|_2^2 + Q_0 \right) + \gamma + \langle h, x(\lambda) \rangle.$$

The solution of OSGA's subproblem can be found by Algorithm 3 (OSS) in [3]. In [3], it is shown that in many convex domains the nonlinear equation (3.25) can be solved explicitly, however, for the bound-constrained problems it can be only solved approximately.

As discussed in [3], 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 [35]. One can also use the MATLAB's `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 OSGA's rational subproblem (2.1) for solving large-scale imaging problems.

4. Applications and numerical experiments. In this section we report some numerical results to show the performance of OSGA compared with some state-of-the-art algorithms on both artificial and real data.

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

<http://homepage.univie.ac.at/masoud.ahookhosh/>.

The package is written in MATLAB, where the parameters

$$\delta = 0.9; \quad \alpha_{max} = 0.7; \quad \kappa = \kappa' = 0.5; \quad \Psi_{target} = -\infty$$

are used. We also use the prox-function (2.4) with $Q_0 = \frac{1}{2}\|x_0\|_2 + \epsilon$, where ϵ is the machine precision. Some examples for each class of problems are available to show how the user can implement it. The interface to each subprogram in the package is fully documented in the corresponding file. Moreover, the OSGA user's manual [2] 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 corresponding papers or packages. All implementations were executed on a Toshiba Satellite Pro L750-176 laptop with Intel Core i7-2670QM Processor and 8 GB RAM.

4.1. Experiment with artificial data. In this subsection we deal with solving the problem (1.1) with the objective functions

$$(4.1) \quad \begin{aligned} f(x) &= \frac{1}{2}\|Ax - b\|_2^2 + \frac{1}{2}\|x\|_2^2 && (L22L22R), \\ f(x) &= \frac{1}{2}\|Ax - b\|_2^2 + \|x\|_1 && (L22L1R), \\ f(x) &= \|Ax - b\|_1 + \frac{1}{2}\|x\|_2^2 && (L1L22R), \\ f(x) &= \|Ax - b\|_1 + \|x\|_1 && (L1L1R). \end{aligned}$$

The problem is generated by

$$[A, z, x] = \text{i_laplace}(n), \quad b = z + 0.1 * \text{rand},$$

where n is the problem dimension and `i_laplace.m` is a code generating an ill-posed test problem using the inverse Laplace transformation from the Regularization Tools MATLAB package, which is available at

<http://www.imm.dtu.dk/~pcha/Regutools/>.

The upper and lower bounds on variables are generated by

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

respectively. Since among the problems (4.1) 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., a version uses BCSS for solving the subproblem (2.1) (OSGA-1) and a version uses the inexact solution described in Subsection 3.2 for solving the subproblem (2.1) (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), see [12].

We solve all of the above-mentioned problems with the dimensions $n = 2000$ and $n = 5000$. The results for L22L22R and L22L1R are illustrated in Table 1 and Figure 1, and the results for L1L22R and L1L1R are summarized in Table 2 and Figure 2. More precisely, Figures 1 and 2 show the relative error of function vales versus iterations

$$(4.2) \quad \delta_k := (f_k - \hat{f}) / (f_0 - \hat{f}),$$

where \hat{f} denotes 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 $\alpha := 5/\sqrt{k}\|g_k\|$ 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 . The algorithms are stopped after 100 iterations.

Table 1: Result summary for L22L22R and L22L1R

	problem	dimension	PSGA-1	PSGA-2	OSGA-1	OSGA-2
f_b	L22L22R	$n = 2000$	81.8427	77.1302	77.1285	77.1285
Time			0.74	0.75	4.15	3.08
f_b	L22L22R	$n = 5000$	4.7561e+2	4.2646e+2	4.2640e+2	4.2645e+2
Time			3.67	3.58	14.09	7.57
f_b	L22L1R	$n = 2000$	1.8922e+2	1.8827e+2	1.8682e+2	1.2367e+2
Time			0.67	0.61	3.91	1.21
f_b	L22L1R	$n = 5000$	7.0679e+2	6.8084e+2	6.7887e+2	6.8064e+2
Time			3.72	3.42	14.20	7.61

Table 2: Result summary for L1L22R and L1L1R

	problem	dimension	PSGA-1	PSGA-2	OSGA-1	OSGA-2
f_b	L1L22R	$n = 2000$	1.8981e+2	1.9420e+2	1.8671e+2	1.8676e+2
Time			0.69	0.75	4.04	2.73
f_b	L1L22R	$n = 5000$	1.9256e+2	3.4612e+2	1.6971e+2	1.6995e+2
Time			3.69	3.57	14.06	6.83
f_b	L1L1R	$n = 2000$	2.6713e+2	2.6936e+2	2.6536e+2	2.6703e+2
Time			0.76	0.75	4.27	3.02
f_b	L1L1R	$n = 5000$	6.9728e+2	7.4536e+2	6.9411e+2	6.9687e+2
Time			3.68	3.57	14.46	7.46

In Figure 1, Subfigures (a) and (b) show that OSGA-1 and OSGA-2 substantially outperform PSGA-1 and PSGA-2 with respect to the relative error of function values δ_k (4.2), however, they need more running time. In this case OSGA-1 and OSGA-2 are competitive, while OSGA-1 performs better. In Figure 1, Subfigure (c) shows that OSGA-2 produces the best results and the others are competitive. Subfigure (d) of Figure 1 demonstrates that OSGA-1 attains the best results and SPGA-2 and OSGA-2 are competitive but much better than SPGA-1. In Figure 2, Subfigures (a) and (b) show that OSGA-1 and OSGA-2 are comparable but much better than PSGA-1 and PSGA-2. Subfigures (c) and (s) show that the best results produced by OSGA-1 and OSGA-2, respectively.

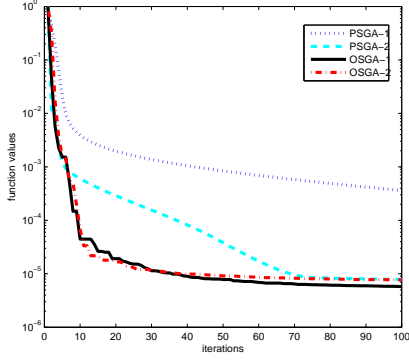
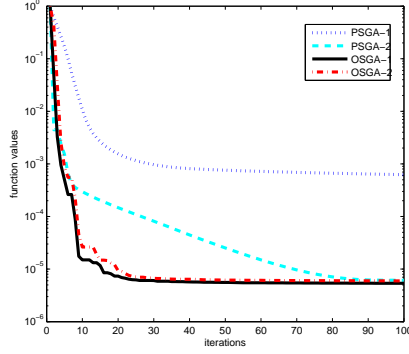
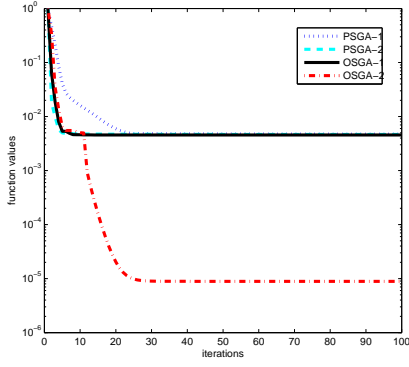
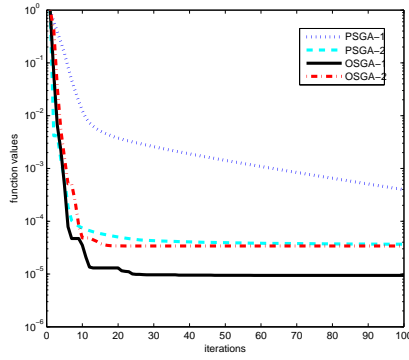
(a) L2L22R, $n = 2000$ (b) L2L22R, $n = 5000$ (c) L2L1R, $n = 2000$ (d) L2L1R, $n = 5000$

Fig. 1: A comparison among PSGA-1, PSGA-2, OSGA-1, and OSGA-2 for solving L2L22R and L2L1R, where the algorithms were stopped after 100 iterations.

4.2. Image deblurring/denoising. Image deblurring/denoising 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 modelled as linear inverse problem

$$(4.3) \quad y = Ax + \omega, \quad x \in \mathcal{V},$$

where \mathcal{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 equation (4.3) is usually underdetermined and ill-conditioned, and ω is not commonly available, so it is not possible to solve it directly, see [34]. Hence the solution is generally approximated by an optimization problem of the form

$$(4.4) \quad \begin{aligned} \min \quad & \frac{1}{2} \|Ax - b\|_2^2 + \lambda \varphi(x) \\ \text{s.t.} \quad & x \in \mathcal{V}, \end{aligned}$$

or

$$(4.5) \quad \begin{aligned} \min \quad & \|Ax - b\|_1 + \lambda \varphi(x) \\ \text{s.t.} \quad & x \in \mathcal{V}, \end{aligned}$$

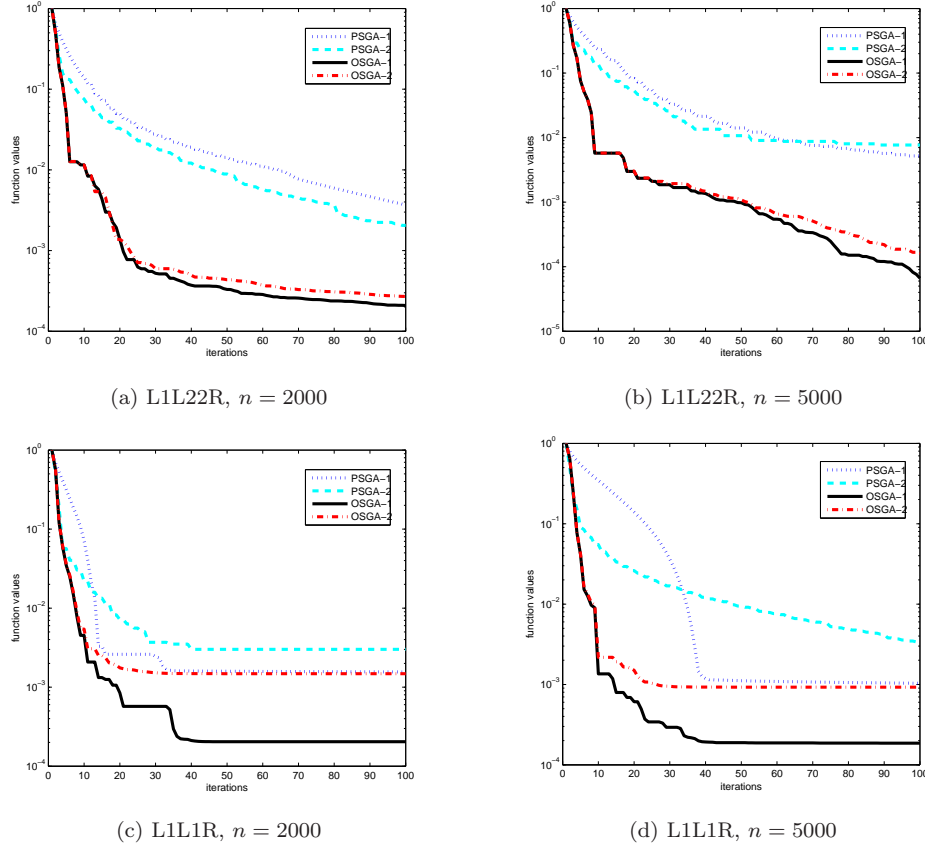


Fig. 2: A comparison among PSGA-1, PSGA-2, OSGA-1, and OSGA-2 for solving L1L22R and L1L1R, where the algorithms were stopped after 100 iterations.

where φ is a smooth or nonsmooth regularizer such as $\varphi(x) = \frac{1}{2}\|x\|_2^2$, $\varphi(x) = \|x\|_1$, $\varphi(x) = \|x\|_{ITV}$, and $\varphi(x) = \|x\|_{ATV}$. Among the various regularizers, the total variation is much more popular due to its strong edge preserving feature. Two important types of the total variation, namely isotropic and anisotropic, see [15], are defined for $x \in \mathbb{R}^{m \times n}$ by

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

and

$$\begin{aligned} \|x\|_{ATV} &= \sum_i^{m-1} \sum_j^{n-1} \{|x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|\} \\ &+ \sum_i^{m-1} |x_{i+1,n} - x_{i,n}| + \sum_i^{n-1} |x_{m,j+1} - x_{m,j}|, \end{aligned}$$

respectively.

The common drawback of the unconstrained problem (4.4) 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 (1.2) in place of the unconstrained problem (4.4), where the bounds are defined by the dynamic range of the images, see [8, 16, 38].

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

$$(4.6) \quad \text{PSNR} = 20 \log_{10} \left(\frac{\sqrt{mn}}{\|x - x_t\|_F} \right)$$

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

$$(4.7) \quad \text{ISNR} = 20 \log_{10} \left(\frac{\|y - x_t\|_F}{\|x - x_t\|_F} \right),$$

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 $[0, 1]$.

4.2.1. Experiment with l_2^2 isotropic total variation. We here consider the image restoration from a blurred/noisy observation using the model (1.2) equipped with the isotropic total variation regularizer. We employ OSGA, MFISTA (a monotone version of FISTA proposed by BECK & TEBoulLE in [8]), ADMM (an alternating direction method proposed by CHAN et al. in [16]), and a projected subgradient algorithms PSGA (with nonsummable diminishing step-size, see [12]). In our implementation we use the original code of MFISTA and ADMM provided by the authors, with minor adaptations to solve the problem form (1.2) and to stop in a fixed number of iterations.

We now 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 standard deviation set to $10^{-3/2}$. 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 $\lambda = 4 \times 10^{-3}$. All algorithms are stopped after 50 iterations. The results of our implementation are summarized in Table 3 and Figures 3 and 4.

Table 3: Result summary for the l_2^2 isotropic total variation

	λ	PSGA	MFISTA	ADMM	OSGA
PSNR		23.69	23.64	23.59	23.74
f_b	1×10^{-2}	1.6804e+2	1.6580e+2	1.6705e+2	1.6531e+2
Time		2.15	30.55	2.05	5.60
PSNR		23.73	23.66	23.67	23.76
f_b	7×10^{-3}	1.5694e+2	1.5543e+2	1.5599e+2	1.5500e+2
Time		2.09	31.31	2.12	5.39
PSNR		23.77	23.67	23.63	23.77
f_b	4×10^{-3}	1.4402e+2	1.4321e+2	1.4329e+2	1.4294e+2
Time		2.09	31.93	2.06	5.49

The results of Table 3 and Figure 3 show that function values, PSNR, and ISNR produced by the algorithms are sensitive to the regularization parameter λ . However,

the function values are less sensitive. Subfigures (a), (c), and (e) show that OSGA gives the best performance in terms of function values. According to subfigures (b), (d), and (f), the best ISNR is attained for $\lambda = 4 \times 10^{-3}$, and the algorithms are comparable with each other, but OSGA outperforms the others slightly. Figure 4 illustrates the resulting deblurred images for $\lambda = 4 \times 10^{-3}$.

4.2.2. Experiment with l_1 isotropic total variation. In this subsection we study the image restoration from a blurred/noisy observation using the model (1.3) equipped with the isotropic total variation regularizer. The optimization problem is solved by DRPD1, DRPD2 (Douglas-Rachford primal-dual algorithms proposed by BOŦ & HENDRICH in [10]), ADMM, and OSGA.

Here we consider recovering the 256×256 blurred/noisy Fingerprint image from a blurred/noisy image constructed by a 7×7 Gaussian kernel with standard deviation 5 and salt-and-pepper impulsive noise with the level 40%. The algorithms are stopped after 50 iterations. Three different regularization parameters $\lambda = 3 \times 10^{-1}$, $\lambda = 1 \times 10^{-1}$, and $\lambda = 8 \times 10^{-2}$ are considered. The results are presented in Table 4 and Figures 5 and 6.

The results of Figure 5 demonstrates that the algorithms are sensitive to the regularization parameter. For example, ADMM get the best function value and an acceptable ISNR compared with the others for $\lambda = 3 \times 10^{-1}$, however, for $\lambda = 1 \times 10^{-1}$ and $\lambda = 8 \times 10^{-2}$, it behaves worse than the others. In the sense of ISNR, DRPD1, DRPD2, and OSGA are competitive, but OSGA get the better results. The resulting images for $\lambda = 8 \times 10^{-2}$ are illustrated in Figure 6, demonstrating that ADMM could not recover the image properly, however, DRPD1, DRPD2, and OSGA reconstruct acceptable approximations, where OSGA attains the best PSNR.

Table 4: Results for the l_1 isotropic total variation

	λ	DRPD-1	DRPD-2	ADMM	OSGA
PSNR		17.95	17.61	18.47	18.67
f_b	3×10^{-1}	1.4581e+4	1.4867e+4	1.4476e+4	1.4564e+4
Time		1.04	0.67	0.81	6.11
PSNR		21.87	21.23	20.10	22.05
f_b	1×10^{-1}	1.3571e+4	1.3635e+4	1.3799e+4	1.3612e+4
Time		1.28	0.83	0.76	6.01
PSNR		22.26	21.56	18.67	22.46
f_b	8×10^{-2}	1.3519e+4	1.3573e+4	1.3879e+4	1.3582e+4
Time		1.00	0.65	0.79	6.11

5. Concluding remarks. This paper addresses an optimal subgradient algorithm, OSGA, for solving bound-constrained convex optimization. It is shown that the solution of OSGA's subproblem has a piecewise linear form in a single dimension. Afterwards, we give two iterative schemes to solve this one-dimensional problem, where the first one solves OSGA's subproblem exactly and the second one inexactly. In particular, the first scheme uses the piecewise linear solution to translate the subproblem into a one-dimensional piecewise rational problem. Subsequently, the optimizer of the

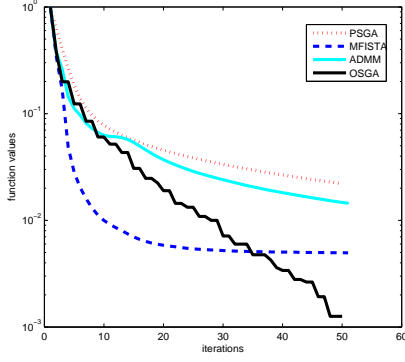
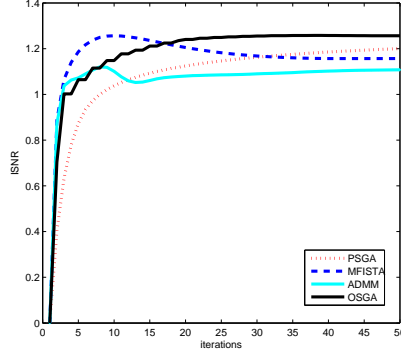
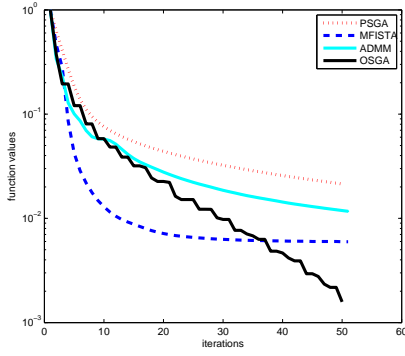
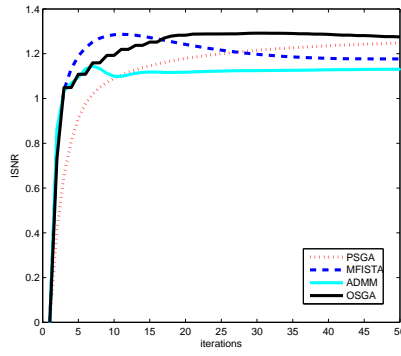
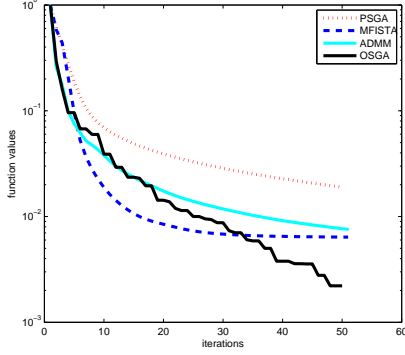
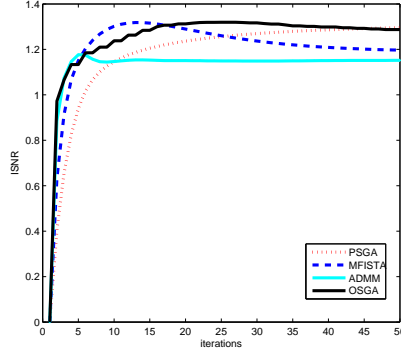
(a) δ_k versus iterations, $\lambda = 1 \times 10^{-3}$ (b) ISNR versus iterations, $\lambda = 1 \times 10^{-3}$ (c) δ_k versus iterations, $\lambda = 6 \times 10^{-4}$ (d) ISNR versus iterations, $\lambda = 6 \times 10^{-4}$ (e) δ_k versus iterations, $\lambda = 4 \times 10^{-4}$ (f) ISNR versus iterations, $\lambda = 4 \times 10^{-4}$

Fig. 3: 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 deviation $10^{-3/2}$. The algorithms were stopped after 50 iterations. The subfigures (a), (c) and, (e) display the relative error δ_k (4.2) of function values versus iterations, and (b), (d), and (f) show ISNR (4.7) versus iterations.



(a) Original image



(b) Blurred/noisy image

(c) PSGA: $f = 1.4402e+2$, PSNR = 23.77, T = 2.09(d) MFISTA: $f = 1.4321e+2$, PSNR = 23.67, T = 31.93(e) ADMM: $f = 1.4329e+2$, PSNR = 23.63, T = 2.06(f) OSGA: $f = 1.4294e+2$, PSNR = 23.77, T = 5.49

Fig. 4: Deblurring of the 512×512 Barbara image with the 9×9 uniform blur and the Gaussian noise with deviation $10^{-3/2}$ by PSGA, MFISTA, ADMM, and OSGA with the regularization parameter $\lambda = 4 \times 10^{-3}$. The algorithms were stopped after 50 iterations.

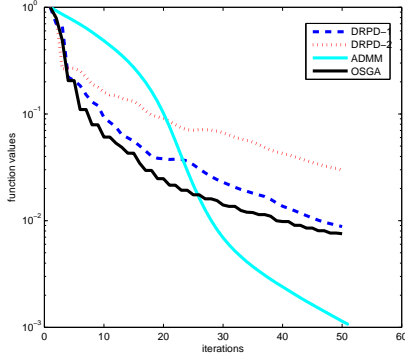
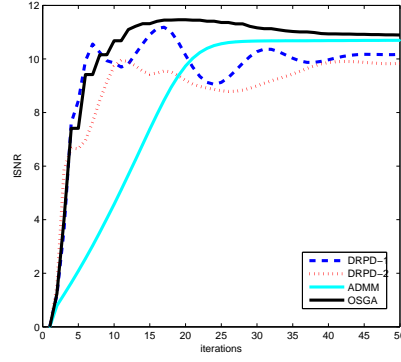
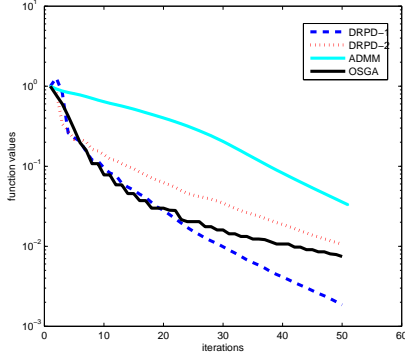
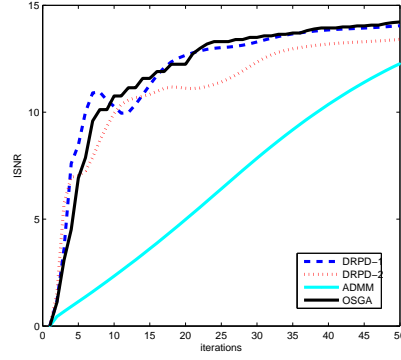
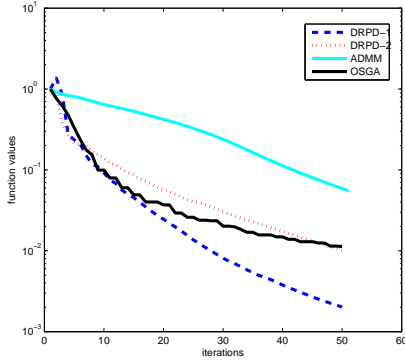
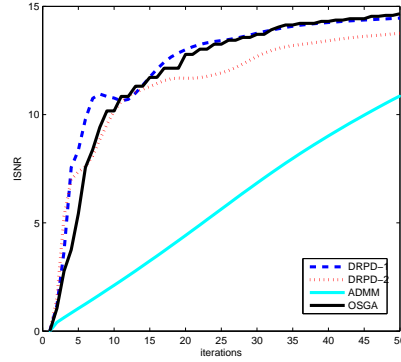
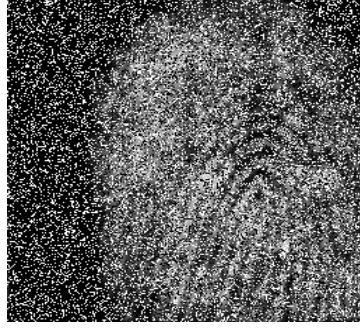
(a) δ_k versus iterations, $\lambda = 2 \times 10^{-1}$ (b) ISNR versus iterations, $\lambda = 2 \times 10^{-1}$ (c) δ_k versus iterations, $\lambda = 8 \times 10^{-2}$ (d) ISNR versus iterations, $\lambda = 8 \times 10^{-2}$ (e) δ_k versus iterations, $\lambda = 6 \times 10^{-2}$ (f) ISNR versus iterations, $\lambda = 6 \times 10^{-2}$

Fig. 5: A comparison among DRPD1, DRPD2, ADMM, and OSGA for deblurring the 256×256 Fingerprint image with the various regularization parameter λ . The blurred/noisy image was constructed by the 7×7 Gaussian kernel with standard deviation 5 and salt-and-pepper impulsive noise with the level 50%. The algorithms were stopped after 50 iterations. Subfigures (a), (c), and (e) display the relative error δ_k (4.2) of function values versus iterations, and (b), (d), and (f) show ISNR (4.7) versus iterations.



(a) Original image



(b) Blurred/noisy image

(c) DRPD-1: $f = 1.3571e + 4$, PSNR = 21.87, T = 1.28(d) DRPD-2: $f = 1.3635e + 4$, PSNR = 21.23, T = 0.83(e) ADMM: $f = 1.3799e + 4$, PSNR = 20.10, T = 0.76(f) OSGA: $f = 1.3612e + 4$, PSNR = 22.04, T = 6.01

Fig. 6: Deblurring of the 256×256 Fingerprint image using DRPD1, DRPD2, ADMM, and OSGA with the regularization parameter $\lambda = 7 \times 10^{-3}$. The algorithms were stopped after 50 iterations. The blurred/noisy image was constructed by the 7×7 Gaussian kernel with standard deviation 5 and salt-and-pepper impulsive noise with the level 50%.

subproblem is founded in each interval and compared to give the global solution. The second scheme substitutes the piecewise linear solution into the subproblem and give a one-dimensional nonlinear equation that can be solved with zero finders to give the optimizer. Numerical results are reported showing the efficiency of OSGA compared with some state-of-the-art algorithms.

Acknowledgement. We would like to thank RADU BOT and MIN TAO for making their codes DRPD-1, DRPD-2, and ADMM available for us.

REFERENCES

- [1] M. AHOOKHOSH, *Optimal subgradient algorithms with application to large-scale linear inverse problems*, submitted (2014), <http://arxiv.org/abs/1402.7291>.
- [2] M. AHOOKHOSH, *User's manual for OSGA (Optimal SubGradient Algorithm)*, http://homepage.univie.ac.at/masoud.ahookhosh/uploads/User's_manual_for_OSGA.pdf, (2014).
- [3] M. AHOOKHOSH AND A. NEUMAIER, *An optimal subgradient algorithm for large-scale convex optimization in simple domains*, submitted (2015), <http://arxiv.org/abs/1501.01451>.
- [4] M. AHOOKHOSH AND A. NEUMAIER, *Solving nonsmooth convex optimization with complexity $O(\epsilon^{-1/2})$* , Manuscript, University of Vienna, (2015).
- [5] M. AHOOKHOSH AND A. NEUMAIER, *An optimal subgradient algorithm with subspace search for costly convex optimization problems*, Manuscript, University of Vienna, (2015).
- [6] M. AHOOKHOSH AND A. NEUMAIER, *High-dimensional convex optimization via optimal affine subgradient algorithms*, in ROKS workshop, 83-84 (2013)
- [7] J. BARDSLEY AND C.R. VOGEL, *A nonnegatively constrained convex programming method for image reconstruction*, SIAM Journal on Scientific Computing, 25 (2003), pp. 1326–1343.
- [8] A. BECK, M. TEOULLE, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring*, IEEE Transactions on Image Processing 18(11) (2009), 2419–2434.
- [9] E.G. BIRGIN, J.M. MARTINEZ, AND M. RAYDAN, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM Journal on Optimization, 10 (2000), pp. 1196–1211.
- [10] R.I. BOT AND C. HENDRICH, *A Douglas-Rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators*, SIAM Journal on Optimization, 23(4) (2013), pp. 2541–2565.
- [11] R.I. BOT, E.R. CSETNEK, AND C. HENDRICH, *A primal-dual splitting algorithm for finding zeros of sums of maximally monotone operators*, SIAM Journal on Optimization, 23 (2013), pp. 2011–2036.
- [12] S. BOYD, L. XIAO, AND A. MUTAPCIC, *Subgradient methods*, http://www.stanford.edu/class/ee392o/subgrad_method.pdf, (2003).
- [13] M.A. BRANCH, T. F. COLEMAN, AND Y. LI, *A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 1–23.
- [14] R.H. BYRD, P. LU, J. NOCEDAL, AND C. ZHU, *A limited memory algorithm for bound constrained optimization*, SIAM Journal on Scientific Computing, 16 (1995), pp. 1190–1208.
- [15] A. CHAMBOLLE, V. CASELLES, D. CREMERS, M. NOVAGA, T. POCK, *An introduction to total variation for image analysis* In: Theoretical Foundations and Numerical Methods for Sparse Recovery, vol. 9, pp. 263340. De Gruyter, Radon Series Comp. Appl. Math. (2010)
- [16] R.H. CHAN, M. TAO, AND X. YUAN, *Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers*, SIAM Journal on Imaging Science, 6(1) (2013), pp. 680–697.
- [17] Y.-H. DAI AND R. FLETCHER, *New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds*, Mathematical Programming, 106 (2006), pp. 403–421.
- [18] T. ELFVING, P. C. HANSEN, AND T. NIKAZAD, *Semiconvergence and relaxation parameters for projected SIRT algorithms*, SIAM Journal on Scientific Computing, 34(4) (2012), pp. A2000–A2017.
- [19] E. ESSER, Y. LOU, AND J. XIN, *A method for finding structured sparse solutions to nonnegative least squares problems with applications*, SIAM Journal on Imaging Science, 6(4) (2013), pp. 2010–2046.
- [20] N. KARMITSA AND M.M. MÄKELÄ, *Limited memory bundle method for large bound constrained nonsmooth optimization: convergence analysis*, Optimization Methods and Software, 25(6)

- (2010), pp. 895–916.
- [21] N. KARMITSA AND M.M. MÄKELÄ, *Adaptive limited memory bundle method for bound constrained large-scale nonsmooth optimization*, Optimization, 59(6) (2010), pp. 945–962.
 - [22] L. KAUFMAN, A. NEUMAIER, *PET regularization by envelope guided conjugate gradients*, IEEE Transactions on Medical Imaging, 15 (1996), pp. 385–389.
 - [23] L. KAUFMAN, A. NEUMAIER, *Regularization of ill-posed problems by envelope guided conjugate gradients*, Journal of Computational and Graphical Statistics, 6(4) (1997), pp. 451–463.
 - [24] D. KIM, S. SRA, AND I. S. DHILLON, *Tackling box-constrained optimization via a new projected quasi-Newton approach*, SIAM Journal on Scientific Computing, 32 (2010), pp. 3548–3563.
 - [25] Hager, W.W., Zhang, H.: A new active set algorithm for box constrained optimization, SIAM Journal on Optimization 17, pp. 526–557 (2006)
 - [26] W.W. HAGER AND H. ZHANG, *The limited memory conjugate gradient method*, SIAM Journal on Optimization, 23 (2013), pp. 2150–2168.
 - [27] R. HELGASON, J. KENNINGTON, H. LALL, *A polynomially bound algorithms for a singly constrained quadratic program*, Mathematical Programming, 18 (1980), pp. 338–343.
 - [28] C.J. LIN, J. J. MORÉ, *Newton’s method for large bound-constrained optimization problems*, SIAM Journal on Optimization, 9 (1999), pp. 1100–1127.
 - [29] S. MORINI, M. PORCELLI, AND R.H. CHAN, *A reduced Newton method for constrained linear least squares problems*, Journal of Computational and Applied Mathematics, 233 (2010), pp. 2200–2212.
 - [30] A.S. Nemirovsky and D.B. Yudin, *Problem complexity and method efficiency in optimization*, Wiley, New York (1983).
 - [31] Y. NESTEROV, *Introductory lectures on convex optimization: A basic course*, Kluwer, Dordrecht (2004).
 - [32] Y. NESTEROV, *Smooth minimization of non-smooth functions*, Mathematical Programming, 103 (2005), 127–152.
 - [33] A. NEUMAIER, *OSGA: a fast subgradient algorithm with optimal complexity*, submitted, <http://arxiv.org/abs/1402.1125>, (2014).
 - [34] A. NEUMAIER, *Solving ill-conditioned and singular linear systems: A tutorial on regularization*, SIAM Review, 40(3) (1998), pp. 636–666.
 - [35] A. NEUMAIER, *Introduction to Numerical Analysis*, Cambridge University Press, Cambridge, (2001).
 - [36] A. NEUMAIER AND B. AZMI, *LMBOPT: a limited memory method for bound-constrained optimization*, Manuscript, University of Vienna, (2015).
 - [37] P.M. PARDALOS, N. KOVOOR, *An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds*, Mathematical Programming, 46 (1990), 321–328.
 - [38] H. WOO AND S. YUN, *Proximal linearized alternating direction method for multiplicative denoising*, SIAM Journal on Scientific Computing, 35 (2013), pp. 336–358.
 - [39] J. ZHANG AND B. MORINI, *Solving regularized linear least-squares problems by the alternating direction method with applications to image restoration*, Electronic Transactions on Numerical Analysis, 40 (2013), pp. 356–372.
 - [40] X. ZHANG, A. SAHA, AND S.V.N. VISHWANATHAN, *Lower bounds on rate of convergence of cutting plane methods*, Manuscript, Department of Computing Sciences, University of Alberta, (2013).