Randomized Block Proximal Methods for Distributed Stochastic Big-Data Optimization

Francesco Farina, *Member, IEEE,* Giuseppe Notarstefano *Member, IEEE*

***Abstract*—In this paper we introduce a class of novel distributed algorithms for solving stochastic big-data convex optimization problems over directed graphs. In the addressed set-up, the dimension of the decision variable can be extremely high and the objective function can be nonsmooth. The general algorithm consists of two main steps: a consensus step and an update on a single block of the optimization variable, which is then broadcast to neighbors. Three special instances of the proposed method, involving particular problem structures, are then presented. In the general case, the convergence of a dynamic consensus algorithm over random row stochastic matrices is shown. Then, the convergence of the proposed algorithm to the optimal cost is proven in expected value. Exact convergence is achieved when using diminishing (local) stepsizes, while approximate convergence is attained when constant stepsizes are employed. The convergence rate is shown to be sublinear and an explicit rate is provided in the case of constant stepsizes. Finally, the algorithm is tested on a distributed classification problem, first on synthetic data and, then, on a real, high-dimensional, text dataset.**

I. Introduction

Recent years have witnessed a steadily growing interest in distributed learning and control over networks consisting of multiple smart agents. Several problems arising in this scenario can be formulated as distributed optimization problems which need to be solved by networks of agents. In this paper, we focus on the following stochastic *big-data* convex optimization problem, which is to be solved over a network of N interconnected agents,

$$\underset{x\in X}{\text{minimize}} \quad \sum_{i=1}^{N} \mathbb{E}[h_i(x;\xi_i)],$$

where $X \subseteq \mathbb{R}^n$ is a convex set, $\xi_i \in \mathbb{R}$ is a random variable and the functions $h_i : \mathbb{R}^n \to \mathbb{R}$ are continuous, convex and possibly non smooth. The optimization variable x is extremely high dimensional and with block structure, i.e., $n = \sum_{\ell=1}^{B} n_\ell$ with n_ℓ being the dimension of the ℓ-th block and $B \gg 1$ the number of blocks. Regarding the role of stochastic functions in the considered set-up, it is worth stressing that they allow agents to deal with various type of problems. Among the others, the case of learning problems involving massive datasets is of particular interest. In this case, the local objective function typically has the form $f_i(x) = \frac{1}{m_i}\sum_{r=1}^{m_i} h_i(x,\xi_i^r)$, where ξ_i^r, $r = 1,\dots,m_i$, are samples uniformly drawn from a certain dataset consisting of m_i elements. When m_i is very large it could be computationally infeasible to compute a subgradient of the entire f_i. On the other side, given ξ_i^r, computing a subgradient of $h_i(x,\xi_i^r)$ is much simpler. Problems of this type are often referred to as sample average approximation problems [2]. Other relevant classes of problems include those of dynamic, or online, optimization problems in which samples generating functions h_i are processed as they become available [3, 4] and settings in which only noisy subgradients of the objective functions are available [5].

Applying classical distributed algorithms to big-data problems may be infeasible due, e.g., to limitations in the communication bandwidth. In fact, they would require agents to communicate a prohibitive amount of data due to the high dimension of the decision variable. This calls for tailored distributed algorithms for big-data optimization problems in which only few blocks of the entire (local) solution estimate are sent to neighbors. Thus, the literature relevant to this paper can be divided in three main (partially overlapping) categories: stochastic optimization methods, block coordinate algorithms and primal distributed algorithms.

Stochastic optimization algorithms: To the best of our knowledge, the first work dealing with stochastic problems has been [6]. Since this seminal work, there has been a steady increase in the interest for this type of problems, and algorithms for solving them (see, e.g., [7] and references therein). Among the others, stochastic approximation approaches were presented in [8, 9] and stochastic mirror descent algorithms have been studied in [10, 11]. Stochastic gradient descent algorithms are particularly appealing in learning problems (see, e.g., [12]) in which extremely large datasets are involved, since they allow for batch processing of the data.

Block coordinate algorithms: Centralized block coordinate methods have a long history (see, e.g., [13] for a survey). They were firstly designed for solving smooth problems, but, in the last years, an increasing number of results have been

F. Farina is in the Artificial Intelligence and Machine Learning group at GSK. However, this work was carried out while the author was at the Department of Electrical, Electronic and Information Engineering "G. Marconi", Università di Bologna, Bologna, Italy. email: francesco.x.farina@gsk.com

G. Notarstefano is with the Department of Electrical, Electronic and Information Engineering "G. Marconi", Università di Bologna, Bologna, Italy. email: giuseppe.notarstefano@unibo.it

A preliminary version of this work has appeared in the Proceedings of the 58-th Control and Decision Conference (CDC 2019) [1]. The current article consider a more general problem set-up, namely a constrained stochastic optimization one. Also the proposed algorithm is more general since local updates are based on generic proximal mappings, agents can be awake or idle at each iteration, blocks can be drawn according to locally defined (possibly non uniform) probability distributions and local stepsize sequences can be employied. Furthermore, the convergence analysis is also carried out under the assumption of constant stepsizes and an explicit convergence rate is provided. Finally, all the complete theoretical proofs are reported.

This result is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART).

provided to deal with nonsmooth objective functions. Two main rules for selecting the block to be updated have been studied: cyclic (or almost cyclic; see, e.g., [14]) or random. In the last case, randomized block coordinate algorithms have been proposed [15, 16, 17, 18, 19]. Particularly relevant for this paper is the work in [11], in which a stochastic block mirror descent method with random block updates is proposed. Parallel block coordinate methods are also a well established strand of optimization literature, see, e.g., [20]. The work in [21] applies to smooth convex functions, while the ones in [22, 23, 24] face up composite optimization problems. A unified framework for nonsmooth optimization using block algorithms has been studied in [25] for centralized and parallel set-ups.

Distributed algorithms: Many distributed optimization algorithms have been proposed in recent years. In [26] a distributed gradient descent algorithm was firstly introduced, which is capable to deal with both deterministic and stochastic convex optimization problems. When the problems to be solved involve nonsmooth objective functions, subgradient-based algorithms have been designed. First examples of such algorithms appeared in [27, 28, 29], while recent advances involve more sophisticated protocols, to deal with directed communication [30, 31, 32, 33, 34]. Many distributed algorithms involving proximal operations have also been proposed (see, e.g.,[35] for a survey on proximal algorithms). Among the others, a proximal gradient method was developed in [36] to deal with unconstrained problems, while in [37, 38] proximal algorithms have been presented to deal with constrained optimization. The stochastic setting has also been treated [5, 39, 40, 41, 42, 43, 44]. In particular, a stochastic subgradient projection algorithm appeared in [5], while a stochastic distributed mirror descent was proposed in [44]. Distributed algorithms over random networks are also relevant to this paper. In [45], consensus protocols were studied using random row-stochastic matrices, while in [46] a distributed subgradient method over random networks with underlying doubly stochastic matrices has been proposed. Distributed algorithms dealing with block communication have started to appear only recently. A block gradient tracking scheme has been presented in [47] for nonconvex problems with nonsmooth regularizers, while [48] proposes an asynchronous algorithm for nonconvex optimization based on the method of multipliers, which is implementable block-wise. A randomized block-coordinate algorithm for smooth problems with common cost function and linear constraints has been presented in [49].

In this paper, we introduce the Distributed Block Proximal Method, which models a class of distributed proximal algorithms, with block communication, for solving stochastic big-data convex optimization problems with nonsmooth objective function. The communication network is modeled as a directed graph admitting a doubly stochastic weight matrix. At each iteration, each node is awake with a certain probability (and idle otherwise). If awake, it performs a consensus step, computes a stochastic subgradient of a local objective function, and performs a proximal-based update (depending on the computed subgradient and on a local stepsize) on a randomly chosen block only. Then, it exchanges with its neighbors only the updated block of the decision variable, thus requiring a small amount of communication bandwidth. We also present three special instances of the proposed algorithm. In the first one, the proximal mapping is based on the squared 2-norm, thus leading to explicit block subgradient steps. In the other two, smooth objective functions and separable (possibly nonsmooth) ones are considered. In both these cases the computational load at each node in the network can be further reduced with respect to the general algorithm. We point out that no global parameter is required in the evolution of the algorithms. In fact, each node is awake and selects blocks with *locally defined* probabilities, and uses *local* stepsizes. The block-wise updates and the communication of a single block induce nontrivial technical challenges in the algorithm analysis. On this regard, it is worth noting that, despite the double stochasticity of the weight matrix, the consensus step on each block turns out to be performed using a sequence of random row-stochastic matrices. The analysis for the Distributed Block Proximal Method is carried out in two parts. First, the convergence properties of a dynamic block consensus protocol over random graphs are studied, by building on block-wise, perturbed consensus dynamics with random matrices. A bound on the expected distance from consensus is provided, which is then specialized to the cases of constant and diminishing stepsizes respectively. Then, a bound on the expected distance from the (globally) optimal cost is provided by properly bounding errors due to the block-wise update and exploiting the probability of drawing blocks. When constant stepsizes are used, approximate convergence (with a constant error term) to the optimal cost is proven in expected value, while asymptotic exact convergence is reached for diminishing stepsizes. Finally, we provide an explicit convergence rate for the proposed algorithm when using constant stepsizes. The rate is sublinear, even though a linear term is present, which can be predominant in the first iterations.

The paper is organized as follows. The problem set-up is introduced in Section II along with some preliminary results. In Section III, the Distributed Block Proximal Method is presented and three special algorithm instances are given in Section IV. Then, the algorithm is analyzed in Section V. Finally, a numerical example involving a distributed classification problem over a syntetic and a real, high-dimensional, text document datasets is dispensed in Section VI and some conclusions are drawn in Section VII.

II. Set-up and preliminaries

A. Notation and definitions

Given a vector $x \in \mathbb{R}^n$, we denote by x_ℓ the ℓ-th block of x, i.e., given a partition of the identity matrix $I = [U_1, \dots, U_B]$, with $U_\ell \in \mathbb{R}^{n \times n_\ell}$ for all ℓ and $\sum_{\ell=1}^B n_\ell = n$, it holds $x = \sum_{\ell=1}^B U_\ell x_\ell$ and $x_\ell = (U_\ell)^\top x$. Moreover we denote by $\|x\|$ the 2-norm of x. Given a vector $x \in \mathbb{R}^n$, with scalar blocks, we define

$$d(x) \triangleq \max_{1 \le \ell \le n} x_\ell - \min_{1 \le \ell \le n} x_\ell.$$

Given a vector $x_i \in \mathbb{R}^n$, we denote by $x_{i,\ell}$ the ℓ-th block of x_i. Moreover, given a constant c, and an index t, we denote by $(c)^t$, c to the power of t, while given a sequence $\{x^t\}_{t\geq 0}$, we denote by x^t the t-th element of the sequence. Given a matrix A, we denote by a_{ij} (or $[A]_{ij}$) the element of A located at row i and column j. Given two matrices A and B, we write $A \geq B$ if $a_{ij} \geq b_{ij}$ for all i and j. Given two vectors $a, b \in \mathbb{R}^n$ we denote by $\langle a, b\rangle$ their scalar product. Given a discrete random variable $r \in \{1, \dots, R\}$, we denote by $P(r = \bar{r})$ the probability of r to be equal to $\bar{r}$. Given a nonsmooth function f, we denote by $\partial f(x)$ its subdifferential computed at x, and by $\partial_{x_\ell} f(x)$ the subdifferential of f with respect to the ℓ-th block of x.

We say that a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ contains a spanning tree if for some $v \in \mathcal{V}$ there exists a directed path from the vertex v to all other vertices $u \in \mathcal{V}$. Given a nonnegative matrix A and some $\delta \in (0, 1)$, we denote by A_δ the matrix whose entries are defined as

$$[A_\delta]_{ij} = \begin{cases} \delta, & \text{if } A_{ij} \geq \delta, \\ 0, & \text{otherwise.} \end{cases}$$

We say that A contains a δ-spanning tree if the graph induced by A_δ contains a spanning tree.

B. Distributed stochastic optimization set-up

As anticipated in the introduction, we consider the following optimization problem,

$$\underset{x\in X}{\text{minimize}} \quad \sum_{i=1}^{N} \mathbb{E}[h_i(x; \xi_i)]. \tag{1}$$

We recall that ξ_i is a random variable, functions $h_i : \mathbb{R}^n \to \mathbb{R}$ are continuous, convex and possibly nonsmooth for every ξ_i, $X \subseteq \mathbb{R}^n$ and $n \gg 1$. We let $f_i(x) = \mathbb{E}[h_i(x; \xi_i)]$ and $f(x) = \sum_{i=1}^{N} f_i(x)$. Moreover, $x^\star \in \mathbb{R}^n$ is a solution of problem (1). The optimization variable $x \in \mathbb{R}^n$ has a block structure, i.e.,

$$x = [x_1^\top, \dots, x_B^\top]^\top.$$

with $x_\ell \in \mathbb{R}^{n_\ell}$ for all ℓ and $\sum_{\ell=1}^{B} n_\ell = n$. We make the following assumption on the problem structure

Assumption 1 (Problem structure)**.**

(A) The constraint set X has the block structure

$$X = X_1 \times \cdots \times X_B,$$

where, for $\ell = 1, \dots, B$, the set $X_\ell \subseteq \mathbb{R}^{n_\ell}$ is closed and convex, and $\sum_{\ell=1}^{B} n_\ell = n$.

(B) Let $g_i(x; \xi_i) \in \partial h_i(x; \xi_i)$ (resp. $g_i(x) \in \partial f_i(x)$) be a subgradient of $h_i(x; \xi_i)$ (resp. $f_i(x)$) computed at x. Then, $g_i(x; \xi_i)$ is an unbiased estimator of the subgradient of f_i, i.e.,

$$\mathbb{E}[g_i(x; \xi_i)] = g_i(x).$$

(C) There exist constants $G_i \in [0, \infty)$ and $\bar{G}_i \in [0, \infty)$ such that

$$\mathbb{E}[\|g_i(x; \xi_i)\|] \leq G_i, \qquad \mathbb{E}[\|g_i(x; \xi_i)\|^2] \leq \bar{G}_i,$$

for all x and ξ_i, for all $i \in \{1, \dots, N\}$. □

Notice that, if $X = \mathbb{R}^n$, Assumption 1(A) is clearly satisfied. Moreover, let us denote by $g_{i,\ell}(x; \xi_i)$ the ℓ-th block of $g_i(x; \xi_i)$ and let $g(x) \in \partial f(x)$ be a subgradient of f computed at x. Then, Assumption 1(C) implies that $\mathbb{E}[\|g_{i,\ell}(x; \xi_i)\|] \leq G_i$ for all $\ell \in \{1, \dots, B\}$ and $\|g_i(x)\| \leq G_i$. Moreover, let $\bar{G} \triangleq \sum_{i=1}^{N} \bar{G}_i$ and $G \triangleq \sum_{i=1}^{N} G_i$. Then, $\|g(x)\| \leq G$ and $\|g_i(x)\| \leq G$ for all i.

Problem (1) is to be solved in a distributed way by a network of N agents. Each agent in the network is assumed to know only a portion of the entire problem, namely agent i knows f_i and the constraint set X only. We make the following assumption on the network structure.

Assumption 2 (Communication structure)**.**

(A) The network is modeled through a weighted *strongly connected* directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ with $\mathcal{V} = \{1, \dots, N\}$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and $W \in \mathbb{R}^{N\times N}$ being the weighted adjacency matrix. We denote by $\mathcal{N}_{i,out}$ the set of out-neighbors of node i, i.e., $\mathcal{N}_{i,out} \triangleq \{j \mid (i, j) \in \mathcal{E}\} \cup \{i\}$. Similarly, the set of in-neighbors of node i is defined as $\mathcal{N}_{i,in} \triangleq \{j \mid (j, i) \in \mathcal{E}\} \cup \{i\}$.

(B) For all $i, j \in \{1, \dots, N\}$, the weights w_{ij} of the weight matrix W satisfy

(i) if $i \neq j$, $w_{ij} > 0$ if and only if $j \in \mathcal{N}_{i,in}$;

(ii) there exists a constant $\eta > 0$ such that $w_{ii} \geq \eta$ and if $w_{ij} > 0$, then $w_{ij} \geq \eta$;

(iii) $\sum_{j=1}^{N} w_{ij} = 1$ and $\sum_{i=1}^{N} w_{ij} = 1$. □

A function ω_ℓ is associated to the ℓ-th block of the optimization variable for all ℓ. Let the function $\omega_\ell : X_\ell \to \mathbb{R}$, be continuously differentiable and σ_ℓ-strongly convex. Functions ω_ℓ are sometimes referred to as distance generating functions. Then, we define the proximal function, also called *Bregman's divergence*, associated to ω_ℓ as

$$\nu_\ell(a, b) = \omega_\ell(b) - \omega_\ell(a) - \langle \nabla\omega_\ell(a), b - a\rangle,$$

for all $a, b \in X_\ell$. The following assumption is made on the functions ν_ℓ.

Assumption 3 (Bregman's divergence separate convexity)**.** For all $\ell \in \{1, \dots, B\}$, the function ν_ℓ satisfies

$$\nu_\ell\left(\sum_{j=1}^{N} \theta_j a_j, b\right) \leq \sum_{j=1}^{N} \theta_j \nu_\ell(a_j, b), \quad \forall a_1, \dots, a_N, b \in X_\ell, \tag{2}$$

where $\sum_{j=1}^{N} \theta_j = 1$ and $\theta_j \geq 0$ for all j. □

Notice that the above assumption is satisfied by many functions (such as the quadratic function, the Boltzmann-Shannon entropy and the exponential function) and conditions on the functions ω_ℓ guaranteeing (2) can be provided (see [50]). Finally, given $a \in X_\ell$, $b \in \mathbb{R}^{n_\ell}$ and $c \in \mathbb{R}$, the proximal mapping associated to ν_ℓ is defined as

$$\text{prox}_\ell(a, b, c) = \arg\min_{u\in X_\ell} \left\{ \langle b, u\rangle + \frac{1}{c}\nu_\ell(a, u) \right\}. \tag{3}$$

C. Preliminary results

Consider a stochastic, discrete-time dynamical system evolving according to

$$x^{t+1} = A^t x^t, \quad \forall t, \tag{4}$$

where $\{A^t\}_{t\geq 0}$ is a sequence of random $n \times n$ row-stochastic matrices. Let $(\Omega, \mathcal{F}, P)$ be a probability space. We assume that the sequence $\{A^t, \mathcal{S}^t\}_{t\geq 0}$ forms an *adapted process*, i.e., $\{A^t\}_{t\geq 0}$ is a stochastic process defined on $(\Omega, \mathcal{F}, P)$, $\{\mathcal{S}^t\}_{t\geq 0}$ is a filtration (i.e., $\mathcal{S}^t \subseteq \mathcal{S}^{t+1}$ and $\mathcal{S}^t \subseteq \mathcal{F}$ for all t) and A^t is measurable with respect to $\mathcal{S}^t$. Given a sequence of matrices $\{A^t\}_{t\geq 0}$, let us define the transition matrix from iteration s to iteration t as

$$\Phi_A^{t,s} \triangleq \begin{cases} A^t A^{t-1} \dots A^s, & \text{if } t > s, \\ A^t, & \text{if } t = s. \end{cases}$$

Then, the following result, adapted from [45, Theorem 3.1], holds true for system (4).

Lemma 1 ([45, Theorem 3.1])**.** *Consider system* (4)*. If there exist $h > 0$, $\delta > 0$ such that $\mathbb{E}[\sum_{t=mh+1}^{(m+1)h} A^t \mid \mathcal{S}^{mh}]$ contains a δ-spanning tree for each m, and $A^t \geq \delta I$ for each t, then, for any given initial distribution of x^0 with $\mathbb{E}[\|x^0\|^p] < \infty$ (which is independent of $\{A^t\}_{t\geq 0}$), and any $p > 0$, it holds*

$$\begin{aligned} \mathbb{E}[d(x^t)^p] &= \mathbb{E}[d(\Phi_A^{t,0} x^0)^p] \\ &\leq (\mu)^t \mathbb{E}[d(x^0)^p] \leq M(\mu)^t \mathbb{E}[\|x^0\|^p], \end{aligned}$$

where $M \in (0, \infty)$ and $\mu \in (0, 1)$. □

Finally, the following three results will be useful in the rest of the paper.

Lemma 2. *Given a scalar $\beta \neq 1$, it holds that*

(i) for any $t \geq r \geq 0$, $\sum_{s=r}^{t} (\beta)^s = \frac{(\beta)^r - (\beta)^{t+1}}{1-\beta}$

(ii) for any $t \geq 0$, $\sum_{s=0}^{t} \sum_{\tau=0}^{t-s} (\beta)^\tau = \frac{t+1-\beta(t+2)+(\beta)^{t+2}}{(1-\beta)^2}$ □

Lemma 3 ([5, Lemma 3.1])**.** *Let $\{\gamma^t\}_{t\geq 0}$ be a scalar sequence.*

(i) If $\lim_{t\to\infty} \gamma^t = \gamma$ and $\beta \in (0, 1)$ then $\lim_{t\to\infty} \sum_{s=0}^{t} (\beta)^{t-s} \gamma^s = \frac{\gamma}{1-\beta}$.

(ii) If $\gamma^t \geq 0$ $\forall t$, $\sum_{t=0}^{\infty} \gamma^t < \infty$ and $\beta \in (0, 1)$, then $\sum_{t=0}^{\infty} \left(\sum_{s=0}^{t} (\beta)^{t-s} \gamma^s \right) < \infty$. □

Lemma 4 (Tower property of conditional expectation)**.** *Let X be a random variable defined on a probability space $(\Omega, \mathcal{F}, P)$. Let $\mathcal{Z} \subseteq \mathcal{Y} \subseteq \mathcal{F}$. Then, $\mathbb{E}[\mathbb{E}[X \mid \mathcal{Y}, \mathcal{Z}] \mid \mathcal{Z}] = \mathbb{E}[X \mid \mathcal{Z}]$.* □

III. Distributed Block Proximal Method

The Distributed Block Proximal Method for solving problem (1) in a distributed way is now introduced. The algorithm works as follows. Each agent i maintains a local solution estimate x_i^t and a local copy of the estimates of its in-neighbors. Let us denote by $x_j^t|_i$ the copy of the solution estimate of agent j at agent i. At the beginning, each node initializes its state with a random (bounded) initial condition x_i^0 which is then shared with its neighbors. At each iteration each agent i is awake with probability $p_{i,on} \in (0, 1]$ and idle with probability $1 - p_{i,on}$. Thus, the proposed algorithm models a particular type of asynchrony in which the communication graph is fixed and agents can communicate or not with their neighbors with a certain probability. If agent i is awake, it picks randomly a block $\ell_i^t \in \{1, \dots, B\}$, some ξ_i^t, and performs two updates:

(i) it computes a weighted average of its in-neighbors' estimates $x_j^t|_i$, $j \in \mathcal{N}_{i,in}$;

(ii) it computes x_i^{t+1} by updating the ℓ_i^t-th block of x_i^t through a proximal mapping step and leaving the other blocks unchanged.

Then, it broadcasts $x_{i,\ell_i^t}^{t+1}$ to its out-neighbors. We model the status (awake or idle) of each node i at each iteration t through a random variable $s_i^t \in \{0, 1\}$ which is 1 (corresponding to being awake) with probability $p_{i,on}$ and 0 with probability $1 - p_{i,on}$. A pseudocode of the method is reported in Algorithm 1.

Algorithm 1 Distributed Block Proximal Method

Initialization: x_i^0

Evolution: for $t = 0, 1, \dots$

UPDATE for all $j \in \mathcal{N}_{i,in}$

$$x_{j,\ell}^t|_i = \begin{cases} x_{j,\ell}^t, & \text{if } \ell = \ell_j^{t-1} \text{ and } s_j^{t-1} = 1 \\ x_{j,\ell}^{t-1}|_i, & \text{otherwise} \end{cases} \tag{5}$$

if $s_i^t = 1$ **then**

PICK $\ell_i^t \in \{1, \dots, B\}$ with $P(\ell_i^t = \ell) = p_{i,\ell} > 0$, $\forall \ell$

COMPUTE

$$y_i^t = \sum_{j \in \mathcal{N}_{i,in}} w_{ij} x_j^t|_i \tag{6}$$

UPDATE

$$x_{i,\ell}^{t+1} = \begin{cases} \text{prox}_\ell(y_{i,\ell}^t, g_{i,\ell}(y_i^t; \xi_i^t), \alpha_i^t), & \text{if } \ell = \ell_i^t \\ x_{i,\ell}^t, & \text{otherwise} \end{cases} \tag{7}$$

BROADCAST $x_{i,\ell_i^t}^{t+1}$ to $j \in \mathcal{N}_{i,out}$

else $x_i^{t+1} = x_i^t$

Notice that all the quantities involved in the above algorithm are local for each node. In fact, each node has locally defined probabilities (both of awakening and block drawing) and local stepsizes.

Moreover, it is worth noting that, despite node i receives from each $j \in \mathcal{N}_i^{in}$ only the block $x_{j,\ell_j^{t-1}}^t$, the consensus step (6) is in fact performed by using the entire x_j^t. Indeed, the other blocks have not changed since the last time they have been received. This is formalized in the next result.

Lemma 5. *Let Assumption 2 hold. Then $x_j^t|_i = x_j^t$ for all t. Moreover, Algorithm 1 can be compactly rewritten as follows. For all $i \in \{1, \dots, N\}$ and all t, if $s_i^t = 1$,*

$$y_i^t = \sum_{j=1}^{N} w_{ij} x_j^t, \tag{8}$$

$$x_{i,\ell}^{t+1} = \begin{cases} \text{prox}_\ell(y_{i,\ell}^t, g_{i,\ell}(y_i^t; \xi_i^t), \alpha_i^t), & \textit{if } \ell = \ell_i^t, \\ x_{i,\ell}^t, & \textit{otherwise}, \end{cases} \tag{9}$$

else, $x_i^{t+1} = x_i^t$.

Proof. The fact that $x_j^t|_i = x_j^t$ for all i and all t follows immediately from the evolution of the algorithm. In fact, the

received block $x^t_{j,\ell^{t-1}_j}|_i$ is the only block that node j has modified in the last iteration, while the others have remained unchanged. Hence, since the graph $\mathcal{G}$ is fixed, it is clear that $x^t_j|_i = x^t_j$ for all i and all t. The reformulation of Algorithm 1 as (8)-(9) is then immediate from Assumption 2(B). □

In virtue of the previous result, in order to lighten the notation in the subsequent analysis, we will use (8)-(9) in place of Algorithm 1, by making the block communication implicit.

As for the block-wise proximal update (9), the ℓ^t_i-th block of a *whole* stochastic subgradient computed at y^t_i is used. Unfortunately, computing a subgradient with respect to the ℓ^t_i-th component only is, in general, *not* equivalent to picking the ℓ^t_i-th block of a *whole* subgradient $g_i(y^t_i;\xi^t_i)$. In fact, in general it holds that, picking $g_1 \in \partial_{y_{i,1}} h_i(y^t_i;\xi^t_i)$,..., $g_B \in \partial_{y_{i,B}} h_i(y^t_i;\xi^t_i)$ does not imply $[g_i^\top,\dots,g_B^\top]^\top \in \partial h_i(y^t_i;\xi^t_i)$. This will turn out to be extremely important in the subsequent analysis. If functions f_i are separable on the blocks, then, only the subgradient with respect to the ℓ^t_i-th component can be computed. Similarly, if the functions f_i are smooth, the ℓ^t_i-th block of the gradient can be directly computed as the gradient with respect to that block. In these cases, the computational load at each node can be further reduced, as it will be shown in Section IV.

The last key feature of the Distributed Block Proximal Method involves the consensus step (8). Let z^t_ℓ be the vector stacking the ℓ-th component of all the x^t_i, i.e., $z^t_\ell \triangleq [(x^t_{1,\ell})^\top,\dots,(x^t_{N,\ell})^\top]^\top$. Also, let D^t_ℓ be a diagonal matrix in which the i-th element of the diagonal is set to 1 if $s^t_i = 1$ and $\ell^t_i = \ell$, and it is set to 0 otherwise, i.e.,

$$[D^t_\ell]_{ij} = \begin{cases} 1, & \text{if } i=j,\ \ell=\ell^t_i \text{ and } s^t_i = 1,\\ 0, & \text{otherwise.}\end{cases}$$

Finally, let $D^t_{-\ell} = I - D^t_\ell$. Now, consider a consensus protocol associated to the Distributed Block Proximal Method, i.e.,

$$y^t_i = \sum_{j=1}^N w_{ij} x^t_j,$$

$$x^{t+1}_{i,\ell} = \begin{cases} y^t_{i,\ell}, & \text{if } \ell=\ell^t_i \text{ and } s^t_i = 1,\\ x^t_{i,\ell}, & \text{otherwise.}\end{cases}$$

This system can be rewritten in terms of z_ℓ as

$$z^{t+1}_\ell = W^t_\ell z^t_\ell,$$

where $W^t_\ell \triangleq D^t_{-\ell} + D^t_\ell W$. It can be easily verified that, for all ℓ and t, the matrix W^t_ℓ is row-stochastic but not doubly stochastic anymore (unless all nodes select the same block ℓ at some iteration t).

Remark 1. It is worth noting that the proposed algorithm, besides being easy to implement, considers challenges that cannot be addressed by other block-wise distributed algorithms [47, 48, 49]. In particular, none of those works deals with stochastic problems. Moreover, in [47] composite objective functions with non-smooth components are considered but the non-smooth part must be common to all the agents. In [48, 49] at least differentiability of the objective is required. Finally, in our algorithm, all the algorithm parameters are local.

IV. Special instances

In this section, three special cases of the Distributed Block Proximal Method are presented. The first one is obtained by choosing the squared 2-norm as distance generating function, while the other two result from smooth and separable objective functions respectively.

A. Distributed Block Subgradient Method

By using $\omega_\ell(x) = \frac{1}{2}\|x\|^2$ for all ℓ, and assuming $X = \mathbb{R}^n$, the proximal mapping (3) has an explicit analytical solution and the update step (7) becomes

$$x^{t+1}_{i,\ell} = \begin{cases} y^t_{i,\ell} - \alpha^t_i g_{i,\ell}(y^t_i;\xi^t_i), & \text{if } \ell = \ell^t_i\\ x^t_{i,\ell}, & \text{otherwise.}\end{cases} \tag{10}$$

Notice that, the proximal step becomes a subgradient step on a single block of the optimization variable. Thus, we call Distributed Block Subgradient Method the resulting algorithm, i.e., the one obtained by replacing (7) with (10) in Algorithm 1. Notice that, in this case, it holds that, for all ℓ, the strong convexity parameter is $\sigma_\ell = 1$, thus resulting in special bounds in the subsequent algorithm analysis.

B. Smooth functions

The update of the solution estimate in (7) requires, in general, for node i at iteration t, the computation of an entire stochastic subgradient at the point y^t_i. However, only the ℓ^t_i-th block of the computed subgradient is used in the update step. When a function h_i is smooth, however, the ℓ^t_i-th block of its gradient can be directly computed as the gradient of h_i with respect to the ℓ^t_i-th block of the optimization variable and (7) can be replaced by

$$x^{t+1}_{i,\ell} = \begin{cases} \text{prox}_\ell(y^t_{i,\ell}, \nabla_\ell h_i(y^t_i;\xi^t_i), \alpha^t_i), & \text{if } \ell = \ell^t_i,\\ x^t_{i,\ell}, & \text{otherwise,}\end{cases} \tag{11}$$

where $\nabla_\ell h_i$ denotes the (partial) gradient of h_i with respect to the ℓ-th block of the optimization variable. Thus, when smooth functions are involved in the problem, the computational load can be reduced by avoiding the computation of the entire (sub)gradient.

C. Separable functions

When functions $h_i(x;\xi_i)$ are separable, i.e.,

$$h_i(x;\xi_i) = \sum_{\ell=1}^B \hat{h}_{i,\ell}(x_\ell,\xi_i),$$

the Distributed Block Proximal Method can be further simplified, allowing for an extra reduction of the computational load at each iteration at a given node. In fact, it holds that $\partial h_i(y^t_i;\xi^t_i) = \partial_{y_{i,1}} h_i(y^t_i;\xi^t_i) \times \dots \partial_{y_{i,B}} h_i(y^t_i;\xi^t_i)$, and hence $g_i(y^t_i;\xi^t_i) = \sum_{\ell=1}^B \hat{g}_{i,\ell}(y^t_{i,\ell},\xi^t_i)$, where $\hat{g}_{i,\ell} \in \partial\hat{h}_{i,\ell}$ is a subgradient of $\hat{h}_{i,\ell}$. This implies that $g_{i,\ell}(y^t_i,\xi^t_i) = \hat{g}_{i,\ell}(y^t_{i,\ell},\xi^t_i)$ and, thus, only the ℓ^t_i-th block of y^t_i is needed in order to compute $g_{i,\ell}(y^t_i,\xi^t_i)$ and hence x^{t+1}_i. Thus the Distributed Block Proximal Method can be simplified by allowing nodes

with a separable function to reduce their computational load. In particular, assume the cost function of node i to be separable. Then, a single block of y_i^t can be updated at each iteration and a subgradient can be directly computed for the corresponding block, without computing an entire subgradient. Hence, the algorithm can be rewritten, by using the equivalent formulation in Lemma 5, as follows. If $s_i^t = 1$,

$$y_{i,\ell}^t = \begin{cases} \sum_{j=1}^N w_{ij} x_{j,\ell}^t, & \text{if } \ell = \ell_i^t, \\ y_{i,\ell}^{t-1}, & \text{else,} \end{cases} \tag{12}$$

$$x_{i,\ell}^{t+1} = \begin{cases} \text{prox}_\ell(y_{i,\ell}^t, \hat{g}_{i,\ell}(y_{i,\ell}^t, \xi_i^t), \alpha_i^t), & \text{if } \ell = \ell_i^t, \\ x_{i,\ell}^t, & \text{otherwise,} \end{cases} \tag{13}$$

else, $x_i^{t+1} = x_i^t$.

V. Algorithm analysis

In this section, the convergence of the Distributed Block Proximal Method is proven in expected value. The proof consists of two main parts. In the first one the consensus of the agents' solution estimates is shown, while in the second one convergence towards the optimal cost is proven. Both results are given, at first, in a general form and, then, specialized to the case of constant stepsizes (in which convergence to a neighborhood is proven) and diminishing stepsizes (in which exact asymptotic convergence is reached).

Define $a^t \triangleq [\alpha_1^t, \dots, \alpha_N^t]^\top$, $a_M^t \triangleq \max_i \alpha_i^t$ and $a_m^t \triangleq \min_i \alpha_i^t$. We summarize in the following two assumptions, the two different choices for the stepsize sequences we consider in the following analysis.

Assumption 4 (Constant stepsize)**.** The sequences $\{\alpha_i^t\}_{t\geq 0}$ satisfy $\alpha_i^t = \alpha_i > 0$ for all t and all i.

Assumption 5 (Diminishing stepsize)**.** The sequences $\{\alpha_i^t\}_{t\geq 0}$ satisfy

$$\sum_{t=0}^\infty \alpha_i^t = \infty, \qquad \sum_{t=0}^\infty (\alpha_i^t)^2 < \infty,$$

for all $i \in \{1, \dots, N\}$. Moreover, $\alpha_i^{t+1} \leq \alpha_i^t$ for all t and all $i \in \{1, \dots, N\}$.

Notice that, under Assumption 4, $a_M^t \triangleq a_M = \max_i \alpha_i$ and $a_m^t \triangleq a_m = \min_i \alpha_i$ for all t, while, under Assumption 5 it can be easily verified that

$$\sum_{t=0}^\infty a_M^t = \infty, \qquad \sum_{t=0}^\infty (a_M^t)^2 < \infty, \qquad a_M^{t+1} \leq a_M^t,$$

and

$$\sum_{t=0}^\infty a_m^t = \infty, \qquad \sum_{t=0}^\infty (a_m^t)^2 < \infty, \qquad a_m^{t+1} \leq a_m^t.$$

Define the vector stacking all local solution estimates as $\boldsymbol{x}(t) \triangleq [(x_1(t))^\top, \dots, (x_N(t))^\top]^\top$, and the average (over the agents) of the local estimates at t as

$$\bar{x}(t) \triangleq \frac{1}{N} \sum_{i=1}^N x_i(t). \tag{14}$$

Then, we make the following assumption on the random variables involved in the algorithm.

Assumption 6 (Random variables)**.**

(A) For a given $i \in \{1, \dots, N\}$, the random variables ℓ_i^t and s_i^t are independent and identically distributed for all t.
(B) For a given t, the random variables s_i^t, ℓ_i^t and ξ_i^t are independent of each other for all $i \in \{1, \dots, N\}$.
(C) There exist constants $C_i \in [0, \infty)$ such that $\mathbb{E}[\|x_i^0\|] \leq C_i$ for all $i \in \{1, \dots, N\}$ and hence $\mathbb{E}[\|\boldsymbol{x}^0\|] \leq C = \sum_{i=1}^N C_i$. □

Before proceeding with the algorithm analysis, let us provide a preliminary instrumental result. Define $q_i^t = x_{i,\ell_i^t}^{t+1} - y_{i,\ell_i^t}^t$ and $q^t = [(q_1^t)^\top, \dots, (q_N^t)^\top]^\top$. Then, the following result applies.

Lemma 6. *Let Assumptions 1(A) and 1(C) hold. Then,*

$$\mathbb{E}[\|q_i^t\|] \leq \frac{G_i}{\sigma} \alpha_i^t,$$

for all $i \in \{1, \dots, N\}$, where $\sigma = \min_\ell \sigma_\ell$.

Proof. The first order necessary optimality condition on (9) for $\ell = \ell_i^t$ reads

$$\langle \alpha_i^t g_{i,\ell_i^t}(y_i^t; \xi_i^t) + \nabla \omega_{\ell_i^t}(x_{i,\ell_i^t}^{t+1}) - \nabla \omega_{\ell_i^t}(y_{i,\ell_i^t}^t), u - x_{i,\ell_i^t}^{t+1} \rangle \geq 0, \tag{15}$$

for all $u \in X_{\ell_i^t}$. Notice now that, by definition, $y_{i,\ell_i^t}^t \in X_{\ell_i^t}$, since it is a weighted average of points lying in $X_{\ell_i^t}$. Thus, by taking $u = y_{i,\ell_i^t}^t$, one obtains

$$\begin{aligned} &\alpha_i^t \langle g_{i,\ell_i^t}(y_i^t; \xi_i^t), y_{i,\ell_i^t}^t - x_{i,\ell_i^t}^{t+1} \rangle \\ &\quad \geq \langle \nabla \omega_{\ell_i^t}(y_{i,\ell_i^t}^t) - \nabla \omega_{\ell_i^t}(x_{i,\ell_i^t}^{t+1}), y_{i,\ell_i^t}^t - x_{i,\ell_i^t}^{t+1} \rangle \\ &\quad \geq \sigma_{\ell_i^t} \|y_{i,\ell_i^t}^t - x_{i,\ell_i^t}^{t+1}\|^2, \end{aligned} \tag{16}$$

where we have used the strong convexity of $\omega_{\ell_i^t}$. By rearranging the terms, one has

$$\begin{aligned} \sigma_{\ell_i^t} \|y_{i,\ell_i^t}^t - x_{i,\ell_i^t}^{t+1}\|^2 &\leq \alpha_i^t \langle g_{i,\ell_i^t}(y_i^t; \xi_i^t), y_{i,\ell_i^t}^t - x_{i,\ell_i^t}^{t+1} \rangle \\ &\leq \alpha_i^t \|g_{i,\ell_i^t}(y_i^t; \xi_i^t)\| \|y_{i,\ell_i^t}^t - x_{i,\ell_i^t}^{t+1}\| \end{aligned} \tag{17}$$

and hence,

$$\|q_i^t\| \leq \frac{\alpha_i^t}{\sigma_{\ell_i^t}} \|g_{i,\ell_i^t}(y_i^t; \xi_i^t)\| \leq \frac{\alpha_i^t}{\sigma} \|g_{i,\ell_i^t}(y_i^t; \xi_i^t)\|.$$

Now, by taking the expected value and using the subgradient boundedness from Assumption 1(C), one gets

$$\mathbb{E}[\|q_i^t\|] \leq \frac{\alpha_i^t}{\sigma} \mathbb{E}[\|g_{i,\ell_i^t}(y_i^t; \xi_i^t)\|] \leq \frac{\alpha_i^t G_i}{\sigma},$$

thus concluding the proof. □

A. Dynamic consensus with random matrices

In this section we show that the sequences $\{x_i^t\}_{t\geq 0}$ and $\{y_i^t\}_{t\geq 0}$ computed by each agent in the network asymptotically achieve consensus in expected value when using diminishing stepsizes. Moreover, an upper bound on the distance from consensus is provided in the case of constant stepsizes.

Let $\mathcal{S}^t \triangleq \{\boldsymbol{x}^\tau \mid \tau \in \{0, \dots, t\}\}$ be the set of estimates generated by the Distributed Block Proximal Method up to

iteration t (which is indeed a filtration). Moreover, define the probability of node i to both be awake and pick block ℓ at each iteration as

$$\pi_{i,\ell} \triangleq p_{i,on} p_{i,\ell}.$$

Then, the following lemma provides a bound on the expected distance between x_i^t and the average $\bar{x}^t$ (defined in (14)).

Lemma 7. *Let Assumptions 1(C), 2, 6 hold. Then, there exist constants $M \in (0, \infty)$ and $\mu_M \in (0, 1)$ such that*

$$\mathbb{E}[\|x_i^t - \bar{x}^t\|] \leq MB\Bigg((\mu_M)^{t-1} C + \frac{G}{\sigma} \sum_{s=0}^{t-2} (\mu_M)^{t-s-2} a_M^s + \frac{G}{\sigma} a_M^{t-1} \Bigg), \quad (18)$$

for all $i \in \{1, \dots, N\}$ and all $t \geq 1$.

Proof. For the sake of presentation, assume that the blocks are scalars, i.e., $B = n$. Let us recall that z_ℓ^t defines the vector stacking the ℓ-th component of all the x_i^t, i.e., $z_\ell^t \triangleq [x_{1,\ell}^t, \dots, x_{N,\ell}^t]^\top$, while the matrix $D_\ell^t \in \mathbb{R}^{N \times N}$ is a diagonal matrix in which the i-th element of the diagonal is set to 1 if $s_i^t = 1$ and $\ell_i^t = \ell$ and it is set to 0 otherwise, i.e.,

$$[D_\ell^t]_{ij} = \begin{cases} 1, & \text{if } i = j,\ \ell = \ell_i^t \text{ and } s_i^t = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Consistently, we let $D_{-\ell}^t = I - D_\ell^t$. Notice that, for all t, D_ℓ^t is a random matrix whose diagonal element $[D_\ell^t]_{ii}$ is 1 with probability $\pi_{i,\ell}$ and 0 with probability $1 - \pi_{i,\ell}$. Define

$$\Pi_\ell \triangleq \text{diag}\{[\pi_{1,\ell}, \dots, \pi_{N,\ell}]\}.$$

Then, by using Assumption 6(A), it can be verified that

$$\mathbb{E}[D_\ell^t \mid \mathcal{S}^t] = \mathbb{E}[D_\ell^t] = \Pi_\ell \quad (19)$$

and, similarly,

$$\mathbb{E}[D_{-\ell}^t \mid \mathcal{S}^t] = \mathbb{E}[D_{-\ell}^t] = I - \Pi_\ell \quad (20)$$

for all t.

Now, Algorithm 1 can be rewritten with respecte to z_ℓ as

$$z_\ell^{t+1} = W_\ell^t z_\ell^t + e_\ell^t, \quad (21)$$

where $W_\ell^t \triangleq D_{-\ell}^t + D_\ell^t W$ is, by definition, a row-stochastic matrix, and $e_\ell^t \triangleq D_\ell^t q^t$. Now, by recursively applying (21), it holds that

$$z_\ell^{t+1} = \Phi_{W_\ell}^{t,0} z_\ell^0 + \sum_{s=0}^{t-1} \Phi_{W_\ell}^{t,s+1} e_\ell^s + e_\ell^t,$$

where $\Phi_{W_\ell}^{t,s}$ is the transition matrix from iteration s to iteration t associated to the matrices W_ℓ^τ, $\tau = s, \dots, t$. Moreover, by applying the $d(\cdot)$ operator on both sides (recall that $d(z_\ell) = \max_{1 \leq i \leq N} z_{\ell,i} - \min_{1 \leq i \leq N} z_{\ell,i}$), one has

$$d(z_\ell^{t+1}) \leq d(\Phi_{W_\ell}^{t,0} z_\ell^0) + \sum_{s=0}^{t-1} d(\Phi_{W_\ell}^{t,s+1} e^s) + d(e^t). \quad (22)$$

Notice now that, by using (19) and (20),

$$\begin{aligned} \mathbb{E}\left[W_\ell^t\right] &= \mathbb{E}[D_{-\ell}^t] + \mathbb{E}[D_\ell^t] W \\ &= I - \Pi_\ell + \Pi_\ell W \end{aligned}$$

for all t. It can be seen that such a matrix is row stochastic and contains a δ-spanning tree with $\delta \geq (\eta \min_i \pi_{i,\ell} + \min_i (1 - \pi_{i,\ell})) > 0$ (since from Assumption 2 the matrix W contains a spanning tree), where η is defined in Assumption 2(B). Moreover, by Assumption 6(A), $\{W_\ell^t\}_{t \geq 0}$ is a sequence of i.i.d. random matrices with $W_\ell^t \geq \eta I$. Hence, from Lemma 1, by taking the expectation on both sides of (22), we get

$$\begin{aligned} \mathbb{E}[d(z_\ell^{t+1})] &\leq \mathbb{E}[d(\Phi_{W_\ell}^{t,0} z_\ell^0)] + \sum_{s=0}^{t-1} \mathbb{E}[d(\Phi_{W_\ell}^{t,s+1} e_\ell^s)] + \mathbb{E}[d(e_\ell^t)] \\ &\leq (\mu_\ell)^t \mathbb{E}[d(z_\ell^0)] + \sum_{s=0}^{t-1} (\mu_\ell)^{t-s-1} \mathbb{E}[d(e_\ell^s)] + \mathbb{E}[d(e_\ell^t)] \\ &\leq M \left((\mu_\ell)^t \mathbb{E}[\|z_\ell^0\|] + \frac{G}{\sigma} \sum_{s=0}^{t-1} (\mu_\ell)^{t-s-1} a_M^s + \frac{G}{\sigma} a_M^t \right). \end{aligned}$$

where we used the fact that, from Lemma 6,

$$\mathbb{E}[\|e_\ell^t\|] \leq \mathbb{E}[\|q^t\|] \leq \sum_{i=1}^N \mathbb{E}[\|q_i^t\|] \leq \frac{G}{\sigma} a_M^t.$$

Let us now define

$$\bar{z}_\ell^t \triangleq \frac{1}{N} \sum_{i=1}^N z_{\ell,i}^t.$$

Since $\min_j z_{\ell,j}^t \leq \bar{z}_\ell^t \leq \max_j z_{\ell,j}^t$, for all t, we have that

$$|z_{\ell,i}^t - \bar{z}_\ell^t| \leq \max_j z_{\ell,j}^t - \min_j z_{\ell,j}^t$$

for all $i \in \{1, \dots, N\}$. Notice now that, by definition $x_{i,\ell}^t = z_{\ell,i}^t$ and $\bar{x}_\ell^t = \bar{z}_\ell^t$. Hence,

$$\begin{aligned} &\mathbb{E}[|x_{i,\ell}^t - \bar{x}_\ell^t|] \\ &\leq M \left((\mu_\ell)^{t-1} \mathbb{E}[\|z_\ell^0\|] + \frac{G}{\sigma} \sum_{s=0}^{t-2} (\mu_\ell)^{t-s-2} a_M^s + \frac{G}{\sigma} a_M^{t-1} \right). \end{aligned}$$

Finally, since $\|x_i^t - \bar{x}^t\| \leq \sum_{\ell=1}^B |x_{i,\ell}^t - \bar{x}_\ell^t|$, one has

$$\begin{aligned} \mathbb{E}[\|x_i^t - \bar{x}^t\|] &\leq \sum_{\ell=1}^B M \Bigg((\mu_\ell)^{t-1} \mathbb{E}[\|z_\ell^0\|] + \frac{G}{\sigma} \sum_{s=0}^{t-2} (\mu_\ell)^{t-s-2} a_M^s + \frac{G}{\sigma} a_M^{t-1} \Bigg) \\ &\leq MB \Bigg((\mu_M)^{t-1} \mathbb{E}[\|\boldsymbol{x}^0\|] + \frac{G}{\sigma} \sum_{s=0}^{t-2} (\mu_M)^{t-s-2} a_M^s + \frac{G}{\sigma} a_M^{t-1} \Bigg), \end{aligned}$$

where $\mu_M = \max_\ell \mu_\ell$. The proof is concluded by using Assumption 6(C). □

Moreover, the expected value of the distance between y_i^t and x_i^t can be bounded, by exploiting the convexity of the norm and using Lemma 7, as stated in the next result.

Lemma 8. *Let Assumptions 1(C), 2, 6 hold. Then,*

$$\mathbb{E}[\|y_i^t - x_i^t\|] \leq 2MB\Bigg((\mu_M)^{t-1}C + \frac{G}{\sigma}\sum_{s=0}^{t-2}(\mu_M)^{t-s-2}a_M^s + \frac{G}{\sigma}a_M^{t-1}\Bigg)$$

for all $i \in \{1,\dots,N\}$ and all $t \geq 1$.

Proof. Form the definition of y_i^t and using the convexity of the norm, one has

$$\begin{aligned}\|y_i^t - x_i^t\| &= \|\sum_{j=1}^N w_{ij}x_j^t - x_i^t\| \\ &\leq \sum_{j=1}^N w_{ij}\|x_j^t - x_i^t\| \\ &\leq \sum_{j=1}^N w_{ij}\left(\|x_j^t - \bar{x}^t\| + \|x_i^t - \bar{x}^t\|\right).\end{aligned}$$

By taking the expected value on both sides and using Lemma 7, the proof follows by noting that $\sum_{j=1}^N w_{ij} = 1$ from Assumption 2(B). □

1) Constant stepsize: The following two results respectively provide an upper bound on the distance of x_i^t from $\bar{x}^t$ as $t \to \infty$ and characterize the quantity $\sum_{\tau=0}^t \mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|]$ for each t in the case of constant stepsizes.

Lemma 9. *Let Assumptions 1(C), 2, 4, 6 hold. Then, there exist constants $M \in (0,\infty)$ and $\mu_M \in (0,1)$ such that*

$$\lim_{t\to\infty}\mathbb{E}[\|x_i^t - \bar{x}^t\|] \leq \bar{S}$$

for all $i \in \{1,\dots,N\}$, with

$$\bar{S} = a_M\frac{MBG}{\sigma}\frac{2-\mu_M}{1-\mu_M}. \tag{23}$$

Proof. Equation (18) in Lemma 7 consists of three terms. For the first one, $\lim_{t\to\infty}(\mu_M)^{t-1}C = 0$, since $\mu_M < 1$. For the second term $\sum_{s=0}^{t-2}(\mu_M)^{t-s-2}a_M^s$, by Assumption 4 and Lemma 3, one has that $\lim_{t\to\infty}\sum_{s=0}^{t-2}(\mu_M)^{t-s-2}a_M^s = \frac{a_M}{1-\mu_M}$. The proof is completed by noting that, under Assumption 4 the last term is constant. □

Lemma 10. *Let Assumptions 1(C), 2, 4, 6 hold. Then,*

$$\sum_{\tau=0}^t\mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] \leq (\mu_M)^t\bar{R} + t\bar{S} + \bar{Q}$$

for all $i \in \{1,\dots,N\}$, with

$$\bar{R} = MB\left(\frac{a_M G}{\sigma(1-\mu_M)^2} - \frac{C}{1-\mu_M}\right), \tag{24}$$

$$\bar{Q} = C - a_M\frac{MBG}{\sigma}\frac{1}{(1-\mu_M)^2}. \tag{25}$$

Proof. By using Assumption 6(C), for $\tau = 0$, one has

$$\begin{aligned}\mathbb{E}[\|x_i^0 - \bar{x}^0\|] &\leq \mathbb{E}[\|x_i^0\|] + \mathbb{E}[\|\bar{x}^0\|] \\ &\leq C_i + \frac{1}{N}\sum_{j=1}^N C_j \leq C_i + \max_j C_j \leq C\end{aligned} \tag{26}$$

Hence, $\sum_{\tau=0}^t \mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] \leq C + \sum_{\tau=1}^t \mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|]$ and, from Lemma 7, we have

$$\begin{aligned}\sum_{\tau=0}^t\mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] &\leq C + \sum_{\tau=1}^t MB\Bigg((\mu_M)^{\tau-1}C \\ &\quad + \frac{G}{\sigma}\sum_{s=0}^{\tau-2}(\mu_M)^{\tau-s-2}a_M + \frac{G}{\sigma}a_M\Bigg) \\ &= C + MBC\sum_{\tau=0}^{t-1}(\mu_M)^\tau \\ &\quad + \frac{MBGa_M}{\sigma}\sum_{\tau=2}^t\sum_{s=0}^{\tau-2}(\mu_M)^{\tau-s-2} \\ &\quad + \sum_{\tau=0}^{t-1}\frac{MBGa_M}{\sigma}\end{aligned}$$

where in the last line we have rearranged the summations. Now, by noting that

$$\sum_{\tau=2}^t\sum_{s=0}^{\tau-2}(\mu_M)^{\tau-s-2} = \sum_{\tau=0}^{t-2}\sum_{s=0}^{t-2-s}(\mu_M)^s$$

and using Lemma 2 the result follows through straightforward manipulations. □

Notice that in virtue of Lemma 8, by using the same reasoning used in the previous two results it is possible to show that

$$\lim_{t\to\infty}\mathbb{E}[\|y_i^t - x_i^t\|] \leq 2\bar{S}$$

and

$$\sum_{\tau=0}^t\mathbb{E}[\|y_i^\tau - x_i^\tau\|] \leq 2\Big((\mu_M)^t\bar{R} + t\bar{S} + \bar{Q})\Big).$$

These bounds will be used in the following optimality analysis.

2) Diminishing stepsize: When adopting diminishing stepsizes, asymptotic (exact) consensus can be reached as stated in the following result.

Lemma 11. *Let Assumptions 1(C), 2, 5, 6 hold. Then,*

$$\lim_{t\to\infty}\mathbb{E}[\|x_i^t - \bar{x}^t\|] = 0$$

for all $i \in \{1,\dots,N\}$.

Proof. The proof is based on the same arguments as in Lemma 9. By noting that $a_M^t \to 0$ as $t \to \infty$ and that, under Assumption 5, from Lemma 3, $\lim_{t\to\infty}\sum_{s=0}^{t-2}(\mu_M)^{t-s-2}a_M^s = 0$, the result follows. □

The next result shows that $\lim_{t\to\infty}\sum_{\tau=0}^t a_m^\tau\mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|]$ is a summable series for all $i \in \{1,\dots,N\}$. Notice that this result does not hold in the case of constant stepsizes.

Lemma 12. *Let Assumptions 1(C), 2, 5, 6 hold. Then,*

$$\lim_{t\to\infty}\sum_{\tau=0}^t a_m^\tau\mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] < \infty$$

for all $i \in \{1,\dots,N\}$.

Proof. As for $\tau = 0$, from (26) we have $a_m^0 \mathbb{E}[\|x_i^0 - \bar{x}^0\|] \leq a_m^0 C$, while, for $\tau \geq 1$, from Lemma 7 it holds that

$$a_m^\tau \mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] \leq a_m^\tau MB\Bigg((\mu_M)^{\tau-1}C + \frac{G}{\sigma}\sum_{s=0}^{\tau-2}(\mu_M)^{\tau-s-2}a_M^s + \frac{G}{\sigma}a_M^{\tau-1}\Bigg).$$

Since, by Assumption 5, $a_m^{\tau+1} \leq a_m^\tau \leq a_M^\tau$ for all τ, one has

$$a_m^\tau \mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] \leq MB\Bigg(a_m^\tau(\mu_M)^{\tau-1}C + \frac{G}{\sigma}\sum_{s=0}^{\tau-2}(\mu_M)^{\tau-s-2}(a_M^s)^2 + \frac{G}{\sigma}(a_M^{\tau-1})^2\Bigg)$$

and then,

$$\begin{aligned}\sum_{\tau=0}^{t} a_m^\tau \mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] &\leq a_m^0 C + MB\Bigg(C\sum_{\tau=1}^{t}(\mu_M)^{\tau-1}a_m^\tau \\ &\quad + \frac{G}{\sigma}\sum_{\tau=1}^{t}\sum_{s=0}^{\tau-2}\mu_M^{\tau-s-2}(a_M^s)^2 + \frac{G}{\sigma}\sum_{\tau=1}^{t}(a_M^{\tau-1})^2\Bigg).\end{aligned}$$

Now, from Assumption 6(C), $C < \infty$. Moreover, by using Assumption 5 and Lemma 3, we have $\lim_{t\to\infty}\sum_{\tau=1}^{t}(\mu_M)^{\tau-1}a_m^\tau < \infty$. Finally, by Assumption 5, $\sum_{\tau=0}^{\infty}(a_M^\tau)^2 < \infty$, so that, from Lemma 3, $\lim_{t\to\infty}\sum_{\tau=1}^{t}\sum_{s=0}^{\tau-2}(\mu_M)^{\tau-s-2}(a_M^s)^2 < \infty$, thus concluding the proof. □

As in the case of constant stepsizes, thanks to Lemma 8, it can be shown that $\lim_{t\to\infty}\mathbb{E}[\|y_i^t - x_i^t\|] = 0$ and $\lim_{t\to\infty}\sum_{\tau=0}^{t} a_m^\tau \mathbb{E}[\|y_i^\tau - x_i^\tau\|] < \infty$.

B. Optimality

In this section, we show the convergence of the Distributed Block Proximal Method. First, a bound on the expected distance from the optimal cost at iteration t is given without any assumption on the stepsize sequence. Then, it is shown that such a distance goes to 0 as $t \to \infty$ for diminishing stepsizes, while it is upper bounded by a finite quantity for constant stepsizes and an explicit convergence rate is provided.

We start by defining the Ljapunov function

$$V_i^\tau \triangleq \sum_{\ell=1}^{B}\pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) \tag{27}$$

and $V^t \triangleq \sum_{i=1}^{N} V_i^t$. Moreover, given a sequence of points $\{z^\tau\}_{\tau=0}^t$, we define

$$f_{best}(z^t) \triangleq \min_{\tau\leq t}\mathbb{E}[f(z^\tau)] \tag{28}$$

Then, the following result holds true.

Theorem 1. *Let Assumptions 1, 2, 3, 6 hold. Then,*

$$\begin{aligned}f_{best}(\bar{x}^t) - f(x^\star) \leq \left(\sum_{\tau=0}^{t} a_m^\tau\right)^{-1}\Bigg(\mathbb{E}[V^0] + \sum_{\tau=0}^{t}\frac{(a_M^\tau)^2\bar{G}}{2\sigma} \\ + \sum_{\tau=0}^{t} a_m^\tau \sum_{i=1}^{N} G_i\mathbb{E}[\|y_i^\tau - x_i^\tau\|] \\ + \sum_{\tau=0}^{t} a_m^\tau \sum_{i=1}^{N} G_i\mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|]\Bigg).\end{aligned} \tag{29}$$

Proof. In order to simplify the notation, let us denote $g_i^\tau = g_i(y_i^\tau)$ and $g_{i,\ell_i^\tau} = g_{i,\ell_i^\tau}(y_i^\tau, \xi_i^\tau)$. From the convexity of f we have that, at a given iteration t,

$$\begin{aligned}\left(\sum_{\tau=0}^{t} a_m^\tau\right)(f_{best}(\bar{x}^t) - f(x^\star)) &= \left(\sum_{\tau=0}^{t} a_m^\tau\right)(\min_{\tau\leq t}(\mathbb{E}[f(\bar{x}^\tau)] - f(x^\star)) \\ &\leq \sum_{\tau=0}^{t} a_m^\tau\left(\mathbb{E}[f(\bar{x}^\tau)] - f(x^\star)\right).\end{aligned} \tag{30}$$

Now, we make some manipulation on the term $\mathbb{E}[f(\bar{x}^\tau)] - f(x^\star) = \mathbb{E}[f(\bar{x}^\tau) - f(x^\star)]$:

$$\begin{aligned}\mathbb{E}[f(\bar{x}^\tau) - f(x^\star)] &= \sum_{i=1}^{N}\mathbb{E}[(f_i(\bar{x}^\tau) - f_i(x^\star))] \\ &= \sum_{i=1}^{N}\mathbb{E}[(f_i(x_i^\tau) - f_i(x^\star) + f_i(\bar{x}^\tau) - f_i(x_i^\tau))] \\ &\leq \sum_{i=1}^{N}\mathbb{E}[(f_i(x_i^\tau) - f_i(x^\star))] + \sum_{i=1}^{N} G_i\mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] \\ &= \sum_{i=1}^{N}\mathbb{E}[(f_i(y_i^\tau) - f_i(x^\star) + f_i(x_i^\tau) - f_i(y_i^\tau))] \\ &\quad + \sum_{i=1}^{N} G_i\mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|] \\ &\leq \sum_{i=1}^{N}\mathbb{E}[(f_i(y_i^\tau) - f_i(x^\star))] \\ &\quad + \sum_{i=1}^{N} G_i\left(\mathbb{E}[\|y_i^\tau - x_i^\tau\|] + \mathbb{E}[\|x_i^\tau - \bar{x}^\tau\|]\right),\end{aligned} \tag{31}$$

where we have used multiple times the convexity of f (and of each f_j) and the subgradient boundedness (Assumption 1(C)). Let us now study the term $\sum_{i=1}^{N}\mathbb{E}[f_i(y_i^\tau) - f_i(x^\star)]$ in (31). By writing the optimality condition for the proximal mapping in the update (9), if $s_i^\tau = 1$, one has

$$\begin{aligned}\nu_{\ell_i^\tau}(x_{i,\ell_i^\tau}^{\tau+1}, x_{\ell_i^\tau}^\star) \leq \nu_{\ell_i^\tau}(y_{i,\ell_i^\tau}^\tau, x_{\ell_i^\tau}^\star) - \alpha_i^\tau\langle U_{\ell_i^\tau}g_i^\tau, y_i^\tau - x^\star\rangle \\ + \frac{(\alpha_i^\tau)^2}{2\sigma}\|g_{i,\ell_i^\tau}\|^2.\end{aligned} \tag{32}$$

Hence,

$$\nu_\ell(x_{i,\ell}^{\tau+1}, x_\ell^\star) \leq \begin{cases}(32), & \text{if } \ell = \ell_i^\tau, \text{ and } s_i^\tau = 1 \\ \nu_\ell(x_{i,\ell}^\tau, x_\ell^\star), & \text{otherwise.}\end{cases} \tag{33}$$

Thus, from (33) and by using (27), one has that, if $s_i^\tau = 1$, it holds

$$V_i^{\tau+1} \leq \sum_{m\neq \ell_i^\tau} \pi_{i,m}^{-1}\nu_m(x_{i,m}^\tau, x_m^\star) + \pi_{i,\ell_i^\tau}^{-1}\Bigg(\nu_{\ell_i^\tau}(y_{i,\ell_i^\tau}^\tau, x_{\ell_i^\tau}^\star) - \alpha_i^\tau\langle U_{\ell_i^\tau} g_i^\tau, y_i^\tau - x^\star\rangle + \frac{(\alpha_i^\tau)^2}{2\sigma}\|g_{i,\ell_i^\tau}\|^2\Bigg), \quad (34)$$

while, if $s_i^\tau = 0$,

$$V_i^{\tau+1} = V_i^\tau = \sum_{\ell=1}^B \pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star). \quad (35)$$

Now, by taking the expected value of $V_i^{\tau+1}$ conditioned to $\mathcal{S}^\tau$, one obtains

$$\mathbb{E}[V_i^{\tau+1} \mid \mathcal{S}^\tau] = (1-p_{i,on})\mathbb{E}[V_i^{\tau+1} \mid \mathcal{S}^\tau, s_i^\tau = 0] + p_{i,on}\mathbb{E}[V_i^{\tau+1} \mid \mathcal{S}^\tau, s_i^\tau = 1], \quad (36)$$

and hence, by substituting (34) and (35) in (36),

$$\begin{aligned}
&\mathbb{E}[V_i^{\tau+1} \mid \mathcal{S}^\tau] \\
&\leq (1-p_{i,on})\sum_{\ell=1}^B \pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) \\
&\quad + p_{i,on}\sum_{\ell=1}^B p_{i,\ell}\Bigg(\sum_{m\neq\ell}\pi_{i,m}^{-1}\nu_m(x_{i,m}^\tau, x_m^\star) \\
&\quad + \pi_{i,\ell}^{-1}\nu_\ell(y_{i,\ell}^\tau, x_\ell^\star) - \alpha_i^\tau\pi_{i,\ell}^{-1}\mathbb{E}[\langle U_\ell g_i^\tau, y_i^\tau - x^\star\rangle] \\
&\quad + \frac{(\alpha_i^\tau)^2}{2\sigma}\pi_{i,\ell}^{-1}\mathbb{E}[\|g_{i,\ell}\|^2]\Bigg). \quad (37)
\end{aligned}$$

Notice now that

$$\begin{aligned}
&\sum_{\ell=1}^B p_{i,\ell}\sum_{m\neq\ell}\pi_{i,m}^{-1}\nu_m(x_{i,m}^\tau, x_m^\star) \\
&= \sum_{\ell=1}^B\Bigg(p_{i,\ell}\sum_{m=1}^B \pi_{i,m}^{-1}\nu_m(x_{i,m}^\tau, x_m^\star) - p_{i,\ell}\pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star)\Bigg) \\
&= \sum_{m=1}^B\sum_{\ell=1}^B p_{i,\ell}\pi_{i,m}^{-1}\nu_m(x_{i,m}^\tau, x_m^\star) - \sum_{\ell=1}^B p_{i,\ell}\pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) \\
&= \sum_{m=1}^B \pi_{i,m}^{-1}\nu_m(x_{i,m}^\tau, x_m^\star) - \sum_{\ell=1}^B p_{i,\ell}\pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) \\
&= \sum_{\ell=1}^B(1-p_{i,\ell})\pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star). \quad (38)
\end{aligned}$$

Moreover, by noting that it holds that $\sum_{\ell=1}^B \mathbb{E}[\|g_{i,\ell}^\tau\|^2] = \mathbb{E}[\sum_{\ell=1}^B \|g_{i,\ell}^\tau\|^2] = \mathbb{E}[\|g_i^\tau\|^2]$ and $\sum_{\ell=1}^B U_\ell g_i^\tau = g_i^\tau$ and by substituting (38) in (37), we obtain

$$\begin{aligned}
&\mathbb{E}[V_i^{\tau+1} \mid \mathcal{S}^\tau] \\
&= \sum_{\ell=1}^B\left((1-p_{i,on}) + p_{i,on} - p_{i,on}p_{i,\ell}\right)\pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) \\
&\quad + \sum_{\ell=1}^B \nu_\ell(y_{i,\ell}^\tau, x_\ell^\star) - \alpha_i^\tau\mathbb{E}[\langle g_i^\tau, y_i^\tau - x^\star\rangle] + \frac{(\alpha_i^\tau)^2}{2\sigma}\mathbb{E}[\|g_i^\tau\|^2] \\
&= \sum_{\ell=1}^B(1-\pi_{i,\ell})\pi_{i,\ell}^{-1}\nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) + \sum_{\ell=1}^B \nu_\ell(y_{i,\ell}^\tau, x_\ell^\star) \\
&\quad - \alpha_i^\tau\mathbb{E}[\langle g_i^\tau, y_i^\tau - x^\star\rangle] + \frac{(\alpha_i^\tau)^2}{2\sigma}\mathbb{E}[\|g_i^\tau\|^2] \\
&= V_i^\tau - \sum_{\ell=1}^B \nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) + \sum_{\ell=1}^B \nu_\ell(y_{i,\ell}^\tau, x_\ell^\star) \\
&\quad - \alpha_i^\tau\langle\mathbb{E}[g_i^\tau], y_i^\tau - x^\star\rangle + \frac{(\alpha_i^\tau)^2}{2\sigma}\mathbb{E}[\|g_i^\tau\|^2] \\
&\leq V_i^\tau - \sum_{\ell=1}^B \nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) + \sum_{j=1}^N w_{ij}\sum_{\ell=1}^B \nu_\ell(x_{j,\ell}^\tau, x_\ell^\star) \\
&\quad - \alpha_i^\tau\langle g_i^\tau, y_i^\tau - x^\star\rangle + \frac{(\alpha_i^\tau)^2\bar{G}_i}{2\sigma} \quad (39)
\end{aligned}$$

where in the last inequality we used the separate convexity of ν_ℓ from Assumptions 3 and the fact that $\mathbb{E}[\|g_i^\tau\|^2] \leq \bar{G}_i$ (Assumption 1(C)), and $\mathbb{E}[g_i^\tau] = g_i^\tau$ (Assumption 1(B)). Now, by summing over i,

$$\begin{aligned}
&\sum_{i=1}^N \mathbb{E}[V_i^{\tau+1} \mid \mathcal{S}^\tau] \\
&\leq \sum_{i=1}^N V_i^\tau - \sum_{i=1}^N\sum_{\ell=1}^B \nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) + \sum_{i=1}^N\sum_{j=1}^N w_{ij}\sum_{\ell=1}^B \nu_\ell(x_{j,\ell}^\tau, x_\ell^\star) \\
&\quad - \sum_{i=1}^N \alpha_i^\tau\langle g_i^\tau, y_i^\tau - x^\star\rangle + \sum_{i=1}^N \frac{(\alpha_i^\tau)^2\bar{G}_i}{2\sigma} \\
&= \sum_{i=1}^N V_i^\tau - \sum_{i=1}^N\sum_{\ell=1}^B \nu_\ell(x_{i,\ell}^\tau, x_\ell^\star) + \sum_{j=1}^N\sum_{\ell=1}^B \nu_\ell(x_{j,\ell}^\tau, x_\ell^\star) \\
&\quad - \sum_{i=1}^N \alpha_i^\tau\langle g_i^\tau, y_i^\tau - x^\star\rangle + \sum_{i=1}^N \frac{(\alpha_i^\tau)^2\bar{G}_i}{2\sigma} \\
&= \sum_{i=1}^N V_i^\tau - \sum_{i=1}^N \alpha_i^\tau\langle g_i^\tau, y_i^\tau - x^\star\rangle + \sum_{i=1}^N \frac{(\alpha_i^\tau)^2\bar{G}_i}{2\sigma} \\
&\leq \sum_{i=1}^N V_i^\tau - \sum_{i=1}^N \alpha_i^\tau\left(f_i(y_i^\tau) - f_i(x^\star)\right) + \frac{(a_M^\tau)^2\bar{G}}{2\sigma}, \quad (40)
\end{aligned}$$

where in the last inequality we used the convexity of f_i. Taking the expected value conditioned to $\mathcal{S}^0$ on both sides of (40), using $\mathbb{E}[V_i^{\tau+1}|\mathcal{S}^\tau]$=$\mathbb{E}[V_i^{\tau+1}|\mathcal{S}^\tau, \mathcal{S}^0]$ and Lemma 4, gives

$$\begin{aligned}
\sum_{i=1}^N \mathbb{E}\left[\mathbb{E}[V_i^{\tau+1} \mid \mathcal{S}^\tau] \mid \mathcal{S}^0\right] &= \sum_{i=1}^N \mathbb{E}[V_i^{\tau+1} \mid \mathcal{S}^0] \\
&\leq \sum_{i=1}^N \mathbb{E}[V_i^\tau \mid \mathcal{S}^0] \\
&\quad - \sum_{i=1}^N \alpha_i^\tau\left(\mathbb{E}[f_i(y_i^\tau) \mid \mathcal{S}^0] - f_i(x^\star)\right) + \frac{(a_M^\tau)^2\bar{G}}{2\sigma}.
\end{aligned}$$

and, rearranging the terms,

$$\sum_{i=1}^{N}\alpha_i^\tau\left(\mathbb{E}[f_i(y_i^\tau)\mid\mathcal{S}^0]-f_i(x^\star)\right)\leq\sum_{i=1}^{N}\mathbb{E}[V_i^\tau\mid\mathcal{S}^0]-\sum_{i=1}^{N}\mathbb{E}[V_i^{\tau+1}\mid\mathcal{S}^0]+\frac{(a_M^\tau)^2\bar{G}}{2\sigma}.$$

Now, by summing over τ, and noting that $\mathbb{E}[V_i^0\mid\mathcal{S}^0]=V_i^0$,

$$\begin{aligned}\sum_{\tau=0}^{t}\sum_{i=1}^{N}\alpha_i^\tau\left(\mathbb{E}[f_i(y_i^\tau)\mid\mathcal{S}^0]-f_i(x^\star)\right)&\leq\sum_{i=1}^{N}V_i^0-\sum_{i=1}^{N}\mathbb{E}[V_i^{t+1}\mid\mathcal{S}^0]+\sum_{\tau=0}^{t}\frac{(a_M^\tau)^2\bar{G}}{2\sigma}\\&\leq V^0+\sum_{\tau=0}^{t}\frac{(a_M^\tau)^2\bar{G}}{2\sigma}.\end{aligned}$$

Moreover, by taking the expected value over $\mathcal{S}^0=\{\boldsymbol{x}^0\}$,

$$\sum_{\tau=0}^{t}\sum_{i=1}^{N}\alpha_i^\tau\left(\mathbb{E}[f_i(y_i^\tau)]-f_i(x^\star)\right)\leq\mathbb{E}[V^0]+\sum_{\tau=0}^{t}\frac{(a_M^\tau)^2\bar{G}}{2\sigma}.\tag{41}$$

Notice now that, since by definition $a_m^\tau\leq\alpha_i^\tau$ for all i and τ, we have

$$\sum_{\tau=0}^{t}a_m^t\sum_{i=1}^{N}\left(\mathbb{E}[f_i(y_i^\tau)]-f_i(x^\star)\right)\leq\sum_{\tau=0}^{t}\sum_{i=1}^{N}\alpha_i^\tau\left(\mathbb{E}[f_i(y_i^\tau)]-f_i(x^\star)\right).\tag{42}$$

Finally, by combining (30), (31), (41) and (42) one obtains (29). □

A similar result can be given also in terms of the sequences of local solution estimates $\{x_i^t\}$ as formalized in the next result.

Corollary 1. *Let Assumptions 1, 2, 3, 6 hold. Then,*

$$\begin{aligned}f_{\textit{best}}(x_i^t)-f(x^\star)\leq\left(\sum_{\tau=0}^{t}a_m^\tau\right)^{-1}\Bigg(&\mathbb{E}[V^0]+\sum_{\tau=0}^{t}\frac{(a_M^\tau)^2\bar{G}}{2\sigma}\\&+\sum_{\tau=0}^{t}a_m^\tau\sum_{j=1}^{N}G_j\mathbb{E}[\|y_j^\tau-x_j^\tau\|]\\&+\sum_{\tau=0}^{t}a_m^\tau\sum_{j=1}^{N}G_j\mathbb{E}[\|x_j^\tau-\bar{x}^\tau\|]\\&+\sum_{\tau=0}^{t}a_m^\tau G\mathbb{E}[\|x_i^\tau-\bar{x}^\tau\|]\Bigg).\end{aligned}\tag{43}$$

for all $i\in\{1,\dots,N\}$.

Proof. The proof follows the same line of the one of Theorem 1, by noting that

$$\begin{aligned}f(x_i^t)-f(x^\star)&=f(x_i^t)-f(\bar{x}^t)+f(\bar{x}^t)-f(x^\star)\\&\leq f(\bar{x}^t)-f(x^\star)+G\|x_i^t-\bar{x}^t\|,\end{aligned}$$

so that in place of (30), one has

$$\begin{aligned}\left(\sum_{\tau=0}^{t}a_m^\tau\right)\left(f_{\textit{best}}(x_i^t)-f(x^\star)\right)&\leq\sum_{\tau=0}^{t}a_m^\tau\left(\mathbb{E}[f(x_i^\tau)]-f(x^\star)\right)\\&\leq\sum_{\tau=0}^{t}a_m^\tau\left(\mathbb{E}[f(\bar{x}^\tau)]-f(x^\star)+G\mathbb{E}[\|x_i^\tau-\bar{x}^\tau\|]\right)\end{aligned}$$

for all $i\in\{1,\dots,N\}$. □

The previous two results hold true without making any assumption on the local stepsize sequences. In the next two subsections the general result of Theorem 1 is specialized to the case of constant and diminishing stepsizes respectively.

1) Constant stepsizes: In the case of constant (local) stepsizes, convergence with a constant error is attained with an explicit sublinear convergence rate.

Theorem 2. *Let Assumptions 1, 2, 3, 4, 6 hold. Then, there exist constants* $M\in(0,\infty)$ *and* $\mu_M\in(0,1)$ *such that*

$$f_{\textit{best}}(x_i^t)-f(x^\star)\leq\frac{Q+(\mu_M)^tR}{t+1}+S\tag{44}$$

for all $i\in\{1,\dots,N\}$*, with*

$$\begin{aligned}Q&=\frac{\mathbb{E}[V^0]}{a_m}+4G\left(\frac{MBC}{1-\mu_M}+\bar{Q}\right),\\R&=4G\bar{R},\\S&=4G\bar{S}+\frac{a_M^2}{a_m}\frac{\bar{G}}{2\sigma}.\end{aligned}$$

Proof. By exploiting Assumption 4, Lemma 10 and Lemma 8, from (43) one obtains

$$\begin{aligned}f_{\textit{best}}(x_i^t)-f(x^\star)\leq\left(\sum_{\tau=0}^{t}a_m\right)^{-1}\Bigg(&\mathbb{E}[V^0]+\sum_{\tau=0}^{t}\frac{(a_M)^2\bar{G}}{2\sigma}\\&+4a_mG\Big((\mu_M)^t\bar{R}+t\bar{S}+\bar{Q}\Big)\end{aligned}$$

which, by rearranging the terms, leads to

$$\begin{aligned}f_{\textit{best}}(x_i^t)-f(x^\star)&\leq\frac{Q+(\mu_M)^tR}{t+1}+\frac{t}{t+1}4G\bar{S}+\frac{a_M^2}{a_m}\frac{\bar{G}}{2\sigma}\\&\leq\frac{Q+(\mu_M)^tR}{t+1}+S\end{aligned}$$

thus concluding the proof. □

The previous result shows that, when constant stepsizes are employed, the value of $f_{\textit{best}}(x_i^t)$ converges to $f(x^\star)$ plus a constant error, which can be retrieved from (44) by taking the limit for $t\to\infty$, i.e., $\lim_{t\to\infty}f_{\textit{best}}(x_i^t)-f(x^\star)\leq S$ with the explicit expression for S being

$$S=a_M\left(\frac{4MBG^2}{\sigma}\frac{2-\mu_M}{1-\mu_M}+\frac{a_M}{a_m}\frac{\bar{G}}{\sigma}\right).$$

It is worth noting that the bound decreases with the maximum stepsize a_M. Regarding the convergence rate, it is sublinear $O\left(Q/t\right)$. However, in the first iterations, the term $(\mu_M)^tR$

can be dominant (if $|R| \gg Q$), thus leading to a linear rate at the beginning of the algorithm (as it will be shown in the numerical example). Notice that Q (and hence the convergence rate) depends both on the number of blocks and on the local probabilities of being awake and drawing blocks. The local probabilities appear in V^0 and (implicitly) in the constant μ_M. In fact, μ_M is related to the randomness of the matrices W_ℓ^t, which depend on such probabilities. In particular notice that, if $B = 1$ the rate is similar to those obtained in [27], in which the proximal mapping $\nu(a, b) = \frac{1}{2}\|a - b\|^2$ is used. Clearly, one may argue that the best rate is achieved by using a single block and hence, communicating in terms of blocks is useless. This is true only if we assume an infinite bandwidth to be available in the communication channels (i.e, transmitting the entire optimization variable or a single block of it requires the same amount of time). However, in typical real world scenarios this is not true and data that exceed the communication bandwidth are transmitted sequentially. If only one block fits the communication channel, our algorithm allows to perform an update at each communication round, while classical ones would need B communication rounds per update. Moreover, in the proposed algorithm, typically, the local computation time at each iteration is *not* negligible, since a minimization problem is to be solved at every step (see (7)). Solving such an optimization problem on the entire optimization or on a single block of it clearly results in completely different computational times, which are clearly lower in the case of block-wise updates. Thus, the benefits of using block-wise updates and communications make the Distributed Block Proximal Method well suited for big-data optimization problems.

2) Diminishing stepsizes: In the case of diminishing (local) stepsizes, asymptotic convergence to the optimal cost can be reached.

Theorem 3. *Let Assumptions 1, 2, 3, 5, 6 hold. Then,*

$$\lim_{t\to\infty} f_{best}(x_i^t) - f(x^\star) = 0,$$

for all $i \in \{1, \dots, N\}$.

Proof. The proof follows by taking the limit for $t \to \infty$, and using Assumption 5 and Lemma 12 in (43). □

Remark 2. We point out that, similarly to, e.g., [37, 11], one can introduce a running averaging mechanism by defining $\hat{x}_i^t = \frac{1}{t}\sum_{\tau=1}^t x_i^t$ and provide the convergence results in terms of $f(\hat{x}_i^t) - f(x^\star)$, in place of $f_{best}(x_i^t) - f(x^\star)$. The convergence proof would follow almost the same line with some adjustments in (30)-(31) (see [11]). However, running averaging mechanisms typically lead to a significantly slower convergence rate. In light of this, we provided our results in terms of $f_{best}(x_i^t) - f(x^\star)$.

VI. Numerical example

Consider a soft margin classification problem in which each agent $i \in \{1, \dots, N\}$ has m_i training samples $q_i^1, \dots, q_i^{m_i} \in \mathbb{R}^d$ each of which has an associated binary label $b_i^r \in \{-1, 1\}$ for all $r \in \{1, \dots, m_i\}$. The goal of the network is to build a linear classifier from the training samples, i.e., to find a hyperplane of the form $\{z \in \mathbb{R}^d \mid \langle \theta, z\rangle + \theta_0 = 0\}$, with $\theta \in \mathbb{R}^d$ and $\theta_0 \in \mathbb{R}$, which better separates the training data. Let us define $x = [\theta^\top, \theta_0]^\top \in \mathbb{R}^{d+1}$ and $\hat{q}_i^r = [(q_i^r)^\top, 1]^\top$. Then, the solution to this problem can be determined by solving the following SVM problem

$$\underset{x\in\mathbb{R}^{d+1}}{\text{minimize}} \quad \sum_{i=1}^N \frac{1}{m_i}\sum_{r=1}^{m_i} \log\left(1 + e^{-b_i^r\langle x, \hat{q}_i^r\rangle}\right) + \lambda\|x\|_1, \tag{45}$$

where $\lambda > 0$ is the regularization weight. Problem (45) can be written in the form of problem (1) by defining $\xi_i^r = (\hat{q}_i^r, b_i^r)$ and

$$\begin{aligned}\mathbb{E}[h_i(x;\xi_i)] &= \frac{1}{m_i}\sum_{r=1}^{m_i} h_i(x;\xi_i^r)\\ &= \frac{1}{m_i}\sum_{r=1}^{m_i}\left(\log\left(1 + e^{-b_i^r\langle x, \hat{q}_i^r\rangle}\right) + \frac{\lambda}{N}\|x\|_1\right)\end{aligned}$$

for all $i \in \{1, \dots, N\}$. Notice that, as long as each data ξ_i^r is uniformly drawn from the dataset, Assumption 1(B) is satisfied.

In the next two sections we will test the Distributed Block Subgradient Method in the presented scenario, first on a synthetic dataset and then on real-world dataset composed of text documents. In both cases we consider a system with $N = 48$ processors. The proposed distributed algorithm has been implemented by using the Python package DISROPT [51], and each processor has been assigned an agent.

A. Synthetic dataset

In order to show how the algorithm performs for different number of blocks, let us consider a relatively low-dimensional problem with $x \in \mathbb{R}^{50}$ and evaluate the algorithm performance for different number of blocks, namely $B \in \{1, 2, 5, 10, 50\}$. We generate a synthetic dataset (with $x \in \mathbb{R}^{50}$) composed of 240 points (taken from two clusters corresponding to labels -1 and 1 respectively) and assign 5 of them to each agent, i.e., $m_1 = \cdots = m_N = 5$. Finally, we set $\lambda = 0.1$, a common (constant) stepsize $\alpha = 0.2$, $p_{i,\ell} = 1/B$ for all i and all ℓ and $s_i^t = 1$ for all i and all t. Regarding the communication graph, it has is generated according to an Erdős-Rènyi random model with connectivity parameter $p = 0.3$. The corresponding weight matrix is built by using the Metropolis-Hastings rule. The evolution of the cost error adjusted with respect to the number of blocks is reported in Figure 1 for the considered block numbers. The results confirm the discussion carried out in Section V-B1 about the role of block communications. In fact, when normalizing the number of iterations with respect to the number of blocks, the convergence rates for the considered number of blocks are comparable. Moreover, the convergence rate exhibits the properties shown in Section V-B1. In fact, it is linear at the beginning and becomes sublinear after some iterations. Moreover, as expected when using constant stepsizes, convergence is reached with a constant error.

B. Text classification

Let us now consider a real-world scenario in which the local training samples are drawn from a dataset of texts. In particular,

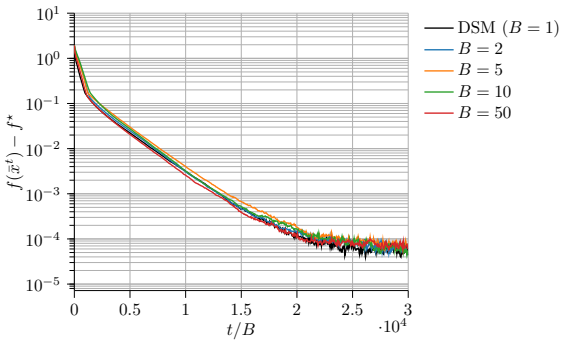

Fig. 1: Numerical example: synthetic dataset. Evolution of the cost error for $B \in \{1, 2, 5, 10, 50\}$. The time scale is normalized on the number of blocks. The case $B = 1$ coincides with the distributed subgradient method [27].

we pick the *20 newsgroups dataset*, a dataset consisting of 18,846 newsgroup posts belonging to 20 topics. Texts are represented by TF-IDF on a dictionary of 130,107 words, so that each sample is a vector in $\mathbb{R}^{130,107}$. Agents have to learn to classify posts belonging to the class *sci.med* from the others. Thus, in order to perform a binary classification, we assign the label 1 to samples belonging to the class *sci.med*, and -1 to all the other samples. In this scenario, the considered 48 agents, are connected over a balanced directed graph, generated according to a binomial random model with connectivity parameter $p = 0.5$ and each agent is awake with probability $p_{i,on} = 0.95$. The entire dataset is split to assign almost the same number of samples to each agent. We run the algorithm for 4000 iterations and for different number of blocks, namely $B \in \{10, 10^2, 10^3, 10^4\}$. Moreover, we set $\lambda = 0.001$ in problem (45), and we select a common (constant) stepsize $\alpha = 0.5$ and $p_{i,\ell} = 1/B$ for all i and all ℓ. Differently from the previous example over synthetic data, in this case computing the exact (centralized) solution of the considered problem is computationally intractable, due to the high dimension of the decision variable (n =130,107) and the large number of samples ($\sum_{i=1}^N m_i$ =18,146). Thus, the performance of the algorithm are evaluated in terms of the accuracy Ψ of the average of the produced solution estimates $\bar{x}^t$, i.e., the number of samples of the dataset that are correctly classified through the hyperplane defined by $\bar{x}^t$. The results are reported in Figure 2.

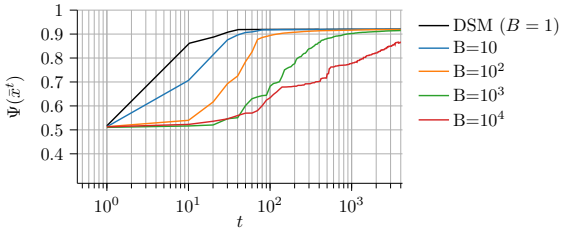

Fig. 2: Numerical example: text classification. Evolution of the accuracy over the entire *20 newsgroups* dataset.

VII. Conclusions

In this paper, we introduced a class of distributed block proximal algorithms for solving stochastic big-data convex optimization problems over networks. In the addressed optimization set-up the dimension of the decision variable is very high and the (stochastic) cost function may be nonsmooth. The main strength of the proposed algorithms is that agents in the network can communicate a single block of the optimization variable per iteration. Under the assumption of diminishing stepsizes, we showed that the agents in the network asymptotically agree on a common solution which is cost-optimal in expected value. When employing constant stepsizes approximate convergence is attained with a constant error on the optimal cost and an explicit convergence rate is provided. Special instances of the algorithm are presented for particular classes of problems. Finally, the proposed algorithm, has been numerically evaluated on a distributed classification problem over both a synthetic dataset and a real, high-dimensional, text document dataset.

References

[1] F. Farina and G. Notarstefano, "A randomized block subgradient approach to distributed big data optimization," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019.

[2] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 479–502, 2002.

[3] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *Journal of Machine Learning Research*, vol. 11, no. Oct, pp. 2543–2596, 2010.

[4] K. I. Tsianos and M. G. Rabbat, "Consensus-based distributed online prediction and optimization," in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 807–810.

[5] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[6] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[7] H. Kushner and G. G. Yin, *Stochastic approximation and recursive algorithms and applications*. Springer Science & Business Media, 2003, vol. 35.

[8] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[9] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework," *SIAM Journal on Optimization*, vol. 22, no. 4, pp. 1469–1492, 2012.

[10] A. Nedic and S. Lee, "On stochastic subgradient mirror-descent algorithm with weighted averaging," *SIAM Journal on Optimization*, vol. 24, no. 1, pp. 84–107, 2014.

[11] C. D. Dang and G. Lan, "Stochastic block mirror descent methods for nonsmooth and stochastic optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 856–881, 2015.

[12] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[13] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[14] Y. Xu and W. Yin, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *Journal of Scientific Computing*, vol. 72, no. 2, pp. 700–734, 2017.

[15] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, no. 1-2, pp. 1–38, 2014.

[16] T. Zhao, M. Yu, Y. Wang, R. Arora, and H. Liu, "Accelerated mini-batch randomized block coordinate descent method," in *Advances in neural information processing systems*, 2014, pp. 3329–3337.

[17] Z. Lu and L. Xiao, "On the complexity analysis of randomized block-coordinate descent methods," *Mathematical Programming*, vol. 152, no. 1-2, pp. 615–642, 2015.

[18] J. Chen and Q. Gu, "Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization," in *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 2016, pp. 132–141.

[19] A. Zhang and Q. Gu, "Accelerated stochastic block coordinate descent with optimal sampling," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 2035–2044.

[20] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[21] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[22] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex big data optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1874–1889, 2015.

[23] P. Richtárik and M. Takáč, "Parallel coordinate descent methods for big data optimization," *Mathematical Programming*, vol. 156, no. 1-2, pp. 433–484, 2016.

[24] I. Necoara and D. Clipici, "Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 197–226, 2016.

[25] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[26] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[27] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, p. 48, 2009.

[28] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.

[29] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[30] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[31] C. Xi and U. A. Khan, "Distributed subgradient projection algorithm over directed graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3986–3992, 2017.

[32] S. Liu, Z. Qiu, and L. Xie, "Convergence rate analysis of distributed optimization with projected subgradient algorithm," *Automatica*, vol. 83, pp. 162–169, 2017.

[33] P. Wang, P. Lin, W. Ren, and Y. Song, "Distributed subgradient-based multiagent optimization with more general step sizes," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2295–2302, 2018.

[34] H. Li, Q. Lü, and T. Huang, "Distributed projection subgradient algorithm over time-varying general unbalanced directed graphs," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1309–1316, 2019.

[35] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[36] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 601–608.

[37] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2012.

[38] K. Margellos, A. Falsone, S. Garatti, and M. Prandini, "Distributed constrained optimization and consensus in uncertain networks via proximal minimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1372–1387, 2018.

[39] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 873–881.

[40] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011.

[41] M. Rabbat, "Multi-agent mirror descent for decentralized stochastic optimization," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2015, pp. 517–520.

[42] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, 2016.

[43] G. Lan, S. Lee, and Y. Zhou, "Communication-efficient algorithms for decentralized and stochastic optimization," *Mathematical Programming*, pp. 1–48, 2017.

[44] J. Li, G. Li, Z. Wu, and C. Wu, "Stochastic mirror descent method for distributed multi-agent optimization," *Optimization Letters*, vol. 12, no. 6, pp. 1179–1197, 2018.

[45] B. Liu, W. Lu, and T. Chen, "Consensus in networks of multiagents with switching topologies modeled as adapted stochastic processes," *SIAM Journal on Control and Optimization*, vol. 49, no. 1, pp. 227–253, 2011.

[46] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, 2011.

[47] I. Notarnicola, Y. Sun, G. Scutari, and G. Notarstefano, "Distributed big-data optimization via block-wise gradient tracking," *arXiv preprint arXiv:1808.07252*, 2018.

[48] F. Farina, A. Garulli, A. Giannitrapani, and G. Notarstefano, "A distributed asynchronous method of multipliers for constrained nonconvex optimization," *Automatica*, vol. 103, pp. 243 – 253, 2019.

[49] I. Necoara, "Random coordinate descent algorithms for multi-agent convex optimization over networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 8, pp. 2001–2012, 2013.

[50] H. H. Bauschke and J. M. Borwein, "Joint and separate convexity of the bregman distance," in *Studies in Computational Mathematics*. Elsevier, 2001, vol. 8, pp. 23–36.

[51] F. Farina, A. Camisa, A. Testa, I. Notarnicola, and G. Notarstefano, "DISROPT: a Python Framework for Distributed Optimization," in *21st IFAC World Congress*, 2020.

Francesco Farina (S'16, M'19) is a GSK.ai Fellow in the Artificial Intelligence and Machine Learning group at GSK. He received the B.Sc. degree in Electronic and Telecommunication Engineering from the University of Florence in 2013, the M.Sc. degree "summa cum laude" in Management Engineering in 2015 and the Ph.D. degree in Information Engineering and Science in 2019, both from the University of Siena. From December 2018 to May 2020 he has been a research fellow at the Department of Electrical, Electronic and Information Engineering G. Marconi at Alma Mater Studiorum Università di Bologna and a research associate at the University of Siena. His current research interests include artificial intelligence, machine learning, optimization and distributed systems.

Giuseppe Notarstefano (M'11) is a Professor in the Department of Electrical, Electronic, and Information Engineering G. Marconi at Alma Mater Studiorum Università di Bologna. He was Associate Professor (June '16 – June '18) and previously Assistant Professor, Ricercatore, (from Feb '07) at the Università del Salento, Lecce, Italy. He received the Laurea degree "summa cum laude" in Electronics Engineering from the Università di Pisa in 2003 and the Ph.D. degree in Automation and Operation Research from the Università di Padova in 2007. He has been visiting scholar at the University of Stuttgart, University of California Santa Barbara and University of Colorado Boulder. His research interests include distributed optimization, cooperative control in complex networks, applied nonlinear optimal control, and trajectory optimization and maneuvering of aerial and car vehicles. He serves as an Associate Editor for IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology and IEEE Control Systems Letters. He has been part of the Conference Editorial Board of IEEE Control Systems Society and EUCA. He is recipient of an ERC Starting Grant 2014.