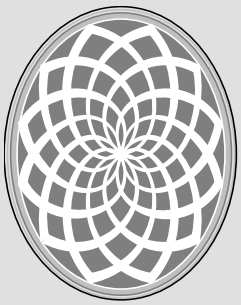

Interdisciplinary Mathematics Institute

2014:01

Convex optimization on Banach Spaces

R.A. DeVore and V.N. Temlyakov

IMI

Preprint Series

College of Arts and Sciences
University of South Carolina

Convex optimization on Banach Spaces

R.A. DeVore and V.N. Temlyakov*

January 1, 2014

Abstract

Greedy algorithms which use only function evaluations are applied to convex optimization in a general Banach space X. Along with algorithms that use exact evaluations, algorithms with approximate evaluations are treated. A priori upper bounds for the convergence rate of the proposed algorithms are given. These bounds depend on the smoothness of the objective function and the sparsity or compressibility (with respect to a given dictionary) of a point in X where the minimum is attained.

1 Introduction

Convex optimization is an important and well studied subject of numerical analysis. The canonical setting for such problems is to find the minimum of a convex function E over a domain in $\mathbb{R}^d$. Various numerical algorithms have been developed for minimization problems and a priori bounds for their performance have been proven. We refer the reader to [1], [9], [10], [11] for the core results in this area.

In this paper, we are concerned with the more general setting where E is defined on a domain D in a general Banach space X with norm $\|\cdot\| = \|\cdot\|_X$. Thus, our main interest is in approximating

$$E^* := \inf_{x\in D} E(x). \tag{1.1}$$

Problems of this type occur in many important application domains, such as statistical estimation and learning, optimal control, and shape optimization. Another important motivation for studying such general problems, even for finite dimensional spaces X, is that when the dimension d of X is large, we would like to obtain bounds on the convergence rate of a proposed algorithm that are independent of this dimension.

Solving (1.1) is an example of a high dimensional problem and is known to suffer the curse of dimensionality without additional assumptions on E which serve to reduce its dimensionality. These additional assumptions take the form of smoothness restrictions on E and assumptions which imply that the minimum in (1.1) is attained on a subset of D with additional structure. Typical assumptions for the latter involve notions of sparsity or compressibility, which are by now heavily employed concepts for high dimensional problems. We will always assume that there is a point

*This research was supported by the Office of Naval Research Contracts ONR-N00014-08-1-1113, ONR N00014-09-1-0107; the NSF Grants DMS 0915231 and DMS-1160841. This research was initiated when the second author was a visiting researcher at TAMU

$x^* \in D$ where the minimum E^* is attained, $E(x^*) = E^*$. We do not assume x^* is unique. The set $D^* = D^*(E) \subset D$ of all points where the minima is attained is convex.

The algorithms studied in this paper utilize dictionaries $\mathcal{D}$ of X. A set of elements $\mathcal{D} \subset X$, whose closed linear span coincides with X is called a *symmetric dictionary* if $\|g\| := \|g\|_X = 1$, for all $g \in \mathcal{D}$, and in addition $g \in \mathcal{D}$ implies $-g \in \mathcal{D}$. The simplest example of a dictionary is $\mathcal{D} = \{\pm\varphi_j\}_{j\in\Gamma}$ where $\{\varphi_j\}_{j\in\Gamma}$ is a Schauder basis for X. In particular for $X = \mathbb{R}^d$, one can take the canonical basis $\{e_j\}_{j=1}^d$.

Given such a dictionary $\mathcal{D}$, there are several types of domains D that are employed in applications. Sometimes, these domains are the natural domain of the physical problem. Other time these are constraints imposed on the minimization problem to ameliorate high dimensionality. We mention the following three common settings.

Sparsity Constraints: *The set $\Sigma_n(\mathcal{D})$ of functions*

$$g = \sum_{g\in\Lambda} c_g g, \quad \#(\Lambda) = n, \tag{1.2}$$

is called the set of sparse functions of order n with respect to the dictionary $\mathcal{D}$. One common assumption is to minimize E on the domain $D = \Sigma_n(\mathcal{D})$, i.e. to look for an n sparse minimizer of (1.1).

ℓ_1 constraints: *A more general setting is to minimize E over the closure $A_1(\mathcal{D})$ (in X) of the convex hull of $\mathcal{D}$. A slightly more general setting is to minimize E over one of the sets*

$$\mathcal{L}_M := \{g \in X : \ g/M \in A_1(\mathcal{D})\}. \tag{1.3}$$

Sometimes M is allowed to vary as in model selection or regularization algorithms from statistics. This is often referred to as ℓ_1 minimization.

Unconstrained optimization: *Imposed constraints, such as sparsity or assuming $D = A_1(\mathcal{D})$, are sometimes artificial and may not reflect the original optimization problem. We consider therefore the unconstrained minimization where $D = X$. We always make the assumption that the minimum of E is actually assumed. Therefore, there is a point $x^* \in X$ where*

$$E^* = E(x^*). \tag{1.4}$$

We do not require that x^* is unique. Notice that in this case the minimum E^* is attained on the set

$$D_0 := \{x \in X : \ E(x) \le E(0)\}. \tag{1.5}$$

In what follows, we refer to minimization over D_0 to be the unconstrained minimization problem.

A typical greedy optimization algorithm builds approximations to E^* of the form $E(G_m)$, $m = 1, 2 \dots$ where the elements G_m are built recursively using the dictionary $\mathcal{D}$ and typically are in $\Sigma_m(\mathcal{D})$. We will always assume that the initial point G_0 is chosen as the 0 element. Given that G_m has been defined, one first searches for a direction $\varphi_m \in \mathcal{D}$ for which $E(G_m + \alpha\varphi_m)$ decreases significantly as α moves away from zero. Once, φ_m is chosen, then one selects $G_{m+1} = G_m + \alpha_m\varphi_m$ or more generally $G_{m+1} = \alpha'_m G_m + \alpha_m\varphi_m$, using some recipe for choosing α_m or more generally α_m, α'_m. Algorithms of this type are referred to as *greedy algorithms* and will be the object of study in this paper.

There are different strategies for choosing φ_m and α_m, α'_m (see, for instance, [20], [13], [2], [3], [6], [4], [8], [19], and [7]). One possibility to choose φ_m is to use the Fréchet derivative $E'(G_{m-1})$ of E to choose a steepest descent direction. This approach has been amply studied and various convergence results for steepest descent algorithms have been proven, even for the general Banach space setting. We refer the reader to the papers [20, 17, 18] which are representative of the convergence results known in this case. The selection of α_m, α'_m is commonly referred to as relaxation and is well studied in numerical analysis, although the Banach space setting needs additional attention.

Our interest in the present paper are greedy algorithms that do not utilize E'. They are preferred since E' is not given to us and therefore, in numerical implementations, must typically be approximated at any given step of the algorithm. We will analyze several different algorithms of this type which are distinguished from one another by how G_{m+1} is gotten from G_m both in the selection of φ_m and the parameters α_m, α'_m. Our algorithms are built with ideas similar to the analogous, well-studied, greedy algorithms for approximation of a given element $f \in X$. We refer the reader to [16] for a comprehensive description of greedy approximation algorithms.

In this introduction, we limit ourselves to two of the main algorithms studied in this paper. The first of these, which we call the Relaxed E-Greedy Algorithm (REGA(co)) was introduced in [20] under the name sequential greedy approximation.

Relaxed E-Greedy Algorithm (REGA(co)): We define $G_0 := 0$. For $m \geq 1$, assuming G_{m-1} has already been defined, we take $\varphi_m \in \mathcal{D}$ and $0 \leq \lambda_m \leq 1$ such that

$$E((1-\lambda_m)G_{m-1} + \lambda_m \varphi_m) = \inf_{0 \leq \lambda \leq 1; g \in \mathcal{D}} E((1-\lambda)G_{m-1} + \lambda g)$$

and define

$$G_m := (1-\lambda_m)G_{m-1} + \lambda_m \varphi_m.$$

We assume that there exist such minimizing φ_m and λ_m.

We note that the REGA(co) is a modification of the classical Frank-Wolfe algorithm [5]. For convenience, we have assumed the existence of a minimizing φ_m and λ_m. However, we also analyze algorithms with only approximate implementation which avoids this assumption.

Observe that this algorithm is in a sense built for $A_1(\mathcal{D})$ because each G_m is obviously in $A_1(\mathcal{D})$. The next algorithm, called the E-Greedy Algorithm with Free Relaxation (EGAFR(co)), makes some modifications in the relaxation step that will allow it to be applied to the more general unconstrained minimization problem on D_0.

E-Greedy Algorithm with Free Relaxation (EGAFR(co)). We define $G_0 := 0$. For $m \geq 1$, assuming G_{m-1} has already been defined, we take $\varphi_m \in \mathcal{D}$, $\alpha_m, \beta_m \in \mathbb{R}$ satisfying (assuming existence)

$$E(\alpha_m G_{m-1} + \beta m \varphi_m) = \inf_{\alpha, \beta \in \mathbb{R}; g \in \mathcal{D}} E(\alpha G_{m-1} + \beta g)$$

and define

$$G_m := \alpha_m G_{m-1} + \beta_m \varphi_m.$$

It is easy to see that each of these algorithms has the following monotonicity

$$E(G_0) \geq E(G_1) \geq E(G_2) \geq \cdots.$$

Our main goal in this paper is to understand what can be said a priori about the convergence rate of a specific greedy optimization algorithm of the above form. Such results are built on two assumptions: (i) the smoothness of E, (ii) assumptions that the minimum is attained at a point x^* satisfying a constraint such as the sparsity or ℓ_1 constraint. In what follows to measure the smoothness of E, we introduce the modulus of smoothness

$$\rho(E,u) := \rho(E,S,u) := \frac{1}{2}\sup_{x\in S,\|y\|=1} |E(x+uy)+E(x-uy)-2E(x)|, \tag{1.6}$$

of E on any given set S. We say that E is uniformly smooth on S if $\rho(E,S,u)/u \to 0$ as $u \to 0$.

The following theorem for REGA(co) is a prototype of the results proved in this paper.

Theorem 1.1 *Let* $E^* := \inf_{x\in A_1(\mathcal{D})} E(x)$.

(i) *If E is uniformly smooth on $A_1(\mathcal{D})$, then the REGA(co) converges:*

$$\lim_{m\to\infty} E(G_m) = E^*. \tag{1.7}$$

(ii) *If in addition, $\rho(E, A_1(\mathcal{D}), u) \le \gamma u^q$, $1 < q \le 2$, then*

$$E(G_m) - E^* \le C(q,\gamma) m^{1-q}, \tag{1.8}$$

with a positive constant $C(q,\gamma)$ which depends only on q and γ.

The case $q = 2$ of this theorem was proved in [20]. We prove this theorem in §2.

As we have already noted, the EGAFR(co) is designed to solve the unconstrained minimization problem where the domain $D = X$. The performance of this algorithm will depend not only on the smoothness of E but also on the compressibility of a point $x^* \in D^*$ where E takes its minimum. To quantify this compressibility, we introduce

$$A(\epsilon) := A(E,\epsilon) := \inf\{M : \exists y \in \mathcal{L}_M \text{ such that } \ E(y) - E^* \le \epsilon\}. \tag{1.9}$$

An equivalent way to quantify this compressibility is the error

$$e(E,M) := \inf_{y\in\mathcal{L}_M} E(y) - E^*. \tag{1.10}$$

Notice that the functions A and e are pseudo-inverses of one another.

The following theorem states the convergence properties of the EGAFR(co).

Theorem 1.2 *Let E be uniformly smooth on X and let* $E^* := \inf_{x\in X} E(x) = \inf_{x\in D_0} E(x)$.

(i) *The EGAFR(co) converges:*

$$\lim_{m\to\infty} E(G_m) = \inf_{x\in X} E(x) = \inf_{x\in D_0} E(x) = E^*.$$

(ii) *If the modulus of smoothness of E satisfies $\rho(E,u) \le \gamma u^q$, $1 < q \le 2$, then, the EGAFR(co) satisfies*

$$E(G_m) - E^* \le C(E,q,\gamma)\epsilon_m, \tag{1.11}$$

where

$$\epsilon_m := \inf\{\epsilon : \ A(\varepsilon)^q m^{1-q} \le \epsilon\}. \tag{1.12}$$

In particular, if for some $r > 0$, we have $e(E,M) \le \tilde{\gamma} M^{-r}$, $M \ge 1$, then

$$E(G_m) - E^* \le C(E,q,\gamma,\tilde{\gamma},r) m^{\frac{1-q}{1+q/r}}. \tag{1.13}$$

We note that the EGAFR(co) is a modification of the Weak Greedy Algorithm with Free Relaxation (WGAFR(co)) studied in [17]. Also note that if $x^* \in \mathcal{L}_M$ then the estimate in Theorem 1.2 reads

$$E(G_m) - E^* \le C(E, q, \gamma) M^q m^{1-q}. \tag{1.14}$$

We show in the following section how Theorem 1.1 and Theorem 1.2 are easily proven using existing results for greedy algorithms. We also introduce and analyze another greedy algorithm for convex minimization.

The most important results of the present paper are in Section 3 and are motivated by numerical considerations. Very often we cannot calculate the values of E exactly. Even if we can evaluate E exactly, we may not be able to find the exact value of, say, the quantity

$$\inf_{0\le\lambda\le 1; g\in\mathcal{D}} E((1-\lambda)G_{m-1} + \lambda g)$$

in the REGA(co). This motivates us to study in §3 various modifications of the above algorithms. For example, the following algorithm, which is an approximate variant of the REGA(co), was introduced in [20].

Relaxed E-Greedy Algorithm with error δ (REGA(δ)). Let $\delta \in (0, 1]$. We define $G_0 := 0$. Then, for each $m \ge 1$ we have the following inductive definition: We take any $\varphi_m \in \mathcal{D}$ and $0 \le \lambda_m \le 1$ satisfying

$$E((1-\lambda_m)G_{m-1} + \lambda_m\varphi_m) \le \inf_{0\le\lambda\le 1; g\in\mathcal{D}} E((1-\lambda)G_{m-1} + \lambda g) + \delta$$

and define

$$G_m := (1-\lambda_m)G_{m-1} + \lambda_m\varphi_m.$$

In Section 3, we give modifications of this type to the above algorithms and then prove convergence results for these modifications. For example, the following convergence result is proven for the REGA(δ).

Theorem 1.3 *Let E be a uniformly smooth on $A_1(\mathcal{D})$ convex function with modulus of smoothness $\rho(E, u) \le \gamma u^q$, $1 < q \le 2$. Then, for the REGA(δ) we have*

$$E(G_m) - E^* \le C(q, \gamma, E, c) m^{1-q}, \quad m \le \delta^{-1/q},$$

where $E^* := \inf_{f\in A_1(\mathcal{D})} E(x)$.

In the case $q = 2$ Theorem 1.3 was proved in [20].

In the REGA(co) and the REGA(δ) we solve the univariate convex optimization problem with respect to λ

$$\inf_{0\le\lambda\le 1} E((1-\lambda)G_{m-1} + \lambda g) \tag{1.15}$$

respectively exactly and with an error δ. It is well known (see [10]) that there are fast algorithms to solve problem (1.15) approximately. We discuss some of them in §4.

In the EGAFR(co) and the EGAFR(δ) (see Section 3 for this algorithm) we solve the convex optimization problem for a function on two variables

$$\inf_{\lambda,w} E((1-w)G_{m-1} + \lambda g) \tag{1.16}$$

respectively exactly and with an error δ. We describe in §5 how univariate optimization algorithms can be used for approximate solution of (1.16).

2 Analysis of greedy algorithms

We begin this section by showing how to prove the results for REGA(co) and EGAFR(co) stated in the introduction, namely Theorems 1.1 and 1.2. The proof of convergence results for greedy algorithms typically is done by establishing a recursive inequality for the error $E(G_n) - E^*$. To analyze the decay of this sequence of errors, will need the following lemma.

Lemma 2.1 *If a sequence a_m, $m \geq 0$, of nonnegative numbers satisfies*

$$a_m \leq a_{m-1}(1 - ca_{m-1}^p), \quad m \geq 1, \tag{2.1}$$

with $c > 0$ and $p > 0$. Then

$$a_n \leq Cn^{-1/p}, \quad n \geq 1, \tag{2.2}$$

with the constant C depending only on p and c.

Proof: In the case $p \geq 1$ which is used in this paper this follows from Lemma 2.16 of [16]. In the case $p \geq 1$ Lemma 2.1 was often used in greedy approximation in Banach spaces (see [16], Chapter 6). For the general case $p > 0$ see Lemma 4.2 of [12]). □

To establish a recursive inequality for the error in REGA(co), we will use the following lemma about REGA(co).

Lemma 2.2 *Let E be a uniformly smooth convex function with modulus of smoothness $\rho(E, u)$. Then, for any $f \in A_1(\mathcal{D})$ and the iterations G_m of the REGA(co), we have*

$$E(G_m) \leq E(G_{m-1}) + \inf_{0 \leq \lambda \leq 1}(-\lambda(E(G_{m-1}) - E(f)) + 2\rho(E, 2\lambda)), \quad m = 1, 2, \ldots. \tag{2.3}$$

Proof: A similar result was proved in Lemma 3.1 of [17] for a different greedy algorithm denoted by WRGA(co) in [17]. In order to distinguish the two algorithms, we denote by $\bar{G}_m$ the output of WRGA(co). The relaxation step in WRGA(co) is exactly the same as in our REGA(co). However the choice of direction $\bar{\varphi}_m$ in WRGA(co) was based on a maximal gradient descent. This means that at each step the $\bar{G}_{m-1}$ is also possibly different than our G_{m-1} of REGA(co). However, an examination of the proof of Lemma 3.1 shows that it did not matter what $\bar{G}_{m-1}$ is as long as it is in $\Sigma_{m-1}(\mathcal{D})$. So Lemma 3.1 holds for our G_{m-1} and if we let $\tilde{G}_m$ denote the result of applying WRGA(co) to our G_{m-1}, then we have

$$E(G_m) \leq E(\tilde{G}_m) \leq E(G_{m-1}) + \inf_{0 \leq \lambda \leq 1}(-\lambda(E(G_{m-1}) - E(f)) + 2\rho(E, 2\lambda)). \tag{2.4}$$

Here, the first inequality is because REGA(co) minimizes error over all choices of directions φ from the dictionary and all choices of the relaxation parameter and thereby is at least as good as the choice from WRGA(co). The last inequality is from Lemma 3.1 of [17]. Thus, we have proven the lemma. □

Proof of Theorem 1.1: The proof of this theorem is similar to the proof of Theorem 3.1 and Theorem 3.2 in [17]. We illustrate the proof of (1.8). If we denote by $a_m := E(G_m) - E^*$, then subtracting E^* from both sides of (2.3) gives the recursive inequality

$$a_m \leq a_{m-1} + \inf_{0 \leq \lambda \leq 1}\{-\lambda a_{m-1} + 2\gamma\lambda^q\}. \tag{2.5}$$

If we choose λ to satisfy

$$\lambda a_{m-1} = 4\gamma(2\lambda)^q \tag{2.6}$$

provided it is not greater than 1 and choose 1 otherwise and use this value in (2.5), we obtain in case $\lambda \leq 1$

$$a_m \leq a_{m-1}(1 - ca_{m-1}^{\frac{1}{q-1}}), \tag{2.7}$$

with $c > 0$ a constant depending only on γ and q. This recursive inequality then gives the decay announced in Theorem 1.1 because of Lemma 2.1. The case $\lambda = 1$ can be treated as in the proof of Theorem 3.2 from [17]. □

Proof of Theorem 1.2: This proof is derived from results in [17] in a similar way to how we have proved Theorem 1.1 for REGA(co). An algorithm, called WGAFR(co), was introduced in [17] which differs from EGAFR(co) only in how each φ_m is chosen. One then uses the analysis in WGAFR(co)

The above discussed algorithms REGA(co) and EGAFR(co) provide sparse approximate solutions to the corresponding optimization problems. These approximate solutions are sparse with respect to the given dictionary $\mathcal{D}$ but they are not obtained as an expansion with respect to $\mathcal{D}$. This means that at each iteration of these algorithms we update all the coefficients of sparse approximants. Sometimes it is important to build an approximant in the form of expansion with respect to $\mathcal{D}$. The reader can find a discussion of greedy expansions in [16], Section 6.7. For comparison with the algorithms we have already introduced, we recall a greedy-type algorithm for unconstrained optimization which uses only function values and builds sparse approximants in the form of expansion that was introduced and analyzed in [18]. Let $\mathcal{C} := \{c_m\}_{m=1}^{\infty}$ be a fixed sequence of positive numbers.

***E*-Greedy Algorithm with coefficients $\mathcal{C}$ (EGA($\mathcal{C}$)).** We define $G_0 := 0$. Then, for each $m \geq 1$ we have the following inductive definition:

(i) Let $\varphi_m \in \mathcal{D}$ be such that (assuming existence)

$$E(G_{m-1} + c_m\varphi_m) = \inf_{g\in\mathcal{D}} E(G_{m-1} + c_m g).$$

(ii) Then define

$$G_m := G_{m-1} + c_m\varphi_m.$$

In the above definition, we can restrict ourselves to positive numbers because of the symmetry of the dictionary $\mathcal{D}$.

For the analysis of this algorithm, we will assume that the sets

$$D_C := \{x : E(x) \leq E(0) + C\}$$

are bounded for all finite C. We recall two results for the EGA($\mathcal{C}$) that were proved in [18] .

Theorem 2.3 *Let $\mu(u) = o(u)$ as $u \to 0$ and let E be a uniformly smooth convex function satisfying*

$$E(x + uy) - E(x) - u\langle E'(x), y\rangle \leq 2\mu(u), \tag{2.8}$$

for $x \in D_2$, $\|y\| = 1$, $|u| \le 1$. Assume that the coefficients sequence $\mathcal{C} := \{c_j\}$, $c_j \in [0,1]$ satisfies the conditions

$$\sum_{k=1}^{\infty} c_k = \infty, \tag{2.9}$$

$$\sum_{k=1}^{\infty} \mu(c_k) \le 1. \tag{2.10}$$

Then, for each dictionary $\mathcal{D}$, the EGA($\mathcal{C}$) satisfies

$$\lim_{m\to\infty} E(G_m) = \inf_{x\in X} E(x) =: E^*.$$

Theorem 2.4 *Let E be a uniformly smooth convex function with modulus of smoothness $\rho(E,u) \le \gamma u^q$, $q \in (1,2]$ on D_2. We set $s := \frac{2}{1+q}$ and $\mathcal{C}_s := \{ck^{-s}\}_{k=1}^{\infty}$ with c chosen in such a way that $\gamma c^q \sum_{k=1}^{\infty} k^{-sq} \le 1$. Then the EGA($\mathcal{C}_s$) converges with the following rate: for any $r \in (0, 1-s)$*

$$E(G_m) - \inf_{x\in A_1(\mathcal{D})} E(x) \le C(r,q,\gamma) m^{-r}.$$

Let us now turn to a brief comparison of the above algorithms and their known convergence rates. The REGA(co) is designed for solving optimization problems on domains $D \subset A_1(\mathcal{D})$ and requires that $D^* \cap A_1(\mathcal{D}) \neq \emptyset$. The EGAFR(co) is not limited to the $A_1(\mathcal{D})$ but applies for any optimization domain as long as E achieves its minimum on a bounded domain. As we have noted earlier, if there is a point $D^* \cap A_1(\mathcal{D}) \neq \emptyset$, then EGAFR(co) provides the same convergence rate ($O(m^{1-q})$) as REGA(co). Thus, EGAFR(co) is more robust and requires the solution of only a slightly more involved minimization at each iteration.

The advantage of EGA($\mathcal{C}$) is that it solves a simpler minimization problem at each iteration since the relaxation parameters are set in advance. However, it requires knowledge of the smoothness order q of E and also gives a poorer rate of convergence than REGA(co) and the EGAFR(co).

To continue this discussion let us consider the very special case where $X = \ell_p^d$ and the dictionary $\mathcal{D}$ is finite, say $\mathcal{D} = \{g_j\}_{j=1}^N$. In such a case, the existence of φ_m in all the above algorithms is easily proven. The EGA($\mathcal{C}$) simply uses Nm function evaluations to make m iterations. The REGA(co) solves a one-dimensional optimization problem at each iteration for each dictionary element, thus N such problems. We discuss this problem in Section 4 and show that each such problem can be solved with exponential accuracy with respect to the number of evaluations needed from E.

3 Approximate greedy algorithms for convex optimization

We turn now to the main topic of this paper which is modifications of the above greedy algorithms to allow imprecise calculations or less strenuous choices for descent directions and relaxation parameters. We begin with a discussion of the Weak Relaxed Greedy Algorithm WRGA(co) which was introduced and analyzed in [17] and which we already referred to in §2 . The WRGA(co) uses the gradient to choose a steepest descent direction at each iteration. The interesting aspect of WRGA(co), relative to imprecise calculations, is that it uses a weakness parameter $t_m < 1$ to allow some relative error in estimating $\sup_{g\in\mathcal{D}} \langle -E'(G_{m-1}), g - G_{m-1}\rangle$. Here and below we use a convenient bracket notation: for a functional $F \in X^*$ and an element $f \in X$ we write $F(f) = \langle F, f\rangle$. We concentrate on a modification of the second step of WRGA(co). Very often we cannot calculate

values of E exactly. Even in case we can evaluate E exactly we may not be able to find the exact value of the $\inf_{0\le\lambda\le1} E((1-\lambda)G_{m-1}+\lambda\varphi_m)$. This motivates us to study the following modification of the WRGA(co).

Weak Relaxed Greedy Algorithm with error δ (WRGA(δ)). Let $\delta \in (0,1]$. We define $G_0 := 0$. Then, for each $m \ge 1$ we have the following inductive definition.

(1) $\varphi_m := \varphi_m^{\delta,\tau} \in \mathcal{D}$ is taken any element satisfying

$$\langle -E'(G_{m-1}), \varphi_m - G_{m-1}\rangle \ge t_m \sup_{g\in\mathcal{D}} \langle -E'(G_{m-1}), g - G_{m-1}\rangle.$$

(2) Then $0 \le \lambda_m \le 1$ is chosen as any number such that

$$E((1-\lambda_m)G_{m-1}+\lambda_m\varphi_m) \le \inf_{0\le\lambda\le1} E((1-\lambda)G_{m-1}+\lambda\varphi_m) + \delta.$$

With these choices, we define

$$G_m := (1-\lambda_m)G_{m-1}+\lambda_m\varphi_m.$$

Thus, this algorithm differs from the REGA(δ) given in the introduction, only in the choice of the direction φ_m at each step. Both of these algorithms are directed at solving the minimization of E over $A_1(\mathcal{D})$. The following theorem analyzes the WRGA(δ).

Theorem 3.1 *Let E be uniformly smooth on $A_1(\mathcal{D})$ whose modulus of smoothness $\rho(E,u)$ satisfies*

$$\rho(E,u) \le \gamma u^q, \quad 1 < q \le 2. \tag{3.1}$$

If $t_k = t$, $k = 1,2,\dots$, then the WRGA(δ) satisfies

$$E(G_m) - E^* \le C(q,\gamma,t,E)m^{1-q}, \quad m \le \delta^{-1/q}, \tag{3.2}$$

where $E^* := \inf_{f\in A_1(\mathcal{D})} E(x)$.

We develop next some results which will be used to prove this theorem. Let us first note that when E is Fréchet differentiable, the convexity of E implies that for any x, y

$$E(y) \ge E(x) + \langle E'(x), y - x\rangle \tag{3.3}$$

or, in other words,

$$E(x) - E(y) \le \langle E'(x), x - y\rangle = \langle -E'(x), y - x\rangle. \tag{3.4}$$

The following simple lemma holds.

Lemma 3.2 *Let E be Fréchet differentiable convex function. Then the following inequality holds for $x \in S$*

$$0 \le E(x+uy) - E(x) - u\langle E'(x), y\rangle \le 2\rho(E, u\|y\|). \tag{3.5}$$

We use these remarks to prove the following.

Lemma 3.3 *Let E be uniformly smooth on $A_1(\mathcal{D})$ with modulus of smoothness $\rho(E,u)$. Then, for any $f \in A_1(\mathcal{D})$ we have that the WRGA(δ) satisfies*

$$E(G_m) \le E(G_{m-1}) + \inf_{0\le\lambda\le 1}(-\lambda t_m(E(G_{m-1}) - E(f)) + 2\rho(E,2\lambda)) + \delta, \quad m = 1,2,\dots$$

and therefore

$$E(G_m) - E^* \le E(G_{m-1}) - E^* + \inf_{0\le\lambda\le 1}(-\lambda t_m(E(G_{m-1}) - E^*) + 2\rho(E,2\lambda)) + \delta, \quad m = 1,2,\dots \tag{3.6}$$

where $E^* := \inf_{f\in A_1(\mathcal{D})} E(x)$.

Proof: We have

$$G_m := (1-\lambda_m)G_{m-1} + \lambda_m\varphi_m = G_{m-1} + \lambda_m(\varphi_m - G_{m-1})$$

and from the definition of λ_m,

$$E(G_m) \le \inf_{0\le\lambda\le 1} E(G_{m-1} + \lambda(\varphi_m - G_{m-1})) + \delta.$$

By Lemma 3.2 we have for any λ

$$E(G_{m-1} + \lambda(\varphi_m - G_{m-1}))$$

$$\le E(G_{m-1}) - \lambda\langle -E'(G_{m-1}), \varphi_m - G_{m-1}\rangle + 2\rho(E,2\lambda) \tag{3.7}$$

and by step (1) in the definition of the WRGA(δ) and Lemma 2.2 from [17] (see also Lemma 6.10, p. 343 of [16]) we get

$$\langle -E'(G_{m-1}), \varphi_m - G_{m-1}\rangle \ge t_m \sup_{g\in\mathcal{D}}\langle -E'(G_{m-1}), g - G_{m-1}\rangle =$$

$$t_m \sup_{\phi\in A_1(\mathcal{D})}\langle -E'(G_{m-1}), \phi - G_{m-1}\rangle \ge t_m\langle -E'(G_{m-1}), f - G_{m-1}\rangle.$$

From (3.4), we obtain

$$\langle -E'(G_{m-1}), f - G_{m-1}\rangle \ge E(G_{m-1}) - E(f).$$

Thus,

$$E(G_m) \le \inf_{0\le\lambda\le 1} E(G_{m-1} + \lambda(\varphi_m - G_{m-1})) + \delta$$

$$\le E(G_{m-1}) + \inf_{0\le\lambda\le 1}(-\lambda t_m(E(G_{m-1}) - E(f)) + 2\rho(E,2\lambda) + \delta, \tag{3.8}$$

which proves the lemma. □

Finally, for the proof of Theorem 3.1, we will need the following result about sequences.

Lemma 3.4 *If a nonnegative sequence $a_0, a_1, \dots, a_N$ satisfies*

$$a_m \le a_{m-1} + \inf_{0\le\lambda\le 1}(-\lambda v a_{m-1} + B\lambda^q) + \delta, \quad B > 0, \quad \delta \in (0,1], \tag{3.9}$$

for $m \le N := [\delta^{-1/q}]$, $q \in (1,2]$, then

$$a_m \le C(q,v,B,a_0)m^{1-q}, \quad m \le N. \tag{3.10}$$

Proof: By taking $\lambda = 0$, (3.9) implies that

$$a_m \le a_{m-1} + \delta, \quad m \le N. \tag{3.11}$$

Therefore, for all $m \le N$ we have

$$a_m \le a_0 + N\delta \le a_0 + 1, \quad 0 \le m \le N.$$

Now fix any value of $m \in [0, N]$ and define $\lambda_1 := \left(\frac{va_{m-1}}{2B}\right)^{\frac{1}{q-1}}$, so that

$$\lambda_1 v a_{m-1} = 2B\lambda_1^q. \tag{3.12}$$

If $\lambda_1 \le 1$ then

$$\inf_{0\le\lambda\le 1}(-\lambda v a_{m-1} + B\lambda^q) \le -\lambda_1 v a_{m-1} + B\lambda_1^q$$

$$= -\frac{1}{2}\lambda_1 v a_{m-1} = -C_1(q, v, B)a_{m-1}^p, \quad p := \frac{q}{q-1}.$$

If $\lambda_1 > 1$ then for all $\lambda \le \lambda_1$ we have $\lambda v a_{m-1} > 2B\lambda^q$ and specifying $\lambda = 1$ we get

$$\inf_{0\le\lambda\le 1}(-\lambda v a_{m-1} + B\lambda^q) \le -\frac{1}{2} v a_{m-1}$$

$$\le -\frac{1}{2} v a_{m-1}^p (a_0 + 1)^{1-p} = -C_1(q, v, a_0)a_{m-1}^p.$$

Thus, in any case, setting $C_2 := C_2(q, v, B, a_0) := \min(C_1(q, v, B), C_1(q, v, a_0))$ we obtain from (3.9)

$$a_m \le a_{m-1} - C_2 a_{m-1}^p + \delta, \tag{3.13}$$

holds for all $0 \le m \le N$.

Now to establish (3.10), we let $n \in [0, N]$ be the smallest integer such that

$$C_2 a_{n-1}^p \le 2\delta. \tag{3.14}$$

If there is no such n, we set $n = N$. In view of (3.13), we have

$$a_m \le a_{m-1} - (C_2/2)a_{m-1}^p, \quad 1 \le m \le n. \tag{3.15}$$

If we modify the sequence a_m by defining it to be zero if $m > n$, then this modified sequence satisfies (3.15) for all m and Lemma 2.1 gives

$$a_m \le C_3 m^{1-q}, \quad 1 \le m \le n, \tag{3.16}$$

with C_3 depending only on C_2 and p.

If $n = N$, we have finished the proof. If $n < N$, then, by (3.11), we obtain for $m \in [n, N]$

$$a_m \le a_{n-1} + (m - n + 1)\delta \le a_{n-1} + N\delta \le a_{n-1} + NN^{-q} \le [\frac{2\delta}{C_2}]^{1/p} + C_4 N^{1-q},$$

where we have used the definition of N. Since $\delta^{1/p} \le N^{-q/p} = N^{-q+1}$, we have

$$a_m \le C_5 N^{1-q} \le C_5 m^{1-q}, \quad n \le m \le N,$$

where C_5 depends only on q, v, B, a_0. This completes the proof of the lemma. □

Proof of Theorem 3.1: We take

$$a_n := E(G_n) - E^* \geq 0.$$

Then, taking into account that $\rho(E,u) \leq \gamma u^q$, we get from Lemma 3.3

$$a_m \leq a_{m-1} + \inf_{0\leq\lambda\leq 1}(-\lambda t a_{m-1} + 2\gamma(2\lambda)^q) + \delta. \tag{3.17}$$

Applying Lemma 3.4 with $v = t$, $B = 2^{1+q}\gamma$ we complete the proof of Theorem 3.1. □

We can establish a similar convergence result for the REGA(δ).

Theorem 3.5 *Let E be a uniformly smooth on $A_1(\mathcal{D})$ convex function with modulus of smoothness $\rho(E,u) \leq \gamma u^q$, $1 < q \leq 2$. Then, for the REGA(δ) we have*

$$E(G_m) - E^* \leq C(q,\gamma,E)m^{1-q}, \quad m \leq \delta^{-1/q},$$

where $E^* := \inf_{f\in A_1(\mathcal{D})} E(x)$.

Proof: From the definition of the REGA(δ), we have

$$E(G_m) \leq \inf_{0\leq\lambda\leq 1; g\in\mathcal{D}} E((1-\lambda)G_{m-1} + \lambda g) + \delta.$$

In the same way that we have proved (2.3), we obtain

$$E(G_m) \leq E(G_{m-1}) + \inf_{0\leq\lambda\leq 1}(-\lambda(E(G_{m-1}) - E^*) + 2\rho(E,2\lambda) + \delta. \tag{3.18}$$

Inequality (3.18) is of the same form as inequality (3.6) from Lemma 3.3. Thus, repeating the above proof of Theorem 3.1 we complete the proof of Theorem 3.5. □

We now introduce and analyze an approximate version of the WGAFR(co).

Weak Greedy Algorithm with Free Relaxation and error δ (WGAFR(δ)). Let $\tau := \{t_m\}_{m=1}^{\infty}$, $t_m \in [0,1]$, be a weakness sequence. We define $G_0 := 0$. Then for each $m \geq 1$ we have the following inductive definition.

(1) $\varphi_m \in \mathcal{D}$ is any element satisfying

$$\langle -E'(G_{m-1}), \varphi_m\rangle \geq t_m \sup_{g\in\mathcal{D}}\langle -E'(G_{m-1}), g\rangle. \tag{3.19}$$

(2) Find w_m and λ_m such that

$$E((1-w_m)G_{m-1} + \lambda_m\varphi_m) \leq \inf_{\lambda,w} E((1-w)G_{m-1} + \lambda\varphi_m) + \delta$$

and define

$$G_m := (1-w_m)G_{m-1} + \lambda_m\varphi_m.$$

Theorem 3.6 *Let E be a uniformly smooth convex function on X with modulus of smoothness $\rho(E, D_1, u) \le \gamma u^q$, $1 < q \le 2$ and let $E^* := \inf_{x\in X} E(x) = \inf_{x\in D_0} E(x)$. Then, for the WGAFR($\delta$), we have*

$$E(G_m) - E^* \le C(E, q, \gamma)\epsilon_m, \quad m \le \delta^{-1/q} \tag{3.20}$$

where

$$\epsilon_m := \inf\{\epsilon : \ A(\varepsilon)^q m^{1-q} \le \epsilon\}. \tag{3.21}$$

and $A(\epsilon)$ is defined by (1.9).

Proof: In the proof of Lemma 4.1 of [17] we established the inequality

$$\inf_{\lambda\ge 0, w} E((1-w)G_{m-1} + \lambda\varphi_m) \le E(G_{m-1})$$

$$+ \inf_{\lambda\ge 0}(-\lambda t_m A(\varepsilon)^{-1}(E(G_{m-1}) - E^*) + 2\rho(E, C_0\lambda)), \quad C_0 = C(D_0), \tag{3.22}$$

under the assumption that φ_m satisfies (3.19) and $G_{m-1} \in D_0$.

In the case of exact evaluations in the WGAFR(co) we had the monotonicity property $E(G_0) \ge E(G_1) \ge \cdots$ which implied that $G_n \in D_0$ for all n. In the case of the WGAFR(δ) inequality (3.22) implies

$$E(G_m) \le E(G_{m-1}) + \delta. \tag{3.23}$$

Therefore, for all $m \le N := [\delta^{-1/q}]$

$$E(G_m) \le E(0) + 1,$$

which implies $G_n \in D_1$ for all $n \le N$.

Denote

$$a_n := E(G_n) - E(f^\varepsilon).$$

Inequality (3.22) implies

$$a_m \le a_{m-1} + \inf_{\lambda\ge 0}(-\lambda t A(\varepsilon)^{-1} a_{m-1} + 2\gamma(C_0\lambda)^q) + \delta.$$

It is similar to (3.17) with the only point that we now cannot guarantee that $a_{m-1} \ge 0$. However, if n is the smallest number from $[1, N]$ such that $a_n < 0$ then for $m \in [n, N]$ (3.23) implies easily $a_m \le Cm^{1-q}$. Thus it is sufficient to assume that $a_n \ge 0$. We apply Lemma 3.4 with $v = tA(\varepsilon)^{-1}$, $B = 2\gamma C_0^q$ and complete the proof. □

We have discussed above two algorithms the WRGA(δ) and the REGA(δ). Results for the REGA(δ) (see Theorem 3.5) were derived from the proof of the corresponding results for the WRGA(δ) (see Theorem 3.1). We now discuss a companion algorithm for the WGAFR(δ) that uses only function evaluations.

***E*-Greedy Algorithm with Free Relaxation and error δ (EGAFR(δ)).** We define $G_0 := 0$. For $m \ge 1$, assuming G_{m-1} has already been defined, we take $\varphi_m \in \mathcal{D}$ $\alpha_m, \beta_m \in \mathbb{R}$ satisfying

$$E(\alpha_m G_{m-1} + \beta m\varphi_m) \le \inf_{\alpha,\beta\in\mathbb{R}; g\in\mathcal{D}} E(\alpha G_{m-1} + \beta g) + \delta$$

and define

$$G_m := \alpha_m G_{m-1} + \beta_m\varphi_m.$$

In the same way as Theorem 3.5 was derived from the proof of Theorem 3.1 one can derive the following theorem from the proof of Theorem 3.6.

Theorem 3.7 *Let E be a uniformly smooth convex function on X with modulus of smoothness $\rho(E, D_1, u) \le \gamma u^q$, $1 < q \le 2$ and let $E^* := \inf_{x \in X} E(x) = \inf_{x \in D_0} E(x)$. Then, for the EGAFR($\delta$), we have*

$$E(G_m) - E^* \le C(E, q, \gamma)\epsilon_m, \quad m \le \delta^{-1/q} \tag{3.24}$$

where

$$\epsilon_m := \inf\{\epsilon : \ A(\varepsilon)^q m^{1-q} \le \epsilon\}. \tag{3.25}$$

and $A(\epsilon)$ is defined by (1.9).

Theorem 2.4 provides the rate of convergence of the EGA($\mathcal{C}$) where we assume that function evaluations are exact and we can find $\inf_{g \in \mathcal{D}}$ exactly. However, in practice we very often cannot evaluate functions exactly and (or) cannot find the exact value of the $\inf_{g \in \mathcal{D}}$. In order to address this issue we modify the EGA($\mathcal{C}$) into the following algorithm EGA($\mathcal{C}, \delta$).

***E*-Greedy Algorithm with coefficients $\mathcal{C}$ and error δ (EGA($\mathcal{C}, \delta$)).** Let $\delta \in (0, 1]$. We define $G_0 := 0$. Then, for each $m \ge 1$ we have the following inductive definition.

(1) $\varphi_m^\delta \in \mathcal{D}$ is such that

$$E(G_{m-1} + c_m \varphi_m^\delta) \le \inf_{g \in \mathcal{D}} E(G_{m-1} + c_m g) + \delta.$$

(2) Let

$$G_m := G_{m-1} + c_m \varphi_m^\delta.$$

We prove an analog of Theorem 2.4 for the EGA($\mathcal{C}, \delta$).

Theorem 3.8 *Let E be a uniformly smooth convex function with modulus of smoothness $\rho(E, u) \le \gamma u^q$, $q \in (1, 2]$ on D_3. We set $s := \frac{2}{1+q}$ and $\mathcal{C}_s := \{ck^{-s}\}_{k=1}^\infty$ with $c \le 1$ chosen in such a way that $\gamma c^q \sum_{k=1}^\infty k^{-sq} \le 1$. Then the EGA($\mathcal{C}_s, \delta$) provides the following rate: for any $r \in (0, 1 - s)$*

$$E(G_m) - E^* \le C(r, q, \gamma) m^{-r}, \qquad m \le \delta^{-\frac{1}{1+r}},$$

where $E^* := \inf_{x \in A_1(\mathcal{D})} E(x)$.

We first accumulate some results that we will use in the proof of this theorem. Let $N := [\delta^{-\frac{1}{1+r}}]$, where $[a]$ is the integer part of a and let G_m, $m \ge 0$ be the sequence generated by the EGA($\mathcal{C}_s, \delta$) .
Claim 1: $G_m \in D_3$, i.e. $E(G_m) \le E(0) + 3$, for all $0 \le m \le N$.

To see this, let $t \in (0, 1)$ and φ_m be such that

$$\langle -E'(G_{m-1}), \varphi_m \rangle \ge t E_{\mathcal{D}}(G_{m-1}), \quad E_{\mathcal{D}}(G) := \sup_{g \in \mathcal{D}} \langle -E'(G), g \rangle. \tag{3.26}$$

Then

$$\inf_{g \in \mathcal{D}} E(G_{m-1} + c_m g) \le E(G_{m-1} + c_m \varphi_m).$$

Thus, it is sufficient to estimate $E(G_{m-1} + c_m\varphi_m)$ with φ_m satisfying (3.26). By (3.5) under assumption that $G_{m-1} \in D_3$ we get with $\mu(u) := \gamma u^q$

$$E(G_{m-1} + c_m\varphi_m) \le E(G_{m-1}) + c_m\langle E'(G_{m-1}), \varphi_m\rangle + 2\mu(c_m).$$

Using the definition of φ_m, we obtain

$$E(G_{m-1} + c_m\varphi_m) \le E(G_{m-1}) - c_m t E_{\mathcal{D}}(G_{m-1}) + 2\mu(c_m). \tag{3.27}$$

We now prove by induction that $G_m \in D_3$ for all $m \le N$. Indeed, clearly $G_0 \in D_3$. Suppose that $G_k \in D_3$, $k = 0, 1, \dots, m-1$, then (3.27) holds for all $k = 1, \dots, m$ instead of m and, therefore,

$$E(G_m) \le E(0) + 2\sum_{k=1}^{m} \mu(c_k) + m\delta \le E(0) + 3,$$

proving the claim.

We also need the following lemma from [18].

Lemma 3.9 *If $f \in \mathcal{L}_A$, then for*

$$G_k := \sum_{j=1}^{k} c_j\varphi_j, \quad \varphi_j \in \mathcal{D}, \quad j = 1, \dots, k,$$

we have

$$E_{\mathcal{D}}(G_k) \ge (E(G_k) - E(f))/(A + A_k), \quad A_k := \sum_{j=1}^{k} |c_j|.$$

Proof of Theorem 3.8: E attains E^* at a point $x^* \in A_1(\mathcal{D})$. If we start with (3.27) and then use the above lemma with $f = x^*$, fact that we obtain

$$E(G_m) \le E(G_{m-1}) - \frac{t c_m (E(G_{m-1}) - E^*)}{1 + A_{m-1}} + 2\gamma c_m^q + \delta. \tag{3.28}$$

The left hand side of (3.28) does not depend on t, therefore the inequality holds with $t = 1$:

$$E(G_m) \le E(G_{m-1}) - \frac{c_m (E(G_{m-1}) - E^*)}{1 + A_{m-1}} + 2\gamma c_m^q + \delta. \tag{3.29}$$

We have

$$A_{m-1} = c\sum_{k=1}^{m-1} k^{-s} \le c(1 + \int_1^m x^{-s}dx) = c(1 + (1-s)^{-1}(m^{1-s} - 1)))$$

and

$$1 + A_{m-1} \le 1 + c(1-s)^{-1}m^{1-s}.$$

Therefore, for $m \ge C_1$ we have with $v := (r + 1 - s)/2$

$$\frac{c_m}{1 + A_{m-1}} \ge \frac{v + 1 - s}{2(m-1)}. \tag{3.30}$$

To conclude the proof, we need the following technical lemma. This lemma is a more general version of Lemma 2.1 from [14] (see also Remark 5.1 in [15] and Lemma 2.37 on p. 106 of [16]).

Lemma 3.10 *Let four positive numbers $\alpha < \beta \le 1$, A, $U \in \mathbb{N}$ be given and let a sequence $\{a_n\}_{n=1}^{\infty}$ have the following properties: $a_1 < A$ and we have for all $n \ge 2$*

$$a_n \le a_{n-1} + A(n-1)^{-\alpha}; \tag{3.31}$$

if for some $\nu \ge U$ we have

$$a_\nu \ge A\nu^{-\alpha}$$

then

$$a_{\nu+1} \le a_\nu(1 - \beta/\nu). \tag{3.32}$$

Then there exists a constant $C = C(\alpha, \beta, A, U)$ such that for all $n = 1, 2, \dots$ we have

$$a_n \le Cn^{-\alpha}.$$

We apply this lemma with $a_n := E(G_n) - E^*$, $n \le N$, $a_n := 0$, $n > N$, $\alpha := r$, $\beta := v := (r+1-s)/2$, $U = C_1$ and A specified later. Let us check the conditions (3.31) and (3.32) of Lemma 3.10. It is sufficient to check these conditions for $m < N$. By the inequality

$$E(G_m) \le E(G_{m-1}) + 2\rho(E, c_m) + \delta \le E(G_{m-1}) + 2\gamma c^q m^{-sq} + \delta$$

the condition (3.31) holds for $A \ge 2\gamma c^q + 1$. Using $sq \ge 1 + r$ we get

$$c_m^q = c^q m^{-sq} \le c^q m^{-1-r}, \qquad \delta \le m^{-1-r}. \tag{3.33}$$

Assume that $a_m \ge Am^{-r}$. Setting A to be big enough to satisfy

$$\delta + 2\gamma c_m^q \le \frac{A(1-s-\beta)}{2m^{1+r}}$$

we obtain from (3.29), (3.30), and (3.33)

$$a_{m+1} \le a_m(1 - \beta/m)$$

provided $a_m \ge Am^{-r}$. Thus (3.32) holds. Applying Lemma 3.10 we get

$$a_m \le C(r, q, \gamma)m^{-r}.$$

This completes the proof of Theorem 3.8. □

4 Univariate convex optimization

The relaxation step in each of the above algorithms involves either a univariate or bivariate optimization of a convex function. The univariate optimization problem called *line search* is well studied in optimization theory (see [10]). The purpose of the remaining two sections of this paper is to show that such problems can be solved efficiently. Results of these two sections are known. We present them here for completeness.

In this section we consider the class F of convex on $[0, 1]$ functions which belong to Lip 1 class with constant 1. We are interested in how many function evaluations are needed in order to find for a given $\epsilon > 0$ and a given $f \in F$ a point $x^\epsilon \in [0, 1]$ such that

$$f(x^\epsilon) \le \min_{x \in [0,1]} f(x) + \epsilon \quad ?$$

We begin with a known upper bound.

Proposition 4.1 *If the algorithm described below is applied to any $f \in F$ and $m \in \mathbb{N}$, then after $3 + 2m$ function evaluations, it produces a point $x_m \in [0, 1]$ such that*

$$f(x_m) \leq \min_{x \in [0,1]} f(x) + 2^{-m}. \tag{4.1}$$

Proof: We begin with three function evaluations $f(0)$, $f(1/2)$, and $f(1)$. Without loss of generality assume that $f(0) \leq f(1)$.
Case 1: $f(0) \leq f(1/2) \leq f(1)$. It follows from convexity that $f(x) \geq f(1/2)$ for all $x \in [1/2, 1]$ and hence we can restrict our search for a point of minimum to the interval $[0, 1/2]$, in other words we delete interval $(1/2, 1]$ from consideration.
Case 2: $f(1/2) < f(0)$**.** We make two more evaluations at the points $1/4$ and $3/4$. It is impossible that both

$$f(1/4) < f(1/2) \quad \text{and} \quad f(3/4) < f(1/2).$$

Therefore, at least one of $f(1/4)$, $f(3/4)$ must be $\geq f(1/2)$. If $f(1/4) \geq f(1/2)$ and $f(3/4) \geq f(1/2)$ in the same way as above we delete intervals $[0, 1/4)$ and $(3/4, 1]$ and continue our search on $[1/4, 3/4]$. If $f(1/4) < f(1/2)$ then we delete $(1/2, 1]$ and if $f(3/4) < f(1/2)$ we delete $[0, 1/2)$.

After one iteration we have added 2 function evaluations and reduced our search for a point of minimum to an interval of length $1/2$ with function values at end points and the middle point known to us. We continue this process to complete the proof of the proposition. □

We next analyze what happens if we do not receive the exact values of f when we query in the above algorithm. We assume that when we query f at a point x, we receive the corrupted value $y(x)$ where $|f(x) - y(x)| \leq \delta$ for each $x \in [0, 1]$. We assume that we know δ.

Proposition 4.2 *Suppose we make function evaluations with an error δ. The algorithm described below applied to $f \in F$ and $m \in \mathbb{N}$ takes $3+2m$ function evaluations and produces a point $x_m \in [0, 1]$ such that*

$$f(x_m) \leq \min_{x \in [0,1]} f(x) + 2^{-m} + (4m + 1)\delta. \tag{4.2}$$

Proof: In the argument that follows, we use the following property of convex functions. For any $0 \leq a < b \leq c < d \leq 1$ we have

$$\frac{f(b) - f(a)}{b - a} \leq \frac{f(d) - f(c)}{d - c}. \tag{4.3}$$

As in the proof of Proposition 4.1 we go by cases. At the first iteration we evaluate our function at $0, 1/2, 1$. Without loss of generality we assume that $y(0) \leq y(1)$.

A. Suppose $y(0) \leq y(1/2)$. Then $f(0) \leq f(1/2) + 2\delta$ and by (4.3) with $a = 0$, $b = 1/2$, $c = 1/2$, $d = x$, $x \in (1/2, 1]$ we obtain

$$f(x) \geq f(1/2) - 2\delta, \quad x \in [1/2, 1].$$

Therefore, restricting our search for a minimum to $[0, 1/2]$ we make an error of at most 2δ.

B. Suppose $y(1/2) < y(0)$. In this case, we make an additional evaluation of the function at $1/4$.

Ba. Suppose $y(1/4) < y(1/2) - 2\delta$. Then $f(1/4) < f(1/2)$ and by (4.3) we obtain that

$$\min_{x \in [1/2,1]} f(x) \geq \min_{x \in [0,1/2]} f(x).$$

Therefore, we can again restrict our search to the interval $[0, 1/2]$.

Bb. Suppose $y(1/4) \geq y(1/2) - 2\delta$. In this case we make an additional evaluation of the function at $3/4$. If $y(3/4) < y(1/2) - 2\delta$ then as in **Ba** we can restrict our search to the interval $[1/2, 1]$. If $y(3/4) \geq y(1/2) - 2\delta$ we argue as in the case **A** and obtain

$$\min_{x\in[0,1/4]} f(x) \geq \min_{x\in[1/4,1/2]} f(x) - 4\delta, \quad \min_{x\in[3/4,1]} f(x) \geq \min_{x\in[1/2,3/4]} f(x) - 4\delta.$$

Therefore, we restrict our search to the interval $[1/4, 3/4]$ with an error at most 4δ.

At each iteration we add two evaluations and then find that we can restrict our search to an interval of half the size of the original while incurring an additional error at most 4δ. Finally, the evaluation of y gives us an error at most δ with that of f. □

We note that convexity of functions from F plays a dominating role in obtaining exponential decay of error in Proposition 4.1. For instance, the following simple known statement holds for the $\mathrm{Lip}_1 1$ class.

Proposition 4.3 *Let $\mathcal{A}(m)$ denote the class of algorithms (adaptive) which use at most m function evaluations and provide an approximate for the minimum value of a function. Then*

$$\inf_{A\in\mathcal{A}(m)} \sup_{f\in \mathrm{Lip}_1 1} |\min_{x\in[0,1]} f(x) - A(f)| = \frac{1}{4m}.$$

Proof: The upper bound follows from evaluating f at the midpoints x_j of the intervals $[(j-1)/m, j/m]$, $j = 1, \ldots, m$ and giving the approximate value $\min_j f(x_j) - \frac{1}{4m}$. The lower bound follows from the following observation. For any m points $0 \leq \xi_1 < \xi_2 < \ldots < \xi_m \leq 1$ there are two functions $f_1, f_2 \in \mathrm{Lip}_1 1$ such that $f_1(\xi_j) = f_2(\xi_j) = 0$ for all j and $\min_x f_1(x) - \min_x f_2(x) \geq \frac{1}{2m}$. □

5 Multivariate convex optimization

In this section, we discuss an analog of Proposition 4.1 for d-variate convex functions on $[0, 1]^d$. The d-variate algorithm is a coordinate wise application of the algorithm from Proposition 4.1 with an appropriate δ. We begin with a simple lemma.

Lemma 5.1 *Let $f(x)$, $x = (x_1, \ldots, x_d) \in [0, 1]^d$ be a convex on $[0, 1]^d$ function. Define $x^d := (x_1, \ldots, x_{d-1}) \in [0, 1]^{d-1}$ and*

$$f_d(x^d) := \min_{x_d} f(x).$$

Then $f_d(x^d)$ is a convex function on $[0, 1]^{d-1}$.

Proof: Let $u, v \in [0, 1]^{d-1}$. Then, there are two points $w, z \in [0, 1]^d$ such that

$$f_d(u) = f(w), \qquad f_d(v) = f(z)$$

and $u = w^d$, $v = z^d$. From the convexity of $f(x)$, we have

$$f(tw + (1-t)z) \leq tf(w) + (1-t)f(z) = tf_d(u) + (1-t)f_d(v), \quad t \in [0, 1]. \tag{5.1}$$

Clearly,

$$f_d((tw + (1-t)z)^d) \leq f(tw + (1-t)z), \quad t \in [0, 1]. \tag{5.2}$$

Inequalities (5.1) and (5.2) imply that $f_d(u)$ is convex. □

Proposition 5.2 *The d-variate minimization algorithm given below takes as input any $f \in F$ and $m \in \mathbb{N}$ and produces after $(3+2m)^d$ function evaluations a point $x_m \in [0,1]^d$ such that*

$$f(x_m) \leq \min_{x \in [0,1]} f(x) + 2^{-m}(4m+2)^d. \tag{5.3}$$

Proof: We construct the algorithm by induction. In the case $d = 1$, we use the univariate algorithm from Proposition 4.2. Suppose, we have given the algorithm such that the proposition holds for $d-1$. Then, we write

$$\min_x f(x) = \min_{x_d} \min_{x^d} f(x)$$

and observe that by Lemma 5.1 the function $g(x_d) := \min_{x^d} f(x)$ is a convex function. Next, we apply the algorithm from Proposition 4.2 with $\delta = 2^{-m}(4m+2)^{d-1}$ to the function g. By our induction assumption we evaluate g with an error at most δ. Thus by Proposition 4.2 we get an error at most

$$2^{-m} + (4m+1)\delta \leq 2^{-m}(4m+2)^d.$$

The total number of evaluations is $(3+2m)^d$. This completes the proof. □

References

[1] J.M. Borwein and A.S. Lewis, Convex Analysis and Nonlinear Optimization. Theory and Examples, Canadian Mathematical Society, Springer, 2006.

[2] V. Chandrasekaran, B. Recht, P.A. Parrilo, and A.S. Willsky, The convex geometry of linear inverse problems, Proceedings of the 48th Annual Allerton Conference on Communication, Control and Computing, 2010, 699–703.

[3] K.L. Clarkson, Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm, ACM Transactions on Algorithms, **6** (2010), Article No. 63.

[4] M. Dudik, Z. Harchaoui, and J. Malick, Lifted coordinate descent for learning with trace-norm regularization, In AISTATS, 2012.

[5] M. Frank and P. Wolfe, An algorithm for quadratic programming, Naval Research Logistics Quarterly, **3** (1956), 95–110.

[6] M. Jaggi, Sparse Convex Optimization Methods for Ma- chine Learning, PhD thesis, ETH Zürich, 2011.

[7] M. Jaggi, Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization, Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.

[8] M. Jaggi and M. Sulovský, A Simple Algorithm for Nuclear Norm Regularized Problems. ICML, 2010.

[9] V.G. Karmanov, Mathematical Programming, Mir Publishers, Moscow, 1989.

[10] A. Nemirovski, Optimization II: Numerical methods for nonlinear continuous optimization, Lecture Notes, Israel Institute of Technology, 1999.

[11] Yu. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Kluwer Academic Publishers, Boston, 2004.

[12] H. Nguyen and G. Petrova, *Greedy strategies for convex optimization*, preprint.

[13] S. Shalev-Shwartz, N. Srebro, and T. Zhang, Trading accuracy for sparsity in optimization problems with sparsity constrains, SIAM Journal on Optimization, **20(6)** (2010), 2807–2832.

[14] V.N. Temlyakov, Greedy Algorithms and m-term Approximation With Regard to Redundant Dictionaries, J. Approx. Theory **98** (1999), 117–145.

[15] V.N. Temlyakov, Greedy-Type Approximation in Banach Spaces and Applications, Constr. Approx., **21** (2005), 257–292.

[16] V.N. Temlyakov, Greedy approximation, Cambridge University Press, 2011.

[17] V.N. Temlyakov, Greedy approximation in convex optimization, IMI Preprint, 2012:03, 1–25; arXiv:1206.0392v1, 2 Jun 2012.

[18] V.N. Temlyakov, Greedy expansions in convex optimization, IMI Preprint, 2012:04, 1–27; arXiv:1206.0393v1, 2 Jun 2012.

[19] A. Tewari, P. Ravikumar, and I.S. Dhillon, Greedy Algorithms for Structurally Constrained High Dimensional Problems, prerint, (2012), 1–10.

[20] T. Zhang, Sequential greedy approximation for certain convex optimization problems, IEEE Transactions on Information Theory, **49(3)** (2003), 682–691.

Ronald A. DeVore, Department of Mathematics, Texas A& M University, College Station, TX 77843, email: rdevore@math.tamu.edu

Vladimir Temlyakov, Department of Mathematics, University of South Carolina, Columbia, SC 29208, email: temlyak@math.sc.edu