Complexity of Adagrad and other first-order methods for nonconvex optimization problems with bounds constraints

S. Gratton*, S. Jerad† and Ph. L. Toint‡

10 VI 2024

Abstract

A parametric class of trust-region algorithms for constrained nonconvex optimization is analyzed, where the objective function is never computed. By defining appropriate first-order stationarity criteria, we are able to extend the Adagrad method to the newly considered problem and retrieve the standard complexity rate of the projected gradient method that uses both the gradient and objective function values. Furthermore, we propose an additional iteration-dependent scaling with slightly inferior theoretical guarantees. In both cases, the bounds are essentially sharp, and curvature information can be used to compute the stepsize. Initial experimental results for noisy bound-constrained instances illustrate the benefits of the objective-free approach.

Keywords: First-order methods, objective-function-free optimization (OFFO), Adagrad, convergence bounds, evaluation complexity, second-order models.

1 Introduction

It is a truism to say that adaptive-gradient methods such as Adagrad, ADAM and cousins are at the heart of machine learning algorithms, and the literature covering them is vast (see [29, 17, 35, 27, 37, 34, 15] to cite only a few significant contributions). In line with current technology for neural network training, most of these minimization methods have been considered for nonconvex *unconstrained* problems. While this is adequate for a broad category of applications, it is however difficult to use them in a context where a priori information on the problem at hand is available, often in the form of constraints. Admittedly, these a can be taken into account by adding penalty terms to the loss/objective function, a technique often used in the Physically Informed Neural Networks (PINNs) branch of research (see [6, 36] for instance), but this introduces new hyper-parameters needing calibration and does not guarantee that constraints are strictly enforced. Moreover, a sufficiently severe penalization of the constraints typically deteriorates the problem's conditioning, possibly resulting in slow

*Université de Toulouse, INP, IRIT, Toulouse, France. Email: serge.gratton@enseeiht.fr. Work partially supported by 3IA Artificial and Natural Intelligence Toulouse Institute (ANITI), French "Investing for the Future - PIA3" program under the Grant agreement ANR-19-PI3A-0004"

†ANITI, Université de Toulouse, INP, IRIT, Toulouse, France. Email: sadok.jerad@enseeiht.fr

‡NAXYS, University of Namur, Namur, Belgium. Email: philippe.toint@unamur.be. Partially supported by ANITI.

convergence, especially if first-order methods are used. This suggests that there is a scope for a more direct approach. The present short paper proposes a step in that direction by considering adaptive gradient algorithms for nonconvex problems which have bound constraints.

Nonconvex optimization with bound constraints is not a new subject, but, to the author's knowledge, nearly all available algorithms use objective function values to ensure global convergence by assessing the effectiveness of a move from one iterate to the next (see, for instance, [13, Chapter 12] for a description of suitable trust-region methods and [8, Section 14.1] for an analysis of their complexity). This can be inconvenient in deep learning applications where sampling techniques cause significant noise ([22] provides a convincing illustration of this argument). Exceptions where the function values are not used in the globalization strategy are the papers [10] and [1]. The first studies a stochastic zero-th order (DFO) algorithm for problems with convex constraints based on the Malahanobis metric where the gradient is approximated by finite-differences and proves that, in expectation, the norm of the projected gradient (in that metric) decreases like $k^{-1/4}$, where k is the iteration counter. This study is motivated by applications in adversarial training (see the references in [10] for details). The second paper considers a purely first-order stochastic version of AMSgrad for weakly convex problems with convex constraints and obtains a similar rate of global convergence. It also mentions applications in adversarial training and reinforcement learning. Although both approaches are based on projections, no specific mention is made of the special case of bound constraints, despite the fact that such constraints are an integral part of efficient formulations of more complex problems with nonconvex constraints. Our aim in the present paper is to complete this picture by considering a general class of adaptive algorithms for general nonconvex problems including the popular Adagrad method, but also allowing the use of curvature information, should it be available at a reasonable cost. The algorithm we will introduce *never evaluates the objective function* (as is the case of the popular aforementioned Adagrad, ADAM and their variants). We refer to this feature by the OFFO acronym, for Objective Function Free Optimization.

Our proposal is deterministic[(1)] and is based on the combination of two existing algorithmic approaches: trust-region based projection methods [13, Chapter 12] (more general and more flexible than pure projected gradient methods) and a recent reinterpretation of adaptive-gradient methods as a particular class of (the same) trust-region techniques [23]. We claim that this combination, although only moderately complex technically, yields a useful class of optimization algorithms, in particular given the interest of the deep-learning community in methods capable of handling nonconvex functions and bound constraints.

A related approach is that using variants of Frank-Wolfe algorithm [19] to minimize a potentially nonconvex function subject to convex constraints, whose complexity was analyed in [28]. This method performs an objective-function-free linesearch along the direction from the current iterate to the minimizer of the first-order model in the feasible set. in this context, the main selling point of the Frank-Wolfe algorithm has always been (see [19]) that it replaces projection on the feasible set (a quadratic optmization problem) by a potentially cheaper linear programming subproblem whenever the feasible set is defined by a set of linear inequalities. Further developments (see [5]) have looked at stochastic versions of the method replacing the linesearch by either a fixed stepsize rule of a "function agnostic" rule based on iteration number only. Some of these variants ([40] or [32] and references therein) make use

[(1)] It is the authors' opinion that a study of the deterministic case is most useful for improving one's understanding of an algorithm's behaviour in more general stochastic contexts.

of the definition of the feasible set as a technique to bias the search direction towards (various forms of) sparse solutions. Our proposal differs from this line of work in that we propose a "function-aware" adaptive stepsize strategy (the trust-region mechanism, possibly using Adagrad-like adaptivity) using first-order information more locally (and thus probably more reliably), and that we also allow taking second-order information into account when possible. We also note that the (first-order) trust-region based step also results from an extremely cheap linear programming computation when the constraints are bounds on the variables, thus avoiding the general quadratic optimization cost of a projection in our case.

While adaptive-gradient methods for constrained nonconvex problems appear to be little explored, this is not the case for problems featuring a convex objective function, for which various methods have been proposed (see [2, 26, 18] for instance). Unfortunately, these methods are difficult to adapt to the nonconvex setting because they typically hinge on an acceleration technique that, so far, require convexity.

Our short paper is organized as follows. We first propose a class of OFFO adaptive-gradient methods for optimization with bound constraints in Section 2. We then specialize this class in Section 3 to derive a suitably modified Adagrad algorithm and analyze its evaluation complexity. We also propose, in Section 4 another specialization of our class, in the spirit of [23]. Some conclusions and perspectives are finally discussed in Section 6.

2 First-order minimization methods with bound constraints

The first optimization problem under consideration is given by

$$\min_{x\in\mathcal{F}} f(x) \tag{2.1}$$

where f is a smooth function from $\mathbb{R}^n$ to $\mathbb{R}$ and

$$\mathcal{F} = \big\{x \in \mathbb{R}^n \mid \ell_i \leq x_i \leq u_i \quad \text{for} \quad i \in \{1,\ldots,n\}\big\}. \tag{2.2}$$

The component-wise lower bounds $\{\ell_i\}_{i=1}^n$ and upper bounds $\{u_i\}_{i=1}^n$ in (2.2) satisfy $\ell_i \leq u_i$ for $i \in \{1,\ldots,n\}$. The values $\ell_i = -\infty$ and $u_i = +\infty$ are allowed. In what follows, we also assume the following:

AS.1: the objective function $f(x)$ is continuously differentiable;

AS.2: its gradient $g(x) \stackrel{\rm def}{=} \nabla_x^1 f(x)$ is Lipschitz continuous with Lipschitz constant $L \geq 0$, that is

$$\|g(x) - g(y)\| \leq L\|x - y\|$$

for all $x, y \in \mathbb{R}^n$;

AS.3: there exists a constant $f_{\rm low}$ such that, for all $x \in \mathcal{F}$, $f(x) \geq f_{\rm low}$.

AS.1, AS.2 and AS.3 are standard for the complexity analysis of optimization methods seeking first-order critical points, AS.3 guaranteeing in particular that the problem is well-posed. Note that we do not assume the gradients to be uniformly bounded, at variance with [38, 15, 39, 23].

Because the purpose of the method we are about to describe is to compute approximate first-order critical points for problem (2.1), we first review what we mean by this statement. Given a vector $x \in \mathcal{F}$, we first define

$$\chi_i(x,\alpha) = \delta_i(x,\alpha)|g(x)_i| \quad \text{where} \quad \delta_i(x,\alpha) = \max\big\{\delta \in [0,\alpha] \mid x - \delta\,\mathrm{sign}[g(x)_i]\,e_i \in \mathcal{F}\big\}, \tag{2.3}$$

with $g(x) = \nabla_x^1 f(x)$ and $\alpha \geq 0$. Observe that $\chi_i(x, \alpha)$ is the maximum decrease of the linear model $g(x)^T s$ achievable along the i-th coordinate vector while preserving feasibility and the inequality $|s_i| \leq \alpha$, that is

$$\chi_i(x, \alpha) = \left| \min \left\{ g(x)^T s \;\text{ such that }\; s = -\delta \, \mathrm{sign}(g_{i,k}) \, e_i, \; \delta \leq \alpha \;\text{ and }\; x + s \in \mathcal{F} \right\} \right| \tag{2.4}$$

Hence $\chi_i(x, 1) = |g_{i,k}|$ when $\mathcal{F} = \mathbb{R}^n$ (i.e., the problem is unconstrained) or when the distance from x to the bounds exceeds one along its i-th coordinate. Moreover $\chi_i(x) \stackrel{\rm def}{=} \chi_i(x, 1)$ can be interpreted as a continuous and backward stable first-order criticality measure for the one-dimensional problem

$$\min_{t \in \mathbb{R}} \left\{ f(x + te_i) \;\text{ such that }\; \ell_i \leq x_i + te_i \leq u_i \right\}$$

(see [24]). As it is standard to define an (unconstrained) ϵ-approximate first-order critical point at a point x such that

$$\|g(x)\| = \sqrt{\sum_{i=1}^{n} |g_i(x)|^2} \leq \epsilon,$$

we extend this notion here by defining a box-constrained ϵ-approximate first-order critical point as a vector $x \in \mathcal{F}$ such that

$$\sqrt{\sum_{i=1}^{n} |\chi_i(x)|^2} \leq \epsilon, \tag{2.5}$$

a backward stable condition whose left-hand side is continuous. If $\epsilon = 0$, (2.5) defines an exact first-order critical point.

Of interest here are iterative methods which generate a sequence of iterates $\{x_k\}_{k \geq 0}$ where the step from x_k to x_{k+1} depends on the gradient of the objective function at x_k and on algorithm-dependent *weights* $\{w_k = w(x_0, \ldots, x_k)\}$ whose main purpose is to control the step's size. These weights are assumed to be bounded below by a strictly positive constant, that is

AS.4: for each $i \in \{1, \ldots, n\}$ there exists a constant $\varsigma_i \in (0, 1]$ such that, $w_{i,k} \geq \varsigma_i$ for all $k \geq 0$,

Given an iterate x_k, we add a subscript k to and drop the argument x_k from quantities of interest to indicate they are evaluated for $x = x_k$, so that

$$g_k = \nabla_x^1 f(x_k), \quad \delta_{i,k}(\alpha) = \delta_i(x_k, \alpha) \quad \text{and} \quad \chi_{i,k} = \chi_i(x_k, 1)$$

for instance.

We now state the ASTR1B algorithm (for Adaptively Scaled Trust Region using 1rst order information with Bounds), a variant of the ASTR1 method of [23] capable of handling bound constraints.

Algorithm 2.1: ASTR1B

Step 0: Initialization. A starting point $x_0 \in \mathcal{F}$ is given. A constant $\tau \in (0, 1]$ is also given. Set $k = 0$.

Step 1: Define the trust-region. Compute $g_k = g(x_k)$ and define

$$\Delta_{i,k} = \frac{\chi_{i,k}}{w_{i,k}}. \tag{2.6}$$

Step 2: Hessian approximation. Select a symmetric Hessian approximation B_k.

Step 3: GCP. Compute a step s_k^L using

$$s_{i,k}^L = -\sum_{i=1}^{n} \delta_{i,k}(\Delta_{i,k}) \operatorname{sign}(g_{i,k}) \quad (i \in \{1, \ldots, n\}) \tag{2.7}$$

and a step s_k^Q given by

$$s_k^Q = \gamma_k s_k^L, \tag{2.8}$$

with

$$\gamma_k = \begin{cases} \min\left[1, \dfrac{|g_k^T s_k^L|}{(s_k^L)^T B_k s_k^L}\right] & \text{if} \quad (s_k^L)^T B_k s_k^L > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{2.9}$$

Finally select a step s_k such that

$$x_k + s_k \in \mathcal{F}, \tag{2.10}$$

$$|s_{i,k}| \leq \Delta_{i,k} \quad (i \in \{1, \ldots, n\}), \tag{2.11}$$

and

$$g_k^T s_k + \tfrac{1}{2} s_k^T B_k s_k \leq \tau \left(g_k^T s_k^Q + \tfrac{1}{2} (s_k^Q)^T B_k s_k^Q \right). \tag{2.12}$$

Step 4: New iterate. Define

$$x_{k+1} = x_k + s_k, \tag{2.13}$$

increment k by one and return to Step 1.

The ASTR1B algorithm, like ASTR1, allows the user to provide (and use) second-order information in B_k, should it be available. There is no need for B_k to be the exact Hessian of f at x_k, and (limited-memory) quasi-Newton approximations are acceptable. If B_k is chosen to be indentically zero on every iteration, the ASTR1B algorithm is then a "purely first-order" optimization method.

Albeit computing a step s_k satisfying (2.10)-(2.12) may seem formidable at first sight, this is actually a simple task, and even more so if $B_k = 0$. Indeed, the step s_k^L (which minimizes the linear model $g_k^T s$ in the intersection of the trust-region and the feasible set) is easily obtained from (2.7) since, using(2.3),

$$\delta_{i,k}(\Delta_{i,k}) = \begin{cases} \min\left[\Delta_{i,k}, u_i - x_i\right] & \text{if} \quad g_{i,k} < 0 \\ \min\left[\Delta_{i,k}, x_i - \ell_i\right] & \text{if} \quad g_{i,k} > 0 \\ 0 & \text{if} \quad g_{i,k} = 0 \end{cases} \tag{2.14}$$

for $i \in \{1, \ldots, n\}$. If $B_k = 0$, we have that $s_k = s_k^Q = s_k^L$. Otherwise the step s_k^Q is computed using (2.8) and (2.9), the latter possibly involving a single matrix vector product, and may be seen a "generalized Cauchy point" (see [23]), that is a step minimizing the quadratic model along a good first-order descent direction while preserving feasiblity with respect to the bounds and the trust-region. Any step preserving (2.10) and the bound (2.11), and decreasing the value of the quadratic model $g_k^T s + \frac{1}{2} s^T B_k s$ is then acceptable as s_k. Such an improved step may be computed for instance by using truncated projected Krylov techniques. This "generalized Cauchy point technique" has been first proposed in [11] and variants have been used extensively since[(2)], for instance in the LANCELOT [12], BOX-QUACAN [16], TRON [30] and GALAHAD [20] packages.

As stated, the algorithm does not impose any restriction on B_k, but it is clear that if curvature can be arbitrarily large, then the step can be arbitrarily small and convergence may be impeded. In the convergence theory below, we therefore assume that

AS.5: There exists a constant $\kappa_{\rm B} \geq 1$ such that $\|B_k\| \leq \kappa_{\rm B}$ for all $k \geq 0$.

Also observe that, at variance with "projected gradient methods" using function values (see [13, Section 12.2.1]), the ASTR1B algorithm does not use explicit projection onto the feasible set, nor does it perform approximate minimization of the quadratic model on the left-hand side of (2.12) along the "projected-gradient path" when B_k is nonzero. This has the advantage, which we judge crucial in potential neural-network training applications, of avoiding more than a single Hessian times vector product,

Because there are many possible choices for the weights $w_{i,k}$ in the ASTR1B algorithm, the latter may effectively be seen as a *class* of more specific methods, two of which we investigate in Sections 3 and 4.

All results in this paper crucially depend on the following two lemmas, which derive lower bounds on the decrease of the first-order Taylor model of f in the neighbourhood of an iterate x_k and on the value of f itself. The first is a substantially modified version of [13, Lemma 12.2.2], while the second is an adaptation of [23, Lemma 2.1].

[(2)] In the non-OFFO context.

Lemma 2.1 Suppose that AS.1 and AS.4 hold. Then, at each iteration $k \geq 0$ generated by the ASTR1B algorithm,

$$|g_{i,k}s^L_{i,k}| \geq \min\left[\frac{\chi^2_{i,k}}{w_{i,k}}, \chi_{i,k}\right] \quad (i \in \{1,\ldots,n\}) \tag{2.15}$$

and

$$|g_k^T s^L_k| \geq \varsigma_{\min}\|s^L_k\|^2 \tag{2.16}$$

where $\varsigma_{\min} = \min_{i\in\{1,\ldots,n\}} \varsigma_i$.

Proof. Let $i \in \{1,\ldots,n\}$ and observe that (2.3) and (2.7) ensure that $s^L_{i,k} = 0$ and $\chi_{i,k} = 0$ whenever $g_{i,k} = 0$, so that (2.15) and (2.16) trivially hold in this case. We therefore assume that $g_{i,k} \neq 0$ for the rest of the proof and let $d_{i,k} = u_i - x_{i,k}$ if $g_{i,k} < 0$, or $d_{i,k} = x_{i,k} - \ell_i$ if $g_{i,k} > 0$. Note that (2.7) implies that

$$|s^L_{i,k}| \leq \Delta_{i,k} \tag{2.17}$$

for all i and k.

If $|s^L_{i,k}| \geq 1$, then $d_{i,k} \geq 1$ and $|g_{i,k}| = \chi_{i,k}$. Moreover (2.7) implies that $\Delta_{i,k} \geq 1$. We thus obtain that

$$|g_{i,k}s^L_{i,k}| \geq \chi_{i,k}\min[d_{i,k}, \Delta_{i,k}] \geq \chi_{i,k}. \tag{2.18}$$

Suppose first that $|s^L_{i,k}| = \Delta_{i,k}$, then, using (2.6), AS.4 and (2.17),

$$|g_{i,k}s^L_{i,k}| = |g_{i,k}|\frac{\chi_{i,k}}{w_{i,k}} = \frac{\chi^2_{i,k}}{w_{i,k}} = w_{i,k}\frac{\chi^2_{i,k}}{w^2_{i,k}} \geq \varsigma_i\Delta^2_{i,k} = \varsigma_i(s^L_{i,k})^2. \tag{2.19}$$

If now $|s^L_{i,k}| = d_{i,k}$, one has $d_{i,k} \leq \frac{\chi_{i,k}}{w_{i,k}}$, which yields

$$|g_{i,k}s^L_{i,k}| = |g_{i,k}|d_{i,k} = \chi_{i,k}d_{i,k} = w_{i,k}\frac{\chi_{i,k}}{w_{i,k}}d_{i,k} \geq w_{i,k}d^2_{i,k} \geq \varsigma_i(s^L_{i,k})^2. \tag{2.20}$$

Suppose now that $|s^L_{i,k}| < 1$. If $|s^L_{i,k}| = \Delta_{i,k}$ and $d_i \geq 1$, then again $|g_{i,k}| = \chi_{i,k}$ and, using (2.6),

$$|g_{i,k}s^L_{i,k}| = \chi_{i,k}\Delta_{i,k} = \frac{\chi^2_{i,k}}{w_{i,k}}, \tag{2.21}$$

while using (2.6), AS.4 and (2.17) now gives that

$$|g_{i,k}s^L_{i,k}| = \chi_{i,k}\Delta_{i,k} = \frac{\chi^2_{i,k}}{w_{i,k}} = w_{i,k}\frac{\chi^2_{i,k}}{w^2_{i,k}} \geq \varsigma_i\Delta^2_{i,k} \geq \varsigma_i(s^L_{i,k})^2. \tag{2.22}$$

If $|s^L_{i,k}| = \Delta_{i,k}$ and $d_i < 1$, then $\chi_{i,k} = |g_{i,k}|d_{i,k}$ and, using (2.17) again,

$$|g_{i,k}s^L_{i,k}| = |g_{i,k}|\Delta_{i,k} = \chi_{i,k}\frac{\Delta_{i,k}}{d_i} \geq \chi_{i,k}\Delta_{i,k} = \frac{\chi^2_{i,k}}{w_{i,k}}. \tag{2.23}$$

We also deduce, using (2.6), (2.17) and AS.4, that, in this case,

$$|g_{i,k}s^L_{i,k}| = \frac{\chi^2_{i,k}}{w_{i,k}} = w_{i,k}\frac{\chi^2_{i,k}}{w^2_{i,k}} \geq \varsigma_i \Delta^2_{i,k} \geq \varsigma_i (s^L_{i,k})^2. \tag{2.24}$$

Finally, if $|s^L_{i,k}| < 1$ and $|s^L_{i,k}| < \Delta_{i,k}$, then one of the bounds on variable i must be active at $x_k + s^L_k$ and thus

$$|g_{i,k}s^L_{i,k}| = \chi_{i,k}, \tag{2.25}$$

and we obtain, using once more (2.6), (2.17), AS.4 and the inequality $|s^L_{i,k}| < 1$, that

$$|g_{i,k}s^L_{i,k}| = w_{i,k}\frac{\chi_{i,k}}{w_{i,k}} \geq \varsigma_i \Delta_{i,k} \geq \varsigma_i |s^L_{i,k}| \geq \varsigma_i (s^L_{i,k})^2. \tag{2.26}$$

Combining (2.18), (2.21), (2.23) and (2.25) then yields (2.15), while combining (2.19), (2.20), (2.22), (2.24), (2.26) and the inequality gives that, for all i and k,

$$|g_{i,k}s^L_{i,k}| \geq \varsigma_i (s^L_{i,k})^2. \tag{2.27}$$

But (2.7) implies that $g_{i,k}s^L_{i,k} < 0$ for all i and k, and thus that, for $k \geq 0$,

$$|g^T_k s^L_k| = \left| -\sum_{i=1}^n |g_{i,k}s^L_{i,k}| \right| = \sum_{i=1}^n |g_{i,k}s^L_{i,k}|.$$

The inequality (2.16) then follows by summing (2.27) for $i \in \{1, \ldots, n\}$ and using the definition of $\varsigma_{\min}$. □

Lemma 2.2 Suppose that AS.1, AS.2, AS.4 and AS.5 hold. Then, at each iteration $k \geq 0$ generated by the ASTR1B algorithm,

$$f(x_{k+1}) \leq f(x_k) - \frac{\tau \varsigma_{\min}}{2\kappa_B} \sum_{i=1}^n \min\left[\frac{\chi^2_{i,k}}{w_{i,k}}, \chi_{i,k}\right] + \tfrac{1}{2}(\kappa_{\rm B} + L) \sum_{i=1}^n \frac{\chi^2_{i,k}}{w^2_{i,k}}. \tag{2.28}$$

Proof. We now consider the quadratic model and suppose first that $(s^L_k)^T B_k s^L_k > 0$ and $\gamma_k < 1$. Then, we deduce from (2.8), (2.9), (2.16) and AS.5 that

$$\begin{aligned} g^T_k s^Q_k + \tfrac{1}{2}(s^Q_k)^T B_k s^Q_k &= \min_\gamma [g^T_k(\gamma s^L_k + \tfrac{1}{2}(\gamma s^L_k)^T B_k (\gamma s^L_k)] \\ &= -\frac{(g^T_k s^L_k)^2}{2(s^L_k)^T B_k s^L_k} \\ &\leq -\frac{\varsigma_{\min}}{2\kappa_B} |g^T_k s^L_k| \end{aligned} \tag{2.29}$$

If now $(s^L_j)^T B_j s^L_j \leq 0$ or $\gamma_j = 1$, then (2.8) and (2.7) give that

$$g^T_k s^Q_k + \tfrac{1}{2}(s^Q_k)^T B_k s^Q_k = g^T_k s^L_k + \tfrac{1}{2}(s^L_k)^T B_k s^L_k \leq \tfrac{1}{2} g^T_k s^L_k < 0 \tag{2.30}$$

and (2.29) then again follows from the bounds $\kappa_{\rm B} \geq 1$ and $\varsigma_i \leq 1$ for $i \in \{1, \ldots, n\}$ (see AS.4).

Successively using AS.1–AS.2, (2.12), (2.29) and (2.6) therefore yields that, for $j \geq 0$,

$$\begin{aligned} f(x_{j+1}) &\leq f(x_j) + g_j^T s_j + \tfrac{1}{2} s_j^T B_j s_j - \tfrac{1}{2} s_j^T B_j s_j + \tfrac{1}{2} L \|s_j\|^2 \\ &\leq f(x_j) + \tau \left(g_j^T s_j^Q + \tfrac{1}{2} (s_j^Q)^T B_j s_j^Q \right) + \tfrac{1}{2} (\kappa_{\rm B} + L) \|s_j\|^2 \\ &\leq f(x_j) - \frac{\tau \varsigma_{\min}}{2\kappa_B} \sum_{i=1}^{n} \min \left[\frac{\chi_{i,k}^2}{w_{i,k}}, \chi_{i,k} \right], + \tfrac{1}{2} (\kappa_{\rm B} + L) \sum_{i=1}^{n} \Delta_{i,j}^2 \\ &\leq f(x_j) - \frac{\tau \varsigma_{\min}}{2\kappa_B} \sum_{i=1}^{n} \min \left[\frac{\chi_{i,k}^2}{w_{i,k}}, \chi_{i,k} \right] + \tfrac{1}{2} (\kappa_{\rm B} + L) \sum_{i=1}^{n} \frac{\chi_{i,k}^2}{w_{i,k}^2}, \end{aligned}$$

which completes the proof. □

3 Adagrad with bound constraints and second-order models

In the unconstrained case, the well-known Adagrad algorithm [17] (in particular reframed as a trust-region method [23]) uses weights given by $w_{i,k} = \sqrt{\varsigma + \sum_{j=0}^{k} g_{i,k}^2}$. We pursue the analogy mentioned above between $g_{i,k}$ in the unconstrained case and $\chi_{i,k}$ in the box-constrained case by defining weights as follows. For given $\varsigma \in (0, 1]$ and $\vartheta \in (0, 1]$ let, for all $i \in \{1, \ldots, n\}$ and for all $k \geq 0$,

$$w_{i,k} \in \left[\sqrt{\vartheta}\, v_{i,k}, v_{i,k} \right] \quad \text{where} \quad v_{i,k} \stackrel{\rm def}{=} \left(\varsigma + \sum_{j=0}^{k} \chi_{i,j}^2 \right)^{\frac{1}{2}}. \tag{3.1}$$

The weights used by the well-know Adagrad algorithm are thus recovered by setting $\vartheta = 1$. *When applied to unconstrained problems,* ASTR1B *with (3.1), $\vartheta = 1$ and $B_k = 0$ is therefore identical to the (deterministic) Adagrad method, and thus generalizes this method to the box-contrained case.*

Note that, for all $k \geq 0$ and all $i \in \{1, \ldots, n\}$, the first part of (3.1) implies that

$$\frac{\chi_{i,k}}{w_{i,k}} \leq \frac{1}{\sqrt{\vartheta}} \quad \text{and} \quad \min \left[\frac{\chi_{i,k}^2}{w_{i,k}}, \chi_{i,k} \right] \geq \sqrt{\vartheta} \frac{\chi_{i,k}^2}{w_{i,k}} \tag{3.2}$$

so that we may rewrite the bound (2.28) as

$$f(x_{k+1}) \leq f(x_k) - \frac{\tau \vartheta \varsigma_{\min}}{2\kappa_B} \sum_{i=1}^{n} \frac{\chi_{i,k}^2}{w_{i,k}} + \tfrac{1}{2} (\kappa_{\rm B} + L) \sum_{i=1}^{n} \frac{\chi_{i,k}^2}{w_{i,k}^2}, \tag{3.3}$$

from which we deduce, by summing over iterations 0 to k, that

$$\frac{\tau \vartheta \varsigma}{2\kappa_B} \sum_{j=0}^{k} \sum_{i=1}^{n} \frac{\chi_{i,j}^2}{w_{i,j}} \leq f(x_0) - f(x_{k+1}) + \tfrac{1}{2} (\kappa_{\rm B} + L) \sum_{j=0}^{k} \sum_{i=1}^{n} \frac{\chi_{i,j}^2}{w_{i,j}^2}. \tag{3.4}$$

Theorem 3.1 Suppose that AS.1–AS.3 and AS.5 hold and that the ASTR1B algorithm is applied to problem (2.1) with its 'Adagrad-like' weights given by (3.1). Then

$$\underset{j\in\{0,\ldots,k\}}{\text{average}} \sum_{i=1}^{n} \chi_{i,j}^2 \leq \frac{\kappa_{\rm adag}}{k+1}, \tag{3.5}$$

with $\Gamma_0 \stackrel{\rm def}{=} f(x_0) - f_{\rm low}$ and

$$\kappa_{\rm adag} = \max\left\{\varsigma, \frac{1}{2}e^{\frac{2\Gamma_0\vartheta}{n(\kappa_B+L)}}, \frac{1}{2\varsigma}\left(\frac{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}{\tau\vartheta^{\frac{5}{2}}}\right)^2 \left|W_{-1}\left(-\frac{\tau\varsigma\vartheta^{\frac{5}{2}}}{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}\right)\right|^2\right\}, \tag{3.6}$$

where W_{-1} is the second branch of the Lambert function [14].

Proof. Given the inequality (3.4), the proof of the theorem is a variation on that of [23, Theorem 3.2], where the i-th component of the gradient $g_{i,k}$ is replaced by $\chi_{i,k}$, the criticality measure for the i-th variable. It is detailed in the Appendix for the sake of completeness. □

As noted in [23], it is possible to give a weaker but more explicit bound on $\kappa_{\rm adag}$ by using an upper bound on the value of the involved Lambert function. This can be obtained from [9, Theorem 1] which states that, for $x > 0$,

$$\left|W_{-1}(-e^{-x-1})\right| \leq 1 + \sqrt{2x} + x. \tag{3.7}$$

Remembering that, for γ_1 and γ_2 given by (A.5), $\log\left(\frac{\gamma_2}{\gamma_1}\right) \geq \log(3) > 1$ and choosing $x = \log\left(\frac{\gamma_2}{\gamma_1}\right) - 1 > 0$ in (3.7) then yields that

$$\left|W_{-1}\left(-\frac{\gamma_1}{\gamma_2}\right)\right| \leq \log\left(\frac{\gamma_2}{\gamma_1}\right) + \sqrt{2\left(\log\left(\frac{\gamma_2}{\gamma_1}\right) - 1\right)}. \tag{3.8}$$

It is also possible to extend the definition of s_k^L in (2.7) by premultiplying it by a stepsize $\alpha_k \in [\alpha_{\min}, 1]$ for some $\alpha_{\min} \in (0, 1]$. Our results again remain valid (with modified constants).

Observe that, if the algorithm is terminated as soon as (2.5) (as we argued in Section 2), it must stop at the latest at iteration

$$k = \kappa_{\rm adag}^2 \epsilon^{-2}. \tag{3.9}$$

This corresponds the the ϵ-order $\mathcal{O}(\epsilon^{-2})$, the standard complexity order of first-order methods using function values (see [8, Chapter 2]).

It also results from [23, Theorem 3.3] (applied for $\mu = \frac{1}{2}$) that the complexity bound given by Theorem 3.1 is essentially sharp (in the sense of [7]), because this is the case for its unconstrained variant. More precisely, we have the following result.

Theorem 3.2 The bound (3.5) is essentially sharp in that, for each $\eta \in (0,1]$, there exists a univariate function f_η satisfying AS.1-AS.3 and AS.5 such that, when applied to minimize f_η without constraints from the origin, the ASTR1B algorithm with (3.1), $B_k = 0$ and $\vartheta = 1$ produce a sequence of gradient norms given by $\chi_{1,0} = g_0 = -2$ and $\chi_{1,k}^2 = g_k^2 = \frac{1}{k^{1+2\eta}}$ for $k \geq 1$.

Proof. See [23, Theorem 3.3] and note that

$$\underset{j\in\{0,\ldots,k\}}{\text{average}}\ \chi_{1,k}^2 = \frac{4}{k+1} + \frac{1}{k+1}\sum_{j=1}^{k}\frac{1}{j^{1+2\eta}} \leq \frac{4+\zeta(1+2\eta)}{k+1},$$

where $\zeta(\cdot)$ is the Riemann zeta function, which is well-defined and finite for arguments exceeding one. □

4 A "diminishing stepsizes" variant

We assume, in this section, that the weights $w_{i,k}$ are chosen such that, for some power parameter $0 < \nu \leq \mu < 1$, all $i \in \{1,\ldots,n\}$ and some constants $\varsigma_i \in (0,1]$ and $\theta > 0$,

$$\max[\varsigma_i, v_{i,k}]\,(k+1)^\nu \leq w_{i,k} \leq \max[\varsigma_i, v_{i,k}]\,(k+1)^\mu \quad (k \geq 0), \tag{4.1}$$

where, for each i, the $v_{i,k}$ satisfy the properties that

$$v_{i,k+1} > v_{i,k} \quad \text{implies that} \quad v_{i,k+1} \leq |\chi_{i,k+1}| \tag{4.2}$$

and

$$v_{i,k} \geq \sqrt{\vartheta}|\chi_{i,k}| \tag{4.3}$$

for some $\vartheta \in (0,1]$. The motivation for considering these alternative class of variants is the interesting numerical performance [23] of the choice

$$v_{i,k} = \max_{j\in\{0,\ldots,k\}} |\chi_{i,j}|$$

which satisfies (4.2) and (4.3). This choice of weights ensures that

$$\frac{\chi_{i,j}}{w_{i,j}} \leq \frac{1}{\sqrt{\vartheta}}$$

and, as in the previous section, (2.28) implies (3.4). We then obtain the following result.

Theorem 4.1 Suppose that AS.1, AS.2, AS.3 and AS.5 hold and that the ASTR1B algorithm is applied to problem (2.1), where the weights $w_{i,k}$ are chosen in accordance with (4.1), (4.2) and (4.3). Then, for any $\eta \in (0, \tau\vartheta\varsigma_{\min})$ and

$$j_\eta \stackrel{\rm def}{=} \left(\frac{\kappa_{\rm B}(\kappa_{\rm B}+L)}{\varsigma_{\min}(\tau\varsigma_{\min}-\eta)} \right)^{\frac{1}{\nu}}, \tag{4.4}$$

there exist a constant $\kappa_\diamond$, a subsequence $\{k_\ell\} \subseteq \{k\}_{j_\eta+1}^{\infty}$ and an index k_ς (where $\kappa_\diamond$ and k_ς only depend on the problem and the algorithmic constants) such that, for all $k_\ell \geq k_\varsigma$,

$$\min_{j\in\{0,\ldots,k_\ell\}} \sum_{i=1}^{n} \chi_{i,j}^2 \leq \kappa_\diamond \frac{(k_\ell+1)^\mu}{k_\ell - j_\eta} \leq \frac{2\kappa_\diamond(j_\eta+1)}{k_\ell^{1-\mu}}. \tag{4.5}$$

Proof. The proof is again a variation of the proof of [23, Theorem 4.2] where the i-th component of the gradient $g_{i,k}$ is replaced by $\chi_{i,k}$, the criticality measure for the i-th variable. The proof is also simpler than that in [23] because our choice of weights $w_{i,k}$ is slightly more restrictive. The details are once more given in the Appendix. □

Because j_θ and k_ς only depends on ν and problem's constants, Theorem 4.1 gives some indication on the rate of convergence for iterations beyond an *a priori* computable iteration index. The formulation of the theorem is nevertheless weaker than that of Theorem 3.1 since (4.5) only holds for iterates along the subsequence $\{k_\ell\}$ and there is no guarantee that the bound given by the right-hand-side is valid at other iterations. Fortunately, the index k_ℓ in this right-hand side is an index in the complete sequence of iterates, which does not depend on the subsequence. A stronger result not involving subsequences has been proved in the unconstrained case under the stronger assumption that gradients remain uniformly bounded [21, Theorem 4.1], and its extension to the bound constrained case is possible much in the same fashion that Theorem 4.2 of [23] has been adapted here.

When μ and ν tend to zero, the k-order of convergence beyond j_θ (as stated by (4.5)) tends to $\mathcal{O}(1/\sqrt{k_\ell})$, which the order derived for the methods of the previous section and is the standard k-order for first-order methods using evaluations of the objective function, albeit the value of j_θ might increase. Moreover this result is essentially sharp, as implied by the following theorem.

Theorem 4.2 The bound (4.5) is essentially sharp in that, for any $\omega > \frac{1}{2}(1-\nu)$, there exists a univariate function $f_\omega(x)$ satisfying AS.1–AS.3 and AS.5 such that the ASTR1B algorithm with (4.1), $\mu = \nu$ and $B_k = 0$ applied to this function without constraints produces a sequence of first-order criticality measures given by $\chi_{1,k} = \|g_k\| = \frac{1}{(k+1)^\omega}$.

Proof. See [23, Theorem 4.3] and the comments in the proof of Theorem 3.2 above. □

	Noise	ASTR1B(0)	ASTR1B(1)	ASTR1B(3)	TRInf(0)	TRInf(1)	TRInf(3)
Performance	0%	0.54	0.48	0.48	0.87	0.90	0.90
Reliability	0%	100.0%	95.5%	95.5%	90.9%	90.9%	90.9%
	1%	99.6%	100.0%	100.0%	5.5%	5.9%	5.0%
	5%	100.0%	99.6%	99.6%	5.0 %	5.0%	5.5%
	15%	100.0%	100.0%	100.0%	not run	not run	not run
	25%	99.1%	100.0%	98.2%	not run	not run	not run

Table 1: Performance and reliability of ASTR1B variants and corresponding trust-region algorithms as a function of relative noise

5 Numerical illustration

Without any ambition of completeness, we now illustrate the behaviour of the ASTR1B algorithm on a small set of 22 bound-constrained problems from the CUTEst collection (as made available in Matlab through S2MPJ [25]). The problems under consideration are given in Table A.2 in Appendix 6. We compare six algorithms.

ASTR1B(0): the ASTR1B algorithm using (3.1) and $B_k = 0$ (i.e. momentumless Adagrad with bounds)

ASTR1B(1): the ASTR1B algorithm using (3.1) and B_k given by a limited BFGS update [31] using one secant pair;

ASTR1B(3): the ASTR1B algorithm using (3.1) and B_k given by a limited BFGS update using three secant pairs;

TRInf(0): the standard trust-region algorithm using an ℓ_∞ trust-region and a purely linear model ($B_k = 0$);

TRInf(1): the standard trust-region algorithm using an ℓ_∞ trust-region and B_k given by a limited BFGS update using one secant pair;

TRInf(3): the standard trust-region algorithm using an ℓ_∞ trust-region and B_k given by a limited BFGS update using three secant pairs.

We also ran these algorithms for an accuracy level $\epsilon = 10^{-3}$ on our small illustrative problem set using 0%, 5%, 15% and 25% of relative Gaussian noise on the gradients (and objective function for the trust-region algorithms), in order to explore their sensitivity to random perturbations. The details of our experimental setup are presented in Appendix 6. Our choice of considering the same level of noise on the objective-function value (when used) and the gradient is motivated by "finite-sum" applications such as deep learning where both the objective-function value and the gradient are computed by sampling. Although the authors are aware that better results can be obtained for the trust-region algorithms if one is ready to substantially increase the sample size for the objective function (see [4] or [3] for instance), this is unnatural and considerably more expensive in the sampling context.

Our results are reported in Table 1. The first line of the table gives the compared performance of the six methods of interest in the absence of noise, measured as the area of the method's curve in a performance profile counting the number of iterations to convergence for

the six methods and for an abscissa between 0 and 10, divided by 10. Hence the closest to one, the best performance (see [33] or [22] for other uses of this synthetic measure). Clearly, the trust-region methods using function values outperform the ASTR1B variants, whose performance could be qualified of mediocre. But the picture changes completely when one considers the reliability of the algorithms, reported in the remaining lines of the table as the percentage of successfully solved problems. From this point of view, the reliability of the trust-region methods becomes extremely poor as soon as some noise is added[(3)], while that of the ASTR1B remains remarkably constant. This observation conforts a similar conclusion obtained in [22] for unconstrained problems.

6 Conclusions

We have combined existing optimization techniques to propose an adaptive gradient algorithm for minimizing general nonconvex objective functions subject to bound constraints and allowing the use of curvature information. We have also analyzed its evaluation complexity and shown that it is (in order) identical to that of both standard algorithms using function values for the same problem and adaptive-gradient algorithms for the unconstrained one. Interestingly, our theoretical results do not require weak-convexity.

We view this proposal as a useful step towards efficient first-order OFFO algorithms for nonconvex problems with nonconvex constraints, a subject of importance in "constraint-aware" machine learning applications such as PINNs, adversarial training and other approaches. Another interesting application of the ideas developped here is the use of sparsity-inducing norms (such as those discussed in [5]) to define the trust region geometry.

Acknowledgements

References

[1] A. Alacaoglu, Y. Malitsky, and V. Cevher. Convergence od adaptive algorithms for constrained weakly-convex optimization. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NEURIPS 2021)*, 2021.

[2] F. Bach and K. Y. Levy. A universal algorithm for variational inequalities adaptive to smoothness and noise. In *Conference on Leaning Theory (COLT)*, pages 164–194, 2019.

[3] S. Bellavia, G. Gurioli, B. Morini, and Ph. L. Toint. The impact of noise on evaluation complexity: The deterministic trust-region case. *Journal of Optimization Theory and Applications*, 196(2):700–729, 2023.

[4] J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg. Convergence rate analysis of a stochastic trust region method via supermartingales. *INFORMS Journal on Optimization*, 1(2):92–119, 2019.

[5] G. Braun, A. Carderera, C. Combettes, H. Hassani, A. Karbasi, A. Mokhtari, and S. Pokutta. Conditional gradient methods. arXiv;2211.14103v2, 2023.

[6] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021.

[7] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. In B. Sirakov, P. de Souza, and M. Viana, editors, *Invited Lectures, Proceedings of the 2018 International Conference of Mathematicians (ICM 2018), vol. 4, Rio de Janeiro*, pages 3729–3768. World Scientific Publishing Co Pte Ltd, 2018.

[(3)] A typical failure of the trust-region approach occurs when noise of the objective-function values results in an erratic ratio of achieved to predicted reduction, itself causing a decrease of the trust-region radius to a point where the algorithm is stalled.

[8] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation complexity of algorithms for nonconvex optimization*. Number 30 in MOS-SIAM Series on Optimization. SIAM, Philadelphia, USA, June 2022.

[9] I. Chatzigeorgiou. Bounds on the Lambert function and their application to the outage analysis of user cooperation. *IEEE Communications Letters*, 17(8):1505–1508, 2013.

[10] X. Chen, S. Liu, X. Li, X. Lin, M. Hong, and D. Cox. ZO-AdaMM: Zeroth-order qadaptive momentum method for black-box optimization. In *Proceedings of the 33rd Conferencve on Neural Information Processing Systems (NEURIPS 2019), Vancouver, Canada*, 2019.

[11] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, 25(182):433–460, 1988. See also same journal 26:764–767, 1989.

[12] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A)*. Number 17 in Springer Series in Computational Mathematics. Springer Verlag, Heidelberg, Berlin, New York, 1992.

[13] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Number 1 in MOS-SIAM Optimization Series. SIAM, Philadelphia, USA, 2000.

[14] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5:329—359, 1996.

[15] A. Défossez, L. Bottou, F. Bach, and N. Usunier. A simple convergence proof for Adam and Adagrad. *Transactions on Machine Learning Research*, October 2022.

[16] M. A. Diniz-Ehrhardt, M. A. Gomes-Ruggiero, and S. A. Santos. Comparing the numerical performance of two trust-region algorithms for large-scale bound-constrained minimization. *Investigación Operativa*, 1998.

[17] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12, July 2011.

[18] A. Ene, H. L. Nguyen, and A. Vladu. Adaptive gradient methods for constrained convex optimization and variational inequalities. In *The 35-th AAAI Conference on Artificial Intelligence*, pages 7314–7321, Washington, USA, 2021. AAAI Press.

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

[20] N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software*, 29(4):353–372, 2003.

[21] S. Gratton, S. Jerad, and Ph. L. Toint. First-order objective-function-free optimization algorithms and their complexity. arXiv:2203.01757, 2022.

[22] S. Gratton, S. Jerad, and Ph. L. Toint. Convergence properties of an objective-function-free optimization regularization algorithm, including an $\mathcal{O}(\epsilon^{-3/2})$ complexity bound. *SIAM Journal on Optimization*, 33(3):1621–1646, 2023.

[23] S. Gratton, S. Jerad, and Ph. L. Toint. Parametric complexity analysis for a class of first-order Adagrad-like algorithms. *Optimization Methods and Software*, (to appear), 2024.

[24] S. Gratton, M. Mouffe, and Ph. L. Toint. Stopping rules and backward error analysis for bound-constrained optimization. *Numerische Mathematik*, 119(1):163–187, 2011.

[25] S. Gratton and Ph. L. Toint. S2MPJ and cutest optimization problems for Matlab, Python and Julia. arXiv:2407.07812, 2024.

[26] P. Joulani, A. Raj, A György, and C. Szepesvári. A simpler approach to accelerated stochastic optimization: Iterative averaging meets optimism. In *International Conference on Machine Learning (ICML)*, 2020.

[27] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings in the International Conference on Learning Representations (ICLR)*, 2015.

[28] S. Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. arXiv:1607.00345, 2016.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[30] C. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, 9(4):1100–1127, 1999.

[31] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming, Series B*, 45(1):503–528, 1989.

[32] L. Miao, X. Luo, T. Chen, W. Chen, D. Liu, and Z . Wang. Learning pruning-friendly networks via Frank-Wolfe: One-shot, any-sparsity, and no retraining. Proceeding of the International Conference on Learning Representations, 2022.

[33] M. Porcelli and Ph. L. Toint. Exploiting problem structure in derivative-free optimization. *ACM Transactions on Mathematical Software*, 48(1), 2021.

[34] S. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. In *Proceedings in the International Conference on Learning Representations (ICLR)*, 2018.

[35] T. Tieleman and G. Hinton. Lecture 6.5-RMSPROP. COURSERA: Neural Networks for Machine Learning, 2012.

[36] S. Wang, X. Yu, and P. Perdikaris. When and why PINNs fail to train: a neural tangent kernel perspective, 2022.

[37] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscapes. In *Proceedings in the International Conference on Machine Learning (ICML2019)*, 2019.

[38] X. Wu, R. Ward, and L. Bottou. WNGRAD: Learn the learning rate in gradient descent. arXiv:1803.02865, 2018.

[39] D. Zhou, J. Chen, Y. Tang, Z. Yang, Y. Cao, and Q. Gu. On the convergence of adaptive gradient methods for nonconvex optimization. arXiv:2080.05671, 2020.

[40] M. Zimmer, Ch. Speigel, and S. Pokutta. Compression-aware training of neural networks using Frank-Wolfe. arXiv:2205.11921, 2022.

Proof of Theorem 3.1

The proof uses the following technical lemma, due to [37, 15].

Lemma A.1 Let $\{a_k\}_{k\geq 0}$ be a non-negative sequence, $\xi > 0$ and define, for each $k \geq 0$, $b_k = \sum_{j=0}^{k} a_j$. Then

$$\sum_{j=0}^{k} \frac{a_j}{(\xi + b_j)} \leq \log\left(\frac{\xi + b_k}{\xi}\right). \tag{A.1}$$

Proof. See [37], [38] or [23, Lemma 3.1]. □

Proof of Theorem 3.1 We see from (3.1) that $w_{i,k}$ verifies **AS.4** and we may thus use Lemma 2.2 and its consequence (3.4).

For each $i \in \{1, \ldots, n\}$, we apply Lemma A.1 with $a_k = \chi_{i,k}^2$ and $\xi = \varsigma$ and obtain that,

$$\sum_{i=1}^{n}\sum_{j=0}^{k} \frac{\chi_{i,k}^2}{w_{i,k}^2} \leq \frac{1}{\vartheta}\sum_{i=1}^{n} \log\left(\frac{1}{\varsigma}\left(\varsigma + \sum_{l=0}^{k} \chi_{i,l}^2\right)\right) \leq \frac{n}{\vartheta}\log\left(1 + \frac{1}{\varsigma}\sum_{l=0}^{k}\sum_{i=1}^{n} \chi_{i,l}^2\right).$$

and substituting this bound in (3.4) then gives that

$$\frac{\tau\vartheta\varsigma}{2\kappa_B}\sum_{j=0}^{k}\sum_{i=1}^{n} \frac{\chi_{i,j}^2}{w_{i,j}} \leq \Gamma_0 + \tfrac{1}{2}(\kappa_{\rm B} + L)\frac{n}{\vartheta}\log\left(1 + \frac{1}{\varsigma}\sum_{l=0}^{k}\sum_{i=1}^{n} \chi_{i,l}^2\right). \tag{A.2}$$

Suppose now that

$$\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2 \geq \max\left[\varsigma, \frac{1}{2}e^{\frac{2\vartheta\Gamma_0}{n(\kappa_B+L)}}\right], \tag{A.3}$$

implying that

$$1+\frac{1}{\varsigma}\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2 \leq \frac{2}{\varsigma}\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2 \;\text{ and }\; \Gamma_0 \leq \frac{n(\kappa_{\rm B}+L)}{2\vartheta}\log\left(\frac{2}{\varsigma}\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2\right).$$

Using (A.2), we obtain then that

$$\frac{\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}}}{2\sqrt{2}\,\kappa_{\rm B}\sqrt{\sum_{\ell=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2}}\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2 \leq \frac{n(\kappa_{\rm B}+L)}{\vartheta}\log\left(\frac{2}{\varsigma}\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2\right),$$

that is

$$\frac{\tau\sqrt{2\varsigma}\vartheta^{\frac{5}{2}}}{4\kappa_{\rm B}}\sqrt{\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2} \leq 2n(\kappa_{\rm B}+L)\log\left(\sqrt{\frac{2}{\varsigma}\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2}\right). \tag{A.4}$$

Now define

$$\gamma_1 \stackrel{\rm def}{=} \frac{\tau\varsigma\vartheta^{\frac{5}{2}}}{4\kappa_{\rm B}}, \quad \gamma_2 \stackrel{\rm def}{=} 2n(\kappa_{\rm B}+L) \quad\text{and}\quad u \stackrel{\rm def}{=} \sqrt{\frac{2}{\varsigma}\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2} \tag{A.5}$$

and observe that that $\gamma_2 > 3\gamma_1$ because $\tau\sqrt{\varsigma}\vartheta^{\frac{3}{2}} \leq 1$ and $\kappa_{\rm B} \geq 1$. The inequality (A.4) can then be rewritten as

$$\gamma_1 u \leq \gamma_2 \log(u). \tag{A.6}$$

Let us denote by $\psi(u) \stackrel{\rm def}{=} \gamma_1 u - \gamma_2\log(u)$. Since $\gamma_2 > 3\gamma_1$, the equation $\psi(u)=0$ admits two roots $u_1 \leq u_2$ and (A.6) holds for $u \in [u_1, u_2]$. The definition of u_2 then gives that

$$\log(u_2) - \frac{\gamma_1}{\gamma_2}u_2 = 0$$

which is

$$u_2 e^{-\frac{\gamma_1}{\gamma_2}u_2} = 1.$$

Setting $z = -\frac{\gamma_1}{\gamma_2}u_2$, we obtain that

$$ze^z = -\frac{\gamma_1}{\gamma_2}$$

Thus $z = W_{-1}(-\frac{\gamma_1}{\gamma_2}) < 0$, where W_{-1} is the second branch of the Lambert function defined over $[-\frac{1}{e}, 0)$. As $-\frac{\gamma_1}{\gamma_2} \geq -\frac{1}{3}$, z is well defined and thus

$$u_2 = -\frac{\gamma_2}{\gamma_1}\,z = -\frac{\gamma_2}{\gamma_1}\,W_{-1}\left(-\frac{\gamma_1}{\gamma_2}\right) > 0.$$

As a consequence, we deduce from (A.6) and (A.5) that

$$\sum_{j=0}^{k}\sum_{i=1}^{n}\chi_{i,j}^2 = \frac{\varsigma}{2}\,u_2^2 = \frac{1}{2\varsigma}\,\left(\frac{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}{\tau\vartheta^{\frac{5}{2}}}\right)^2\left|W_{-1}\left(-\frac{\tau\varsigma\vartheta^{\frac{5}{2}}}{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}\right)\right|^2.$$

and

$$\underset{j\in\{0,\ldots,k\}}{\text{average}} \sum_{i=1}^{n} \chi_{i,j}^2 \leq \frac{1}{2\varsigma} \left(\frac{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}{\tau\vartheta^{\frac{5}{2}}}\right)^2 \left|W_{-1}\left(-\frac{\tau\varsigma\vartheta^{\frac{5}{2}}}{8n\kappa_{\rm B}(\kappa_{\rm B}+L)}\right)\right|^2 \cdot \frac{1}{k+1}. \tag{A.7}$$

If (A.3) does not hold, we have that

$$\underset{j\in\{0,\ldots,k\}}{\text{average}} \sum_{i=1}^{n} \chi_{i,j}^2 < \max\left\{\varsigma, \frac{1}{2}\, e^{\frac{2\Gamma_0\vartheta}{n(\kappa_B+L)}}\right\} \cdot \frac{1}{k+1}. \tag{A.8}$$

Combining (A.7) and (A.8) gives (3.5). □

Proof of Theorem 4.1

From (3.4) and AS.3, using $w_{\min,j} \stackrel{\rm def}{=} \min_{i\in\{1,\ldots,n\}} w_{i,k}$ ensures that

$$\Gamma_0 \geq f(x_0) - f(x_{k+1}) \geq \sum_{j=0}^{k}\sum_{i=1}^{n} \frac{\chi_{i,j}^2}{2\kappa_{\rm B} w_{i,j}} \left[\tau\vartheta\varsigma_{\min} - \frac{\kappa_{\rm B}(\kappa_{\rm B}+L)}{w_{\min,j}}\right]. \tag{A.9}$$

Consider now an arbitrary $\eta \in (0, \tau\vartheta\varsigma_{\min})$ and suppose first that, for some j,

$$\left[\tau\vartheta\varsigma_{\min} - \frac{\kappa_B+L}{w_{\min,j}}\right] \leq \eta, \tag{A.10}$$

i.e., using (4.1),

$$\varsigma_{\min}\, j^{\nu} \leq w_{\min,j} \leq \frac{\kappa_{\rm B}(\kappa_{\rm B}+L)}{\tau\vartheta\varsigma_{\min}-\eta}.$$

But this is impossible for $j > j_\eta$ for j_η given by (4.4), and hence (A.10) fails for all $j > j_\eta$. As a consequence, we have that, for $k > j_\eta$,

$$\begin{aligned}
f(x_{j_\eta+1}) - f(x_k) &\geq \eta \sum_{j=j_\eta+1}^{k}\sum_{i=1}^{n} \frac{\chi_{i,j}^2}{2\kappa_{\rm B} w_{i,j}} \\
&\geq \frac{\eta}{2\kappa_{\rm B}} \sum_{j=j_\eta+1}^{k}\sum_{i=1}^{n} \frac{\chi_{i,j}^2}{\max[\varsigma_i, v_{i,j}]\,\theta\,(j+1)^{\mu}} \\
&\geq \frac{\eta}{2\kappa_{\rm B}(k+1)^{\mu}} \sum_{j=j_\eta+1}^{k}\sum_{i=1}^{n} \min\left[\frac{\chi_{i,j}^2}{\varsigma_i}, \frac{\chi_{i,j}^2}{v_{i,j}}\right] \\
&\geq \frac{\eta(k-j_\eta)}{2\kappa_{\rm B}(k+1)^{\mu}} \min_{j\in\{j_\eta+1,\ldots,k\}} \left(\sum_{i=1}^{n} \min\left[\frac{\chi_{i,j}^2}{\varsigma_i}, \frac{\chi_{i,j}^2}{v_{i,j}}\right]\right)
\end{aligned} \tag{A.11}$$

But we also know from (2.28), (4.1) and (4.3) that

$$\begin{aligned} f(x_0) - f(x_{j_\eta+1}) &\geq \sum_{j=0}^{j_\eta}\sum_{i=1}^{n} \frac{\tau\vartheta\varsigma_{\min}\chi_{i,j}^2}{2\kappa_{\rm B} w_{i,j}} - \tfrac{1}{2}\,(\kappa_{\rm B}+L)\sum_{j=0}^{j_\eta}\sum_{i=1}^{n}\frac{\chi_{i,j}^2}{w_{i,j}^2} \\ &\geq -\tfrac{1}{2}\kappa_{\rm B}(\kappa_{\rm B}+L)\sum_{j=0}^{j_\eta}\sum_{i=1}^{n}\frac{\chi_{i,j}^2}{w_{i,j}^2} \\ &\geq -\frac{n\kappa_{\rm B}(\kappa_{\rm B}+L)}{2\vartheta}\,j_\eta \end{aligned} \tag{A.12}$$

Combining (A.11) and (A.12), we obtain that

$$\Gamma_0 \geq f(x_0) - f(x_{k+1}) \geq -\frac{n\kappa_{\rm B}(\kappa_{\rm B}+L)}{2\vartheta}\,j_\eta + \frac{\eta(k-j_\eta)}{2\kappa_{\rm B}(k+1)^\mu} \min_{j\in\{j_\eta+1,\ldots,k\}} \left(\sum_{i=1}^{n} \min\left[\frac{\chi_{i,j}^2}{\varsigma_i}, \frac{\chi_{i,j}^2}{v_{i,j}}\right]\right)$$

and thus that

$$\min_{j\in\{j_\eta+1,\ldots,k\}} \left(\sum_{i=1}^{n} \min\left[\frac{\chi_{i,j}^2}{\varsigma_i}, \frac{\chi_{i,j}^2}{v_{i,j}}\right]\right) \leq \frac{2\kappa_{\rm B}(k+1)^\mu}{\eta(k-j_\eta)}\left[\Gamma_0 + \frac{n\kappa_{\rm B}(\kappa_{\rm B}+L)}{2\vartheta}\,j_\eta\right]$$

and we deduce that there must exist a subsequence $\{k_\ell\} \subseteq \{k\}_{j_\eta+1}^{\infty}$ such that, for each ℓ,

$$\sum_{i=1}^{n} \min\left[\frac{\chi_{i,k_\ell}^2}{\varsigma_i}, \frac{\chi_{i,jk_\ell}^2}{v_{i,k_\ell}}\right] \leq \frac{2\kappa_{\rm B}(k_\ell+1)^\mu}{\eta(k_\ell-j_\eta)}\left[\Gamma_0 + \frac{n\kappa_{\rm B}(\kappa_{\rm B}+L)}{2\vartheta}\,j_\eta\right]. \tag{A.13}$$

But

$$\frac{(k_\ell+1)^\mu}{k_\ell-j_\eta} < \frac{2^\mu k_\ell^\mu}{k_\ell-j_\eta} < \frac{2k_\ell^\mu}{k_\ell-j_\eta} = \frac{2k_\ell^\mu k_\ell}{(k_\ell-j_\eta)k_\ell} = \frac{k_\ell}{k_\ell-j_\eta}\cdot\frac{2}{k_\ell^{1-\mu}} \leq \frac{2(j_\eta+1)}{k_\ell^{1-\mu}}, \tag{A.14}$$

where we used the facts that $\mu < 1$ and that $\frac{k_\ell}{k_\ell-j_\theta}$ is a decreasing function for $k_\ell \geq j_\theta + 1$. Using this inequality, we thus obtain from (A.13) that, for each ℓ,

$$\sum_{i=1}^{n} \min\left[\frac{\chi_{i,k_\ell}^2}{\varsigma_i}, \frac{\chi_{i,k_\ell}^2}{v_{i,k_\ell}}\right] \leq \frac{4\kappa_{\rm B}(j_\eta+1)}{\eta\, k_\ell^{1-\mu}}\left[\Gamma_0\theta + n\frac{\kappa_{\rm B}(\kappa_{\rm B}+L)}{2\vartheta}\,j_\eta\right].$$

As a consequence,

$$k_\varsigma \stackrel{\rm def}{=} \left(\frac{4\kappa_{\rm B}(j_\eta+1)\left[\Gamma_0 + \frac{n\kappa_{\rm B}(\kappa_{\rm B}+L)}{2\vartheta}\,j_\eta\right]}{\eta\varsigma_{\min}}\right)^{\frac{1}{1-\mu}}$$

is such that, for all $k_\ell \geq k_\varsigma$,

$$\min\left[\frac{\chi_{i,k_\ell}^2}{\varsigma_i,}, \frac{\chi_{i,k_\ell}^2}{v_{i,k\ell}}\right] \leq \varsigma_{\min}. \tag{A.15}$$

But (4.3) ensures that

$$\min\left[\frac{\chi_{i,k_\ell}^2}{\varsigma_i}, \frac{\chi_{i,k_\ell}}{\sqrt{\vartheta}}\right] \leq \min\left[\frac{\chi_{i,k_\ell}^2}{\varsigma_i,}, \frac{\chi_{i,k_\ell}^2}{v_{i,k\ell}}\right] \leq \varsigma_{\min}.$$

Now this inequality and the bounds $\varsigma_i \in (0,1)$, $\vartheta \in (0,1)$ and $\varsigma_{\min} \in (0,1)$ together imply that $\chi_{i,k_\ell} \in (0,1)$ and hence that $\chi^2_{i,k_\ell} < \chi_{i,k_\ell}$. We thus obtain from (A.15) that, for all $k_\ell \geq k_\varsigma$,

$$\sum_{i=1}^{n} \frac{\chi^2_{i,k_\ell}}{\max[\varsigma_i, \sqrt{\vartheta}]} \leq \frac{2\kappa_{\rm B}(k_\ell+1)^\mu}{\eta(k_\ell - j_\eta)} \left[\Gamma_0 + \frac{n\kappa_{\rm B}(\kappa_{\rm B}+L)}{2\vartheta}\, j_\eta\right]$$

which, because $\max[\varsigma_i, \sqrt{\vartheta}] \leq 1$, gives that, for all $k_\ell \geq k_\varsigma$,

$$\sum_{i=1}^{n} \chi^2_{i,k_\ell} \leq \frac{(k_\ell+1)^\mu}{k_\ell - j_\eta} \left(\frac{4\kappa_{\rm B}}{\eta}\right) \left[\Gamma_0 + n\frac{\kappa_{\rm B}(\kappa_{\rm B}+L)}{\vartheta}\, j_\eta\right], \tag{A.16}$$

finally implying, because of (A.14), that (4.5) holds with

$$\kappa_\diamond = 2(j_\eta+1)\left(\frac{4\kappa_{\rm B}}{\eta}\right)\left[\Gamma_0 + n\frac{\kappa_{\rm B}(\kappa_{\rm B}+L)}{\vartheta} j_\eta\right].$$

□

Details of the experimental setup

The test problem used and their dimension are given in Table A.2.

BQPGABIM	50	HADAMALS	400	NCVXBQP1	500	OBSTCLBL	625
BQPGASIM	50	JNLBRNG1	625	NCVXBQP2	500	OBSTCLBM	625
EXPLIN	600	JNLBRNG2	625	NCVXBQP3	500	OBSTCLBU	625
EXPLIN2	600	JNLBRNGA	625	NOBNDTOR	1024	QINGB	500
EXPQUAD	120	JNLBRNGB	625	OBSTCLAE	625		
GENROSEB	500	LINVERSE	999	OBSTCLAL	625		

Table A.2: The CUTEst/S2MPJ problems used for illustration

Our test were performed in Matlab R2023b on a Dell Precision with 64GB of memory and running Ubuntu 20.04. All algorithms were stopped after a maximum number of 100000 iterations (remember we are mostly using first-order methods). All variants of the ASTR1B algorithms use the constants given by

$$\tau = 1, \quad \vartheta = 1 \quad \text{and} \quad \varsigma = 0.01.$$

The trust-region algorithm was used with an acceptance threshold $\eta_1 = 10^{-4}$, a trust-region expansion threshold $\eta_2 = 0.95$, a radius contraction factor of $\frac{1}{2}$ and a radius expansion factor of 2. In all cases where a nonzero B_k was used, the approximate quadratic model was minimized using a truncated Lanczos iteration with a maximum number of inner iterations given by $3n$ and a relative gradient accuracy request of 10^{-4}. Whenever random noise is present in the evaluations, the gradient's and (when relevant) the objective-function's values are perturbed according to

$$f(x) = f(x)\big(1+\tau\mathcal{N}(0,1)\big), \quad [g(x)]_i = [g(x)]_i\big(1+\tau\mathcal{N}(0,1)\big), \quad (i \in \{1,\ldots,n\}),$$

where $\tau \in [0,1]$ is the noise level and $\mathcal{N}(0,1)$ is the standard normal distribution. In these cases, the reported results are based on the rounded averages of ten independent runs.