A Universal Catalyst for First-Order Optimization

Hongzhou Lin, Julien Mairal, Zaid Harchaoui

A Universal Catalyst for First-Order Optimization

Hongzhou Lin[1], **Julien Mairal**[1] **and Zaid Harchaoui**[1,2]
[1]Inria [2]NYU
{hongzhou.lin,julien.mairal}@inria.fr
zaid.harchaoui@nyu.edu

Abstract

We introduce a generic scheme for accelerating first-order optimization methods in the sense of Nesterov, which builds upon a new analysis of the accelerated proximal point algorithm. Our approach consists of minimizing a convex objective by approximately solving a sequence of well-chosen auxiliary problems, leading to faster convergence. This strategy applies to a large class of algorithms, including gradient descent, block coordinate descent, SAG, SAGA, SDCA, SVRG, Finito/MISO, and their proximal variants. For all of these methods, we provide acceleration and explicit support for non-strongly convex objectives. In addition to theoretical speed-up, we also show that acceleration is useful in practice, especially for ill-conditioned problems where we measure significant improvements.

1 Introduction

A large number of machine learning and signal processing problems are formulated as the minimization of a composite objective function $F : \mathbb{R}^p \to \mathbb{R}$:

$$\min_{x \in \mathbb{R}^p} \Big\{ F(x) \triangleq f(x) + \psi(x) \Big\}, \tag{1}$$

where f is convex and has Lipschitz continuous derivatives with constant L and ψ is convex but may not be differentiable. The variable x represents model parameters and the role of f is to ensure that the estimated parameters fit some observed data. Specifically, f is often a large sum of functions

$$f(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(x), \tag{2}$$

and each term $f_i(x)$ measures the fit between x and a data point indexed by i. The function ψ in (1) acts as a regularizer; it is typically chosen to be the squared ℓ_2-norm, which is smooth, or to be a non-differentiable penalty such as the ℓ_1-norm or another sparsity-inducing norm [2]. Composite minimization also encompasses constrained minimization if we consider extended-valued indicator functions ψ that may take the value $+\infty$ outside of a convex set $\mathcal{C}$ and 0 inside (see [11]).

Our goal is to accelerate gradient-based or *first-order* methods that are designed to solve (1), with a particular focus on large sums of functions (2). By "accelerating", we mean generalizing a mechanism invented by Nesterov [17] that improves the convergence rate of the gradient descent algorithm. More precisely, when $\psi = 0$, gradient descent steps produce iterates $(x_k)_{k \geq 0}$ such that $F(x_k) - F^* = O(1/k)$, where F^* denotes the minimum value of F. Furthermore, when the objective F is strongly convex with constant μ, the rate of convergence becomes linear in $O((1 - \mu/L)^k)$. These rates were shown by Nesterov [16] to be suboptimal for the class of first-order methods, and instead optimal rates—$O(1/k^2)$ for the convex case and $O((1 - \sqrt{\mu/L})^k)$ for the μ-strongly convex one—could be obtained by taking gradient steps at well-chosen points. Later, this acceleration technique was extended to deal with non-differentiable regularization functions ψ [4, 19].

For modern machine learning problems involving a large sum of n functions, a recent effort has been devoted to developing fast *incremental* algorithms [6, 7, 14, 24, 25, 27] that can exploit the particular

structure of (2). Unlike full gradient approaches which require computing and averaging n gradients $\nabla f(x) = (1/n)\sum_{i=1}^n \nabla f_i(x)$ at every iteration, incremental techniques have a cost per-iteration that is independent of n. The price to pay is the need to store a moderate amount of information regarding past iterates, but the benefit is significant in terms of computational complexity.

Main contributions. Our main achievement is a *generic acceleration scheme* that applies to a large class of optimization methods. By analogy with substances that increase chemical reaction rates, we call our approach a "catalyst". A method may be accelerated if it has linear convergence rate for strongly convex problems. This is the case for full gradient [4, 19] and block coordinate descent methods [18, 21], which already have well-known accelerated variants. More importantly, it also applies to incremental algorithms such as SAG [24], SAGA [6], Finito/MISO [7, 14], SDCA [25], and SVRG [27]. Whether or not these methods could be accelerated was an important open question. It was only known to be the case for dual coordinate ascent approaches such as SDCA [26] or SDPC [28] for strongly convex objectives. Our work provides a universal positive answer regardless of the strong convexity of the objective, which brings us to our second achievement.

Some approaches such as Finito/MISO, SDCA, or SVRG are only defined for strongly convex objectives. A classical trick to apply them to general convex functions is to add a small regularization $\varepsilon\|x\|^2$ [25]. The drawback of this strategy is that it requires choosing in advance the parameter ε, which is related to the target accuracy. A consequence of our work is to automatically provide a *direct support for non-strongly convex objectives*, thus removing the need of selecting ε beforehand.

Other contribution: Proximal MISO. The approach Finito/MISO, which was proposed in [7] and [14], is an incremental technique for solving smooth unconstrained μ-strongly convex problems when n is larger than a constant $\beta L/\mu$ (with $\beta = 2$ in [14]). In addition to providing acceleration and support for non-strongly convex objectives, we also make the following specific contributions:

- we extend the method and its convergence proof to deal with the composite problem (1);
- we fix the method to remove the "big data condition" $n \geq \beta L/\mu$.

The resulting algorithm can be interpreted as a variant of proximal SDCA [25] with a different step size and a more practical optimality certificate—that is, checking the optimality condition does not require evaluating a dual objective. Our construction is indeed purely *primal*. Neither our proof of convergence nor the algorithm use duality, while SDCA is originally a dual ascent technique.

Related work. The catalyst acceleration can be interpreted as a variant of the proximal point algorithm [3, 9], which is a central concept in convex optimization, underlying augmented Lagrangian approaches, and composite minimization schemes [5, 20]. The proximal point algorithm consists of solving (1) by minimizing a sequence of auxiliary problems involving a quadratic regularization term. In general, these auxiliary problems cannot be solved with perfect accuracy, and several notations of inexactness were proposed, including [9, 10, 22]. The catalyst approach hinges upon (i) an acceleration technique for the proximal point algorithm originally introduced in the pioneer work [9]; (ii) a more practical inexactness criterion than those proposed in the past.[1] As a result, we are able to control the rate of convergence for approximately solving the auxiliary problems with an optimization method $\mathcal{M}$. In turn, we are also able to obtain the computational complexity of the global procedure for solving (1), which was not possible with previous analysis [9, 10, 22]. When instantiated in different first-order optimization settings, our analysis yields systematic acceleration.

Beyond [9], several works have inspired this paper. In particular, accelerated SDCA [26] is an instance of an inexact accelerated proximal point algorithm, even though this was not explicitly stated in [26]. Their proof of convergence relies on different tools than ours. Specifically, we use the concept of *estimate sequence* from Nesterov [17], whereas the direct proof of [26], in the context of SDCA, does not extend to non-strongly convex objectives. Nevertheless, part of their analysis proves to be helpful to obtain our main results. Another useful methodological contribution was the convergence analysis of inexact proximal gradient methods of [23]. Finally, similar ideas appear in the independent work [8]. Their results overlap in part with ours, but both papers adopt different directions. Our analysis is for instance more general and provides support for non-strongly convex objectives. Another independent work with related results is [13], which introduce an accelerated method for the minimization of finite sums, which is not based on the proximal point algorithm.

[1] Note that our inexact criterion was also studied, among others, in [22], but the analysis of [22] led to the conjecture that this criterion was too weak to warrant acceleration. Our analysis refutes this conjecture.

2 The Catalyst Acceleration

We present here our generic acceleration scheme, which can operate on any first-order or gradient-based optimization algorithm with linear convergence rate for strongly convex objectives.

Linear convergence and acceleration. Consider the problem (1) with a μ-strongly convex function F, where the strong convexity is defined with respect to the ℓ_2-norm. A minimization algorithm $\mathcal{M}$, generating the sequence of iterates $(x_k)_{k\geq 0}$, has a *linear convergence rate* if there exists $\tau_{\mathcal{M},F}$ in $(0,1)$ and a constant $C_{\mathcal{M},F}$ in $\mathbb{R}$ such that

$$F(x_k) - F^* \leq C_{\mathcal{M},F}(1-\tau_{\mathcal{M},F})^k, \tag{3}$$

where F^* denotes the minimum value of F. The quantity $\tau_{\mathcal{M},F}$ controls the convergence rate: the larger is $\tau_{\mathcal{M},F}$, the faster is convergence to F^*. However, for a given algorithm $\mathcal{M}$, the quantity $\tau_{\mathcal{M},F}$ depends usually on the ratio L/μ, which is often called the *condition number* of F.

The catalyst acceleration is a general approach that allows to wrap algorithm $\mathcal{M}$ into an accelerated algorithm $\mathcal{A}$, which enjoys a faster linear convergence rate, with $\tau_{\mathcal{A},F} \geq \tau_{\mathcal{M},F}$. As we will also see, the catalyst acceleration may also be useful when F is not strongly convex—that is, when $\mu = 0$. In that case, we may even consider a method $\mathcal{M}$ that requires strong convexity to operate, and obtain an accelerated algorithm $\mathcal{A}$ that can minimize F with near-optimal convergence rate $\tilde{O}(1/k^2)$.[2]

Our approach can accelerate a wide range of first-order optimization algorithms, starting from classical gradient descent. It also applies to randomized algorithms such as SAG, SAGA, SDCA, SVRG and Finito/MISO, whose rates of convergence are given in expectation. Such methods should be contrasted with stochastic gradient methods [15, 12], which minimize a different non-deterministic function. Acceleration of stochastic gradient methods is beyond the scope of this work.

Catalyst action. We now highlight the mechanics of the catalyst algorithm, which is presented in Algorithm 1. It consists of replacing, at iteration k, the original objective function F by an auxiliary objective G_k, close to F up to a quadratic term:

$$G_k(x) \triangleq F(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2, \tag{4}$$

where κ will be specified later and y_k is obtained by an extrapolation step described in (6). Then, at iteration k, the accelerated algorithm $\mathcal{A}$ minimizes G_k up to accuracy ε_k.

Substituting (4) to (1) has two consequences. On the one hand, minimizing (4) only provides an approximation of the solution of (1), unless $\kappa = 0$; on the other hand, the auxiliary objective G_k enjoys a better condition number than the original objective F, which makes it easier to minimize. For instance, when $\mathcal{M}$ is the regular gradient descent algorithm with $\psi = 0$, $\mathcal{M}$ has the rate of convergence (3) for minimizing F with $\tau_{\mathcal{M},F} = \mu/L$. However, owing to the additional quadratic term, G_k can be minimized by $\mathcal{M}$ with the rate (3) where $\tau_{\mathcal{M},G_k} = (\mu+\kappa)/(L+\kappa) > \tau_{\mathcal{M},F}$. In practice, there exists an "optimal" choice for κ, which controls the time required by $\mathcal{M}$ for solving the auxiliary problems (4), and the quality of approximation of F by the functions G_k. This choice will be driven by the convergence analysis in Sec. 3.1-3.3; see also Sec. C for special cases.

Acceleration via extrapolation and inexact minimization. Similar to the classical gradient descent scheme of Nesterov [17], Algorithm 1 involves an extrapolation step (6). As a consequence, the solution of the auxiliary problem (5) at iteration $k+1$ is driven towards the extrapolated variable y_k. As shown in [9], this step is in fact sufficient to reduce the number of iterations of Algorithm 1 to solve (1) when $\varepsilon_k = 0$—that is, for running the *exact* accelerated proximal point algorithm.

Nevertheless, to control the total computational complexity of an accelerated algorithm $\mathcal{A}$, it is necessary to take into account the complexity of solving the auxiliary problems (5) using $\mathcal{M}$. This is where our approach differs from the classical proximal point algorithm of [9]. Essentially, both algorithms are the same, but we use the weaker inexactness criterion $G_k(x_k) - G_k^* \leq \varepsilon_k$, where the sequence $(\varepsilon_k)_{k\geq 0}$ is fixed beforehand, and only depends on the initial point. This subtle difference has important consequences: (i) in practice, this condition can often be checked by computing duality gaps; (ii) in theory, the methods $\mathcal{M}$ we consider have linear convergence rates, which allows us to control the complexity of step (5), and then to provide the computational complexity of $\mathcal{A}$.

[2] In this paper, we use the notation $O(.)$ to hide constants. The notation $\tilde{O}(.)$ also hides logarithmic factors.

Algorithm 1 Catalyst

input initial estimate $x_0 \in \mathbb{R}^p$, parameters κ and α_0, sequence $(\varepsilon_k)_{k\geq 0}$, optimization method $\mathcal{M}$;

1: Initialize $q = \mu/(\mu+\kappa)$ and $y_0 = x_0$;
2: **while** the desired stopping criterion is not satisfied **do**
3: Find an approximate solution of the following problem using $\mathcal{M}$

$$x_k \approx \arg\min_{x\in\mathbb{R}^p} \Big\{ G_k(x) \triangleq F(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \Big\} \quad \text{such that} \quad G_k(x_k) - G_k^* \leq \varepsilon_k. \tag{5}$$

4: Compute $\alpha_k \in (0,1)$ from equation $\alpha_k^2 = (1-\alpha_k)\alpha_{k-1}^2 + q\alpha_k$;
5: Compute

$$y_k = x_k + \beta_k(x_k - x_{k-1}) \quad \text{with} \quad \beta_k = \frac{\alpha_{k-1}(1-\alpha_{k-1})}{\alpha_{k-1}^2 + \alpha_k}. \tag{6}$$

6: **end while**

output x_k (final estimate).

3 Convergence Analysis

In this section, we present the theoretical properties of Algorithm 1, for optimization methods $\mathcal{M}$ with deterministic convergence rates of the form (3). When the rate is given as an expectation, a simple extension of our analysis described in Section 4 is needed. For space limitation reasons, we shall sketch the proof mechanics here, and defer the full proofs to Appendix B.

3.1 Analysis for μ-Strongly Convex Objective Functions

We first analyze the convergence rate of Algorithm 1 for solving problem 1, regardless of the complexity required to solve the subproblems (5). We start with the μ-strongly convex case.

Theorem 3.1 (Convergence of Algorithm 1, μ-Strongly Convex Case).
Choose $\alpha_0 = \sqrt{q}$ with $q = \mu/(\mu+\kappa)$ and

$$\varepsilon_k = \frac{2}{9}(F(x_0) - F^*)(1-\rho)^k \quad \textit{with} \quad \rho < \sqrt{q}.$$

Then, Algorithm 1 generates iterates $(x_k)_{k\geq 0}$ such that

$$F(x_k) - F^* \leq C(1-\rho)^{k+1}(F(x_0) - F^*) \quad \textit{with} \quad C = \frac{8}{(\sqrt{q}-\rho)^2}. \tag{7}$$

This theorem characterizes the linear convergence rate of Algorithm 1. It is worth noting that the choice of ρ is left to the discretion of the user, but it can safely be set to $\rho = 0.9\sqrt{q}$ in practice. The choice $\alpha_0 = \sqrt{q}$ was made for convenience purposes since it leads to a simplified analysis, but larger values are also acceptable, both from theoretical and practical point of views. Following an advice from Nesterov[17, page 81] originally dedicated to his classical gradient descent algorithm, we may for instance recommend choosing α_0 such that $\alpha_0^2 + (1-q)\alpha_0 - 1 = 0$.

The choice of the sequence $(\varepsilon_k)_{k\geq 0}$ is also subject to discussion since the quantity $F(x_0) - F^*$ is unknown beforehand. Nevertheless, an upper bound may be used instead, which will only affects the corresponding constant in (7). Such upper bounds can typically be obtained by computing a duality gap at x_0, or by using additional knowledge about the objective. For instance, when F is non-negative, we may simply choose $\varepsilon_k = (2/9)F(x_0)(1-\rho)^k$.

The proof of convergence uses the concept of estimate sequence invented by Nesterov [17], and introduces an extension to deal with the errors $(\varepsilon_k)_{k\geq 0}$. To control the accumulation of errors, we borrow the methodology of [23] for inexact proximal gradient algorithms. Our construction yields a convergence result that encompasses both strongly convex and non-strongly convex cases. Note that estimate sequences were also used in [9], but, as noted by [22], the proof of [9] only applies when using an extrapolation step (6) that involves the true minimizer of (5), which is unknown in practice. To obtain a rigorous convergence result like (7), a different approach was needed.

Theorem 3.1 is important, but it does not provide yet the global computational complexity of the full algorithm, which includes the number of iterations performed by $\mathcal{M}$ for approximately solving the auxiliary problems (5). The next proposition characterizes the complexity of this inner-loop.

Proposition 3.2 (Inner-Loop Complexity, μ-Strongly Convex Case).
Under the assumptions of Theorem 3.1, let us consider a method $\mathcal{M}$ generating iterates $(z_t)_{t\geq 0}$ for minimizing the function G_k with linear convergence rate of the form

$$G_k(z_t) - G_k^* \leq A(1-\tau_{\mathcal{M}})^t (G_k(z_0) - G_k^*). \tag{8}$$

When $z_0 = x_{k-1}$, the precision ε_k is reached with a number of iterations $T_{\mathcal{M}} = \tilde{O}(1/\tau_{\mathcal{M}})$, where the notation $\tilde{O}$ hides some universal constants and some logarithmic dependencies in μ and κ.

This proposition is generic since the assumption (8) is relatively standard for gradient-based methods [17]. It may now be used to obtain the global rate of convergence of an accelerated algorithm. By calling F_s the objective function value obtained after performing $s = kT_{\mathcal{M}}$ iterations of the method $\mathcal{M}$, the true convergence rate of the accelerated algorithm $\mathcal{A}$ is

$$F_s - F^* = F\left(x_{\frac{s}{T_{\mathcal{M}}}}\right) - F^* \leq C(1-\rho)^{\frac{s}{T_{\mathcal{M}}}} (F(x_0) - F^*) \leq C\left(1 - \frac{\rho}{T_{\mathcal{M}}}\right)^s (F(x_0) - F^*). \tag{9}$$

As a result, algorithm $\mathcal{A}$ has a global linear rate of convergence with parameter

$$\tau_{\mathcal{A},F} = \rho/T_{\mathcal{M}} = \tilde{O}(\tau_{\mathcal{M}}\sqrt{\mu}/\sqrt{\mu+\kappa}),$$

where $\tau_{\mathcal{M}}$ typically depends on κ (the greater, the faster is $\mathcal{M}$). Consequently, κ will be chosen to maximize the ratio $\tau_{\mathcal{M}}/\sqrt{\mu+\kappa}$. Note that for other algorithms $\mathcal{M}$ that do not satisfy (8), additional analysis and possibly a different initialization z_0 may be necessary (see Appendix D for example).

3.2 Convergence Analysis for Convex but Non-Strongly Convex Objective Functions

We now state the convergence rate when the objective is *not strongly convex*, that is when $\mu = 0$.

Theorem 3.3 (Convergence of Algorithm 1, Convex, but Non-Strongly Convex Case).
When $\mu = 0$, choose $\alpha_0 = (\sqrt{5}-1)/2$ and

$$\varepsilon_k = \frac{2(F(x_0) - F^*)}{9(k+2)^{4+\eta}} \quad \text{with } \eta > 0. \tag{10}$$

Then, Algorithm 1 generates iterates $(x_k)_{k\geq 0}$ such that

$$F(x_k) - F^* \leq \frac{8}{(k+2)^2}\left(\left(1+\frac{2}{\eta}\right)^2 (F(x_0) - F^*) + \frac{\kappa}{2}\|x_0 - x^*\|^2\right). \tag{11}$$

This theorem is the counter-part of Theorem 3.1 when $\mu = 0$. The choice of η is left to the discretion of the user; it empirically seem to have very low influence on the global convergence speed, as long as it is chosen small enough (e.g., we use $\eta = 0.1$ in practice). It shows that Algorithm 1 achieves the optimal rate of convergence of first-order methods, but *it does not take into account the complexity of solving the subproblems (5)*. Therefore, we need the following proposition:

Proposition 3.4 (Inner-Loop Complexity, Non-Strongly Convex Case).
Assume that F has bounded level sets. Under the assumptions of Theorem 3.3, let us consider a method $\mathcal{M}$ generating iterates $(z_t)_{t\geq 0}$ for minimizing the function G_k with linear convergence rate of the form (8). Then, there exists $T_{\mathcal{M}} = \tilde{O}(1/\tau_{\mathcal{M}})$, such that for any $k \geq 1$, solving G_k with initial point x_{k-1} requires at most $T_{\mathcal{M}}\log(k+2)$ iterations of $\mathcal{M}$.

We can now draw up the global complexity of an accelerated algorithm $\mathcal{A}$ when $\mathcal{M}$ has a linear convergence rate (8) for κ-strongly convex objectives. To produce x_k, $\mathcal{M}$ is called at most $kT_{\mathcal{M}}\log(k+2)$ times. Using the global iteration counter $s = kT_{\mathcal{M}}\log(k+2)$, we get

$$F_s - F^* \leq \frac{8T_{\mathcal{M}}^2 \log^2(s)}{s^2}\left(\left(1+\frac{2}{\eta}\right)^2 (F(x_0) - F^*) + \frac{\kappa}{2}\|x_0 - x^*\|^2\right). \tag{12}$$

If $\mathcal{M}$ is a first-order method, this rate is *near-optimal*, up to a logarithmic factor, when compared to the optimal rate $O(1/s^2)$, which may be the price to pay for using a generic acceleration scheme.

4 Acceleration in Practice

We show here how to accelerate existing algorithms $\mathcal{M}$ and compare the convergence rates obtained before and after catalyst acceleration. For all the algorithms we consider, we study rates of convergence in terms of *total number of iterations* (in expectation, when necessary) to reach accuracy ε. We first show how to accelerate full gradient and randomized coordinate descent algorithms [21]. Then, we discuss other approaches such as SAG [24], SAGA [6], or SVRG [27]. Finally, we present a new proximal version of the incremental gradient approaches Finito/MISO [7, 14], along with its accelerated version. Table 4.1 summarizes the acceleration obtained for the algorithms considered.

Deriving the global rate of convergence. The convergence rate of an accelerated algorithm $\mathcal{A}$ is driven by the parameter κ. In the strongly convex case, the best choice is the one that maximizes the ratio $\tau_{\mathcal{M},G_k}/\sqrt{\mu+\kappa}$. As discussed in Appendix C, this rule also holds when (8) is given in expectation and in many cases where the constant $\mathcal{C}_{\mathcal{M},G_k}$ is different than $A(G_k(z_0)-G_k^*)$ from (8). When $\mu = 0$, the choice of $\kappa > 0$ only affects the complexity by a multiplicative constant. A rule of thumb is to maximize the ratio $\tau_{\mathcal{M},G_k}/\sqrt{L+\kappa}$ (see Appendix C for more details).

After choosing κ, the global iteration-complexity is given by Comp $\leq k_{\text{in}}\, k_{\text{out}}$, where k_{in} is an upper-bound on the number of iterations performed by $\mathcal{M}$ per inner-loop, and k_{out} is the upper-bound on the number of outer-loop iterations, following from Theorems 3.1-3.3. Note that for simplicity, we always consider that $L \gg \mu$ such that we may write $L - \mu$ simply as "L" in the convergence rates.

4.1 Acceleration of Existing Algorithms

Composite minimization. Most of the algorithms we consider here, namely the proximal gradient method [4, 19], SAGA [6], (Prox)-SVRG [27], can handle composite objectives with a regularization penalty ψ that admits a proximal operator prox_ψ, defined for any z as

$$\text{prox}_\psi(z) \triangleq \arg\min_{y\in\mathbb{R}^p} \left\{ \psi(y) + \frac{1}{2}\|y-z\|^2 \right\} .$$

Table 4.1 presents convergence rates that are valid for proximal and non-proximal settings, since most methods we consider are able to deal with such non-differentiable penalties. The exception is SAG [24], for which proximal variants are not analyzed. The incremental method Finito/MISO has also been limited to non-proximal settings so far. In Section 4.2, we actually introduce the extension of MISO to composite minimization, and establish its theoretical convergence rates.

Full gradient method. A first illustration is the algorithm obtained when accelerating the regular "full" gradient descent (FG), and how it contrasts with Nesterov's accelerated variant (AFG). Here, the optimal choice for κ is $L - 2\mu$. In the strongly convex case, we get an accelerated rate of convergence in $\tilde{O}(n\sqrt{L/\mu}\log(1/\varepsilon))$, which is the same as AFG up to logarithmic terms. A similar result can also be obtained for randomized coordinate descent methods [21].

Randomized incremental gradient. We now consider randomized incremental gradient methods, resp. SAG [24] and SAGA [6]. When $\mu > 0$, we focus on the "ill-conditioned" setting $n \leq L/\mu$, where these methods have the complexity $O((L/\mu)\log(1/\varepsilon))$. Otherwise, their complexity becomes $O(n\log(1/\varepsilon))$, which is independent of the condition number and seems theoretically optimal [1].

For these methods, the best choice for κ has the form $\kappa = a(L-\mu)/(n+b) - \mu$, with $(a,b) = (2,-2)$ for SAG, $(a,b) = (1/2, 1/2)$ for SAGA. A similar formula, with a constant L' in place of L, holds for SVRG; we omit it here for brevity. SDCA [26] and Finito/MISO [7, 14] are actually related to incremental gradient methods, and the choice for κ has a similar form with $(a,b) = (1,1)$.

4.2 Proximal MISO and its Acceleration

Finito/MISO was proposed in [7] and [14] for solving the problem (1) when $\psi = 0$ and when f is a sum of n μ-strongly convex functions f_i as in (2), which are also differentiable with L-Lipschitz derivatives. The algorithm maintains a list of quadratic lower bounds—say $(d_i^k)_{i=1}^n$ at iteration k—of the functions f_i and randomly updates one of them at each iteration by using strong-convexity

	Comp. $\mu > 0$	Comp. $\mu = 0$	Catalyst $\mu > 0$	Catalyst $\mu = 0$
FG	$O\left(n\left(\frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(n\sqrt{\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$\tilde{O}\left(n\frac{L}{\sqrt{\varepsilon}}\right)$
SAG [24]	$O\left(\frac{L}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$		$\tilde{O}\left(\sqrt{\frac{nL}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	
SAGA [6]				
Finito/MISO-Prox		not avail.		
SDCA [25]				
SVRG [27]	$O\left(\frac{L'}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$		$\tilde{O}\left(\sqrt{\frac{nL'}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	
Acc-FG [19]	$O\left(n\sqrt{\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\sqrt{\varepsilon}}\right)$	no acceleration	
Acc-SDCA [26]	$\tilde{O}\left(\sqrt{\frac{nL}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	not avail.		

Table 1: Comparison of rates of convergence, before and after the catalyst acceleration, resp. in the strongly-convex and non strongly-convex cases. **To simplify, we only present the case where $n \leq L/\mu$ when $\mu > 0$.** For all incremental algorithms, there is indeed no acceleration otherwise. The quantity L' for SVRG is the average Lipschitz constant of the functions f_i (see [27]).

inequalities. The current iterate x_k is then obtained by minimizing the lower-bound of the objective

$$x_k = \operatorname*{arg\,min}_{x \in \mathbb{R}^p} \left\{ D_k(x) = \frac{1}{n}\sum_{i=1}^{n} d_i^k(x) \right\}. \tag{13}$$

Interestingly, since D_k is a lower-bound of F we also have $D_k(x_k) \leq F^*$, and thus the quantity $F(x_k) - D_k(x_k)$ can be used as an optimality certificate that upper-bounds $F(x_k) - F^*$. Furthermore, this certificate was shown to converge to zero with a rate similar to SAG/SDCA/SVRG/SAGA under the condition $n \geq 2L/\mu$. In this section, we show how to remove this condition and how to provide support to non-differentiable functions ψ whose proximal operator can be easily computed. We shall briefly sketch the main ideas, and we refer to Appendix D for a thorough presentation.

The first idea to deal with a nonsmooth regularizer ψ is to change the definition of D_k:

$$D_k(x) = \frac{1}{n}\sum_{i=1}^{n} d_i^k(x) + \psi(x),$$

which was also proposed in [7] without a convergence proof. Then, because the d_i^k's are quadratic functions, the minimizer x_k of D_k can be obtained by computing the proximal operator of ψ at a particular point. The second idea to remove the condition $n \geq 2L/\mu$ is to modify the update of the lower bounds d_i^k. Assume that index i_k is selected among $\{1, \dots, n\}$ at iteration k, then

$$d_i^k(x) = \begin{cases} (1-\delta)d_i^{k-1}(x) + \delta(f_i(x_{k-1}) + \langle \nabla f_i(x_{k-1}), x - x_{k-1}\rangle + \frac{\mu}{2}\|x - x_{k-1}\|^2) & \text{if } \ i = i_k \\ d_i^{k-1}(x) & \text{otherwise} \end{cases}$$

Whereas the original Finito/MISO uses $\delta = 1$, our new variant uses $\delta = \min(1, \mu n/2(L-\mu))$. The resulting algorithm turns out to be very close to variant "5" of proximal SDCA [25], which corresponds to using a different value for δ. The main difference between SDCA and MISO-Prox is that the latter does not use duality. It also provides a different (simpler) optimality certificate $F(x_k) - D_k(x_k)$, which is guaranteed to converge linearly, as stated in the next theorem.

Theorem 4.1 (Convergence of MISO-Prox).
Let $(x_k)_{k\geq 0}$ be obtained by MISO-Prox, then

$$\mathbb{E}[F(x_k)] - F^* \leq \frac{1}{\tau}(1-\tau)^{k+1}\left(F(x_0) - D_0(x_0)\right) \quad \textit{with} \quad \tau \geq \min\left\{\frac{\mu}{4L}, \frac{1}{2n}\right\}. \tag{14}$$

Furthermore, we also have fast convergence of the certificate

$$\mathbb{E}[F(x_k) - D_k(x_k)] \leq \frac{1}{\tau}(1-\tau)^{k}\left(F^* - D_0(x_0)\right).$$

The proof of convergence is given in Appendix D. Finally, we conclude this section by noting that MISO-Prox enjoys the catalyst acceleration, leading to the iteration-complexity presented in Table 4.1. Since the convergence rate (14) does not have exactly the same form as (8), Propositions 3.2 and 3.4 cannot be used and additional analysis, given in Appendix D, is needed. Practical forms of the algorithm are also presented there, along with discussions on how to initialize it.

5 Experiments

We evaluate the Catalyst acceleration on three methods that have never been accelerated in the past: SAG [24], SAGA [6], and MISO-Prox. We focus on ℓ_2-regularized logistic regression, where the regularization parameter μ yields a lower bound on the strong convexity parameter of the problem.

We use three datasets used in [14], namely real-sim, rcv1, and ocr, which are relatively large, with up to $n = 2\,500\,000$ points for ocr and $p = 47\,152$ variables for rcv1. We consider three regimes: $\mu = 0$ (no regularization), $\mu/L = 0.001/n$ and $\mu/L = 0.1/n$, which leads significantly larger condition numbers than those used in other studies ($\mu/L \approx 1/n$ in [14, 24]). We compare MISO, SAG, and SAGA with their default parameters, which are recommended by their theoretical analysis (step-sizes $1/L$ for SAG and $1/3L$ for SAGA), and study several accelerated variants. The values of κ and ρ and the sequences $(\varepsilon_k)_{k\geq 0}$ are those suggested in the previous sections, with $\eta = 0.1$ in (10). Other implementation details are presented in Appendix E.

The restarting strategy for $\mathcal{M}$ is key to achieve acceleration in practice. All of the methods we compare store n gradients evaluated at previous iterates of the algorithm. We always use the gradients from the previous run of $\mathcal{M}$ to initialize a new one. We detail in Appendix E the initialization for each method. Finally, we evaluated a heuristic that constrain $\mathcal{M}$ to always perform at most n iterations (one pass over the data); we call this variant AMISO2 for MISO whereas AMISO1 refers to the regular "vanilla" accelerated variant, and we also use this heuristic to accelerate SAG.

The results are reported in Table 1. We always obtain a huge speed-up for MISO, which suffers from numerical stability issues when the condition number is very large (for instance, $\mu/L = 10^{-3}/n = 4.10^{-10}$ for ocr). Here, not only does the catalyst algorithm accelerate MISO, but it also stabilizes it. Whereas MISO is slower than SAG and SAGA in this "small μ" regime, AMISO2 is almost systematically the best performer. We are also able to accelerate SAG and SAGA in general, even though the improvement is less significant than for MISO. In particular, SAGA without acceleration proves to be the best method on ocr. One reason may be its ability to adapt to the unknown strong convexity parameter $\mu' \geq \mu$ of the objective near the solution. When $\mu'/L \geq 1/n$, we indeed obtain a regime where acceleration does not occur (see Sec. 4). Therefore, this experiment suggests that adaptivity to unknown strong convexity is of high interest for incremental optimization.

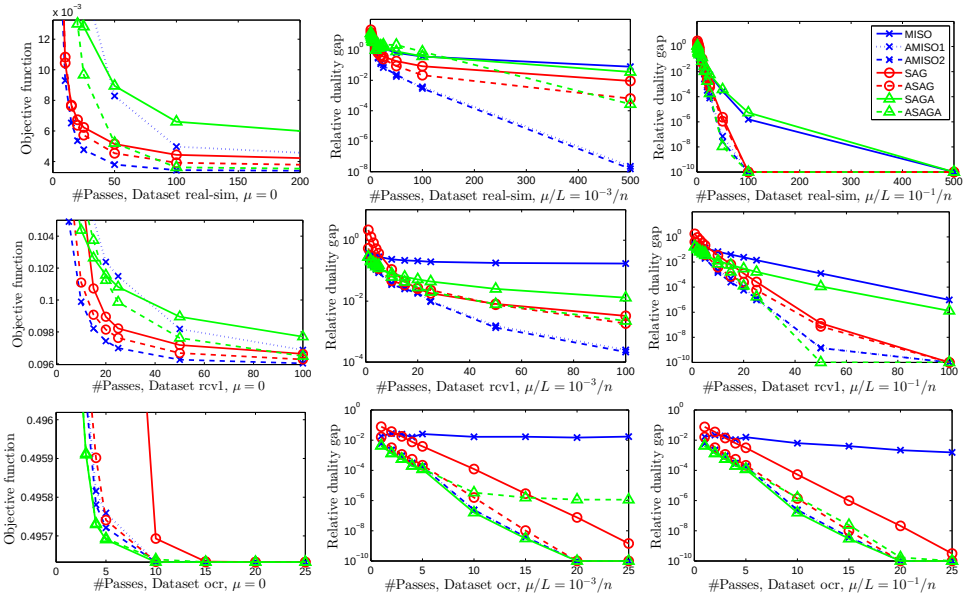

Figure 1: Objective function value (or duality gap) for different number of passes performed over each dataset. The legend for all curves is on the top right. AMISO, ASAGA, ASAG refer to the accelerated variants of MISO, SAGA, and SAG, respectively.

Acknowledgments

This work was supported by ANR (MACARON ANR-14-CE23-0003-01), MSR-Inria joint centre, CNRS-Mastodons program (Titan), and NYU Moore-Sloan Data Science Environment.

References

[1] A. Agarwal and L. Bottou. A lower bound for the optimization of finite sums. In *Proc. International Conference on Machine Learning (ICML)*, 2015.

[2] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[3] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[5] D. P. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015.

[6] A. J. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Adv. Neural Information Processing Systems (NIPS)*, 2014.

[7] A. J. Defazio, T. S. Caetano, and J. Domke. Finito: A faster, permutable incremental gradient method for big data problems. In *Proc. International Conference on Machine Learning (ICML)*, 2014.

[8] R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Un-regularizing: approximate proximal point algorithms for empirical risk minimization. In *Proc. International Conference on Machine Learning (ICML)*, 2015.

[9] O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[10] B. He and X. Yuan. An accelerated inexact proximal point algorithm for convex minimization. *Journal of Optimization Theory and Applications*, 154(2):536–548, 2012.

[11] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer, 1996.

[12] A. Juditsky and A. Nemirovski. First order methods for nonsmooth convex large-scale optimization. *Optimization for Machine Learning, MIT Press*, 2012.

[13] G. Lan. An optimal randomized incremental gradient method. *arXiv:1507.02000*, 2015.

[14] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[15] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

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

[17] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2004.

[18] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[19] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[20] N. Parikh and S.P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[21] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[22] S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

[23] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Adv. Neural Information Processing Systems (NIPS)*, 2011.

[24] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2013.

[25] S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *arXiv:1211.2717*, 2012.

[26] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 2015.

[27] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[28] Y. Zhang and L. Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proc. International Conference on Machine Learning (ICML)*, 2015.

In this appendix, Section A is devoted to the construction of an object called *estimate sequence*, originally introduced by Nesterov (see [17]), and introduce extensions to deal with inexact minimization. This section contains a generic convergence result that will be used to prove the main theorems and propositions of the paper in Section B. Then, Section C is devoted to the computation of global convergence rates of accelerated algorithms, Section D presents in details the proximal MISO algorithm, and Section E gives some implementation details of the experiments.

A Construction of the Approximate Estimate Sequence

The estimate sequence is a generic tool introduced by Nesterov for proving the convergence of accelerated gradient-based algorithms. We start by recalling the definition given in [17].

Definition A.1 (Estimate Sequence [17]).
A pair of sequences $(\varphi_k)_{k\geq 0}$ and $(\lambda_k)_{k\geq 0}$, with $\lambda_k \geq 0$ and $\varphi_k : \mathbb{R}^p \to \mathbb{R}$, is called an *estimate sequence* of function F if

$$\lambda_k \to 0,$$

and for any x in $\mathbb{R}^p$ and all $k \geq 0$, we have

$$\varphi_k(x) \leq (1-\lambda_k)F(x) + \lambda_k \varphi_0(x).$$

Estimate sequences are used for proving convergence rates thanks to the following lemma

Lemma A.2 (Lemma 2.2.1 from [17]).
If for some sequence $(x_k)_{k\geq 0}$ we have

$$F(x_k) \leq \varphi_k^* \triangleq \min_{x\in\mathbb{R}^p} \varphi_k(x),$$

for an estimate sequence $(\varphi_k)_{k\geq 0}$ of F, then

$$F(x_k) - F^* \leq \lambda_k(\varphi_0(x^*) - F^*),$$

where x^* is a minimizer of F.

The rate of convergence of $F(x_k)$ is thus directly related to the convergence rate of λ_k. Constructing estimate sequences is thus appealing, even though finding the most appropriate one is not trivial for the catalyst algorithm because of the approximate minimization of G_k in (5). In a nutshell, the main steps of our convergence analysis are

1. define an "approximate" estimate sequence for F corresponding to Algorithm 1—that is, finding a function φ that almost satisfies Definition A.1 up to the approximation errors ε_k made in (5) when minimizing G_k, and control the way these errors sum up together.
2. extend Lemma A.2 to deal with the approximation errors ε_k to derive a generic convergence rate for the sequence $(x_k)_{k\geq 0}$.

This is also the strategy proposed by Güler in [9] for his inexact accelerated proximal point algorithm, which essentially differs from ours in its stopping criterion. The estimate sequence we choose is also different and leads to a more rigorous convergence proof. Specifically, we prove in this section the following theorem:

Theorem A.3 (Convergence Result Derived from an Approximate Estimate Sequence).
Let us denote

$$\lambda_k = \prod_{i=0}^{k-1}(1-\alpha_i), \tag{15}$$

where the α_i's are defined in Algorithm 1. Then, the sequence $(x_k)_{k\geq 0}$ satisfies

$$F(x_k) - F^* \leq \lambda_k \left(\sqrt{S_k} + 2\sum_{i=1}^{k} \sqrt{\frac{\varepsilon_i}{\lambda_i}} \right)^2, \tag{16}$$

where F^* is the minimum value of F and

$$S_k = F(x_0) - F^* + \frac{\gamma_0}{2}\|x_0 - x^*\|^2 + \sum_{i=1}^{k} \frac{\varepsilon_i}{\lambda_i} \quad \textit{where} \quad \gamma_0 = \frac{\alpha_0\left((\kappa+\mu)\alpha_0 - \mu\right)}{1-\alpha_0}, \tag{17}$$

where x^* is a minimizer of F.

Then, the theorem will be used with the following lemma from [17] to control the convergence rate of the sequence $(\lambda_k)_{k\geq 0}$, whose definition follows the classical use of estimate sequences [17]. This will provide us convergence rates both for the strongly convex and non-strongly convex cases.

Lemma A.4 (Lemma 2.2.4 from [17]).
If in the quantity γ_0 defined in (17) satisfies $\gamma_0 \geq \mu$, then the sequence $(\lambda_k)_{k\geq 0}$ from (15) satisfies

$$\lambda_k \leq \min\left\{(1-\sqrt{q})^k\,,\ \frac{4}{\left(2+k\sqrt{\frac{\gamma_0}{\kappa+\mu}}\right)^2}\right\}. \tag{18}$$

We may now move to the proof of the theorem.

A.1 Proof of Theorem A.3

The first step is to construct an estimate sequence is typically to find a sequence of lower bounds of F. By calling x_k^* the minimizer of G_k, the following one is used in [9]:

Lemma A.5 (Lower Bound for F near x_k^*).
For all x in $\mathbb{R}^p$,

$$F(x) \geq F(x_k^*) + \langle \kappa(y_{k-1} - x_k^*), x - x_k^*\rangle + \frac{\mu}{2}\|x - x_k^*\|^2. \tag{19}$$

Proof. By strong convexity, $G_k(x) \geq G_k(x_k^*) + \frac{\kappa+\mu}{2}\|x - x_k^*\|^2$, which is equivalent to

$$F(x) + \frac{\kappa}{2}\|x - y_k\|^2 \geq F(x_k^*) + \frac{\kappa}{2}\|x_k^* - y_{k-1}\|^2 + \frac{\kappa+\mu}{2}\|x - x_k^*\|^2.$$

After developing the quadratic terms, we directly obtain (19). □

Unfortunately, the exact value x_k^* is unknown in practice and the estimate sequence of [9] yields in fact an algorithm where the definition of the anchor point y_k involves the unknown quantity x_k^* instead of the approximate solutions x_k and x_{k-1} as in (6), as also noted by others [22]. To obtain a rigorous proof of convergence for Algorithm 1, it is thus necessary to refine the analysis of [9]. To that effect, we construct below a sequence of functions that approximately satisfies the definition of estimate sequences. Essentially, we replace in (19) the quantity x_k^* by x_k to obtain an approximate lower bound, and control the error by using the condition $G_k(x_k) - G_k(x_k^*) \leq \varepsilon_k$. This leads us to the following construction:

1. $\phi_0(x) = F(x_0) + \frac{\gamma_0}{2}\|x - x_0\|^2$;
2. For $k \geq 1$, we set

$$\phi_k(x) = (1-\alpha_{k-1})\phi_{k-1}(x) + \alpha_{k-1}[F(x_k) + \langle \kappa(y_{k-1} - x_k), x - x_k\rangle + \frac{\mu}{2}\|x - x_k\|^2],$$

where the value of γ_0, given in (17) will be explained later. Note that if one replaces x_k by x_k^* in the above construction, it is easy to show that $(\phi_k)_{k\geq 0}$ would be exactly an estimate sequence for F with the relation λ_k given in (15).

Before extending Lemma A.2 to deal with the approximate sequence and conclude the proof of the theorem, we need to characterize a few properties of the sequence $(\phi_k)_{k\geq 0}$. In particular, the functions ϕ_k are quadratic and admit a canonical form:

Lemma A.6 (Canonical Form of the Functions ϕ_k).
For all $k \geq 0$, ϕ_k can be written in the canonical form

$$\phi_k(x) = \phi_k^* + \frac{\gamma_k}{2}\|x - v_k\|^2,$$

where the sequences $(\gamma_k)_{k\geq 0}$, $(v_k)_{k\geq 0}$, and $(\phi_k^*)_{k\geq 0}$ are defined as follows

$$\gamma_k = (1-\alpha_{k-1})\gamma_{k-1} + \alpha_{k-1}\mu, \tag{20}$$

$$v_k = \frac{1}{\gamma_k}\left((1-\alpha_{k-1})\gamma_{k-1}v_{k-1} + \alpha_{k-1}\mu x_k - \alpha_{k-1}\kappa(y_{k-1}-x_k)\right), \tag{21}$$

$$\begin{aligned}\phi_k^* = {} & (1-\alpha_{k-1})\phi_{k-1}^* + \alpha_{k-1}F(x_k) - \frac{\alpha_{k-1}^2}{2\gamma_k}\|\kappa(y_{k-1}-x_k)\|^2 \\ & + \frac{\alpha_{k-1}(1-\alpha_{k-1})\gamma_{k-1}}{\gamma_k}\left(\frac{\mu}{2}\|x_k - v_{k-1}\|^2 + \langle \kappa(y_{k-1}-x_k), v_{k-1}-x_k\rangle\right),\end{aligned} \tag{22}$$

Proof. We have for all $k \geq 1$ and all x in $\mathbb{R}^p$,

$$\begin{aligned}\phi_k(x) = {} & (1-\alpha_{k-1})\left(\phi_{k-1}^* + \frac{\gamma_{k-1}}{2}\|x - v_{k-1}\|^2\right) \\ & + \alpha_{k-1}\left(F(x_k) + \langle \kappa(y_{k-1}-x_k), x - x_k\rangle + \frac{\mu}{2}\|x-x_k\|^2\right) \\ = {} & \phi_k^* + \frac{\gamma_k}{2}\|x - v_k\|^2.\end{aligned} \tag{23}$$

Differentiate twice the relations (23) gives us directly (20). Since v_k minimizes ϕ_k, the optimality condition $\nabla\phi_k(v_k) = 0$ gives

$$(1-\alpha_{k-1})\gamma_{k-1}(v_k - v_{k-1}) + \alpha_{k-1}\left(\kappa(y_{k-1}-x_k) + \mu(v_k - x_k)\right) = 0,$$

and then we obtain (21). Finally, apply $x = x_k$ to (23), which yields

$$\phi_k(x_k) = (1-\alpha_{k-1})\left(\phi_{k-1}^* + \frac{\gamma_{k-1}}{2}\|x_k - v_{k-1}\|^2\right) + \alpha_{k-1}F(x_k) = \phi_k^* + \frac{\gamma_k}{2}\|x_k - v_k\|^2.$$

Consequently,

$$\phi_k^* = (1-\alpha_{k-1})\phi_{k-1}^* + \alpha_{k-1}F(x_k) + (1-\alpha_{k-1})\frac{\gamma_{k-1}}{2}\|x_k - v_{k-1}\|^2 - \frac{\gamma_k}{2}\|x_k - v_k\|^2 \tag{24}$$

Using the expression of v_k from (21), we have

$$v_k - x_k = \frac{1}{\gamma_k}\left((1-\alpha_{k-1})\gamma_{k-1}(v_{k-1} - x_k) - \alpha_{k-1}\kappa(y_{k-1}-x_k)\right).$$

Therefore

$$\begin{aligned}\frac{\gamma_k}{2}\|x_k - v_k\|^2 = {} & \frac{(1-\alpha_{k-1})^2\gamma_{k-1}^2}{2\gamma_k}\|x_k - v_{k-1}\|^2 \\ & - \frac{(1-\alpha_{k-1})\alpha_{k-1}\gamma_{k-1}}{\gamma_k}\langle v_{k-1} - x_k, \kappa(y_{k-1}-x_k)\rangle + \frac{\alpha_{k-1}^2}{2\gamma_k}\|\kappa(y_{k-1}-x_k)\|^2.\end{aligned}$$

It remains to plug this relation into (24), use once (20), and we obtain the formula (22) for ϕ_k^*. □

We may now start analyzing the errors ε_k to control how far is the sequence $(\phi_k)_{k\geq 0}$ from an exact estimate sequence. For that, we need to understand the effect of replacing x_k^* by x_k in the lower bound (19). The following lemma will be useful for that purpose.

Lemma A.7 (Controlling the Approximate Lower Bound of F).
If $G_k(x_k) - G_k(x_k^*) \leq \varepsilon_k$, then for all x in $\mathbb{R}^p$,

$$F(x) \geq F(x_k) + \langle \kappa(y_{k-1}-x_k), x - x_k\rangle + \frac{\mu}{2}\|x-x_k\|^2 + (\kappa+\mu)\langle x_k - x_k^*, x - x_k\rangle - \varepsilon_k. \tag{25}$$

Proof. By strong convexity, for all x in $\mathbb{R}^p$,

$$G_k(x) \geq G_k^* + \frac{\kappa+\mu}{2}\|x - x_k^*\|^2,$$

where G_k^* is the minimum value of G_k. Replacing G_k by its definition (5) gives

$$\begin{aligned}
F(x) &\geq G_k^* + \frac{\kappa+\mu}{2}\|x - x_k^*\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2 \\
&= G_k(x_k) + (G_k^* - G_k(x_k)) + \frac{\kappa+\mu}{2}\|x - x_k^*\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2 \\
&\geq G_k(x_k) - \varepsilon_k + \frac{\kappa+\mu}{2}\|(x - x_k) + (x_k - x_k^*)\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2 \\
&\geq G_k(x_k) - \varepsilon_k + \frac{\kappa+\mu}{2}\|x - x_k\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2 + (\kappa+\mu)\langle x_k - x_k^*, x - x_k\rangle.
\end{aligned}$$

We conclude by noting that

$$\begin{aligned}
G_k(x_k) + \frac{\kappa}{2}\|x - x_k\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2 &= F(x_k) + \frac{\kappa}{2}\|x_k - y_{k-1}\|^2 + \frac{\kappa}{2}\|x - x_k\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2 \\
&= F(x_k) + \langle \kappa(y_{k-1} - x_k), x - x_k\rangle.
\end{aligned}$$

□

We can now show that Algorithm 1 generates iterates $(x_k)_{k\geq 0}$ that approximately satisfy the condition of Lemma A.2 from Nesterov [17].

Lemma A.8 (Relation between $(\phi_k)_{k\geq 0}$ and Algorithm 1).
Let ϕ_k be the estimate sequence constructed above. Then, Algorithm 1 generates iterates $(x_k)_{k\geq 0}$ such that

$$F(x_k) \leq \phi_k^* + \xi_k,$$

where the sequence $(\xi_k)_{k\geq 0}$ is defined by $\xi_0 = 0$ and

$$\xi_k = (1 - \alpha_{k-1})(\xi_{k-1} + \varepsilon_k - (\kappa+\mu)\langle x_k - x_k^*, x_{k-1} - x_k\rangle).$$

Proof. We proceed by induction. For $k = 0$, $\phi_0^* = F(x_0)$ and $\xi_0 = 0$.
Assume now that $F(x_{k-1}) \leq \phi_{k-1}^* + \xi_{k-1}$. Then,

$$\begin{aligned}
\phi_{k-1}^* &\geq F(x_{k-1}) - \xi_{k-1} \\
&\geq F(x_k) + \langle \kappa(y_{k-1} - x_k), x_{k-1} - x_k\rangle + (\kappa+\mu)\langle x_k - x_k^*, x_{k-1} - x_k\rangle - \varepsilon_k - \xi_{k-1} \\
&= F(x_k) + \langle \kappa(y_{k-1} - x_k), x_{k-1} - x_k\rangle - \xi_k/(1 - \alpha_{k-1}),
\end{aligned}$$

where the second inequality is due to (25). By Lemma A.6, we now have,

$$\begin{aligned}
\phi_k^* &= (1 - \alpha_{k-1})\phi_{k-1}^* + \alpha_{k-1}F(x_k) - \frac{\alpha_{k-1}^2}{2\gamma_k}\|\kappa(y_{k-1} - x_k)\|^2 \\
&\quad + \frac{\alpha_{k-1}(1 - \alpha_{k-1})\gamma_{k-1}}{\gamma_k}\left(\frac{\mu}{2}\|x_k - v_{k-1}\|^2 + \langle \kappa(y_{k-1} - x_k), v_{k-1} - x_k\rangle\right) \\
&\geq (1 - \alpha_{k-1})\left(F(x_k) + \langle \kappa(y_{k-1} - x_k), x_{k-1} - x_k\rangle\right) - \xi_k + \alpha_{k-1}F(x_k) \\
&\quad - \frac{\alpha_{k-1}^2}{2\gamma_k}\|\kappa(y_{k-1} - x_k)\|^2 + \frac{\alpha_{k-1}(1 - \alpha_{k-1})\gamma_{k-1}}{\gamma_k}\langle \kappa(y_{k-1} - x_k), v_{k-1} - x_k\rangle. \\
&= F(x_k) + (1 - \alpha_{k-1})\langle \kappa(y_{k-1} - x_k), x_{k-1} - x_k + \frac{\alpha_{k-1}\gamma_{k-1}}{\gamma_k}(v_{k-1} - x_k)\rangle \\
&\quad - \frac{\alpha_{k-1}^2}{2\gamma_k}\|\kappa(y_{k-1} - x_k)\|^2 - \xi_k \\
&= F(x_k) + (1 - \alpha_{k-1})\langle \kappa(y_{k-1} - x_k), x_{k-1} - y_{k-1} + \frac{\alpha_{k-1}\gamma_{k-1}}{\gamma_k}(v_{k-1} - y_{k-1})\rangle \\
&\quad + \left(1 - \frac{(\kappa+2\mu)\alpha_{k-1}^2}{2\gamma_k}\right)\kappa\|(y_{k-1} - x_k)\|^2 - \xi_k.
\end{aligned}$$

We now need to show that the choice of the sequences $(\alpha_k)_{k\geq 0}$ and $(y_k)_{k\geq 0}$ will cancel all the terms involving $y_{k-1} - x_k$. In other words, we want to show that

$$x_{k-1} - y_{k-1} + \frac{\alpha_{k-1}\gamma_{k-1}}{\gamma_k}(v_{k-1} - y_{k-1}) = 0, \tag{26}$$

and we want to show that

$$1 - (\kappa + \mu)\frac{\alpha_{k-1}^2}{\gamma_k} = 0, \tag{27}$$

which will be sufficient to conclude that $\phi_k^* + \xi_k \geq F(x_k)$. The relation (27) can be obtained from the definition of α_k in (6) and the form of γ_k given in (20). We have indeed from (6) that

$$(\kappa + \mu)\alpha_k^2 = (1 - \alpha_k)(\kappa + \mu)\alpha_{k-1}^2 + \alpha_k \mu.$$

Then, the quantity $(\kappa + \mu)\alpha_k^2$ follows the same recursion as γ_{k+1} in (20). Moreover, we have

$$\gamma_1 = (1 - \alpha_0)\gamma_0 + \mu\alpha_0 = (\kappa + \mu)\alpha_0^2,$$

from the definition of γ_0 in (17). We can then conclude by induction that $\gamma_{k+1} = (\kappa + \mu)\alpha_k^2$ for all $k \geq 0$ and (27) is satisfied.

To prove (26), we assume that y_{k-1} is chosen such that (26) is satisfied, and show that it is equivalent to defining y_k as in (6). By lemma A.6,

$$\begin{aligned}
v_k &= \frac{1}{\gamma_k}\left((1 - \alpha_{k-1})\gamma_{k-1}v_{k-1} + \alpha_{k-1}\mu x_k - \alpha_{k-1}\kappa(y_{k-1} - x_k)\right) \\
&= \frac{1}{\gamma_k}\left(\frac{(1 - \alpha_{k-1})}{\alpha_{k-1}}((\gamma_k + \alpha_{k-1}\gamma_{k-1})y_{k-1} - \gamma_k x_{k-1}) + \alpha_{k-1}\mu x_k - \alpha_{k-1}\kappa(y_{k-1} - x_k)\right) \\
&= \frac{1}{\gamma_k}\left(\frac{(1 - \alpha_{k-1})}{\alpha_{k-1}}((\gamma_{k-1} + \alpha_{k-1}\mu)y_{k-1} - \gamma_k x_{k-1}) + \alpha_{k-1}(\mu + \kappa)x_k - \alpha_{k-1}\kappa y_{k-1}\right) \\
&= \frac{1}{\gamma_k}\left(\frac{1}{\alpha_{k-1}}(\gamma_k - \mu\alpha_{k-1}^2)y_{k-1} - \frac{(1 - \alpha_{k-1})}{\alpha_{k-1}}\gamma_k x_{k-1} + \frac{\gamma_k}{\alpha_{k-1}}x_k - \alpha_{k-1}\kappa y_{k-1}\right) \\
&= \frac{1}{\alpha_{k-1}}(x_k - (1 - \alpha_{k-1})x_{k-1}),
\end{aligned} \tag{28}$$

As a result, using (26) by replacing $k - 1$ by k yields

$$y_k = x_k + \frac{\alpha_{k-1}(1 - \alpha_{k-1})}{\alpha_{k-1}^2 + \alpha_k}(x_k - x_{k-1}),$$

and we obtain the original equivalent definition of (6). This concludes the proof. □

With this lemma in hand, we introduce the following proposition, which brings us almost to Theorem A.3, which we want to prove.

Proposition A.9 (Auxiliary Proposition for Theorem A.3).
Let us consider the sequence $(\lambda_k)_{k\geq 0}$ defined in (15). Then, the sequence $(x_k)_{k\geq 0}$ satisfies

$$\frac{1}{\lambda_k}(F(x_k) - F^* + \frac{\gamma_k}{2}\|x^* - v_k\|^2) \leq \phi_0(x^*) - F^* + \sum_{i=1}^{k}\frac{\varepsilon_i}{\lambda_i} + \sum_{i=1}^{k}\frac{\sqrt{2\varepsilon_i\gamma_i}}{\lambda_i}\|x^* - v_i\|,$$

where x^* is a minimizer of F and F^* its minimum value.

Proof. By the definition of the function ϕ_k, we have

$$\begin{aligned}
\phi_k(x^*) &= (1 - \alpha_{k-1})\phi_{k-1}(x^*) + \alpha_{k-1}[F(x_k) + \langle\kappa(y_{k-1} - x_k), x^* - x_k\rangle + \frac{\mu}{2}\|x^* - x_k\|^2] \\
&\leq (1 - \alpha_{k-1})\phi_{k-1}(x^*) + \alpha_{k-1}[F(x^*) + \varepsilon_k - (\kappa + \mu)\langle x_k - x_k^*, x^* - x_k\rangle],
\end{aligned}$$

where the inequality comes from (25). Therefore, by using the definition of ξ_k in Lemma A.8,

$$\begin{aligned}
& \phi_k(x^*) + \xi_k - F^* \\
\leq\ & (1 - \alpha_{k-1})(\phi_{k-1}(x^*) + \xi_{k-1} - F^*) + \varepsilon_k - (\kappa + \mu)\langle x_k - x_k^*, (1 - \alpha_{k-1})x_{k-1} + \alpha_{k-1}x^* - x_k\rangle \\
=\ & (1 - \alpha_{k-1})(\phi_{k-1}(x^*) + \xi_{k-1} - F^*) + \varepsilon_k - \alpha_{k-1}(\kappa + \mu)\langle x_k - x_k^*, x^* - v_k\rangle \\
\leq\ & (1 - \alpha_{k-1})(\phi_{k-1}(x^*) + \xi_{k-1} - F^*) + \varepsilon_k + \alpha_{k-1}(\kappa + \mu)\|x_k - x_k^*\|\|x^* - v_k\| \\
\leq\ & (1 - \alpha_{k-1})(\phi_{k-1}(x^*) + \xi_{k-1} - F^*) + \varepsilon_k + \alpha_{k-1}\sqrt{2(\kappa + \mu)\varepsilon_k}\|x^* - v_k\| \\
=\ & (1 - \alpha_{k-1})(\phi_{k-1}(x^*) + \xi_{k-1} - F^*) + \varepsilon_k + \sqrt{2\varepsilon_k\gamma_k}\|x^* - v_k\|,
\end{aligned}$$

where the first equality uses the relation (28), the last inequality comes from the strong convexity relation $\varepsilon_k \geq G_k(x_k) - G_k(x_k^*) \geq (1/2)(\kappa+\mu)\|x_k^* - x_k\|^2$, and the last equality uses the relation $\gamma_k = (\kappa+\mu)\alpha_{k-1}^2$.

Dividing both sides by λ_k yields

$$\frac{1}{\lambda_k}(\phi_k(x^*) + \xi_k - F^*) \leq \frac{1}{\lambda_{k-1}}(\phi_{k-1}(x^*) + \xi_{k-1} - F^*) + \frac{\varepsilon_k}{\lambda_k} + \frac{\sqrt{2\varepsilon_k\gamma_k}}{\lambda_k}\|x^* - v_k\|.$$

A simple recurrence gives,

$$\frac{1}{\lambda_k}(\phi_k(x^*) + \xi_k - F^*) \leq \phi_0(x^*) - F^* + \sum_{i=1}^{k}\frac{\varepsilon_i}{\lambda_i} + \sum_{i=1}^{k}\frac{\sqrt{2\varepsilon_i\gamma_i}}{\lambda_i}\|x^* - v_i\|.$$

Finally, by lemmas A.6 and A.8,

$$\phi_k(x^*) + \xi_k - F^* = \frac{\gamma_k}{2}\|x^* - v_k\|^2 + \phi_k^* + \xi_k - F^* \geq \frac{\gamma_k}{2}\|x^* - v_k\|^2 + F(x_k) - F^*.$$

As a result,

$$\frac{1}{\lambda_k}(F(x_k) - F^* + \frac{\gamma_k}{2}\|x^* - v_k\|^2) \leq \phi_0(x^*) - F^* + \sum_{i=1}^{k}\frac{\varepsilon_i}{\lambda_i} + \sum_{i=1}^{k}\frac{\sqrt{2\varepsilon_i\gamma_i}}{\lambda_i}\|x^* - v_i\|. \quad (29)$$

□

To control the error term on the right and finish the proof of Theorem A.3, we are going to borrow some methodology used to analyze the convergence of inexact proximal gradient algorithms from [23], and use an extension of a lemma presented in [23] to bound the value of $\|v_i - x^*\|$. This lemma is presented below.

Lemma A.10 (Simple Lemma on Non-Negative Sequences).
Assume that the nonnegative sequences $(u_k)_{k\geq 0}$ and $(a_k)_{k\geq 0}$ satisfy the following recursion for all $k \geq 0$:

$$u_k^2 \leq S_k + \sum_{i=1}^{k} a_i u_i, \quad (30)$$

where $(S_k)_{k\geq 0}$ is an increasing sequence such that $S_0 \geq u_0^2$. Then,

$$u_k \leq \frac{1}{2}\sum_{i=1}^{k} a_i + \sqrt{\Big(\frac{1}{2}\sum_{i=1}^{k} a_i\Big)^2 + S_k}. \quad (31)$$

Moreover,

$$S_k + \sum_{i=1}^{k} a_i u_i \leq \left(\sqrt{S_k} + \sum_{i=1}^{k} a_i\right)^2.$$

Proof. The first part—that is, Eq. (31)—is exactly Lemma 1 from [23]. The proof is in their appendix. Then, by calling b_k the right-hand side of (31), we have that for all $k \geq 1$, $u_k \leq b_k$. Furthermore $(b_k)_{k\geq 0}$ is increasing and we have

$$S_k + \sum_{i=1}^{k} a_i u_i \leq S_k + \sum_{i=1}^{k} a_i b_i \leq S_k + \Big(\sum_{i=1}^{k} a_i\Big) b_k = b_k^2,$$

and using the inequality $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$, we have

$$b_k = \frac{1}{2}\sum_{i=1}^{k} a_i + \sqrt{\Big(\frac{1}{2}\sum_{i=1}^{k} a_i\Big)^2 + S_k} \;\leq\; \frac{1}{2}\sum_{i=1}^{k} a_i + \sqrt{\Big(\frac{1}{2}\sum_{i=1}^{k} a_i\Big)^2} + \sqrt{S_k} \;=\; \sqrt{S_k} + \sum_{i=1}^{k} a_i.$$

As a result,

$$S_k + \sum_{i=1}^{k} a_i u_i \leq b_k^2 \leq \left(\sqrt{S_k} + \sum_{i=1}^{k} a_i\right)^2.$$

□

We are now in shape to conclude the proof of Theorem A.3. We apply the previous lemma to (29):

$$\frac{1}{\lambda_k}\Big(\frac{\gamma_k}{2}\|x^*-v_k\|^2+F(x_k)-F^*\Big)\leq \phi_0(x^*)-F^*+\sum_{i=1}^k\frac{\varepsilon_i}{\lambda_i}+\sum_{i=1}^k\frac{\sqrt{2\varepsilon_i\gamma_i}}{\lambda_i}\|x^*-v_i\|.$$

Since $F(x_k)-F^*\geq 0$, we have

$$\underbrace{\frac{\gamma_k}{2\lambda_k}\|x^*-v_k\|^2}_{u_k^2}\leq \underbrace{\phi_0(x^*)-F^*+\sum_{i=1}^k\frac{\varepsilon_i}{\lambda_i}}_{S_k}+\sum_{i=1}^k\underbrace{\frac{\sqrt{2\varepsilon_i\gamma_i}}{\lambda_i}\|x^*-v_i\|}_{a_iu_i},$$

with

$$u_i=\sqrt{\frac{\gamma_i}{2\lambda_i}}\|x^*-v_i\| \quad \text{and} \quad a_i=2\sqrt{\frac{\varepsilon_i}{\lambda_i}} \text{ and } S_k=\phi_0(x^*)-F^*+\sum_{i=1}^k\frac{\varepsilon_i}{\lambda_i}.$$

Then by Lemma A.10, we have

$$F(x_k)-F^*\leq \lambda_k\left(S_k+\sum_{i=1}^k a_iu_i\right)\leq \lambda_k\left(\sqrt{S_k}+\sum_{i=1}^k a_i\right)^2=\lambda_k\left(\sqrt{S_k}+2\sum_{i=1}^k\sqrt{\frac{\varepsilon_i}{\lambda_i}}\right)^2,$$

which is the desired result.

B Proofs of the Main Theorems and Propositions

B.1 Proof of Theorem 3.1

Proof. We simply use Theorem A.3 and specialize it to the choice of parameters. The initialization $\alpha_0=\sqrt{q}$ leads to a particularly simple form of the algorithm, where $\alpha_k=\sqrt{q}$ for all $k\geq 0$. Therefore, the sequence $(\lambda_k)_{k\geq 0}$ from Theorem A.3 is also simple. For all $k\geq 0$, we indeed have $\lambda_k=(1-\sqrt{q})^k$. To upper-bound the quantity S_k from Theorem A.3, we now remark that $\gamma_0=\mu$ and thus, by strong convexity of F,

$$F(x_0)+\frac{\gamma_0}{2}\|x_0-x^*\|^2-F^*\leq 2(F(x_0)-F^*).$$

Therefore,

$$\begin{aligned}
\sqrt{S_k}+2\sum_{i=1}^k\sqrt{\frac{\varepsilon_i}{\lambda_i}} &= \sqrt{F(x_0)+\frac{\gamma_0}{2}\|x_0-x^*\|^2-F^*+\sum_{i=1}^k\frac{\varepsilon_i}{\lambda_i}}+2\sum_{i=1}^k\sqrt{\frac{\varepsilon_i}{\lambda_i}}\\
&\leq \sqrt{F(x_0)+\frac{\gamma_0}{2}\|x_0-x^*\|^2-F^*}+3\sum_{i=1}^k\sqrt{\frac{\varepsilon_i}{\lambda_i}}\\
&\leq \sqrt{2(F(x_0)-F^*)}+3\sum_{i=1}^k\sqrt{\frac{\varepsilon_i}{\lambda_i}}\\
&= \sqrt{2(F(x_0)-F^*)}\left[1+\sum_{i=1}^k\underbrace{\left(\sqrt{\frac{1-\rho}{1-\sqrt{q}}}\right)^i}_{\eta}\right]\\
&= \sqrt{2(F(x_0)-F^*)}\,\frac{\eta^{k+1}-1}{\eta-1}\\
&\leq \sqrt{2(F(x_0)-F^*)}\,\frac{\eta^{k+1}}{\eta-1}.
\end{aligned}$$

Therefore, Theorem A.3 combined with the previous inequality gives us

$$\begin{aligned}
F(x_k) - F^* &\leq 2\lambda_k(F(x_0) - F^*) \left(\frac{\eta^{k+1}}{\eta - 1}\right)^2 \\
&= 2\left(\frac{\eta}{\eta - 1}\right)^2 (1-\rho)^k (F(x_0) - F^*) \\
&= 2\left(\frac{\sqrt{1-\rho}}{\sqrt{1-\rho} - \sqrt{1-\sqrt{q}}}\right)^2 (1-\rho)^k (F(x_0) - F^*) \\
&= 2\left(\frac{1}{\sqrt{1-\rho} - \sqrt{1-\sqrt{q}}}\right)^2 (1-\rho)^{k+1} (F(x_0) - F^*).
\end{aligned}$$

Since $\sqrt{1-x} + \frac{x}{2}$ is decreasing in $[0,1]$, we have $\sqrt{1-\rho} + \frac{\rho}{2} \geq \sqrt{1-\sqrt{q}} + \frac{\sqrt{q}}{2}$. Consequently,

$$F(x_k) - F^* \leq \frac{8}{(\sqrt{q} - \rho)^2} (1-\rho)^{k+1} (F(x_0) - F^*).$$

□

B.2 Proof of Proposition 3.2

To control the number of calls of $\mathcal{M}$, we need to upper bound $G_k(x_{k-1}) - G_k^*$ which is given by the following lemma:

Lemma B.1 (Relation between $G_k(x_{k-1})$ and ε_{k-1}).
Let $(x_k)_{k\geq 0}$ and $(y_k)_{k\geq 0}$ be generated by Algorithm 1. Remember that by definition of x_{k-1},

$$G_{k-1}(x_{k-1}) - G_{k-1}^* \leq \varepsilon_{k-1}.$$

Then, we have

$$G_k(x_{k-1}) - G_k^* \leq 2\varepsilon_{k-1} + \frac{\kappa^2}{\kappa + \mu} \|y_{k-1} - y_{k-2}\|^2. \tag{32}$$

Proof. We first remark that for any x, y in $\mathbb{R}^p$, we have

$$G_k(x) - G_{k-1}(x) = G_k(y) - G_{k-1}(y) + \kappa\langle y - x, y_{k-1} - y_{k-2}\rangle, \quad \forall k \geq 2,$$

which can be shown by using the respective definitions of G_k and G_{k-1} and manipulate the quadratic term resulting from the difference $G_k(x) - G_{k-1}(x)$.

Plugging $x = x_{k-1}$ and $y = x_k^*$ in the previous relation yields

$$\begin{aligned}
G_k(x_{k-1}) - G_k^* &= G_{k-1}(x_{k-1}) - G_{k-1}(x_k^*) + \kappa\langle x_k^* - x_{k-1}, y_{k-1} - y_{k-2}\rangle \\
&= G_{k-1}(x_{k-1}) - G_{k-1}^* + G_{k-1}^* - G_{k-1}(x_k^*) + \kappa\langle x_k^* - x_{k-1}, y_{k-1} - y_{k-2}\rangle \\
&\leq \varepsilon_{k-1} + G_{k-1}^* - G_{k-1}(x_k^*) + \kappa\langle x_k^* - x_{k-1}, y_{k-1} - y_{k-2}\rangle \\
&\leq \varepsilon_{k-1} - \frac{\mu + \kappa}{2} \|x_k^* - x_{k-1}^*\|^2 + \kappa\langle x_k^* - x_{k-1}, y_{k-1} - y_{k-2}\rangle,
\end{aligned} \tag{33}$$

where the last inequality comes from the strong convexity inequality of

$$G_{k-1}(x_k^*) \geq G_{k-1}^* + \frac{\mu + \kappa}{2} \|x_k^* - x_{k-1}^*\|^2.$$

Moreover, from the inequality $\langle x, y\rangle \leq \frac{1}{2}\|x\|^2 + \frac{1}{2}\|y\|^2$, we also have

$$\kappa\langle x_k^* - x_{k-1}^*, y_{k-1} - y_{k-2}\rangle \leq \frac{\mu + \kappa}{2} \|x_k^* - x_{k-1}^*\|^2 + \frac{\kappa^2}{2(\kappa + \mu)} \|y_{k-1} - y_{k-2}\|^2, \tag{34}$$

and

$$\begin{aligned}
\kappa\langle x_{k-1}^* - x_{k-1}, y_{k-1} - y_{k-2}\rangle &\leq \frac{\mu + \kappa}{2} \|x_{k-1}^* - x_{k-1}\|^2 + \frac{\kappa^2}{2(\kappa + \mu)} \|y_{k-1} - y_{k-2}\|^2 \\
&\leq \varepsilon_{k-1} + \frac{\kappa^2}{2(\kappa + \mu)} \|y_{k-1} - y_{k-2}\|^2.
\end{aligned} \tag{35}$$

Summing inequalities (33), (34) and (35) gives the desired result. □

Next, we need to upper-bound the term $\|y_{k-1}-y_{k-2}\|^2$, which was also required in the convergence proof of the accelerated SDCA algorithm [26]. We follow here their methodology.

Lemma B.2 (Control of the term $\|y_{k-1}-y_{k-2}\|^2$.).
Let us consider the iterates $(x_k)_{k\geq 0}$ and $(y_k)_{k\geq 0}$ produced by Algorithm 1, and define

$$\delta_k = C(1-\rho)^{k+1}(F(x_0)-F^*),$$

which appears in Theorem 3.1 and which is such that $F(x_k)-F^*\leq \delta_k$. Then, for any $k\geq 3$,

$$\|y_{k-1}-y_{k-2}\|^2 \leq \frac{72}{\mu}\delta_{k-3}.$$

Proof. We follow here [26]. By definition of y_k, we have

$$\begin{aligned}\|y_{k-1}-y_{k-2}\| &= \|x_{k-1}+\beta_{k-1}(x_{k-1}-x_{k-2})-x_{k-2}-\beta_{k-2}(x_{k-2}-x_{k-3})\| \\ &\leq (1+\beta_{k-1})\|x_{k-1}-x_{k-2}\|+\beta_{k-2}\|x_{k-2}-x_{k-3}\| \\ &\leq 3\max\left\{\|x_{k-1}-x_{k-2}\|, \|x_{k-2}-x_{k-3}\|\right\},\end{aligned}$$

where β_k is defined in (6). The last inequality was due to the fact that $\beta_k\leq 1$. Indeed, the specific choice of $\alpha_0=\sqrt{q}$ in Theorem A.3 leads to $\beta_k=\frac{\sqrt{q}-q}{\sqrt{q}+q}\leq 1$ for all k. Note, however, that this relation $\beta_k\leq 1$ is true regardless of the choice of α_0:

$$\beta_k^2 = \frac{\left(\alpha_{k-1}-\alpha_{k-1}^2\right)^2}{\left(\alpha_{k-1}^2+\alpha_k\right)^2} = \frac{\alpha_{k-1}^2+\alpha_{k-1}^4-2\alpha_{k-1}^3}{\alpha_k^2+2\alpha_k\alpha_{k-1}^2+\alpha_{k-1}^4} = \frac{\alpha_{k-1}^2+\alpha_{k-1}^4-2\alpha_{k-1}^3}{\alpha_{k-1}^2+\alpha_{k-1}^4+q\alpha_k+\alpha_k\alpha_{k-1}^2}\leq 1,$$

where the last equality uses the relation $\alpha_k^2+\alpha_k\alpha_{k-1}^2=\alpha_{k-1}^2+q\alpha_k$ from Algorithm 1. To conclude the lemma, we notice that by triangle inequality

$$\|x_k-x_{k-1}\|\leq \|x_k-x^*\|+\|x_{k-1}-x^*\|,$$

and by strong convexity of F

$$\frac{\mu}{2}\|x_k-x^*\|^2\leq F(x_k)-F(x^*)\leq \delta_k.$$

As a result,

$$\begin{aligned}\|y_{k-1}-y_{k-2}\|^2 &\leq 9\max\left\{\|x_{k-1}-x_{k-2}\|^2, \|x_{k-2}-x_{k-3}\|^2\right\} \\ &\leq 36\max\left\{\|x_{k-1}-x^*\|^2, \|x_{k-2}-x^*\|^2, \|x_{k-3}-x^*\|^2\right\} \\ &\leq \frac{72}{\mu}\delta_{k-3}.\end{aligned}$$

□

We are now in shape to conclude the proof of Proposition 3.2.

By Proposition B.1 and lemma B.2, we have for all $k\geq 3$,

$$G_k(x_{k-1})-G_k^*\leq 2\varepsilon_{k-1}+\frac{\kappa^2}{\kappa+\mu}\frac{72}{\mu}\delta_{k-3}\leq 2\varepsilon_{k-1}+\frac{72\kappa}{\mu}\delta_{k-3}.$$

Let $(z_t)_{t\geq 0}$ be the sequence of using $\mathcal{M}$ to solve G_k with initialization $z_0=x_{k-1}$. By assumption (8), we have

$$G_k(z_t)-G_k^*\leq A\left(1-\tau_{\mathcal{M}}\right)^t(G_k(x_{k-1})-G_k^*)\leq A\,e^{-\tau_{\mathcal{M}}t}(G_k(x_{k-1})-G_k^*).$$

The number of iterations $T_{\mathcal{M}}$ of $\mathcal{M}$ to guarantee an accuracy of ε_k needs to satisfy

$$A\,e^{-\tau_{\mathcal{M}}T_{\mathcal{M}}}(G_k(x_{k-1})-G_k^*)\leq \varepsilon_k,$$

which gives

$$T_{\mathcal{M}}=\left\lceil \frac{1}{\tau_{\mathcal{M}}}\log\left(\frac{A(G_k(x_{k-1})-G_k^*)}{\varepsilon_k}\right)\right\rceil. \tag{36}$$

Then, it remains to upper-bound

$$\frac{G_k(x_{k-1}) - G_k^*}{\varepsilon_k} \leq \frac{2\varepsilon_{k-1} + \frac{72\kappa}{\mu}\delta_{k-3}}{\varepsilon_k} = \frac{2(1-\rho) + \frac{72\kappa}{\mu} \cdot \frac{9C}{2}}{(1-\rho)^2} = \frac{2}{1-\rho} + \frac{2592\kappa}{\mu(1-\rho)^2(\sqrt{q}-\rho)^2}.$$

Let us denote R the right-hand side. We remark that this upper bound holds for $k \geq 3$. We now consider the cases $k = 1$ and $k = 2$.

When $k = 1$, $G_1(x) = F(x) + \frac{\kappa}{2}\|x - y_0\|^2$. Note that $x_0 = y_0$, then $G_1(x_0) = F(x_0)$. As a result,

$$G_1(x_0) - G_1^* = F(x_0) - F(x_1^*) - \frac{\kappa}{2}\|x_1^* - y_0\|^2 \leq F(x_0) - F(x_1^*) \leq F(x_0) - F^*.$$

Therefore,

$$\frac{G_1(x_0) - G_1^*}{\varepsilon_1} \leq \frac{F(x_0) - F^*}{\varepsilon_1} = \frac{9}{2(1-\rho)} \leq R.$$

When $k = 2$, we remark that $y_1 - y_0 = (1 + \beta_1)(x_1 - x_0)$. Then, by following similar steps as in the proof of Lemma B.2, we have

$$\|y_1 - y_0\|^2 \leq 4\|x_1 - x_0\|^2 \leq \frac{32\delta_0}{\mu},$$

which is smaller than $\frac{72\delta_{-1}}{\mu}$. Therefore, the previous steps from the case $k \geq 3$ apply and $\frac{G_2(x_1) - G_2^*}{\varepsilon_2} \leq R$. Thus, for any $k \geq 1$,

$$T_{\mathcal{M}} \leq \left\lceil \frac{\log(AR)}{\tau_{\mathcal{M}}} \right\rceil, \tag{37}$$

which concludes the proof.

B.3 Proof of Theorem 3.3.

We will again Theorem A.3 and specialize it to the choice of parameters. To apply it, the following Lemma will be useful to control the growth of $(\lambda_k)_{k\geq 0}$.

Lemma B.3 (Growth of the Sequence $(\lambda_k)_{k\geq 0}$).
Let $(\lambda_k)_{k\geq 0}$ be the sequence defined in (15) where $(\alpha_k)_{k\geq 0}$ is produced by Algorithm 1 with $\alpha_0 = \frac{\sqrt{5}-1}{2}$ and $\mu = 0$. Then, we have the following bounds for all $k \geq 0$,

$$\frac{4}{(k+2)^2} \geq \lambda_k \geq \frac{2}{(k+2)^2}.$$

Proof. Note that by definition of α_k, we have for all $k \geq 1$,

$$\alpha_k^2 = (1 - \alpha_k)\alpha_{k-1}^2 = \prod_{i=1}^{k}(1 - \alpha_i)\alpha_0^2 = \lambda_{k+1}\frac{\alpha_0^2}{1 - \alpha_0} = \lambda_{k+1}.$$

With the choice of α_0, the quantity γ_0 defined in (17) is equal to κ. By Lemma A.4, we have $\lambda_k \leq \frac{4}{(k+2)^2}$ for all $k \geq 0$ and thus $\alpha_k \leq \frac{2}{k+3}$ for all $k \geq 1$ (it is also easy to check numerically that this is also true for $k = 0$ since $\frac{\sqrt{5}-1}{2} \approx 0.62 \leq \frac{2}{3}$). We now have all we need to conclude the lemma:

$$\lambda_k = \prod_{i=0}^{k-1}(1 - \alpha_i) \geq \prod_{i=0}^{k-1}\left(1 - \frac{2}{i+3}\right) = \frac{2}{(k+2)(k+1)} \geq \frac{2}{(k+2)^2}.$$

□

With this lemma in hand, we may now proceed and apply Theorem A.3. We have remarked in the proof of the previous lemma that $\gamma_0 = \kappa$. Then,

$$\begin{aligned}\sqrt{S_k} + 2\sum_{i=1}^{k}\sqrt{\frac{\varepsilon_i}{\lambda_i}} &= \sqrt{F(x_0) - F^* + \frac{\kappa}{2}\|x_0 - x^*\|^2 + \sum_{i=1}^{k}\frac{\varepsilon_i}{\lambda_i}} + 2\sum_{i=1}^{k}\sqrt{\frac{\varepsilon_i}{\lambda_i}} \\ &\leq \sqrt{F(x_0) - F^* + \frac{\kappa}{2}\|x_0 - x^*\|^2} + 3\sum_{i=1}^{k}\sqrt{\frac{\varepsilon_i}{\lambda_i}} \\ &\leq \sqrt{\frac{\kappa}{2}\|x_0 - x^*\|^2} + \sqrt{F(x_0) - F^*}\left(1 + \sum_{i=1}^{k}\frac{1}{(i+2)^{1+\eta/2}}\right),\end{aligned}$$

where the last inequality uses Lemma B.3 to upper-bound the ratio ε_i/λ_i. Moreover,

$$\sum_{i=1}^{k}\frac{1}{(i+2)^{1+\eta/2}} \leq \sum_{i=2}^{\infty}\frac{1}{i^{1+\eta/2}} \leq \int_1^{\infty}\frac{1}{x^{1+\eta/2}}\,\mathrm{d}x = \frac{2}{\eta}.$$

Therefore, by (16) from Theorem A.3,

$$\begin{aligned}F(x_k) - F^* &\leq \lambda_k\left(\sqrt{S_k} + 2\sum_{i=1}^{k}\sqrt{\frac{\varepsilon_i}{\lambda_i}}\right)^2 \\ &\leq \frac{4}{(k+2)^2}\left(\sqrt{F(x_0) - F^*}\left(1 + \frac{2}{\eta}\right) + \sqrt{\frac{\kappa}{2}\|x_0 - x^*\|^2}\right)^2 \\ &\leq \frac{8}{(k+2)^2}\left(\left(1 + \frac{2}{\eta}\right)^2(F(x_0) - F^*) + \frac{\kappa}{2}\|x_0 - x^*\|^2\right).\end{aligned}$$

The last inequality uses $(a+b)^2 \leq 2(a^2 + b^2)$.

B.4 Proof of Proposition 3.4

When $\mu = 0$, we remark that Proposition B.1 still holds but Lemma B.2 does not. The main difficulty is thus to find another way to control the quantity $\|y_{k-1} - y_{k-2}\|$.

Since $F(x_k) - F^*$ is bounded by Theorem 3.3, we may use the bounded level set assumptions to ensure that there exists $B > 0$ such that $\|x_k - x^*\| \leq B$ for any $k \geq 0$ where x^* is a minimizer of F. We can now follow similar steps as in the proof of Lemma B.2, and show that

$$\|y_{k-1} - y_{k-2}\|^2 \leq 36B^2.$$

Then by Proposition B.1,

$$G_k(x_{k-1}) - G_k^* \leq 2\varepsilon_{k-1} + 36\kappa B^2.$$

Since $\kappa > 0$, G_k is strongly convex, then using the same argument as in the strongly convex case, the number of calls for $\mathcal{M}$ is given by

$$\left\lceil \frac{1}{\tau_{\mathcal{M}}}\log\left(\frac{A(G_k(x_{k-1}) - G_k^*)}{\varepsilon_k}\right)\right\rceil. \tag{38}$$

Again, we need to upper bound it

$$\frac{G_k(x_{k-1}) - G_k^*}{\varepsilon_k} \leq \frac{2\varepsilon_{k-1} + 36\kappa B^2}{\varepsilon_k} = \frac{2(k+1)^{4+\eta}}{(k+2)^{4+\eta}} + \frac{162\kappa B^2(k+2)^{4+\eta}}{(F(x_0) - F^*)}.$$

The right hand side is upper-bounded by $O((k+2)^{4+\eta})$. Plugging this relation into (38) gives the desired result.

C Derivation of Global Convergence Rates

We give here a generic "template" for computing the *optimal choice of* κ to accelerate a given algorithm $\mathcal{M}$, and therefore compute the rate of convergence of the accelerated algorithm $\mathcal{A}$.

We assume here that $\mathcal{M}$ is a *randomized* first-order optimization algorithm, *i.e.* the iterates (x_k) generated by $\mathcal{M}$ are a sequence of random variables; specialization to a deterministic algorithm is straightforward. Also, for the sake of simplicity, we shall use simple notations to denote the stopping time to reach accuracy ε. Definition and notation using filtrations, σ-algebras, etc. are unnecessary for our purpose here where the quantity of interest has a clear interpretation.

Assume that algorithm $\mathcal{M}$ enjoys a linear rate of convergence, in expectation. There exists constants $\mathcal{C}_{\mathcal{M},F}$ and $\tau_{\mathcal{M},F}$ such that the sequence of iterates $(x_k)_{k\geq 0}$ for minimizing a strongly-convex objective F satisfies

$$\mathbb{E}\left[F(x_k) - F^*\right] \leq \mathcal{C}_{\mathcal{M},F}\left(1 - \tau_{\mathcal{M},F}\right)^k. \tag{39}$$

Define the random variable $T_{\mathcal{M},F}(\varepsilon)$ (stopping time) corresponding to the minimum number of iterations to guarantee an accuracy ε in the course of running $\mathcal{M}$

$$T_{\mathcal{M},F}(\varepsilon) := \inf\left\{k \geq 1,\ F(x_k) - F^* \leq \varepsilon\right\} \tag{40}$$

Then, an upper bound on the expectation is provided by the following lemma.

Lemma C.1 (Upper Bound on the expectation of $T_{\mathcal{M},F}(\varepsilon)$).
Let $\mathcal{M}$ be an optimization method with the expected rate of convergence (39). Then,

$$\mathbb{E}[T_{\mathcal{M}}(\varepsilon)] \leq \frac{1}{\tau_{\mathcal{M}}}\log\left(\frac{2C_{\mathcal{M}}}{\tau_{\mathcal{M}}\cdot\varepsilon}\right) + 1 = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\log\left(\frac{C_{\mathcal{M}}}{\varepsilon}\right)\right), \tag{41}$$

where we have dropped the dependency in F to simplify the notation.

Proof. We abbreviate $\tau_{\mathcal{M}}$ by τ. Set

$$T_0 = \frac{1}{\tau}\log\left(\frac{1}{1-e^{-\tau}}\frac{C_{\mathcal{M}}}{\varepsilon}\right).$$

For any $k \geq 0$, we have

$$\mathbb{E}[F(x_k) - F^*] \leq C_{\mathcal{M}}(1-\tau)^k \leq C_{\mathcal{M}}\, e^{-k\tau}.$$

By Markov's inequality,

$$\mathbb{P}[F(x_k) - F^* > \varepsilon] = \mathbb{P}[T_{\mathcal{M}}(\varepsilon) > k] \leq \frac{\mathbb{E}[F(x_k) - F^*]}{\varepsilon} \leq \frac{C_{\mathcal{M}}\, e^{-k\tau}}{\varepsilon}. \tag{42}$$

Together with the fact $\mathbb{P} \leq 1$ and $k \geq 0$. We have

$$\mathbb{P}[T_{\mathcal{M}}(\varepsilon) \geq k+1] \leq \min\left\{\frac{C_{\mathcal{M}}}{\varepsilon}e^{-k\tau}, 1\right\}.$$

Therefore,

$$\begin{aligned}
\mathbb{E}[T_{\mathcal{M}}(\varepsilon)] &= \sum_{k=1}^{\infty}\mathbb{P}[T_{\mathcal{M}}(\varepsilon) \geq k] = \sum_{k=1}^{T_0}\mathbb{P}[T_{\mathcal{M}}(\varepsilon) \geq k] + \sum_{k=T_0+1}^{\infty}\mathbb{P}[T_{\mathcal{M}}(\varepsilon) \geq k] \\
&\leq T_0 + \sum_{k=T_0}^{\infty}\frac{C_{\mathcal{M}}}{\varepsilon}e^{-k\tau} = T_0 + \frac{C_{\mathcal{M}}}{\varepsilon}e^{-T_0\tau}\sum_{k=0}^{\infty}e^{-k\tau} \\
&= T_0 + \frac{C_{\mathcal{M}}}{\varepsilon}\frac{e^{-\tau T_0}}{1-e^{-\tau}} = T_0 + 1.
\end{aligned}$$

As simple calculation shows that for any $\tau \in (0,1)$, $\frac{\tau}{2} \leq 1 - e^{-\tau}$ and then

$$\mathbb{E}[T_{\mathcal{M}}(\varepsilon)] \leq T_0 + 1 = \frac{1}{\tau}\log\left(\frac{1}{1-e^{-\tau}}\frac{C_{\mathcal{M}}}{\varepsilon}\right) + 1 \leq \frac{1}{\tau}\log\left(\frac{2C_{\mathcal{M}}}{\tau\varepsilon}\right) + 1.$$

□

Note that the previous lemma mirrors Eq. (36-37) in the proof of Prop. 3.1 in Appendix B. For all optimization methods of interest, the rate $\tau_{\mathcal{M},G_k}$ is independent of k and varies with the parameter κ. We may now compute the iteration-complexity (in expectation) of the accelerated algorithm $\mathcal{A}$—that is, for a given ε, the expected total number of iterations performed by the method $\mathcal{M}$. Let us now fix $\varepsilon > 0$. Calculating the iteration-complexity decomposes into three steps:

1. Find κ that maximizes the ratio $\tau_{\mathcal{M},G_k}/\sqrt{\mu+\kappa}$ for algorithm $\mathcal{M}$ when F is μ-strongly convex. In the non-strongly convex case, we suggest maximizing instead the ratio $\tau_{\mathcal{M},G_k}/\sqrt{L+\kappa}$. Note that the choice of κ is less critical for non-strongly convex problems since it only affects multiplicative constants in the global convergence rate.
2. Compute the upper-bound of the number of outer iterations k_{out} using Theorem 3.1 (for the strongly convex case), or Theorem 3.3 (for the non-strongly convex case), by replacing κ by the optimal value found in step 1.
3. Compute the upper-bound of the expected number of inner iterations
$$\max_{k=1,\dots,k_{\text{out}}} \mathbb{E}[T_{\mathcal{M},G_k}(\varepsilon_k)] \leq k_{\text{in}},$$
by replacing the appropriate quantities in Eq. 41 for algorithm $\mathcal{M}$; for that purpose, the proofs of Propositions 3.2 of 3.4 my be used to upper-bound the ratio $\mathcal{C}_{\mathcal{M},G_k}/\varepsilon_k$, or another dedicated analysis for $\mathcal{M}$ may be required if the constant $\mathcal{C}_{\mathcal{M},G_k}$ does not have the required form $A(G_k(z_0) - G_k^*)$ in (8).

Then, the iteration-complexity (in expectation) denoted Comp. is given by
$$\text{Comp} \leq k_{\text{in}} \times k_{\text{out}} \,. \tag{43}$$

D A Proximal MISO/Finito Algorithm

In this section, we present the algorithm MISO/Finito, and show how to extend it in two ways. First, we propose a proximal version to deal with composite optimization problems, and we analyze its rate of convergence. Second, we show how to remove a large sample condition $n \geq 2L/\mu$, which was necessary for the convergence of the algorithm. The resulting algorithm is a variant of proximal SDCA [25] with a different stepsize and a stopping criterion that does not use duality.

D.1 The Original Algorithm MISO/Finito

MISO/Finito was proposed in [14] and [7] for solving the following smooth unconstrained convex minimization problem
$$\min_{x\in\mathbb{R}^p} \left\{ f(x) \triangleq \frac{1}{n}\sum_{i=1}^n f_i(x) \right\}, \tag{44}$$
where each f_i is differentiable with L-Lipschitz continuous derivatives and μ-strongly convex. At iteration k, the algorithm updates a list of lower bounds d_i^k of the functions f_i, by randomly picking up one index i_k among $\{1,\cdots,n\}$ and performing the following update
$$d_i^k(x) = \begin{cases} f_i(x_{k-1}) + \langle \nabla f_i(x_{k-1}), x - x_{k-1}\rangle + \frac{\mu}{2}\|x - x_{k-1}\|^2 & \text{if} \quad i = i_k \\ d_i^{k-1}(x) & \text{otherwise} \end{cases},$$
which is a lower bound of f_i because of the μ-strong convexity of f_i. Equivalently, one may perform the following updates
$$z_i^k = \begin{cases} x_{k-1} - \frac{1}{\mu}\nabla f_i(x_{k-1}) & \text{if} \quad i = i_k \\ z_i^{k-1} & \text{otherwise} \end{cases},$$
and all functions d_i^k have the form
$$d_i^k(x) = c_i^k + \frac{\mu}{2}\|x - z_i^k\|^2,$$
where c_i^k is a constant. Then, MISO/Finito performs the following minimization to produce the iterate (x_k):
$$x_k = \arg\min_{x\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^n d_i^k(x) = \frac{1}{n}\sum_{i=1}^n z_i^k,$$

which is equivalent to

$$x_k \leftarrow x_{k-1} - \frac{1}{n}\left(z_{i_k}^k - z_{i_k}^{k-1}\right).$$

In many machine learning problems, it is worth remarking that each function $f_i(x)$ has the specific form $f_i(x) = l_i(\langle x, w_i\rangle) + \frac{\mu}{2}\|x\|^2$. In such cases, the vectors z_i^k can be obtained by storing only $O(n)$ scalars.[3] The main convergence result of [14] is that the procedure above converges with a linear rate of convergence of the form (3), with $\tau_{\text{MISO}} = 1/3n$ (also refined in $1/2n$ in [7]), when the large sample size constraint $n \geq 2L/\mu$ is satisfied.

Removing this condition and extending MISO to the composite optimization problem (1) is the purpose of the next section.

D.2 Proximal MISO

We now consider the composite optimization problem below,

$$\min_{x\in\mathbb{R}^p}\left\{F(x) = \frac{1}{n}\sum_{i=1}^n f_i(x) + \psi(x)\right\},$$

where the functions f_i are differentiable with L-Lipschitz derivatives and μ-strongly convex. As in typical composite optimization problems, ψ is convex but not necessarily differentiable. We assume that the proximal operator of ψ can be computed easily. The algorithm needs to be initialized with some lower bounds for the functions f_i:

$$f_i(x) \geq \frac{\mu}{2}\|x - z_i^0\|^2 + c_i^0, \tag{A1}$$

which are guaranteed to exist due to the μ-strong convexity of f_i. For typical machine learning applications, such initialization is easy. For example, logistic regression with ℓ_2-regularization satisfies (A1) with $z_i^0 = 0$ and $c_i^0 = 0$. Then, the MISO-Prox scheme is given in Algorithm 2. Note that if no simple initialization is available, we may consider any initial estimate $\bar{z}_0$ in $\mathbb{R}^p$ and define $z_i^0 = \bar{z}_0 - (1/\mu)\nabla f_i(\bar{z}_0)$, which requires performing one pass over the data.

Then, we remark that under the large sample size condition $n \geq 2L/\mu$, we have $\delta = 1$ and the update of the quantities z_i^k in (45) is the same as in the original MISO/Finito algorithm. As we will see in the convergence analysis, the choice of δ ensures convergence of the algorithm even in the small sample size regime $n < 2L/\mu$.

Relation with Proximal SDCA [25]. The algorithm MISO-Prox is almost identical to variant 5 of proximal SDCA [25], which performs the same updates with $\delta = \mu n/(L + \mu n)$ instead of $\delta = \min(1, \frac{\mu n}{2(L-\mu)})$. It is however not clear that MISO-Prox actually performs dual ascent steps in the sense of SDCA since the proof of convergence of SDCA cannot be directly modified to use the stepsize of proximal MISO and furthermore, the convergence proof of MISO-Prox does not use the concept of duality. Another difference lies in the optimality certificate of the algorithms. Whereas Proximal-SDCA provides a certificate in terms of linear convergence of a duality gap based on Fenchel duality, Proximal-SDCA ensures linear convergence of a gap that relies on strong convexity but not on the Fenchel dual (at least explicitly).

Optimality Certificate and Stopping Criterion. Similar to the original MISO algorithm, Proximal MISO maintains a list (d_i^k) of lower bounds of the functions f_i, which are updated in the following fashion

$$d_i^k(x) = \begin{cases} (1-\delta)d_i^{k-1}(x) + \delta\left(f_i(x_{k-1}) + \langle \nabla f_i(x_{k-1}), x - x_{k-1}\rangle + \frac{\mu}{2}\|x - x_{k-1}\|^2\right) & \text{if } \; i = i_k \\ d_i^{k-1}(x) & \text{otherwise} \end{cases} \tag{46}$$

[3] Note that even though we call this algorithm MISO (or Finito), it was called MISOμ in [14], whereas "MISO" was originally referring to an incremental majorization-minimization procedure that uses upper bounds of the functions f_i instead of lower bounds, which is appropriate for non-convex optimization problems.

Algorithm 2 MISO-Prox: an improved MISO algorithm with proximal support.

input $(z_i^0)_{i=1,\dots,n}$ such that (A1) holds; N (number of iterations);

initialize $\bar{z}_0 = \frac{1}{n}\sum_{i=1}^n z_i^0$ and $x_0 = \operatorname{prox}_{\psi/\mu}[\bar{z}_0]$;

define $\delta = \min\left(1, \frac{\mu n}{2(L-\mu)}\right)$;

for $k = 1, \dots, N$ **do**

randomly pick up an index i_k in $\{1, \dots, n\}$;

update

$$\begin{aligned} z_i^k &= \begin{cases} (1-\delta) z_i^{k-1} + \delta\left(x_{k-1} - \frac{1}{\mu}\nabla f_i(x_{k-1})\right) & \text{if } \ i = i_k \\ z_i^{k-1} & \text{otherwise} \end{cases} \\ \bar{z}_k &= \bar{z}_{k-1} - \frac{1}{n}\left(z_{i_k}^k - z_{i_k}^{k-1}\right) = \frac{1}{n}\sum_{i=1}^n z_i^k \\ x_k &= \operatorname{prox}_{\psi/\mu}[\bar{z}_k]. \end{aligned} \tag{45}$$

end for

output x_N (final estimate).

Then, the following function is a lower bound of the objective F:

$$D_k(x) = \frac{1}{n}\sum_{i=1}^n d_i^k(x) + \psi(x), \tag{47}$$

and the update (45) can be shown to exactly minimize D_k. As a lower bound of F, we have that $D_k(x_k) \leq F^*$ and thus

$$F(x_k) - F^* \leq F(x_k) - D_k(x_k).$$

The quantity $F(x_k) - D_k(x_k)$ can then be interpreted as an optimality gap, and the analysis below will show that it converges linearly to zero. In practice, it also provides a convenient stopping criterion, which yields Algorithm 3.

Algorithm 3 MISO-Prox with stopping criterion.

input $(z_i^0, c_i^0)_{i=1,\dots,n}$ such that (A1) holds; ε (target accuracy);

initialize $\bar{z}_0 = \frac{1}{n}\sum_{i=1}^n z_i^0$ and $c_i'^0 = c_i^0 + \frac{\mu}{2}\|\bar{z}_0\|^2$ for all i in $\{1, \dots, n\}$ and $x_0 = \operatorname{prox}_{\psi/\mu}[\bar{z}_0]$;

Define $\delta = \min\left(1, \frac{\mu n}{2(L-\mu)}\right)$ and $k = 0$;

while $\frac{1}{n}\sum_{i=1}^n f_i(x_k) - c_i'^k + \mu\langle \bar{z}_k, x_k\rangle - \frac{\mu}{2}\|x_k\|^2 > \varepsilon$ **do**

for $l = 1, \dots, n$ **do**

$k \leftarrow k + 1$;

randomly pick up an index i_k in $\{1, \dots, n\}$;

perform the update (45);

update

$$c_i'^k = \begin{cases} (1-\delta) c_i'^{k-1} + \delta\left(f_i(x_{k-1}) - \langle \nabla f_i(x_{k-1}), x_{k-1}\rangle + \frac{\mu}{2}\|x_{k-1}\|^2\right) & \text{if } \ i = i_k \\ c_i'^{k-1} & \text{otherwise} \end{cases}. \tag{48}$$

end for

end while

output x_N (final estimate such that $F(x_N) - F^* \leq \varepsilon$).

To explain the stopping criterion in Algorithm 3, we remark that the functions d_i^k are quadratic and can be written

$$d_i^k(x) = c_i^k + \frac{\mu}{2}\|x - z_i^k\|^2 = c_i'^k - \mu\langle x, z_i^k\rangle + \frac{\mu}{2}\|x\|^2, \tag{49}$$

where the c_i^k's are some constants and $c_i'^k = c_i^k + \frac{\mu}{2}\|z_i^k\|^2$. Equation (48) shows how to update recursively these constants $c_i'^k$, and finally

$$D_k(x_k) = \left(\frac{1}{n}\sum_{i=1}^n c_i'^k\right) - \mu\langle x_k, \bar{z}_k\rangle + \frac{\mu}{2}\|x_k\|^2 + \psi(x_k),$$

and

$$F(x_k) - D_k(x_k) = \left(\frac{1}{n}\sum_{i=1}^n f_i(x_k) - c_i'^k\right) + \mu\langle x_k, \bar{z}_k\rangle - \frac{\mu}{2}\|x_k\|^2,$$

which justifies the stopping criterion. Since computing $F(x_k)$ requires scanning all the data points, the criterion is only computed every n iterations.

Convergence Analysis. The convergence of MISO-Prox is guaranteed by Theorem 4.1 from the main part of paper. Before we prove this theorem, we note that this rate is slightly better than the one proven in MISO [14], which converges as $(1 - \frac{1}{3n})^k$. We start by recalling a classical lemma that provides useful inequalities. Its proof may be found in [17].

Lemma D.1 (Classical Quadratic Upper and Lower Bounds).
For any function $g : \mathbb{R}^p \to \mathbb{R}$ which is μ-strongly convex and differentiable with L-Lipschitz derivatives, we have for all x, y in $\mathbb{R}^p$,

$$\frac{\mu}{2}\|x - y\|^2 \leq g(x) - g(y) + \langle \nabla g(y), x - y\rangle \leq \frac{L}{2}\|x - y\|^2.$$

To start the proof, we need a sequence of upper and lower bounds involving the functions D_k and D_{k-1}. The first one is given in the next lemma

Lemma D.2 (Lower Bound on D_k).
For all $k \geq 1$ and x in $\mathbb{R}^p$,

$$D_k(x) \geq D_{k-1}(x) - \frac{\delta(L-\mu)}{2n}\|x - x_{k-1}\|^2, \quad \forall x \in \mathbb{R}^p. \tag{50}$$

Proof. For any $i \in \{1, \ldots, n\}$, f_i satisfies the assumptions of Lemma D.1, and we have for all $k \geq 0$, x in $\mathbb{R}^p$, and for $i = i_k$,

$$\begin{aligned} d_i^k(x) &= (1-\delta)d_i^{k-1}(x) + \delta[f_i(x_{k-1}) + \langle \nabla f_i(x_{k-1}), x - x_{k-1}\rangle + \frac{\mu}{2}\|x - x_{k-1}\|^2] \\ &\geq (1-\delta)d_i^{k-1}(x) + \delta f_i(x) - \frac{\delta(L-\mu)}{2}\|x - x_{k-1}\|^2 \\ &\geq d_i^{k-1}(x) - \frac{\delta(L-\mu)}{2}\|x - x_{k-1}\|^2, \end{aligned}$$

where the definition of d_i^k is given in (46). The first inequality uses Lemma D.1, and the last one uses the inequality $f_i \geq d_i^{k-1}$. From this inequality, we can obtain (50) by simply using $D_k(x) = \sum_{i=1}^n d_i^k(x) + \psi(x) = D_{k-1}(x) + \frac{1}{n}\left(d_{i_k}^k(x) - d_{i_k}^{k-1}(x)\right)$. □

Next, we prove the following lemma to compare D_k and D_{k-1}.

Lemma D.3 (Relation between D_k and D_{k-1}).
For all $k \geq 0$, for all x and y in $\mathbb{R}^p$,

$$D_k(x) - D_k(y) = D_{k-1}(x) - D_{k-1}(y) - \mu\langle \bar{z}_k - \bar{z}_{k-1}, x - y\rangle. \tag{51}$$

Proof. Remember that the functions d_i^k are quadratic and have the form (49), that D_k is defined in (47), and that $\bar{z}_k$ minimizes $\frac{1}{n}\sum_{i=1}^n d_i^k$. Then, there exists a constant A_k such that

$$D_k(x) = A_k + \frac{\mu}{2}\|x - \bar{z}_k\|^2 + \psi(x).$$

This gives

$$D_k(x) - D_k(y) = \frac{\mu}{2}\|x - \bar{z}_k\|^2 - \frac{\mu}{2}\|y - \bar{z}_k\|^2 + \psi(x) - \psi(y). \tag{52}$$

Similarly,

$$D_{k-1}(x) - D_{k-1}(y) = \frac{\mu}{2}\|x - \bar{z}_{k-1}\|^2 - \frac{\mu}{2}\|y - \bar{z}_{k-1}\|^2 + \psi(x) - \psi(y). \tag{53}$$

Subtracting (52) and (53) gives (51). □

Then, we are able to control the value of $D_k(x_{k-1})$ in the next lemma.

Lemma D.4 (Controlling the value $D_k(x_{k-1})$).
For any $k \geq 1$,

$$D_k(x_{k-1}) - D_k(x_k) \leq \frac{\mu}{2}\|\bar{z}_k - \bar{z}_{k-1}\|^2. \tag{54}$$

Proof. Using Lemma D.3 with $x = x_{k-1}$ and $y = x_k$ yields

$$D_k(x_{k-1}) - D_k(x_k) = D_{k-1}(x_{k-1}) - D_{k-1}(x_k) - \mu\langle \bar{z}_k - \bar{z}_{k-1}, x_{k-1} - x_k\rangle.$$

Moreover x_{k-1} is the minimum of D_{k-1} which is μ-strongly convex. Thus,

$$D_{k-1}(x_{k-1}) + \frac{\mu}{2}\|x_k - x_{k-1}\|^2 \leq D_{k-1}(x_k).$$

Adding the two previous inequalities gives the first inequality below

$$D_k(x_{k-1}) - D_k(x_k) \leq -\frac{\mu}{2}\|x_k - x_{k-1}\|^2 - \mu\langle \bar{z}_k - \bar{z}_{k-1}, x_{k-1} - x_k\rangle \leq \frac{\mu}{2}\|\bar{z}_k - \bar{z}_{k-1}\|^2,$$

and the last one comes from the basic inequality $\frac{1}{2}\|a\|^2 + \langle a, b\rangle + \frac{1}{2}\|b\|^2 \geq 0$. □

We have now all the inequalities in hand to prove Theorem 4.1.

Proof of Theorem 4.1.
We start by giving a lower bound of $D_k(x_{k-1}) - D_{k-1}(x_{k-1})$.
Take $x = x_{k-1}$ in (51). Then, for all y in $\mathbb{R}^p$,

$$\begin{aligned} D_k(x_{k-1}) - D_{k-1}(x_{k-1}) &= D_k(y) - D_{k-1}(y) + \mu\langle \bar{z}_k - \bar{z}_{k-1}, y - x_{k-1}\rangle \\ &\text{by (50)} \geq -\frac{\delta(L-\mu)}{2n}\|y - x_{k-1}\|^2 + \mu\langle \bar{z}_k - \bar{z}_{k-1}, y - x_{k-1}\rangle \end{aligned}$$

Choose y that maximizes the above quadratic function, i.e.

$$y = x_{k-1} + \frac{n\mu}{\delta(L-\mu)}(\bar{z}_k - \bar{z}_{k-1}),$$

and then

$$\begin{aligned} D_k(x_{k-1}) - D_{k-1}(x_{k-1}) &\geq \frac{n\mu^2}{2\delta(L-\mu)}\|\bar{z}_k - \bar{z}_{k-1}\|^2 \\ &\text{by (54)} \geq \frac{n\mu}{\delta(L-\mu)}\left[D_k(x_{k-1}) - D_k(x_k)\right]. \end{aligned} \tag{55}$$

Then, we start introducing expected values.
By construction

$$D_k(x_{k-1}) = D_{k-1}(x_{k-1}) + \frac{\delta}{n}(f_{i_k}(x_{k-1}) - d_{i_k}^{k-1}(x_{k-1})).$$

After taking expectation, we obtain the relation

$$\mathbb{E}[D_k(x_{k-1})] = \left(1 - \frac{\delta}{n}\right)\mathbb{E}[D_{k-1}(x_{k-1})] + \frac{\delta}{n}\mathbb{E}[F(x_{k-1})]. \tag{56}$$

We now introduce an important quantity

$$\tau = \left(1 - \frac{\delta(L-\mu)}{n\mu}\right)\frac{\delta}{n},$$

and combine (55) with (56) to obtain

$$\tau\mathbb{E}[F(x_{k-1})] - \mathbb{E}[D_k(x_k)] \leq -(1-\tau)\mathbb{E}[D_{k-1}(x_{k-1})].$$

We reformulate this relation as

$$\tau\left(\mathbb{E}[F(x_{k-1})] - F^*\right) + \left(F^* - \mathbb{E}[D_k(x_k)]\right) \leq (1-\tau)\left(F^* - \mathbb{E}[D_{k-1}(x_{k-1})]\right). \tag{57}$$

On the one hand, since $F(x_{k-1}) \geq F^*$, we have

$$F^* - \mathbb{E}[D_k(x_k)] \leq (1-\tau)\left(F^* - \mathbb{E}[D_{k-1}(x_{k-1})]\right).$$

This is true for any $k \geq 1$, as a result

$$F^* - \mathbb{E}[D_k(x_k)] \leq (1-\tau)^k\left(F^* - D_0(x_0)\right). \tag{58}$$

On the other hand, since $F^* \geq D_k(x_k)$, then

$$\tau\left(\mathbb{E}[F(x_{k-1})] - F^*\right) \leq (1-\tau)\left(F^* - \mathbb{E}[D_{k-1}(x_{k-1})]\right) \leq (1-\tau)^k\left(F^* - D_0(x_0)\right),$$

which gives us the relation (14) of the theorem. We conclude giving the choice of δ. We choose it to maximize the rate of convergence, which turns to maximize τ. This is a quadratic function, which is maximized at $\delta = \frac{n\mu}{2(L-\mu)}$. However, by definition $\delta \leq 1$. Therefore, the optimal choice of δ is given by

$$\delta = \min\left\{1, \frac{n\mu}{2(L-\mu)}\right\}.$$

Note now that

1. When $\frac{n\mu}{2(L-\mu)} \leq 1$, we have $\delta = \frac{n\mu}{2(L-\mu)}$ and $\tau = \frac{\mu}{4(L-\mu)}$.
2. When $1 \leq \frac{n\mu}{2(L-\mu)}$, we have $\delta = 1$ and $\tau = \frac{1}{n} - \frac{L-\mu}{n^2\mu} \geq \frac{1}{2n}$.

Therefore, $\tau \geq \min\left(\frac{1}{2n}, \frac{\mu}{4(L-\mu)}\right)$, which concludes the first part of the theorem.

To prove the second part, we use (58) and (14), which gives

$$\begin{aligned}
\mathbb{E}[F(x_k) - D_k(x_k)] &= \mathbb{E}[F(x_k)] - F^* + F^* - \mathbb{E}[D_k(x_k)] \\
&\leq \frac{1}{\tau}(1-\tau)^{k+1}(F^* - D_0(x_0) + (1-\tau)^k(F^* - D_0(x_0)) \\
&= \frac{1}{\tau}(1-\tau)^k(F^* - D_0(x_0)).
\end{aligned}$$

□

D.3 Accelerating MISO-Prox

The convergence rate of MISO (or also SDCA) requires a special handling since it does not satisfy exactly the condition (8) from Proposition 3.2. The rate of convergence is linear, but with a constant proportional to $F^* - D_0(x_0)$ instead of $F(x_0) - F^*$ for many classical gradient-based approaches.

To achieve acceleration, we show in this section how to obtain similar guarantees as Proposition 3.2 and 3.4—that is, how to solve efficiently the subproblems (5). This essentially requires the right initialization each time MISO-Prox is called. By initialization, we mean initializing the variables z_i^0.

Assume that MISO-Prox is used to obtain x_{k-1} from Algorithm 1 with $G_{k-1}(x_{k-1}) - G_k^* \leq \varepsilon_{k-1}$, and that one wishes to use MISO-Prox again on G_k to compute x_k. Then, let us call D' the lower-bound of G_{k-1} produced by MISO-Prox when computing x_{k-1} such that

$$x_{k-1} = \operatorname*{arg\,min}_{x\in\mathbb{R}^p}\left\{D'(x) = \frac{1}{n}\sum_{i=1}^n d_i'(x) + \psi(x)\right\},$$

with

$$d_i'(x) = \frac{\mu+\kappa}{2}\|x - z_i'\|^2 + c_i'.$$

Note that we do not index these quantities with $k-1$ or k for the sake of simplicity. The convergence of MISO-Prox may ensure that not only do we have $G_{k-1}(x_{k-1}) - G_k^* \leq \varepsilon_{k-1}$, but in fact we have the stronger condition $G_{k-1}(x_{k-1}) - D'(x_{k-1}) \leq \varepsilon_{k-1}$. Remember now that

$$G_k(x) = G_{k-1}(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 - \frac{\kappa}{2}\|x - y_{k-2}\|^2,$$

and that D' is a lower-bound of G_{k-1}. Then, we may set for all i in $\{1, \dots, n\}$

$$d_i^0(x) = d_i'(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 - \frac{\kappa}{2}\|x - y_{k-2}\|^2,$$

which is equivalent to initializing the new instance of MISO-Prox with

$$z_i^0 = z_i' + \frac{\kappa}{\kappa+\mu}(y_{k-1} - y_{k-2}),$$

and by choosing appropriate quantities c_i^0. Then, the following function is a lower bound of G_k

$$D_0(x) = \frac{1}{n}\sum_{i=1}^n d_i^0(x) + \psi(x).$$

and the new instance of MISO-Prox to minimize G_k and compute x_k will produce iterates, whose first point, which we call x^0, minimizes D_0. This leads to the relation

$$x^0 = \operatorname{prox}_{\psi/(\kappa+\mu)}\left[\bar{z}^0\right] = \operatorname{prox}_{\psi/(\kappa+\mu)}\left[\bar{z}' + \frac{\kappa}{\kappa+\mu}(y_{k-1} - y_{k-2})\right],$$

where we use the notation $\bar{z}^0 = \frac{1}{n}\sum_{i=1}^n z_i^0$ and $\bar{z}' = \frac{1}{n}\sum_{i=1}^n z_i'$ as in Algorithm 2.

Then, it remains to show that the quantity $G_k^* - D_0(x^0)$ is upper bounded in a similar fashion as $G_k(x_{k-1}) - G_k^*$ in Propositions 3.2 and 3.4 to obtain a similar result for MISO-Prox and control the number of inner-iterations. This is indeed the case, as stated in the next lemma.

Lemma D.5 (Controlling $G_k(x_{k-1}) - G_k^*$ for MISO-Prox).
When initializing MISO-Prox as described above, we have

$$G_k^* - D_0(x^0) \leq \varepsilon_{k-1} + \frac{\kappa^2}{2(\kappa+\mu)}\|y_{k-1} - y_{k-2}\|^2.$$

Proof. By strong convexity, we have

$$D_0(x^0) + \frac{\kappa}{2}\|x^0 - y_{k-2}\|^2 - \frac{\kappa}{2}\|x^0 - y_{k-1}\|^2 = D_0'(x^0) \geq D_0'(x_{k-1}) + \frac{\kappa+\mu}{2}\|x^0 - x_{k-1}\|^2.$$

Consequently,

$$\begin{aligned}
D_0(x^0) &\geq D'(x_{k-1}) - \frac{\kappa}{2}\|x^0 - y_{k-2}\|^2 + \frac{\kappa}{2}\|x^0 - y_{k-1}\|^2 + \frac{\kappa+\mu}{2}\|x^0 - x_{k-1}\|^2 \\
&= D_0(x_{k-1}) + \frac{\kappa}{2}\|x_{k-1} - y_{k-2}\|^2 - \frac{\kappa}{2}\|x_{k-1} - y_{k-1}\|^2 - \frac{\kappa}{2}\|x^0 - y_{k-2}\|^2 + \frac{\kappa}{2}\|x^0 - y_{k-1}\|^2 \\
&\quad + \frac{\kappa+\mu}{2}\|x^0 - x_{k-1}\|^2 \\
&= D_0(x_{k-1}) - \kappa\langle x^0 - x_{k-1}, y_{k-1} - y_{k-2}\rangle + \frac{\kappa+\mu}{2}\|x^0 - x_{k-1}\|^2 \\
&\geq D_0(x_{k-1}) - \frac{\kappa^2}{2(\kappa+\mu)}\|y_{k-1} - y_{k-2}\|^2,
\end{aligned}$$

where the last inequality is using a simple relation $\frac{1}{2}\|a\|^2 + 2\langle a, b\rangle + \frac{1}{2}\|b\|^2 \geq 0$. As a result,

$$\begin{aligned}
G_k^* - D_0(x^0) &\leq G_k^* - D_0(x_{k-1}) + \frac{\kappa^2}{2(\kappa+\mu)}\|y_{k-1} - y_{k-2}\|^2 \\
&\leq G_k(x_{k-1}) - D_0(x_{k-1}) + \frac{\kappa^2}{2(\kappa+\mu)}\|y_{k-1} - y_{k-2}\|^2 \\
&= G_{k-1}(x_{k-1}) - D'(x_{k-1}) + \frac{\kappa^2}{2(\kappa+\mu)}\|y_{k-1} - y_{k-2}\|^2 \\
&\leq \varepsilon_{k-1} + \frac{\kappa^2}{2(\kappa+\mu)}\|y_{k-1} - y_{k-2}\|^2
\end{aligned}$$

□

We remark that this bound is half of the bound shown in (32). Hence, a similar argument gives the bound on the number of inner iterations. We may finally compute the iteration-complexity of accelerated MISO-Prox.

Proposition D.6 (Iteration-Complexity of Accelerated MISO-Prox)**.**
When F is μ-strongly convex, the accelerated MISO-Prox algorithm achieves the accuracy ε with an expected number of iteration upper bounded by

$$O\left(\min\left\{\frac{L}{\mu}, \sqrt{\frac{nL}{\mu}}\right\}\log\left(\frac{1}{\varepsilon}\right)\log\left(\frac{L}{\mu}\right)\right).$$

Proof. When $n > 2(L-\mu)/\mu$, there is no acceleration. The optimal value for κ is zero, and we may use Theorem 4.1 and Lemma C.1 to obtain the complexity

$$O\left(\frac{L}{\mu}\log\left(\frac{L}{\mu}\frac{F(x_0)-D_0(x_0)}{\varepsilon}\right)\right).$$

When $n < 2(L-\mu)/\mu$, there is an acceleration, with $\kappa = 2(L-\mu)/\mu \ -\mu$. Let us compute the global complexity using the "template" presented in Appendix C. The number of outer iteration is given by

$$k_{\text{out}} = O\left(\sqrt{\frac{L}{n\mu}}\log\left(\frac{F(x_0)-F^*}{\varepsilon}\right)\right).$$

At each inner iteration, we initialize with the value x^0 described above, and we use Lemma D.5:

$$G_k^* - D_0(x^0) \leq \varepsilon_{k-1} + \frac{\kappa}{2}\|y_{k-1}-y_{k-2}\|^2.$$

Then,

$$\frac{G_k^* - D_0(x^0)}{\varepsilon_k} \leq \frac{R}{2},$$

where

$$R = \frac{2}{1-\rho} + \frac{2592\kappa}{\mu(1-\rho)^2(\sqrt{q}-\rho)^2} = O\left(\left(\frac{L}{n\mu}\right)^2\right).$$

With Miso-Prox, with have $\tau_{G_k} = \frac{1}{2n}$, thus the expected number of inner iteration is given by Lemma C.1:

$$k_{\text{in}} = O(n\log(n^2R)) = O\left(n\log\left(\frac{L}{\mu}\right)\right).$$

As a result,

$$\text{Comp} = O\left(\sqrt{\frac{nL}{\mu}}\log\left(\frac{F(x_0)-F^*}{\varepsilon}\right)\log\left(\frac{L}{\mu}\right)\right).$$

To conclude, the complexity of the accelerated algorithm is given by

$$O\left(\min\left\{\frac{L}{\mu}, \sqrt{\frac{nL}{\mu}}\right\}\log\left(\frac{1}{\varepsilon}\right)\log\left(\frac{L}{\mu}\right)\right).$$

□

E Implementation Details of Experiments

In the experimental section, we compare the performance with and without acceleration for three algorithms SAG, SAGA and MISO-Prox on l_2-logistic regression problem. In this part, we clarify some details about the implementation of the experiments.

Firstly, we normalize the observed data before running the regression. Then we apply Catalyst using parameters according to the theoretical settings. Standard analysis of the logistic function shows that the Lipschitz gradient parameter L is $1/4$ and strongly convex parameter $\mu = 0$ when there is no

regularization. Adding properly a l_2 term generates the strongly-convex regimes. Several parameters need to be fixed at the beginning stage. The parameter κ is set to its optimal value suggested by theory, which only depends on n, μ and L. More precisely, κ writes as $\kappa = a(L-\mu)/(n+b) - \mu$, with $(a,b) = (2,-2)$ for SAG, $(a,b) = (1/2,1/2)$ for SAGA and $(a,b) = (1,1)$ for MISO-Prox. The parameter α_0 is initialized as the positive solution of $x^2 + (1-q)x - 1 = 0$ where $q = \sqrt{\mu/(\mu+\kappa)}$. Furthermore, since the objective function is always positive, $F(x_0) - F^*$ can be upper bounded by $F(x_0)$ which allow us to set the $\varepsilon_k = (2/9)F(x_0)(1-\rho)^k$ in the strongly convex case and $\varepsilon_k = 2F(x_0)/9(k+2)^{4+\eta}$ in the non-strongly convex case. Finally, we set the free parameter in the expression of ε_k as follows. We simply set $\rho = 0.9\sqrt{q}$ in the strongly convex case and $\eta = 0.1$ in the non strongly convex case.

To solve the subproblem at each iteration, the step-sizes parameter for SAG, SAGA and MISO are set to the values suggested by theory, which only depend on μ, L and κ. All of the methods we compare store n gradients evaluated at previous iterates of the algorithm. For MISO, the convergence analysis in Appendix D leads to the initialization $x_{k-1} + \frac{\kappa}{\mu+\kappa}(y_{k-1} - y_{k-2})$ that moves x_{k-1} closer to y_{k-1} and further away from y_{k-2}. We found that using this initial point for SAGA was giving slightly better results than x_{k-1}.