

OSGA: A fast subgradient algorithm with optimal complexity

Arnold Neumaier

*Fakultät für Mathematik, Universität Wien
Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria
email: Arnold.Neumaier@univie.ac.at
WWW: <http://www.mat.univie.ac.at/~neum/>*

May 9, 2015

Abstract. This paper presents an algorithm for approximately minimizing a convex function in simple, not necessarily bounded convex, finite-dimensional domains, assuming only that function values and subgradients are available. No global information about the objective function is needed apart from a strong convexity parameter (which can be put to zero if only convexity is known). The worst case number of iterations needed to achieve a given accuracy is independent of the dimension and – apart from a constant factor – best possible under a variety of smoothness assumptions on the objective function.

Keywords: complexity bound, convex optimization, optimal subgradient method, large-scale optimization, Nesterov’s optimal method, nonsmooth optimization, optimal first-order method, smooth optimization, strongly convex

2010 MSC Classification: primary 90C25; secondary 90C60, 49M37, 65K05, 68Q25

1 Introduction

In the recent years, first order methods for convex optimization have become prominent again as they are able to solve large-scale problems in millions of variables (often arising from applications to image processing, compressed sensing, or machine learning), where matrix-based interior point methods cannot even perform a single iteration. (However, a matrix-free interior point method by FOUNTOULAKIS et al. [15] works well in some large compressed sensing problems.)

In 1983, NEMIROVSKY & YUDIN [23] proved lower bounds on the complexity of first order methods (measured in the number of subgradient calls needed to achieve a given accuracy) for convex optimization under various regularity assumptions for the objective functions. (See NESTEROV [25, Sections 2.1.2 and 3.2.1] for a simplified account.) They constructed convex, piecewise linear functions in dimensions $n > k$, where no first order method can have function values more accurate than $O(k^{-1/2})$ after k subgradient evaluations. This implies the need for at least $\Omega(\varepsilon^{-2})$ subgradient evaluations in the worst case if f is a nondifferentiable but Lipschitz continuous convex function. They also constructed convex quadratic functions in dimensions $n \geq 2k$ where no first order method can have function values more accurate than

$O(k^{-2})$ after k gradient evaluations. This implies the need for at least $\Omega(\varepsilon^{-1/2})$ gradient evaluations in the worst case if f is an arbitrarily often differentiable convex function. However in case of strongly convex functions with Lipschitz continuous gradients, the known lower bounds on the complexity allow a dimension-independent linear rate of convergence $\Omega(q^k)$ with $0 < q = 1 - \sqrt{\sigma/L}$, where σ is a strong convexity constant of f and L a Lipschitz constant of its gradient.

Algorithms by NESTEROV [25, 26, 29] (dating back in the unconstrained, not strongly convex case to 1983 [24]), achieve the optimal complexity order in all three cases. These algorithms need as input the knowledge of global constants – a global Lipschitz constant for the objective functions in the nonsmooth case, a global Lipschitz constant for the gradient in the smooth case, and an explicit constant of strong convexity in the strongly convex case. Later many variants were described (see, e.g., AUSLENDER & TEBoulLE [7], LAN et al. [21]), some of which are adaptive in the sense that they estimate all required constants during the execution of the algorithm. BECK & TEBoulLE [11] developed an adaptive proximal point algorithm called FISTA, popular in image restauration applications. Like all proximal point based methods, the algorithm needs more information about the objective function than just subgradients, but delivers in return a higher speed of convergence. TSENG [33] gives a common uniform derivation of several variants of fast first order algorithms based on proximal points. BECKER et al. [12] add (among other things) adaptive features to Tseng’s class of algorithms, making them virtually independent of global (and hence often pessimistic) Lipschitz information. Other such adaptive algorithms include GONZAGA et al. [16, 17] and MENG & CHEN [22]. DEVOLDER et al. [14] show that both the nonsmooth case and the smooth case can be understood in a common way in terms of inexact gradient methods. Recent work by NESTEROV [31] discusses methods where the amount of smoothness is measured in terms of Hölder conditions, interpolating between the assumptions of Lipschitz continuity and that of Lipschitz continuity of the gradient, and where no knowledge of the relevant constants is needed. It is likely that similar results as in these papers can be proved for OSGA.

If the Lipschitz constant is very large, the methods with optimal complexity for the smooth case are initially much slower than the methods that have an optimal complexity for the nonsmooth case. This counterintuitive situation was remedied by LAN [20], who provides an algorithm that needs (expensive auxiliary computations but) no knowledge about the function except convexity and has the optimal complexity, both in the nonsmooth case and in the smooth case, without having to know whether or not the function is smooth. However, its worst case behavior on strongly convex problem is unknown. Similarly, if the constant of strong convexity is very tiny, the methods with optimal complexity for the strongly convex case are initially much slower than the methods that do not rely on strong convexity. Prior to the present work, no algorithm was known with optimal complexity both for the general nonsmooth case and for the strongly convex case.

Content. In this paper, we derive an algorithm for approximating a solution $\hat{x} \in C$ of the convex optimization problem

$$f(\hat{x}) = \hat{f} := \min_{x \in C} f(x) \tag{1}$$

using first order information (function values f and subgradients g) only. Here $f : C \rightarrow \mathbb{R}$ is a proper and closed convex function defined on a nonempty, closed and convex subset C of a finite-dimensional real vector space V with bilinear pairing $\langle h, z \rangle$ defined for $z \in V$ and h in

the dual space V^* . The minimum in (1) exists if there is a point $x_0 \in C$ such that the level set $\{x \in C \mid f(x) \leq f(x_0)\}$ is bounded.

Our method is based on monotonically reducing bounds on the error $f(x_b) - \hat{f}$ of the function value of the currently best point x_b . These bounds are derived from suitable linear relaxations and inequalities obtained with the help of a prox function. The solvability of an auxiliary optimization subproblem involving the prox function is assumed. In many cases, this auxiliary subproblem has a cheap, closed form solution; this is shown here for the unconstrained case with a quadratic prox function, and in AHOOKHOSH & NEUMAIER [4, 5, 6] for more general cases involving simple and practically important convex sets C .

The OSGA algorithm presented here provides a fully adaptive alternative to current optimal first order methods. If no strong convexity is assumed, it shares the uniformity, the lack of need to estimate Lipschitz constants, and the optimal complexity properties of Lan’s method, but has a far simpler structure and derivation. Beyond that, it also gives the optimal complexity in the strongly convex case, though it needs in this case – like all other known methods with provable optimal linear convergence rate – the knowledge of an explicit constant of strong convexity. Furthermore – like Nesterov’s $O(k^{-2})$ algorithm from [26] for the smooth case, but unlike his linearly convergent algorithm for the strongly convex case, scheme (2.2.19) in NESTEROV [25], the algorithm derived here does not evaluate f and g outside their domain. The method for analyzing the complexity of OSGA is also new; neither Tseng’s complexity analysis nor Nesterov’s estimating sequences are applicable to OSGA.

The OSGA algorithm can be used in place of Nesterov’s optimal algorithms for smooth convex optimization and its variants whenever the latter are traditionally employed. Thus it may be used as the smooth solver with methods for solving nonsmooth convex problems via smoothing (NESTEROV [26]), and for solving very large linear programs (see, e.g., AYBAT & IYENGAR [9], CHEN & BURER [13], GU et al. [18], NESTEROV [27, 28], RICHTARIK [32])

Related results. For unconstrained problems ($C = V$), AHOOKHOSH [1] gives extensive numerical experiments and comparisons of OSGA with popular first-order methods (e.g., FISTA and several of Nesterov’s methods) for applications to inverse problems involving multi-term composite functions but no constraints. Remarkably, the performance is often close to that expected for smooth problems even when the problem is nonsmooth, whereas the other methods often perform close to the worst case. Therefore, OSGA is significantly better in speed and accuracy than earlier methods designed for nonsmooth problems.

In many case where proximal point procedures need to resort to approximations, subgradients can be calculated exactly, making OSGA more easily applicable. For example, this is the case in total variation imaging problems, where the results of [1] show that OSGA is superior to FISTA and related algorithms although the latter have a stronger complexity guarantee.

The only method comparable in quality is the 1983 method of NESTEROV [24] designed for smooth problems but “misused” for nonsmooth problems by pretending that the subgradient is a gradient. However, there is currently no convergence proof for Nesterov’s 1983 smooth method when used in this way.

AHOOKHOSH & NEUMAIER [2, 3] solve unconstrained convex problems involving costly linear

operators by combining OSGA and a multi-dimensional subspace search technique. AHOOKHOSH & NEUMAIER [4, 5, 6] discuss OSGA for various classes of constrained problems. [4] proves that a class of structured nonsmooth convex constrained problems generalizing the problem class considered by NESTEROV [30, 31], may be rewritten as smooth problems with simple constraints solvable with OSGA with a complexity of order $O(\varepsilon^{-1/2})$. [5] shows that the auxiliary subproblem can be solved for bound-constrained problems. [6] shows that for many other types of constraints appearing in applications, the auxiliary subproblem can be solved efficiently, either in a closed form or by a simple iterative scheme.

Future work. The new approach opens a number of lines for further research. I'd like to mention in particular two extensions.

Many first order algorithms for convex optimization remain well-behaved even when auxiliary (e.g., proximal point) computations are only done approximately. It should be possible to show for OSGA that the approximate solution of the subproblem and the approximate calculation of subgradients calculations does not significantly affect the quality of the iteration before the function values match the optimum within a reasonable accuracy.

It would be very interesting to extend the technique to problems involving convex functional constraints. That this might be possible is suggested by the existence of algorithms such as the CoMirror method (BECK et al. [10]) that handle convex problems with simple constraints and a single convex functional constraint $g(x) \leq 0$ (where $g(x)$ may be taken as the maximum of several convex functions to cover multiple functional constraints).

Acknowledgment. I'd like to thank Masoud Ahookhosh for numerous useful remarks on earlier versions of the manuscript. Thanks also to the referees for a number of suggestions that improved the paper.

2 Bounds from prox functions

In this section we motivate the new algorithm by proving that the solution of a simple auxiliary problem leads to a bound on the difference between a function value and the optimal function value. We then show how to solve the auxiliary problem in the unconstrained case.

In the following, V denotes a finite-dimensional Banach space with norm $\|\cdot\|$, and V^* is the dual Banach space with the dual norm $\|\cdot\|_*$. C is a nonempty, closed and convex subset of V . The objective function $f : C \rightarrow \mathbb{R}$ is assumed to be proper, closed and convex, and $g(x)$ denotes a particular computable subgradient of f at $x \in C$.

2.1 The basic idea

The method is based on monotonically reducing bounds on the error $f(x_b) - \hat{f}$ of the function value of the currently best point x_b . These bounds are derived from suitable linear relaxations

$$f(z) \geq \gamma + \langle h, z \rangle \quad \text{for all } z \in C \tag{2}$$

(where $\gamma \in \mathbb{R}$ and $h \in V^*$) with the help of a continuously differentiable **prox function** $Q : C \rightarrow \mathbb{R}$ satisfying

$$Q_0 := \inf_{z \in C} Q(z) > 0, \quad (3)$$

$$Q(z) \geq Q(x) + \langle g_Q(x), z - x \rangle + \frac{1}{2} \|z - x\|^2 \quad \text{for all } x, z \in C, \quad (4)$$

where $g_Q(x)$ denotes the gradient of Q at $x \in C$. (Thus Q is strongly convex with strong convexity parameter $\sigma = 1$. Choosing $\sigma = 1$ simplified the formulas, and is no restriction of generality, as we may always rescale a prox function to enforce $\sigma = 1$.) We require that

$$E(\gamma, h) := - \inf_{z \in C} \frac{\gamma + \langle h, z \rangle}{Q(z)} = \sup_{z \in C} E_{\gamma, h}(z), \quad (5)$$

, where $E_{\gamma, h}(z) = -\frac{\gamma + \langle h, z \rangle}{Q(z)}$, is attained for each $\gamma \in \mathbb{R}$ and $h \in V^*$ at some $z = U(\gamma, h) \in C$.

2.1 Proposition. *Let*

$$\gamma_b := \gamma - f(x_b), \quad u := U(\gamma_b, h), \quad \eta := E(\gamma_b, h). \quad (6)$$

Then

$$0 \leq f(x_b) - \hat{f} \leq \eta Q(\hat{x}). \quad (7)$$

In particular, if x_b is not yet optimal then the choice (6) implies $E(\gamma_b, h) > 0$.

Proof. Our requirements imply that, for arbitrary $\gamma_b \in \mathbb{R}$ and $h \in V^*$,

$$\gamma_b + \langle h, z \rangle \geq -E(\gamma_b, h)Q(z) \quad \text{for all } z \in C.$$

Now (7) follows from (2) for $z = \hat{x}$ together with (3). If x_b is not optimal then the left inequality in (7) is strict, and since $Q(z) \geq Q_0 > 0$, we conclude that $0 < \eta = E(\gamma_b, h)$. \square

Note that the form of the auxiliary optimization problem (5) is forced by this argument. Although this is a nonconvex optimization problem, it is shown in AHOOKHOSH & NEUMAIER [4, 5, 6] that there are many important cases where for appropriate prox functions, $\eta = E(\gamma, h)$ and $u = U(\gamma, h)$ are cheap to compute. In particular, we shall show in Subsection 2.3 that this is the case when $C = V$ and the prox function is quadratic. Typically, u and η in (6) are computed together.

If an upper bound for $Q(\hat{x})$ is known or assumed, the bound (7) translates into a computable error estimate for the minimal function value. But even in the absence of such an upper bound, we can solve the optimization problem (1) to a target accuracy

$$0 \leq f(x_b) - \hat{f} \leq \varepsilon Q(\hat{x}) \quad (8)$$

if we manage to decrease the error factor η from its initial value until $\eta \leq \varepsilon$ for some target tolerance $\varepsilon > 0$. This will be achieved by Algorithm 3.4 defined below. We shall prove for this algorithm complexity bounds on the number of iterations that are independent of the dimension of V (which may be infinite), and – apart from a constant factor – best possible under a variety of assumptions on the objective function.

2.2 Optimality conditions for the auxiliary optimization problem

The optimality conditions for the optimization problem (5) associated play an important role both for the construction of methods for solving (5) and for the derivation of bounds for the error factor η .

2.2 Proposition. *Let $\eta = E(\gamma, h) > 0$ and $u = U(\gamma, h)$. Then*

$$\gamma + \langle h, u \rangle = -\eta Q(u), \quad (9)$$

$$\langle \eta g_Q(u) + h, z - u \rangle \geq 0 \quad \text{for all } z \in C, \quad (10)$$

$$\gamma + \langle h, z \rangle \geq \eta \left(\frac{1}{2} \|z - u\|^2 - Q(z) \right) \quad \text{for all } z \in C. \quad (11)$$

Proof. The definition (5) implies that the function $\phi : C \rightarrow \mathbb{R}$ defined by

$$\phi(z) := \gamma + \langle h, z \rangle + \eta Q(z)$$

is nonnegative and vanishes for $z = u := U(\gamma, h)$. In particular, (9) holds. Since $\phi(z)$ is continuously differentiable with gradient $g_\phi(z) = h + \eta g_Q(z)$, the first order optimality condition

$$\langle g_\phi(u), z - u \rangle \geq 0 \quad \text{for all } z \in C \quad (12)$$

holds, and (10) follows. Since $\eta > 0$ and $Q(z)$ is strongly convex with parameter 1, $\phi(z)$ is strongly convex with parameter η . Therefore

$$\phi(z) - \phi(u) - \langle g_\phi(u), z - u \rangle \geq \frac{\eta}{2} \|z - u\|^2,$$

and (11) follows from (12). □

2.3 Unconstrained problems with quadratic prox function

To use the algorithm in practice, we need prox functions for which $E(\gamma, h)$ and $U(\gamma, h)$ can be evaluated easily. A number of simple domains C and prox functions for which this is possible (based on Proposition 2.2) are discussed in AHOOKHOSH & NEUMAIER [4, 5, 6].

Here we only discuss the simplest case, where the original optimization problem is unconstrained (so that $C = V$) and the norm on V is Euclidean,

$$\|z\| := \sqrt{\langle Bz, z \rangle},$$

where the **preconditioner** B is a symmetric and positive definite linear mapping $B : V \rightarrow V^*$. The associated dual norm on V^* is then given by

$$\|h\|_* := \|B^{-1}h\| = \sqrt{\langle h, B^{-1}h \rangle}.$$

Given the preconditioner, it is natural to consider the quadratic prox function

$$Q(z) := Q_0 + \frac{1}{2}\|z - z_0\|^2, \quad (13)$$

where Q_0 is a positive number and $z_0 \in V$. As the quotient of a linear and a positive quadratic function, $E_{\gamma,h}(z)$ is arbitrarily small outside a ball of sufficiently large radius. Therefore the level sets for positive function values are compact, and the supremum is attained on any of these sets set.

By Proposition 2.1, we may assume that $\eta := E(\gamma, h) > 0$. Since $C = V$ and $g_Q(z) = B(z - z_0)$, we conclude from the proposition that $E(\gamma, h)B(u - z_0) + h = 0$, where $u = U(\gamma, h)$, so that

$$U(\gamma, h) = z_0 - \eta^{-1}B^{-1}h. \quad (14)$$

Inserting this into (9), we find

$$\eta\left(Q_0 + \frac{1}{2}\| - \eta^{-1}B^{-1}h\|^2\right) = \eta Q(u) = -\gamma - \langle h, z_0 - \eta^{-1}B^{-1}h \rangle,$$

which simplifies to the quadratic equation

$$Q_0\eta^2 + \beta\eta - \frac{1}{2}\|h\|_*^2 = 0, \quad \beta = \gamma + \langle h, z_0 \rangle.$$

Since the left hand side is negative at $\eta = 0$, there is exactly one positive solution, which therefore is the unique maximizer. It is given by

$$E(\gamma, h) = \frac{-\beta + \sqrt{\beta^2 + 2Q_0\|h\|_*^2}}{2Q_0} = \frac{\|h\|_*^2}{\beta + \sqrt{\beta^2 + 2Q_0\|h\|_*^2}}. \quad (15)$$

(The first form is numerically stable when $\beta \leq 0$, the second when $\beta > 0$.)

A reasonable choice is to take for z_0 the starting point of the iteration, and to use an order of magnitude guess $Q_0 \approx \frac{1}{2}\|\hat{x} - z_0\|^2$.

3 The OSGA algorithm

In this section we derive all details needed to formulate the new algorithm.

Constructing linear relaxations. The convexity of f implies for $x, z \in C$ the bound

$$f(z) \geq f(x) + \langle g(x), z - x \rangle, \quad (16)$$

where $g(x)$ denotes a subgradient of f at $x \in C$. Therefore (2) always holds with

$$\gamma = f(x_b) - \langle g(x_b), x_b \rangle, \quad h = g(x_b).$$

We can find more general relaxations of the form (2) by accumulating past information. Indeed, if (2) holds, $\alpha \in [0, 1]$, and $x \in C$ then (2) remains valid when we substitute

$$\bar{\gamma} := \gamma + \alpha(f(x) - \langle g(x), x \rangle - \gamma),$$

$$\bar{h} := h + \alpha(g(x) - h)$$

in place of γ, h , as by (16),

$$\begin{aligned} f(z) &= (1 - \alpha)f(z) + \alpha f(z) \\ &\geq (1 - \alpha)(\gamma + \langle h, z \rangle) + \alpha(f(x) + \langle g(x), z - x \rangle) \\ &= (1 - \alpha)\gamma + \alpha(f(x) - \langle g(x), x \rangle) + \langle (1 - \alpha)h + \alpha g(x), z \rangle \\ &= \bar{\gamma} + \langle \bar{h}, z \rangle. \end{aligned}$$

For appropriate choices of x and α , this may give much improved error bounds. We discuss suitable choices for x later.

Step size selection. The **step size parameter** α controls the fraction of the new information (16) incorporated into the new relaxation. It is chosen with the hope for a reduction factor of approximately $1 - \alpha$ in the current error factor η , and must therefore be adapted to the actual progress made.

First we note that in practice, $Q(\hat{x})$ is unknown; hence the numerical value of η is meaningless in itself. However, quotients of η at different iterations have a meaning, quantifying the amount of progress made.

In the following, we use bars to denote quantities tentatively modified in the current iteration, but they replace the current values of these quantities only if an acceptance criterion is met that we now motivate. We measure progress in terms of the quantity

$$R := \frac{\eta - \bar{\eta}}{\lambda \alpha \eta}, \tag{17}$$

where $\lambda \in]0, 1[$ is a fixed number. A value $R \geq 1$ indicates that we made sufficient progress in that

$$\bar{\eta} = (1 - \lambda R \alpha) \eta \tag{18}$$

was reduced at least by a fraction λ of the designed improvement of η by $\alpha \eta$; thus the step size is acceptable or may even be increased if $R > 1$. On the other hand, if $R < 1$, the step size must be reduced significantly to improve the chance of reaching the design goal. Introducing a maximal step size $\alpha_{\max} \in]0, 1[$ and two parameters with $0 < \kappa' \leq \kappa$ to control the amount of increase or decrease in α , we update the step size according to

$$\bar{\alpha} := \begin{cases} \alpha e^{-\kappa} & \text{if } R < 1, \\ \min(\alpha e^{\kappa'(R-1)}, \alpha_{\max}) & \text{if } R \geq 1. \end{cases} \tag{19}$$

Updating the linear relaxation and u makes sense only when η was improved. This suggests the following update scheme, in which α is always modified, while h, γ, η , and u are changed only when $\bar{\eta} < \eta$; if this is not the case, the barred quantities are simply discarded.

3.1 Algorithm. (Update scheme)

global tuning parameters: $\lambda \in]0, 1[$; $\alpha_{\max} \in]0, 1[$; $\kappa > 0$;

$\kappa' \in]0, \kappa]$;

input: $\alpha, \eta, \bar{h}, \bar{\gamma}, \bar{\eta}, \bar{u}$;

output: $\alpha, h, \gamma, \eta, u$;

$R = (\eta - \bar{\eta}) / (\lambda \alpha \eta)$;

if $R < 1$, $\bar{\alpha} = \alpha e^{-\kappa}$;

else $\bar{\alpha} = \min(\alpha e^{\kappa'(R-1)}, \alpha_{\max})$;

end;

$\alpha = \bar{\alpha}$;

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

$h = \bar{h}$; $\gamma = \bar{\gamma}$; $\eta = \bar{\eta}$; $u = \bar{u}$;

end;

Parameters that work well in practice [1] are $\lambda = 0.9$, $\kappa = \kappa' = 0.5$, $\alpha_{\max} = 0.7$.

If α_{\min} denotes the smallest actually occurring step size (which is not known in advance), we have global linear convergence with a convergence factor of $1 - e^{-\kappa} \alpha_{\min}$. However, α_{\min} and hence this global rate of convergence may depend on the target tolerance ε ; thus the convergence speed in the limit $\varepsilon \rightarrow 0$ may be linear or sublinear depending on the properties of the specific function minimized.

Strongly convex relaxations. If f is strongly convex, we may know a number $\mu > 0$ such that $f - \mu Q$ is still convex. In this case, we have in place of (16) the stronger inequality

$$f(z) - \mu Q(z) \geq f(x) - \mu Q(x) + \langle g(x) - \mu g_Q(x), z - x \rangle \quad \text{for } x, z \in C. \quad (20)$$

In the following, we only assume that $\mu \geq 0$, thus covering the case of linear relaxations, too.

(20) allows us to construct strongly convex relaxations of the form

$$f(z) \geq \gamma + \langle h, z \rangle + \mu Q(z) \quad \text{for all } z \in C. \quad (21)$$

For example, (21) always holds with

$$h = g(x_b) - \mu g_Q(x_b), \quad \gamma = f(x_b) - \mu Q(x_b) - \langle h, x_b \rangle.$$

Again more general relaxations of the form (21) are found by accumulating past information.

3.2 Proposition. *Suppose that $x \in C$, $\alpha \in [0, 1]$, and let*

$$\bar{h} = h + \alpha(g - h), \quad \bar{\gamma} = \gamma + \alpha \left(f(x) - \mu Q(x) - \langle g, x \rangle - \gamma \right),$$

where

$$g = g(x) - \mu g_Q(x).$$

If (21) holds and $f - \mu Q$ is convex then (21) also holds with $\bar{\gamma}$ and \bar{h} in place of γ and h .

Proof. By (20) and the assumptions,

$$\begin{aligned}
f(z) - \mu Q(z) &= (1 - \alpha)(f(z) - \mu Q(z)) + \alpha(f(z) - \mu Q(z)) \\
&\geq (1 - \alpha)(\gamma + \langle h, z \rangle) \\
&\quad + \alpha \left(f(x) - \mu Q(x) + \langle g(x), z - x \rangle - \mu \langle g_Q(x), z - x \rangle \right) \\
&= \bar{\gamma} + \langle \bar{h}, z \rangle.
\end{aligned}$$

□

The relaxations (21) lead to the following error bound.

3.3 Proposition. *Let*

$$\gamma_b := \gamma - f(x_b), \quad \eta := E(\gamma_b, h) - \mu.$$

Then (21) implies

$$0 \leq f(x_b) - \hat{f} \leq \eta Q(\hat{x}). \quad (22)$$

Proof. By definition of $E(\gamma_b, h) = \eta + \mu$ and (21), we have

$$-(\eta + \mu)Q(z) \leq \gamma_b + \langle h, z \rangle = \gamma - f(x_b) + \langle h, z \rangle \leq f(z) - f(x_b) - \mu Q(z).$$

for all $z \in C$. Substituting $z = \hat{x}$ gives (22). □

Note that for $\mu = 0$, we simply recover the previous results for general convex functions.

An optimal subgradient algorithm. For a nonsmooth convex function, the subgradient at a point does not always determine a direction of descent. However, we may hope to find better points by moving from the best point x_b into the direction of the point (6) used to determine our error bound. We formulate on this basis the following algorithm, for which optimal complexity bounds will be proved in Section 5.

Note that the strong convexity parameter μ needs to be specified to use the algorithm. If μ is unknown, one may always put $\mu = 0$ (ignoring possible strong convexity), at the cost of possibly slower worst case asymptotic convergence. (Techniques like those used in JUDITSKY & NESTEROV [19] or GONZAGA & KARAS [16] for choosing μ adaptively can probably be applied to the above algorithm to remove the dependence on having to know μ . However, [19] requires an explicit knowledge of a Lipschitz constant for the gradient, while [16] proves only sublinear convergence. It is not yet clear how to modify OSGA to avoid both problems simultaneously.)

3.4 Algorithm. (Optimal subgradient algorithm, OSGA)

global tuning parameters: $\lambda \in]0, 1[$; $\alpha_{\max} \in]0, 1[$; $\kappa > 0$;

$\kappa' \in]0, \kappa]$;

input parameters: $\mu \geq 0$; $\varepsilon > 0$; f_{target} ;

output: x_b ;

assumptions: $f - \mu Q$ is convex;

begin

choose $x_b \in C$; stop if $f(x_b) \leq f_{\text{target}}$;

$h = g(x_b) - \mu g_Q(x_b)$; $\gamma = f(x_b) - \mu Q(x_b) - \langle h, x_b \rangle$;

$\gamma_b = \gamma - f(x_b)$; $u = U(\gamma_b, h)$; $\eta = E(\gamma_b, h) - \mu$;

$\alpha = \alpha_{\max}$;

while 1,

$x = x_b + \alpha(u - x_b)$; $g = g(x) - \mu g_Q(x)$;

$\bar{h} = h + \alpha(g - h)$; $\bar{\gamma} = \gamma + \alpha(f(x) - \mu Q(x) - \langle g, x \rangle - \gamma)$;

$x'_b = \operatorname{argmin}_{z \in \{x_b, x\}} f(z)$;

$\gamma'_b = \bar{\gamma} - f(x'_b)$; $u' = U(\gamma'_b, \bar{h})$; $x' = x_b + \alpha(u' - x_b)$;

choose \bar{x}_b with $f(\bar{x}_b) \leq \min(f(x'_b), f(x'))$;

$\bar{\gamma}_b = \bar{\gamma} - f(\bar{x}_b)$; $\bar{u} = U(\bar{\gamma}_b, \bar{h})$; $\bar{\eta} = E(\bar{\gamma}_b, \bar{h}) - \mu$;

$x_b = \bar{x}_b$;

stop if some user-defined test is passed;

update $\alpha, h, \gamma, \eta, u$ by Algorithm 3.1;

end;

end;

The analysis of the algorithm will be independent of the choice of \bar{x}_b allowed in Algorithm 3.4. The simplest choice is

$$\bar{x}_b = \operatorname{argmin}_{z \in \{x'_b, x'\}} f(z).$$

If the best function value $f(x_b)$ is stored and updated, each iteration then requires the computation of two function values $f(x)$ and $f(x')$ and one subgradient $g(x)$.

However, the algorithm allows the incorporation of heuristics to look for improved function values before deciding on the choice of \bar{x}_b . This may involve additional function evaluations at points selected by a line search procedure (see, e.g., BECK & TEBoulLE [11]), a bundle optimization (see, e.g., LAN [20]), or a local quadratic approximation (see, e.g., YU et al. [34]).

For numerical results see the remarks at the end of the introduction.

4 Inequalities for the error factor

The possibility to get worst case complexity bounds rests on the establishment of a strong upper bound on the error factor η . This bound depends on global information about the

function f ; while not necessary for executing the algorithm itself, it is needed for the analysis. Depending on the properties of f , global information of different strength can be used, resulting in inequalities of corresponding strength. The key steps in the analysis rely on the lower bound for the term $\gamma + \langle h, z \rangle$ derived in Proposition 2.2.

4.1 Theorem. *In Algorithm 3.4, the error factors are related by*

$$\bar{\eta} - (1 - \alpha)\eta \leq \frac{\alpha^2 \|g(x)\|_*^2}{2(1 - \alpha)(\eta + \mu)Q_0}, \quad (23)$$

where $\|\cdot\|_*$ denotes the norm dual to $\|\cdot\|$.

Proof. We first establish some inequalities needed for the later estimation. By convexity of Q and the definition of \bar{h} ,

$$\begin{aligned} \alpha\mu \left(Q(\bar{u}) - Q(x) + \langle g_Q(x), x \rangle \right) &\geq \alpha\mu \langle g_Q(x), \bar{u} \rangle = \langle h - \bar{h} + \alpha(g(x) - h), \bar{u} \rangle \\ &= (1 - \alpha)\langle h, \bar{u} \rangle + \langle \alpha g(x) - \bar{h}, \bar{u} \rangle. \end{aligned}$$

By definition of x , we have

$$(1 - \alpha)(x_b - x) = -\alpha(u - x).$$

Hence (20) (with $\mu = 0$) implies

$$(1 - \alpha)(f(x_b) - f(x)) \geq (1 - \alpha)\langle g(x), x_b - x \rangle = -\alpha\langle g(x), u - x \rangle.$$

By definition of $\bar{\gamma}$, we conclude from these two inequalities that

$$\begin{aligned} \bar{\gamma} - f(x) + \alpha\mu Q(\bar{u}) &= (1 - \alpha)(\gamma - f(x)) - \alpha\langle g(x), x \rangle + \alpha\mu \left(Q(\bar{u}) - Q(x) + \langle g_Q(x), x \rangle \right) \\ &\geq (1 - \alpha) \left(\gamma - f(x) + \langle h, \bar{u} \rangle \right) + \alpha\langle g(x), \bar{u} - x \rangle - \langle \bar{h}, \bar{u} \rangle \\ &\geq (1 - \alpha) \left(\gamma - f(x_b) + \langle h, \bar{u} \rangle \right) + \alpha\langle g(x), \bar{u} - u \rangle - \langle \bar{h}, \bar{u} \rangle. \end{aligned}$$

Using this, (9) (with $\bar{\gamma}_b = \bar{\gamma} - f(\bar{x}_b)$ in place of γ and \bar{h} in place of h), and $E(\bar{\gamma}_b, \bar{h}) = \bar{\eta} + \mu$ gives

$$\begin{aligned} (\bar{\eta} + \mu - \alpha\mu)Q(\bar{u}) &= f(\bar{x}_b) - \bar{\gamma} - \langle \bar{h}, \bar{u} \rangle - \alpha\mu Q(\bar{u}) \\ &\leq f(\bar{x}_b) - f(x) - \alpha\langle g(x), \bar{u} - u \rangle - (1 - \alpha) \left(\gamma - f(x_b) + \langle h, \bar{u} \rangle \right). \end{aligned} \quad (24)$$

Since $E(\gamma_b, h) > 0$ by Proposition 2.1, we may use (11) with $\gamma_b = \gamma - f(x_b)$ in place of γ and $\eta + \mu = E(\gamma_b, h)$, and find

$$(\eta + \mu)Q(\bar{u}) \geq f(x_b) - \gamma - \langle h, \bar{u} \rangle + \frac{\eta + \mu}{2} \|\bar{u} - u\|^2. \quad (25)$$

Now (24) and (25) imply

$$\begin{aligned}
(\bar{\eta} - (1 - \alpha)\eta)Q(\bar{u}) &= (\bar{\eta} + \mu - \alpha\mu)Q(\bar{u}) - (1 - \alpha)(\eta + \mu)Q(\bar{u}) \\
&\leq f(\bar{x}_b) - f(x) - (1 - \alpha)\left(\gamma - f(x_b) + \langle h, \bar{u} \rangle\right) \\
&\quad - \alpha\langle g(x), \bar{u} - u \rangle \\
&\quad - (1 - \alpha)\left(f(x_b) - \gamma - \langle h, \bar{u} \rangle + \frac{\eta + \mu}{2}\|\bar{u} - u\|^2\right) \\
&= f(\bar{x}_b) - f(x) + \bar{S},
\end{aligned}$$

where

$$\begin{aligned}
\bar{S} &:= -\alpha\langle g(x), \bar{u} - u \rangle - \frac{(1 - \alpha)(\eta + \mu)}{2}\|\bar{u} - u\|^2 \\
&\leq \alpha\|g(x)\|_*\|\bar{u} - u\| - \frac{(1 - \alpha)(\eta + \mu)}{2}\|\bar{u} - u\|^2 \\
&= \frac{\alpha^2\|g(x)\|_*^2 - (\alpha\|g(x)\|_* + (1 - \alpha)(\eta + \mu)\|\bar{u} - u\|)^2}{2(1 - \alpha)(\eta + \mu)} \leq \frac{\alpha^2\|g(x)\|_*^2}{2(1 - \alpha)(\eta + \mu)}.
\end{aligned} \tag{26}$$

If $\bar{\eta} \leq (1 - \alpha)\eta$ then (23) holds trivially. Thus we assume that $\bar{\eta} > (1 - \alpha)\eta$. Then

$$(\bar{\eta} - (1 - \alpha)\eta)Q_0 \leq (\bar{\eta} - (1 - \alpha)\eta)Q(\bar{u}) \leq f(\bar{x}_b) - f(x) + \bar{S}. \tag{27}$$

Since $f(\bar{x}_b) \leq f(x)$, we conclude again that (23) holds. Thus (23) holds generally. \square

Note that the arguments used in this proof did not make use of x' ; thus (23) even holds when one sets $x' = x$ in the algorithm, saving some work.

4.2 Theorem. *If f has Lipschitz continuous gradients with Lipschitz constant L then, in Algorithm 3.4,*

$$\bar{\eta} > (1 - \alpha)\eta \quad \Rightarrow \quad (1 - \alpha)(\eta + \mu) < \alpha^2 L. \tag{28}$$

Proof. The proof follows the general line of the preceding proof, but now we must consider the information provided by x' .

Since E is monotone decreasing in its first argument and $f(x'_b) \geq f(\bar{x}_b)$, the hypothesis of (28) implies that

$$\eta' := E(\bar{\gamma} - f(x'_b), \bar{h}) - \mu \geq E(\bar{\gamma} - f(\bar{x}_b), \bar{h}) - \mu = \bar{\eta} > (1 - \alpha)\eta.$$

By convexity of Q and the definition of \bar{h} ,

$$\begin{aligned}
\alpha\mu\left(Q(u') - Q(x) + \langle g_Q(x), x \rangle\right) &\geq \alpha\mu\langle g_Q(x), u' \rangle = \langle h - \bar{h} + \alpha(g(x) - h), u' \rangle \\
&= (1 - \alpha)\langle h, u' \rangle + \langle \alpha g(x) - \bar{h}, u' \rangle.
\end{aligned}$$

By definition of x , we have

$$(1 - \alpha)(x_b - x) = -\alpha(u - x).$$

Hence (20) (with $\mu = 0$) implies

$$(1 - \alpha)(f(x_b) - f(x)) \geq (1 - \alpha)\langle g(x), x_b - x \rangle = -\alpha\langle g(x), u - x \rangle.$$

By definition of $\bar{\gamma}$, we conclude from the last two inequalities that

$$\begin{aligned} \bar{\gamma} - f(x) + \alpha\mu Q(u') &= (1 - \alpha)(\gamma - f(x)) - \alpha\langle g(x), x \rangle + \alpha\mu\left(Q(u') - Q(x) + \langle g_Q(x), x \rangle\right) \\ &\geq (1 - \alpha)\left(\gamma - f(x) + \langle h, u' \rangle\right) + \alpha\langle g(x), u' - x \rangle - \langle \bar{h}, u' \rangle \\ &\geq (1 - \alpha)\left(\gamma - f(x_b) + \langle h, u' \rangle\right) + \alpha\langle g(x), u' - u \rangle - \langle \bar{h}, u' \rangle. \end{aligned}$$

Using this, (9) (with $\gamma'_b = \bar{\gamma} - f(x'_b)$ in place of γ and \bar{h} in place of h), and $E(\gamma'_b, \bar{h}) = \eta' + \mu$ gives

$$\begin{aligned} (\eta' + \mu - \alpha\mu)Q(u') &= f(x'_b) - \bar{\gamma} - \langle \bar{h}, u' \rangle - \alpha\mu Q(u') \\ &\leq f(x'_b) - f(x) - \alpha\langle g(x), u' - u \rangle \\ &\quad - (1 - \alpha)\left(\gamma - f(x'_b) + \langle h, u' \rangle\right). \end{aligned} \tag{29}$$

Using (11) (with $\gamma_b = \gamma - f(x_b)$ in place of γ) and $\eta + \mu = E(\gamma_b, h)$, we find

$$(\eta + \mu)Q(u') \geq f(x_b) - \gamma - \langle h, u' \rangle + \frac{\eta + \mu}{2}\|u' - u\|^2. \tag{30}$$

Now (29) and (30) imply

$$\begin{aligned} (\eta' - (1 - \alpha)\eta)Q(u') &= (\eta' + \mu - \alpha\mu)Q(u') - (1 - \alpha)(\eta + \mu)Q(u') \\ &\leq f(x'_b) - f(x) - (1 - \alpha)\left(\gamma - f(x_b) + \langle h, u' \rangle\right) \\ &\quad - \alpha\langle g(x), u' - u \rangle \\ &\quad - (1 - \alpha)\left(f(x_b) - \gamma - \langle h, u' \rangle + \frac{\eta + \mu}{2}\|u' - u\|^2\right) \\ &= f(x'_b) - f(x) + S', \end{aligned}$$

where

$$S' := -\alpha\langle g(x), u' - u \rangle - \frac{(1 - \alpha)(\eta + \mu)}{2}\|u' - u\|^2,$$

giving

$$(\eta' - (1 - \alpha)\eta)Q_0 \leq f(x'_b) - f(x) + S'.$$

Now

$$\begin{aligned} f(x'_b) &\leq f(x') \leq f(x) + \langle g(x), x' - x \rangle + \frac{L}{2}\|x' - x\|^2 \\ &= f(x) + \alpha\langle g(x), u' - u \rangle + \frac{\alpha^2 L}{2}\|u' - u\|^2, \end{aligned} \tag{31}$$

so that under the hypothesis of (28)

$$0 < (\eta' - (1 - \alpha)\eta)Q_0 \leq \frac{\alpha^2 L - (1 - \alpha)(\eta + \mu)}{2}\|u' - u\|^2.$$

Thus $\alpha^2 L - (1 - \alpha)(\eta + \mu) > 0$, and the conclusion of (28) holds. \square

5 Bounds for the number of iterations

We now use the inequalities from Theorem 4.1 and Theorem 4.2 to derive bounds for the number of iterations. The weakest global assumption, mere convexity, leads to the weakest bounds and guarantees sublinear convergence only, while the strongest global assumption, strong convexity and Lipschitz continuous gradients, leads to the strongest bounds guaranteeing R -linear convergence. Our main result shows that, asymptotically as $\varepsilon \rightarrow 0$, the number of iterations needed by the OSGA algorithm matches the lower bounds on the complexity derived by NEMIROVSKI & YUDIN [23], apart from constant factors:

5.1 Theorem. *Suppose that $f - \mu Q$ is convex. Then:*

(i) (**Nonsmooth complexity bound**)

If the points generated by Algorithm 3.4 stay in a bounded region of the interior of C , or if f is Lipschitz continuous in C , the total number of iterations needed to reach a point with $f(x) \leq f(\hat{x}) + \varepsilon$ is at most $O((\varepsilon^2 + \mu\varepsilon)^{-1})$. Thus the asymptotic worst case complexity is $O(\varepsilon^{-2})$ when $\mu = 0$ and $O(\varepsilon^{-1})$ when $\mu > 0$.

(ii) (**Smooth complexity bound**)

If f has Lipschitz continuous gradients with Lipschitz constant L , the total number of iterations needed by Algorithm 3.4 to reach a point with $f(x) \leq f(\hat{x}) + \varepsilon$ is at most $O(\varepsilon^{-1/2})$ if $\mu = 0$, and at most $O(|\log \varepsilon| \sqrt{L/\mu})$ if $\mu > 0$.

In particular, if f is strongly convex and differentiable with Lipschitz continuous gradients, $\mu > 0$ holds with arbitrary quadratic prox functions, and we get a complexity bound similar to that achieved by the preconditioned conjugate gradient method for linear systems; cf. AXELSSON & LINDSKOG [8].

Note that (28) generalizes to other situations by replacing (31) with a weaker smoothness property of the form

$$f(z) \leq f(x) + \langle g(x), z - x \rangle + \phi(\|z - x\|) \quad (32)$$

with ϕ convex and monotone increasing. For example, this holds with $\phi(t) = L_1 t$ if f has subgradients with bounded variation, and with $\phi(t) = L_s t^{s+1}$ if f has Hölder continuous gradients with exponent $s \in]0, 1[$, and with linear combinations thereof in the composite case considered by LAN [20]. Imitating the analysis below of the two cases stated in the theorem then gives corresponding complexity bounds matching those obtained by Lan.

Theorem 5.1 follows from the two propositions below covering the different cases, giving in each case explicit upper bounds on the number $K_\mu(\alpha, \eta)$ of further iterations needed to complete the algorithm from a point where the values of α and η given as arguments of K_μ were achieved. We write α_0 and η_0 for the initial values of α and η . Only the dependence on μ , α , and η is made explicit.

5.2 Proposition. *Suppose that the dual norm of the subgradients $g(x)$ encountered during the iteration remains bounded by the constant c_0 . Define*

$$c_1 := \frac{c_0^2}{2Q_0}, \quad c_2 := \max \left(\frac{e^\kappa c_1}{(1-\lambda)(1-\alpha_{\max})}, \frac{\eta_0(\eta_0 + \mu)}{\alpha_0} \right), \quad c_3 = \frac{c_2}{2\lambda}.$$

(i) In each iteration,

$$\eta(\eta + \mu) \leq \alpha c_2. \quad (33)$$

(ii) The algorithm stops after at most

$$K_\mu(\alpha, \eta) := 1 + \kappa^{-1} \log \frac{c_2 \alpha}{\varepsilon(\varepsilon + \mu)} + \frac{c_3}{\varepsilon(\varepsilon + \mu)} - \frac{c_3}{\eta(\eta + \mu)} \quad (34)$$

further iterations.

In particular, (i) and (ii) hold when the iterates stay in a bounded region of the interior of C , or when f is Lipschitz continuous in C .

Note that any convex function is Lipschitz continuous in any closed and bounded domain inside the interior of its support. Hence if the iterates stay in a bounded region R of the interior of C , $\|g\|$ is bounded by the Lipschitz constant of f in the closure of the region R .

Proof. (i) Condition (33) holds initially, and is preserved in each update unless α is reduced. But then $R < 1$, hence $\bar{\eta} \geq (1 - \lambda)\eta$. Combining this with the upper bound on $\bar{\eta}$ from Theorem 4.1 gives

$$(1 - \lambda)\alpha\eta \leq \bar{\eta} - (1 - \alpha)\eta \leq \frac{\alpha^2 c_1}{(1 - \alpha)(\eta + \mu)}.$$

This implies

$$(1 - \lambda)(1 - \alpha)\eta(\eta + \mu) \leq \alpha c_1.$$

Since $\lambda < 1$ and $\alpha \leq e^\kappa \bar{\alpha}$ by (19), we conclude that

$$\bar{\eta}(\bar{\eta} + \mu) \leq \eta(\eta + \mu) \leq \frac{\alpha c_1}{(1 - \lambda)(1 - \alpha)} \leq \bar{\alpha} c_2.$$

Thus (33) holds with $\bar{\eta}$ and $\bar{\alpha}$ in place of η and α , and hence always.

(ii) As the algorithm stops once $\eta \leq \varepsilon$, (33) implies that in each iteration $c_2 \alpha \geq \varepsilon(\varepsilon + \mu)$. As α is reduced only when $R < 1$, and then by a fixed factor $e^{-\kappa}$, this cannot happen more than $\kappa^{-1} \log \frac{c_2 \alpha}{\varepsilon(\varepsilon + \mu)}$ times in turn. Thus after some number of α -reductions we must always have another step with $R \geq 1$. By (18), this gives a reduction of η by a factor of at least $1 - \lambda\alpha$. But this implies that the stopping criterion $\eta \leq \varepsilon$ is eventually reached. Therefore the algorithm stops eventually. Since $R \geq 0$, (19) implies $\bar{\alpha} \leq \alpha e^{\kappa(R-1)}$. Therefore

$$\log(\alpha/\bar{\alpha}) \geq \kappa(1 - R). \quad (35)$$

Now (35), (18), and (33) imply

$$\begin{aligned}
K_\mu(\alpha, \eta) - K_\mu(\bar{\alpha}, \bar{\eta}) &= \frac{\log(\alpha/\bar{\alpha})}{\kappa} + \frac{c_3}{\bar{\eta}(\bar{\eta} + \mu)} - \frac{c_3}{\eta(\eta + \mu)} \\
&\geq 1 - R + \frac{c_3}{\bar{\eta}(\bar{\eta} + \mu)} - \frac{c_3}{\eta(\eta + \mu)} \\
&= 1 - R + \frac{c_3}{(1 - \lambda R\alpha)\eta((1 - \lambda R\alpha)\eta + \mu)} - \frac{c_3}{\eta(\eta + \mu)} \\
&= 1 - R + c_3 \frac{(\eta + \mu) - (1 - \lambda R\alpha)((1 - \lambda R\alpha)\eta + \mu)}{(1 - \lambda R\alpha)\eta(\eta + \mu)((1 - \lambda R\alpha)\eta + \mu)} \\
&= 1 - R + c_3 \frac{\lambda R\alpha((2 - \lambda R\alpha)\eta + \mu)}{(1 - \lambda R\alpha)\eta(\eta + \mu)((1 - \lambda R\alpha)\eta + \mu)} \\
&\geq 1 - R + c_3 \frac{2\lambda R\alpha}{\eta(\eta + \mu)} = 1 - R + \frac{2c_3\lambda R}{c_2} = 1.
\end{aligned}$$

This implies the complexity bound by reverse induction, since immediately before the last iteration, $c_2\alpha \geq \varepsilon(\varepsilon + \mu)$ and $\eta > \varepsilon$, hence $K_\mu(\alpha, \eta) \geq 1$. \square

5.3 Proposition. *Suppose that f has Lipschitz continuous gradients with Lipschitz constant L , and put*

$$c_4 = \max\left(\frac{\eta_0 + \mu}{\alpha_0^2}, \frac{e^{2\kappa}L}{1 - \alpha_{\max}}\right), \quad c_5 = \frac{4c_4}{\lambda^2}, \quad c_6 = \sqrt{\frac{c_4}{\mu}}, \quad c_7 = \frac{c_6}{\lambda}.$$

(i) *In each iteration*

$$\eta + \mu \leq \alpha^2 c_4, \tag{36}$$

(ii) *The algorithm stops after at most $K_\mu(\alpha, \eta)$ further iterations. Here*

$$K_0(\alpha, \eta) := 1 + \kappa^{-1} \log\left(\alpha \sqrt{\frac{c_4}{\varepsilon}}\right) + \sqrt{\frac{c_5}{\varepsilon}} - \sqrt{\frac{c_5}{\eta}}, \tag{37}$$

$$K_\mu(\alpha, \eta) := 1 + \frac{\log(c_6\alpha)}{\kappa} + c_7 \log \frac{\eta}{\varepsilon} \quad \text{for } \mu > 0. \tag{38}$$

Proof. (i) (36) holds initially, and is preserved in each update unless α is reduced. But then $R < 1$, hence by Theorem 4.2, $(1 - \alpha)(\eta + \mu) = \alpha^2 L$ before the reduction. Therefore

$$\bar{\eta} + \mu \leq \eta + \mu \leq \frac{\alpha^2 L}{1 - \alpha} \leq \frac{\alpha^2 L}{1 - \alpha_{\max}} \leq \alpha^2 e^{-2\kappa} c_4 \leq \bar{\alpha}^2 c_4.$$

Thus (36) holds again after the reduction, and hence always. As in the previous proof, we find that the algorithm stops eventually, and (35) holds.

(ii) If $\mu = 0$ then (18), (35), and (36) imply

$$\begin{aligned} K_0(\alpha, \eta) - K_0(\bar{\alpha}, \bar{\eta}) &= \frac{\log(\alpha/\bar{\alpha})}{\kappa} + \sqrt{\frac{c_5}{(1 - \lambda R\alpha)\eta}} - \sqrt{\frac{c_5}{\bar{\eta}}} \\ &\geq 1 - R + \left(1 - \sqrt{1 - \lambda R\alpha}\right) \sqrt{\frac{c_5}{(1 - \lambda R\alpha)\eta}} \\ &\geq 1 - R + \frac{\lambda R\alpha}{2} \sqrt{\frac{c_5}{\eta}} \geq 1 - R + \frac{\lambda R}{2} \sqrt{\frac{c_5}{c_4}} = 1. \end{aligned}$$

This implies the complexity bound by reverse induction, since immediately before the last iteration, $\alpha \geq \sqrt{\varepsilon/c_4}$, hence $K_0(\alpha, \eta) \geq 1$.

(iii) If $\mu > 0$ then (36) shows that always $c_6\alpha \geq 1$, hence $\bar{\eta} = (1 - \lambda R\alpha)\eta \leq (1 - R/c_7)\eta$. Therefore

$$K_\mu(\alpha, \eta) - K_\mu(\bar{\alpha}, \bar{\eta}) = \frac{\log(\alpha/\bar{\alpha})}{\kappa} + c_7 \log \frac{\eta}{\bar{\eta}} \geq 1 - R + c_7 \log \frac{1}{1 - R/c_7} \geq 1,$$

and the result follows as before. \square

Proof of Theorem 5.1.

(i) We apply Proposition 5.2(ii) to the first iteration, and note that $K_0(\alpha, \eta) = O(e^{-2})$ and $K_\mu(\alpha, \eta) = O(e^{-1})$ if $\mu > 0$.

(i) We apply Proposition 5.3(ii) to the first iteration, and note that $K_0(\alpha, \eta) = O(e^{-1/2})$ and $K_\mu(\alpha, \eta) = O(\log \varepsilon^{-1})$ if $\mu > 0$.

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 and A. Neumaier, High-dimensional convex optimization via optimal affine subgradient algorithms, in ROKS workshop (2013), 83–84.
- [3] M. Ahookhosh and A. Neumaier, An optimal subgradient algorithm with subspace search for costly convex optimization problems, Submitted, (2015) http://www.optimization-online.org/DB_FILE/2015/04/4852.pdf
- [4] M. Ahookhosh and A. Neumaier, Solving nonsmooth convex optimization with complexity $O(\varepsilon^{-1/2})$, http://www.optimization-online.org/DB_FILE/2015/05/4900.pdf
- [5] M. Ahookhosh and A. Neumaier, An optimal subgradient algorithms for large-scale bound-constrained convex optimization, Submitted, (2015) <http://arxiv.org/abs/1501.01497>

- [6] M. Ahookhosh and A. Neumaier, An optimal subgradient algorithms for large-scale convex optimization in simple domains, Submitted, (2015) <http://arxiv.org/abs/1501.01451>
- [7] A. Auslender and M. Teboulle, Interior gradient and proximal methods for convex and conic optimization, *SIAM J. Optimization* 16 (2006), 697–725.
- [8] O. Axelsson and G. Lindskog, On the rate of convergence of the conjugate gradient method, *Numer. Math.* 48 (1986), 499–523.
- [9] N.S. Aybat and G. Iyengar, A first-order augmented Lagrangian method for compressed sensing, *SIAM J. Optim.*, 22(2) (2012), 429–459.
- [10] A. Beck, A. Ben-Tal, N. Guttman–Beck and L. Tetruashvili, The CoMirror algorithm for solving nonsmooth constrained convex problems, *Operations Research Letters* 38 (2010), 493–498.
- [11] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.* 2 (2009), 183–202.
- [12] S.R. Becker, E.J. Candès, and M.C. Grant, Templates for convex cone problems with applications to sparse signal recovery, *Math. Programming Comput.* 3 (2011), 165–218.
- [13] J. Chen and S. Burer, A first-order smoothing technique for a class of large-scale linear programs, *SIAM J. Optim.* 24 (2014), 598–620.
- [14] O. Devolder, F. Glineur, and Y. Nesterov, First-order methods of smooth convex optimization with inexact oracle, *Math. Programming* 146 (2014), 37–75.
- [15] K. Fountoulakis, J. Gondzio and P. Zhlobich, Matrix-free interior point method for compressed sensing problems, *Math. Programming Comput.* 6 (2014), 1–31.
- [16] C.C. Gonzaga and E.W. Karas, Fine tuning Nesterov’s steepest descent algorithm for differentiable convex programming, *Math. Programming* 138 (2013), 141–166.
- [17] C.C. Gonzaga, E.W. Karas, and D.R. Rossetto, An optimal algorithm for constrained differentiable convex optimization, *SIAM J. Optim.* 23 (2013), 1939–1955.
- [18] M. Gu, L.-H. Lim and C.J. Wu, PARNES: A rapidly convergent algorithm for accurate recovery of sparse and approximately sparse signals, *Numer. Algor.* 64 (2013), 321–347.
- [19] A. Juditsky and Y. Nesterov, Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization, *Stochastic Systems*, 4(1) (2014), 44–80.
- [20] G. Lan, Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization, *Mathematical Programming*, (2013) DOI 10.1007/s10107-013-0737-x.
- [21] G. Lan, Z. Lu and R.D.C. Monteiro, Primal-dual first-order methods with $O(1/\varepsilon)$ iteration-complexity for cone programming, *Math. Programming* 126 (2011), 1–29.
- [22] X. Meng and H. Chen, Accelerating Nesterov’s method for strongly convex functions with Lipschitz gradient, *Arxiv preprint arXiv:1109.6058* (2011).

- [23] A.S. Nemirovsky and D.B. Yudin, Problem complexity and method efficiency in optimization, Wiley, New York 1983.
- [24] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$ (in Russian), Doklady AN SSSR 269 (1983), 543-547. Engl. translation: Soviet Math. Dokl. 27 (1983), 372–376.
- [25] Y. Nesterov, Introductory lectures on convex optimization: A basic course, Kluwer, Dordrecht 2004.
- [26] Y. Nesterov, Smooth minimization of non-smooth functions, Math. Programming 103 (2005), 127–152.
- [27] Y. Nesterov, Rounding of convex sets and efficient gradient methods for linear programming problems, Optim. Methods Softw. 23 (2008), 109–128.
- [28] Y. Nesterov, Unconstrained convex minimization in relative scale, Math. Operations Res. 34 (2009), 180–193.
- [29] Y. Nesterov, Primal-dual subgradient methods for convex problems, Math. Programming 120 (2009), 221–259.
- [30] Y. Nesterov, Gradient methods for minimizing composite objective function, Math. Programming, 140 (2013), pp. 125–161.
- [31] Y. Nesterov, Universal gradient methods for convex optimization problems, Math. Programming, (2014), DOI 10.1007/s10107-014-0790-0.
- [32] P. Richtarik, Improved algorithms for convex minimization in relative scale, SIAM J. Optimization 21 (2011), 1141–1167.
- [33] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, Technical report, Math. Dept., Univ. of Washington (2008).
<http://pages.cs.wisc.edu/~brecht/cs726docs/Tseng.APG.pdf>
- [34] J. Yu, S.V.N. Vishvanathan, S. Günter and N.N. Schraudolph, A quasi-Newton approach to nonsmooth convex optimization problems in machine learning, J. Machine Learning Res. 11 (2010), 1145–1200.