Noname manuscript No.
(will be inserted by the editor)

Asynchronous Parallel Algorithms for Nonconvex Big-Data Optimization
Part II: Complexity and Numerical Results

Loris Cannelli · Francisco Facchinei · Vyacheslav Kungurtsev · Gesualdo Scutari

January 18, 2017

Abstract We present complexity and numerical results for a new asynchronous parallel algorithmic method for the minimization of the sum of a smooth nonconvex function and a convex nonsmooth regularizer, subject to both convex and nonconvex constraints. The proposed method hinges on successive convex approximation techniques and a novel probabilistic model that captures key elements of modern computational architectures and asynchronous implementations in a more faithful way than state-of-the-art models. In the companion paper [3] we provided a detailed description on the probabilistic model and gave convergence results for a diminishing stepsize version of our method. Here, we provide theoretical complexity results for a fixed stepsize version of the method and report extensive numerical comparisons on both convex and nonconvex problems demonstrating the efficiency of our approach.

Keywords Asynchronous algorithms · big-data · convergence rate · nonconvex constrained optimization.

1 Introduction

The rise of the big data challenge has created a strong demand for highly parallelizable algorithms solving huge optimization problems quickly and reliably. Recent research on algorithms incorporating asynchrony has given promising theoretical and computational results but, as discussed thoroughly in the

Loris Cannelli and Gesualdo Scutari, School of Industrial Engineering, Purdue University, USA E-mail: <lcannelli, gscutari>@purdue.edu. · Francisco Facchinei, Department of Computer, Control, and Management Engineering Antonio Ruberti, University of Rome La Sapienza, Roma, Italy E-mail: francisco.facchinei@uniroma1.it. · Vyacheslav Kungurtsev Dept. of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech E-mail: vyacheslav.kungurtsev@fel.cvut.cz.

The work of Cannelli and Scutari was supported by the USA National Science Foundation (NSF) under Grants CIF 1564044, CCF 1632599 and CAREER Award No. 1555850, and the Office of Naval Research (ONR) Grant N00014-16-1-2244.

companion paper [3], to which we refer the reader for a comprehensive bibliographical review, there is still room for huge improvements. In [3] we presented a novel probabilistic model we believe gives a more accurate description of asynchrony as encountered in modern computational architectures and proposed a new diminishing stepsize minimization method for which we proved convergence to stationary points. Here, we complete the analysis by proving complexity results for a fixed stepsize version of the method proposed in [3] and by reporting numerical test results comparing our algorithm to state-of-the-art methods on both convex and nonconvex problems.

We consider the minimization of a smooth (possibly) *nonconvex* function f and a nonsmooth block-separable convex one G subject to convex constraints $\mathcal{X}$ and local nonconvex ones $c_{j_i}(\mathbf{x}_i) \leq 0$. The formulation reads

$$\begin{aligned} \min_{\mathbf{x}=(\mathbf{x}_1,\ldots,\mathbf{x}_N)} \quad & F(\mathbf{x}) \triangleq f(\mathbf{x}) + G(\mathbf{x}) \\ \text{subject to} \quad & \left.\begin{aligned} & \mathbf{x}_i \in \mathcal{X}_i, \quad i = 1,\ldots,N \\ & c_{j_1}(\mathbf{x}_1) \leq 0, \quad j_1 = 1,\ldots,m_1 \\ & \vdots \\ & c_{j_N}(\mathbf{x}_N) \leq 0, \quad j_N = m_{N-1}+1,\ldots,m_N. \end{aligned}\right\} \triangleq \mathcal{K} \end{aligned} \tag{1}$$

We denote by $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_N$ the Cartesian product of the lower dimensional closed, convex sets $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$.We consider the presence of $m_i - m_{i-1}$ nonconvex local constraints $c_{j_i} : \mathcal{X}_i \to \mathbb{R}$ with $j_i = m_{i-1},\ldots,m_i$, for each block of variables $\mathbf{x}_i$, with $m_i \geq 0$, for $i \in \mathcal{N} \triangleq \{1,\ldots,N\}$ and $m_0 = 0$. We denote by $\mathcal{K}_i$ the set $\mathcal{K}_i \triangleq \{\mathbf{x}_i \in \mathcal{X}_i : c_{j_i}(\mathbf{x}_i) \leq 0, \quad j_i = m_{i-1}+1,\ldots,m_i\}$. The function f is smooth (not necessarily convex or separable) and G is convex, separable, and possibly nondifferentiable.

Assumption A We make the following blanket assumptions.

(A1) Each $\mathcal{X}_i$ is nonempty, closed and convex;

(A2) f is C^1 on an open set containing $\mathcal{X}$;

(A3) $\nabla_{\mathbf{x}_i} f$ is Lipschitz continuous on $\mathcal{X}_i$ with a Lipschitz constant L_f which is independent of i;

(A4) $G(\mathbf{x}) \triangleq \sum_i g_i(\mathbf{x}_i)$, and each $g_i(\mathbf{x}_i)$ is continuous, convex, and Lipschitz continuous with constant L_g on $\mathcal{X}_i$ (but possibly nondifferentiable);

(A5) $\mathcal{K}$ is compact;

(A6) Each $c_{j_i} : \mathcal{X}_i \to \mathbb{R}$ is continuously differentiable on $\mathcal{X}_i$, for all $i \in \mathcal{N}$ and $j_i \in \{m_{i-1}+1,\ldots,m_i\}$, $m_i \geq 0$ and $m_0 = 0$.

In this paper we study the complexity of a version of the algorithm proposed in the companion paper [3] that uses a fixed stepsize. We prove that the expected value of an appropriate stationarity measure becomes smaller than ϵ after a number of iterations proportional to $1/\epsilon$. We also show that if the stepsize is small enough, then a linear speedup can be expected as the number of cores increases. Although comparisons are difficult both because our probabilistic model is different from and more accurate than most usually used in the literature, our complexity results seem comparable to the ones in the literature for both convex problems (see for example [8, 13, 14, 16, 17]), and nonconvex problems ([10] and [4,5]). In [10] complexity is analyzed for a stochastic gradient methods for *unconstrained,* smooth nonconvex problems. It is shown that a number of iterations proportional to $1/\epsilon$ is needed in order to drive the expected value of the gradient below ϵ. Similar results are proved [4,5] for nonconvex, constrained problems. However, recall that in [3] it was observed that the probabilistic models used in [4, 5, 10] or, from another point of view, the implicit assumptions made in these papers, are problematic. Furthermore it should also be observed that the methods for nonconvex, constrained problems in [4, 5] require the global solution of nonconvex subproblems, making these methods of uncertain practical use, in general. Therefore, the complexity results presented in this paper are of novel value and represent an advancement in the state of the art of asynchronous algorithms for large-scale (nonconvex) optimization problems.

We also provide extensive numerical results for the diminishing stepsize method proposed in the companion paper [3]. We compare the performance and the speedup of this method with the most advanced state of the art algorithms for both convex and nonconvex problems. The results show that our method compares favorably to existing asynchronous methods.

The paper is organized as follows. In the next section, after briefly summarizing some necessary elements from [3], we describe the algorithm and in Section 3 we give the complexity results. In Section 4 we report the numerical results and finally draw some conclusions in Section 5. All proofs are given in the Appendix.

2 Algorithm

In this section we describe the asynchronous model and algorithm as proposed in the companion paper [3, Sec. 4]. The difference with [3] is that here, in order to study the convergence rate of the algorithm, we enforce a fixed stepsize (rather than using a diminishing stepsize).

We use a global index k to count iterations: whenever a core updates a block-component of the current $\mathbf{x}$, a new iteration $k \to k+1$ is triggered (this iteration counter is not required by the cores themselves to compute the updates). Therefore, at each iteration k, there is a core that updates in an independent and asynchronous fashion a block-component $\mathbf{x}_{i^k}$ of $\mathbf{x}^k$ randomly chosen, thus generating the vector $\mathbf{x}^{k+1}$. Hence, $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$ only differ in the i^k-th component. To update block i^k, a core generally does not have access to the global vector $\mathbf{x}^k$, but will instead use the possibly out-of-sync, delayed coordinates $\tilde{\mathbf{x}}^k = \mathbf{x}^{k-\mathbf{d}^k} \triangleq (\mathbf{x}_1^{k-d_1^k}, \ldots, \mathbf{x}_N^{k-d_N^k})$, where d_i^k are some nonnegative integer numbers. Given $\tilde{\mathbf{x}}^k$, to update the i^k-th component, the following strongly convex problems is first solved

$$\hat{\mathbf{x}}_i^k(\tilde{\mathbf{x}}^k) = \underset{\mathbf{x}_{i^k}\in\mathcal{K}_{i^k}(\tilde{\mathbf{x}}^k_{i^k})}{\arg\min}\ \tilde{F}_{i^k}(\mathbf{x}_{i^k};\tilde{\mathbf{x}}^k) \triangleq \tilde{f}_{i^k}(\mathbf{x}_{i^k};\tilde{\mathbf{x}}^k) + g_{i^k}(\mathbf{x}_{i^k}), \tag{2}$$

where $\tilde{f}_i : \mathcal{X}_i \times \mathcal{K} \to \mathbb{R}$ and $\mathcal{K}_i(\bullet)$, with $i = 1, \dots, N$, represent convex approximations of f and $\mathcal{K}_i$, respectively, defined according to the rules listed below. Then block i^k is updated according to

$$\mathbf{x}_{i^k}^{k+1} = \mathbf{x}_{i^k}^{k} + \gamma(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^{k}); \tag{3}$$

where γ is a positive constant.

We require the following assumptions on the surrogate function $\tilde{f}_i$.

Assumption B (On the surrogate functions). Each $\tilde{f}_i$ is a function continuously differentiable with respect to the first argument such that:

(B1) $\tilde{f}_i(\bullet; \mathbf{y})$ is uniformly strongly convex on $\mathcal{X}_i$ for all $\mathbf{y} \in \mathcal{K}$ with a strong convexity constant $c_{\tilde{f}}$ which is independent of i and k;

(B2) $\nabla \tilde{f}_i(\mathbf{y}_i; \mathbf{y}) = \nabla_{\mathbf{y}_i} f(\mathbf{y})$, for all $\mathbf{y} \in \mathcal{K}$;

(B3) $\nabla \tilde{f}_i(\mathbf{y}_i; \bullet)$ is Lipschitz continuous on $\mathcal{K}$, for all $\mathbf{y}_i \in \mathcal{X}_i$, with a Lipschitz constant L_B which is independent of i and k;

(B4) $\nabla \tilde{f}_i(\bullet; \mathbf{y})$ is Lipschitz continuous on $\mathcal{X}_i$, for all $\mathbf{y} \in \mathcal{K}$, with a Lipschitz constant L_E which is independent of i and k.

$\mathcal{K}_i(\mathbf{y}_i)$ is a convex approximation of $\mathcal{K}_i$ defined by

$$\mathcal{K}_i(\mathbf{y}_i) \triangleq \begin{cases} \tilde{c}_{j_i}(\mathbf{x}_i; \mathbf{y}_i) \leq 0, & j_i = m_{i-1} + 1, \dots, m_i \\ \mathbf{x}_i \in \mathcal{X}_i & \end{cases}, \quad i = 1, \dots, N, \tag{4}$$

where $\tilde{c}_{j_i} : \mathcal{X}_i \times \mathcal{K}_i \to \mathbb{R}$ is required to satisfy the following assumptions.

Assumption C (On $\tilde{c}_{j_i}$'s). For each $i \in \mathcal{N}$ and $j_i \in \{m_{i-1} + 1, \dots, m_i\}$, it holds:

(C1) $\tilde{c}_{j_i}(\bullet; \mathbf{y})$ is convex on $\mathcal{X}_i$ for all $\mathbf{y} \in \mathcal{K}_i$;

(C2) $\tilde{c}_{j_i}(\mathbf{y}; \mathbf{y}) = c_{j_i}(\mathbf{y})$, for all $\mathbf{y} \in \mathcal{K}_i$;

(C3) $c_{j_i}(\mathbf{z}) \leq \tilde{c}_{j_i}(\mathbf{z}; \mathbf{y})$ for all $\mathbf{z} \in \mathcal{X}_i$ and $\mathbf{y} \in \mathcal{K}_i$;

(C4) $\tilde{c}_{j_i}(\bullet; \bullet)$ is continuous on $\mathcal{X}_i \times \mathcal{K}_i$;

(C5) $\nabla c_{j_i}(\mathbf{y}) = \nabla_1 \tilde{c}_{j_i}(\mathbf{y}; \mathbf{y})$, for all $\mathbf{y} \in \mathcal{K}_i$;

(C6) $\nabla \tilde{c}_{j_i}(\bullet; \bullet)$ is continuous on $\mathcal{X}_i \times \mathcal{K}_i$;

(C7) Each $\tilde{c}_{j_i}(\bullet; \bullet)$ is Lipschitz continuous on $\mathcal{X}_i \times \mathcal{K}_i$;

where $\nabla \tilde{c}_{j_i}$ is the partial gradient of $\tilde{c}_{j_i}$ with respect to the first argument.

The randomness associated with the block selection procedure and the delayed information being used to compute the solution to the subproblems is described in our model with the index-delay pair $(i, \mathbf{d})$ used at each iteration k to update $\mathbf{x}^k$ being a realization of a random vector $\underline{\boldsymbol{\omega}}^k \triangleq (\underline{i}^k, \underline{\mathbf{d}}^k)$, taking values on $\mathcal{N} \times \mathcal{D}$ with some probability $p_{i,\mathbf{d}}^k \triangleq \mathbb{P}((\underline{i}^k, \underline{\mathbf{d}}^k) = (i, \mathbf{d}))$, where $\mathcal{D}$ is the set of all possible delay vectors. Since each delay $d_i \leq \delta$ (see Assumption C below), $\mathcal{D}$ is the set of all possible N-length vectors whose components are integers between 0 and δ. More formally, let Ω be the sample space of all the sequences $\{(i^k, \mathbf{d}^k)\}_{k \in \mathbb{N}_+}$, and let us define the discrete-time, discrete-value stochastic process $\underline{\boldsymbol{\omega}}$, where $\{\underline{\boldsymbol{\omega}}^k(\omega)\}_{k \in \mathbb{N}_+}$ is a sample path of the process. The

k-th entry $\underline{\boldsymbol{\omega}}^k(\omega)$ of $\underline{\boldsymbol{\omega}}(\omega)$—the k-th element of the sequence ω—is a realization of the random vector $\underline{\boldsymbol{\omega}}^k = (\underline{i}^k, \underline{\mathbf{d}}^k) : \Omega \mapsto \mathcal{N} \times \mathcal{D}$.

The stochastic process $\underline{\boldsymbol{\omega}}$ is fully defined once the joint finite-dimensional probability mass functions $p_{\underline{\boldsymbol{\omega}}^{0:k}}(\boldsymbol{\omega}^{0:k}) \triangleq \mathbb{P}(\underline{\boldsymbol{\omega}}^{0:k} = \boldsymbol{\omega}^{0:k})$ are given, for all admissible tuples $\boldsymbol{\omega}^{0:k}$ and k, where we used the shorthand notation $\underline{\boldsymbol{\omega}}^{0:t} \triangleq (\underline{\boldsymbol{\omega}}^0, \underline{\boldsymbol{\omega}}^1, \ldots, \underline{\boldsymbol{\omega}}^t)$ (the first $t+1$ random variables), and $\boldsymbol{\omega}^{0:t} \triangleq (\boldsymbol{\omega}^0, \boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^t)$ ($t+1$ possible values for the random variables $\underline{\boldsymbol{\omega}}^{0:t}$). In fact, this joint distribution induces a valid probability space $(\Omega, \mathcal{F}, P)$ over which $\underline{\boldsymbol{\omega}}$ is well-defined and has $p_{\underline{\boldsymbol{\omega}}^{0:k}}$ as its finite-dimensional distributions.

This process fully describes the evolution of Algorithm 1. Indeed, given a starting point $\mathbf{x}^0$, the trajectories of the variables $\mathbf{x}^k$ and $\mathbf{x}^{k-\mathbf{d}}$ are completely determined once a sample path $\{(i^k, \mathbf{d}^k)\}_{k\in\mathbb{N}_+}$ is drawn by $\underline{\boldsymbol{\omega}}$.

Finally, we need to define the conditional probabilities $p((i, \mathbf{d}) \,|\, \boldsymbol{\omega}^{0:k}) \triangleq \mathbb{P}(\underline{\boldsymbol{\omega}}^{k+1} = (i, \mathbf{d}) | \underline{\boldsymbol{\omega}}^{0:k} = \boldsymbol{\omega}^{0:k})$. We require a few minimal conditions on the probabilistic model, as stated next.

Assumption D Given the global model described in Algorithm 1 and the stochastic process $\underline{\boldsymbol{\omega}}$, suppose that

(D1) There exists a $\delta \in \mathbb{N}_+$, such that $d_i^k \leq \delta$, for all i and k;

(D2) For all $i = 1, \ldots, N$ and $\boldsymbol{\omega}^{0:k-1}$ such that $p_{\underline{\boldsymbol{\omega}}^{0:k-1}}(\boldsymbol{\omega}^{0:k-1}) > 0$, it holds

$$\sum_{\mathbf{d}\in\mathcal{D}} p((i, \mathbf{d}) \,|\, \boldsymbol{\omega}^{0:k-1}) \geq p_{\min},$$

for some $p_{\min} > 0$;

(D3) For a given $\omega = (\omega^k)_k \in \bar{\Omega}$, where $\bar{\Omega} \subseteq \Omega$ such that $\mathbb{P}(\bar{\Omega}) = 1$, there exists a set $\mathcal{V}(\omega) \subseteq \mathcal{N} \times \mathcal{D}$ such that
$p((i, \mathbf{d})|\boldsymbol{\omega}^{0:k-1}) \geq \Delta > 0$ for all $(i, \mathbf{d}) \in \mathcal{V}(\omega)$ and k, and
$p((i, \mathbf{d})|\boldsymbol{\omega}^{0:k-1}) = 0$ otherwise.

(D4) The block variables are partitioned in P sets $I_1, \ldots, I_P$, where P is the number of available cores, and each core processes a different set I_j of block variables. This implies: $\mathbf{x}_{i^k}^k = \tilde{\mathbf{x}}_i^k$.

Note that D4 is necessary in order to preserve feasibility of the iterates. This is because the feasible set is nonconvex, and so a convex combination of the two feasible vectors $\hat{\mathbf{x}}(\tilde{\mathbf{x}}^k)$ and $\mathbf{x}^k$ may not be feasible. However, by construction of the subproblem, $\tilde{\mathbf{x}}^k$ and $\hat{\mathbf{x}}(\tilde{\mathbf{x}}^k)$ are both feasible and lie in a convex subset of the feasible set, and thus any convex combination of them is feasible for the original constraints.

We point out that this setting (D4) has proved to be very effective from an experimental point of view even in the absence of nonconvex constraints. This has been observed also in [12], and indeed is the setting in which the experiments were performed, despite the convergence proof drawn up without this arrangement.

Algorithm 1 AsyFLEXA: A Global Description

```
Initialization: $k = 0$, $\mathbf{x}^0 \in \mathcal{X}$, $\gamma > 0$, and $k$.
while a termination criterion is not met do
    (S.1): The random variables $(\underline{i}^k, \underline{\mathbf{d}}^k)$ take realization $(i, \mathbf{d})$;
    (S.2): Compute $\hat{\mathbf{x}}_i(\mathbf{x}^{k-\mathbf{d}})$ [cf. (2)];
    (S.3): Read $\mathbf{x}_i^k$;
    (S.4): Update $\mathbf{x}_i$ by setting
        $$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \gamma(\hat{\mathbf{x}}_i(\mathbf{x}^{k-\mathbf{d}}) - \mathbf{x}_i^k); \qquad (5)$$
    (S.5): Update the iteration counter $k \leftarrow k + 1$;
end while
return $\mathbf{x}^k$
```

3 Complexity

Assumptions A-D in the previous two sections correspond exactly to the assumptions used in the companion paper [3] in order to analyze the convergence properties of Algorithm 1 therein when applied to the general non convex problem (1); more precisely they correspond to Assumptions A′, B′, E, and C respectively. Note that Algorithm 1 and Algorithm AsyFLEXA-NCC in [3, Section 4] are identical with the only exception that *in [3] we use a diminishing stepsize while in Algorithm 1 we use a fixed stepsize* γ. The difference is standard and derives from the necessity to get complexity results; as a part of our complexity result we will also show convergence of this fixed step size variant of Algorithm 1 in [3]. Besides the rather standard Assumptions A-D, in order to get complexity results we also need that the solution mapping $\hat{\mathbf{x}}_i(\cdot)$ is Lipschitz continuous on $\mathcal{X}$.

Assumption E $\hat{\mathbf{x}}_i(\cdot)$ is Lipschitz continuous with constant $L_{\hat{x}}$ on $\mathcal{X}$

This assumption is rather mild. While, in order to concentrate on the complexity result, we postpone its more detailed analysis to the Appendix, we mention here that, in our setting and supposing $\mathcal{X}$ is bounded, it is automatically satisfied if the feasible region of problem (1) is convex or, in case of non convex constraints, if some constraint qualifications are satisfied.

We will use the norm of the following quantity as a measure of optimality:

$$M_F(\mathbf{x}) = \mathbf{x} - \underset{\mathbf{y} \in \mathcal{K}_1(\mathbf{x}_1) \times \ldots \times \mathcal{K}_N(\mathbf{x}_N)}{\arg\min} \{\nabla f(\mathbf{x})^{\mathrm{T}}(\mathbf{y} - \mathbf{x}) + g(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2\} \quad (6)$$

This is a valid measure of stationarity because $M_F(\mathbf{x})$ is continuous and $M_F(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ is a stationary solution of Problem (1). Note that it is an extension of the optimality measure given in [19] for Problem (1) but without the nonconvex constraints. We will study the rate of decrease of $\mathbb{E}(\|M_F(\underline{\mathbf{x}}^k)\|_2^2)$. More precisely, the following theorem gives an upper bound on the number of iterations needed to decrease $\mathbb{E}(\|M_F(\underline{\mathbf{x}}^k)\|_2^2)$ below a certain chosen value ϵ, provided that the stepsize γ is smaller than a certain given constant.

Theorem 1 *Consider Assumptions A-E and let* $\{\mathbf{x}^k\}$ *be the sequence generated by Algorithm 1, with* γ *such that:*

$$0 < \gamma \leq \min\left\{\frac{(1-\rho^{-1})}{2(1+L_{\hat{x}}N(3+2\psi))}; \frac{c_{\tilde{f}}}{L_f + \frac{\delta\psi' L_f}{2\Delta}}\right\}, \tag{7}$$

where $\rho > 1$ *is any given number. Define* K_ϵ *to be the first iteration such that* $\mathbb{E}(\|M_F(\underline{\mathbf{x}}^k)\|_2^2) \leq \epsilon$. *Then:*

$$K_\epsilon \leq \frac{1}{\epsilon}\frac{4(1+(1+L_B+L_E)(1+L_E L_B \delta\psi'\gamma^2))}{p_{min}\gamma(2\Delta(c_{\tilde{f}}-\gamma L_f)-\gamma\delta\psi' L_f)}(F(\mathbf{x}^0)-F^*), \tag{8}$$

where $F^* = \min_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x})$, $\psi \triangleq \sum_{t=1}^{\delta} \rho^{\frac{t}{2}}$, *and* $\psi' \triangleq \sum_{t=1}^{\delta} \rho^t$.

Proof See Appendix.

Remark 1 There is an inverse proportionality relation between the desired optimality tolerance satisfaction and the required number of iterations, i.e., $K_\epsilon \propto \frac{1}{\epsilon}$. This is consistent with the sequential complexity result for SCA algorithms for nonconvex problems [19] from which it can also be seen that the constants match when asynchrony is removed. Similarly, it matches results on asynchronous methods for nonconvex problems as appearing in [5,10]. This suggests that the actual complexity rate is tight, despite a more complex, and thus difficult to analyze, model.

Remark 2 The bounds (7) and (8) are rather intricate and depend in a complex way on all constants involved. However, it is of particular interest to try to understand how the speedup of the method is influenced by the number of cores used in the computations, all other parameters of the problem being fixed. We consider the most common shared memory architecture and denote by N_r the number of cores. To perform the analysis we assume that γ is "very small". By this we mean that not only does γ satisfy (7) but (a) it is such that variations of N_r will make (7) still satisfied by the chosen value of γ and (b) the chosen value of γ will make all terms involving γ^2 in (8) negligible with respect to the other terms. Under these circumstances (8) reads, for some suitable constant C_1,

$$K_\epsilon \lesssim \frac{1}{\epsilon}\frac{C_1}{\gamma}$$

thus implying a linear speedup with the number of cores. Note that when N_r increases the value on the right-hand-side of (7) will decrease, because we can reasonably assume that $N_r \approx \delta$, so that if N_r increases ψ and ψ' also increase. Once the right-hand-side of (7) hits the chosen value of γ the analysis above fails and linear speedup is harder to establish. Therefore we expect that the smaller the chosen γ the larger the number of cores for which we can guarantee theoretical linear speedup. One should take into account that this type of results is mainly of theoretical interest because on the one hand (7) and (8) only give (worst case scenario) upper estimates and, on the other hand, in practice one would prefer to use larger values of γ.

An interesting result can be easily derived from Theorem 1 if we consider a synchronous version of Algorithm 1.

Corollary 1 *Consider a synchronous version of Algorithm 1, i.e.* $\mathbf{x}^{k-\mathbf{d}} = \mathbf{x}^k \, \forall \, k$, *under assumptions A-D, where at each iteration a block is randomly picked to be updated according to a uniform distribution. Choose* γ *such that:*

$$0 < \gamma \leq \min \left\{ \frac{(1-\rho^{-1})}{2(1+3L_{\hat{x}}N)}; \frac{c_{\tilde{f}}}{L_f} \right\}. \tag{9}$$

Then:

$$K_\epsilon \sim \mathcal{O}\left(\frac{N^3}{\epsilon}\right). \tag{10}$$

Proof Since we consider a synchronous scheme, we have $\mathcal{D} = \{\mathbf{0}\}$, which implies $\delta = 0$ and $\psi = \psi' = 0$. The uniform random block selection implies that $p_{\min} = \varDelta = \frac{1}{N}$. Substituting these values in (7) and (8) we get respectively (9) and (10).

The Corollary states, as expected, that in a synchronous implementation of our algorithm, the iteration complexity does not depend (ideally) on the number of cores running. This of course comes from the fact that in this setting the convergence speed of the algorithm is no longer affected by the use of old information $\tilde{\mathbf{x}}^k$.

4 Numerical Results

In this section we test our algorithm on LASSO problems and on a nonconvex sparse learning problem and compare it to state-of-art methods. In particular, we test the diminishing-stepsize version of the method proposed and studied in the companion paper [3]. Results for the fixed-stepsize version of the method whose complexity has been studied in the previous section are not reported. Indeed, as usual for these methods, if the theoretical stepsize (7) is used, the algorithms simply do not make any practical progress towards optimality in a reasonable number of iterations. If, on the other hand, we disregard the bound (7) that, we recall, is an upper bound, and use a “large” stepsize (in particular we found the value of 0.95 practically effective), the numerical results are essentially the same as those obtained with the diminishing stepsize version for which we report the results. Since this latter version, as shown in [3], has theoretical convergence guarantee for the chosen stepsize rule, we present the results only for this version of the method.

4.1 Implementation

All tested codes have been written in C++ using Open- MP. The algorithms were tested on the Purdue's Community Cluster Snyder on a machine with two 10-Core Intel Xeon-E5 processors (20 cores in total) and 256 GB of RAM.

•**AsyFLEXA**: We tested Algorithm 1 from [3], always setting $N = n$, i.e. we considered scalar subproblems. These subproblems, for the choice of $\tilde{f}_i$ that we used and that we specify below for each class of problems, can be solved in closed form using the soft-thresholding operator [1]. For the stepsize sequence $\{\gamma^k\}_{k\in\mathbb{N}_+}$ we used the rule $\gamma^{k+1} = \gamma^k(1 - \mu\gamma^k)$ with $\gamma^0 = 1$ and $\mu = 10^{-6}$.

We compared AsyFLEXA with the following state-of-the-art asynchronous schemes: AsySPCD and ARock. We underline that the theoretical stepsize rules required for the convergence of AsySPCD and ARock lead to practical non-convergence, since they prescribe extremely small stepsizes. For both algorithms we made several attempts to identify practical rules that bring the best numerical behaviour, as detailed below.

• **AsySPCD**: This is the asynchronous parallel stochastic proximal gradient coordinate descent algorithm for the minimization of convex and nonsmooth objective functions presented in [12]. Since the algorithm was observed to perform poorly if the stepsize γ is chosen according to [12, Th. 4.1]−the value guaranteeing theoretical convergence−in our experiments we used $\gamma = 1$, which violates [12, Th. 4.1] but was effective on our test problems. Note also that this is the value actually used in the numerical tests reported in [12]. We underline that in order to implement the algorithm it is also required to estimate some Lipschitz-like constants.
• **ARock**: ARock [17] is an asynchronous parallel algorithm proposed to compute fixed-points of a nonexpansive operator. Thus it can be used to solve convex optimization problems. The algorithm requires a stepsize and the knowledge of the Lipschitz constant of ∇f. Here, again, the use of stepsizes that guarantee theoretical convergence leads to very poor practical performance. In this case we found that the most practical, effective version of the method could be obtained by using for the stepsize the same rule employed in our AsyFLEXA, with a safeguard that guarantees that the stepsize never becomes smaller than 0.1.

We also tested the stochastic version of Asynchronous PALM [4], an asynchronous version of the Proximal Alternating Linearized Minimization (PALM) algorithm for the minimization of nonsmooth and nonconvex objective functions. We did not report the results, because its practical implementation (using unitary stepsize rather than the one guaranteeing convergence, for the reasons explained above) basically coincides with the one of AsySPCD.

Before reporting the numerical results it is of interest to briefly contrast these algorithms in order to better highlight some interesting properties of AsyFLEXA.

1. In all tests we partitioned the variables among the cores, with only one core per partition. Therefore, each core is in charge of updating its assigned variables and no other core will update those variables. There is consensus in the literature that this configuration brings better results experimentally in shared memory architectures, see e.g. [12,16]. Furthermore, if one is considering a message passing architecture, this is actually the only possible choice. It is then interesting to note that this choice is fully covered by our theory, while the theory of ARock [17] requires that each core has access to all variables and therefore can not handle it. The theory supporting AsySPCD [12] requires that at each iteration a variable is updated, with all variables selected with *equal* probability. The equal probability requirement in the partitioned configuration can not be theoretically excluded, but is totally unlikely: it would require that all cpu are identical and that all subproblems require exactly the same amount of time to be processed. In any case, even not considering the partitioned variables issue, the practical choice for the stepsizes adopted are not covered by the theory in [12] and [17]. We believe this clearly shows our analysis to be robust and well matched to practical computational architectures.
2. It is of interest to note that we run experiments also on *nonconvex* problems. Our algorithm has theoretical convergence guarantees for nonconvex problems while neither ARock nor AsySPCD has any such guarantees.
3. Both AsySPCD and ARock (and also PALM) theoretically require, for their implementation, the estimation of some Lipschitz constants of the problem functions. Although in our test problems the estimations could be reasonably done, this requirement in general can be extremely time consuming if at all possible on other problems.
4. One last difference worth mentioning is that contrary to AsyFLEXA and ARock, AsySPCD and PALM require a memory lock of the variable(s) being updated while the solution of the corresponding subproblems are computed. In the partitioned configuration we adopted this is not an issue, because nobody else can update the variables being updated by a given core. However, when other configurations are chosen this can become a source of delays, especially if the subproblems are complex and take some time to be solved. This could easily happen, for example, if the subproblems involve more than one variable or if they are nonconvex, as those that could appear in PALM

We are now ready to illustrate the numerical results.

4.2 LASSO

Consider the LASSO problem, i.e. Problem (1), with
$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$, $G(\mathbf{x}) = \lambda \cdot \|\mathbf{x}\|_1$, and $\mathcal{X} = \mathbb{R}^n$, with $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. In all implementations of the algorithms, we precomputed the matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and the vector $\mathbf{A}^{\mathrm{T}}\mathbf{b}$ offline. In all the experiments, the starting point of all the algorithms was set to the zero vector. For problems in which the optimal

value F^* is not known, in order to compute the relative error we estimated F^* by running a synchronous version of AsyFLEXA until variations in the objective value were undetectable.

For AsyFLEXA we set $\tilde{f}_i(x_i; \mathbf{x}^{k-\mathbf{d}}) = f(x_i; \mathbf{x}_{-i}^{k-\mathbf{d}}) + \frac{\tau^k}{2}(x_i - x_i^{k-d_i})^2$, where $\mathbf{x}_{-i}$ denotes the vector obtained from $\mathbf{x}$ by deleting the block $\mathbf{x}_i$. The sequence $\{\tau^k\}_{k\in\mathbb{N}_+}$, shared among all the cores, was updated every n iterations, according to the heuristic proposed for FLEXA [6].

We run tests on LASSO problems generated according to three different methods; all curves reported are averaged over five independent random problem realizations.

Gaussian problems [12]: In this setting we generated the LASSO problem according to the procedure used in [12]; where the matrix $\mathbf{A}$ has samples taken from a Gaussian $\mathcal{N}(0,1)$ distribution, $\bar{\mathbf{x}} \in \mathbb{R}^n$ is a sparse vector with s nonzeros and $\mathbf{b} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{e}$, with $\mathbf{e} \in \mathbb{R}^m \sim \mathcal{N}(0, \sigma^2)$. In particular we chose $m = 20000$, $n = 40000$, $s = 40$ and $\sigma = 0.01$. For the regularization parameter we used the value suggested in [12]: $\lambda = 20\sqrt{m\log(n)}\sigma$. In Fig. 1 we plot the relative error on the objective function versus the CPU time, using 2 and 20 cores ((c) = cores). The figure shows that when increasing the number of cores, all the algorithms converge quickly to the solution with comparable performances. In this particular setting, and contrary to what happens in all other problems, AsySPCD has a slightly better behavior than AsyFLEXA but it does not have convergence guarantees.

In order to quantify the scalability of the algorithms, in Fig. 2 we plot the speedup achieved by each of the algorithms versus the number of cores (of course we run all algorithms also for values between $c = 1$ and $c = 20$, more precisely for $c = 1, 2, 4, 8, 10, 20$). We defined the speedup as the ratio between the runtime on a single core and the runtime on multiple cores. The runtimes we used are the CPU times needed to reach a relative error strictly less than 10^{-4}. The figure shows that all the algorithms obtain a good gain in the performances by increasing the number of cores which is not far from the ideal speedup.

Nesterov's problems [15]: Here we generated a LASSO problem using the random generator proposed by Nesterov in [15], which permits us to control the sparsity of the solution. We considered a problem with 40000 variables and matrix $\mathbf{A}$ having 20000 rows, and set $\lambda = 1$; the percentage of nonzero in the solution is 1%. In Fig. 4 we plot the relative error on the objective function (note that for Nesterov's model the optimal solution is known) versus the CPU time, using 2 and 20 cores. The figure clearly shows that AsyFLEXA significantly outperforms all the other algorithms on these problems. Moreover, the empirical convergence speed significantly increases with the number of cores, which instead is not observed for the other algorithms, see Fig. 4, where we only report data for AsyFLEXA, given that the other algorithms do not reach the prefixed threshold error value of 10^{-4} in one hour of computation time.

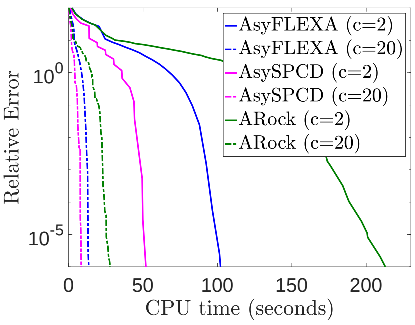

Fig. 1: LASSO: comparison in terms of relative error versus CPU time (in seconds) for Liu and Wright's problems [12]

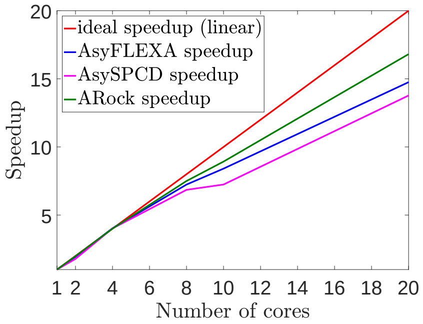

Fig. 2: LASSO: speedup of the tested algorithms for Liu and Wright's problems [12]

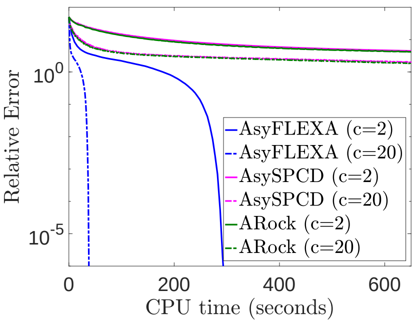

Fig. 3: LASSO: comparison in terms of relative error versus CPU time (in seconds) for Nesterov's problems [15]

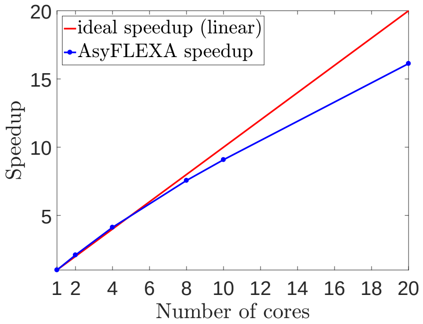

Fig. 4: LASSO: speedup of AsyFLEXA for Nesterov's problems [15]

Gondzio's problem: We generated these LASSO problems using the generator proposed by Gondzio in [7]. The key feature of this generator is the possibility of choosing the condition number of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, and we set it to 10^4. We generated a matrix $\mathbf{A}$ with 2^{14} rows and $\lceil 1.01n \rceil$ columns, where the ceiling function $\lceil \cdot \rceil$ returns the smallest integer greater than or equal to its argument. The sparsity in the solution is 0.1% and we set $\lambda = 1$. Fig. 5 shows the relative error with respect to the CPU time for the different algorithms we tested, when using 2 and 16 cores. Fig. 6 that shows the speedup achieved by our algorithm on these problems (in this case we run the algorithm for $c = 1, 2, 4, 8, 16$). We see that the behavior of the algorithm is qualitatively very similar to the one obtained for the previous set of problems.

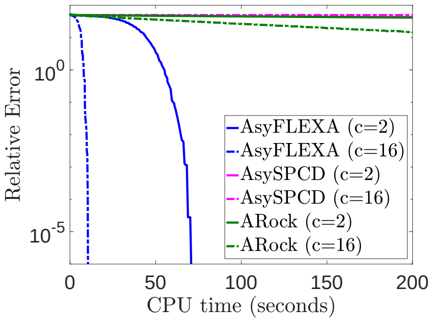

Fig. 5: LASSO: comparison in terms of relative error versus CPU time (in seconds) for Gondzio's problems [7]

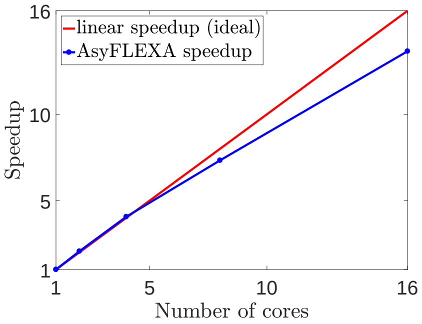

Fig. 6: LASSO: speedup of AsyFLEXA for Gondzio's problems [7]

4.3 Nonconvex Sparse Learning

We consider now the following nonconvex instance of problem (1):

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\; F(\mathbf{x}) \triangleq \underbrace{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2}_{=m(\mathbf{x})} + \lambda H(\mathbf{x}), \tag{11}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\lambda > 0$ and $H(\mathbf{x})$ is a regularizer used to balance the amount of sparsity in the solution. (11) is a standard formulation used for doing regression in order to recover a sparse signal from a noisy observation vector $\mathbf{b}$ [23]. Common choices for the regularizer H are surrogates of the l_0 norm, as the frequently used l_1 norm (in this case we recover the LASSO problem of the previous section). However, it is known that nonconvex surrogates of the l_0 norm can provide better solutions than the l_1 norm, in terms of balance between compression and reconstruction error, see e.g. [9]. For this reason we tested here two nonconvex regularizer functions, that we already used in [21]: the exponential one $H(\mathbf{x}) = H_{\exp}(\mathbf{x}) = \sum_{i=1}^{n} h_{\exp}(x_i)$, with $h_{\exp}(x_i) = 1 - e^{-\theta_{\exp}|x|}$ and $\theta_{\exp} > 0$, and the logarithmic one $H(\mathbf{x}) = H_{\log}(\mathbf{x}) = \sum_{i=1}^{n} h_{\log}(x_i)$, with $h_{\log}(x_i) = \frac{\log(1+\theta_{\log}|x|)}{\log(1+\theta_{\log})}$ and $\theta_{\log} > 0$. Note that Problem (11) does not immediately appear to be in the format needed by our algorithm, i.e. the sum of a smooth term and of a possibly nondifferentiable convex one. But we can put the problem in this form, as explained next. In both cases the function h possesses a DC structure that allows us to rewrite it as

$$h(x) = \underbrace{\eta(\theta)|x|}_{\triangleq h^+(x)} - \underbrace{\eta(\theta)|x| - h(x)}_{\triangleq h^-(x)}, \tag{12}$$

with $\eta_{\exp}(\theta_{\exp}) = \theta_{\exp}$ and $\eta_{\log}(\theta_{\log}) = \frac{\theta_{\log}}{\log(1+\theta_{\log})}$. It is easily verified that h^+ is convex and that, slightly more surprisingly, h^- is continuously differentiable with a Lipschitz gradient [9]. Given these facts, it is now easily seen that problem (11) can be put in the required format by setting $f \triangleq h^-$ and $G \triangleq m + h^+$. With this in mind, we can implement AsyFLEXA to solve problem (11) by considering scalar blocks ($N = n$) and the following formulation for the best-response $\hat{x}_i$, for $i = 1, \dots, n$:

$$\hat{x}_i(\mathbf{x}^{k-\mathbf{d}}) = \\ \arg\min_{x_i \in \mathbb{R}} \left\{ f(x_i; \mathbf{x}_{-i}^{k-\mathbf{d}}) - \lambda \nabla h^-(x_i^{k-d_i})^{\mathrm{T}}(x - x_i^{k-d_i}) + \lambda h^+(x_i) + \tfrac{\tau^k}{2}(x_i - x_i^{k-d_i})^2 \right\}, \tag{13}$$

Note that, once again, the solution of (13) can be computed in closed-form through the soft-thresholding operator [1]. For the positive constant τ^k we use again the same heuristic rule used for the LASSO problems.

In order to generate the 5 random instances of the problem we must specify how we generate the quadratic function $m(\cdot)$. In order to favour AsySPCD and ARock we use the Liu and Wright's generator used for the first set of LASSO problems discussed above. In particular, we generated the underlying sparse linear model according to: $\mathbf{b} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{e}$ where $\mathbf{A}$ has 20000 rows (with values normalized to one) and 40000 columns. $\mathbf{A}$, $\bar{\mathbf{x}}$ and $\mathbf{e}$ have i.i.d. elements coming from a Gaussian $\mathcal{N}(0, \sigma^2)$ distribution, with $\sigma = 1$ for $\mathbf{A}$ and $\bar{\mathbf{x}}$, and $\sigma = 0.1$ for the noise vector $\mathbf{e}$. To impose sparsity on $\bar{\mathbf{x}}$, we randomly set to zero 95% of its component.

Since (11) is nonconvex, we compared the performance of the algorithms using the following distance to stationarity: $\|\hat{\mathbf{x}}(\mathbf{x}^k) - \mathbf{x}^k\|_\infty$. Note that this is a valid stationarity measure: it is continuous and $\|\hat{\mathbf{x}}(\mathbf{x}^*) - \mathbf{x}^*\|_\infty = 0$ if and only if $\mathbf{x}^*$ is a stationary solution of (11). In Figure 7 we plot the stationarity measure versus the CPU time, for all the algorithms using 2 and 20 cores, for problem (11) where $H(\mathbf{x}) = H_{\log}(\mathbf{x})$ with $\theta_{\log} = 20$; the curves are averaged over 5 independent realizations. All the algorithms were observed to converge to the same stationary solution of (11) even if, we recall, AsySPCD and ARock have no formal proof of convergence in this nonconvex setting. The figure shows that, even in the nonconvex case, AsyFLEXA has good performances and actually behaves better that all the other algorithms while being also guaranteed to converge. As a side issue it is interesting to illustrate the utility of our nonconvex regularizers with respect to the more standard l_1 norm regularizer. In Fig. 8 and Fig. 9 we plot respectively the Normalized Mean Square Error (NMSE) (defined as: $\mathrm{NMSE}(\mathbf{x}) = \|\mathbf{x} - \bar{\mathbf{x}}\|_2^2 / \|\bar{\mathbf{x}}\|_2^2$) and the percentage of nonzeros obtained by solving the aforementioned problem with AsyFLEXA for different values of the regularization parameter λ and for different surrogates of the l_0 norm: the l_1 norm, the exponential function ($\theta_{\exp} = 20$) and the logarithmic function ($\theta_{\log} = 20$). AsyFLEXA is terminated when the merit function goes below 10^{-4} or after $100n$ iterations. These two figures interestingly show that the two nonconvex regularizers obtain their lowest NMSE for a value of λ which corresponds to a good compression result, close to the number of nonzeros of the original signal. On the other side, the l_1 norm attains its lowest NMSE by reconstructing the signal with more of

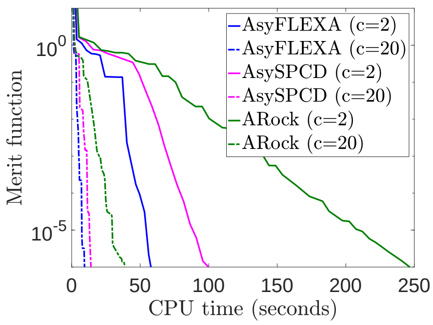

Fig. 7: Nonconvex problem: comparison in terms of relative error versus CPU time (in seconds)

15% of nonzeros, much more than the original 5%. In order to get close to the desired number of nonzeros in the reconstruction with the l_1 norm, it is necessary to increase the value of λ, but this leads to a worsening in terms of NMSE of at least two times with respect to the minimum NMSE attainable.

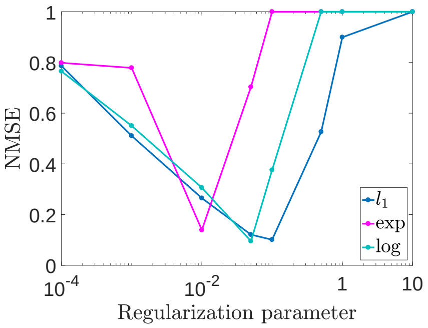

Fig. 8: Nonconvex problem: NMSE for different values of the regularization parameter λ and for different surrogates of the l_0 norm

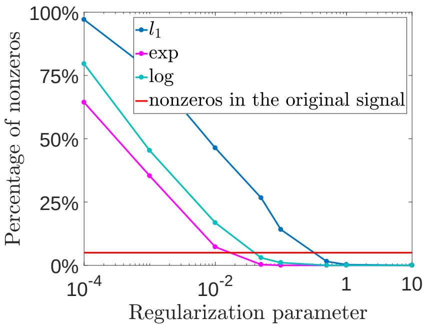

Fig. 9: Nonconvex problem: percentage of nonzeros in the solution for different values of the regularization parameter λ and for different surrogates of the l_0 norm

A full understanding of the numerical behavior of our algorithm and its comparison with existing alternatives certainly needs further numerical tests, on both larger and more complex problems. But the substantial results already reported here seem to clearly indicate that our method is reliable and robust and capable to efficiently solve problems that other methods cannot tackle. The improved performance of AsyFLEXA with respect to AsySPCD and ARock on some classes of problems seems due, in our experience, to the much wider flexibility we have in the choice of the model $\tilde{f}$ which is not linked in any way to Lipschitz constants of the problem functions.

5 Conclusions

Leveraging the new theoretical framework we introduced in the companion paper [3], in this work, we studied the convergence rate of a novel parallel asynchronous algorithm for the minimization of the sum of a nonconvex smooth function and a convex nonsmooth one subject to nonconvex constraints. Furthermore, we presented numerical results on convex and nonconvex problems showing that our method is widely applicable and outperforms asynchronous state-of-the-art-schemes in at least some classes of important problems.

Acknowledgments

We very gratefully acknowledge useful discussions with Daniel Raph on stability issues related to Proposition 2.

References

1. A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, Jan. 2009.
2. D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.
3. Loris Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari. Asynchronous parallel algorithms for nonconvex big-data optimization - part i: Model and convergence. *Submitted to Mathematical Programming*, 2016.
4. D. Davis. The asynchronous palm algorithm for nonsmooth nonconvex problems. *arXiv preprint arXiv:1604.00526*, 2016.
5. D. Davis, B. Edmunds, and M. Udell. The sound of apalm clapping: Faster nonsmooth nonconvex optimization with stochastic asynchronous palm. *arXiv preprint arXiv:1606.02338*, 2016.
6. F. Facchinei, G. Scutari, and S. Sagratella. Parallel selective algorithms for nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63(7):1874–1889, 2015.
7. K. Fountoulakis and J. Gondzio. A second-order method for strongly convex\ ell _1-regularization problems. *Mathematical Programming*, 156(1-2):189–219, 2016.
8. M. Hong. A distributed, asynchronous and incremental algorithm for nonconvex optimization: An admm based approach. *arXiv preprint arXiv:1412.6058*, 2014.
9. H. A. Le Thi, T P. Dinh, H. M. Le, and X. T. Vo. Dc approximation approaches for sparse optimization. *European Journal of Operational Research*, 244(1):26–46, 2015.
10. X. Lian, Y. Huang, Y. Li, and J. Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 2719–2727, 2015.
11. J. Liu. Sensitivity analysis in nonlinear programs and variational inequalities via continuous selections. *SIAM Journal on Control and Optimization*, 33:1040–1060, 1955.
12. J. Liu and S. J. Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.
13. J. Liu, S. J. Wright, C. Ré, V. Bittorf, and S. Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. *The Journal of Machine Learning Research*, 16(1):285–322, 2015.

14. H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *arXiv:1507.06970*, 2016.
15. Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140:125–161, August 2013.
16. F. Niu, B. Recht, C. Re, and S. J. Wright. Hogwild: a lock-free approach to parallelizing stochastic gradient descent. *Advances in Neural Information Processing Systems*, pages 693–701, 2011.
17. Z. Peng, Y. Xu, M. Yan, and W. Yin. Arock: an algorithmic framework for asynchronous parallel coordinate updates. *arXiv preprint arXiv:1506.02396*, 2015.
18. D. Ralph and S. Dempe. Directional derivatives of the solution of a parametric nonlinear program. *Mathematical programming*, 70(1-3):159–172, 1995.
19. M. Razaviyayn, M. Hong, Z. Q. Luo, and J. S. Pang. Parallel successive convex approximation for nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, pages 1440–1448, 2014.
20. G. Scutari, F. Facchinei, and L. Lampariello. Parallel and distributed methods for constrained nonconvex optimization-part i: Theory. *IEEE Transactions on Signal Processing, published online DOI: 10.1109/TSP.2016.2637317*, 2016.
21. G. Scutari, F. Facchinei, L. Lampariello, S. Sardellitti, and P. Song. Parallel and distributed methods for constrained nonconvex optimization–part ii: Applications in communications and machine learning. *IEEE Transactions on Signal Processing, published online DOI: 10.1109/TSP.2016.2637314*, 2016.
22. W. H. Young. On classes of summable functions and their fourier series. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 87(594):225–229, 1912.
23. S. Zhang and J. Xin. Minimization of transformed l_1 penalty: Theory, difference of convex function algorithm, and robust application in compressed sensing. *arXiv preprint arXiv:1411.5735*, 2014.

6 Appendix

6.1 Preliminaries

We first introduce some preliminary definitions and results that will be instrumental to prove Theorem (1). Some of the following results have already been discussed in [3], to which we refer the interested reader for more details.

In the rest of the Appendix it will be convenient to use the following notation for the random variables and their realizations: underlined symbols denote random variables, e.g., $\underline{\mathbf{x}}^k$, $\underline{\tilde{\mathbf{x}}}^k$ whereas the same symbols with no underline are the corresponding realizations, e.g., $\mathbf{x}^k \triangleq \underline{\mathbf{x}}^k(\omega)$ and $\tilde{\mathbf{x}}^k \triangleq \underline{\tilde{\mathbf{x}}}^k(\omega)$.

1. **Properties of the best response $\hat{\mathbf{x}}(\bullet)$ and Assumption E.** The following proposition is a direct consequence of [20, Lemma 7].

Proposition 1 *Given $\hat{\mathbf{x}}(\bullet)$ as defined in (2), under Assumptions A-C the following holds. For any $i \in \mathcal{N}$ and $\mathbf{y} \in \mathcal{K}$,*

$$(\hat{\mathbf{x}}_i(\mathbf{y}) - \mathbf{y}_i)^T \nabla_{\mathbf{y}_i} f(\mathbf{y}) + g_i(\hat{\mathbf{x}}_i(\mathbf{y})) - g_i(\mathbf{y}_i) \leq -c_{\tilde{f}} \|\hat{\mathbf{x}}_i(\mathbf{y}) - \mathbf{y}_i\|_2^2 \,. \qquad (14)$$

The discussion of when Assumption E is satisfied, i.e. of when the Lipschitz continuity of $\hat{\mathbf{x}}(\bullet)$ holds, is in general complex. Fortunately, for the kind of problems we are interested in, it simplifies considerably even if a case by case analysis may be needed. The following proposition shows in the two fundamental cases in which we are interested that Assumption E either holds automatically or can be guaranteed by suitable constraint qualifications.

Proposition 2 *Suppose that Assumptions A-C hold and that $\mathcal{X}$ is compact. Then the following two assertions hold.*

(a) *If the feasible set of problem* (1) *is convex (and therefore there are no nonconvex constraints c), then Assumption E is satisfied.*

(b) *Suppose, for simplicity of presentation only, that the sets $\mathcal{X}_i$ are specified by a finite set of convex constraints $h_i(\mathbf{x}_i) \leq 0$, with each component of h_i convex and continuously differentiable. Consider problem* (1) *in the most common case in which $G(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$ for some positive constant λ. Assume that $\tilde{f}$ and $\tilde{g}$ are C^2 and that, for each $\mathbf{y} \in \mathcal{X}$, problem* (2) *satisfies the Mangasarian-Fromovitz constraints qualification (MFCQ) and the Constant Rank constraint qualification (CRCQ) in $\mathbf{y}$. Then Assumption E is satisfied.*

Proof Case (a) is nothing else but [6, Proposition 8 (a)]. Thus, we focus next on case (b). Because of the compactness of $\mathcal{X}_i$ it is enough to show that every $\hat{\mathbf{x}}_i(\bullet)$ is locally Lipschitz. $\hat{\mathbf{x}}_i(\bullet)$ is the unique solution of the strongly convex problem (2). This problem can be equivalently rewritten, by adding extra variables $\mathbf{t} \triangleq (t_1, t_2, \ldots, t_{n_i})$, as

$$\begin{aligned} \min_{\mathbf{x}_i,\mathbf{t}} \quad & \tilde{f}_i(\mathbf{x}_i;\mathbf{x}^k) + \lambda\,(t_1 + \cdots + t_{n_i}) \\ \text{subject to } & \tilde{c}_{j_i}(\mathbf{x}_i;\mathbf{x}_i^k) \leq 0, \quad j_i = m_{i-1}+1, \ldots, m_i \\ & h_i(\mathbf{x}_i) \leq 0 \\ & -t_\ell \leq (\mathbf{x}_i)_\ell \leq t_\ell, \quad \ell = 1, \ldots, n_i. \end{aligned} \tag{15}$$

Since problem (2) has the unique solution $\hat{\mathbf{x}}_i(\mathbf{x}^k)$, also problem (15) has the unique solution $(\hat{\mathbf{x}}_i(\mathbf{x}^k), \hat{t}_1, \ldots, \hat{t}_{n_i})$, with $\hat{t}_\ell \triangleq |[\hat{\mathbf{x}}_i(\mathbf{x}^k)]_\ell|$. Note also that, by the particular structure of the new linear constraints added in (15) with respect to (2), and by the assumptions made on problem (2), problem (15) satisfies the MFCQ and the CRCQ at its solution. Finally, observe that assumption B1 ensures that the Lagrangian of problem (15) is positive definite at the solution of this problem. Then, it is easy to see that the Theorem holds by, for example, [18, Theorem 2] or [11, Theorem 3.6].

Remark 3 We note that the assumption that $\mathcal{X}$ be compact can always be satisfied in our setting by suitably redefining $\mathcal{X}$, if needed. In fact, A5 guarantees that $\mathcal{K}$ is compact,and therefore if $\mathcal{X}$ is not compact we can simply redefine it by intersecting it with a suitably large ball without changing problem (1)

Remark 4 Note that the line of proof used in Proposition 2 (b) can be adapted to deal with all cases in which the "epigraphical transformation" of the problem leads to smooth constraints and does not destroys the MFCQ and CRCQ of the original problem. For example we can cover in this way the case in which the function G represents a group ℓ_2 or ℓ_∞ regularizations.

2. Young's Inequality [22]. Consider the Young's inequality in the following form:

$$\mu_1\mu_2 \leq \frac{1}{2}(\alpha\mu_1^2 + \alpha^{-1}\mu_2^2)\,, \tag{16}$$

for any $\alpha, \mu_1, \mu_2 > 0$.

3. Further definitions. In order to define a σ-algebra on $\mathit{\Omega}$ we consider, for every $k \geq 0$ and every $\boldsymbol{\omega}^{0:k} \in \mathcal{N} \times \mathcal{D}$, the cylinder

$$C^k(\boldsymbol{\omega}^{0:k}) \triangleq \{\omega \in \mathit{\Omega} : \boldsymbol{\omega}_{0:k} = \boldsymbol{\omega}^{0:k}\},$$

i.e., $C^k(\boldsymbol{\omega}^{0:k})$ is the subset of $\mathit{\Omega}$ of all elements ω whose first k elements are equal to $\boldsymbol{\omega}^0, \ldots \boldsymbol{\omega}^k$. With a little abuse of notation, we indicate by $\boldsymbol{\omega}_k$ the k-th element of the sequence $\omega \in \mathit{\Omega}$. Let us now denote by $\mathcal{C}^k$ the set of all possible $C^k(\boldsymbol{\omega}^{0:k})$ when $\boldsymbol{\omega}^t$, $t = 0, \ldots, k$, takes all possible values. Denoting by $\sigma\left(\mathcal{C}^k\right)$ the sigma-algebra generated by $\mathcal{C}^k$, define for all k,

$$\mathcal{F}^k \triangleq \sigma\left(\mathcal{C}^k\right) \qquad \text{and} \qquad \mathcal{F} \triangleq \sigma\left(\cup_{t=0}^{\infty}\mathcal{C}^t\right). \tag{17}$$

We have $\mathcal{F}^k \subseteq \mathcal{F}^{k+1} \subseteq \mathcal{F}$ for all k. The latter inclusion is obvious, the former derives easily from the fact that any cylinder in $\mathcal{C}^{k-1}$ can be obtained as a finite union of cylinders in $\mathcal{C}^k$.

Finally, let us define the vectors $\mathbf{w}_{\mathbf{x}}^k$ such $[\mathbf{w}_{\mathbf{x}}^k]_i = [\hat{\mathbf{x}}(\tilde{\mathbf{x}}^k(\mathbf{d}_{j_{i,k}}))]_i$ where $j_{i,k}$ is defined to be $j_{i,k} = \text{argmax}_{j:\,(i,\mathbf{d}_j)\in\mathcal{V}(\omega)} \|\hat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d}_j)) - \mathbf{x}_i^k\|_2$. If the argmax is not a singleton, we just pick the first index among those satisfying the operation.

4. Inconsistent read. For any given $\omega \in \mathit{\Omega}$, recall that for simplicity of notation we define: $\tilde{\mathbf{x}}^k = \mathbf{x}^{k-\mathbf{d}^k}$. Since at each iteration only one block of variables is updated, it is not difficult to see that $\tilde{\mathbf{x}}^k$ can be written as

$$\tilde{\mathbf{x}}^k = \mathbf{x}^k + \sum_{l\in K(\mathbf{d}^k)} (\mathbf{x}^l - \mathbf{x}^{l+1}), \tag{18}$$

where $K(\mathbf{d}^k) \subseteq \{k-\delta, \ldots, k-1\}$ [cf. Assumption D1]. When we need to explicitly specify the dependence of $\tilde{\mathbf{x}}^k$ on a given realization $\mathbf{d}$ of $\underline{\mathbf{d}}^k$, we will write $\tilde{\mathbf{x}}^k(\mathbf{d})$.

5. Lemma. In order to prove the complexity result we will use the following lemma.

Lemma 1 *Set* $\gamma \leq \frac{(1-\rho^{-1})}{2(1+L_{\hat{x}}N(3+2\psi))}$. *Given* $\mathbf{x}^0$ *and* $\omega \in \bar{\mathit{\Omega}}$, *the sequence generated by the proposed algorithm under all the previous assumptions satisfies the following condition for any* $k \geq 0$*:*

$$\|\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1}\|_2^2 \leq \rho\|\mathbf{w}_{\mathbf{x}}^k - \mathbf{x}^k\|_2^2. \tag{19}$$

Proof The proof is done by induction and parallels, to some extent, a similar one presented in [12].
Let us consider a given $\omega \in \bar{\mathit{\Omega}}$ and $k \geq 0$.

We start by relying on the following trivial implication, which holds true for any two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$:

$$\|\mathbf{a}-\mathbf{b}\|_2^2 = \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 - 2\mathbf{a}^T\mathbf{b} \geq 0 \Longrightarrow \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 \geq 2\mathbf{a}^T\mathbf{b}, \tag{20}$$

that, combined with Cauchy-Schwartz inequality, leads to:

$$\begin{aligned}&\|\mathbf{a}\|_2^2 - \|\mathbf{b}\|_2^2 = 2\|\mathbf{a}\|_2^2 - (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2) \leq 2\|\mathbf{a}\|_2^2 - 2\mathbf{a}^{\mathrm{T}}\mathbf{b}\\ &= 2\mathbf{a}^{\mathrm{T}}(\mathbf{a}-\mathbf{b}) \leq 2\|\mathbf{a}\|_2\|\mathbf{b}-\mathbf{a}\|_2.\end{aligned} \tag{21}$$

The following holds:

$$\begin{aligned}&\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 - \|\gamma(\mathbf{w}_{\mathbf{x}}^{k} - \mathbf{x}^{k})\|_2^2\\ &\overset{(a)}{\leq} 2\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2\|\gamma(\mathbf{w}_{\mathbf{x}}^{k} - \mathbf{x}^{k}) - \gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2\\ &\overset{(b)}{\leq} 2\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2(\gamma\|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2 + \gamma L_{\hat{x}}\sum_{i=1}^{N}\|\tilde{\mathbf{x}}^k(\mathbf{d}_{j_{i,k}}) - \tilde{\mathbf{x}}^{k-1}(\mathbf{d}_{j_{i,k-1}})\|_2)\\ &\leq 2\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2(\gamma\|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2 + \gamma L_{\hat{x}}N\|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2\\ &+\gamma L_{\hat{x}}\sum_{i=1}^{N}(\|\mathbf{x}^k - \tilde{\mathbf{x}}^k(\mathbf{d}_{j_{i,k}})\|_2 + \|\mathbf{x}^{k-1} - \tilde{\mathbf{x}}^{k-1}(\mathbf{d}_{j_{i,k-1}})\|_2))\end{aligned} \tag{22}$$

where in (a) we used (21) and (b) comes from Assumption E. We will prove the Lemma by induction, so let us analyze what happens for $k = 1$. (22) simply becomes:

$$\begin{aligned}&\|\gamma(\mathbf{w}_{\mathbf{x}}^{0} - \mathbf{x}^{0})\|_2^2 - \|\gamma(\mathbf{w}_{\mathbf{x}}^{1} - \mathbf{x}^{1})\|_2^2 \leq 2\|\gamma(\mathbf{w}_{\mathbf{x}}^{0} - \mathbf{x}^{0})\|_2((1 + L_{\hat{x}}N)\gamma\|\mathbf{x}^1 - \mathbf{x}^0\|_2\\ &+ \gamma L_{\hat{x}}\sum_{i=1}^{N}(\|\mathbf{x}^1 - \tilde{\mathbf{x}}^1(\mathbf{d}_{j_{i,1}})\|_2 + \|\mathbf{x}^0 - \tilde{\mathbf{x}}^0(\mathbf{d}_{j_{i,0}})\|_2))\end{aligned} \tag{23}$$

For $k = 1$ and for any $j = 1, \dots, |\mathcal{V}(\omega)|$, we can bound the terms in (23) as:

$$\begin{aligned}&\|\mathbf{x}^1 - \tilde{\mathbf{x}}^1(\mathbf{d}_{j_{i,1}})\|_2 + \|\mathbf{x}^0 - \tilde{\mathbf{x}}^0(\mathbf{d}_{j_{i,0}})\|_2 \overset{(a)}{=} \|\mathbf{x}^1 - \tilde{\mathbf{x}}^1(\mathbf{d}_{j_{i,1}})\|_2 + \|\mathbf{x}^0 - \mathbf{x}^0\|_2\\ &\leq \|\mathbf{x}^1 - \mathbf{x}^0\|_2,\end{aligned} \tag{24}$$

where (a) comes from (18) and the fact that $K(\mathbf{d}^0) = \emptyset$ for any $\mathbf{d}^0 \in \mathcal{D}$ and $K(\mathbf{d}^1) \subseteq \{0\}$ for any $\mathbf{d}^1 \in \mathcal{D}$ (see the definition of the set $K(\mathbf{d}^k)$ in Section 6.1).

Substituting (24) in (23), we have:

$$\begin{aligned}&\|\gamma(\mathbf{w}_{\mathbf{x}}^{0} - \mathbf{x}^{0})\|_2^2 - \|\gamma(\mathbf{w}_{\mathbf{x}}^{1} - \mathbf{x}^{1})\|_2^2 \leq 2\gamma(1 + L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^{0} - \mathbf{x}^{0})\|_2\|\mathbf{x}^1 - \mathbf{x}^0\|_2\\ &+ 2\gamma L_{\hat{x}}N\|\gamma(\mathbf{w}_{\mathbf{x}}^{0} - \mathbf{x}^{0})\|_2\|\mathbf{x}^1 - \mathbf{x}^0\|_2 = 2\gamma(1 + 2L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^{0} - \mathbf{x}^{0})\|_2\|\mathbf{x}^1 - \mathbf{x}^0\|_2\\ &\overset{(a)}{\leq} \gamma(1 + 2L_{\hat{x}}N)(\|\gamma(\mathbf{w}_{\mathbf{x}}^{0} - \mathbf{x}^{0})\|_2^2 + \|\mathbf{x}^1 - \mathbf{x}^0\|_2^2)\\ &= \gamma(1 + 2L_{\hat{x}}N)(\|\gamma(\mathbf{w}_{\mathbf{x}}^{0} - \mathbf{x}^{0})\|_2^2 + \|\gamma(\hat{\mathbf{x}}_{i^0}(\tilde{\mathbf{x}}^0(\mathbf{d}^0)) - \mathbf{x}_{i^0}^0)\|_2^2)\end{aligned} \tag{25}$$

$$\leq 2\gamma(1+2L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^0-\mathbf{x}^0)\|_2^2,$$

where (a) follows from the Young's inequality with $\alpha=1$ and the last inequality follows from the definition of $\mathbf{w}_{\mathbf{x}}$ in Section 6.1. We can now derive the base of the induction we were seeking, just rearranging the terms in (25):

$$\|\gamma(\mathbf{w}_{\mathbf{x}}^0-\mathbf{x}^0)\|_2^2 \leq (1-2\gamma(1+2L_{\hat{x}}N))^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^1-\mathbf{x}^1)\|_2^2 \leq \rho\|\gamma(\mathbf{w}_{\mathbf{x}}^1-\mathbf{x}^1)\|_2^2. \tag{26}$$

The steps in (26) are valid only if, for some $\rho>1$:

$$\gamma \leq \frac{(1-\rho^{-1})}{2(1+2L_{\hat{x}}N)}. \tag{27}$$

In this way we proved the base of the induction. Now we start again from (22) in order to finish the proof of the Lemma. Let us search a bound for the following quantity for any $j=1,\dots,|\mathcal{V}(\omega)|$:

$$\begin{aligned}
&\|\mathbf{x}^k-\tilde{\mathbf{x}}^k(\mathbf{d}_{j_{i,k}})\|_2+\|\mathbf{x}^{k-1}-\tilde{\mathbf{x}}^{k-1}(\mathbf{d}_{j_{i,k-1}})\|_2\\
&\overset{(a)}{=}\|\sum_{l\in K^k(\mathbf{d}_j)}(\mathbf{x}^{l+1}-\mathbf{x}^l)\|_2+\|\sum_{l\in K^{k-1}(\mathbf{d}_j)}(\mathbf{x}^{l+1}-\mathbf{x}^l)\|_2\\
&\leq \sum_{l\in K^k(\mathbf{d}_{j_{i,k}})}\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2+\sum_{l\in K^{k-1}(\mathbf{d}_{j_{i,k-1}})}\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2\\
&\overset{(b)}{\leq}\sum_{l=k-\delta}^{k-1}\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2+\sum_{l=k-1-\delta}^{k-2}\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2\\
&\leq 2\sum_{l=k-1-\delta}^{k-1}\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2 \;=\; 2\sum_{l=k-1-\delta}^{k-1}\|\mathbf{x}_{i^l}^{l+1}-\mathbf{x}_{i^l}^l\|_2
\end{aligned} \tag{28}$$

where (a) comes from (18) and (b) from D1. Plugging this last result into (22):

$$\begin{aligned}
&\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1}-\mathbf{x}^{k-1})\|_2^2-\|\gamma(\mathbf{w}_{\mathbf{x}}^k-\mathbf{x}^k)\|_2^2\\
&\leq 2\gamma(1+L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1}-\mathbf{x}^{k-1})\|_2\|\mathbf{x}^k-\mathbf{x}^{k-1}\|_2\\
&+4\gamma L_{\hat{x}}N\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1}-\mathbf{x}^{k-1})\|_2\sum_{l=k-1-\delta}^{k-1}\|\mathbf{x}_{i^l}^{l+1}-\mathbf{x}_{i^l}^l\|_2\\
&=2\gamma(1+3L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1}-\mathbf{x}^{k-1})\|_2\|\mathbf{x}^k-\mathbf{x}^{k-1}\|_2\\
&+4\gamma L_{\hat{x}}N\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1}-\mathbf{x}^{k-1})\|_2\sum_{l=k-1-\delta}^{k-2}\|\mathbf{x}_{i^l}^{l+1}-\mathbf{x}_{i^l}^l\|_2.
\end{aligned} \tag{29}$$

By using Young's inequality with $\alpha_l>0$ for any l, we get:

$$\begin{aligned}
&\|\mathbf{x}_{i^l}^{l+1}-\mathbf{x}_{i^l}^l\|_2\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1}-\mathbf{x}^{k-1})\|_2 \;\leq\; \frac{1}{2}(\alpha_l\|\mathbf{x}_{i^l}^{l+1}-\mathbf{x}_{i^l}^l\|_2^2+\alpha_l^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1}-\mathbf{x}^{k-1})\|_2^2)\\
&=\frac{1}{2}(\alpha_l\|\gamma(\hat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l(\mathbf{d}^l))-\mathbf{x}_{i^l}^l)\|_2^2+\alpha_l^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1}-\mathbf{x}^{k-1})\|_2^2)
\end{aligned} \tag{30}$$

$$\le \frac{1}{2}(\alpha_l\|\gamma(\mathbf{w}_{\mathbf{x}}^l - \mathbf{x}^l)\|_2^2 + \alpha_l^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2)$$

Substituting (30) in (29) and using again Young's inequality with $\alpha = 1$, we get the following result:

$$\begin{aligned}
&\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 - \|\gamma(\mathbf{w}_{\mathbf{x}}^k - \mathbf{x}^k)\|_2^2 \\
&\le \gamma(1 + 3L_{\hat{x}}N)(\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 + \|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2^2) \\
&+ 2\gamma N L_{\hat{x}} \sum_{l=k-1-\delta}^{k-2} (\alpha_l\|\gamma(\mathbf{w}_{\mathbf{x}}^l - \mathbf{x}^l)\|_2^2 + \alpha_l^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2) \qquad (31) \\
&= \gamma(1 + 3L_{\hat{x}}N)(\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 + \|\gamma(\hat{\mathbf{x}}_{i^{k-1}}(\tilde{\mathbf{x}}^{k-1}(\mathbf{d}^{k-1})) - \mathbf{x}_{i^{k-1}}^{k-1})\|_2^2) \\
&+ 2\gamma N L_{\hat{x}} \sum_{l=k-1-\delta}^{k-2} (\alpha_l\|\gamma(\mathbf{w}_{\mathbf{x}}^l - \mathbf{x}^l)\|_2^2 + \alpha_l^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2) \\
&\le 2\gamma(1 + 3L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 \\
&+ 2\gamma N L_{\hat{x}} \sum_{l=k-1-\delta}^{k-2} (\alpha_l\|\gamma(\mathbf{w}_{\mathbf{x}}^l - \mathbf{x}^l)\|_2^2 + \alpha_l^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2)
\end{aligned}$$

Assuming the inductive step to hold true up to the step $\|\mathbf{w}_{\mathbf{x}}^{k-2} - \mathbf{x}^{k-2}\|_2^2 \le \rho\|\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1}\|_2^2$, we obtain:

$$\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 - \|\gamma(\mathbf{w}_{\mathbf{x}}^k - \mathbf{x}^k)\|_2^2 \ \le\ 2\gamma(1 + 3L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2$$

$$(32)$$

$$+ 2\gamma N L_{\hat{x}} \sum_{l=k-\delta-1}^{k-2} (\alpha_l \rho^{k-l-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 + \alpha_l^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2).$$

Setting $\alpha_l = (\rho^{\frac{1+l-k}{2}})^{-1}$ and noticing that $\sum\limits_{l=k-\delta-1}^{k-2} \rho^{\frac{k-l-1}{2}} = \sum\limits_{t=1}^{\delta} \rho^{\frac{t}{2}} = \psi$ we get:

$$\begin{aligned}
&\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 - \|\gamma(\mathbf{w}_{\mathbf{x}}^k - \mathbf{x}^k)\|_2^2 \le 2\gamma(1 + 3L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 \\
&+ 4\gamma N L_{\hat{x}} \sum_{l=k-\delta-1}^{k-2} \rho^{\frac{k-l-1}{2}}\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 \qquad (33) \\
&= 2\gamma(1 + 3L_{\hat{x}}N)\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 + 4\gamma N L_{\hat{x}}\psi\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2.
\end{aligned}$$

Rearranging the terms we get the desired result

$$\begin{aligned}
&\|\gamma(\mathbf{w}_{\mathbf{x}}^{k-1} - \mathbf{x}^{k-1})\|_2^2 \qquad (34) \\
&\le (1 - 2\gamma(1 + L_{\hat{x}}N(3 + 2\psi)))^{-1}\|\gamma(\mathbf{w}_{\mathbf{x}}^k - \mathbf{x}^k)\|_2^2 \ \le\ \rho\|\gamma(\mathbf{w}_{\mathbf{x}}^k - \mathbf{x}^k)\|_2^2.
\end{aligned}$$

(34) holds true if:

$$\gamma \le \frac{(1 - \rho^{-1})}{2(1 + L_{\hat{x}}N(3 + 2\psi))} \qquad (35)$$

with $\rho > 1$. Note that if we set: $\gamma \leq \frac{(1-\rho^{-1})}{2(1+L_{\hat{x}}N(3+2\psi))}$, both condition (27) and (35) are satisfied.

6.2 Proof of Theorem 1

Let us define:

$$\hat{\mathbf{y}}_i^k = \underset{\mathbf{y}_i \in \mathcal{K}_i(\mathbf{x}_i^k)}{\arg\min} \{\nabla_{\mathbf{x}_i} f(\mathbf{x}^k)^{\mathrm{T}}(\mathbf{y}_i - \mathbf{x}_i^k) + g_i(\mathbf{y}_i) + \frac{1}{2}\|\mathbf{y}_i - \mathbf{x}_i^k\|_2^2\} \tag{36}$$

and note that $M_F(\mathbf{x}) = [\mathbf{x}_1^k - \hat{\mathbf{y}}_1^k, \ldots, \mathbf{x}_N^k - \hat{\mathbf{y}}_N^k]^{\mathrm{T}}$. Let us consider in the following a given realization $\omega \in \bar{\Omega}$ and $k \geq 0$. Relying on the first order optimality conditions for $\hat{\mathbf{y}}_{i^k}^k$ and using convexity of g_{i^k} we can write the following inequality, that holds true for any $\mathbf{z}_{i^k} \in \mathcal{K}_{i^k}(\mathbf{x}_{i^k}^k)$:

$$(\nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k) + \hat{\mathbf{y}}_{i^k} - \mathbf{x}_{i^k}^k)^{\mathrm{T}}(\mathbf{z}_{i^k} - \hat{\mathbf{y}}_{i^k}) + g_{i^k}(\mathbf{z}_{i^k}) - g_{i^k}(\hat{\mathbf{y}}_{i^k}) \geq 0. \tag{37}$$

In a similar way we can use first order optimality condition for $\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)$ and convexity of g_{i^k}, obtaining:

$$\nabla \tilde{f}_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \tilde{\mathbf{x}}^k)^{\mathrm{T}}(\mathbf{w}_{i^k} - \hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)) + g_{i^k}(\mathbf{w}_{i^k}) - g_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k))) \geq 0\,, \tag{38}$$

that holds true for any $\mathbf{w}_i \in \mathcal{K}_{i^k}(\tilde{\mathbf{x}}_i^k) = \mathcal{K}_{i^k}(\mathbf{x}_i^k)$ (cf. D4). We can now sum up (37) and (38) together setting $\mathbf{w}_{i^k} = \hat{\mathbf{y}}_{i^k}$ and $\mathbf{z}_{i^k} = \hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)$:

$$(\nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k) - \nabla \tilde{f}_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \tilde{\mathbf{x}}^k) + \hat{\mathbf{y}}_{i^k} - \mathbf{x}_{i^k}^k)^{\mathrm{T}}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}) \geq 0. \tag{39}$$

Summing and subtracting $\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)$ inside the first parenthesis on the left hand side and using the gradient consistency assumption B2 we get:

$$(\nabla \tilde{f}_{i^k}(\mathbf{x}_{i^k}^k; \mathbf{x}^k) - \nabla \tilde{f}_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \tilde{\mathbf{x}}^k) + \hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k)^{\mathrm{T}}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}) \geq \|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}\|_2^2\,. \tag{40}$$

Applying Cauchy-Schwartz inequality to upper-bound the left hand side of (40) we obtain:

$$\|\nabla \tilde{f}_{i^k}(\mathbf{x}_{i^k}^k; \mathbf{x}^k) - \nabla \tilde{f}_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \tilde{\mathbf{x}}^k) + \hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2 \geq \|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}\|_2\,, \tag{41}$$

We can proceed summing and subtracting $\nabla \tilde{f}_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \mathbf{x}^k)$ inside the norm on the left hand side and then applying triangular inequality to get:

$$\begin{aligned} &\|\nabla \tilde{f}_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \mathbf{x}^k) - \nabla \tilde{f}_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \tilde{\mathbf{x}}^k)\|_2 \\ &+\|\nabla \tilde{f}_{i^k}(\mathbf{x}_{i^k}^k; \mathbf{x}^k) - \nabla \tilde{f}_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k); \mathbf{x}^k)\|_2 + \|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}_{i^k}^k\|_2 \\ &\geq \|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}\|_2, \end{aligned} \tag{42}$$

and we can further upper-bound the left hand side relying on the Lipschitz continuity assumptions B3 and B4:

$$\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}\|_2 \leq (1 + L_E)\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2 + L_B\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2\,. \tag{43}$$

Taking the square both sides we have:

$$\begin{aligned}\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}\|_2^2 \;\leq\; (1 + L_E)^2\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2 + L_B^2\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2\\ + 2L_B(1 + L_E)\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2.\end{aligned} \tag{44}$$

We are now interested in bounding $\|\mathbf{x}^k_{i^k} - \hat{\mathbf{y}}_{i^k}\|_2^2$:

$$\begin{aligned}&\|\mathbf{x}^k_{i^k} - \hat{\mathbf{y}}_{i^k}\|_2^2 = \|\mathbf{x}^k_{i^k} - \hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) + \hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}\|_2^2 &(45)\\ &\leq\; 2\left(\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2 + \|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \hat{\mathbf{y}}_{i^k}\|_2^2\right) \overset{(a)}{\leq} (2 + 2(1 + L_E)^2)\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2^2\\ &+ 2L_B^2\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2^2 + 4L_B(1 + L_E)\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k) - \mathbf{x}^k_{i^k}\|_2\|\mathbf{x}^k - \tilde{\mathbf{x}}^k\|_2\end{aligned}$$

where (a) comes from (44). Taking now conditional expectation and using D2:

$$\mathbb{E}(\|\mathbf{x}^k_{\underline{i}^k} - \hat{\mathbf{y}}_{\underline{i}^k}\|_2^2|\mathcal{F}^{k-1})(\omega) \;=\; \sum_{i=1}^N p(i|\omega^{0:k-1})\|\mathbf{x}^k_i - \hat{\mathbf{y}}_i\|_2^2 \;\geq\; p_{\min}\|\mathbf{x}^k - \hat{\mathbf{y}}\|_2^2, \tag{46}$$

It follows:

$$\begin{aligned}&p_{\min}\|\mathbf{x}^k - \hat{\mathbf{y}}\|_2^2\\ &\overset{(a)}{\leq} (2 + 2(1 + L_E)^2)\mathbb{E}(\|\hat{\mathbf{x}}_{\underline{i}^k}(\underline{\tilde{\mathbf{x}}}^k) - \mathbf{x}^k_{\underline{i}^k}\|_2^2|\mathcal{F}^{k-1})(\omega) + 2L_B^2\mathbb{E}(\|\mathbf{x}^k - \underline{\tilde{\mathbf{x}}}^k\|_2^2|\mathcal{F}^{k-1})(\omega)\\ &+ 4L_B(1 + L_E)\mathbb{E}(\|\hat{\mathbf{x}}_{\underline{i}^k}(\underline{\tilde{\mathbf{x}}}^k) - \mathbf{x}^k_{\underline{i}^k}\|_2\|\mathbf{x}^k - \underline{\tilde{\mathbf{x}}}^k\|_2|\mathcal{F}^{k-1})(\omega)\\ &\overset{(b)}{\leq} 2(1 + (1 + L_E)(1 + L_B + L_E))\|\mathbf{w}^k_{\mathbf{x}} - \mathbf{x}^k\|_2^2\\ &+ 2L_B(1 + L_B + L_E)\mathbb{E}(\|\mathbf{x}^k - \underline{\tilde{\mathbf{x}}}^k\|_2^2|\mathcal{F}^{k-1})(\omega),\end{aligned} \tag{47}$$

where in (a) we combined (45) and (46) together while (b) follows from the Young's inequality with $\alpha = 1$ and the definition of $\mathbf{w}_{\mathbf{x}}$.
In order to bound (47) we need the following result:

$$\begin{aligned}&\mathbb{E}(\|\mathbf{x}^k - \underline{\tilde{\mathbf{x}}}^k\|_2^2|\mathcal{F}^{k-1})(\omega) \;\overset{(a)}{\leq}\; \mathbb{E}\left(\left(\sum_{l=k-\delta}^{k-1}\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\right)^2|\mathcal{F}^{k-1}\right)(\omega)\\ &= \left(\sum_{l=k-\delta}^{k-1}\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\right)^2 \;\overset{(b)}{\leq}\; \delta\sum_{l=k-\delta}^{k-1}\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2 &(48)\\ &= \delta\gamma^2\sum_{l=k-\delta}^{k-1}\|\hat{\mathbf{x}}_{i^l}(\tilde{\mathbf{x}}^l) - \mathbf{x}^l_{i^l}\|_2^2 \;\leq\; \delta\gamma^2\sum_{l=k-\delta}^{k-1}\|\mathbf{w}^l_{\mathbf{x}} - \mathbf{x}^l\|_2^2 \;\overset{(c)}{\leq}\; \delta\psi'\gamma^2\|\mathbf{w}^k_{\mathbf{x}} - \mathbf{x}^k\|_2^2\end{aligned}$$

where (a) follows from (18); (b) from the Jensen's inequality; and (c) from Lemma 1.

The left hand side of (47) is nothing else than $\|M_F(\mathbf{x}^k)\|_2^2$, which is our optimality measure; substituting (48) into (47):

$$p_{\min}\|M_F(\mathbf{x}^k)\|_2^2 \leq 2(1+(1+L_B+L_E)(1+L_E L_B \delta\psi'\gamma^2))\|\mathbf{w}_{\mathbf{x}}^k-\mathbf{x}^k\|_2^2. \tag{49}$$

We now need a bound for the quantity $\|\mathbf{w}_{\mathbf{x}}^k-\mathbf{x}^k\|_2^2$. For any $k \geq 0$ we have:

$$\begin{aligned}
&F(\mathbf{x}^{k+1})\\
&= f(\mathbf{x}^{k+1}) + g(\mathbf{x}^{k+1})\\
&\overset{(a)}{=} f(\mathbf{x}^{k+1}) + \sum_{i\neq i^k} g_i(\mathbf{x}_i^{k+1}) + g_{i^k}(\mathbf{x}_{i^k}^{k+1})\\
&\overset{(b)}{=} f(\mathbf{x}^{k+1}) + \sum_{i\neq i^k} g_i(\mathbf{x}_i^{k}) + g_{i^k}(\mathbf{x}_{i^k}^{k+1})\\
&\overset{(c)}{\leq} f(\mathbf{x}^k) + \gamma\nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k)^{\mathrm{T}}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\mathbf{x}_{i^k}^k) + \tfrac{\gamma^2 L_f}{2}\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\mathbf{x}_{i^k}^k\|_2^2\\
&+ \sum_{i\neq i^k} g_i(\mathbf{x}_i^{k}) + g_{i^k}(\mathbf{x}_{i^k}^{k+1})\\
&= f(\mathbf{x}^k) + \gamma\nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k)^{\mathrm{T}}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\mathbf{x}_{i^k}^k)\\
&+(\nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k)-\nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k))^{\mathrm{T}}(\gamma(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\mathbf{x}_{i^k}^k)) + \tfrac{\gamma^2 L_f}{2}\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\mathbf{x}_{i^k}^k\|_2^2\\
&+ \sum_{i\neq i^k} g_i(\mathbf{x}_i^{k}) + g_{i^k}(\mathbf{x}_{i^k}^{k+1})\\
&\overset{(d)}{\leq} f(\mathbf{x}^k) + \gamma\nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k)^{\mathrm{T}}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\tilde{\mathbf{x}}_{i^k}^k)\\
&+(\nabla_{\mathbf{x}_{i^k}} f(\mathbf{x}^k)-\nabla_{\mathbf{x}_{i^k}} f(\tilde{\mathbf{x}}^k))^{\mathrm{T}}(\gamma(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\tilde{\mathbf{x}}_{i^k}^k)) + \tfrac{\gamma^2 L_f}{2}\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\tilde{\mathbf{x}}_{i^k}^k\|_2^2\\
&+ \sum_{i\neq i^k} g_i(\mathbf{x}_i^{k}) + \gamma g_{i^k}(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)) + g_{i^k}(\mathbf{x}_{i^k}^{k}) - \gamma g_{i^k}(\tilde{\mathbf{x}}_{i^k}^k)\\
&\overset{(e)}{\leq} F(\mathbf{x}^k) - \gamma(c_{\tilde{f}} - \tfrac{\gamma L_f}{2})\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\tilde{\mathbf{x}}_{i^k}^k\|_2^2 + L_f\|\mathbf{x}^k-\tilde{\mathbf{x}}^k\|_2\|\gamma(\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\tilde{\mathbf{x}}_{i^k}^k)\|_2\\
&\overset{(f)}{\leq} F(\mathbf{x}^k) - \gamma(c_{\tilde{f}} - \gamma L_f)\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\tilde{\mathbf{x}}_{i^k}^k\|_2^2 + \tfrac{L_f}{2}\|\mathbf{x}^k-\tilde{\mathbf{x}}^k\|_2^2\\
&\overset{(g)}{=} F(\mathbf{x}^k) - \gamma(c_{\tilde{f}} - \gamma L_f)\|\hat{\mathbf{x}}_{i^k}(\tilde{\mathbf{x}}^k)-\mathbf{x}_{i^k}^k\|_2^2 + \tfrac{L_f}{2}\|\mathbf{x}^k-\tilde{\mathbf{x}}^k\|_2^2,
\end{aligned} \tag{50}$$

where in (a) we used the separability of g; (b) follows from the updating rule of the algorithm; in (c) we applied the Descent Lemma [2] on f; (d) comes from the convexity of g_i and D4; in (e) we used Proposition 1 and Assumption A3; (f) is due to the Young's inequality, with $\alpha = 1$; and (g) comes from D4.

Taking conditional expectations both sides, we have:

$$\begin{aligned}
&\mathbb{E}(F(\underline{\mathbf{x}}^{k+1})|\mathcal{F}^{k-1})(\omega)\\
&\leq F(\mathbf{x}^k) - \gamma(c_{\tilde{f}} - \gamma L_f)\mathbb{E}(\|\hat{\mathbf{x}}_{\underline{i}^k}(\underline{\tilde{\mathbf{x}}}^k) - \mathbf{x}^k_{\underline{i}^k}\|_2^2|\mathcal{F}^{k-1})(\omega) + \tfrac{L_f}{2}\mathbb{E}(\|\mathbf{x}^k - \underline{\tilde{\mathbf{x}}}^k\|_2^2|\mathcal{F}^{k-1})(\omega)\\
&\overset{(a)}{\leq} F(\mathbf{x}^k) - \gamma(c_{\tilde{f}} - \gamma L_f)\sum_{(i,\mathbf{d})\in\mathcal{V}(\omega)} p((i,\mathbf{d})|\omega^{0:k-1})\|\hat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d})) - \mathbf{x}_i^k\|_2^2\\
&+\tfrac{\delta\psi'\gamma^2 L_f}{2}\|\mathbf{w}^k_{\mathbf{x}} - \mathbf{x}^k\|_2^2\\
&\overset{(b)}{\leq} F(\mathbf{x}^k) - \Delta\gamma(c_{\tilde{f}} - \gamma L_f)\sum_{i=1}^N \|\hat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(\mathbf{d}_{j_{i,k}})) - \mathbf{x}_i^k\|_2^2 + \tfrac{\delta\psi'\gamma^2 L_f}{2}\|\mathbf{w}^k_{\mathbf{x}} - \mathbf{x}^k\|_2^2\\
&= F(\mathbf{x}^k) - \gamma\Delta(c_{\tilde{f}} - \gamma L_f)\|\mathbf{w}^k_{\mathbf{x}} - \mathbf{x}^k\|_2^2 + \tfrac{\delta\psi'\gamma^2 L_f}{2}\|\mathbf{w}^k_{\mathbf{x}} - \mathbf{x}^k\|_2^2\\
&\overset{(c)}{=} F(\mathbf{x}^k) - \gamma\left(\Delta(c_{\tilde{f}} - \gamma L_f) - \tfrac{\gamma\delta\psi' L_f}{2}\right)\|\mathbf{w}^k_{\mathbf{x}} - \mathbf{x}^k\|_2^2,
\end{aligned} \tag{51}$$

where (a) follows from (48); (b) comes from Assumption D3 and holds true for $\gamma \leq \frac{c_{\tilde{f}}}{L_f}$; in (c) we require: $\gamma \leq \frac{2\Delta c_{\tilde{f}}}{2\Delta L_f + \delta\psi' L_f}$. Taking expectations both sides of (51) and rearranging the terms, we get:

$$\mathbb{E}(\|\underline{\mathbf{w}}^k_{\mathbf{x}} - \underline{\mathbf{x}}^k\|_2^2) \leq \frac{2}{\gamma(2\Delta(c_{\tilde{f}} - \gamma L_f) - \gamma\delta\psi' L_f)}\mathbb{E}(F(\underline{\mathbf{x}}^k) - F(\underline{\mathbf{x}}^{k+1})). \tag{52}$$

Using this result in (49) we finally have:

$$\mathbb{E}(\|M_F(\mathbf{x}^k)\|_2^2) \leq \frac{4(1 + (1 + L_B + L_E)(1 + L_E L_B \delta\psi'\gamma^2))}{p_{\min}\gamma(2\Delta(c_{\tilde{f}} - \gamma L_f) - \gamma\delta\psi' L_f)}\mathbb{E}(F(\underline{\mathbf{x}}^k) - F(\underline{\mathbf{x}}^{k+1})), \tag{53}$$

and:

$$\begin{aligned}
K_\epsilon\epsilon &\leq \sum_{k=0}^{K_\epsilon}\mathbb{E}(\|M_F(\mathbf{x}^k)\|_2^2)\\
&\leq \sum_{k=0}^{K_\epsilon}\tfrac{4(1+(1+L_B+L_E)(1+L_E L_B\delta\psi'\gamma^2))}{p_{\min}\gamma(2\Delta(c_{\tilde{f}}-\gamma L_f)-\gamma\delta\psi' L_f)}\mathbb{E}(F(\underline{\mathbf{x}}^k) - F(\underline{\mathbf{x}}^{k+1}))\\
&= \tfrac{4(1+(1+L_B+L_E)(1+L_E L_B\delta\psi'\gamma^2))}{p_{\min}\gamma(2\Delta(c_{\tilde{f}}-\gamma L_f)-\gamma\delta\psi' L_f)}\mathbb{E}(F(\mathbf{x}^0) - F(\underline{\mathbf{x}}^{K_\epsilon+1}))\\
&\leq \tfrac{4(1+(1+L_B+L_E)(1+L_E L_B\delta\psi'\gamma^2))}{p_{\min}\gamma(2\Delta(c_{\tilde{f}}-\gamma L_f)-\gamma\delta\psi' L_f)}(F(\mathbf{x}^0) - F^*).
\end{aligned} \tag{54}$$

This completes the proof.