A multilevel stochastic regularized first-order method with application to finite sum minimization

Filippo Marini* Margherita Porcelli†,‡ Elisa Riccietti§

November 27, 2025

Abstract

In this paper, we propose a multilevel stochastic framework for the solution of nonconvex unconstrained optimization problems. The proposed approach uses random regularized first-order models that exploit an available hierarchical description of the problem, being either in the classical variable space or in the function space, meaning that different levels of accuracy for the objective function are available. We propose a convergence analysis showing an almost sure global convergence of the method to a first order stationary point. The numerical behavior is tested on the solution of finite sum minimization problems. Differently from classical deterministic multilevel schemes, our stochastic method does not require the finest approximation to coincide with the original objective function along all the optimization process. This allows for significantly decreasing their cost, for instance in data-fitting problems, where considering all the data at each iteration can be avoided.

1 Introduction

Many modern applications require the solution of large scale stochastic optimization problems, i.e., problems of the form:

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where f is a function that is assumed to be smooth and bounded from below, whose value can only be computed with some noise [2]. When considering this problem, it is usually assumed that random realizations of f are computable, with variable accuracy [8, 20]. While this framework is borrowed from derivative free optimization, it applies to derivative-based optimization as well, cf. [20]. Expected risk minimization and problems in which the objective function is the outcome of a stochastic simulation are examples that fit this framework [12, 20].

Methods to address (1) usually rely on the assumption that the accuracy of the function approximations grows asymptotically, which leads to iterations that are more and more expensive [8, 20]. Moreover, such methods control the accuracy of the function estimates but not the size of the variables. An important challenge in this context is thus to develop scalable stochastic methods, able to handle the increasing costs of large scale stochastic optimization problems.

*Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale Morgagni 40/44, 50134 Firenze, Italia. Email: filippo.marini@unifi.it

†Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale Morgagni 40/44, 50134 Firenze, Italia. Member of the INdAM Research Group GNCS. Email: margherita.porcelli@unifi.it

‡ISTI–CNR, Via Moruzzi 1, Pisa, Italia

§ENS de Lyon, Univ Lyon, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France. Email: elisa.riccietti@ens-lyon.fr

In classical scientific computing, *multilevel methods* represent powerful techniques that have been developed to cope with structured large scale optimization problems where the limiting factor is the number of variables. When the structure of the problem at hand allows for a hierarchical description of the problem itself, these methods reduce the cost of the problem's solution by computing cheap steps by exploiting function approximations defined on subspaces of progressively smaller dimension. Thanks to this, they achieve not only considerable speed-ups but also an improved quality solution in various applications, spanning from the solution of partial differential equations to image reconstruction [28, 33, 34, 37].

Existing multilevel methods are limited to a *deterministic* context and thus are unsuitable to address stochastic optimization problems. Moreover, they have always been used to exploit hierarchies in the variables space, such as discretizations of infinite dimensional problems on selected grids. However, in many modern applications the limiting factor can be the accuracy of the function estimates rather than the size of the model.

In this work we propose an extension of multilevel methods to a stochastic setting. More precisely, we assume to have access to a hierarchy of randomly chosen computable representations of f, built either by reducing the dimension in the variables space or by reducing the noise of the function approximation, or both. Our multilevel adaptation decreases the cost of the iterative solution by coupling standard "fine" stochastic steps with "coarse" steps, which are less expensive and keep a low cost for all the iterative process. Such steps can be built either:

- By reducing the dimension of the problem in the variables space, as in classical multilevel schemes, but possibly in a stochastic way.
- Using low accuracy approximations of the objective function even in later steps of the optimization process.

A level ℓ will correspond to a subset of variables and to a noise level in the function approximation. The fine steps will still be computed considering all the variables and increasingly accurate function approximations, while the coarse steps will be computed by taking into account just small subsets of variables and/or inaccurate function approximations. While the fine iterations will still become more and more expensive, as in standard stochastic methods, their number will be reduced, and progress and speed will be ensured by the coarse iterations.

Our multilevel framework thus extends classical multilevel schemes by allowing for the solution of stochastic problems such as (1), with hierarchies built also in the function space. Differently from the classical setting, the steps are stochastic, as well as the generated sequence of iterations. The stochasticity of the framework allows one to mitigate one of the main limitations of deterministic multilevel methods: the need of periodically handling the full original problem at fine scale, which limits the size of affordable problems. In this context indeed, as it is the case for classical stochastic methods [8, 10, 15, 20], this will be necessary only towards the end of the optimization procedure.

Inspired by the stochastic trust-region framework STORM in [20], we propose a stochastic multilevel Adaptive Regularization (AR) technique[1] named MU$^\ell$STREG for MUltilevel STochastic REegularized Gradient method, where ℓ refers to the number of levels in the hierarchical description of the problem. As in standard AR techniques, the automatic step selection choice is made possible at each level by the use of models regularized by an adaptive regularization parameter, but the fine level models are stochastic. As for STORM, we prove that our method converges to a first order stationary point as long as the local models of the objective function at the finest level are fully linear and the function estimates are sufficiently accurate, both with

[1]We choose to employ AR techniques rather than trust-region techniques as they are easier to adapt to a multilevel context, cf. [16].

sufficiently high probability. The proposed analysis extends that proposed in [20] to adaptive regularization methods while including also the multilevel steps.

From a practical point of view, we test the multilevel paradigm on the solution of *finite sum minimization problems*. Such problems have their origin in large scale data analysis applications where models depending on a large number of parameters n are fitted to a large set of N samples, thus either n or N (or both) are really large, see e.g. [11, 12]. Several methods have been developed to cope with the large sizes of the datasets. In particular, optimization techniques based on subsampling techniques have been proposed, among them the numerous variations of classical stochastic gradient descent (SGD), see e.g. [6, 11, 12, 21, 31] and references therein.

When considering expected risk minimization problems, there is a natural way of building a hierarchy of computable function approximations, through the definition of nested subsample sets and of finite sum minimization problems. A multilevel method in this context alternates "fine steps", i.e., steps computed considering increasingly large subsets of data and "coarse steps" computed taking into account just small subsets of data (mini-batches). In this context they can be viewed as a way for accelerating adaptive sampling strategies. Moreover, the coarse steps are computed by minimizing a model that is built from the coarse level approximations (finite sum problems defined on mini-batches) by adding a correction term, usually known as "first order coherence" in the multilevel literature, which (in this context) accounts for the discrepancy between the fine gradient and the coarse one. When the full gradient is used at fine level, this is reminiscent of the term added in the reduced variance gradient estimate of the mini-batch version of SVRG [31]. Multilevel methods in this case can thus also be interpreted as variance reduction methods, cf. [13], with the advantage of allowing for an automatic choice of the step size.

Contributions

- This is the first stochastic framework for multilevel methods, that are currently limited to the deterministic case.
- The multilevel framework allows for hierarchies in the function space, i.e., built by considering function approximations with variable accuracy.
- The stochastic multilevel framework mitigates the limiting factor of classical deterministic multilevel methods, whose convergence theory requires the fine level function to coincide with the original target function at every iteration.
- Our multilevel framework, and thus the stochastic analysis, also covers the classical one-level case.
- The multilevel method offers a strategy to accelerate methods for stochastic optimization such as STORM.

Related work *Multilevel methods.* As a natural extension of multigrid methods [14] to a non-linear context, multilevel methods were first proposed by Nash through the MG/OPT framework [37] and later extended to trust region schemes [28]. Recently these methods have been extended to other contexts: high-order models [16], non-smooth optimization [34, 38], machine learning [26, 27, 32]. A multilevel method that exploits hierarchies in the function space has been explored in [13], for deterministic finite sum convex problems leveraging the multilevel scheme of MG/OPT developed in [37]. Recent research [26] proposes a (deterministic) multilevel version of the OFFO method that does not require function evaluations and that is based on the classical multilevel scheme constructed on the variable space.

Derivative free optimization An idea close to that of multilevel methods to alternate between accurate steps and cheap steps using more or less information can be found in full-low evaluation derivative free optimization for direct search methods [7, 41]. Another technique that has been considered to reduce the cost of DFO problems is random subset selection [18].

Stochastic regularization methods In [8] the authors propose a Levenberg-Marquardt adaptation of the STORM framework for noisy least squares problems. As in our work, the step size in this context is updated through a regularization parameter. We inherited from this work the dependence of the regularization parameter from the norm of the gradient (cf. (3) below) and the definition of accurate models (cf. Definition 1). The recent literature on variants of the standard trust-region method based on the use of random models is very extensive, we refer to [3, 4, 6, 29, 40] to name a few and references therein. In addition, stochastic adaptive second order regularization methods are considered in [30, 42], while higher order models are considered in [5].

Organization of the paper In section 2 we introduce our MU$^\ell$STREG method and we propose its convergence analysis in section 3. In section 4 we specialize the MU$^\ell$STREG framework to finite sum minimization and we analyze its numerical performance in section 5. We draw some conclusions and present some perspectives in section 6.

2 The multilevel stochastic regularized gradient method

In this section we describe our new MUltilevel STochastic REegularized Gradient method (MU$^\ell$-STREG)[2] for the solution of problem (1).

Hierarchical representation of problem (1) We assume to have at disposal, at each iteration k, a fine level computable approximation f_k of f and a hierarchy of coarse computable approximations, $\{\phi_k^\ell\}_{\ell=1}^{\ell_{\max}-1}$, where 1 corresponds to the coarsest level. More precisely, we assume that, for each k, ϕ_k^ℓ is less costly to compute than $\phi_k^{\ell+1}$ and that $\phi_k^{\ell_{\max}-1}$ is less costly to compute than f_k, either because it is less accurate and/or because it is defined on a smaller dimensional space. We also assume that the functions ϕ_k^ℓ are randomly chosen at iteration k, but once they have been defined, they are deterministic functions. As in classical multilevel methods, we assume to have at disposal some transfer operators R_k^ℓ (restriction) and P_k^ℓ (prolongation) to transfer the information (variables and gradients) from level ℓ to level $\ell-1$ and vice-versa, such that $R_k^\ell = \nu_k (P_k^\ell)^T$ for some $\nu_k > 0$ [14]. Unlike the classical framework, such operators can be random and often vary from one iteration k to another. If the hierarchy is built just in the functions space all the variables will have the same dimension and the transfer operators will thus just be the identity. We present here some problems that fit this framework. In the following we will refer to the fine level with the index $\ell_{\max}$.

Example 1. Expected risk minimization problems *[12] Assume to have two sets $\mathcal{Z}, \mathcal{Y}$, a loss function $\mathcal{L} : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$, and a probability distribution P on $\mathcal{Z} \times \mathcal{Y}$. The expected risk minimization problem is*

$$\min_{x \in \mathbb{R}^n} \mathbb{E}_P[\mathcal{L}(y, m_x(z))] \tag{2}$$

for a given parametric model $m_x : \mathcal{Z} \to \mathcal{Y}$. Given a data set $\{(z_i, y_i)\}$, $i = 1, \ldots, N$, drawn from the distribution $P(z, y)$, the fine approximation at iteration k can be defined as the averaged sum

[2] The ℓ denotes the number of levels in the hierarchical problem description.

of the the functions $f^{(i)}(x) := \mathcal{L}(y_i, m_x(z_i))$*, that is as*

$$\frac{1}{|\mathcal{S}_k^{\ell_{\max}}|} \sum_{i \in \mathcal{S}_k^{\ell_{\max}}} f^{(i)}(x) \quad x \in \mathbb{R}^n,$$

over increasingly larger subsets $\mathcal{S}_k^{\ell_{\max}} \subseteq \{1, \dots, N\}$*, such that* $\mathcal{S}_k^{\ell_{\max}} \subseteq \mathcal{S}_{k+1}^{\ell_{\max}}$ *(cf. section 4). The hierarchy of functions approximations at coarse levels can then be built in two different ways:*

1. Hierarchy in the samples space *The coarse approximations can be defined, at iteration* k*, as the averaged sum of the* $f^{(i)}$ *over nested subsets of* $\mathcal{S}_k^{\ell_{\max}}$*, that is* $\phi_k^\ell := f^{S_k^\ell}$ *where:*

$$f^{S_k^\ell}(x) = \frac{1}{|\mathcal{S}_k^\ell|} \sum_{i \in \mathcal{S}_k^\ell} f^{(i)}(x) \quad x \in \mathbb{R}^n,$$

   for $\mathcal{S}_k^\ell \subseteq \mathcal{S}_k^{\ell+1} \subseteq \mathcal{S}_k^{\ell_{\max}}$ *for all* ℓ *and for all* k*.*

2. Hierarchy on the variables *The coarse approximations can still be defined as the averaged sum of the* $f^{(i)}$ *over* $\mathcal{S}_k^{\ell_{\max}}$*, but considering a (possibly random) subset of the variables:*

$$\phi_k^\ell(\bar{x}) = \frac{1}{|\mathcal{S}_k^{\ell_{\max}}|} \sum_{i \in \mathcal{S}_k^{\ell_{\max}}} f^{(i)}(\bar{x}) \quad \bar{x} \in \mathbb{R}^{n_k^\ell}$$

   with $n_k^\ell \leq n_k^{\ell+1} \leq n$ *for all* ℓ *and for all* k*.*

Notice that the sets $\mathcal{S}_k^\ell$ *are drawn randomly, but once they are drawn, the function approximations are deterministic.*

Example 2. Montecarlo simulations *[24] Let* ξ *be a random variable defined on the probability space* (Ω, P)*. Let* $f(x) = \mathbb{E}(F(x, \xi))$*, with* $F : \mathbb{R}^n \times \Omega \to \mathbb{R}$*.* $F(x, \xi)$ *can represent for instance the fit of the solution of a stochastic PDE to some data.* F *can be approximated at fine level by the stochastic approximations* $F_{h_k}(x, \xi)$*, the output of a PDE solver with mesh size* $h_k > 0$ *and* f *can then be approximated by the Montecarlo estimator*

$$\hat{f}_{N_k, h_k}(x) = \frac{1}{N_k} \sum_{i=1}^{N_k} F_{h_k}(x, \xi_i)$$

where $\xi_1, \dots, \xi_{N_k}$ *are iid samples drawn from* P*. The coarse approximations can be built either:*

1. In the function space, *by defining*

$$\phi_k^\ell(x) = \frac{1}{N_k^\ell} \sum_{i=1}^{N_k^\ell} F_{h_k^\ell}(x, \xi_i)$$

   with $h_k^\ell \geq h_k^{\ell+1} \geq h_k$ *and/or* $N_k^\ell \leq N_k^{\ell+1} \leq N_k$ *for al* ℓ *and for all* k*.*

2. In the variables space *by defining*

$$\phi_k^\ell(\bar{x}) = \frac{1}{N_k} \sum_{i=1}^{N_k} F_{h_k}(\bar{x}, \xi_i), \quad \bar{x} \in \mathbb{R}^{n_k^\ell}$$

   with $n_k^\ell \leq n_k^{\ell+1} \leq n$ *for all* ℓ *and for all* k*.*

The step computation For any level ℓ and at each iteration k, our multilevel gradient method can choose between two different types of stochastic steps: a gradient step, which is known as the *fine step*, or a *coarse step* computed by exploiting the approximations of f. Notice that, differently from classical deterministic multilevel schemes, the steps are all stochastic.

At the finest level, if the fine step is chosen, a model $m_k(x_k+s) := f_k + g_k^T s$ approximating f around the current iterate x_k is built, where f_k and g_k are approximations of $f(x_k)$ and $\nabla_x f(x_k)$, respectively, and a classical stochastic gradient step $s_k = -\frac{g_k}{\lambda_k \|g_k\|}$ is taken[3] for some $\lambda_k > 0$, with g_k a realization of $\nabla_x f(x_k)$.

If the coarse step is chosen at iteration k, the step is found by recursively minimizing a sequence of regularized models $m_k^{R,\ell}$, for $\ell = \ell_{\max}-1, \dots, 1$, built exploiting the random approximations $\{\phi_k^\ell\}_{\ell=1}^{\ell_{\max}-1}$ of f and are thus either defined in a lower dimensional space, or employs inaccurate function approximations, or both. More precisely, at each level $\ell > 1$ (including the finest level $\ell_{\max}$) a coarse model is defined for the immediately coarser level, which will serve as an objective function for level $\ell-1$ when the algorithm is called recursively. Starting at the fine level and considering the highest coarse approximation in the hierarchy $\phi_k^{\ell_{\max}-1}$, at iteration k we define

$$\begin{aligned}\varphi_k^{\ell_{\max}-1}(s) = \phi_k^{\ell_{\max}-1}(R_k^{\ell_{\max}-1}x_k + s) \\ + (R_k^{\ell_{\max}-1}g_k - \nabla_s \phi_k^{\ell_{\max}-1}(R_k^{\ell_{\max}-1}x_k))^T s,\end{aligned}$$

i.e., $\varphi_k^{\ell_{\max}-1}$ is a modification of the coarse function $\phi_k^{\ell_{\max}-1}$ through the addition of a correction term. This correction aims to enforce the following relation:

$$\nabla_s \varphi_k^{\ell_{\max}-1}(0) = R_k^{\ell_{\max}-1} g_k,$$

which ensures that the behaviour of the coarse model is coherent with the fine objective function approximation, up to order one. The step is defined as an approximate minimizer of the regularized model

$$m_k^{R,\ell_{\max}-1}(s) = \varphi_k^{\ell_{\max}-1}(s) + \frac{\lambda_k^{\ell_{\max}-1}\|g_k\|}{2}\|s\|^2, \tag{3}$$

with $\lambda_k^{\ell_{\max}-1} > 0$, by calling the multilevel procedure in a recursive way. At the first level of the recursive call, the algorithm will take as an objective function $m_k^{R,\ell_{\max}-1}$, and take either a gradient step for $m_k^{R,\ell_{\max}-1}$ or a coarse step. To define the coarse step, a coarse model for $m_k^{R,\ell_{\max}-1}(s)$ is built, which involves the approximation $\phi_k^{\ell_{\max}-2}$, the restriction of the correction vector and the regularization term from level $\ell_{\max}-1$ and a new correction term and a new regularization for level $\ell_{\max}-2$. The procedure is repeated until the coarsest level is reached, which is minimized directly without further recursion. Once a coarse step s^* is found at the end of the recursive procedure, we set $s_k = P_k^{\ell_{\max}} s^*$.

At a generic level ℓ, the recursive call is stopped as soon as a step $s_k^{\ell-1}$ is found that satisfies the following conditions:

$$m_k^{R,\ell-1}(s_k^{\ell-1}) < m_k^{R,\ell-1}(0), \quad \left\|\nabla_s m_k^{R,\ell-1}(s_k^{\ell-1})\right\| \le \epsilon^{\ell-1}\|s_k^{\ell-1}\|, \tag{4}$$

for some $\epsilon^{\ell-1} > 0$, and we set $s_k^\ell := P_k^\ell s_k^{\ell-1}$. As we will see, these conditions will ensure the convergence of the multilevel method in the spirit of the Adaptive-Regularization algorithm with

[3]This amounts to minimize $m_k(x_k+s) + \frac{\lambda_k\|g_k\|}{2}\|s\|^2$ wrt s. Notice that the stepsize depends on the norm of the gradient as in [8], cf. discussion in [8, section 3.1].

a first-order model described e.g., in [17, Sec. 2.4.1]. Note that even if we use a first order model at fine level, we could use a higher order method to minimize the lower level models.

In order to be meaningful, the coarse steps are restricted to iterations such that

$$\|R_k^\ell g_k^\ell\| \geq \kappa^\ell \|g_k^\ell\|$$

with g_k^ℓ a realization of the gradient of the objective function at level ℓ, and for $\kappa^\ell \in (0, \min_k \min\{1, \|R_k^\ell\|\})$ [28].

This framework is flexible and encompasses several actual implementations: at each iteration k one needs to choose whether to employ the fine or the coarse step. A sketch of a possible MU$^\ell$STREG cycle of iterations is depicted in Figure 1.

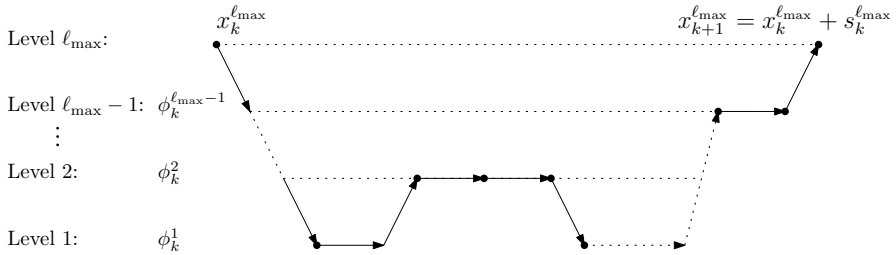

Figure 1: Sketch of a possible iteration scheme for MU$^\ell$STREG. Horizontal arrows represent fine steps.

The step acceptance The step s_k^ℓ obtained at each level is used to define a trial point $x_k^\ell + s_k^\ell$ and an estimate $f_k^{\ell,s}$ of $m_k^{R,\ell}(x_k^\ell + s_k^\ell)$. The step acceptance is based on the ratio of the achieved reduction over the predicted reduction:

$$\rho_k = \frac{f_k^\ell - f_k^{\ell,s}}{m_k^\ell(0) - m_k^\ell(s_k^\ell)}, \tag{5}$$

where f_k^ℓ is an estimate of $m_k^{R,\ell}(x_k^\ell)$, and for the fine step $m_k^\ell - m_k^{\ell,s} = -(g_k^\ell)^T s_k^\ell$ with g_k^ℓ an estimate of $\nabla_s m_k^{R,\ell}(x_k^\ell)$ and for the coarse step $m_k^\ell - m_k^{\ell,s} = \varphi_k^{\ell-1}(0) - \varphi_k^{\ell-1}(s_k^{\ell-1})$. A *successful* iteration is declared if the model is accurate, i.e., ρ_k is larger than or equal to a chosen threshold $\eta_1 \in (0,1)$ and $\|g_k^\ell\| \geq \frac{\eta_2}{\lambda_k^\ell}$ for some $\eta_2 > 0$; otherwise the iteration is declared *unsuccessful* and the step is rejected. The test for the step acceptance is combined with the update of the regularization parameter λ_k^ℓ for the next iteration. The update is still based on the ratio (5). If the step is successful, the regularization parameter is decreased, otherwise it is increased.

The full multilevel procedure with ℓ levels, specialized for finite sum minimization problems, is described in Algorithm 2 and will be introduced in section 4. In the following section, for sake of simplicity, we detail the procedure in the two-level case.

2.1 MU^{2}STREG: the two-level case

We assume here that we have just two levels, i.e., just a coarse approximation to f. We therefore omit the superscript ℓ and we denote by ϕ_k the coarse approximation at iteration k. Moreover, let $n_c \leq n$ be the dimension of the coarse space, and let R_k and P_k be the grid operators. We sketch the MU$^\ell$STREG procedure for $\ell = 2$ in Algorithm 1 where we rename it as MU^{2}STREG. In the following, we collect the main assumptions on the algorithmic steps that will be used in the convergence analysis in the next section.

Assumption 1. *At each iteration k of Algorithm 1, given $f_k \in \mathbb{R}$ and $g_k \in \mathbb{R}^n$, approximations to $f(x_k)$ and $\nabla_x f(x_k)$ respectively, let the step $s_k \in \mathbb{R}^n$ be computed so that either:*

$$s_k = -\frac{g_k}{\lambda_k \|g_k\|}, \qquad \textit{(fine step) or} \tag{6}$$

$$s_k = P_k s^*,\ s^* \in \mathbb{R}^{n_c}, \qquad \textit{(coarse step)} \tag{7}$$

with s^* satisfying

$$m_k^R(s^*) \leq m_k^R(0) \quad \textit{and} \quad \|\nabla_s m_k^R(s^*)\| \leq \theta \|s^*\|,\ \theta > 0 \tag{8}$$

where, for $s \in \mathbb{R}^{n_c}$

$$\varphi_k(s) := \phi_k(R_k x_k + s) + (R_k g_k - \nabla_s \phi_k(R_k x_k))^T s, \tag{9a}$$

$$m_k^R(s) := \varphi_k(s) + \frac{\lambda_k \|g_k\|}{2}\|s\|^2. \tag{9b}$$

The definition of the coarse model ensures that

$$\nabla_s \varphi_k(0) = R_k g_k \tag{10}$$

and that, when the coarse model is used, it holds:

$$\varphi_k(s^*) - \varphi_k(0) \leq -\frac{1}{2}\lambda_k \|g_k\| \|s^*\|^2. \tag{11}$$

The use of the coarse step is restricted to iterations k such that

$$\|R_k g_k\| \geq \kappa_H \|g_k\| \tag{12}$$

for $\kappa_H \in (0, \min_k \min\{1, \|R_k\|\})$. We assume that $R_k = \nu_k P_k^T$ with $\nu_k = 1$ for all k, without loss of generality, and that, for all k, $\|R_k\| \leq \kappa_R$ with $\kappa_R > 0$. This in particular, from (10), ensures that, if $s_k = P_k s^*$ then

$$\nabla_s \varphi_k(0)^T s^* = g_k^T s_k. \tag{13}$$

3 Convergence theory

In this section, we provide a theoretical analysis of the proposed multilevel method proving the almost sure global convergence to first order critical points. Note that, as the method is recursive, we can restrict the analysis to the two-levels case. We thus focus on MU^{2}STREG as described in section 2.1. The analysis follows the scheme proposed in [20] and is extended to adaptive regularization methods while including the multilevel steps. Let us now first state some regularity assumptions as in [9].

Assumption 2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $\phi_k : \mathbb{R}^{n_c} \to \mathbb{R}$ with $n \geq n_c$, be continuously differentiable and bounded from below functions, for all k. Let us assume that the gradients of f and of ϕ_k are Lipschitz continuous, i.e., that there exist constants L_f and L_{ϕ_k} such that*

$$\|\nabla_x f(x) - \nabla_x f(y)\| \leq L_f \|x - y\| \quad \textit{for all} \quad x, y \in \mathbb{R}^n,$$
$$\|\nabla_x \phi_k(x) - \nabla_x \phi_k(y)\| \leq L_{\phi_k} \|x - y\| \quad \textit{for all} \quad x, y \in \mathbb{R}^{n_c}.$$

Let $L_\phi := \max_k L_{\phi_k}$.

Algorithm 1 MU^{2}STREG(x_0, λ_0) two-levels stochastic regularized gradient method

1: • **Initialization:** Choose $x_0 \in \mathbb{R}^n$ and $\lambda_0 > 0$. Set the constants $\eta_1 \in (0,1)$, $\eta_2 > 0$ and $\gamma \in (0,1)$. Set $k = 0$.

2: • **Operators and functions computation:** Choose a (possibly random) restriction operator R_k and compute g_k, a fine approximation of $\nabla_x f(x_k)$.

3: • **Model choice:** If (12) holds, choose if to use the fine level model and go to Step 4, or the lower level model and go to Step 5. Otherwise go to Step 4.

4: • **Fine step computation:** Build a model $m_k(x_k + s) = f_k + g_k^T s$ that approximates $f(x)$ around x_k. Set $s_k = -\frac{g_k}{\lambda_k \|g_k\|}$, $m_k^0 = m_k(x_k)$ and $m_k^s = m_k(x_k + s_k)$. Go to Step 6.

5: • **Coarse step computation:** Randomly define a coarse approximation ϕ_k of f, and the lower level model and its regularized version as:

$$\varphi_k(s) = \phi_k(R_k\, x_k + s) + [R_k\, g_k - \nabla_x \phi_k(R_k\, x_k)]^T s,$$
$$m_k^R(s) = \varphi_k(s) + \frac{1}{2}\lambda_k \|g_k\| \|s\|^2.$$

Approximately minimize m_k^R, yielding an approximate solution s^* satisfying (8). Define $s_k = P_k s^*$ and set $m_k^0 = \varphi_k(0)$ and $m_k^s = \varphi_k(s^*)$.

6: • **Acceptance of the trial point and regularization parameter update:** Obtain an estimate f_k^s of $f(x_k + s_k)$ and compute $\rho_k = \dfrac{f_k - f_k^s}{m_k^0 - m_k^s}$.
If $\rho_k \geq \eta_1$ and $\|g_k\| \geq \eta_2/\lambda_k$ **then** set $x_{k+1} = x_k + s_k$ and $\lambda_{k+1} = \gamma \lambda_k$.
Else set $x_{k+1} = x_k$ and $\lambda_{k+1} = \gamma^{-1}\lambda_k$.

7: • **Check stopping criterion**. If satisfied stop, otherwise set $k = k + 1$ and go to Step 2.

Remark 1. *In the setting of finite sum minimization, cf. Example 1, the assumption is satisfied if all the $f^{(i)}$ are smooth, which is quite a common assumption in this context [23].*

Moreover, we assume that the models in this work are random functions and so is their behavior and influence on the iterations. Hence, M_k will denote a random model in the k-th iteration, while we will use the notation $m_k = M_k(\omega)$ for its realizations. As a consequence of using random models, the iterates X_k, the regularization parameter Λ_k and the steps S_k are also random quantities, and so $x_k = X_k(\omega)$, $\lambda_k = \Lambda_k(\omega)$, $s_k = S_k(\omega)$ will denote their respective realizations. Similarly, let random quantities F_k, F_k^s denote the estimates of $f(X_k)$ and $f(X_k + S_k)$, with their realizations denoted by $f_k = F_k(\omega)$ and $f_k^s = F_k^s(\omega)$. In other words, Algorithm 1 results in a stochastic process $\{M_k, X_k, S_k, \Lambda_k, F_k, F_k^s\}$. Our goal is to show that under certain conditions on the sequences $\{M_k\}$ and $\{F_k, F_k^s\}$ the resulting stochastic process has desirable convergence properties with probability one. In particular, we will assume that models M_k and estimates F_k, F_k^s are sufficiently accurate with sufficiently high probability, conditioned on the past. To formalize conditioning on the past, let $\mathcal{F}_{k-1}^{M\cdot F}$ denote the σ-algebra generated by $M_0, \ldots, M_{k-1}$ and $F_0, \ldots, F_{k-1}$ and let $\mathcal{F}_{k-1/2}^{M\cdot F}$ denote the σ-algebra generated by $M_0, \ldots, M_k$ and $F_0, \ldots, F_{k-1}$. To formalize sufficient accuracy we use the measure for the accuracy introduced in [8], which adapts to regularized models those originally proposed in [20].

Definition 1. *Suppose that ∇f is Lipschitz continuous. Given $\lambda_k > 0$, a function m is a κ-fully linear model of f around the iterate x_k provided, for $\kappa = (\kappa_f, \kappa_g)$, that for all y in a*

neighbourhood of x_k:

$$\|\nabla_x f(y) - \nabla_x m(y)\| \leq \frac{\kappa_g}{\lambda_k}, \tag{14}$$

$$|f(y) - m(y)| \leq \frac{\kappa_f}{\lambda_k^2}. \tag{15}$$

We will ask for this requirement on the fine level model $m_k(x_k + s) = f_k + g_k s^T$. Specifically, we will consider probabilistically fully-linear models, according to the following definition [20]:

Definition 2. *A sequence of random models $\{M_k\}$ is said to be α-probabilistically κ-fully linear with respect to the corresponding sequence $\{X_k, \Lambda_k\}$ if the events*

$$I_k = \{M_k \text{ is a } \kappa \text{ -fully linear model of } f \text{ around } X_k\}$$

satisfy the condition

$$\mathbb{P}(I_k | \mathcal{F}_{k-1}^{M\,F}) \geq \alpha,$$

where $\mathcal{F}_{k-1}^{M\,F}$ is the σ-algebra generated by $M_0, \ldots, M_{k-1}$ and $F_0, \ldots, F_{k-1}$.

Notice that imposing this condition on the fine level only will be enough to ensure convergence of the method. The first order correction imposed on the coarse model will ensure a link between the two approximations of f and consequently the coarse step will point in the good direction thanks to the link with the fine one[4], cf. (13). Thus the accuracy of the coarse approximations does not need to increase along with the iterations, or the increase can be much slower than at fine level.

We will also require function estimates to be sufficiently accurate.

Definition 3. *The estimates f_k and f_k^s are said to be ϵ_f-accurate estimates of $f(x_k)$ and $f(x_k + s_k)$ respectively, for a given λ_k if*

$$|f_k - f(x_k)| \leq \frac{\epsilon_f}{\lambda_k^2} \text{ and } |f_k^s - f(x_k + s_k)| \leq \frac{\epsilon_f}{\lambda_k^2}.$$

In particular we will consider probabilistically accurate estimates as in [20]:

Definition 4. *A sequence of random estimates $\{F_k, F_k^s\}$ is said to be β-probabilistically ϵ_f-accurate with respect to the corresponding sequence $\{X_k, \Lambda_k, S_k\}$ if the events*

$$J_k = \{F_k, F_k^s \text{ are } \epsilon_f\text{-accurate estimates of } f(x_k) \text{ and } f(x_k + s_k), \text{ respectively, around } X_k\}$$

satisfy the condition

$$\mathbb{P}(J_k | \mathcal{F}_{k-1/2}^{M\,F}) \geq \beta,$$

where ϵ_f is a fixed constant and $\mathcal{F}_{k-1/2}^{M\,F}$ is the σ-algebra generated by $M_0, \ldots, M_k$ and $F_0, \ldots, F_{k-1}$.

Following [20], in our analysis we will require that our method has access to α-probabilistically κ-fully linear models, for some fixed κ and to β-probabilistically ϵ_f accurate function estimates, for some fixed, sufficiently small ϵ_f. Cf. [20, Section 5] for procedures for constructing probabilistically fully linear models, and probabilistically accurate estimates. Basically, when the function approximations come from a subsampling this construction is possible if the model accounts for enough samples.

[4] If R_k is the identity, the coherence term makes the coarse models fully linear in a neighbourhood of x_k.

3.1 Convergence analysis

We start by recalling two useful relations, following from Taylor's theorem, see for example [17, Corollary A.8.4].

Lemma 1. *Let $h : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradient, with L the corresponding Lipschitz constant. Given its first order truncated Taylor series in x, $T[h](s) := h(x) + \nabla_x h(x)^T s$, it holds:*

$$h(x+s) = T[h](s) + \int_0^1 [\nabla_x h(x+\xi s) - \nabla_x h(x)]^T s \, d\xi, \tag{16}$$

$$|h(x+s) - T[h](s)| \leq \frac{L}{2}\|s\|^2, \tag{17}$$

We now propose two technical lemmas on the coarse step.

Lemma 2. *Let Assumptions 1 and 2 hold. Consider a realization of Algorithm 1 where at iteration k the coarse model is used and let $s_k = Ps^*$ be the resulting step. Then it holds:*

$$|\varphi_k(0) + \varphi_k(s^*) - g_k^T s_k| \leq \frac{L_\phi}{2}\|s^*\|^2. \tag{18}$$

Proof. Using the first order Taylor expansion of φ_k and (16) applied to φ_k, and considering that from (10), $\nabla_s \varphi_k(0)^T s^* = g_k^T s_k$, we can write:

$$\varphi_k(0) - \varphi_k(s^*) = -g_k^T s_k - \int_0^1 \left[\nabla_s \varphi_k(\xi s^*) - \nabla_s \varphi_k(0)\right]^T s^* \, d\xi.$$

Using Assumption 2 and recalling that φ_k and ϕ_k just differ by a linear term, we obtain:

$$\begin{aligned} |\varphi_k(0) - \varphi_k(s^*) + g_k^T s_k) - | &\leq \int_0^1 |\left[\nabla_s \varphi_k(\xi s^*) - \nabla_s \varphi(0)\right]^T s^*| \, d\xi \\ &\leq \int_0^1 \|\nabla_s \varphi_k(\xi s^*) - \nabla_s \varphi_k(0)\| \|s^*\| \, d\xi \leq \frac{L_\phi}{2}\|s^*\|^2. \end{aligned}$$

□

Lemma 3. *Under Assumptions 1 and 2, for any realization of Algorithm 1 and for each iteration k where the coarse step is used it holds:*

$$\|Rg_k\| \leq (L_\phi + \theta + \lambda_k \|g_k\|)\|s^*\|. \tag{19}$$

Proof. From (8) and (9), since $\nabla_s m_k^R(s) = \nabla_s \varphi_k(s) + \lambda_k \|g_k\| s$, it follows:

$$\begin{aligned} \|Rg_k\| &\leq \|Rg_k - \nabla_s \varphi_k(s^*)\| + \|\nabla_s \varphi_k(s^*) + \lambda_k \|g_k\| s^*\| + \lambda_k \|g_k\| \|s^*\| \\ &\leq \|Rg_k - \nabla_s \phi_k(R_k x_k + s^*) - Rg_k + \nabla_s \phi_k(Rx_k)\| + \theta\|s^*\| + \lambda_k \|g_k\| \|s^*\| \\ &\leq L_\phi \|s^*\| + \theta\|s^*\| + \lambda_k \|g_k\| \|s^*\|. \end{aligned}$$

Thus we finally obtain the thesis. □

The following lemma relates the coarse step size and the regularization parameter λ_k.

Lemma 4. *Let Assumptions 1 and 2 hold. Assume that at iteration k the coarse step is used. Assume that*

$$\frac{1}{\lambda_k} \leq \min\left\{\frac{1}{L_\phi + \theta}, \frac{1}{2L_\phi}\right\}\|g_k\|, \tag{20}$$

Then

$$\frac{\kappa_H}{2\lambda_k} \leq \|s^*\| \leq \frac{4\kappa_R}{\lambda_k}. \tag{21}$$

Proof. The first inequality follows from:

$$\|s^*\| \overset{(19)}{\geq} \frac{\|R_k g_k\|}{L_\phi + \theta + \lambda_k\|g_k\|} \overset{(12)}{\geq} \frac{\kappa_H\|g_k\|}{L_\phi + \theta + \lambda_k\|g_k\|} \overset{(20)}{\geq} \frac{\kappa_H}{2\lambda_k}. \tag{22}$$

The second inequality follows from (18):

$$|\varphi_k(s^*) - \varphi_k(0)| - |\nabla_s\varphi_k(0)^T s^*| \leq |\varphi_k(s^*) - \varphi_k(0) - \nabla_s\varphi_k(0)^T s^*| \leq \frac{L_\phi}{2}\|s^*\|^2,$$

where we have used the fact that from (9) and Assumption 2, φ_k is L_ϕ-smooth. Thus, from (13) and since $\varphi_k(0) \geq \varphi_k(s^*)$, we obtain

$$\begin{aligned}\varphi_k(0) - \varphi_k(s^*) &\leq |\nabla_s\varphi_k(0)^T s^*| + \frac{L_\phi}{2}\|s^*\|^2 = |g_k^T s_k| + \frac{L_\phi}{2}\|s^*\|^2 \\ &\leq \|g_k\|\|s_k\| + \frac{L_\phi}{2}\|s^*\|^2 \leq \kappa_R\|g_k\|\|s^*\| + \frac{L_\phi}{2}\|s^*\|^2.\end{aligned}$$

Combining this with (11) we have:

$$\frac{1}{2}\lambda_k\|g_k\|\|s^*\|^2 \leq \varphi_k(0) - \varphi_k(s^*) \leq \kappa_R\|g_k\|\|s^*\| + \frac{L_\phi}{2}\|s^*\|^2.$$

Thus

$$\left(\frac{1}{2}\lambda_k\|g_k\| - \frac{L_\phi}{2}\right)\|s^*\|^2 \leq \kappa_R\|g_k\|\|s^*\|.$$

From (20) we have $\frac{1}{2}\lambda_k\|g_k\| - \frac{L_\phi}{2} \geq \frac{1}{4}\lambda_k\|g_k\|$ and thus

$$\frac{1}{4}\lambda_k\|g_k\|\|s^*\| \leq \kappa_R\|g_k\|.$$

□

In the following lemma we measure the decrease predicted by the model.

Lemma 5. *Let Assumptions 1 and 2 hold. For any realization of Algorithm 1 and for each k it holds:*

$$m_k^s - m_k^0 \leq \begin{cases} -\frac{\|g_k\|}{\lambda_k} & \text{if fine step,} \\ -\frac{\lambda_k\|g_k\|}{2}\|s^*\|^2 & \text{if coarse step.} \end{cases} \tag{23}$$

Proof. If the fine step is used,

$$m_k^s - m_k^0 = g_k^T s_k = -\frac{\|g_k\|^2}{\lambda_k\|g_k\|} = -\frac{\|g_k\|}{\lambda_k}.$$

If the coarse step is used:

$$m_k^s - m_k^0 = \varphi_k(s^*) - \varphi_k(0) \overset{(11)}{\leq} -\frac{\lambda_k\|g_k\|}{2}\|s^*\|^2.$$

□

We now prove some auxiliary lemmas that provide conditions under which the decrease of the true objective function f is guaranteed. The first lemma states that if the regularization parameter is large enough relative to the size of the fine model gradient and if the fine model is fully linear, then the step s_k provides a decrease in f proportional to the size of the model gradient.

Lemma 6. *Under Assumptions 1 and 2, suppose that $m_k(x_k + s) := f_k + g_k^T s$ is a (κ_f, κ_g)-fully linear model of f in a neighbourhood of x_k. If*

$$\frac{1}{\lambda_k} \leq \min\left\{\frac{1}{L_\phi + \theta}, \frac{\kappa_H^2}{64\kappa_f}, \frac{1}{2L_\phi}\right\}\|g_k\|, \tag{24}$$

then the trial step s_k leads to an improvement in $f(x_k + s_k)$ such that

$$f(x_k + s_k) - f(x_k) \leq -\frac{\kappa_H^2}{32}\frac{\|g_k\|}{\lambda_k}.$$

Proof. We distinguish two cases depending on the used step.

1. In the fine step case, we get

$$\begin{aligned} f(x_k + s_k) - f(x_k) &= f(x_k + s_k) - m_k(x_k + s_k) + m_k(x_k + s_k) - m_k(x_k) + m_k(x_k) - f(x_k) \\ &\overset{(15)+(23)}{\leq} \frac{2\kappa_f}{\lambda_k^2} - \frac{\|g_k\|}{\lambda_k} \overset{(24)}{\leq} -\frac{1}{2}\frac{\|g_k\|}{\lambda_k} \leq -\frac{\kappa_H^2}{32}\frac{\|g_k\|}{\lambda_k}, \end{aligned}$$

   where we have used that, from (24), $\frac{1}{\lambda_k} \leq \frac{\kappa_H^2}{64\kappa_f}\|g_k\| \leq \frac{1}{4\kappa_f}\|g_k\|$, since $\kappa_H \leq 1$.

2. When the coarse step is used, we have, for m_k the fine model, that

$$\begin{aligned} f(x_k + s_k) - f(x_k) = {} & f(x_k + s_k) - m_k(x_k + s_k) \\ & + m_k(x_k + s_k) - m_k(x_k) - \varphi_k(s^*) + \varphi_k(0) \\ & - \varphi_k(0) + \varphi_k(s^*) \\ & + m_k(x_k) - f(x_k). \end{aligned}$$

   The first and the last terms are bounded by $\frac{\kappa_f}{\lambda_k^2}$ from (15). The second term from Lemma 2 is bounded by $\frac{L_\phi}{2}\|s^*\|^2$. The third term is bounded by $-\frac{\lambda_k\|g_k\|}{2}\|s^*\|^2$ from (11). Thus

$$\begin{aligned} f(x_k + s_k) - f(x_k) &\leq \frac{2\kappa_f}{\lambda_k^2} + \left(\frac{L_\phi}{2} - \frac{\lambda_k\|g_k\|}{2}\right)\|s^*\|^2 \\ &\overset{(24)}{\leq} \frac{2\kappa_f}{\lambda_k^2} - \frac{\lambda_k\|g_k\|}{4}\|s^*\|^2 \\ &\overset{(21)}{\leq} \frac{2\kappa_f}{\lambda_k^2} - \frac{\|g_k\|\kappa_H^2}{16\lambda_k} \\ &\overset{(24)}{\leq} -\frac{\kappa_H^2}{32}\frac{\|g_k\|}{\lambda_k}. \end{aligned}$$

□

The next lemma shows that for a sufficiently large regularization parameter λ_k relative to the size of the true gradient $\nabla_x f(x_k)$, the guaranteed decrease in the objective function, provided by s_k, is proportional to the size of the true gradient.

Lemma 7. *Let Assumptions 1 and 2 hold and suppose that m_k is a (κ_f, κ_g)-fully linear model of f in a neighbourhood of x_k. If*

$$\frac{1}{\lambda_k} \leq \min\left\{\frac{1}{L_\phi + \theta + \kappa_g}, \frac{1}{(64\kappa_f/\kappa_H^2) + \kappa_g}, \frac{1}{2L_\phi + \kappa_g}\right\}\|\nabla_x f(x_k)\|, \tag{25}$$

then the trial step s_k leads to an improvement in $f(x_k + s_k)$ such that

$$f(x_k + s_k) - f(x_k) \leq -C_1 \frac{\|\nabla_x f(x_k)\|}{\lambda_k},$$

with $C_1 := \frac{\kappa_H^2}{32} \max\left\{\frac{L_\phi+\theta}{L_\phi+\theta+\kappa_g}, \frac{64\kappa_f}{64\kappa_f+\kappa_g\kappa_H^2}, \frac{2L_\phi}{2L_\phi+\kappa_g}\right\}$.

Proof. We first prove that the assumption of Lemma 6 is satisfied, and we use its result to deduce the decrease of the objective function in terms of $\|\nabla_x f(x_k)\|$ rather than $\|g_k\|$, by linking these two quantities through the assumption of κ-fully linear model, which yields that

$$\|g_k\| \geq \|\nabla_x f(x_k)\| - \frac{\kappa_g}{\lambda_k}. \tag{26}$$

From assumption (25) it holds

$$\|\nabla_x f(x_k)\| \geq \max\{L_\phi + \theta + \kappa_g, 64\kappa_f/\kappa_H^2 + \kappa_g, 2L_\phi + \kappa_g\}\frac{1}{\lambda_k},$$

and thus from (26) we have

$$\|g_k\| \geq \|\nabla_x f(x_k)\| - \frac{\kappa_g}{\lambda_k} \geq \max\{L_\phi + \theta, 64\kappa_f/\kappa_H^2, 2L_\phi\}\frac{1}{\lambda_k}.$$

Thus the assumption of Lemma 6 is satisfied and

$$f(x_k + s_k) - f(x_k) \leq -\frac{\kappa_H^2}{32}\frac{\|g_k\|}{\lambda_k}.$$

In the same way from (25) and (26) we have

$$\begin{aligned}
\|g_k\| &\geq \|\nabla_x f(x_k)\| - \frac{\kappa_g}{\lambda_k} \\
&\geq \|\nabla_x f(x_k)\| - \kappa_g \min\left\{\frac{1}{L_\phi + \theta + \kappa_g}, \frac{1}{64\kappa_f/\kappa_H^2 + \kappa_g}, \frac{1}{2L_\phi + \kappa_g}\right\}\|\nabla_x f(x_k)\| \\
&= \max\left\{\frac{L_\phi + \theta}{L_\phi + \theta + \kappa_g}, \frac{64\kappa_f}{64\kappa_f + \kappa_g\kappa_H^2}, \frac{2L_\phi}{2L_\phi + \kappa_g}\right\}\|\nabla_x f(x_k)\| := \tilde{C}_1\|\nabla_x f(x_k)\|.
\end{aligned}$$

We conclude that

$$f(x_k + s_k) - f(x_k) \leq -\frac{\kappa_H^2}{32}\frac{\|g_k\|}{\lambda_k} \leq -\frac{\kappa_H^2\tilde{C}_1}{32}\frac{\|\nabla_x f(x_k)\|}{\lambda_k} := -C_1\frac{\|\nabla_x f(x_k)\|}{\lambda_k}.$$

□

We now prove a lemma that states that, if the estimates are sufficiently accurate, the fine model is fully-linear and the regularization parameter is large enough relatively to the size of the model gradient, then a successful step is guaranteed.

Lemma 8. *Let Assumptions 1 and 2 hold. Suppose that m_k is a (κ_f, κ_g)-fully linear model in a neighbourhood of x_k and that the estimates $\{f_k, f_k^s\}$ are ϵ_f-accurate with $\epsilon_f \leq \kappa_f$. If*

$$\frac{1}{\lambda_k} \leq \min\left\{\frac{1}{L_\phi + \theta}, \frac{1}{\eta_2}, \frac{1-\eta_1}{32\kappa_f/\kappa_H^2 + L_\phi}\right\}\|g_k\|, \tag{27}$$

then the k-th iteration is successful.

Proof. Let us consider ρ_k in Step 6 of Algorithm 1. When the fine step is used, it holds:

$$\rho_k = \frac{f_k - f_k^s}{m_k^0 - m_k^s} = \frac{m_k^0 - m_k^s}{m_k^0 - m_k^s} + \frac{m_k^s - f(x_k + s_k)}{m_k^0 - m_k^s} + \frac{f(x_k + s_k) - f_k^s}{m_k^0 - m_k^s}. \tag{28}$$

From the assumption on the function estimates (cf. Definition 3) it follows:

$$|f_k^s - f(x_k + s_k)| \leq \frac{\epsilon_f}{\lambda_k^2} \leq \frac{\kappa_f}{\lambda_k^2}.$$

From the assumption of κ-fully linearity of the model the numerator of the second term is bounded by (15). Consequently, the numerator of $|\rho_k - 1|$ is bounded by $\frac{2\kappa_f}{\lambda_k^2}$. The denominator is bounded from (23). Thus by (27)

$$|\rho_k - 1| \leq \frac{2\kappa_f}{\lambda_k\|g_k\|} \leq 1 - \eta_1.$$

If the coarse step is used we have:

$$\begin{aligned}\rho_k &= \frac{f_k - f_k^s}{m_k^0 - m_k^s} \\ &= \frac{f_k + g_k^T s_k - f(x_k + s_k)}{\varphi_k(0) - \varphi_k(s^*)} + \frac{\varphi_k(s^*) - \varphi_k(0) - g_k^T s_k}{\varphi_k(0) - \varphi_k(s^*)} + \frac{\varphi_k(0) - \varphi_k(s^*)}{\varphi_k(0) - \varphi_k(s^*)} \\ &+ \frac{f(x_k + s_k) - f_k^s}{\varphi_k(0) - \varphi_k(s^*)}.\end{aligned}$$

Considering the first and the last term, the numerators can be bounded as in the first case. We thus have

$$\left|\frac{f_k + g_k^T s_k - f(x_k + s_k)}{\varphi_k(0) - \varphi_k(s^*)} + \frac{f(x_k + s_k) - f_k^s}{\varphi_k(0) - \varphi_k(s^*)}\right| \overset{(11)}{\leq} \frac{\frac{2\kappa_f}{\lambda_k^2}}{\frac{\lambda_k\|g_k\|}{2}\|s^*\|^2} \overset{(21)}{\leq} \frac{\frac{2\kappa_f}{\lambda_k^2}}{\frac{\lambda_k\|g_k\|}{2}\frac{\kappa_H^2}{4\lambda_k^2}} = \frac{32\kappa_f}{\lambda_k\|g_k\|\kappa_H^2}.$$

For the second term we have:

$$\left|\frac{\varphi_k(s^*) - \varphi_k(0) - g_k^T s_k}{m_k^0 - m_k^s}\right| \overset{(18)+(11)}{\leq} \frac{\frac{L_\phi}{2}\|s^*\|^2}{\frac{\lambda_k\|g_k\|}{2}\|s^*\|^2} = \frac{L_\phi}{\lambda_k\|g_k\|}.$$

Thus from (27)

$$|\rho_k - 1| \leq \frac{32\kappa_f/\kappa_H^2 + L_\phi}{\lambda_k\|g_k\|} \leq 1 - \eta_1.$$

Hence in every case $\rho_k \geq 1$. Moreover, since $\|g_k\| \geq \frac{\eta_2}{\lambda_k}$ from (27), the k-th iteration is successful. □

Finally, we state and prove the lemma that guarantees an amount of decrease of the objective function on a true successful iteration.

Lemma 9. *Under Assumptions 1 and 2 , suppose that the estimates $\{f_k, f_k^s\}$ are ϵ_f-accurate with $\epsilon_f < \frac{\eta_1\eta_2\kappa_H^2}{16}$. If a trial step s_k is accepted then the improvement in f is bounded below by:*

$$f(x_{k+1}) - f(x_k) \leq -\frac{C_2}{\lambda_k^2}$$

where $C_2 = \frac{\eta_1\eta_2\kappa_H^2}{8} - 2\epsilon_f > 0$.

Proof. If the iteration is successful, this means that $\|g_k\| \geq \frac{\eta_2}{\lambda_k}$ and $\rho_k \geq \eta_1$. Thus, if the fine step is used,

$$f_k - f_k^s \geq \eta_1(m_k^0 - m_k^s) \overset{(23)}{\geq} \eta_1 \frac{\|g_k\|}{\lambda_k} \geq \frac{\eta_1\eta_2}{\lambda_k^2}.$$

If the coarse step is used

$$\begin{aligned} f_k - f_k^s \geq \eta_1(m_k^0 - m_k^s) &\overset{(23)}{\geq} \frac{\eta_1}{2}\lambda_k\|g_k\|\|s^*\|^2 \\ &\overset{(21)}{\geq} \frac{\eta_1\kappa_H^2}{8}\|g_k\|\frac{1}{\lambda_k} \geq \frac{\eta_1\eta_2\kappa_H^2}{8}\frac{1}{\lambda_k^2}. \end{aligned}$$

Then, since the estimates are ϵ_f-accurate, we have that the improvement in f can be bounded as

$$f(x_k + s_k) - f(x_k) = f(x_k + s_k) - f_k^s + f_k^s - f_k + f_k - f(x_k) \leq -\frac{C_2}{\lambda_k^2},$$

where $C_2 = \frac{\eta_1\eta_2\kappa_H^2}{8} - 2\epsilon_f > 0$. □

To prove convergence of Algorithm 1 we need to assume that the fine-level models $\{M_k\}$ and the estimates $\{F_k, F_k^s\}$ are sufficiently accurate with sufficiently high probability.

Assumption 3. *Given values of $\alpha, \beta \in (0, 1)$ and $\epsilon_f > 0$, there exist κ_g and κ_f such that the sequence of fine-level models $\{M_k\}$ and estimates $\{F_k, F_k^s\}$ generated by Algorithm 1 are, respectively, α-probabilistically (κ_f, κ_g)-fully-linear and β-probabilistically ϵ_f-accurate.*

The following theorem states that the regularization parameter λ_k converges to $+\infty$ with probability one. Together with its corollary it gives conditions on the existence of κ_g and κ_f given α, β and ϵ_f.

Theorem 1. *Let Assumptions 1, 2 and 3 be satisfied and assume that in Algorithm 1 the following holds.*

- *The step acceptance parameter η_2 is chosen so that*

$$\eta_2 \geq \max\{L_\phi + \theta, 24\kappa_f\}.$$

- *The accuracy parameter of the estimates satisfies*

$$\epsilon_f \leq \min\left\{\kappa_f, \frac{\eta_1\eta_2\kappa_H^2}{32}\right\}.$$

Then α and β can be chosen so that, if Assumption 3 holds for these values, then the sequence of regularization parameters $\{\Lambda_k\}$ generated by Algorithm 1 satisfies

$$\sum_{k=0}^{\infty} \frac{1}{\Lambda_k^2} < \infty$$

almost surely.

Proof. The scheme of the proof is the same as that of [20, Theorem 4.11]. We outline here just the differences. Let C_1 be defined as in Lemma 7. We define

$$h_k = \nu f(X_k) + (1-\nu)\frac{1}{\Lambda_k^2}$$

with $\nu \in (0,1)$ such that

$$\frac{\nu}{1-\nu} > \max\left\{\frac{4}{\gamma^2 \zeta C_1}, \frac{16}{\gamma^2 \eta_1 \eta_2 \kappa_H^2}, \frac{1}{\gamma^2 3\kappa_f}\right\}$$

where

$$\zeta \geq \kappa_g + \max\left\{\eta_2, \frac{64\kappa_f/\kappa_H^2 + L_\phi}{1-\eta_1}\right\}.$$

Under this assumption, the results in the proof of [20, Theorem 4.11] hold with

$$\begin{aligned}
b_1 &:= (1-\nu)(\gamma^2-1)\frac{1}{\lambda_k^2}, \\
b_2 &:= -\nu C_1 \|\nabla_x f(x_k)\| \frac{1}{\lambda_k} + (1-\nu)\left(\frac{1}{\gamma^2}-1\right)\frac{1}{\lambda_k^2}, \\
b_3 &:= \nu C_3 \|\nabla_x f(x_k)\| \frac{1}{\lambda_k} + (1-\nu)\left(\frac{1}{\gamma^2}-1\right)\frac{1}{\lambda_k^2},
\end{aligned}$$

with $C_3 := \frac{40L_f}{\zeta} + 4$. In particular, there exists $\sigma > 0$ such that for all k

$$\mathbb{E}[h_{k+1} - h_k | \mathcal{F}_{k-1}^{MF}] \leq -\sigma \frac{1}{\Lambda_k^2} < 0.$$

□

The choice of α and β is specified in the following corollary.

Corollary 1. *Let all assumptions of Theorem 1 hold. The statement of Theorem 1 holds if α and β are chosen to satisfy the following conditions:*

$$\frac{\alpha\beta - \frac{1}{2}}{(1-\alpha)(1-\beta)} \geq \frac{\frac{40L_f}{\zeta} + 4}{C_1}$$

and

$$(1-\alpha)(1-\beta) \leq \frac{\frac{1}{\gamma^2} - 1}{\frac{1}{\gamma^4} - 1 + \frac{1}{\gamma^2}(40L_f + 4\zeta)\max\left\{\frac{4}{\zeta C_1}, \frac{16}{\eta_1\eta_2\kappa_H^2}, \frac{1}{3\kappa_f}\right\}},$$

with $C_1 = \frac{\kappa_H^2}{32}\max\left\{\frac{L_\phi+\theta}{L_\phi+\theta+\kappa_g}, \frac{64\kappa_f}{64\kappa_f+\kappa_g\kappa_H^2}, \frac{2L_\phi}{2L_\phi+\kappa_g}\right\}$ and $\zeta = \kappa_g + \eta_2$.

The following results can be derived as in [20, Lemma 4.17] and [20, Theorem 4.18], their proof is therefore omitted for sake of brevity.

Theorem 2. *Let the assumptions of Theorem 1 hold. Let $\{X_k\}$ and $\{\Lambda_k\}$ be the sequences of random iterates and random regularization parameters generated by Algorithm 1. Fix $\epsilon > 0$ and define the sequence $\{K_\epsilon\}$ consisting of the natural numbers k for which $\|\nabla_x f(X_k)\| > \epsilon$. Then almost surely*

$$\sum_{k \in \{K_\epsilon\}} \frac{1}{\Lambda_k} < \infty.$$

Theorem 3. *Let the assumptions of Theorem 1 hold. Let $\{X_k\}$ be the sequence of random iterates generated by Algorithm 1. Then, almost surely,*

$$\lim_{k \to \infty} \|\nabla_x f(X_k)\| = 0.$$

4 MU$^\ell$STREG for finite sum minimization

In this section we describe how to adapt Algorithm 1 to the solution of finite sum minimization problems of the form

$$\min_{x \in \mathbb{R}^n} F(x) = \frac{1}{N} \sum_{i=1}^{N} f^{(i)}(x) \tag{29}$$

using a multilevel setting with ℓ levels. We assume that $N \gg n$ and we consider hierarchies built just in the samples space, thus at each level the iterates belong to $\mathbb{R}^n$ and the operators R_k and P_k are the identity. We thus drop here the indexes ℓ from the iterates and the steps.

Since the objective function in (29) is the average of the set of functions $\left\{f^{(i)}\right\}_{i=1}^{N}$, we can easily define a hierarchy of approximations by subsampling. In particular, given the number of levels $\ell_{\max} \geq 2$, for every $\ell \in \{1, ..., \ell_{\max}\}$ we define the subsampled function as:

$$f^{S_k^\ell}(x) := \frac{1}{|\mathcal{S}_k^\ell|} \sum_{i \in \mathcal{S}_k^\ell} f^{(i)}(x); \tag{30}$$

where $\mathcal{S}_k^\ell$ is a subsample set such that $\emptyset \neq \mathcal{S}_k^1 \subset ... \subset \mathcal{S}_k^\ell \subset ... \subset \mathcal{S}_k^{\ell_{\max}-1} \subset \mathcal{S}_k^{\ell_{\max}} \subseteq \{1, ..., N\}$.

The stochastic framework of Algorithm 1 allows for inexact approximations of F at the finest level, thus avoiding the need of the full sample evaluation at each iteration. Following [20], in order to ensure that the model at fine level remains a fully linear model for F we choose, at iteration k, $\mathcal{S}_k^{\ell_{\max}}$ a set of randomly drawn samples such that $|\mathcal{S}_k^{\ell_{\max}}| = p_k := \min\{N, \max\{100k + n + 2, \lambda_k^2\}\}$. The lower level approximations ϕ_k^ℓ are defined as in (30) with $\mathcal{S}_k^\ell \subset \mathcal{S}_k^{\ell_{\max}}$ and $|\mathcal{S}_k^\ell|$ a fixed fraction[5] of p_k. The cardinality of the sample set thus changes at each fine iteration, and the increase is much slower at the coarse levels.

In order to define the objective functions for each level ℓ, we introduce the following notation. Given a function g, we define $[g]^{\mathcal{S}}$ its subsampled version, i.e., if g is the average of functions $g^{(i)}$ over a sample set including $\mathcal{S}$, $[g]^{\mathcal{S}}$ is the average of the $g^{(i)}$ restricted to just the samples in $\mathcal{S}$. If g is not defined on a sample set, $[g]$ is just g.

We use the functions $\left\{f^{S_k^\ell}\right\}_{\ell=1}^{\ell_{\max}}$ to define the regularized models that are minimized at each level in a recursive way. In particular, at level $1 < \ell \leq \ell_{\max}$, given the objective function of that

[5] Note that they could as well be chosen constant.

level f_k^ℓ and an iterate x_k, we define the objective function $m_k^{R,\ell-1}$ at x_k for the lower level $\ell-1$ as

$$m_k^{R,\ell-1}(s) = \left[f_k^\ell\right]^{\mathcal{S}_k^{\ell-1}}(x_k+s) + (v_k^{\ell-1})^T s + \frac{1}{2}\lambda_k^\ell \left\|\nabla_x f_k^\ell(x_k)\right\| \|s\|^2 \tag{31}$$

with $\lambda_k^\ell > 0$ and

$$v_k^{\ell-1} = \nabla_x f_k^\ell(x_k) - \nabla_x \left[f_k^\ell\right]^{\mathcal{S}^{\ell-1}}(x_k). \tag{32}$$

At the finest level $f_k^\ell = f^{\mathcal{S}_k^{\ell_{\max}}}$, thus $\left[f^{\mathcal{S}_k^{\ell_{\max}}}\right]^{\mathcal{S}_k^{\ell_{\max}-1}}$ is simply $f^{\mathcal{S}_k^{\ell_{\max}-1}}$. However, when $\ell < \ell_{\max}$, f_k^ℓ represents the regularized model built at the immediately upper level $\ell+1$ and incorporates also the regularization and the vector $v_k^{\ell+1}$. Given that this quantities are not defined on a samples set, the subsampled version of f_k^ℓ will contain the term $f^{\mathcal{S}_k^\ell}$ that is subsampled on $\mathcal{S}_k^{\ell-1}$, while the correction and the regularization vectors remain unchanged[6].

The stopping criterion depends on the level. At fine level it checks if the norm of the approximated gradient is below some tolerance ϵ:

$$\left\|\nabla_x f^{\mathcal{S}_k^{\ell_{\max}}}(x_k)\right\| \leq \epsilon. \tag{33}$$

If the full gradient is not evaluated (i.e., $|\mathcal{S}_k^{\ell_{\max}}| < N$), case expected for most of the fine iterations, the stopping criterion (33) might not be meaningful. Therefore we use a heuristic stopping test. When (33) is satisfied for the first time, after a fine or a coarse step, a new set of p_k randomly chosen samples is drawn and fine steps are taken until (33) is satisfied again. In practice the stopping criterion is however, in most cases, satisfied when the full sample size is reached. A safeguard is also added that imposes a maximum number of fine iterations. If $\ell < \ell_{\max}$, we use the stopping condition (4) where $m_k^{R,\ell}$ is defined in (31), with $\epsilon^{\ell-1} > 0$, and impose a maximum number maxit$_\ell$ of iterations.

We report in Algorithm 2 the complete MU$^\ell$STREG algorithm for problem (29). The call to the algorithm at fine level is MU$^\ell$STREG($\ell_{\max}$,x_0,-,$\epsilon^{\ell_{\max}}$,$\lambda_0^{\ell_{\max}}$,-), with $x_0 \in \mathbb{R}^n$. Note that the choice of the alternate scheme between the coarse and fine steps is left to the user.

5 Numerical experiments

In this section we illustrate the performance of MU$^\ell$STREG for the solution of finite sum minimization problems (29) arising in binary classification.

Considered methods We use 3 levels (adding more levels did not improve the results) and we rename our method MU^{3}STREG. We compare MU^{3}STREG against the 1-level MU^{1}STREG. We remark that MU^{1}STREG can be interpreted as a STORM-like approach where the trust region constraint is replaced by a quadratic regularization, resulting in fact in an adaptive sampling strategy.

We also consider two first-order stochastic algorithms widely used in the solution of (29): SVRG [31] and Adagrad [22, 36]. The choice of SVRG is due to the variance reduction interpretation of multilevel methods mentioned in the introduction, while Adagrad was chosen among the Ada-like methods for its good complexity properties [25]. As our approach, Adagrad uses an adaptive strategy for the stepsize, while SVRG uses a constant steplength (or learning rate) α. Both algorithms use a mini-batch of size b for evaluating the approximated gradient but SVRG also need to re-evaluate the full gradient every m iterations [31].

[6]Notice that each time we go down a level we accumulate in the regularized model a regularization term and a correction vector.

Performance measures In order to compare the efficiency of the various methods, we consider the number of weighted gradient and function evaluations performed during the execution: a full-gradient evaluation is counted as 1, while a function evaluation is counted as $\frac{1}{n}$, taking into account that the size of the gradients is n. Therefore, the sub-sampled gradients are weighted as $\frac{|\mathcal{S}_k^\ell|}{N}$, the size of the sub-sample set. From now on, we will refer to this measure as computational effort or more simply weighted number of evaluations, which will be denoted by **#f/g**.

Beside the efficiency of the methods, we also take into account the quality of the solutions found. In particular, we consider the classification accuracy (in percentage) on the testing set that will be denoted by **%tA**.

Implementation issues Both the versions of MU$^\ell$STREG with one and three levels have been implemented following Algorithm 2 with parameters chosen as follows:

$$\eta_1 = 0.5,\ \eta_2 = 10^{-3},\ \eta_3 = 0.75,\ \gamma_1 = 0.5,\ \gamma_2 = 0.3,\ \gamma_3 = 2,\ \lambda_{\min} = 10^{-4}.$$

Moreover, for the 3 level version, we set $|\mathcal{S}_k^1| = 0.001\, p_k$ and $|\mathcal{S}_k^2| = 0.01\, p_k$. The recursion scheme encompasses a fine step after each recursive call, as depicted in Figure 2 where the horizontal arrows represent the fine steps.

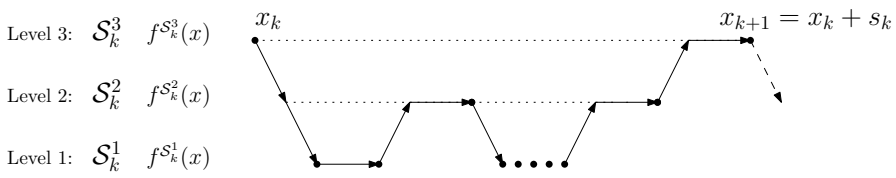

Figure 2: Iteration scheme used in our implementation of MU$^\ell$STREG for problem (29).

Algorithm 2 terminates when condition (33) holds with $\epsilon = 10^{-3}$ and we set $\epsilon^{\ell-1} = 10^{-3}$ in (4) and a maximum number of coarse iterations $\text{maxit}_\ell = 5$. Finally, we set $\lambda_0^\ell = 10^{-4}$ for $\ell > 1$ and $\lambda_0^1 = 10^{-3}$.

For the implementation of SVRG and Adagrad, we followed [39, 43] and, also taking into account our own extensive tuning work, we set the mini-batch size $b = 20$ for both methods and chose the learning rate $\alpha = 0.01$ and $m = N/b$ for SVRG. Regarding the stopping criterion, SVRG terminates when the norm of the gradient computed on the full sample set is below $\epsilon = 10^{-3}$ or 10^4 iterations are performed. Imposing a stopping criterion on Adagrad is less obvious and we only set a maximum number of weighted number of gradients and functions.

All algorithms have been implemented in MATLAB R2024a using HPE ProLiant DL560 Gen10 with 4 Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz with 512 Gb RAM[7].

Results on binary classification problems We consider a binary classification problem where the objective function is obtained by composing a least-squares loss with the sigmoid function, that is

$$\min_{x\in\mathbb{R}^n} F(x) = \frac{1}{2N}\sum_{i=1}^{N}\left(y_i - \frac{1}{1+\exp(-y_i x^T z_i)}\right)^2, \tag{Pb-FS}$$

with $(z_i, y_i) \in \mathbb{R}^n \times \{0,1\}$, for every $i = 1,\dots,N$. The tests are performed with four different datasets for binary classification: `MNIST` [35], `MUSH` [1], `A9A` and `IJCNN1` [19]. The data sets are divided into a training set and a testing set as specified in Table 1.

[7]We kindly acknowledge the Department of Mathematics of the University of Bologna for making the department's HPC resources available for this work.

Data set	nr. of features (n)	Training set size (N)	Testing set size (N_t)
MNIST	784	60000	10000
MUSH	112	6503	1621
A9A	123	22793	9768
IJCNN1	22	49990	91701

Table 1: Data sets with number of features n and number of instances of the training set N and the testing set N_t.

For each dataset we perform five tests with random initial guesses and then we show the averaged values (see Table 2 and Figures 4-5). In Table 2 we consider the computational effort **#f/g** of the solvers to satisfy the convergence stopping criteria[8].

MUSH			
	SVRG	MU^{1}STREG	MU^{3}STREG
Avg. %tA	98.69	97.68	97.74
StD %tA	0.25	0.50	0.48
Avg. #f/g	341.89	216.78	35.48
StD #f/g	722.65	30.69	9.95
A9A			
	SVRG	MU^{1}STREG	MU^{3}STREG
Avg. %tA	85.00	84.65	84.83
StD %tA	0.05	0.04	0.07
Avg. #f/g	840.77	207.68	90.87
StD #f/g	300.51	53.97	30.31
IJCNN1			
	SVRG	MU^{1}STREG	MU^{3}STREG
Avg. %tA	91.68	91.10	91.13
StD %tA	0.00	0.25	0.09
Avg. #f/g	553.87	6.20	7.09
StD #f/g	1.64	0.97	0.64
MNIST			
	SVRG	MU^{1}STREG	MU^{3}STREG
Avg. %tA	90.43	89.80	89.82
StD %tA	0.13	0.04	0.05
Avg. #f/g	17843.40	2923.02	414.15
StD #f/g	6336.56	497.15	16.37

Table 2: (Pb-FS) Comparison between MU^{1}STREG, MU^{3}STREG and SVRG. Average of maximum classification accuracy reached and number of evaluations, with corresponding standard deviation.

We first observe that MU^{3}STREG is in general much more efficient than the one-level version, while reaching the same accuracy level. Also, Figure 3 shows the evolution of the cardinality p_k of the sample set at the finest level along the finest iterations, for one illustrative run of the five performed. We can observe that MU^{3}STREG often requires a smaller sample set, especially in the first phase of the iteration history.

Moreover, both MU$^\ell$STREG solvers requires definitely less weighted number of evaluations than SVRG with similar solution quality. We also remark that the reported results for SVRG are the best obtained after a consistent tuning work for the learning rate α, while the stepsize selection is automatic in our approach.

[8]Adagrad is not included, since the stopping criterion is based on a maximum number of function evaluations only.

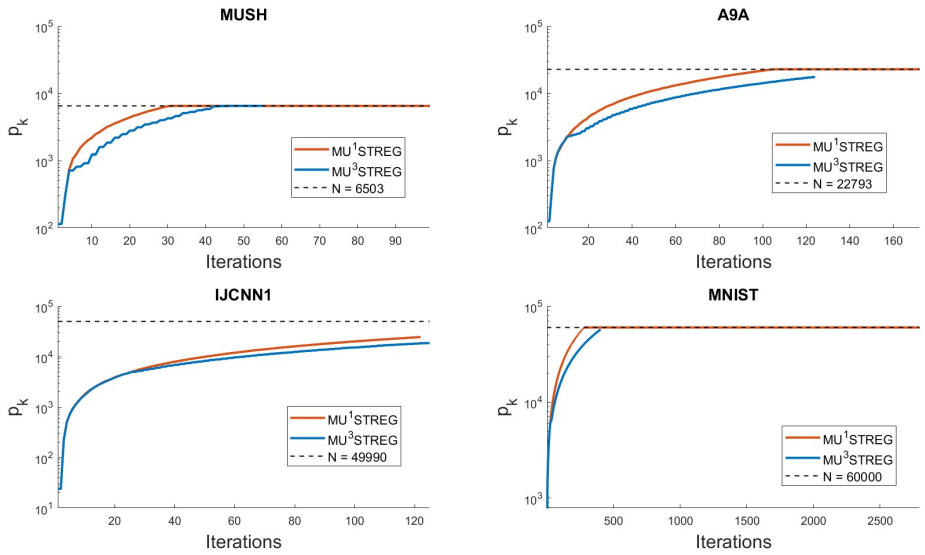

Figure 3: (Pb-FS) Cardinality of the sample set at the finest level for MU^{1}STREG and MU^{3}STREG and the full size N along the iterations.

We now compare our method with Adagrad, imposing a maximum budget of 100 weighted function and gradient evaluations **#f/g** [9]. Figures 4 and 5 show the value of the objective function and of the classification accuracy along **#f/g**, respectively. Here, the solid curves represent the mean values and the shaded area takes into account the standard deviation. Figure 4 shows that the objective decrease is faster for MU$^\ell$STREG solvers than for Adagrad, while Figure 5 highlights an oscillatory behaviour of Adagrad that reaches a lower accuracy than MU$^\ell$STREG (especially for the `A9A` and `MNIST` datasets) and comparable in case of `MUSH` .

6 Conclusions

We have proposed a new framework for the multilevel solution of stochastic problems, assuming that the stochastic objective function admits a hierarchical representation. Our framework encompasses both hierarchies in the variable space and in the function space, meaning that the function can be represented at different levels of accuracy.

We propose MU$^\ell$STREG, a new multilevel stochastic gradient method based on adaptive regularization that generalizes the AR1 method [17] and we propose a stochastic convergence analysis for it. This convergence theory is the first stochastic convergence study for multilevel methods.

We detailed our method for the solution of finite sum minimization problems and we made experiments on binary classification problems. We show that the proposed multi-level method outperforms the adaptive sampling one-level counterpart and is competitive with the tested subsampling methods (Adagrad and SVRG).

In particular, the finite sum minimization setting allows us to show the main advantage of a

[9] We do not consider SVRG in this analysis, as the required budget to get an accurate solution for SVRG is much larger than 100 and therefore the plots of its performance curves are unreadable.

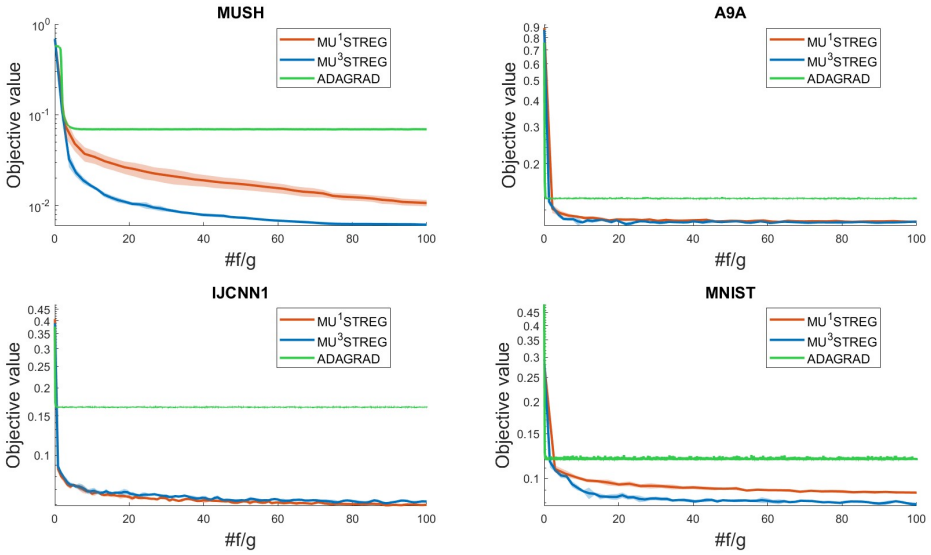

Figure 4: (Pb-FS) Objective function value along the number of weighted evaluations of gradients and functions.

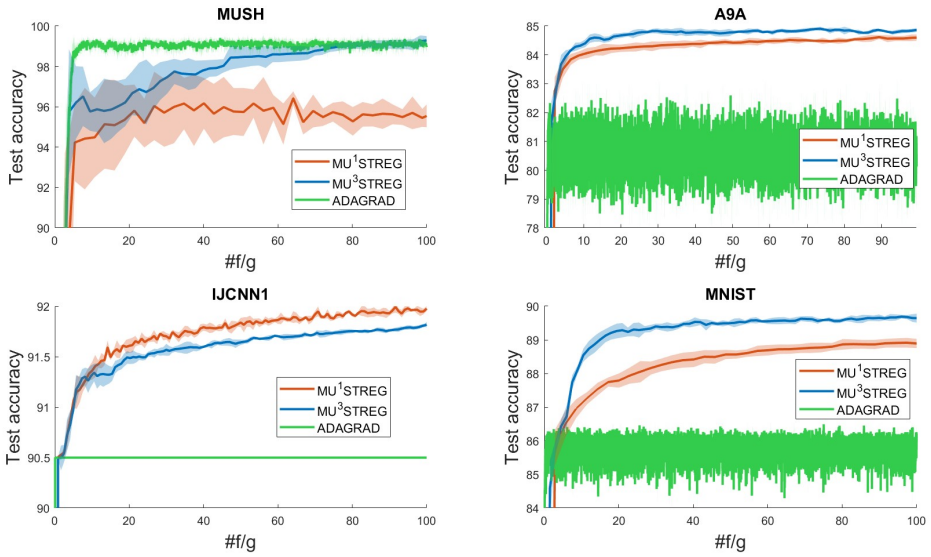

Figure 5: (Pb-FS) Classification accuracy on the testing set along the number of weighted evaluations of gradients and functions.

stochastic multilevel framework with respect to the classical deterministic one: it does not need the evaluation of the function/gradient over the full samples set along all the iterations.

Our framework covers different contexts (cf. the examples in section 2). Testing its effectiveness in other practical contexts (Montecarlo simulations, hierarchies in the variables space,

notably, e.g., when random projection operators are used) is an open research direction.

Moreover, our analysis assumes that the stochastic objective function f admits a hierarchy of computable approximations, i.e., that the functions ϕ_k^ℓ, once drawn randomly, are deterministic. The development of a fully stochastic analysis, where the functions ϕ_k^ℓ are assumed to be stochastic like f, is another meaningful perspective.

Acknowledgments

The authors wish to thank the anonymous referee for his/her careful reading and suggestions, which led to significant improvement of the manuscript.

The work of the first and second author was partially supported by INdAM-GNCS under the INdAM-GNCS project CUP_E53C22001930001. Part of the work of the F.M. was started during the author's Ph.D. thesis at the Università di Bologna supported by the program "Programma Operativo Nazionale Ricerca e Innovazione 2014-2020 (CCI2014IT16M2OP005)" - Azione IV.5 "Dottorati e contratti di ricerca su tematiche green" XXXVII ciclo, code DOT1303154-4, CUP J35F21003200006. The research of M.P. was partially granted by the Italian Ministry of University and Research (MUR) through the PRIN 2022 "MOLE: Manifold constrained Optimization and LEarning", code: 2022ZK5ME7 MUR D.D. financing decree n. 20428 of Nov. 6th, 2024 (CUP B53C24006410006), and by PNRR - Missione 4 Istruzione e Ricerca - Componente C2 Investimento 1.1, Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN) funded by the European Commission under the NextGeneration EU programme, project "Advanced optimization METhods for automated central veIn Sign detection in multiple sclerosis from magneTic resonAnce imaging (AMETISTA)", code: P2022J9SNP, MUR D.D. financing decree n. 1379 of 1st Sept. 2023 (CUP E53D23017980001). The work of E.R. was partially funded by the Fondation Simone et Cino Del Duca - Institut de France and by MEPHISTO (ANR-24-CE23-7039-01) project of the French National Agency for Research (ANR).

References

[1] Mushroom. UCI Machine Learning Repository, 1981. https://doi.org/10.24432/C5959T.

[2] S. Alarie, C. Audet, A. E. Gheribi, M. Kokkolaras, and S. Le Digabel. Two decades of blackbox optimization applications. *EURO J. Comput. Optim.*, 9:100011, 2021.

[3] A.S. Bandeira, K. Scheinberg, and L.N. Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM J. Optim.*, 24(3):1238–1264, 2014.

[4] S. Bellavia, B. Gurioli, G. Morini, and Ph. L. Toint. Trust-region algorithms: Probabilistic complexity and intrinsic noise with applications to subsampling techniques. *EURO J. Comput. Optim.*, 10:100043, 2022.

[5] S. Bellavia, G. Gurioli, B. Morini, and Ph L Toint. Adaptive regularization for nonconvex optimization using inexact function values and randomly perturbed derivatives. *Journal of Complexity*, 68:101591, 2022.

[6] S. Bellavia, N. Krejić, B. Morini, and S. Rebegoldi. A stochastic first-order trust-region method with inexact restoration for finite-sum minimization. *Comput. Optim. Appl.*, 84(1):53–84, 2023.

[7] A. S. Berahas, O. Sohab, and L. N. Vicente. Full-low evaluation methods for derivative-free optimization. *Optim. Methods Softw.*, 38(2):386–411, 2023.

[8] E. H. Bergou, Y. Diouane, V. Kungurtsev, and C. W. Royer. A stochastic Levenberg–Marquardt method using random models with complexity results. *SIAM/ASA J. Uncert. Quant.*, 10(1):507–536, 2022.

[9] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Math. Program.*, 163:359–368, 2017.

[10] 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.

[11] R. Bollapragada, R. Byrd, and J. Nocedal. Adaptive sampling strategies for stochastic optimization. *SIAM Journal on Optimization*, 28(4):3312–3343, 2018.

[12] L. Bottou, F.E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Rev.*, 60(2):223–311, 2018.

[13] V. Braglia, A. Kopaničáková, and R. Krause. A multilevel approach to training. *ArXiv preprint 2006.15602*, 2020.

[14] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, 2000.

[15] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu. Sample size selection in optimization methods for machine learning. *Mathematical programming*, 134(1):127–155, 2012.

[16] H. Calandra, S. Gratton, E. Riccietti, and X. Vasseur. On high-order multilevel optimization strategies. *SIAM J. Optim.*, 31(1):307–330, 2021.

[17] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation complexity of algorithms for nonconvex optimization*. MOS-SIAM Series on Optimization, 2022.

[18] C. Cartis and R. Roberts. Scalable subspace methods for derivative-free nonlinear least-squares optimization. *Math. Program.*, 199:461–524, 2023.

[19] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *tist*, 2:27:1–27:27, 2011. Software available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html`.

[20] R. Chen, M. Menickelly, and K. Scheinberg. Stochastic optimization using a trust-region method and random models. *Math. Program.*, 169:447–487, 2018.

[21] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Adv. Neural Inf. Process. Syst.*, 27, 2014.

[22] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[23] G. Garrigos and R. M. Gower. Handbook of convergence theorems for (stochastic) gradient methods. *ArXiv preprint 2301.11235*, 2023.

[24] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations research*, 56(3):607–617, 2008.

[25] S. Gratton, S. Jerad, and Ph. L. Toint. Complexity of a class of first-order objective-function-free optimization algorithms. *Optim. Methods Softw.*, pages 1–31, 2024.

[26] S. Gratton, A. Kopaničáková, and P. L. Toint. Multilevel objective-function-free optimization with an application to neural networks training. *SIAM J. Optim.*, 33(4):2772–2800, 2023.

[27] S. Gratton, V. Mercier, E. Riccietti, and Ph. L. Toint. A block-coordinate approach of multi-level optimization with an application to physics-informed neural networks. *Comput. Optim. Appl.*, 2024.

[28] S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM J. Optim.*, 19:414–444, 2008.

[29] Y. Ha, S. Shashaani, and R. Pasupathy. Complexity of zeroth-and first-order stochastic trust-region algorithms. *SIAM J. Optim.*, 35(3):2098–2127, 2025.

[30] B. Jin, K. Scheinberg, and M. Xie. Sample complexity analysis for adaptive optimization algorithms with stochastic oracles. *Math. Program.*, 209(1):651–679, 2025.

[31] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Adv. in Neural Information Proc. Sys.*, 26, 2013.

[32] A. Kopaničáková and R. Krause. Globally Convergent Multilevel Training of Deep Residual Networks. *SIAM J. Sci. Comput.*, 0(0):S254–S280, 2022.

[33] G. Lauga, A. Repetti, E. Riccietti, N. Pustelnik, P. Gonçalves, and Y. Wiaux. A multilevel framework for accelerating usara in radio-interferometric imaging. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pages 2287–2291, 2024.

[34] G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves. IML FISTA: A multilevel framework for inexact and inertial forward-backward. application to image restoration. *SIAM J. Imaging Sc.*, 17(3):1347–1376, 2024.

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

[36] H. B. McMahan and M. Streeter. Adaptive bound optimization for online convex optimization. *ArXiv preprint 1002.4908*, 2010.

[37] S. G. Nash. A multigrid approach to discretized optimization problems. *Optim. Methods Softw.*, 14(1-2):99–116, 2000.

[38] P. Parpas. A multilevel proximal gradient algorithm for a class of composite optimization problems. *SIAM J. Sci. Comput.*, 39(5):S681–S701, 2017.

[39] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323. PMLR, 2016.

[40] F. Rinaldi, L.N. Vicente, and D. Zeffiro. Stochastic trust-region and direct-search methods: A weak tail bound condition and reduced sample sizing. *SIAM J. Optim.*, 34(2):2067–2092, 2024.

[41] C. W. Royer, O. Sohab, and L. N. Vicente. Full-low evaluation methods for bound and linearly constrained derivative-free optimization. *Comput. Optim. Appl.*, pages 1–37, 2024.

[42] K. Scheinberg and M. Xie. Stochastic adaptive regularization method with cubics: A high probability complexity bound. In *2023 Winter Simulation Conference (WSC)*, pages 3520–3531. IEEE, 2023.

[43] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *Journal of Machine Learning Research*, 21(219):1–30, 2020.

Algorithm 2 MU$^\ell$STREG for finite-sum minimization - MU$^\ell$STREG(ℓ,x_0,f^ℓ,ϵ^ℓ,λ^ℓ,$\mathcal{S}^\ell$)

Input: ℓ index of the level, $x_0 \in \mathbb{R}^n$ starting point, $f^\ell : \mathbb{R}^n \to \mathbb{R}$ objective function for level $\ell < \ell_{\max}$, $\epsilon^\ell > 0$ tolerance for the stopping criterion, $\lambda^\ell > 0$, $\mathcal{S}^\ell$ sample set for the current level.
Given $0 < \eta_1 \leq \eta_3 < 1$, $\eta_2 > 0$, $0 < \gamma_2 \leq \gamma_1 < 1 < \gamma_3$.

Set $k = 0$, $\lambda_0^\ell = \lambda^\ell$ and $\mathcal{S}_0^\ell = \mathcal{S}^\ell$.
while the stop criterion for level ℓ is not satisfied **do**
Sample set construction
  if $\ell = \ell_{\max}$ **then**
    Set $p_k := \min\{N, \max\{100k + n + 2, \lambda_k^2\}\}$ and build $\mathcal{S}_k^{\ell_{\max}} \subseteq \{1, \ldots, N\}$ drawing p_k indices randomly. Set $f_k^{\ell_{\max}} = f^{\mathcal{S}_k^{\ell_{\max}}}$ as in (30).
  else
    Set $f_k^\ell = f^\ell$.
  end if
Model choice
  if $\ell > 1$ **then**
    Choose to go to Step 13 or to Step 14.
  else
    Go to Step 13.
  end if
Regularized Taylor step
  Define $m_k^\ell(s) = f_k^\ell(x_k) + \nabla_x f_k^\ell(x_k)^T s$. Set $s_k = -\frac{\nabla_x f_k^\ell(x_k)}{\lambda_k^\ell \left\|\nabla_x f_k^\ell(x_k)\right\|}$. Go to Step 19.
Sub-sampled model
  Build $\mathcal{S}_k^{\ell-1} \subset \mathcal{S}_k^\ell$ randomly. Set $f^{\mathcal{S}_k^{\ell-1}}$ as in (30).
  Build the lower level approximation $f_k^{\ell-1}$ of f_k^ℓ from $f^{\mathcal{S}_k^{\ell-1}}$ (cf. discussion in section 4, below (32).). Compute the correction vector $v_k^{\ell-1}$ as in (32). Define the lower level model $\varphi_k^{\ell-1}(s)$ and its regularization $m_k^{R,\ell-1}(s)$ as

$$\varphi_k^{\ell-1}(s) = f_k^{\ell-1}(x_k + s) + (v_k^{\ell-1})^T s;$$
$$m_k^{R,\ell-1}(s) = \varphi_k^{\ell-1}(s) + \frac{1}{2}\lambda_k^\ell \left\|\nabla_x f_k^\ell(x_k)\right\| \|s\|^2.$$

  Recursive call
  Choose $\epsilon^{\ell-1}$ and call MU$^\ell$STREG($\ell - 1$,0,$m_k^{R,\ell-1}$,$\epsilon^{\ell-1}$,λ_k^ℓ,$\mathcal{S}_k^{\ell-1}$) to find an approximate solution s^* of the problem

$$\min_{s \in \mathbb{R}^n} m_k^{R,\ell-1}(s),$$

such that condition (4) is satisfied.
  Set $s_k = s^*$ and $m_k^\ell(s) = \varphi_k^{\ell-1}(s)$.
Step acceptance of trial point
  Compute $\rho_k^\ell := \frac{f_k^\ell(x_k) - f_k^\ell(x_k + s_k)}{m_k^\ell(0) - m_k^\ell(s_k)}$.
  if $\rho_k^\ell \geq \eta_1$ and $\left\|\nabla_x f_k^\ell(x_k)\right\| \geq \eta_2/\lambda_k^\ell$ **then**
    $x_{k+1} = x_k + s_k$
  else
    $x_{k+1} = x_k$
  end if
Regularization parameter update
  if $\rho_k^\ell \geq \eta_1$ and $\left\|\nabla_x f_k^\ell(x_k)\right\| \geq \eta_2/\lambda_k^\ell$ **then**

$$\lambda_{k+1}^\ell = \begin{cases} \max\{\lambda_{\min}, \gamma_2 \lambda_k^\ell\}, & \text{if } \rho_k^\ell \geq \eta_3, \\ \max\{\lambda_{\min}, \gamma_1 \lambda_k^\ell\}, & \text{if } \rho_k^\ell < \eta_3 \end{cases}$$

  else
    $\lambda_{k+1}^\ell = \gamma_3 \lambda_k^\ell$.
  end if
  $k = k + 1$
end while