SIAM J. OPTIM.
Vol. 30, No. 1, pp. 262–289

© 2020 Society for Industrial and Applied Mathematics

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

SHARPNESS, RESTART, AND ACCELERATION[*]

VINCENT ROULET[†] AND ALEXANDRE D'ASPREMONT[‡]

Abstract. The Łojasiewicz inequality shows that sharpness bounds on the minimum of convex optimization problems hold almost generically. Sharpness directly controls the performance of restart schemes, as observed by Nemirovskii and Nesterov [*USSR Comput. Math. Math. Phys.*, 25 (1985), pp. 21–30]. The constants quantifying these sharpness bounds are of course unobservable, but we show that optimal restart strategies are robust, and searching for the best scheme only increases the complexity by a logarithmic factor compared to the optimal bound. Overall then, restart schemes generically accelerate accelerated methods.

Key words. error bounds, sharpness, restart, accelerated methods

AMS subject classification. 90C25

DOI. 10.1137/18M1224568

Introduction. We study[1] convex optimization problems of the form

$$\text{minimize} \quad f(x), \tag{P}$$

where f is a convex function defined on $\mathbb{R}^n$. The complexity of these problems using first order methods is usually controlled by smoothness assumptions on f such as Lipschitz continuity of its gradient. Additional assumptions such as strong and uniform convexity provide, respectively, linear and faster polynomial rates of convergence [33, 20]. However, these assumptions are often too restrictive to be applicable. Here, we make a much more generic assumption that describes the growth of the function around its minimizers using constants $\mu > 0$ and $r \geq 1$ such that

$$\frac{\mu}{r} d(x, X^*)^r \leq f(x) - f^* \quad \text{for every } x \in K, \tag{Loja}$$

where f^* is the minimum of f, $K \supset X^*$ is a given set, and $d(x, X^*) = \min_{y \in X^*} \|x-y\|_2$ is the Euclidean distance from x to the set X^* of minimizers of f. This defines a *lower bound* on the function around its minimizers and quantifies the sharpness of the minimum. We exploit this property using restart schemes on classical convex optimization algorithms.

The sharpness assumption (Loja) is also known as a Hölderian error bound on the distance to the set of minimizers. Hoffman [19] first introduced error bounds to study systems of linear inequalities. Natural extensions were then developed for convex optimization by Robinson [40], Mangasarian [28], and Auslender and Crouzeix [3], notably through the concept of sharp minimum [11, 10]. But the most striking

[*] Received by the editors November 5, 2018; accepted for publication (in revised form) October 14, 2019; published electronically January 14, 2020. A subset of these results appeared at the NIPS 2017 conference under the same title.
https://doi.org/10.1137/18M1224568
Funding: The work of the authors was supported by the Fonds AXA Pour la Recherche, a gift from Société Générale Cross Asset Quantitative Research, an AMX fellowship, and a Google Focused award.

[†] Department of Statistics, University of Washington, Seattle, WA 98195 (vincent.roulet.1509@gmail.com).
[‡] Département d'Informatique, CNRS & École Normale Supérieure, 75005 Paris, France (aspremon@ens.fr).

[1] A subset of these results appeared at the NIPS 2017 conference under the same title.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

result in this vein is due to Łojasiewicz [25, 26], who proved that inequality (Loja) holds generically for real analytic and subanalytic functions. This result has then been extended to nonsmooth subanalytic convex functions by Bolte, Daniilidis, and Lewis [7]. Overall then, condition (Loja) essentially measures the sharpness of minimizers and holds generically. On the other hand, this inequality is purely implicit as r and μ are neither observed nor known a priori, and deriving adaptive schemes is thus crucial to ensure practical relevance.

Łojasiewicz inequalities either in the form of (Loja) or as gradient dominated properties [37] led to new convergence results for composite problems and for alternating or splitting methods [2, 9, 15, 21]. Here we use this inequality to produce accelerated rates for restart schemes.

Restart schemes have already been studied for strongly or uniformly convex functions in, e.g., [29, 32, 20, 23]. In particular, Nemirovskii and Nesterov [29] link a "strict minimum" condition akin to (Loja) with faster convergence rates using restart schemes which form the basis of our results, but they do not study the cost of adaptation and do not tackle the nonsmooth case. In a similar spirit, weaker versions of this strict minimum condition were used more recently to study the performance of restart schemes in [38, 16, 41].

The fundamental question regarding restart schemes is to define when to restart. Several heuristics have been presented that used some criterion on the iterates to restart the accelerated algorithm and speed up convergence [35, 42, 18]. However, they did not theoretically establish improved complexity bounds. The robustness of restart schemes was also studied by Fercoq and Qu [13] for quadratic error bounds, i.e., (Loja) with $r = 2$, satisfied by the LASSO problem, for example. Fercoq and Qu [14] recently extended this work to produce adaptive restarts with theoretical guarantees of optimal performance, again for quadratic error bounds. In the same vein, Liu and Yang [24] presented adaptive accelerated methods given Hölderian error bounds, but their results are not adaptive to the exponent of the error bound. The references above focus on smooth problems, but error bounds appear also for nonsmooth ones, with Gilpin, Pena, and Sandholm [17] proving, for example, linear convergence of restart schemes in bilinear matrix games where the minimum is sharp, i.e., (Loja) with $r = 1$. Recently Renegar and Grimmer [39] presented simple generic schemes inspired by an early draft of this work and provided adaptive schemes in all regimes (not only the smooth case).

Our contribution here is to derive optimal scheduled restart schemes for general convex optimization problems on smooth, nonsmooth, or Hölder smooth functions satisfying a sharpness assumption. We then show that for smooth functions these schemes can be made adaptive with nearly optimal complexity (up to a squared log term) for a wide array of sharpness assumptions. We also analyze restart schemes based on a sufficient decrease of the primal gap, when the optimal value of the problem is known. In that case, restart schemes are shown to be optimal without requiring a log scale grid search on the parameters. Our proofs only rely on having access to the convergence bound of an accelerated method; therefore, our results are directly extended to the non-Euclidean case with composite objective and to nonsmooth functions that can be smoothed.

1. Regularity assumptions.

1.1. Smoothness. Convex optimization problems (P) are generally divided into two classes: smooth problems, for which f has Lipschitz continuous gradients, and nonsmooth problems, for which f is not differentiable. Following Nesterov [34], we

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

use a unified framework that extends the definition of Hölder smooth functions.

Definition 1.1. *A function f is s-smooth for given $1 \leq s \leq 2$ if there exists a constant $L > 0$ such that*

$$\text{(Hölder)} \qquad \|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1} \quad \textit{for all } x, y \in \mathbf{dom}\, f$$

and any subgradients $\nabla f(x) \in \partial f(x), \nabla f(y) \in \partial f(y)$ of f at x, y, respectively. We write $\mathcal{H}_{s,L}$ as the set of s-smooth functions with parameter L.

For $s = 2$, we retrieve the classical definition of smoothness [33]. For $s = 1$, we get a classical assumption made in nonsmooth convex optimization, i.e., that subgradients of the function are bounded. For $s \in]1, 2[$ we get the definition of Hölder smooth functions. We generalize our results for functions that are smooth with respect to a non-Euclidean norm in section 5.

1.2. Sharpness / error bounds / growth property. We study convex optimization problems whose objective satisfies a growth condition as defined below.

Definition 1.2. *A function f satisfies a Łojasiewicz growth condition on a set K if there exist constants $r \geq 1$, $\mu > 0$, such that*

$$\text{(Loja)} \qquad \frac{\mu}{r} d(x, X^*)^r \leq f(x) - f^* \quad \textit{for every } x \in K,$$

where f^* is the minimum of f, and $d(x, X^*) = \min_{y \in X^*} \|x - y\|_2$ is the Euclidean distance from x to the set X^* of minimizers of f. We write $\mathcal{L}_{r,\mu}(K)$ as the set of functions satisfying a Łojasiewicz growth condition on a set K with parameters $r \geq 1$, $\mu > 0$.

Condition (Loja) holds almost generically and is notably satisfied by analytic and subanalytic functions (see [8] for more details). However, the proof (see, e.g., [6, Theorem 6.4]) uses topological arguments that are far from constructive. Hence, outside of some particular cases (e.g., strong convexity), we cannot assume that the constants in (Loja) are known—even approximately.

Error bounds are directly related to a Łojasiewicz inequality bounding the magnitude of the gradient [8]. These properties underlie many recent results in optimization [2, 15, 9]. Here, the sharpness condition in (Loja) allows us to accelerate convex optimization algorithms using restart schemes.

Our analysis relies on the condition that (Loja) is satisfied for any output of the algorithms we restart. By enforcing monotonicity of the objective values produced by those algorithms, this reduces to assuming that (Loja) is satisfied on sublevel sets of the objective.

1.3. Sharpness and smoothness. Given a convex function $f \in \mathcal{H}_{s,L}$, by using its Taylor expansion and the smoothness property, we get $f(x) \leq f^* + \frac{L}{s}\|x - y\|_2^s$ for $x \in \mathbf{dom}\, f$ and $y \in X^*$. Setting y to be the projection of x onto X^*, this yields the following *upper bound* on suboptimality:

$$(1) \qquad f(x) - f^* \leq \frac{L}{s} d(x, X^*)^s.$$

Now, assume moreover that $f \in \mathcal{L}_{r,\mu}(K)$ for a given set K such that $X^* \subset K \subset \mathbf{dom}\, f$. Combining (1) and (Loja) leads to

$$\frac{s\mu}{rL} \leq d(x, X^*)^{s-r}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

for every $x \in K \setminus X^*$. This means that necessarily $s \le r$ by taking x close enough to X^*. Moreover, if $s < r$, the set K must satisfy $\sup_{x \in K} d(x, X^*) < +\infty$.

For the following, we define

$$\kappa \triangleq \frac{L^{\frac{2}{s}}}{\mu^{\frac{2}{r}}} \qquad \text{and} \qquad \tau \triangleq 1 - \frac{s}{r} \in [0,1), \tag{2}$$

a generalized condition number for the function f and a condition number based on the ratio of powers in inequalities (Hölder) and (Loja), respectively. Note that if $r = s = 2$, κ matches the classical condition number of the function.

2. Scheduled restarts for smooth convex problems.

In this section, we seek to solve (P) assuming that the function f is smooth, i.e., satisfies (Hölder) with $s = 2$ and $L > 0$. Without further assumptions on f, an optimal algorithm to solve the smooth convex optimization problem (P) is Nesterov's accelerated gradient method [30]. Given an initial point x_0, this algorithm outputs, after t iterations, a point

$$x = \mathcal{A}(x_0, t) \qquad \text{such that} \qquad f(x) - f^* \le \frac{cL}{t^2} d(x_0, X^*)^2, \tag{3}$$

where $c > 0$ is a universal constant (whose value will be allowed to vary in what follows, with $c = 4$ here). The accelerated algorithm can be enforced to output solutions whose objective decays monotonically as detailed in Appendix A. Consequently, if f satisfies a Łojasiewicz growth condition on the initial sublevel set $K = \{x : f(x) \le f(x_0)\}$, then it is satisfied for any point output by the algorithm.

Note that the arguments that we develop below are not specific to the algorithm of Nesterov [30] and would apply to any method satisfying the complexity bound (3), as shown, for example, in section 5, which generalizes the results to the non-Euclidean setting. We now describe a restart scheme exploiting the extra regularity (Loja) to improve the computational complexity of solving problem (P) using accelerated methods.

2.1. Scheduled restarts.

Here, we schedule the number of iterations t_k made by the accelerated gradient algorithm between restarts, with t_k being the number of (inner) iterations at the kth algorithm run (outer iteration). Our scheme is described in Algorithm 1 below.

Algorithm 1 Scheduled restarts for smooth convex minimization.

```
Inputs : $x_0 \in \mathbb{R}^n$ and a sequence $t_k$ for $k = 1, \dots, R$.
for $k = 1, \dots, R$ do
    $x_k := \mathcal{A}(x_{k-1}, t_k)$
end for
Output : $\hat{x} := x_R$
```

The analysis of this scheme and the following schemes relies on two steps. We first choose schedules that ensure linear convergence of the objective values $f(x_k)$ with respect to k at a given rate. We then adjust this linear rate to minimize complexity, i.e., the total number of inner iterations. We begin with a technical lemma which assumes linear convergence holds, and connects the growth of t_k, the precision reached, and the total number of inner iterations N.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Lemma 2.1. *Let x_k be a sequence whose kth iterate is generated from the previous one by an algorithm that runs t_k iterations, and write $N = \sum_{k=1}^R t_k$, the total number of iterations to output a point x_R. Suppose setting $t_k = Ce^{\alpha k}$ $(k = 1, \dots, R)$ for some $C > 0$ and $\alpha \geq 0$ ensures that the outer iterations satisfy*

$$f(x_k) - f^* \leq \nu e^{-\gamma k} \tag{4}$$

for all $k \geq 0$ where $\nu \geq 0$ and $\gamma \geq 0$. Then, precision at the output is given by

$$f(x_R) - f^* \leq \nu \exp(-\gamma N/C) \quad \textit{when } \alpha = 0$$

and

$$f(x_R) - f^* \leq \frac{\nu}{(\alpha e^{-\alpha} C^{-1} N + 1)^{\frac{\gamma}{\alpha}}} \quad \textit{when } \alpha > 0.$$

Proof. When $\alpha = 0$, $N = RC$, and inserting this into (4) at the last point x_R yields the desired result. On the other hand, when $\alpha > 0$, we have $N = \sum_{k=1}^R t_k = Ce^{\alpha}\frac{e^{\alpha R}-1}{e^{\alpha}-1}$, which gives $R = \log\left(\frac{e^{\alpha}-1}{e^{\alpha}C}N + 1\right)/\alpha$. Inserting this into (4) at the last point, we get

$$f(x_R) - f^* \leq \nu \exp\left(-\frac{\gamma}{\alpha}\log\left(\frac{e^{\alpha}-1}{e^{\alpha}C}N + 1\right)\right) \leq \frac{\nu}{(\alpha e^{-\alpha} C^{-1} N + 1)^{\frac{\gamma}{\alpha}}},$$

where we used $e^x - 1 \geq x$. This yields the second part of the result. □

The last approximation in the case $\alpha > 0$ simplifies the analysis that follows without significantly affecting the bounds. We also show in Appendix B that using integer values $\tilde{t}_k = \lceil t_k \rceil$ does not significantly affect the bounds above.

We now analyze restart schedules t_k that ensure linear convergence. Our choice of t_k will heavily depend on the ratio between r and s (with $s = 2$ for smooth functions here), measured by $\tau = 1 - s/r$ defined in (2). Below, we show that if $\tau = 0$, a constant schedule is sufficient to ensure linear convergence. When $\tau > 0$, we need a geometrically increasing number of iterations for each cycle.

Proposition 2.2. *Let f be a convex function and $x_0 \in \mathbf{dom}\, f$. Denote $K = \{x : f(x) \leq f(x_0)\}$ and assume that $f \in \mathcal{H}_{2,L} \cap \mathcal{L}_{r,\mu}(K)$. Run Algorithm 1 from x_0 with iteration schedule $t_k = C^*_{\kappa,\tau} e^{\tau k}$, for $k = 1, \dots, R$, where*

$$C^*_{\kappa,\tau} \triangleq e^{1-\tau}(c\kappa)^{\frac{1}{2}}(f(x_0) - f^*)^{-\frac{\tau}{2}}, \tag{5}$$

with κ and τ defined in (2) and $c = 4e^{2/e}$ here. The precision reached at the last point $\hat{x}$ is given by

$$f(\hat{x}) - f^* \leq \exp\left(-2e^{-1}(c\kappa)^{-\frac{1}{2}}N\right)(f(x_0) - f^*) = O\left(\exp(-\kappa^{-\frac{1}{2}}N)\right) \quad \textit{when } \tau = 0, \tag{6}$$

while

$$f(\hat{x}) - f^* \leq \frac{f(x_0) - f^*}{\left(\tau e^{-1}(f(x_0) - f^*)^{\frac{\tau}{2}}(c\kappa)^{-\frac{1}{2}}N + 1\right)^{\frac{2}{\tau}}} = O\left(N^{-\frac{2}{\tau}}\right) \quad \textit{when } \tau > 0, \tag{7}$$

where $N = \sum_{k=1}^R t_k$ is the total number of iterations.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. Our strategy is to choose t_k such that the objective is linearly decreasing, i.e.,

$$f\left(x_k\right) - f^* \leq e^{-\gamma k}(f(x_0) - f^*) \tag{8}$$

for some $\gamma \geq 0$ depending on the choice of t_k. This directly holds for $k = 0$ and any $\gamma \geq 0$. Combining (Loja) with the complexity bound in (3), we get

$$f\left(x_k\right) - f^* \leq \frac{c\kappa}{t_k^2}\left(f\left(x_{k-1}\right) - f^*\right)^{\frac{2}{r}}, \tag{9}$$

where $c = 4e^{2/e}$ using that $r^{2/r} \leq e^{2/e}$. Assuming recursively that (8) is satisfied at iteration $k-1$ for a given γ, we have

$$f\left(x_k\right) - f^* \leq \frac{c\kappa e^{-\gamma\frac{2}{r}(k-1)}}{t_k^2}(f(x_0) - f^*)^{\frac{2}{r}},$$

and to ensure (8) at iteration k, we impose

$$\frac{c\kappa e^{-\gamma\frac{2}{r}(k-1)}}{t_k^2}(f(x_0) - f^*)^{\frac{2}{r}} \leq e^{-\gamma k}(f(x_0) - f^*).$$

Rearranging terms in this last inequality, using τ defined in (2), we get

$$t_k \geq e^{\frac{\gamma(1-\tau)}{2}}(c\kappa)^{\frac{1}{2}}(f(x_0) - f^*)^{-\frac{\tau}{2}}e^{\frac{\tau\gamma}{2}k}. \tag{10}$$

For a given $\gamma \geq 0$, we can set $t_k = Ce^{\alpha k}$ where

$$C = e^{\frac{\gamma(1-\tau)}{2}}(c\kappa)^{\frac{1}{2}}(f(x_0) - f^*)^{-\frac{\tau}{2}} \qquad \text{and} \qquad \alpha = \tau\gamma/2, \tag{11}$$

and Lemma 2.1 then yields

$$f(\hat{x}) - f^* \leq \exp\left(-\gamma e^{-\frac{\gamma}{2}}(c\kappa)^{-\frac{1}{2}}N\right)(f(x_0) - f^*)$$

when $\tau = 0$, while

$$f(\hat{x}) - f^* \leq \frac{f(x_0) - f^*}{\left(\frac{\tau}{2}\gamma e^{-\frac{\gamma}{2}}(c\kappa)^{-\frac{1}{2}}(f(x_0) - f^*)^{\frac{\tau}{2}}N + 1\right)^{\frac{2}{\tau}}}$$

when $\tau > 0$. These bounds are minimal for $\gamma = 2$, which yields the desired result. □

When $\tau = 0$, bound (6) matches the classical complexity bound for smooth strongly convex functions [33]. When $\tau > 0$, on the other hand, bound (7) highlights a *faster convergence rate than accelerated gradient methods.* The sharper the function (i.e., the closer r is to 2), the faster the convergence. This matches the lower bounds for optimizing smooth and sharp functions up to constant factors [29, eq. 1.21]. Also, setting $t_k = C^*_{\kappa,\tau}e^{\tau k}$ yields continuous bounds on precision, i.e., when $\tau \to 0$, bound (7) converges to bound (6), which also shows that for τ near zero, constant restart schemes are almost optimal.

Note that for $N \leq C^*_{\kappa,\tau}$, the bounds (6) and (7) are not informative. Precisely, the lower bounds for this problem as presented in [29, eq. 1.21] are not informative for small N. In that case, the optimal rate is given by the accelerated scheme and consequently by Algorithm 1 before the first restart.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2.2. Adaptive scheduled restart. The previous restart schedules depend on the sharpness parameters (r, μ) in (Loja). In general, of course, these values are neither observed nor known a priori. Making the restart scheme adaptive is thus crucial to its practical performance. Fortunately, we show below that a simple logarithmic grid search on these parameters is enough to guarantee nearly optimal performance.

We begin with the following proposition, which stems from the proof of Proposition 2.2.

PROPOSITION 2.3. *Let f be a convex function and $x_0 \in \mathbf{dom}\, f$. Denote $K = \{x : f(x) \leq f(x_0)\}$ and assume that $f \in \mathcal{H}_{2,L} \cap \mathcal{L}_{r,\mu}(K)$. Run Algorithm 1 from x_0 with general schedules of the form*

$$\begin{cases} t_k = C & \text{if } \tau = 0, \\ t_k = Ce^{\alpha k} & \text{if } \tau > 0. \end{cases}$$

If $\tau = 0$ and $C \geq C^*_{\kappa,0}$, then

$$f(\hat{x}) - f^* \leq \left(\frac{c\kappa}{C^2}\right)^{\frac{N}{C}} (f(x_0) - f^*), \tag{12}$$

while if $\tau > 0$ and $C \geq C(\alpha)$, then

$$f(\hat{x}) - f^* \leq \frac{f(x_0) - f^*}{(\alpha e^{-\alpha} C^{-1} N + 1)^{\frac{2}{\tau}}}, \tag{13}$$

where

$$C(\alpha) \triangleq e^{\frac{\alpha(1-\tau)}{\tau}} (c\kappa)^{\frac{1}{2}} (f(x_0) - f^*)^{-\frac{\tau}{2}}, \tag{14}$$

and $N = \sum_{k=1}^R t_k$ is the total number of iterations.

Proof. Given general schedules of the form

$$\begin{cases} t_k = C & \text{if } \tau = 0, \\ t_k = Ce^{\alpha k} & \text{if } \tau > 0, \end{cases}$$

the best value of γ satisfying condition (10) for any $k \geq 0$ in Proposition 2.2 is given by

$$\begin{cases} \gamma = \log\left(\frac{C^2}{c\kappa}\right) & \text{if } \tau = 0 \text{ and } C \geq C^*_{\kappa,0}, \\ \gamma = \frac{2\alpha}{\tau} & \text{if } \tau > 0 \text{ and } C \geq C(\alpha). \end{cases}$$

As in Proposition 2.2, plugging these values into the bounds of Lemma 2.1 yields the desired result. □

We run several schemes with a fixed number of inner iterations N to perform a log-scale grid search on τ and κ. We define these schemes as follows:

$$\begin{cases} \mathcal{S}_{i,0} : \text{Restart Algorithm 1 with } t_k = C_i, \\ \mathcal{S}_{i,j} : \text{Restart Algorithm 1 with } t_k = C_i e^{\tau_j k}, \end{cases} \tag{15}$$

where $C_i = 2^i$ and $\tau_j = 2^{-j}$. We stop each of these schemes when the total number of its inner iterations has exceeded N, i.e., at the smallest R such that $\sum_{k=1}^R t_k \geq N$. The size of the grid search in C_i is naturally bounded as we cannot restart the algorithm after more than N total inner iterations, so $i \in [1, \ldots, \lfloor \log_2 N \rfloor]$. We also show that

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

when τ is smaller than $1/N$, a constant schedule where $t_k = C$ performs as well as the optimal geometrically increasing schedule where $t_k = C^*_{\kappa,\tau} e^{\tau k}$. This crucially means we can also choose $j \in [1, \dots, \lceil \log_2 N \rceil]$, hence limiting the cost of the grid search.

The following proposition details the convergence of this grid search, using the same notation as in Proposition 2.2. As observed at the end of section 2.1, the optimal bounds (6) and (7) are only informative after a sufficient number of iterations, which is why we analyze the adaptive scheme only for a number of iterations $N \geq 2C^*_{\kappa,\tau}$. To get optimal bounds in all regimes, it suffices to run an additional nonrestarted algorithm that will also capture the best rate in the case $N < 2C^*_{\kappa,\tau}$.

PROPOSITION 2.4. *Let f be a convex function and $x_0 \in \mathbf{dom}\, f$. Denote $K = \{x : f(x) \leq f(x_0)\}$, assume that $f \in \mathcal{H}_{2,L} \cap \mathcal{L}_{r,\mu}(K)$, and denote by $N \geq 2C^*_{\kappa,\tau}$ a given number of iterations.*

Run schemes $\mathcal{S}_{i,j}$ defined in (15) *to solve* (P) *for $i \in [1, \dots, \lfloor \log_2 N \rfloor]$ and $j \in [0, \dots, \lceil \log_2 N \rceil]$, stopping each time after N total inner algorithm iterations, i.e., for R such that $\sum_{k=1}^R t_k \geq N$.*

If $\tau = 0$, there exists $i \in [1, \dots, \lfloor \log_2 N \rfloor]$ such that the scheme $\mathcal{S}_{i,0}$ achieves a precision given by

$$f(\hat{x}) - f^* \leq \exp\left(-e^{-1}(c\kappa)^{-\frac{1}{2}} N\right)(f(x_0) - f^*).$$

If $\tau > 0$, there exist $i \in [1, \dots, \lfloor \log_2 N \rfloor]$ and $j \in [0, \dots, \lceil \log_2 N \rceil]$ such that the scheme $\mathcal{S}_{i,j}$ achieves a precision given by

$$f(\hat{x}) - f^* \leq \frac{f(x_0) - f^*}{\left(\tau e^{-1}(c\kappa)^{-\frac{1}{2}}(f(x_0) - f^*)^{\frac{\tau}{2}}(N-1)/4 + 1\right)^{\frac{2}{\tau}}}.$$

Overall, running the logarithmic grid search has a complexity $(\log_2 N)^2$ *times higher than running N iterations using the optimal (oracle) scheme.*

Proof. Denoting by R the number of restarts of a scheme S_{ij}, we have, for $j = 0$, $R = \lceil N/C_i \rceil$ and, for $j \neq 0$, $R = \lceil \log(\frac{e^{\tau_j}-1}{e^{\tau_j} C_i} N + 1)/\tau_j \rceil$. Denote by $N' = \sum_{k=1}^R t_k \geq N$ the number of iterations of a scheme $\mathcal{S}_{i,j}$. We necessarily have $N' \leq 2e^{1/2}N$ for our choice of C_i and τ_j. Hence the cost of running all methods is on the order of $N(\log_2 N)^2$.

If $\tau = 0$ and $N \geq 2C^*_{\kappa,0}$, then $i = \lceil \log_2 C^*_{\kappa,0} \rceil \leq \lfloor \log_2 N \rfloor$. Therefore, $\mathcal{S}_{i,0}$ has been run, and bound (12) shows then that the last iterate $\hat{x}$ satisfies

$$f(\hat{x}) - f^* \leq \left(\frac{c\kappa}{C_i^2}\right)^{\frac{N}{C_i}} (f(x_0) - f^*).$$

Using that $C^*_{\kappa,0} \leq C_i \leq 2C^*_{\kappa,0}$, we have

$$f(\hat{x}) - f^* \leq \left(\frac{c\kappa}{(C^*_{\kappa,0})^2}\right)^{\frac{N}{2C^*_{\kappa,0}}} (f(x_0) - f^*) \leq \exp\left(-e^{-1}(c\kappa)^{-\frac{1}{2}} N\right)(f(x_0) - f^*).$$

If $\tau \geq \frac{1}{N}$ and $N \geq 2C^*_{\kappa,\tau}$, then $j = \lceil -\log_2 \tau \rceil \leq \lceil \log_2 N \rceil$ and $i = \lceil \log_2 C^*_{\kappa,\tau} \rceil \leq \lfloor \log_2 N \rfloor$. Therefore, scheme $\mathcal{S}_{i,j}$ has been run. As $C_i \geq C^*_{\kappa,\tau} \geq C(\tau_j)$, where $C(\tau_j)$ is as defined in (14), bound (13) shows that the last iterate $\hat{x}$ of scheme $\mathcal{S}_{i,j}$ satisfies

$$f(\hat{x}) - f^* \leq \frac{f(x_0) - f^*}{\left(\tau_j e^{-\tau_j} C_i^{-1} N + 1\right)^{\frac{2}{\tau}}}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Finally, by definition of i and j, $2\tau_j \geq \tau$ and $C_i \leq 2C^*_{\kappa,\tau}$, so

$$f(\hat{x}) - f^* \leq \frac{f(x_0) - f^*}{\left(\tau e^{-\tau_j}(C^*_{\kappa,\tau})^{-1}N/4 + 1\right)^{\frac{2}{\tau}}} = \frac{f(x_0) - f^*}{\left(\tau e^{-1}(c\kappa)^{-\frac{1}{2}}(f(x_0) - f^*)^{\frac{\tau}{2}}N/4 + 1\right)^{\frac{2}{\tau}}},$$

where we concluded by expanding $C^*_{\kappa,\tau} = e^{1-\tau}(c\kappa)^{\frac{1}{2}}(f(x_0) - f^*)^{-\frac{\tau}{2}}$ and using that $\tau \geq \tau_j$.

If $\frac{1}{N} > \tau > 0$ and $N > 2C^*_{\kappa,\tau}$, then $i = \lceil \log_2 C^*_{\kappa,\tau} \rceil \leq \lfloor \log_2 N \rfloor$, so scheme $\mathcal{S}_{i,0}$ has been run. As in (9), its iterates x_k satisfy, with $1 - \tau = 2/r$,

$$\begin{aligned} f(x_k) - f^* &\leq \frac{c\kappa}{C_i^2}(f(x_{k-1}) - f^*)^{\frac{2}{r}} \\ &\leq \left(\frac{c\kappa}{C_i^2}\right)^{\left(1-(1-\tau)^k\right)/\tau} (f(x_0) - f^*)^{(1-\tau)^k} \\ &\leq \left(\frac{c\kappa(f(x_0) - f^*)^{-\tau}}{C_i^2}\right)^{\left(1-(1-\tau)^k\right)/\tau} (f(x_0) - f^*). \end{aligned}$$

Now $C_i \geq C^*_{\kappa,\tau} = e^{1-\tau}(c\kappa)^{\frac{1}{2}}(f(x_0) - f^*)^{-\frac{\tau}{2}}$ and $C_i R \geq N$; therefore, the last iterate $\hat{x}$ satisfies

$$f(\hat{x}) - f^* \leq \exp\left(-2(1-\tau)\frac{1 - (1-\tau)^{N/C_i}}{\tau}\right)(f(x_0) - f^*).$$

As $N \geq C_i$, since $h(\tau) = \frac{(1-\tau)(1-(1-\tau)^{\frac{N}{C_i}})}{1-(1-\tau)}$ is decreasing with τ and $\frac{1}{N} > \tau > 0$, we have

$$\begin{aligned} f(\hat{x}) - f^* &\leq \exp\left(-2(N-1)\left(1 - \left(1 - \frac{1}{N}\right)^{N/C_i}\right)\right)(f(x_0) - f^*) \\ &\leq \exp\left(-2(N-1)\left(1 - \exp\left(-\frac{1}{C_i}\right)\right)\right)(f(x_0) - f^*) \\ &\leq \exp\left(-2\frac{N-1}{C_i}\left(1 - \frac{1}{2C_i}\right)\right)(f(x_0) - f^*), \end{aligned}$$

having used the facts that (i) $(1 + ax)^{\frac{b}{x}} \leq \exp(ab)$ if $ax \geq -1$, and $\frac{b}{x} \geq 0$; (ii) $1 - x + \frac{x^2}{2} \geq \exp(-x)$ when $x \geq 0$. As $C_i = 2^i \geq 1$, we finally get

$$\begin{aligned} f(\hat{x}) - f^* &\leq \exp\left(-\frac{N-1}{C_i}\right)(f(x_0) - f^*) \\ &\leq \exp\left(-\frac{N-1}{2C^*_{\kappa,\tau}}\right)(f(x_0) - f^*) \\ &\leq \frac{f(x_0) - f^*}{\left(\tau(C^*_{\kappa,\tau})^{-1}(N-1)/4 + 1\right)^{\frac{2}{\tau}}} \\ &\leq \frac{f(x_0) - f^*}{\left(\tau(f(x_0) - f^*)^{\frac{\tau}{2}}e^{-1}(c\kappa)^{-\frac{1}{2}}(N-1)/4 + 1\right)^{\frac{2}{\tau}}}, \end{aligned}$$

using the fact that $e^\tau \geq 1$. ∎

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

In the strongly convex case, this adaptive bound is similar to the one of [32] to optimize smooth strongly convex functions in the sense that we lose approximately a log factor of the condition number of the function. However, our assumptions are weaker and our bound also handles all sharpness regimes, i.e., any exponent $r \in [2, +\infty]$—not just the strongly convex case. Finally, the step size chosen for the grid search was set to 2. The proof can be adapted for a generic step size h; the size of the grid may be reduced, but corresponding bounds will suffer an h^2 approximation loss compared to the best schedule.

Note that the scheduled restart schemes we present here adapt to a global sharpness hypothesis on the sublevel set defined by the initial point and are not locally adaptive to potentially better constant μ on smaller sublevel sets. On the other hand, restart schemes based on a primal gap, presented in section 4, do adapt to the local value of μ, although these schemes require having access to the primal gap.

2.3. Comparison to gradient descent. We end this section by analyzing the behavior of gradient descent in light of the sharpness assumption in order to compare the advantage of the restarted accelerated method over plain gradient descent. While the bounds we obtain using the basic gradient method are suboptimal compared to the ones above, the gradient algorithm having no memory will automatically adapt to the best "restart" schedule. Given only the smoothness hypothesis, the gradient descent algorithm, presented in, e.g., [34], starts from a point x_0 and outputs iterates

$$x_t = \mathcal{G}(x_0, t) \quad \text{such that} \quad f(x_t) - f^* \leq \frac{L}{t} d(x_0, X^*)^2.$$

While accelerated methods use the last two iterates to compute the next one, simple gradient descent algorithms use only the last iterate, so the algorithm can be seen as (implicitly) restarting at each iteration. Formally we use that its convergence can be bounded as, for $k \geq 1$,

$$f(x_{k+t}) - f^* \leq \frac{L}{t} d(x_k, X^*)^2, \tag{16}$$

and we analyze it in light of the restart interpretation using the sharpness property.

PROPOSITION 2.5. *Let f be a convex function and $x_0 \in \mathbf{dom}\, f$. Denote $K = \{x : f(x) \leq f(x_0)\}$ and assume that $f \in \mathcal{H}_{2,L} \cap \mathcal{L}_{r,\mu}(K)$. Denote by $x_t = \mathcal{G}(x_0, t)$ the iterate sequence generated by the gradient descent algorithm started at x_0 to solve* (P) *and define*

$$t_k = e^{1-\tau} c\kappa (f(x_0) - f^*)^\tau e^{\tau k},$$

with κ and τ defined in (2) *and $c = e^{2/e}$ here. The precision reached after $N = \sum_{k=1}^n t_k$ iterations is given by*

$$f(x_N) - f^* \leq \exp\left(-e^{-1}(c\kappa)^{-1} N\right)(f(x_0) - f^*) = O\left(\exp(-\kappa^{-1} N)\right) \quad \textit{when } \tau = 0,$$

while

$$f(x_N) - f^* \leq \frac{f(x_0) - f^*}{(\tau e^{-1}(c\kappa)^{-1}(f(x_0) - f^*)^\tau N + 1)^{\frac{1}{\tau}}} = O\left(N^{-\frac{1}{\tau}}\right) \quad \textit{when } \tau > 0.$$

Proof. For a given $\gamma \geq 0$, we construct a subsequence $x_{\phi(k)}$ of x_t such that

$$f(x_{\phi(k)}) - f^* \leq e^{-\gamma k}(f(x_0) - f^*). \tag{17}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Define $x_{\phi(0)} = x_0$. Assume that (17) is true at iteration $k-1$; then, combining complexity bound (16) and (Loja), for any $t \geq 1$, we have

$$\begin{aligned} f(x_{\phi(k-1)+t}) - f^* &\leq \frac{c\kappa}{t}(f(x_{\phi(k-1)}) - f^*)^{\frac{2}{r}} \\ &\leq \frac{c\kappa}{t} e^{-\gamma\frac{2}{r}(k-1)}(f(x_0) - f^*)^{\frac{2}{r}}, \end{aligned}$$

where $c = e^{2/e}$, using that $r^{2/r} \leq e^{2/e}$. Taking $t_k = e^{\gamma(1-\tau)}c\kappa(f(x_0)-f^*)^{-\tau}e^{\gamma\tau k}$ and $\phi(k) = \phi(k-1) + t_k$ ensures that (17) holds at iteration k. Using Lemma 2.1, we obtain at iteration $N = \phi(n) = \sum_{k=1}^n t_k$

$$f(x_N) - f^* \leq \exp\left(-\gamma e^{-\gamma}(c\kappa)^{-1}N\right)(f(x_0) - f^*) \quad \text{if } \tau = 0$$

and

$$f(x_N) - f^* \leq \frac{f(x_0) - f^*}{(\tau\gamma e^{-\gamma}(c\kappa)^{-1}(f(x_0) - f^*)^{\tau}N + 1)^{\frac{1}{\tau}}} \quad \text{if } \tau > 0.$$

These bounds are minimal for $\gamma = 1$ and the results follow. ◻

We observe that restarting accelerated gradient methods reduce complexity from $O(\epsilon^{-\tau})$ to $O(\epsilon^{-\tau/2})$ compared to simple gradient descent. More general results on the convergence of (sub)gradient descent algorithms under a Łojasiewicz inequality assumption were developed by Bolte et al. [8].

3. Universal scheduled restarts for convex problems.

In this section, we generalize previous results to s-smooth functions as defined in Definition 1.1 to tackle both smooth and nonsmooth convex optimization problems. Without further assumptions on f, the optimal rate of convergence for this class of functions is bounded as $O(1/N^{\rho})$, where N is the total number of iterations and

$$\rho = 3s/2 - 1, \tag{18}$$

which gives $\rho = 2$ for smooth functions and $\rho = 1/2$ for nonsmooth functions. The universal fast gradient method [34] achieves this rate by requiring only a target accuracy ϵ and a starting point x_0. It outputs after t iterations a point

$$x = \mathcal{U}(x_0, \epsilon, t), \quad \text{such that} \quad f(x) - f^* \leq \frac{\epsilon}{2} + \frac{cL^{\frac{2}{s}}d(x_0, X^*)^2}{\epsilon^{\frac{2}{s}}t^{\frac{2\rho}{s}}}\frac{\epsilon}{2}, \tag{19}$$

where c is a constant ($c = 2^{\frac{4s-2}{s}}$). A simplified implementation of the universal fast gradient method that enforces monotonicity in objective values of the outputs of the algorithm is presented in Appendix A.

We assume again that f satisfies a Łojasiewicz growth condition on its initial sublevel set. The key difference from the smooth case described in the previous section is that here we schedule *both* the target accuracy ϵ_k used by the algorithm *and* the number of iterations t_k made at the kth run of the algorithm. Our scheme is described in Algorithm 2.

Our strategy is to choose a sequence t_k that ensures

$$f(x_k) - f^* \leq \epsilon_k$$

for the geometrically decreasing sequence ϵ_k. The overall complexity of our method will then depend on the growth of t_k as described in Lemma 2.1.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm 2 Universal scheduled restarts for convex minimization.

```
Inputs : $x_0 \in \mathbb{R}^n$, $\epsilon_0 \geq f(x_0) - f^*$, $\gamma \geq 0$, and a sequence $t_k$ for $k = 1, \ldots, R$.
for $k = 1, \ldots, R$ do
    $\epsilon_k := e^{-\gamma}\epsilon_{k-1}, \qquad x_k := \mathcal{U}(x_{k-1}, \epsilon_k, t_k)$
end for
Output : $\hat{x} := x_R$
```

PROPOSITION 3.1. *Let f be a convex function and $x_0 \in \mathbf{dom}\, f$. Denote $K = \{x : f(x) \leq f(x_0)\}$ and assume that $f \in \mathcal{H}_{s,L} \cap \mathcal{L}_{r,\mu}(K)$. Run Algorithm 2 from x_0 for a given $\epsilon_0 \geq f(x_0) - f^*$ with*

$$\gamma = \rho, \qquad t_k = C^*_{\kappa,\tau,\rho} e^{\tau k}, \quad \textit{where} \quad C^*_{\kappa,\tau,\rho} \triangleq e^{1-\tau}(c\kappa)^{\frac{s}{2\rho}} \epsilon_0^{-\frac{\tau}{\rho}},$$

where ρ is as defined in (18), *κ and τ are as defined in* (2), *and $c = 8e^{2/e}$ here. The precision reached at the last point $\hat{x}$ is given by*

$$f(\hat{x}) - f^* \;\leq\; \exp\left(-\rho e^{-1}(c\kappa)^{-\frac{s}{2\rho}} N\right)\epsilon_0 \;=\; O\left(\exp(-\kappa^{-\frac{s}{2\rho}} N)\right) \quad \textit{when } \tau = 0$$

while

$$f(\hat{x}) - f^* \;\leq\; \frac{\epsilon_0}{\left(\tau e^{-1}(c\kappa)^{-\frac{s}{2\rho}} \epsilon_0^{\frac{\tau}{\rho}} N + 1\right)^{-\frac{\rho}{\tau}}} \;=\; O\left(N^{-\frac{\rho}{\tau}}\right) \quad \textit{when } \tau > 0,$$

where $N = \sum_{k=1}^{R} t_k$ is the total number of iterations.

Proof. Our goal is to ensure that the target accuracy is reached at each restart, i.e.,

$$f(x_k) - f^* \leq \epsilon_k. \tag{20}$$

By assumption, (20) holds for $k = 0$. Assume that (20) is true at iteration $k - 1$, combining (Loja) with the complexity bound in (19); then

$$f(x_k) - f^* \leq \frac{\epsilon_k}{2} + \frac{c\kappa(f(x_{k-1}) - f^*)^{\frac{2}{r}}}{\epsilon_k^{\frac{2}{s}} t_k^{\frac{2\rho}{s}}} \frac{\epsilon_k}{2} \leq \frac{\epsilon_k}{2} + \frac{c\kappa}{t_k^{\frac{2\rho}{s}}} \frac{\epsilon_{k-1}^{\frac{2}{r}}}{\epsilon_k^{\frac{2}{s}}} \frac{\epsilon_k}{2}, \tag{21}$$

where $c = 8e^{2/e}$ using that $r^{2/r} \leq e^{2/e}$. By definition $\epsilon_k = e^{-\gamma k}\epsilon_0$, so to ensure (20) at iteration k this imposes

$$\frac{c\kappa e^{\gamma\frac{2}{r}} e^{-\gamma\left(\frac{2}{r} - \frac{2}{s}\right)k}}{t_k^{\frac{2\rho}{s}}} \epsilon_0^{\frac{2}{r} - \frac{2}{s}} \leq 1.$$

Rearranging terms in last inequality, using τ as defined in (2),

$$t_k \geq e^{\gamma\frac{1-\tau}{\rho}} (c\kappa)^{\frac{s}{2\rho}} \epsilon_0^{-\frac{\tau}{\rho}} e^{\frac{\gamma\tau}{\rho}k}.$$

Choosing $t_k = Ce^{\alpha k}$, where

$$C = e^{\gamma\frac{1-\tau}{\rho}} (c\kappa)^{\frac{s}{2\rho}} \epsilon_0^{-\frac{\tau}{\rho}} \qquad \text{and} \qquad \alpha = \frac{\gamma\tau}{\rho},$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

and using Lemma 2.1 then yields

$$f(\hat{x}) - f^* \leq \exp(-\gamma e^{-\frac{\gamma}{\rho}} (c\kappa)^{-\frac{s}{2\rho}} N)\epsilon_0 \tag{22}$$

when $\tau = 0$, while

$$f(\hat{x}) - f^* \leq \frac{\epsilon_0}{\left(\frac{\gamma\tau}{\rho} e^{-\frac{\gamma}{\rho}} (c\kappa)^{-\frac{s}{2\rho}} \epsilon_0^{\frac{\tau}{\rho}} N + 1\right)^{\frac{\rho}{\tau}}} \tag{23}$$

when $\tau > 0$. These bounds are minimal for $\gamma = \rho$, and the results follow. □

This bound matches the lower bounds for optimizing smooth and sharp functions up to constant factors [29, eq. 1.21]. Notice that, compared to [29], we can tackle nonsmooth convex optimization by using the universal fast gradient algorithm of [34]. The rate of convergence in Proposition 3.1 is controlled by the ratio between τ and ρ. If these are unknown, a log-scale grid search will not be able to reach the optimal rate, even if ρ is known, since we will miss the optimal rate by a constant factor; see Appendix C. If both are known, in the case of nonsmooth strongly convex functions, for example, a grid search on C recovers a nearly optimal bound. Finally, note that our bound is provided with respect to the number of iterations of the accelerated algorithms; the corresponding bounds in terms of numbers of calls to the oracles can be found by analyzing the line-search cost of the universal fast gradient method.

4. Restart with known primal gap.

Here, we assume that we know the optimum f^* of (P). This is the case, for example, in zero-sum matrix game problems or overparametrized least-squares without regularization. We assume again that f satisfies the generic smoothness assumption (Hölder) and the Łojasiewicz growth condition (Loja) on its initial sublevel set. We use again the universal gradient method $\mathcal{U}$. Here, however, we can stop the algorithm when it reaches the target accuracy as we know the optimum f^*; i.e., we stop after t_ϵ inner iterations such that $x = \mathcal{U}(x_0, \epsilon, t_\epsilon)$ satisfies $f(x) - f^* \leq \epsilon$ and write $x \triangleq \mathcal{C}(x_0, \epsilon)$, the output of this method.

Here we simply restart this method and decrease the target accuracy by a constant factor after each restart. Our scheme is described in Algorithm 3. The following proposition describes its convergence.

Algorithm 3 Restart with known primal gap for convex minimization.

```
Inputs : $x_0 \in \mathbb{R}^n, f^*, \gamma \geq 0, \epsilon_0 = f(x_0) - f^*$
for $k = 1, \dots, R$ do
    $\epsilon_k := e^{-\gamma}\epsilon_{k-1}, \qquad x_k := \mathcal{C}(x_{k-1}, \epsilon_k)$
end for
Output : $\hat{x} := x_R$
```

Proposition 4.1. *Let f be a convex function and $x_0 \in \mathbf{dom}\, f$. Denote $K = \{x : f(x) \leq f(x_0)\}$ and assume that $f \in \mathcal{H}_{s,L} \cap \mathcal{L}_{r,\mu}(K)$. Run Algorithm 3 from x_0 with parameter $\gamma = 1$. The precision reached at the last point $\hat{x}$ is given by*

$$f(\hat{x}) - f^* \leq \exp\left(-e^{-\frac{1}{\rho}} (c\kappa)^{-\frac{s}{2\rho}} N\right) (f(x_0) - f^*) = O\left(\exp(-\kappa^{-\frac{s}{2\rho}} N)\right) \quad \textit{when } \tau = 0$$

while

$$f(\hat{x}) - f^* \leq \frac{f(x_0) - f^*}{\left(\frac{\tau}{\rho} e^{-\frac{1}{\rho}} (c\kappa)^{-\frac{s}{2\rho}} (f(x_0) - f^*)^{\frac{\tau}{\rho}} N + 1\right)^{\frac{\rho}{\tau}}} = O\left(N^{-\frac{\rho}{\tau}}\right) \quad \textit{when } \tau > 0,$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where N is the total number of iterations, ρ is as defined in (18)*, κ and τ are as defined in* (2)*, and $c = 8e^{2/e}$ here. Those bounds are suboptimal to the best scheduled restarts by a factor of at most $e/2 \approx 1.3$.*

Proof. Given $\gamma \geq 0$, the linear convergence of our scheme is ensured by our choice of target accuracies ϵ_k. It remains to compute the number of iterations t_{ϵ_k} needed by the algorithm before the kth restart. Following the proof of Proposition 3.1, for $k \geq 1$ we know that the target accuracy is necessarily reached after

$$\bar{t}_k = e^{\gamma \frac{1-\tau}{\rho}} (c\kappa)^{\frac{s}{2\rho}} \epsilon_0^{-\frac{\tau}{\rho}} e^{\frac{\gamma\tau}{\rho} k}$$

iterations, such that $t_{\epsilon_k} \leq \bar{t}_k$. So Algorithm 3 achieves linear convergence while needing fewer inner iterates than the scheduled restart presented in Proposition 3.1; its convergence is therefore at least as good. For a given γ, bounds (22) and (23) follow with $\epsilon_0 = f(x_0) - f^*$. The dependency in γ of the restart scheme in bounds (22) and (23) is a factor

$$h(\gamma) = \gamma e^{-\gamma/\rho}$$

of the number of iterations, whose maximum value is reached for $\gamma = \rho$. Taking $\gamma = 1$ then leads to a bound suboptimal by a constant factor of at most $h(\rho)/h(1) \leq e/2 \approx 1.3$ for $\rho \in [1/2, 2]$, so running this scheme with $\gamma = 1$ makes it parameter-free while producing nearly optimal complexity. ∎

When f^* is known, the above restart scheme is adaptive, contrary to the general nonsmooth case in Proposition 3.1. It can even adapt to the local values of L or μ as we use a criterion instead of a preset schedule. Here, stopping using $f(x_k) - f^*$ implicitly yields optimal choices of C and τ. Note that this approach generalizes to algorithms for which a bound on the primal gap is available, as in the Frank–Wolfe algorithm; see [22].

5. Extensions.

Previous analyses of restart schemes only require bounds of the form (3) or (19). Our results extend then readily to non-Euclidean composite settings or structured objectives as presented below.

5.1. Composite problems and Bregman divergences.

We extend previous schemes to more general convex optimization problems of the form

$$\text{minimize } f(x) \triangleq \phi(x) + g(x), \tag{24}$$

where g is a simple convex function (the meaning of "simple" will be clarified later), ϕ is a convex s-smooth function with respect to a given norm $\|\cdot\|$ (potentially non-Euclidean) as defined below, and ϕ is defined on an open set containing $\mathbf{dom}\, g$, i.e., $\mathbf{dom}\, f = \mathbf{dom}\, g$.

Definition 5.1. *A function ϕ is s-smooth for a given $1 \leq s \leq 2$ with respect to a norm $\|\cdot\|$ if there exists a constant $L > 0$ such that*

$$\|\nabla\phi(x) - \nabla\phi(y)\|_* \leq L\|x - y\|^{s-1} \quad \textit{for all } x, y \in \mathbf{dom}\, \phi$$

and any subgradients $\nabla\phi(x) \in \partial\phi(x), \nabla\phi(y) \in \partial\phi(y)$ of ϕ at x, y, respectively, with $\|\cdot\|_*$ being the dual norm of $\|\cdot\|$. We denote by $\mathcal{H}_{s,L,\|\cdot\|}$ the set of s-smooth functions with respect to a norm $\|\cdot\|$ with parameter L.

To exploit the smoothness of ϕ with respect to a generic norm, we assume that we have access to a potential function h with $\mathbf{dom}(f) \subset \mathbf{dom}(h)$, strongly convex

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

with respect to the norm $\|\cdot\|$ with convexity parameter equal to one, which means

$$h(y) \geq h(x) + \nabla h(x)^T(y-x) + \frac{1}{2}\|x-y\|^2 \quad \text{for any } x, y \in \mathbf{dom}(h).$$

We define the Bregman divergence associated to h as, for given $x, y \in \mathbf{dom}(h)$,

$$D_h(y;x) = h(y) - h(x) - \nabla h(x)^T(y-x) \geq \frac{1}{2}\|x-y\|^2.$$

For $h(x) = \frac{1}{2}\|x\|_2^2$, we get $D_h(y;x) = \frac{1}{2}\|x-y\|_2^2$ and recover the Euclidean setting. Given the problem geometry, appropriate choices of potential functions and associated Bregman divergences can lead to significant performance gains in high dimensional settings. We now formally state the assumption that g is simple. Given $x, y \in \mathbf{dom}(f)$ and $\lambda \geq 0$ we assume that

$$\min_z \left\{ y^T z + g(z) + \lambda D_h(z;x) \right\} \tag{25}$$

can be solved either in a closed form or by some fast computational procedure.

This setting includes some constrained optimization problems, where g is the indicator function of a closed convex set, on which we can easily project the points. It also includes sparse optimization problems, such as the LASSO, where $\phi(x) = \|Ax-b\|_2^2$, with $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $g(x) = \lambda\|x\|_1$, with $\lambda \geq 0$ and $h(x) = \frac{1}{2}\|x\|_2^2$. To apply our analysis of restart schemes we need two things: an accelerated algorithm and an appropriate notion of sharpness. In the spirit of [4, 27], we thus introduce the notion of relative sharpness.

Definition 5.2. *A function f satisfies a relative Łojasiewicz growth condition with respect to a strictly convex function h on a set K if there exist $r \geq 1$, $\mu > 0$ such that*

$$\frac{\mu}{r} D_h(x, X^*)^{\frac{r}{2}} \leq f(x) - f^* \quad \textit{for any } x \in K, \tag{26}$$

where $D_h(x, X^*) = \min_{x^* \in X^*} D_h(x^*;x)$ and D_h is the Bregman divergence associated to h. We denote by $\mathcal{L}_{r,\mu,h}(K)$ the set of functions satisfying a relative Łojasiewicz growth condition with respect to h on a set K with parameters r, μ.

If $h = \frac{1}{2}\|x\|_2^2$, we recover the definition of the Łojasiewicz growth in the Euclidean setting (with slightly modified constants). This assumption is as generic as our first one in (Loja) as it is satisfied if f and h are subanalytic [6, Th. 6.4].

The algorithms are essentially the same as before, except that the distance to the set of minimizers is replaced by the Bregman divergence to the set of minimizers. We keep the same notation for the algorithms as the implementations are the same as presented in Appendix A. Formally, if ϕ is smooth with respect to a norm $\|\cdot\|$, the accelerated algorithm outputs after t iterations a point

$$x = \mathcal{A}(x_0, t) \qquad \text{such that} \qquad f(x) - f^* \leq \frac{cL}{t^2} D_h(x_0, X^*), \tag{27}$$

where $c = 8$ here. The next corollary generalizes Proposition 2.2.

Corollary 5.3. *Let $f = \phi + g$ be a composite convex function, $x_0 \in \mathbf{dom}\, f$, and $K = \{x : f(x) \leq f(x_0)\}$. Assume that $\phi \in \mathcal{H}_{2,L,\|\cdot\|}$ for a given norm $\|\cdot\|$, that $f \in \mathcal{L}_{r,\mu,h}(K)$ for h strongly convex with respect to $\|\cdot\|$, and that g is simple such that*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

problem (25) *can be computed efficiently. Run Algorithm* 1 *from* x_0 *with iteration schedule* $t_k = C^*_{\kappa,\tau} e^{\tau k}$ *for* $k = 1, \dots, R$, *where*

$$C^*_{\kappa,\tau} \triangleq e^{1-\tau}(c\kappa)^{\frac{1}{2}}(f(x_0)-f^*)^{-\frac{\tau}{2}},$$

with κ *and* τ *defined as in* (2) *and* $c = 8e^{2/e}$. *The precision reached at the last point* $\hat{x}$ *is given by*

$$f(\hat{x}) - f^* \le \exp\left(-2e^{-1}(c\kappa)^{-\frac{1}{2}}N\right)(f(x_0)-f^*) = O\left(\exp(-\kappa^{-\frac{1}{2}}N)\right) \quad \textit{when } \tau = 0,$$

while

$$f(\hat{x}) - f^* \le \frac{f(x_0)-f^*}{\left(\tau e^{-1}(f(x_0)-f^*)^{\frac{\tau}{2}}(c\kappa)^{-\frac{1}{2}}N+1\right)^{\frac{2}{\tau}}} = O\left(N^{-\frac{2}{\tau}}\right) \quad \textit{when } \tau > 0,$$

where $N = \sum_{k=1}^R t_k$ *is the total number of iterations.*

Proof. The proof of Proposition 2.2 only relies on the bound in (9) that combines the growth condition (Loja) with the complexity bound in (3). For the case with composite problems and Bregman divergences we combine (26) with the bound (27), which ensures that, for the kth iterate of the restart scheme, $f(x_k)-f^* \le \frac{c\kappa}{t_k^2}(f(x_{k-1})-f^*)^{\frac{2}{r}}$, with $c = 8e^{2/e}$ here. The rest of the proof follows as in Proposition 2.2. □

For general convex functions, given a target accuracy ϵ and an initial point x_0, the universal fast gradient method outputs after t iterations a point

$$x = \mathcal{U}(x_0, \epsilon, t) \quad \text{such that} \quad f(x) - f^* \le \frac{\epsilon}{2} + \frac{cL^{\frac{2}{s}} D_h(x_0, X^*)}{\epsilon^{\frac{2}{s}} t^{\frac{2\rho}{s}}} \frac{\epsilon}{2}, \tag{28}$$

where $c = 2^{\frac{5s-2}{s}}$ here. Then the following corollary generalizes Proposition 3.1.

COROLLARY 5.4. *Let* $f = \phi + g$ *be a composite convex function,* $x_0 \in \mathbf{dom}\, f$, *and* $K = \{x : f(x) \le f(x_0)\}$. *Assume that* $\phi \in \mathcal{H}_{s,L,\|\cdot\|}$ *for a given norm* $\|\cdot\|$, *that* $f \in \mathcal{L}_{r,\mu,h}(K)$ *for* h *strongly convex with respect to* $\|\cdot\|$, *and that* g *is simple such that problems* (25) *can be computed efficiently. Run Algorithm* 2 *from* x_0 *for given* $\epsilon_0 \ge f(x_0) - f^*$,

$$\gamma = \rho, \qquad t_k = C^*_{\kappa,\tau,\rho} e^{\tau k}, \quad \textit{where} \quad C^*_{\kappa,\tau,\rho} \triangleq e^{1-\tau}(c\kappa)^{\frac{s}{2\rho}} \epsilon_0^{-\frac{\tau}{\rho}},$$

where ρ *is defined as in* (18), κ *and* τ *are defined as in* (2), *and* $c = 16e^{2/e}$. *The precision reached at the last point* $\hat{x}$ *is given by*

$$f(\hat{x}) - f^* \ \le \ \exp\left(-\rho e^{-1}(c\kappa)^{-\frac{s}{2\rho}}N\right)\epsilon_0 \ = \ O\left(\exp(-\kappa^{-\frac{s}{2\rho}}N)\right) \quad \textit{when } \tau = 0,$$

while

$$f(\hat{x}) - f^* \ \le \ \frac{\epsilon_0}{\left(\tau e^{-1}(c\kappa)^{-\frac{s}{2\rho}}\epsilon_0^{\frac{\tau}{\rho}}N+1\right)^{\frac{\rho}{\tau}}} \ = \ O\left(N^{-\frac{\rho}{\tau}}\right) \quad \textit{when } \tau > 0,$$

where $N = \sum_{k=1}^R t_k$ *is the total number of iterations.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. The proof of Proposition 3.1 only relies on the bound in (21) that combines the growth condition (Loja) with the complexity bound in (19). For the case with composite problems and Bregman divergences we combine (26) with the bound (28), which ensures for the kth iterate of the restart scheme, $f(x_k) - f^* \leq \frac{\epsilon_k}{2} + \frac{c\kappa(f(x_{k-1})-f^*)^{\frac{2}{r}}}{\epsilon_k^{\frac{2}{s}} t_k^{\frac{2\rho}{s}}} \frac{\epsilon_k}{2}$ with $c = 16e^{2/e}$ here. The rest of the proof follows as in Proposition 3.1. □

The results regarding adaptive schemes and those for which f^* is known, i.e., Propositions 2.4 and 4.1, respectively, generalize similarly under the relative growth assumption. Those results apply then to generic $\ell_{1,p}$ regularized prediction problems where g is an $\ell_{1,p}$ norm and ϕ is a data-fitting term. Indeed, error bounds were proven to hold for those problems by Zhou, Zhang, and So [44]. Those error bounds are then equivalent to quadratic growth conditions, i.e., (Loja) with $r = 2$ [12]. Previous works demonstrate then linear convergence of proximal gradient descent [8]. Here the restart schemes allow us to get accelerated rates similar to those for smooth strongly convex problems. Note that adaptive schemes were also developed by Fercoq and Qu [14] in that case.

5.2. Smoothing nonsmooth problems. Our approach extends also to problems that can be smoothed, i.e., problems of the form

$$\text{minimize} \quad f(x) \triangleq \phi(Ax) + g(x), \tag{29}$$

where $A \in \mathbb{R}^{m\times n}$, g is a simple convex function, and ϕ is a nonsmooth convex function whose inf-convolution with some smooth convex function ψ can be computed analytically; i.e., one has access for any $\mu > 0$ to

$$\phi_{\mu\psi^\star}(x) = \sup_{u\in\mathbf{dom}\,\phi^\star} \left\{u^\top x - \phi^\star(u) - \mu\psi^\star(u)\right\}, \tag{30}$$

where for a function f we denote by $f^\star$ its convex conjugate. Those problems were notably considered by Nesterov [31], who proved that, though they a priori suffer from their nonsmoothness, they can still be solved in $O(\varepsilon)$ calls to an oracle by using their structure. Formally, we have access to an algorithm $\mathcal{S}$ that, given an initial point x_0 and a target accuracy ε, outputs after t iterations a point

(31)

$$x = \mathcal{S}(x_0, \epsilon, t) \quad \text{such that} \quad f(x) - f^* \leq \frac{\epsilon}{2} + \frac{cL^2_{\psi^\star,A} D_h(x, X^*)}{\epsilon^2 t^2}\frac{\epsilon}{2}, \quad \text{and} \quad f(x) \leq f(x_0),$$

where h is some potential function and $L_{\psi^\star,A}$ is a smoothing constant; see Appendix A for more details. The scheduled restarts of this algorithm will follow the same strategy as for the universal fast gradient method as presented in the following proposition.

PROPOSITION 5.5. *Let $f(x) = \phi(Ax) + g(x)$ be a nonsmooth objective that can be smoothed using* (30)*, $x_0 \in \mathbf{dom}\, f$, and $K = \{x : f(x) \leq f(x_0)\}$. Assume that we have access to a smoothing method $\mathcal{S}$ ensuring* (31) *for a given strongly convex function h and that $f \in \mathcal{L}_{r,\mu,h}(K)$. Given $\epsilon_0 \geq f(x_0) - f^*$, restart the method $\mathcal{S}$ such that for $k \geq 1$,*

$$x_k = \mathcal{S}(x_{k-1}, \epsilon_k, t_k), \qquad \epsilon_k = e^{-1}\epsilon_{k-1}, \qquad t_k = \tilde{C}^*_{\kappa,\tau} e^{\tau k}, \qquad \tilde{C}^*_{\kappa,\tau} \triangleq e^{1-\tau}(c\kappa)^{\frac{1}{2}}\epsilon_0^{-\tau},$$

where κ and τ are defined as in (2) *with $s = 1$ and $L_{\psi^*,A}$ in place of L.*

The precision reached at a point $\hat{x} = x_R$ after R restarts is given by

$$f(\hat{x}) - f^* \;\leq\; \exp\left(-e^{-1}(c\kappa)^{-\frac{1}{2}}N\right)\epsilon_0 \;=\; O\left(\exp(-\kappa^{-\frac{1}{2}}N)\right) \quad \textit{when } \tau = 0,$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

while

$$f(\hat{x}) - f^* \;\leq\; \frac{\epsilon_0}{\left(\tau e^{-1}(c\kappa)^{-\frac{1}{2}}\epsilon_0^{\tau} N + 1\right)^{\frac{1}{\tau}}} \;=\; O\left(N^{-\frac{1}{\tau}}\right) \quad \textit{when } \tau > 0,$$

where $N = \sum_{k=1}^{R} t_k$ *is the total number of iterations.*

Proof. The smoothing method has a bound of the same form as the universal fast gradient method; i.e., we have

$$x = \mathcal{S}(x_0, \epsilon, t) \qquad \text{such that} \qquad f(x) - f^* \leq \frac{\epsilon}{2} + \frac{cL^{\frac{2}{s}} D_h(x_0, X^*)^2}{\epsilon^{\frac{2}{s}} t^{\frac{2\rho}{s}}} \frac{\epsilon}{2},$$

with $L = L_{\psi^\star, A}$, $s = 1$ and $\rho = 1$ here. The optimal restart schedule and corresponding rates follow then from Proposition 3.1 by replacing $s = 1$ and $\rho = 1$. □

As for the universal fast gradient method, a grid search will not get optimal rates if r, and so τ, is unknown. However, if it is known, a grid search will ensure optimal rates up to a constant factor. It is illustrated for sparse recovery problems by Roulet, Boumal, and d'Aspremont [41].

If f^* is known, Proposition 4.1 is modified into the following proposition. Note that the resulting restart scheme is the one presented by Gilpin, Pena, and Sandholm [17] for zero-sum matrix games.

Proposition 5.6. *Let* $f(x) = \phi(Ax) + g(x)$ *be a nonsmooth objective that can be smoothed using* (30), $x_0 \in \mathbf{dom}\, f$, *and* $K = \{x : f(x) \leq f(x_0)\}$. *Assume that* f^* *is known, that we have access to a smoothing method* $\mathcal{S}$ *ensuring* (31) *for a given strongly convex function* h, *and that* $f \in \mathcal{L}_{r,\mu,h}(K)$. *Denoting* $\epsilon_0 = f(x_0) - f^*$, *consider the restart scheme defined by*

$$\begin{aligned} x_k &= S(x_{k-1}, \epsilon_k, t_k) \quad \textit{s.t.} \quad \epsilon_k = e^{-1}\epsilon_0, \\ t_k &= \operatorname{argmin}\{t : x = \mathcal{S}(x_{k-1}, \epsilon_k, t) \textit{ satisfies } f(x) - f^* \leq \epsilon_k\}. \end{aligned}$$

The precision reached at a point $\hat{x} = x_R$ *after* R *restarts is given by*

$$f(\hat{x}) - f^* \;\leq\; \exp\left(-e^{-1}(c\kappa)^{-\frac{1}{2}} N\right)\epsilon_0 \;=\; O\left(\exp(-\kappa^{-\frac{1}{2}} N)\right) \quad \textit{when } \tau = 0,$$

while

$$f(\hat{x}) - f^* \;\leq\; \frac{\epsilon_0}{\left(\tau e^{-1}(c\kappa)^{-\frac{1}{2}}\epsilon_0^{\tau} N + 1\right)^{\frac{1}{\tau}}} \;=\; O\left(N^{-\frac{1}{\tau}}\right) \quad \textit{when } \tau > 0,$$

where $N = \sum_{k=1}^{R} t_k$ *is the total number of iterations and* κ *and* τ *are defined as in* (2) *with* $s = 1$ *and* $L_{\psi^*, A}$ *in place of* L.

Proof. As in Proposition 4.1, the proposed scheme with a termination criterion on the gap cannot do worse than the optimal scheduled restart. The rate is then given by Proposition 5.5. □

6. Numerical results.

We illustrate our results by testing our adaptive restart schemes, the adaptive scheme *Adap* of section 2.2, and the scheme with stopping criterion on the primal gap *Crit* in section 4 on several problems to compare them against simple gradient descent (*Grad*), accelerated gradient methods (*Acc*), and the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

restart heuristic enforcing monotonicity (*Mono*) proposed by O'Donoghue and Candes [35]. For *Adap* we plot the convergence of the best method found by grid search to compare with the restart heuristic. This implicitly assumes that the grid search is run in parallel with enough servers. For *Crit* we use the optimal f^* found by another solver. This gives an overview of its performance when such information is available. All restart schemes were performed using the accelerated gradient with backtracking line search detailed in Appendix A, with large dots representing restart iterations.

In Figure 1, we solve classification problems with various losses on the UCI *Sonar* data set [1]. For the least-square loss on the *Sonar* data set, we observe much faster convergence of the restart schemes compared to the accelerated method. These results were already observed by O'Donoghue and Candes [35]. For the logistic loss, we observe that restart does not provide much improvement for a budget of 1000 iterations. For the hinge loss, we regularize by a squared norm and optimize the dual, which means solving a quadratic problem with box constraints. We observe here that the scheduled restart scheme converges much faster, while restart heuristics may be activated too late. We observe similar results for the LASSO problem. This highlights the benefits of a sharpness assumption for these last two problems. In general, *Crit* ensures the theoretical accelerated rate, but *Adap* exhibits more consistent behavior. Again, precisely quantifying sharpness from data/problem structure is a key open problem.

To account for the grid search effort, in Figure 2, we multiplied the number of iterations made by the *Adap* method by the size of the grid. This is for the LASSO problem on the *Sonar* data set with a grid step size of 4. This shows that the benefits of the restart schemes make the grid search effort acceptable both on paper and in practice. More clever grid search strategies for scheduled restarts run in parallel would even reduce this effort.

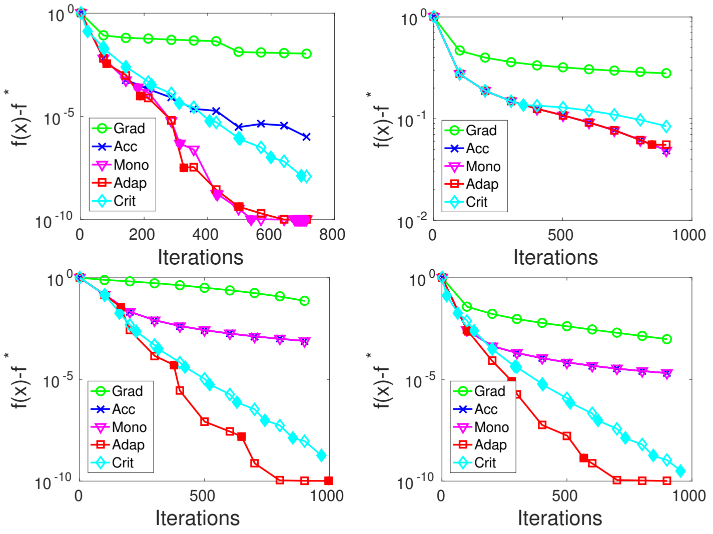

FIG. 1. *Sonar data set. From left to right: Least-square loss, logistic loss, dual SVM problem, and LASSO. We use adaptive restarts (Adap), gradient descent (Grad), accelerated gradient (Acc), and restart heuristic enforcing monotonicity (Mono). Large dots represent the restart iterations. Regularization parameters for dual SVM and LASSO were set to one.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

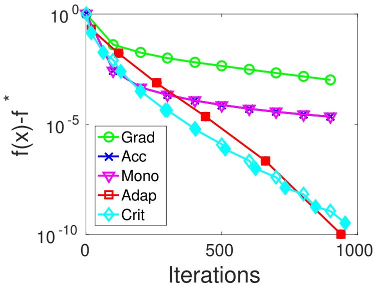

FIG. 2. *Comparison of the methods for the LASSO problem on Sonar dataset where the number of iterations of the Adaptive method is multiplied by the size of the grid. Grid search step size is set to* 4.

Appendix A. Algorithms and complexity bounds.

We present here the algorithms that we restart. In particular, we present a simplified version of the universal fast gradient method of Nesterov [34] and show how we can enforce monotonicity of the objective values while keeping the optimal rate. The classical accelerated algorithm for smooth convex function is then derived as a special case of the universal method.

In both cases the smoothness constant does not need to be known in advance. A line search is provided whose complexity can be bounded as in [34]. In practice, when restarting the algorithm we use the last smoothness constant provided by the algorithm.

A.1. Problem formulation.

We focus on composite optimization problems of the form

$$\text{minimize} \quad f(x) = \phi(x) + g(x) \tag{32}$$

where ϕ, g are proper lower semicontinuous convex functions on $\mathbb{R}^n$ and ϕ is defined on an open set containing $\mathbf{dom}\, g$, i.e., $\mathbf{dom}\, f = \mathbf{dom}\, g$. We denote by $\|\cdot\|$ a given norm on $\mathbb{R}^n$ and by $\|\cdot\|_*$ the dual norm of $\|\cdot\|$.

We assume that ϕ is s-smooth with respect to $\|\cdot\|$ for a given $s \in [1, 2]$, i.e., that there exists $L > 0$ such that

$$\|\nabla\phi(x) - \nabla\phi(y)\|_* \leq L\|x - y\|^{s-1} \tag{33}$$

for all $x, y \in \mathbf{dom}\, f$ and any $\nabla\phi(x) \in \partial\phi(x), \nabla\phi(y) \in \partial\phi(y)$. We assume that we have access to a function h which is differentiable on its domain $\mathbf{dom}\, h \supseteq \mathbf{dom}\, f$ and strongly convex with respect to the norm $\|\cdot\|$ with convexity parameter equal to one, i.e.,

$$h(y) \geq h(x) + \nabla h(x)^T(y - x) + \frac{1}{2}\|x - y\|^2 \quad \text{for any } x, y \in \mathbf{dom}\, h.$$

We define the Bregman divergence associated to h as, for given $x, y \in \mathbf{dom}\, h$,

$$D_h(y; x) = h(y) - h(x) - \nabla h(x)^T(y - x) \geq \frac{1}{2}\|x - y\|^2. \tag{34}$$

Finally, we assume that, for any $x, y \in \mathbf{dom}\, f$ and $\lambda \geq 0$, we can solve

$$\min_z y^T z + g(z) + \lambda D_h(z; x) \tag{35}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

either in a closed form or by some cheap computational procedure. In the following, we denote, for any $x, y \in \mathbf{dom}\, f$,

$$\ell_f(x; y) = \phi(y) + \nabla\phi(y)^\top (x - y) + g(x),$$

where $\nabla\phi(y) \in \partial\phi(y)$ is any subgradient of ϕ at y. This partial linearization of the objective is convex and satisfies, by convexity of ϕ,

$$\ell_f(x; y) \le f(x). \tag{36}$$

A.2. Universal fast gradient method. Our simplified version of the universal fast gradient method is presented in Algorithm 4. Proposition A.1 shows the convergence of Algorithm 4. Proposition A.2 ensures that the line searches terminate. The total number of oracle calls can be bounded as done by Nesterov [34], using a termination criterion; we only give the complexity in terms of iterations of the algorithm. Monotonicity is enforced by simply taking the best of the new and old iterates at each iteration.

PROPOSITION A.1. *Consider problem* (32) *where ϕ satisfies* (33) *with parameters* (s, L). *Algorithm* 4, *started at an initial point x_0 for a target accuracy ϵ and an initial estimate L_0, outputs after t iterations a point x_t such that*

$$f(x_t) - f^* \le \frac{\epsilon}{2} + \frac{2^{\frac{5s-2}{s}} L^{\frac{2}{s}} D_h(x_0, X^*)}{\epsilon^{\frac{2}{s}} t^{\frac{2\rho}{s}}} \frac{\epsilon}{2} \quad \textit{and} \quad f(x_t) \le f(x_0),$$

where $D_h(x, X^*) = \min_{x^* \in X^*} D_h(x^*; x)$ *with* $X^* = \operatorname{argmin}_x f(x)$.

Proof. Monotonicity of the objective values is ensured by (43). We fix in the following $x \in \mathbf{dom}\, f$. Consider the kth iteration of Algorithm 4 for $k \ge 1$. We have

$$\begin{aligned}
f(x_k) &\overset{(43)}{\le} f(\tilde{x}_k) \overset{(42)}{\le} \ell_f(\tilde{x}_k; y_{k-1}) + \frac{L_k}{2}\|\tilde{x}_k - y_{k-1}\|^2 + \frac{\epsilon\theta_k}{2} \\
&\overset{(i)}{\le} (1-\theta_k)\ell_f(x_{k-1}; y_{k-1}) + \theta_k \ell_f(z_k; y_{k-1}) + \frac{L_k\theta_k^2}{2}\|z_k - z_{k-1}\|^2 + \frac{\epsilon\theta_k}{2} \\
&\overset{(34)}{\le} (1-\theta_k)\ell_f(x_{k-1}; y_{k-1}) + \theta_k\left(\ell_f(z_k; y_{k-1}) + L_k\theta_k D_h(z_k; z_{k-1})\right) + \frac{\epsilon\theta_k}{2} \\
&\overset{(45)}{\le} (1-\theta_k)\ell_f(x_{k-1}; y_{k-1}) + \theta_k\left(\ell_f(x; y_{k-1}) + L_k\theta_k[D_h(x; z_{k-1}) - D_h(x; z_k)]\right) + \frac{\epsilon\theta_k}{2} \\
&\overset{(36)}{\le} (1-\theta_k)f(x_{k-1}) + \theta_k f(x) + L_k\theta_k^2[D_h(x; z_{k-1}) - D_h(x; z_k)] + \frac{\epsilon\theta_k}{2},
\end{aligned}$$

where in (i) we used that $\tilde{x}_k = (1-\theta_k)x_{k-1} + \theta_k z_k$, the convexity of $\ell_f(\cdot; y_{k-1})$, and $\tilde{x}_k - y_{k-1} = \theta_k(z_k - z_{k-1})$. Subtracting $f(x)$ on both sides and dividing by $L_k\theta_k^2$, we get

$$\frac{1}{L_k\theta_k^2}(f(x_k) - f(x)) \le \frac{1-\theta_k}{L_k\theta_k^2}(f(x_{k-1}) - f(x)) + D_h(x; z_{k-1}) - D_h(x; z_k) + \frac{\epsilon}{2L_k\theta_k}.$$

If $k = 1$, we have, using the initialization $\theta_1 = 1$, $z_0 = x_0$,

$$\frac{1}{L_1\theta_1^2}(f(x_1) - f(x)) \le D_h(x; x_0) - D_h(x; z_1) + \frac{\epsilon}{2L_1\theta_1}. \tag{37}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Otherwise we have, using the definition of θ_k in (41),

$$\frac{1}{L_k\theta_k^2}(f(x_k)-f(x)) \leq \frac{1}{L_{k-1}\theta_{k-1}^2}(f(x_{k-1})-f(x))+D_h(x;z_{k-1})-D_h(x;z_k)+\frac{\epsilon}{2L_k\theta_k}. \tag{38}$$

Using inequality (38) recursively from k to 2 and inequality (37) for $k=1$, we get

$$\begin{aligned}\frac{1}{L_k\theta_k^2}(f(x_k)-f(x)) &\leq D_h(x;x_0)-D_h(x;z_{k+1})+\sum_{j=1}^{k}\frac{\epsilon}{2L_j\theta_j}\\ &\overset{(47)}{=} D_h(x;x_0)-D_h(x;z_{k+1})+\frac{\epsilon}{2L_k\theta_k^2}.\end{aligned}$$

Finally, a bound on $L_k\theta_k^2$ can be found by combining Propositions A.2 and A.1 such that $L_k\theta_k^2 \leq \frac{2^{\frac{5s-2}{s}}L^{\frac{2}{s}}}{\epsilon^{\frac{2}{s}}k^{\frac{3s-2}{s}}}\frac{\epsilon}{2}$, and we get, denoting $\rho=\frac{3s-2}{2}$,

$$f(x_k)-f(x) \leq \frac{\epsilon}{2}+\frac{2^{\frac{5s-2}{s}}L^{\frac{2}{s}}D_h(x;x_0)}{\epsilon^{\frac{2}{s}}k^{\frac{2\rho}{s}}}\frac{\epsilon}{2}. \tag{39}$$

Taking $x \in \operatorname{argmin}_{x\in X^*} D_h(x;x_0)$ concludes the proof. □

PROPOSITION A.2. *Consider problem* (32) *where* ϕ *satisfies* (33) *with parameters* (s,L)*. The line searches of Algorithm* 4 *terminate with*

$$L_k \leq 2\epsilon^{\frac{s-2}{s}}L^{\frac{2}{s}}\theta_k^{\frac{s-2}{s}}.$$

Proof. Lemma A.4 ensures that the line search for x_1 stops for $\hat{L}_1 \geq \epsilon^{\frac{s-2}{s}}L^{\frac{2}{s}}$. Therefore, we have $L_1 \leq 2\epsilon^{\frac{s-2}{s}}L^{\frac{2}{s}}$. For $k \geq 2$, during the line search procedure, the parameter θ_k reads, denoting $a=\theta_{k-1}^2L_{k-1}/\hat{L}_k$, $\theta_k(\hat{L}_k)=\frac{-a+\sqrt{a^2+4a}}{2}=\frac{2}{1+\sqrt{1+4/a}}=\frac{2}{1+\sqrt{1+4\hat{L}_k/(\theta_{k-1}^2L_{k-1})}}$. Using again Lemma A.4, the stopping criterion (42) is then ensured if there exists $\hat{L}_k$ such that

$$\hat{L}_k \geq \left(\theta_k(\hat{L}_k)\epsilon\right)^{\frac{s-2}{s}}L^{\frac{2}{s}}=(2\epsilon)^{\frac{s-2}{s}}L^{\frac{2}{s}}\left(1+\sqrt{1+4\hat{L}_k/(\theta_{k-1}^2L_{k-1})}\right)^{\frac{2-s}{s}}. \tag{40}$$

Denote $c : x \to x-\alpha(1+\sqrt{1+\beta x})^{\gamma}$ for $\alpha>0, \beta>0, 0\leq\gamma\leq 1$. We have $\lim_{x\to+\infty}c(x)=+\infty$. Therefore, there exists $\hat{L}_k>0$ satisfying (40). Moreover, the line search terminates with $L_k \leq 2\left(\theta_k\epsilon\right)^{\frac{s-2}{s}}L^{\frac{2}{s}}$ as otherwise $L_k/2 > (\theta_k(L_k)\epsilon)^{\frac{s-2}{s}}L^{\frac{2}{s}} \geq (\theta_k(L_k/2)\epsilon)^{\frac{s-2}{s}}L^{\frac{2}{s}}$, and the line search would have stopped before. □

A.3. Accelerated algorithm.

The accelerated algorithm is obtained as a special case of the universal fast gradient algorithm for $s=2$ and a choice of $\epsilon=0$, i.e., $\mathcal{A}(x_0,t)=\mathcal{U}(x_0,t,0)$. Its rate follows from the one given by the universal fast gradient method as recalled below.

COROLLARY A.3. *Consider problem* (32) *where* ϕ *satisfies* (33) *with parameters* $(2,L)$*. Algorithm* 4 *with* $\epsilon=0$*, started at an initial point* x_0 *with an initial estimate* L_0*, outputs after* t *iterations a point* x_t *such that*

$$f(x_t)-f^* \leq \frac{8L}{t^2}D_h(x_0,X^*) \quad \textit{and} \quad f(x_t)\leq f(x_0),$$

where $D_h(x,X^*)=\min_{x^*\in X^*}D_h(x^*;x)$ *with* $X^*=\operatorname{argmin}_x f(x)$*.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm 4 Simplified universal fast gradient method $x = \mathcal{U}(x_0, t, \epsilon)$.

```
**Problem oracles:** Convex functions $\phi, g$, first order oracles $(x, y, \lambda) \to \operatorname{argmin}_z y^T z + g(z) + \lambda D_h(z; x)$ and $x \to \nabla\phi(x)$ with $\nabla\phi(x) \in \partial\phi(x)$
**Inputs:** Initial point $x_0$, number of iterations $t$, target accuracy $\epsilon$, smoothness estimate $L_0$
**Initialize:** $z_0 = x_0, \theta_1 = 1$
**for** $k = 1, \ldots, t$ **do**
    Initialize line search by $\hat{L}_k = L_{k-1}/2$
    **repeat**
        **if** $k > 1$ **then**
            Compute $\theta_k \geq 0$ s.t.
(41)            $$\frac{1-\theta_k}{\hat{L}_k \theta_k^2} = \frac{1}{L_{k-1}\theta_{k-1}^2}$$
        **end if**
        Compute
            $$\begin{aligned} y_{k-1} &= (1-\theta_k)x_{k-1} + \theta_k z_{k-1} \\ z_k &= \operatorname*{argmin}_z \ell_f(z; y_{k-1}) + \hat{L}_k \theta_k D_h(z; z_{k-1}) \\ \tilde{x}_k &= (1-\theta_k)x_{k-1} + \theta_k z_k \end{aligned}$$
        **if** $f(\tilde{x}_k) > \ell_f(\tilde{x}_k; y_{k-1}) + \frac{\hat{L}_k}{2}\|\tilde{x}_k - y_{k-1}\|^2 + \frac{\theta_k \epsilon}{2}$ **then** $\hat{L}_k \leftarrow 2\hat{L}_k$ **end if**
    **until**
(42)    $$f(\tilde{x}_k) \leq \ell_f(\tilde{x}_k; y_{k-1}) + \frac{\hat{L}_k}{2}\|\tilde{x}_k - y_{k-1}\|^2 + \frac{\theta_k \epsilon}{2}$$
    Pick any $x_k$ such that
(43)    $$f(x_k) \leq \min(f(\tilde{x}_k), f(x_{k-1}))$$
    Update $L_k = \hat{L}_k$
**end for**
**Output:** $x_t$
```

A.4. Lemmas for proving convergence.

LEMMA A.4 ([34, Lemma 2]). *Let ϕ satisfy* (33). *Then for any $\delta > 0$ and $\hat{L} \geq \left(\frac{2-s}{s}\frac{1}{\delta}\right)^{\frac{2-s}{s}} L^{\frac{2}{s}}$, or a fortiori any $\hat{L} \geq \delta^{\frac{s-2}{s}} L^{\frac{2}{s}}$, we have, for any $x, y \in \mathbf{dom}\,\phi$ and $\nabla\phi(x) \in \partial\phi(x)$,*

$$\phi(y) \leq \phi(x) + \nabla\phi(x)^\top (y-x) + \frac{\hat{L}}{2}\|x-y\|^2 + \frac{\delta}{2}. \tag{44}$$

LEMMA A.5 ([43, Property 1]). *For any proper lower semicontinuous convex function ψ, and any $z \in \mathbf{dom}\,h$, denote $z_+ = \operatorname{argmin}_x\{\psi(x) + D_h(x; z)\}$. Then, for any $x \in \mathbf{dom}\,h$,*

$$\psi(x) + D_h(x; z) \geq \psi(z_+) + D_h(z_+; z) + D_h(x; z_+). \tag{45}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

LEMMA A.6. *Consider two sequences* $(L_k)_{k\geq 1}, (\theta_k)_{k\geq 1}$ *initialized by* $L_1 > 0, \theta_1 = 1$ *and satisfying*

$$\text{(46)} \qquad \textit{for } k \geq 1 \qquad L_k \leq \frac{\alpha}{\theta_k^\beta}, \qquad \textit{and for } k \geq 2 \qquad \frac{1-\theta_k}{L_k\theta_k^2} = \frac{1}{L_{k-1}\theta_{k-1}^2},$$

for some $\alpha \geq 0, \beta \in [0,1]$, *with* $\theta_k \geq 0$. *Then, for any* $k \geq 1$,

$$\text{(47)} \qquad \frac{1}{L_k\theta_k^2} = \sum_{j=1}^{k} \frac{1}{L_j\theta_j} \qquad \textit{and} \qquad L_k\theta_k^2 \leq \frac{\alpha 2^{2-\beta}}{k^{2-\beta}}.$$

Proof. The first property of (47) is true for $k = 1$ since $\theta_1 = 1$. The definition of θ_k for $k \geq 2$ reads then $\frac{1}{L_k\theta_k^2} = \frac{1}{L_{k-1}\theta_{k-1}^2} + \frac{1}{L_k\theta_k}$, which shows the first property of (47) by induction.

For $k \geq 1$, denote $a_k = \frac{1}{L_k\theta_k}$ and $A_k = \sum_{j=1}^k a_j$, such that $A_k = \frac{1}{L_k\theta_k^2} = L_k a_k^2$. We have from (46) $L_k^{1-\beta} \leq \alpha a_k^\beta$. Therefore, $A_k^{1-\beta} \leq \alpha a_k^{2-\beta}$, which gives $A_k^{\frac{1-\beta}{2-\beta}} \leq \alpha^{\frac{1}{2-\beta}} a_k$. Denote $\gamma = \frac{1}{2-\beta}$ and $A_0 = 0$. Since $\gamma \geq 1/2$ and $A_k \geq A_{k-1}$, we have for any $k \geq 1$

$$A_k^\gamma - A_{k-1}^\gamma \geq \frac{A_k - A_{k-1}}{A_k^{1-\gamma} + A_{k-1}^{1-\gamma}} \geq \frac{A_k - A_{k-1}}{2A_k^{1-\gamma}} \geq \frac{1}{2\alpha^{\frac{1}{2-\beta}}}.$$

Therefore, we conclude that $A_k \geq \frac{k^{\frac{1}{\gamma}}}{2^{\frac{1}{\gamma}}\alpha} = \frac{k^{2-\beta}}{2^{2-\beta}\alpha}$. □

A.5. Smoothing nonsmooth problems. We present here the smoothing algorithm used in section 5. We recall the problem

$$\text{(48)} \qquad \text{minimize} \quad f(x) \triangleq \phi(Ax) + g(x)$$

where $A \in \mathbb{R}^{m\times n}$, g is a simple convex function, and ϕ is a nonsmooth convex function whose inf-convolution with some smooth convex function ψ can be computed analytically, i.e., one has access for any $\mu > 0$ to

$$\phi_{\mu\psi^\star}(x) = \inf_y \left\{\phi(y) + \mu\psi\left(\frac{x-y}{\mu}\right)\right\} = \sup_u \left\{u^\top x - \phi^\star(u) - \mu\psi^\star(u)\right\},$$

where for a function f we denote by $f^\star$ its convex conjugate; see [5] for a detailed exposition. For $\psi^\star$ 1-strongly convex with respect to a given norm $\|\cdot\|_\beta$ (i.e., ψ 1-smooth with respect to $\|\cdot\|_\beta^*$), the function $\phi_{\mu\psi^\star}$ is $1/\mu$-smooth with respect to the norm $\|\cdot\|_\beta^*$. Moreover, it approximates ϕ as (see, e.g., [36, Proposition 41])

$$\text{(49)} \quad \mu \inf_{u\in\mathbf{dom}\,\phi^\star} \psi^\star(u) \leq \phi(x) - \phi_{\mu\psi^\star}(x) \leq \mu \sup_{u\in\mathbf{dom}\,\phi^\star} \psi^\star(u) \qquad \text{for any } x \in \mathbf{dom}\,\phi.$$

The smoothed objective is a composite objective as in (32), i.e.,

$$\text{(50)} \qquad f_{\mu\psi^\star}(x) = \phi_{\mu\psi^\star}(Ax) + g(x),$$

where, denoting $\|A\|_{\alpha,\beta} = \sup_{\|x\|_\alpha\leq 1, \|u\|_\beta\leq 1} u^\top Ax$, we have that $x \to \phi_{\mu\psi^\star}(Ax)$ is $\|A\|_{\alpha,\beta}^2/\mu$ smooth with respect to a norm $\|\cdot\|_\alpha$ (see, e.g., [36, Lemma 42]). We can then apply the accelerated algorithm on (50) with a potential function h strongly

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

convex with respect to the norm $\|\cdot\|_\alpha$, assuming that g is simple such that we have access to oracles of the form (25). Precisely, given an initial point x_0 and a target accuracy ϵ, by applying the accelerated algorithm on (50) with $\mu = \epsilon/(2D)$, where $D = \sup_{u\in\mathbf{dom}\,\phi^*}\psi^*(u) - \inf_{u\in\mathbf{dom}\,\phi^*}\psi^*(u)$, we get after t iterations a point $\tilde{x}$ such that, for $x^*\in\operatorname{argmin} f$,

$$f(\tilde{x}) - f^* \overset{(49)}{\leq} f_{\mu\psi^*}(\tilde{x}) - f_{\mu\psi^*}(x^*) + D\mu \overset{(i)}{\leq} \frac{\epsilon}{2} + \frac{16D\|A\|^2_{\alpha,\beta}}{\epsilon t^2} D_h(x^*;x_0),$$

where in (i) we use the definition of μ and the convergence bound (39) of the accelerated algorithm ($\epsilon = 0$) applied on $x = x^*$. We denote then by $x = \mathcal{S}(x_0,\epsilon,t)$ any point such that $f(x)\leq \min\{f(\tilde{x}), f(x_0)\}$ such that it both satisfies the rate above and belongs to the initial sublevel set. The bound presented in (31) is obtained by taking $x^*\in\operatorname{argmin}_{x\in X^*} D_h(x;x_0)$ and defining $c = 32$ and $L_{\psi^*,A} = \sqrt{D}\|A\|_{\alpha,\beta}$.

Appendix B. Rounding issues. We presented convergence bounds for real sequences of iterate counts $(t_k)_{k=1}^{\infty}$, but in practice these are integer sequences. The following lemma details the convergence of our schemes for an approximate choice $\tilde{t}_k = \lceil t_k\rceil$.

LEMMA B.1. *Let x_k be a sequence whose kth iterate is generated from the previous one by an algorithm that needs t_k iterations, and denote by $N = \sum_{k=1}^R t_k$ the total number of iterations to output a point $\hat{x} = x_R$. Suppose setting $t_k = \lceil Ce^{\alpha k}\rceil$, $k = 1,\dots,R$, for some $C > 0$ and $\alpha\geq 0$ ensures that objective values $f(x_k)$ converge linearly, i.e.,*

$$f\left(x_k\right) - f^* \leq \nu e^{-\gamma k} \tag{51}$$

for all $k\geq 0$ with $\nu\geq 0$ and $\gamma\geq 0$. Then precision at the output is given by

$$f(\hat{x}) - f^* \leq \nu\exp(-\gamma N/(C+1)) \quad \textit{when } \alpha = 0$$

and, denoting $N' = N - \frac{\log\left((e^\alpha-1)e^{-\alpha}C^{-1}N+1\right)}{\alpha}$,

$$f(\hat{x}) - f^* \leq \frac{\nu}{(\alpha e^{-\alpha}C^{-1}N'+1)^{\frac{\gamma}{\alpha}}} \quad \textit{when } \alpha > 0.$$

Proof. At the Rth point generated, $N = \sum_{k=1}^R t_k$. If $t_k = \lceil C\rceil$, define $\epsilon = \lceil C\rceil - C$ such that $0\leq\epsilon<1$. Then $N = R(C+\epsilon)$, and injecting it into (51) at the Rth point, we get

$$f(\hat{x}) - f^* \leq \nu e^{-\gamma\frac{N}{C+\epsilon}} \leq \nu e^{-\gamma\frac{N}{C+1}}.$$

Now, if $t_k = \lceil Ce^{\alpha k}\rceil$, define $\epsilon_k = \lceil Ce^{\alpha k}\rceil - Ce^{\alpha k}$, such that $0\leq\epsilon_k<1$. On one hand, $N\geq\sum_{k=1}^R Ce^{\alpha k}$, such that $R\leq\frac{\log\left((e^\alpha-1)e^{-\alpha}C^{-1}N+1\right)}{\alpha}$. On the other hand,

$$\begin{aligned}
N = \sum_{k=1}^R t_k = \frac{Ce^\alpha}{e^\alpha-1}(e^{\alpha R}-1) + \sum_{k=1}^R \epsilon_k &\leq \frac{Ce^\alpha}{e^\alpha-1}(e^{\alpha R}-1) + R\\
&\leq \frac{Ce^\alpha}{e^\alpha-1}(e^{\alpha R}-1) + \frac{\log\left((e^\alpha-1)e^{-\alpha}C^{-1}N+1\right)}{\alpha},
\end{aligned}$$

such that $R\geq\frac{\log\left(\alpha e^{-\alpha}C^{-1}N'+1\right)}{\alpha}$. Injecting it into (51) at the Rth point the result follows. □

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Appendix C. Grid-search for universal restart schemes. We briefly explain why a grid search on the parameters for the general case $s \in [1,2]$ does not provide near-optimal bounds. Consider general restart schemes as presented in Algorithm 2 for a function $f \in \mathcal{H}_{s,L} \cap \mathcal{L}_{r,\mu}(K)$ with K the initial sublevel set of f at a given x_0. Assume that the decreasing factor is γ and the schedules have the form $t_k = Ce^{\alpha k}$ such that

$$t_k \geq e^{\gamma \frac{1-\tau}{\rho}} (c\kappa)^{\frac{s}{2\rho}} \epsilon_0^{-\frac{\tau}{\rho}} e^{\frac{\gamma\tau}{\rho} k},$$

which is $C \geq e^{\gamma \frac{1-\tau}{\rho}} (c\kappa)^{\frac{s}{2\rho}} \epsilon_0^{-\frac{\tau}{\rho}}$ and $\alpha \geq \frac{\gamma\tau}{\rho}$. Consider the case $\tau > 0$. Then following the proof of Proposition 3.1, we get that $f(x_R) - f^* \leq \gamma^k \epsilon_0$, and applying Lemma 2.1, we obtain a convergence rate

$$f(x_R) - f^* \leq \frac{\epsilon_0}{(\alpha e^{-\alpha} C^{-1} N + 1)^{\frac{\gamma}{\alpha}}},$$

where N is the total number of iterations. For this rate to be optimal or nearly optimal we need $\frac{\gamma}{\alpha} = \frac{\rho}{\tau}$. Any grid search on this ratio will then suffer from a constant factor such that we will not get a rate of the form $N^{-\frac{\rho}{\tau}}$ except if we know r and s, which gives us ρ/τ.

REFERENCES

[1] A. Asuncion and D. Newman, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml/index.php, 2007.

[2] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality*, Math. Oper. Res., 35 (2010), pp. 438–457.

[3] A. Auslender and J.-P. Crouzeix, *Global regularity theorems*, Math. Oper. Res., 13 (1988), pp. 243–253.

[4] H. H. Bauschke, J. Bolte, and M. Teboulle, *A descent lemma beyond Lipschitz gradient continuity: First-order methods revisited and applications*, Math. Oper. Res., 42 (2016), pp. 330–348.

[5] A. Beck and M. Teboulle, *Smoothing and first order methods: A unified framework*, SIAM J. Optim., 22 (2012), pp. 557–580, https://doi.org/10.1137/100818327.

[6] E. Bierstone and P. D. Milman, *Semianalytic and subanalytic sets*, Inst. Hautes Études Sci. Publ. Math., 67 (1988), pp. 5–42.

[7] J. Bolte, A. Daniilidis, and A. Lewis, *The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems*, SIAM J. Optim., 17 (2007), pp. 1205–1223, https://doi.org/10.1137/050644641.

[8] J. Bolte, T. P. Nguyen, J. Peypouquet, and B. W. Suter, *From error bounds to the complexity of first-order descent methods for convex functions*, Math. Program., 165 (2017), pp. 471–507.

[9] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program., 146 (2014), pp. 459–494.

[10] J. Burke and S. Deng, *Weak sharp minima revisited. I. Basic theory*, Control Cybernet., 31 (2002), pp. 439–469.

[11] J. V. Burke and M. C. Ferris, *Weak sharp minima in mathematical programming*, SIAM J. Control Optim., 31 (1993), pp. 1340–1359, https://doi.org/10.1137/0331063.

[12] D. Drusvyatskiy and A. S. Lewis, *Error bounds, quadratic growth, and linear convergence of proximal methods*, Math. Oper. Res., 43 (2018), pp. 919–948.

[13] O. Fercoq and Z. Qu, *Restarting Accelerated Gradient Methods with a Rough Strong Convexity Estimate*, preprint, https://arxiv.org/abs/1609.07358, 2016.

[14] O. Fercoq and Z. Qu, *Adaptive restart of accelerated gradient methods under local quadratic growth condition*, IMA J. Numer. Anal., 39 (2019), pp. 2069–2095.

[15] P. Frankel, G. Garrigos, and J. Peypouquet, *Splitting methods with variable metric for Kurdyka–Łojasiewicz functions and general convergence rates*, J. Optim. Theory Appl., 165 (2015), pp. 874–900.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[16] R. M. Freund and H. Lu, *New computational guarantees for solving convex optimization problems with first order methods, via a function growth condition measure*, Math. Program., 170 (2018), pp. 445–477.

[17] A. Gilpin, J. Pena, and T. Sandholm, *First-order algorithm with $\mathcal{O}(\log 1/\epsilon)$ convergence for ϵ-equilibrium in two-person zero-sum games*, Math. Program., 133 (2012), pp. 279–298.

[18] P. Giselsson and S. Boyd, *Monotonicity and restart in fast gradient methods*, in Proceedings of the 53rd IEEE Conference on Decision and Control, IEEE, Washington, DC, 2014, pp. 5058–5063.

[19] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Research Nat. Bur. Standards, 49 (1952), pp. 263–265.

[20] A. Iouditski and Y. Nesterov. *Primal-Dual Subgradient Methods for Minimizing Uniformly Convex Functions*, preprint, https://arxiv.org/abs/1401.1792, 2014.

[21] H. Karimi, J. Nutini, and M. Schmidt, *Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition*, in Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, New York, 2016, pp. 795–811.

[22] T. Kerdreux, A. d'Aspremont, and S. Pokutta, *Restarting Frank-Wolfe*, in Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Okinawa, Japan, 2019, pp. 1275–1283.

[23] Q. Lin and L. Xiao, *An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization*, in Proceedings of the 31st International Conference on Machine Learning, Vol. 32, Beijing, China, 2014, pp. 73–81.

[24] M. Liu and T. Yang, *Adaptive accelerated gradient converging method under Hölderian error bound condition*, in Advances in Neural Information Processing Systems 2017, Long Beach, CA, 2017, pp. 3104–3114.

[25] S. Łojasiewicz, *Une propriété topologique des sous-ensembles analytiques réels*, in Les Équations aux Dérivées Partielles (Paris, 1962), Éditions du Centre National de la Recherche Scientifique, Paris, France, 1963, pp. 87–89.

[26] S. Łojasiewicz, *Sur la géométrie semi- et sous-analytique*, Ann. Inst. Fourier (Grenoble), 43 (1993), pp. 1575–1595.

[27] H. Lu, R. M. Freund, and Y. Nesterov, *Relatively smooth convex optimization by first-order methods, and applications*, SIAM J. Optim., 28 (2018), pp. 333–354, https://doi.org/10.1137/16M1099546.

[28] O. L. Mangasarian, *A condition number for differentiable convex inequalities*, Math. Oper. Res., 10 (1985), pp. 175–179.

[29] A. Nemirovskii and Y. Nesterov, *Optimal methods of smooth convex minimization*, USSR Comput. Math. Math. Phys., 25 (1985), pp. 21–30.

[30] Y. Nesterov, *A method of solving a convex programming problem with convergence rate* $O(1/k^2)$, Soviet Math. Dokl., 27 (1983), pp. 372–376.

[31] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.

[32] Y. Nesterov, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161.

[33] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Springer, New York, 2004.

[34] Y. Nesterov, *Universal gradient methods for convex optimization problems*, Math. Program., 152 (2015), pp. 381–404.

[35] B. O'Donoghue and E. Candes, *Adaptive restart for accelerated gradient schemes*, Found. Comput. Math., 15 (2015), pp. 715–732.

[36] K. Pillutla, V. Roulet, S. M. Kakade, and Z. Harchaoui, *A smoother way to train structured prediction models*, in Advances in Neural Information Processing Systems 32, Montreal, Canada, 2018, pp. 4766–4778.

[37] B. T. Polyak, *Gradient methods for minimizing functionals*, Ž. Vyčisl. Mat. i Mat. Fiz., 3 (1963), pp. 643–653 (in Russian).

[38] J. Renegar, *Efficient First-Order Methods for Linear Programming and Semidefinite Programming*, preprint, https://arxiv.org/abs/1409.5832, 2014.

[39] J. Renegar and B. Grimmer, *A Simple Nearly-Optimal Restart Scheme for Speeding-Up First Order Methods*, preprint, https://arxiv.org/abs/1803.00151, 2018.

[40] S. M. Robinson, *An application of error bounds for convex programming in a linear space*, SIAM J. Control, 13 (1975), pp. 271–273, https://doi.org/10.1137/0313015.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[41] V. Roulet, N. Boumal, and A. d'Aspremont, *Computational complexity versus statistical performance on sparse recovery problems*, Inf. Inference, to appear, https://doi.org/10.1093/imaiai/iay020.

[42] W. Su, S. Boyd, and E. Candes, *A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights*, in Advances in Neural Information Processing Systems 27, Montreal, Canada, 2014, pp. 2510–2518.

[43] P. Tseng, *On Accelerated Proximal Gradient Methods for Convex-Concave Optimization*, Technical report, University of Washington, Seattle, WA, 2008.

[44] Z. Zhou, Q. Zhang, and A. M.-C. So, *$\ell_{1,p}$-norm regularization: Error bounds and convergence rate analysis of first-order methods*, in Proceedings of the 32nd International Conference on Machine Learning, Vol. 37, Lille, France, 2015, pp. 1501–1510.

Downloaded 01/15/20 to 129.93.16.3. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.