AN OPTIMAL SUBGRADIENT ALGORITHM FOR LARGE-SCALE CONVEX OPTIMIZATION IN SIMPLE DOMAINS

MASOUD AHOOKHOSH∗ AND ARNOLD NEUMAIER†

Abstract. This paper shows that the optimal subgradient algorithm, OSGA, proposed in [59] can be used for solving structured large-scale convex constrained optimization problems. Only first-order information is required, and the optimal complexity bounds for both smooth and nonsmooth problems are attained. More specifically, we consider two classes of problems: (i) a convex objective with a simple closed convex domain, where the orthogonal projection on this feasible domain is efficiently available; (ii) a convex objective with a simple convex functional constraint. If we equip OSGA with an appropriate prox-function, the OSGA subproblem can be solved either in a closed form or by a simple iterative scheme, which is especially important for large-scale problems. We report numerical results for some applications to show the efficiency of the proposed scheme. A software package implementing OSGA for above domains is available.

Key words. structured convex optimization, sparse optimization, nonsmooth optimization, projection operator, optimal complexity, first-order black-box information, high-dimensional data

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

1. Introduction. Convex optimization has been shown to provide efficient algorithms for computing reliable solutions in a broad range of applications. Many applications arising in applied sciences and engineering such as signal and image processing, machine learning, statistics, and general inverse problems can be addressed by a convex optimization problem involving high-dimensional data. In practice, solving a nonsmooth convex problem is usually more difficult and costly than a smooth one. More precisely, for a prescribed accuracy parameter ε, the optimal complexity to achieve an ε-solution of nonsmooth Lipschitz continuous problems is $O(\varepsilon^{-2})$, the superior complexity $O(\varepsilon^{-1/2})$ for smooth problems with Lipschitz continuous gradient, see [52, 53].

Thanks to the low memory requirement and simple structure, first-order methods have received much attention during the past few decades. Indeed, they deal successfully with large-scale problems. In general, convex optimization problems can be solved by gradient-type algorithms [3, 21, 22, 38], conjugate gradient methods [41, 45, 46] and spectral gradient methods [12, 23, 63] for smooth objectives and by subgradient-type methods [27, 51, 57], proximal gradient methods [62, 32], smoothing techniques [15, 24, 34, 55], bundle-type algorithms [48, 49], and primal-dual first-order methods [25, 26, 28] for nonsmooth objectives. Moreover, both classes can be addressed by (zero-order) coordinate descent methods and derivative-free methods. The current paper only addresses first-order methods and assumes that first-order black-box information – function values and subgradients – of the objective function are available.

Historically, gradient descent and subgradient methods were the first numerical schemes proposed to solve optimization problems with smooth and nonsmooth convex objective functions, respectively. In practice, they are too slow, especially for badly scaled problems. This can be addressed by their worst-case complexity bounds to reach an ε-solution, while the gradient descent method achieve the complexity of the

∗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
order $O(\varepsilon^{-1})$ which is not optimal for smooth problems, the subgradient methods attain the worst-case complexity of the order $O(\varepsilon^{-2})$. In 1983, Nemirovski & Yudin in [52] derived optimal worst-case complexity bounds of first-order methods to achieve an $\varepsilon$-solution for several class of problems such as Lipschitz continuous nonsmooth problems and smooth problems with Lipschitz continuous gradient. If an algorithm attains the optimal worst-case complexity bound for a class of problems, it is called optimal. Optimal first-order methods dating back to Nesterov [54] in 1983. This optimal first-order method is interesting both theoretically and computationally, attracting many researchers to work in the development of such schemes, for example Auslander & Teboulle [9], Beck & Teboulle [16], Devolder et al. [33], Gonzaga et al. [39, 40], Lan [49], Lan et al. [50], Nesterov [55, 56, 58], Neumaier [59] and Tseng [65]. Computational comparisons for composite functions show that optimal Nesterov-type first-order methods are substantially superior to the gradient descent and subgradient methods, see, for example, Ahookhosh [1] and Becker et al. [18].

Content. In this paper we consider structured convex constrained optimization problems frequently observed in applications and develop OSGA to efficiently solve such problems. Two classes of convex domains are considered, namely, simple convex domains such that the orthogonal projection is cheaply feasible, and sublevel set of a convex function referred as functional domain. For problems with a simple domain, we first introduce an appropriate prox-function and then show that the solution of OSGA’s subproblem is obtained by a projection on the domain followed by solving a one-dimensional nonlinear equation. It is shown that if explicit formula for projection is available, the nonlinear equation can be solved in a closed form in many interesting cases. We also establish the optimality condition for functional domain and show for some simple functions that results to in a closed form solution. Finally, we report some numerical results for applications to show the efficiency OSGA in comparison with some state-of-the-art algorithms.

The remainder of this paper is organized as follows. In the next section, we review the basic idea of OSGA. Section 3 considers the structured convex constrained optimization and how to solve the associated OSGA subproblem. We report numerical results in Section 4 and our conclusions are derived in Section 5.

Notation and preliminaries. Let $V$ be a real finite-dimensional vector space endowed with the norm $\| \cdot \|$, and $V^*$ denotes the dual space of all linear functional on $V$ where the bilinear pairing $\langle g, x \rangle$ denotes the value of the functional $g \in V^*$ at $x \in V$. If $V = \mathbb{R}^n$, then

$$\|x\|_2 := \left( \sum_{i=1}^{n} |x_i|^2 \right)^{1/2}.$$ 

If $x \in \mathbb{R}^{m \times n}$, then the Schatten $\infty$-norm is $\|\sigma(x)\|_{\infty}$ where $\sigma : \mathbb{R}^{m \times n} \to \mathbb{R}^{\min\{m,n\}}$ is the function that takes a matrix $x \in \mathbb{R}^{m \times n}$ and returns a vector of singular values in nonincreasing order. If $x$ is a positive definite matrix, we denote it by $x \succ 0$. We also denote by $x = \sum_{i=1}^{n} \lambda_i u_i u_i^T$ and $x = \sum_{i=1}^{n} \sigma_i u_i v_i^T$ the eigenvalue decomposition and the singular value decomposition of $x$. For a function $f : V \to \overline{\mathbb{R}} = \mathbb{R} \cup \{\pm \infty\}$, we denote by

$$\text{dom}f := \{x \in V \mid f(x) < +\infty\}.$$
its **effective domain** and call \( f \) **proper** if \( \text{dom} f \neq \emptyset \) and \( f(x) > -\infty \) for all \( x \in V \). The vector \( g \in V^* \) is called a **subgradient** of \( f \) at \( x \) if \( f(x) \in \mathbb{R} \) and
\[
f(y) \geq f(x) + \langle y, y - x \rangle \quad \text{for all} \quad y \in V.
\]
The set \( \partial f(x) \) of all subgradients is called the **subdifferential** of \( f \) at \( x \).

We call a nonempty, closed, and convex subset \( C \) of \( V \) a **simple convex domain** if the orthogonal projection
\[
P_C(y) := \text{argmin}_{x \in C} \frac{1}{2} \|x - y\|^2
\]
of \( y \) to \( C \) can be found efficiently for every \( y \in V \). Note that \( P_C(y) \) is unique since \( \frac{1}{2} \|x - y\|^2 \) is strongly convex. Computing the orthogonal projection is a well-studied topic on convex optimization, and the projection operator is available for many domains \( C \) either in a closed form or by a simple iterative scheme. Table 1 gives some practically interesting convex domains, associated projection operators, and references for the formulas or iterative schemes.

Table 1: List of some available projection operators for \( C = \{x \in V \mid c(x)\} \)

<table>
<thead>
<tr>
<th>Defining constraint ( c(x) )</th>
<th>Projection operator</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Ax = b )</td>
<td>( u = y - A^T(Ay - b) )</td>
<td>[62]</td>
</tr>
<tr>
<td>( (a, x) = b )</td>
<td>( u = y - (a, y) - b/|a|^2 ) a</td>
<td>[13]</td>
</tr>
<tr>
<td>( (a, x) \leq b )</td>
<td>( u = y - (a, y) - b/|a|^2 ) a</td>
<td>[13]</td>
</tr>
<tr>
<td>(</td>
<td>(a, x)</td>
<td>\leq b )</td>
</tr>
<tr>
<td>( \underline{b} \leq Ax \leq \overline{b} )</td>
<td>( u = x - \sum_{i=1}^N \lambda_i(x)/|A_i|^2 ) ( A_i ), ( \lambda_i(x) := \begin{cases} 0 &amp; \text{if } \underline{b}_i \leq (A_i, x) \leq \overline{b}_i, \ (A_i, x) - \overline{b}_i &amp; \text{if } (A_i, x) &gt; \overline{b}_i, \ (A_i, x) - \underline{b}_i &amp; \text{if } (A_i, x) &lt; \underline{b}_i. \end{cases} )</td>
<td>[13]</td>
</tr>
<tr>
<td>( x \in [a, \overline{a}] )</td>
<td>( u = \sup { x, \inf { y, \overline{a} } } )</td>
<td>[13]</td>
</tr>
<tr>
<td>( x \geq 0 )</td>
<td>( u = (y)_+ := \max(y, 0) )</td>
<td>[62]</td>
</tr>
<tr>
<td>( |x|_1 \leq \xi )</td>
<td>iterative scheme</td>
<td>[36, 62]</td>
</tr>
<tr>
<td>( |x|_2 \leq \xi )</td>
<td>( u = \begin{cases} \xi y/|y|_2 &amp; \text{if } |y|_2 &gt; \xi \ y &amp; \text{if } |y|_2 \leq \xi \end{cases} )</td>
<td>[13]</td>
</tr>
<tr>
<td>( |x|_\infty \leq \xi )</td>
<td>( u = \sup { -\xi t, \inf { y, \xi t } } )</td>
<td>[62]</td>
</tr>
<tr>
<td>( {x, t \mid |x|_2 \leq t } )</td>
<td>( u = \begin{cases} 0 &amp; \text{if } |y|_2 \leq -t \ (y, t) &amp; \text{if } |y|_2 \leq t \ 1/2(1 + |y|_2)(y, |y|_2) &amp; \text{if } |y|_2 \geq</td>
<td>t</td>
</tr>
</tbody>
</table>

Exponential cone iterative scheme [62]

Epigraphs iterative scheme [13]

Sublevel sets iterative scheme [13]

Simplex iterative scheme [62]

\( x \geq 0, x = \sum_{i=1}^n \lambda_i u_i u_i^T \) \( u = \sum_{i=1}^n (\lambda_i)_{+} u_i u_i^T \) [62]

\( x \geq 0, \text{tr}(x) = 1 \) iterative scheme [62]

\( \|\sigma(x)\|_\infty \leq 1, x = \sum_{i=1}^n \sigma_i u_i u_i^T \) \( u = \sum_{i=1}^n \max(\lambda_i, 1) u_i u_i^T \) [62]
2. A review of OSGA. In what follows we briefly review the main idea of optimal subgradient algorithm proposed by Neumaier in [59]. To this end, we first consider the convex constrained minimization problem

\begin{equation}
\min_{x \in C} f(x),
\end{equation}

where $f : C \to \mathbb{R}$ is a convex function defined on a nonempty, closed and convex subset $C$ of $\mathcal{V}$. The main objective is to find a solution $u \in C$ by using the first-order information, i.e., function values and subgradients.

OSGA (see Algorithm 1) is an optimal subgradient algorithm for problem (2.1) that constructs a sequence of iterations whose related function values converge to the minimum with the optimal complexity. Moreover, OSGA requires no information regarding global parameters such as Lipschitz constants of function values and gradients. The primary objective is to monotonically reduce bounds on the error $f(x_b) - \hat{f}$ of function values, where $\hat{f}$ is the minimum and $x_b$ is the best known point.

OSGA considers the linear relaxations

\begin{equation}
f(z) \geq \gamma + \langle h, z \rangle \quad \text{for all } z \in C,
\end{equation}

of $f$ at $z$, where $\gamma \in \mathbb{R}$ and $h \in \mathcal{V}^*$, and a continuously differentiable prox-function $Q : C \to \mathbb{R}$ satisfying

\begin{equation}
Q_0 := \inf_{z \in C} Q(z) > 0
\end{equation}

and

\begin{equation}
Q(z) \geq Q(x) + \langle g_Q(x), z - x \rangle + \frac{\sigma}{2} \|z - x\|^2 \quad \text{for all } x, z \in C,
\end{equation}

where $\sigma = 1$, $g_Q(x)$ denotes the gradient of $Q$ at $x \in C$ and $\|\cdot\|$ is a norm defined on $\mathcal{V}$. OSGA solves a sequence of minimization problems of the form

\begin{equation}
\sup_{x \in C} E_{\gamma, h}(x)
\end{equation}

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

\begin{equation}
E_{\gamma, h}(x) := \frac{-\gamma + \langle h, x \rangle}{Q(x)}.
\end{equation}

If $u = U(\gamma, h) \in C$ is the solution of this problem, then it is assumed that $e = E(\gamma, h)$ and $u = U(\gamma, h)$ are readily computable.

In [59], it is shown that OSGA attains the following bound on function values

\begin{equation}
0 \leq f(x_b) - \hat{f} \leq \eta Q(\bar{x}).
\end{equation}

Hence, by decreasing the error factor $\eta$, the convergence to an $\varepsilon$-minimizer $x_b$ is guaranteed by

\begin{equation}
0 \leq f(x_b) - \hat{f} \leq \varepsilon.
\end{equation}
for the accuracy tolerance $\varepsilon > 0$. In [59], it is shown that the number of iterations to achieve the optimizer is in the order $O\left(\varepsilon^{-1/2}\right)$ for smooth $f$ with Lipschitz continuous gradients and in the order $O\left(\varepsilon^{-2}\right)$ for Lipschitz continuous nonsmooth $f$, which is optimal in both cases, cf. Nemirovsky & Yudin [52] and Nesterov [53]. The algorithm does not need to know about the global Lipschitz parameters and has the low memory requirement. Hence if the subproblem (2.5) can be solved efficiently, OSGA is appropriate for solving large-scale problems. Numerical results reported by Ahookhosh in [1] and Ahookhosh & Neumaier in [4, 5], for unconstrained problems, and Ahookhosh & Neumaier in [6, 7], for constrained problems, show the promising behavior of OSGA for practical problems. In the next section we show that by selecting a suitable prox-function, OSGA’s subproblem (2.5) can be solved efficiently for structured convex constrained problems.

Algorithm 1: OSGA (optimal subgradient algorithm)

**Input:** $\delta, \alpha_{\text{max}} \in ]0, 1[, \kappa’ \leq \kappa$; 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  
$\alpha \leftarrow \alpha_{\text{max}}$;  
while stopping criteria do not hold do  
$x = x_b + \alpha(u - x_b); \text{compute } f_x \text{ and } g_x; \quad g = g - \mu g_Q(x_b); \quad \overline{h} = h + \alpha(g - h); \quad \overline{\gamma} = \gamma + \alpha(f_x - \mu Q(x_b) - \langle g, x_b \rangle - \gamma); \quad x’_b = \text{argmin}\{x \in \{x_b, x\} f(z, v_z); \quad f_{x’_b} = \min\{f_{x_b}, f_x\}; \quad \gamma’_b = \overline{\gamma} - f_{x’_b}; u’ = U(\gamma’_b, \overline{h}); \quad x’ = x_b + \alpha(u’ - x_b); \text{compute } f_{x’}; \quad \text{choose } \overline{x}_b \text{ in such a way that } f_{\overline{x}_b} \leq \min\{f’_{x_b}, f_{x’}\}; \quad \overline{\gamma}_b = \overline{\gamma} - f_{x_b}; \quad \overline{\mu} = U(\gamma_b, \overline{h}); \quad \overline{\eta} = E(\gamma_b, \overline{h}) - \mu; \quad x_b = \overline{x}_b; \quad f_{x_b} = f_{x_b};$  
if $f_{x_b} \leq f_{\text{target}}$ then  
stop;  
else  
update the parameters $\alpha, h, \gamma, \eta$ and $u$ using UPS;  
end  
end

As discussed in [59], OSGA uses the following scheme for updating the given parameters $\alpha, h, \gamma, \eta$ and $u$:
Algorithm 2: PUS (parameters updating scheme)

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

**Output:** $\alpha$, $h$, $\gamma$, $\eta$, $u$;

**begin**

$R \leftarrow (\eta - \bar{\eta})/(\delta \alpha \eta)$;

if $R < 1$ then

$h \leftarrow \bar{h}$;

else

$\alpha \leftarrow \min(\alpha e^{\kappa'(R-1)}, \alpha_{\text{max}})$;

end

$\eta \leftarrow \min(\bar{\eta}, \alpha)$;

if $\bar{\eta} < \eta$ then

$h \leftarrow \bar{h}$;

$\gamma \leftarrow \bar{\gamma}$;

$\eta \leftarrow \eta$;

$u \leftarrow \bar{u}$;

end

**end**

3. Structured convex constrained problems in simple domains. In this paper we consider the convex constrained optimization problem

\[
\min_{x \in C} f(Ax)
\]

where $f : C \to \mathbb{R}$ is convex and lower semicontinuous, $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator, and $C$ is a simple convex domain. We call problem (3.1) a simple domain problem. This problem appears in many applications such as signal and image processing, machine learning, statistics, and inverse problem.

**Example 3.1. (Image restoration)** The process of reconstructing or estimating a true image from a degraded observation is known as the image restoration, also called deblurring or deconvolution. Image restoration is addressed by solving a constraint satisfaction problem of the form

\[
Ax = b, \quad x \in C,
\]

where $C$ a convex domain $C$ that is commonly a box or the nonnegativity constraint. This is an ill-posed problem, see Neumaier [60], and normally handled by the regularized least-squares problem

\[
\min_{x \in C} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \varphi(x)
\]

or the regularized $l_1$ problem

\[
\min_{x \in C} \|Ax - b\|_1 + \lambda \varphi(x)
\]

where $\varphi : C \to \mathbb{R}$ is a convex regularization function such as $\| \cdot \|_2^2$, $\| \cdot \|_1$, $\| \cdot \|_{\text{ITV}}$, and $\| \cdot \|_{\text{ATV}}$. The regularizers $\| \cdot \|_{\text{ITV}}$ and $\| \cdot \|_{\text{ATV}}$ are respectively called isotropic and anisotropic total variation, see, for example, [29], where they are defined by

\[
\|x\|_{\text{ITV}} = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} \sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2} + \sum_{i=1}^{m-1} |x_{i+1,n} - x_{i,n}| + \sum_{i=1}^{m} |x_{m,j+1} - x_{m,j}|
\]
and
\[
\|x\|_{AV} = \sum_{i}^{m-1} \sum_{j}^{n-1} \left\{ |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}| \right\} \\
+ \sum_{i}^{m-1} |x_{i+1,n} - x_{i,n}| + \sum_{i}^{n-1} |x_{m,j+1} - x_{m,j}|,
\]
for \( x \in \mathbb{R}^{m \times n} \).

**Example 3.2. (Basis pursuit problem)** Let \( A : \mathbb{R}^{n} \to \mathbb{R}^{m} \) be a linear operator with \( m < n \) and \( y \in \mathbb{R}^{m} \). The basis pursuit problem is the constrained minimization problem
\[
(3.4) \quad \min \|x\|_{1} \quad \text{s.t.} \quad Ax = y,
\]
which determines an \( l_1 \)-minimal solution \( \hat{x} \) of the undetermined linear system \( Ax = y \). This problem appears in many applications such signal and image processing and compressed sensing, see [19, 20, 31, 35, 67, 68, 69] and references therein.

According to the features of objective functions, (3.2) can be solved by Nesterov-type optimal methods, however, (3.3) and (3.4) cannot be solved by Nesterov-type optimal methods. Since OSGA only needs first-order information, it can deal with all of these problems without considering the structure of problems. In the remainder of this section, we establish how OSGA can be used to efficiently solve the problem (3.1). Since the underlying problem (3.1) is a special case of the problem (2.1) considered in [59], the complexity of OSGA remains valid for both smooth and nonsmooth problems.

The quadratic function
\[
(3.5) \quad Q(z) := \frac{1}{2} \|z\|^{2} + Q_{0},
\]
is a prox-function, see e.g. [1]. We now show that the solution of OSGA’s subproblem (2.5) can be found either in a closed form or by a simple iterative scheme. In particular, we address some convex domains that a closed form solution for associated OSGA’s subproblem (2.5) can be found.

The next result shows that the solution of the auxiliary subproblem (2.5) is given by the orthogonal projection (1.1) of \( y := e^{-1}h \) on the domain \( C \) followed by solving a one-dimensional nonlinear equation to determine \( e \).

**Theorem. 3.1.** Let \( u \) be a minimizer of (2.5) and also let \( e = E_{\gamma,h}(u) > 0 \). Then
\[
u = \hat{u}(e) := P_{C}(y), \quad y := -e^{-1}h;
\]
where, \( e \) is a solution of the univariate equation
\[
\varphi(e) = 0
\]
with
\[
(3.6) \quad \varphi(e) := e \left( \frac{1}{2} \|\hat{u}(e)\|^{2} + Q_{0} \right) + \gamma + \langle h, \hat{u}(e) \rangle.
\]

**Proof.** From Proposition 5.1 in [59], at the minimizer \( u \), we obtain
\[
(3.7) \quad eQ(u) = -\gamma - \langle h, u \rangle
\]
and

\[(3.8) \quad \langle eu + h, z - u \rangle \geq 0 \quad \text{for all } z \in C.\]

By setting $z = u$ in this variational inequality, it follows that $u$ is a solution of the minimization problem

\[
\inf_{z \in C} \langle eu + h, z - u \rangle.
\]

The first-order optimality condition for this problem is

\[(3.9) \quad 0 \in eu + h + N_C(u),\]

where

\[N_C(u) := \{ p \in V | \forall y \in C, \langle p, u - y \rangle \geq 0 \}\]

denotes the normal cone to $C$ at $u$. Since $e > 0$, $u$ satisfies

\[
u = \arg\min_{z \in C} \frac{1}{2} \|ez + h\|^2 = \arg\min_{z \in C} \frac{1}{2} \|z - y\|^2 = P_C(y) = \hat{u}(e),
\]

where $y = -e^{-1}h$ giving the result. \[\square\]

Theorem 3.1 gives a way to compute a solution of OSGA’s subproblem (2.5) involving a projection on the domain $C$ and solving the one-dimensional nonlinear equation. This equation can be solved exactly for some projection operators, see Table 2. However, one can solve this nonlinear equation approximately using zero finding schemes, see e.g. Chapter 5 of [61]. We apply the results of Theorem 3.1 in the next scheme to solve OSGA’s subproblem (2.5):

**Algorithm 3: OSS (OSGA’s subproblem solver)**

**Input**: $Q_0$, $\gamma$, $h$, a program for evaluating $\varphi(e)$ defined in (3.6);

**Output**: $u$, $e$;

**begin**

solve the nonlinear equation $\varphi(e) = 0$ either in a closed form or approximately by a root finding solver;

set $u = \hat{u}(e)$.

**end**

To implement Algorithm 3 (OSS), we first need to solve the projection problem (1.1) effectively, see Table 1. If one solves the equation $\varphi(e) = 0$ approximately, and an initial interval $[a, b]$ is available such that $\varphi(a)\varphi(b) < 0$, then a solution can be computed to $\varepsilon$-accuracy using the bisection scheme in $O(\log_2((b - a)/\varepsilon))$ iterations, see, for example, [61]. However, it is preferable to use a more sophisticated zero finder like the secant bisection scheme (Algorithm 5.2.6, [61]). If an interval $[a, b]$ with sign change is available\(^1\), one can also use MATLAB’s `fzero` function combining the bisection scheme, the inverse quadratic interpolation, and the secant method.

In the following we investigate special domains $C$, where the nonlinear equation $\varphi(e) = 0$ can be solved explicitly, see Table 2.

\(^1\) Without a sign change, `fzero` is unreliable; it fails on the simple quadratic $x^2 - 0.0001 = 0$ with starting point 0.2.
Table 2: List of domains $C$ where $\varphi(e) = 0$ can be solved explicitly

<table>
<thead>
<tr>
<th>defining constraint $c(x)$</th>
<th>solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Ax = b$</td>
<td>Proposition 3.3</td>
</tr>
<tr>
<td>$(x, x) = b$</td>
<td>Corollary 3.4</td>
</tr>
<tr>
<td>$(x, x) \leq b$</td>
<td>Proposition 3.5</td>
</tr>
<tr>
<td>$x \geq 0$</td>
<td>Proposition 3.6</td>
</tr>
<tr>
<td>$|x|_2 \leq \xi$</td>
<td>Proposition 3.7</td>
</tr>
</tbody>
</table>

**Proposition 3.3.** If $C = \{ x \in V \mid Ax = b \}$ is an affine set, then the subproblem (2.5) is solved by $u = PC(-e^{-1}h)$, where

$$ P_C(y) = y - A^t(Ay - b). $$

and

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

with

$$ \beta_1 := \frac{1}{2}\|A^t b\|_2^2 + Q_0, \quad \beta_2 := \langle A^t(Ah), A^t b \rangle + \gamma, \quad \beta_3 := \frac{1}{2}\|A^t(Ah)\|_2^2 + \frac{1}{2}\|h\|_2^2. $$

**Proof.** The projection operator on $C$ is given by (3.10). This and $y = -e^{-1}h$ give

$$ P_C(-e^{-1}h) = -e^{-1}(A^t(Ah + eb) - h). $$

This, together with (3.7), yields

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

$$ = \frac{1}{2}\|A^t(Ah + eb)\|_2^2 + \frac{1}{2}\|h\|_2^2 - \langle A^t(Ah + eb), h \rangle + Q_0e^2 $$

$$ + \gamma e + \langle A^t(Ah + eb) - h, h \rangle $$

$$ = \left(\frac{1}{2}\|A^t b\|_2^2 + Q_0\right)e^2 + (\langle A^t(Ah), A^t b \rangle + \gamma) e $$

$$ + \frac{1}{2}\|A^t(Ah)\|_2^2 + \frac{1}{2}\|h\|_2^2 $$

$$ = \beta_1e^2 + \beta_2e + \beta_3 = 0, $$

where $\beta_1, \beta_2$, and $\beta_3$ are defined in (3.12). Since the subproblem (2.5) is the maximization, the bigger root of this equation is selected, which is given by (3.11).

**Corollary 3.4.** If $C = \{ x \in V \mid a^T x = b \}$ is a hyperplane, then the subproblem (2.5) is solved by $u = PC(-e^{-1}h)$, where

$$ P_C(y) = y - \left(\frac{\langle a, y \rangle - b}{\|a\|_2^2}\right)a, $$
and \( e \) is given by (3.11) with

\[
\beta_1 := \frac{b}{2\|a\|^2} + Q_0, \quad \beta_2 := \frac{b\langle a, h \rangle}{\|a\|^2} + \gamma, \quad \beta_3 := \frac{1}{2} \frac{(a, h)^2}{\|a\|^2} - \frac{1}{2} \|h\|^2.
\]  

Proof. Since the hyperplane \( C = \{ x \in V \mid a^T x = b \} \) is an affine set, this is a special case of Proposition 3.3.

**Proposition 3.5.** If \( C = \{ x \in V \mid \langle a, x \rangle \leq b \} \) is a halfspace, then the subproblem (2.5) is solved by \( u = P_C(-e^{-1}h) \), where

\[
P_C(y) = y - \frac{\langle a, y \rangle - b}{\|a\|^2} a
\]

and \( e \) is given by (3.11) with

\[
\beta_1 := Q_0, \quad \beta_2 := \gamma, \quad \beta_3 := -\frac{1}{2} \|h\|^2,
\]

say \( e_1 \), and with \( \beta_1, \beta_2, \text{and} \beta_3 \) is given in (3.14), say \( e_2 \). If \( \langle a, h \rangle \geq e_1^{-1}b \) and \( \langle a, h \rangle \leq e_2^{-1}b \), then \( e = e_1 \). If \( \langle a, h \rangle \leq e_1^{-1}b \) and \( \langle a, h \rangle < e_2^{-1}b \), then \( e = e_2 \). If \( \langle a, h \rangle \geq e_1^{-1}b \) and \( \langle a, h \rangle < e_2^{-1}b \), then \( e = \max\{e_1, e_2\} \).

Proof. The projection operator on \( C \) is given by (3.15). This gives

\[
P_C(-e^{-1}h) = -e^{-1} \left( h + \frac{\langle a, h \rangle + eb}{\|a\|^2} a \right).
\]

If \( \langle a, h \rangle \geq -eb \), we obtain

\[
P_C(-e^{-1}h) = -e^{-1}h,
\]

leading to

\[
eQ(P_C(-e^{-1}h)) + \gamma + \langle h, P_C(-e^{-1}h) \rangle = \frac{1}{2} e^{-1} \|h\|^2 + Q_0e + \gamma - e^{-1} \|h\|^2
\]

\[
= Q_0e^2 + \gamma e - \frac{1}{2} \|h\|^2 = \beta_1 e^2 + \beta_2 e + \beta_3 = 0,
\]

where \( \beta_1 := Q_0, \beta_2 := \gamma, \text{and} \beta_3 := -\frac{1}{2} \|h\|^2. \) This identity leads to a solution of the form (3.11), say \( e_1 \). If \( \langle a, h \rangle < -eb \), (3.13) is valid and \( e \) is computed by (3.11) where \( \beta_1, \beta_2, \text{and} \beta_3 \) is defined in (3.14), say \( e_2 \). After computing \( e_1 \) and \( e_2 \), we check whether the inequalities \( \langle a, h \rangle \geq -e_1b \) and \( \langle a, h \rangle < -e_2b \) are satisfied. Since the subproblem (2.5) has a solution, at least one of the conditions has to be satisfied. If one of them is satisfied, the corresponding \( e \) and (3.17) give the solution. If both of them hold, we consider the solution with bigger \( e \). \( \Box \)

**Proposition 3.6.** If \( C = \{ x \in \mathbb{R}^n \mid x_i \geq 0 \quad i = 1, \cdots, n \} \) is the nonnegative orthant, then the subproblem (2.5) is solved by \( u = P_C(-e^{-1}h) \), where

\[
P_C(y) = (y)_+
\]

and \( e \) is given by (3.11) with

\[
\beta_1 := Q_0, \quad \beta_2 := \gamma, \quad \beta_3 := \frac{1}{2} \|h\|^2 - \langle h, (h) \rangle.
\]
Proof. The projection operator on $C$ is given by (3.18) leading to

$$PC(-e^{-1}h) = -e^{-1}(h)_.$$

This and (3.7) imply

$$eQ(PC(-e^{-1}h)) + \gamma + \langle h, PC(-e^{-1}h) \rangle = \frac{1}{2}e^{-1}\| (h)_- \|_2^2 + Q_0e + \gamma - e^{-1}\langle h, (h)_- \rangle$$

$$= Q_0e^2 + \gamma e + \frac{1}{2}\| (h)_- \|_2^2 - \langle h, (h)_- \rangle$$

$$= \beta_1e^2 + \beta_2e + \beta_3 = 0,$$

where $\beta_1$, $\beta_2$, and $\beta_3$ are defined in (3.19), giving the result. \[ \square \]

**Proposition 3.7.** Let $C = \{ x \in \mathbb{R}^n \mid \| x \|_2 \leq \xi \}$ be the Euclidean ball. Then

(3.20)

$$PC(y) = \begin{cases} \frac{\xi y}{\| y \|_2} & \| y \|_2 > \xi, \\ y & \| y \|_2 \leq \xi, \end{cases}$$

If $\| e^{-1}h \|_2 \leq \xi$ where $e$ is given by (3.11) with

(3.21)

$$\beta_1 := Q_0, \quad \beta_2 := \gamma, \quad \beta_3 := -\frac{1}{2}\| h \|_2^2,$$

then $u = -e^{-1}h$; otherwise, the solution of OSGA’s subproblem (2.5) is given by

$$u = -\frac{\xi}{\| h \|_2}h, \quad e = -\frac{2(\gamma + \xi\| h \|_2)}{\xi^2 + 2Q_0}.$$

Proof. The projection operator on $C$ is given by (3.20), leading to

$$PC(-e^{-1}h) = \begin{cases} -\xi h/\| h \|_2 & \| h \|_2 > e\xi, \\ -e^{-1}h & \| h \|_2 \leq e\xi. \end{cases}$$

We first assume that $\| h \|_2 \leq e\xi$ implying $PC(-e^{-1}h) = -e^{-1}h$. Substituting this into (3.7) yields

$$eQ(PC(-e^{-1}h)) + \gamma + \langle h, PC(-e^{-1}h) \rangle = \frac{1}{2}e^{-1}\| h \|_2^2 + Q_0e + \gamma - e^{-1}\| h \|_2^2$$

$$= Q_0e^2 + \gamma e - \frac{1}{2}\| h \|_2^2 = \beta_1e^2 + \beta_2e + \beta_3 = 0,$$

where $\beta_1 := Q_0$, $\beta_2 := \gamma$, and $\beta_3 := -\frac{1}{2}\| h \|_2^2$. Hence $e$ is given by (3.11). If this $e$ satisfies $\| h \|_2 \leq e\xi$, then $u = -e^{-1}h$. Otherwise, we assume that $\| h \|_2 > e\xi$. Substituting $PC(-e^{-1}h) = -e\xi h/\| h \|_2$ into (3.7) yields

$$e \left( \frac{1}{2}e^2 + Q_0 \right) + \gamma - e\xi\| h \|_2 = 0,$$

implying

$$e = -\frac{2(\gamma + \xi\| h \|_2)}{\xi^2 + 2Q_0}.$$
and \( u = -\xi \varepsilon / \| h \|_2 \). This completes the proof. \( \square \)

To solve bound-constrained problems with OSGA, we developed and algorithm that can find the global solution of the subproblem (2.5) by solving a sequence of one-dimensional rational optimization problems, see Algorithm 3 in [6]. Notice that the constraint \( C := \{ x \in V \mid \| x \|_\infty \leq \xi \} \) is a special case of bound-constrained problem with \( x = -\xi \mathbf{1} \) and \( \mathbf{1} = \xi \mathbf{1} \) where \( \mathbf{1} \) is an \( n \)-dimensional vector with all elements equal to unity.

4. Solving structured problems with a functional constraint. In this subsection we consider the structured convex constrained problem

\[
\begin{align*}
\min & \quad f(Ax) \\
\text{s.t.} & \quad \phi(x) \leq \xi,
\end{align*}
\]

where \( \phi : C \to \mathbb{R} \) is a simple smooth or nonsmooth, real-valued, and convex loss function, and \( \xi \) is a real constant. We call the problem (4.1) a functional constraint problem. While it the special case of (3.1) with

\[
C := \{ x \in V \mid \phi(x) \leq \xi \},
\]

one can solve OSGA’s subproblem (2.5) directly by using the KKT optimality conditions, especially when no efficient method for finding the projection on \( C \) is known. Indeed, if a nonsmooth problem can be reformulated in the form (3.1) with a smooth \( f \) and a nonsmooth \( \phi \), then OSGA can solve this nonsmooth problem with the complexity of the order \( O(\varepsilon^{-1/2}) \), which is optimal for smooth problems.

Example 4.1. (Linear inverse problem) Let \( A : \mathbb{R}^n \to \mathbb{R}^m \) be an ill-conditioned or singular linear operator and \( y \in \mathbb{R}^m \) be a vector of observations. The linear inverse problem is the quest of finding \( x \in \mathbb{R}^n \) such that

\[
y = Ax + \nu,
\]

with unknown but small additive noise \( \nu \in \mathbb{R}^m \). The problem is solvable if one knows additional qualitative information about \( x \). This qualitative information is encoded in a constraint on \( x \), under which the Euclidean norm of \( \nu \) is minimized. Constrained optimization problems resulting from two typical qualitative constraints are

\[
\min \frac{1}{2} \| y - Ax \|_2^2 \quad \text{s.t.} \quad \| x \|_2 \leq \xi,
\]

\[
\min \frac{1}{2} \| y - Ax \|_2^2 \quad \text{s.t.} \quad \| x \|_{1,2} \leq \xi,
\]
in which \( \xi \) is a nonnegative real constant. This problem often occurs in applied sciences and engineering, see [44, 64].

In the reminder of this section we assume that the functional constraint satisfies the Cottle constraint qualification [10]

(H1) For all \( x \in C \), either \( \phi(x) < 0 \) or \( 0 \not\in \partial \phi(x) \).

We also need the following result.

Proposition 4.2. (see, e.g., [5]) Let \( \phi : V \to \mathbb{R} \), \( \phi(x) = \| x \| \). Then the subdifferential of \( \phi \) is

\[
\partial \phi(x) = \begin{cases} 
\{ g \mid \| g \| \leq 1 \} & \text{if } x = 0, \\
\{ g \mid \| g \| = 1, \langle g, x \rangle = \| x \| \} & \text{if } x \neq 0.
\end{cases}
\]
Moreover, if $\|\cdot\|$ is self-dual, then
\[
\partial \phi(x) = \begin{cases} 
\{ g : \|g\| \leq 1 \} & \text{if } x = 0, \\
{x}/\|x\| & \text{if } x \neq 0.
\end{cases}
\]

The next result gives the optimality conditions for solving the problem (3.1).

**Theorem 4.1.** Let (H1) satisfies for the problem (4.1). Then, for a real constant $\xi$, the solution $u$ of OSGA’s subproblem
\[
\min \quad -\gamma - \langle h, x \rangle \quad \text{s.t.} \quad Q(x) \phi(x) \leq \xi,
\]
satisfies either
\[(4.5) \quad u = -e^{-1}h, \quad \mu = 0, \quad \phi(u) < \xi\]
or
\[(4.6) \quad \frac{1 - eu - h}{\mu Q(u)} \in \partial \phi(u), \quad \mu > 0, \quad \phi(u) = \xi,
\]
where $e := -(\gamma + \langle h, u \rangle)/Q(u)$.

**Proof.** Let’s define the function
\[E_{\gamma,h} : C \to \mathbb{R}, \quad E_{\gamma,h}(x) := -\gamma - \langle h, x \rangle \quad \text{s.t.} \quad Q(x) \phi(x) \leq \xi,
\]
Since this function is differentiable, by differentiating both sides of the equality $E_{\gamma,h}(x)Q(x) = -\gamma - \langle h, x \rangle$ with respect to $x$, we obtain
\[(4.7) \quad \partial E_{\gamma,h}(x) = \left\{ \frac{-E_{\gamma,h}(x)x - h}{Q(x)} \right\}.
\]
In view of the KKT optimality conditions for inequality constrained nonsmooth problems, see [10], we have the optimality condition
\[(4.8) \quad \begin{cases} 
0 \in \partial E_{\gamma,h}(u) + \mu \partial \phi(u), \\
\phi(u) \leq \xi, \\
\mu \geq 0, \\
\mu(\phi(u) - \xi) = 0,
\end{cases}
\]
for (3.1). Now, by substituting (4.7) into (4.8), setting $e := -(\gamma + \langle h, u \rangle)/Q(u)$, and distinguishing between $\mu = 0$ and $\mu > 0$, we obtain either (4.5) or (4.6). □

Theorem 4.1 gives the optimality conditions for general function $\phi$, however, in view of Theorem 3.1, it is especially useful when the projection in $C = \{ x : \phi(x) \leq \xi \}$ is not efficiently available. In the remainder of this subsection, we derive the solution of OSGA’s subproblem (2.5) for some $\phi$ such as $\|\cdot\|_2$ and $\|\cdot\|_1$ that appear in many applications. We already solve OSGA’s subproblem (2.5) with the constraint $C = \{ x : \|x\|_2 \leq \xi \}$ in Proposition 3.7, but to show how to apply Theorem 4.1 we study it in the next result.

**Proposition 4.2.** Let $\mathcal{V}$ be a real finite-dimensional Hilbert space with the induced norm $\phi(\cdot) = \|\cdot\|_2$. Then OSGA’s subproblem (2.5) is solved by
\[u = -e^{-1}h, \quad e = \frac{-\beta_2 + \sqrt{\beta_2^2 - 4\beta_1 \beta_3}}{-2\beta_1}, \quad \mu = 0,\]
where
\[ \beta_1 := Q_0, \quad \beta_2 := \gamma, \quad \beta_3 := \frac{1}{2} \|h\|_2^2, \]
if \( \phi(u) < \xi \); Otherwise it is solved by
\[ u = \frac{\xi}{\|h\|_2} h, \quad e = \frac{2\|h\|_2(\gamma \|h\|_2 + \xi \|h\|_2)}{\xi^2\|h\|_2^2 + 2Q_0\|h\|_2^2}, \quad \mu = \frac{2(\|h\|_2 + e\xi)\|h\|_2^3}{\|h\|_2^2 + 2Q_0\|h\|_2^2}. \]

**Proof.** Since \( \|\cdot\|_2 \) is self-dual, Proposition 4.2 implies
\[ \partial \phi(u) = \begin{cases} \{ g \in V^* \mid \|g\|_2 \leq 1 \} & \text{if } u = 0, \\ \frac{u}{\|u\|_2} & \text{if } u \neq 0. \end{cases} \]
As \( u = 0 \) is not useful in our optimization setting, we seek only \( u \neq 0 \). We now apply Theorem 4.1 leading to two cases: (i) (4.5) holds; (ii) (4.6) holds.

Case (i). The condition (4.5) holds. Then we have \( u = -e^{-1}h \). By substituting this into the identity \( E_{\gamma,h}(u) = e \), we get
\[ e = -\frac{\gamma - \|h\|_2 e^{-1}}{\frac{3}{2}\|h\|_2 e^{-2} + Q_0}, \]
implicating
\[ Q_0e^2 + \gamma e - \frac{1}{2}\|h\|_2^2 = 0. \]
By using the bigger root of this equation, we have
\[ e = \frac{-\beta_2 + \sqrt{\beta_2^2 - 4\beta_1\beta_3}}{-2\beta_1}, \]
where \( \beta_1 = Q_0, \beta_2 = \gamma, \) and \( \beta_3 = \frac{1}{2}\|h\|_2^2. \)

Case (ii). The condition (4.6) holds. Then we have
\[ -eu - h = -\mu \frac{u}{\frac{3}{2}\|u\|_2^2 + Q_0}, \]
giving
\[ (-eu - h)\|u\|_2 + \mu \left( \frac{1}{2}\|u\|_2^2 + Q_0 \right) u = 0, \]
leading to
\[ (-e\|u\|_2 + \frac{1}{2}\mu\|u\|_2^2 + \mu Q_0) u = \|u\|_2 h. \]
This implies that there exist \( \lambda \) such that \( u = \lambda h \). By substituting this into \( \phi(u) = \|u\|_2 = \xi \) we get
\[ \lambda = \frac{\xi}{\|h\|_2}. \]
Now, substituting $u$ into (4.9), we obtain

\[(4.10) \quad \mu = \frac{2(\|h\|_2 + e\xi)\|h\|_2^2}{\|h\|_2^2 + 2Q_0\|h\|_2^2}.\]

It follows from $E_{\gamma,h}(u) = e$ that

\[e = -\frac{2\|h\|_2(\gamma\|h\|_2 + \xi\|h\|_2)}{\xi^2\|h\|_2^2 + 2Q_0\|h\|_2^2}.\]

This gives the result. □

In 2004, YUAN and LIN in [70] proposed an interesting regularizer called grouped LASSO for the linear regression. Later KIM et al. in [44] proposed a constrained ridge regression model using the constraint

\[\|x\|_{1,2} \leq \xi,\]

where

\[\|x\|_{1,2} := \sum_{i=1}^{m} \|x_{g_i}\|_2,\]

where $x = (x_{g_1}, \ldots, x_{g_m})$ and $\|x\|_{1,2}$ is a so-called the $l_{1,2}$ group norm. We consider this constraint in the next result.

**Proposition 4.3.** Let $\mathcal{V}$ be a real finite-dimensional vector space with the induced norm $\phi(\cdot) = \|\cdot\|_{1,2}$. Then OSGA’s subproblem (2.5) is solved by

\[u_{g_i} = -\frac{1}{2}h_{g_i}, \quad \text{for all } i = 1, \ldots, m,\]

and

\[e = -\frac{\beta_2 + \sqrt{\beta_2^2 - 4\beta_1\beta_3}}{-2\beta_1}, \quad \mu = 0,\]

where

\[\beta_1 := Q_0, \quad \beta_2 := \gamma, \quad \beta_2 := \frac{1}{2}\|h\|_2^2 - \sum_{i=1}^{m} \|h_{g_i}\|_2^2,\]

if $\phi(u) < \xi$; Otherwise it is solved by

\[u_i = \rho_i h_{g_i}, \quad \rho_i = \frac{\|h_{g_i}\|_2 - \mu}{e\|h_{g_i}\|_2} \left(\frac{\frac{1}{2}\xi^2 + Q_0}{\|h_{g_i}\|_2}\right) \quad \text{for all } i = 1, \ldots, m,\]

and

\[e = \frac{\gamma + \langle h, u \rangle}{\frac{1}{2}\xi^2 + Q_0} = 2(\gamma + \sum_{i=1}^{m} \tau_i^2\|h_{g_i}\|_2^2) \left(\frac{\|h_{g_i}\|_2 + e\xi}{\|h_{g_i}\|_2 + 2Q_0}\right) \quad \mu = 2(\sum_{i=1}^{m} \tau_i^2\|h_{g_i}\|_2^2 + e\xi) \left(\frac{1}{m(\sum_{i=1}^{m} \tau_i^2\|h_{g_i}\|_2^2 + 2Q_0)}\right).\]

**Proof.** Similar to Proposition 4.2, we consider $u \neq 0$. In view of Proposition (4.2), we get

\[\partial \phi(u_{g_i}) = \left\{ \frac{u_{g_i}}{\|u_{g_i}\|_2} \right\} \quad \text{for all } i = 1, \ldots, m,\]
leading to
\[
\partial \phi(u) = \left\{ \left( \frac{u_{g_1}}{\|u_{g_1}\|_2}, \ldots, \frac{u_{g_m}}{\|u_{g_m}\|_2} \right) \right\}.
\]

We now apply Theorem 4.1 leading to two cases: (i) (4.5) holds; (ii) (4.6) holds.

Case (i). The condition (4.5) holds. Then we have
\[u_{g_i} = -e^{-1}h_{g_i}\text{ for } i = 1, \ldots, n\]
By substituting \(u = (u_{g_1}, \ldots, u_{g_m})\) into the identity \(E_{\gamma,h}(u) = e\), we get
\[e = \frac{-\gamma + \sum_{i=1}^m \|h_{g_i}\|_2^2}{1/2 \|h\|_2^2} e^{-1},\]
implying
\[Q_0 e^2 + \gamma e + \frac{1}{2}\|h\|_2^2 - \sum_{i=1}^m \|h_{g_i}\|_2^2 = 0.\]
By using the bigger root of this equation, we get
\[e = -\beta_2 + \sqrt{\beta_2^2 - 4\beta_1 \beta_3},\]
where \(\beta_1 := Q_0, \beta_2 := \gamma, \text{ and } \beta_3 := \frac{1}{2}\|h\|_2^2 - \sum_{i=1}^m \|h_{g_i}\|_2^2.\)

Case (ii). The condition (4.6) holds. Then we have
\[-e u_{g_i} - h_{g_i} = -\mu \frac{u_{g_i}}{\|u_{g_i}\|_2} \quad \text{for all } i = 1, \ldots, m.\]
Since \(\phi(u) = \|u\| = \xi\), we equivalently get
\[\left( \frac{1}{2\|u\|_2^2} + Q_0 \right) \left( -\frac{e}{2\|u\|_2^2} + \frac{\mu}{\|u_{g_i}\|_2} \right) u_{g_i} = h_{g_i},\]
implying \(u_{g_i} = \tau_i h_{g_i}\). If \(h_{g_i} = 0\), then \(u_{g_i} = 0\). Now let \(h_{g_i} \neq 0\). Substituting \(u_{g_i} = \tau_i h_{g_i}\) into the previous identity, it follows that
\[\left( \frac{1}{2} \sum_{i=1}^m \tau_i^2 \|h_{g_i}\|_2^2 + Q_0 \right) \left( -\frac{e}{2} \sum_{i=1}^m \tau_i^2 \|h_{g_i}\|_2^2 + Q_0 + \frac{\mu}{\tau_i \|h_{g_i}\|_2} \right) \tau_i h_{g_i} = h_{g_i},\]
giving
\[-e \tau_i \|h_{g_i}\|_2 + \mu \left( \frac{1}{2} \sum_{i=1}^m \tau_i^2 \|h_{g_i}\|_2^2 + Q_0 \right) = \|h_{g_i}\|_2 \quad \text{for all } i = 1, \ldots, m.\]
Applying a summation from both sides, together with \(\sum_{i=1}^m \tau_i \|h_{g_i}\|_2 = \xi\), yields
\[\frac{1}{m} \sum_{i=1}^m \tau_i^2 \|h_{g_i}\|_2^2 + Q_0 = \sum_{i=1}^m \|h_{g_i}\|_2.\]
implying
\[\mu = \frac{2(\sum_{i=1}^m \|h_{g_i}\|_2 + \epsilon \xi)}{m(\sum_{i=1}^m \tau_i^2 \|h_{g_i}\|_2^2 + 2Q_0)}\].
By substituting this into (4.11), we have
\[ \tau_i = -\frac{1}{m \epsilon} \|h_{g_i}\|_2 \left( m \|h_{g_i}\|_2 - \sum_{i=1}^{m} \|h_{g_i}\|_2 - \epsilon \xi \right) \]
leading to
\[ u = (\tau_1 h_{g_1}, \ldots, \tau_m h_{g_m}). \]
By substituting this into \( E_{\gamma,h}(u) = e \), we get
\[ e = -\frac{\gamma + \langle h, u \rangle}{\frac{1}{2} \xi^2 + Q_0} = -\frac{2(\gamma + \sum_{i=1}^{n} \tau_i^2 \|h_{g_i}\|_2^2)}{\sum_{i=1}^{n} \tau_i^2 \|h_{g_i}\|_2^2 + 2Q_0}, \]
giving the result.

The package is written in MATLAB; it uses the parameters
\[ \delta = 0.9; \quad \alpha_{\max} = 0.7; \quad \kappa = \kappa' = 0.5; \quad \Psi_{\text{target}} = -\infty. \]
and the prox-function (3.5) with \( Q_0 = \frac{1}{2} \|x_0\|_2 + \epsilon \), where \( \epsilon \) is the machine precision.
A user manual [2] describes the design and use of the package. Some examples are included as illustrations.

This section discusses numerical results and comparisons of OSGA with some state-of-the-art first-order solvers on some ridge regression and image deblurring problems. All numerical results were created with version 1.1 of the above software. The algorithms used for comparison use the default parameter values reported in the corresponding papers or packages. All numerical experiments were executed on a Toshiba Satellite Pro L750-176 laptop with Intel Core i7-2670QM processor and 8 GB RAM.

5.1. Ridge regression. In this subsection we consider a \( l_2 \)-constrained least squares of the form (4.3) (so-called ridge regression, see [47]) and report some numerical results.
The problem is generated by
\[ \begin{align*}
[A,z,x] &= \text{i_laplace}(n), \quad y = z + 0.1 \ast \text{rand},
\end{align*} \]
where \( n = 5000 \) is the problem dimension and \text{i_laplace.m} is an ill-posed test problem generator using the inverse Laplace transformation from Regularization Tools MATLAB package, which is available in
Since (4.3) is smooth and the projection on \( C = \{ x \in \mathbb{R}^n \mid \|x\| \leq \xi \} \) is available (see Table 1), we employ gradient projection algorithm (PGA), the spectral gradient projection [23] with the GRIPPO et al. nonmonotone term [37] (SPG-G), the spectral gradient projection with the AMINI et al. nonmonotone term [8] (SPG-A), and OSGA (see Proposition 4.2) to solve this minimization problem. The parameters of SPG-G and SPG-A are the same as those reported in the associated papers, but SPG-A uses
\[ \eta_k = \begin{cases} 
\eta_0/2 & \text{if } k = 1, \\
(\eta_{k-1} + \eta_{k-2})/2 & \text{if } k \geq 2.
\end{cases} \]
The algorithms are stopped after 500 iterations.

In Table 3 we consider $\xi = 10, 15, 20, 25$ and report the best attained function values and the running time. The results imply that OSGA attains the best running time and except for $\xi = 15$ gives the best function values. To see the results of implementation in more details, we demonstrate the relative error of function values

\begin{equation}
\delta_k := \frac{f_k - \hat{f}}{f_0 - \hat{f}}
\end{equation}

in Figure 1, where $\hat{f}$ denotes the minimum and $f_0$ shows the function value on an initial point $x_0$.

5.2. Image deblurring with nonnegativity constraint. As discussed in Section 3, inverse problems are appearing in many fields of applied sciences and Engineering. This is particularly happen when researchers use digital images to record and analyze results from experiments in many fields such as astronomy, medical sciences, biology, geophysics, and physics. In these cases, observing blurred and noisy images is a common phenomenon happening frequently because of environmental effects and imperfections in the imaging system.

In many applications, the variable $x$ describes physical quantities, which is meaningful if each component of $x$ is restricted to be nonnegative. This constraint is referred as the nonnegativity constraint; it is especially useful for restoring blurred and noisy images, see [11, 42, 43, 66].

We restore the $256 \times 256$ blurred and noisy MR-brain image using the model (3.2) equipped with the isotropic total variation regularizer. The true image is available in http://graphics.stanford.edu/data/voldata/.

The blurred/noisy image $y$ is generated by a $9 \times 9$ uniform blur and adding a Gaussian noise with zero mean and standard deviation set to $10^{-3}$. For restoring the image, we use OSGA (see Proposition 3.6), MFISTA (a monotone version of FISTA proposed by Beck & Teboulle in [17]), ADMM (an alternating direction method proposed by Chan et al. in [30]), and PSGA (a projected subgradient scheme with nonsummable diminishing step size), see [27]. The original codes of MFISTA and ADMM provided by the authors are used. Since the methods are sensitive to the regularization parameter $\lambda$, three different regularization parameters are used. The algorithms are stopped after 100 iterations. The comparison concerning the quality of the recovered image is

Table 3: Result summary for the ridge regression

<table>
<thead>
<tr>
<th>$\xi$</th>
<th>PGA</th>
<th>SPG-G</th>
<th>SPG-A</th>
<th>OSGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_b$</td>
<td>$101.70e-3$</td>
<td>$7.60e-3$</td>
<td>$6.41e-3$</td>
<td>$3.60e-3$</td>
</tr>
<tr>
<td>Time(s)</td>
<td>$77.78$</td>
<td>$30.08$</td>
<td>$31.20$</td>
<td>$22.09$</td>
</tr>
<tr>
<td>$f_b$</td>
<td>$48.23e-3$</td>
<td>$1.70e-3$</td>
<td>$1.31e-3$</td>
<td>$1.52e-3$</td>
</tr>
<tr>
<td>Time(s)</td>
<td>$66.54$</td>
<td>$25.00$</td>
<td>$24.24$</td>
<td>$21.55$</td>
</tr>
<tr>
<td>$f_b$</td>
<td>$23.08e-2$</td>
<td>$2.01e-2$</td>
<td>$1.74e-2$</td>
<td>$8.60e-3$</td>
</tr>
<tr>
<td>Time(s)</td>
<td>$64.60$</td>
<td>$28.47$</td>
<td>$27.11$</td>
<td>$21.40$</td>
</tr>
<tr>
<td>$f_b$</td>
<td>$23.00e-2$</td>
<td>$2.22e-2$</td>
<td>$1.24e-2$</td>
<td>$8.96e-3$</td>
</tr>
<tr>
<td>Time(s)</td>
<td>$62.55$</td>
<td>$30.20$</td>
<td>$31.18$</td>
<td>$26.50$</td>
</tr>
</tbody>
</table>
Fig. 1: A comparison among PGA, SPG-G, SPG-A, and OSGA for solving the problem (4.3) based on the relative error of function values $\delta_k$ (5.1). The algorithms were stopped after 500 iterations.

made via the so-called peak signal-to-noise ratio (PSNR) defined by

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

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

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

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 $[0, 1]$. The results of implementation are summarized in Table 4 and Figures 2 and 3.

In Table 4 we report PSNR, the best available approximation $f_b$ of the minimimum, and the running time in seconds for three different regularization parameters. The results reported in Figure 2 regarding function values and ISNR show that the algorithms considered are sensitive to the parameter $\lambda$, however, the best results
Fig. 2: A comparison among PSGA, MFISTA, ADMM, and OSGA for deblurring the 256 × 256 MR-brain image with the 9 × 9 uniform blur and the Gaussian noise with deviation $10^{-3}$. The algorithms were stopped after 100 iterations. Subfigures (a), (c), and (e) display the relative error of function values $\delta_k$ (5.1) versus iterations, and Subfigures (b), (d), and (f) display ISNR (5.3) versus iterations.
Fig. 3: Deblurring of the $256 \times 256$ MR-brain image with the $9 \times 9$ uniform blur and the Gaussian noise with deviation $10^{-3}$ by PSGA, MFISTA, ADMM, and OSGA with the regularization parameter $\lambda = 10^{-4}$. The algorithms were stopped after 100 iterations.
Table 4: Result summary for L22ITV

<table>
<thead>
<tr>
<th>λ</th>
<th>PSNR</th>
<th>MFISTA</th>
<th>ADMM</th>
<th>OSGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_b = 5 \times 10^{-4}$</td>
<td>32.59</td>
<td>32.67</td>
<td>32.66</td>
<td>32.73</td>
</tr>
<tr>
<td>Time(s)</td>
<td>1.14</td>
<td>7.61</td>
<td>1.11</td>
<td>1.82</td>
</tr>
<tr>
<td>PSNR</td>
<td>33.23</td>
<td>33.96</td>
<td>33.95</td>
<td>33.97</td>
</tr>
<tr>
<td>$f_b = 1 \times 10^{-4}$</td>
<td>0.1184</td>
<td>0.0960</td>
<td>0.0958</td>
<td>0.0980</td>
</tr>
<tr>
<td>Time(s)</td>
<td>1.14</td>
<td>7.34</td>
<td>1.04</td>
<td>1.71</td>
</tr>
<tr>
<td>PSNR</td>
<td>33.24</td>
<td>34.45</td>
<td>34.49</td>
<td>34.46</td>
</tr>
<tr>
<td>$f_b = 5 \times 10^{-5}$</td>
<td>0.1174</td>
<td>0.0653</td>
<td>0.0651</td>
<td>0.0669</td>
</tr>
<tr>
<td>Time(s)</td>
<td>1.15</td>
<td>6.51</td>
<td>1.06</td>
<td>1.67</td>
</tr>
</tbody>
</table>

obtained for $\lambda = 10^{-4}$. More specifically, the results about function values in subfigures (a), (c), and (e) demonstrate that OSGA outperforms PGA, which means it performs much better than the lower complexity bound $O(\varepsilon^{-2})$, however, it cannot perform similar to MFISTA attaining the complexity of the order $O(\varepsilon^{-1/2})$. Subfigures (b), (d), and (f) show that OSGA is comparable with MFISTA and ADMM and even better than them in the sense of ISNR. The deblurred images by the algorithms considered are illustrated in Figure 3 for $\lambda = 10^{-4}$.

We also consider the restoration of the 641 \times 641 blurred/noisy Dione image using (3.3). The true image is available in http://photojournal.jpl.nasa.gov/Help/ImageGallery.html. The blurred/noisy image is constructed from the 7 \times 7 Gaussian kernel with standard deviation 5 and salt-and-pepper impulsive noise with the level 50%. To recover the image, we use DRPD-1, DRPD-2 (Douglas-Rachford primal-dual schemes proposed by Bot & Hendrich in [25]), ADMM, and OSGA. The algorithms are stopped after 100 iterations, and three different regularization parameters are considered. The results of implementation are reported in Table 5 and Figures 4 and 5.

The results of Table 5 shows that OSGA outperforms the others in the sense of PSNR. Figure 4 indicates that OSGA attains the best function values for $\lambda = 10^{-1}$ and $\lambda = 5 \times 10^{-2}$, however, ADMM get the best function value for $\lambda = 5 \times 10^{-1}$. It also implies that OSGA are comparable or even better that the others regarding ISNR. The resulted images for $\lambda = 10^{-1}$ are illustrated in Figure 5, demonstrating that the algorithms can restore the image by acceptable qualities while OSGA obtains the best function value and PSNR.

6. Conclusions. In this paper an optimal subgradient method, OSGA, is addressed for solving structured convex constrained optimization. More specifically, finding a solution of OSGA’s subproblem is investigated in the presence of some convex constraints. Two types of convex constraints are considered, namely, simple convex domains, in which the orthogonal projection in the domains is effectively available, and functional constraints, defined as the sublevel sets of simple convex functions. In each case some interesting examples are discussed for which OSGA’s subproblem can be solved efficiently. Numerical results and comparisons with some state-of-the-art algorithms are reported showing that OSGA is efficient and reliable for solving convex optimization problems in applications.
Fig. 4: A comparison among DRPD-1, DRPD-2, ADMM, and OSGA for deblurring the 641 × 641 Dione image with the various regularization parameter $\lambda$. 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 100 iterations. Subfigures (a), (c), and (e) display the relative error of function values $\delta_k$ (5.1) versus iterations, and (b), (d), and (f) demonstrate ISNR (5.3) versus iterations.
Fig. 5: Deblurring of the 641 × 641 Dione image using DRPD-1, DRPD-2, ADMM and OSGA with the parameter $\lambda = 10^{-1}$. The algorithms were stopped after 100 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%.
## Table 5: Results summary for L1ITV

<table>
<thead>
<tr>
<th>λ</th>
<th>PSNR</th>
<th>DRPD-1</th>
<th>DRPD-2</th>
<th>ADMM</th>
<th>OSGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 × 10⁻¹</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f_b</td>
<td>37.43</td>
<td>36.66</td>
<td>37.42</td>
<td>37.50</td>
<td>37.50</td>
</tr>
<tr>
<td>Time</td>
<td>10.86</td>
<td>6.83</td>
<td>8.57</td>
<td>9.01</td>
<td></td>
</tr>
<tr>
<td>1 × 10⁻¹</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f_b</td>
<td>38.70</td>
<td>38.11</td>
<td>38.35</td>
<td>38.73</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>10.43</td>
<td>6.68</td>
<td>8.46</td>
<td>8.32</td>
<td></td>
</tr>
<tr>
<td>5 × 10⁻²</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f_b</td>
<td>37.09</td>
<td>36.77</td>
<td>30.06</td>
<td>37.06</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>10.26</td>
<td>6.27</td>
<td>8.25</td>
<td>9.23</td>
<td></td>
</tr>
</tbody>
</table>

### 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


2434. [18]


[38] A. A. GOLOSTEIN, On steepest descent, SIAM Journal on Control, 3 (1965), pp. 147–151. [1]


[43] L. KAUFMAN, A. NEUMAER, Regularization of ill-posed problems by envelope guided conjugate
OSGA for convex optimization with simple constraints


