Distributed Signal Processing and Optimization based on In-Network Subspace Projections

Paolo Di Lorenzo, *Senior Member, IEEE*, Sergio Barbarossa, *Fellow, IEEE*, Stefania Sardellitti, *Member, IEEE*

***Abstract*—We study distributed optimization and processing of subspace-constrained signals in multi-agent networks with sparse connectivity. We introduce the first optimization framework based on distributed subspace projections, aimed at minimizing a network cost function depending on the specific processing task, while imposing subspace constraints on the final solution. The proposed method hinges on (sub)gradient optimization techniques while leveraging distributed projections as a mechanism to enforce subspace constraints in a cooperative and distributed fashion. Asymptotic convergence rates to optimal solutions of the problem are established under different assumptions (e.g., nondifferentiability, nonconvexity, etc.) on the objective function. We also introduce an extension of the framework that works with constant step-sizes, thus enabling faster convergence to optimal solutions of the optimization problem. Our algorithmic framework is very flexible and can be customized to a variety of problems in distributed signal processing. Finally, numerical tests on synthetic and realistic data illustrate how the proposed methods compare favorably to existing distributed algorithms.**

***Index Terms*—Distributed Optimization, Signal Processing, Networks, Subspace Projections, Convergence Analysis.**

I. Introduction

Distributed signal processing aims at performing learning tasks from data that is naturally distributed over a multi-agent network having, typically, a sparse topology [1], [2]. Such inference goals typically arise in multiple real-world scenarios including, among others, wireless sensor networks [1], [3], data mining in peer-to-peer networks [4], distributed databases [5], and mobile edge computing in 5G systems [6]. Common to these applications is the necessity of performing a completely decentralized computation/optimization. For instance, when data are collected/stored over a distributed network, sharing local information with a central processor is either unfeasible or not economical/efficient, owing to the large size of the network and volume of data, time-varying network topology, energy constraints, and/or privacy issues. Performing learning in a centralized fashion may raise robustness concerns as well, since the central processor represents an isolate point of failure. Due to these reasons, nowadays, the need for fully distributed inference is recognized as a defining characteristic of many real-world big data applications [7]. Additional emphasis is also provided by the Internet-of-Things (IoT) paradigm, which envisions that several *trillions* of smart devices will be connected in the very near future [8].

To be more specific, let us consider a network composed of N sensors, where the i-th node collects a measurement y_i of the signal value x_i at its local geographic position. Let $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ be the vector collecting the signal values at every node of the network. The gathered measurements $\{y_i\}_{i=1}^N$ may be highly unreliable due to observation noise, presence of outliers, missing data, etc. Improving the reliability of the individual node is typically unfeasible because of increased complexity and cost, which are fundamental design constraints in large scale networks. A way to recover reliability is to properly fuse the measurements collected over all the network in order to reach some globally optimal decision. This is possible if the set of data gathered by the network exhibits some kind of *structure*, which is typically the case in many physical fields of interest, e.g., the distribution of temperatures or the concentration of a contaminant. In mathematical terms, this means that the observed signal field belongs to a low-dimensional subspace, i.e., the vector $\boldsymbol{x}$ can be cast as:

$$\boldsymbol{x} = \mathbf{U}\boldsymbol{s}, \tag{1}$$

where $\mathbf{U}$ is an $N \times r$ matrix, with $r \leq N$, and $\boldsymbol{s}$ is an $r \times 1$ column vector. The columns of $\mathbf{U}$ are assumed to be linearly independent and thus constitute a basis spanning the signal subspace. In many applications, the signal is a smooth function, which can be very well modeled by choosing the columns of $\mathbf{U}$ as the low frequency components of the Fourier basis, or low-order polynomials, for example. In practice, the dimension r of the signal subspace is typically much smaller than the dimension N of the observation space [9], [10].

In this paper, we consider a broad family of signal processing tasks that can be written as instances of the following subspace-constrained optimization framework:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}; \boldsymbol{y}) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{R}(\mathbf{U}) \end{aligned} \tag{2}$$

where $f(\cdot)$ is the network objective function, which depends on the observed vector $\boldsymbol{y}$ and on the vector $\boldsymbol{x}$ to be recovered, and $\mathcal{R}(\mathbf{U})$ denotes the range space of the full-column rank matrix $\mathbf{U}$, i.e., the subspace where $\boldsymbol{x}$ lies. The properties of $f(\cdot)$ [e.g., (non)convexity, (non)differentiability] depend on the specific processing task, as specified in the examples below.

Signal Processing Applications. The framework in (2) subsumes several different signal processing tasks. We provide next a few instances of applications that fall within (2).

1) Noise reduction via subspace projection: Given noisy measurements $\{y_i\}_{i=1}^N$ of the signal values $\{x_i\}_{i=1}^N$, a fundamental task is to "clean" the observations to reduce the effect of noise. Since the signal belongs to the low-dimensional subspace $\mathcal{R}(\mathbf{U})$, a strong noise reduction may be obtained by computing the least square estimator for $\boldsymbol{x}$, which reads as:

$$\begin{aligned} \widehat{\boldsymbol{x}} \; = \; \arg\min_{\boldsymbol{x}} \quad & \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \\ \text{subject to } & \boldsymbol{x} \in \mathcal{R}(\mathbf{U}). \end{aligned} \tag{3}$$

The authors are with the Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome, Via Eudossiana 18, 00184, Rome, Italy; E-mail: {paolo.dilorenzo,sergio.barbarossa,stefania.sardellitti}@uniroma1.it.

Problem (3) is clearly an instance of the framework in (2), and its optimal solution is given by the projection of the observation vector $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ onto the signal subspace:

$$\widehat{\boldsymbol{x}} \;=\; \mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y} \;=\; \mathbf{U}\Big(\mathbf{U}^T\mathbf{U}\Big)^{-1}\mathbf{U}^T\boldsymbol{y}, \tag{4}$$

where

$$\mathcal{P}_{\mathcal{R}(\mathbf{U})} \;=\; \mathbf{U}\Big(\mathbf{U}^T\mathbf{U}\Big)^{-1}\mathbf{U}^T \tag{5}$$

is the operator that projects onto the subspace $\mathcal{R}(\mathbf{U})$. If the additive noise in $\{y_i\}_{i=1}^N$ is Gaussian, with zero mean and variance σ_v^2 for all i, the expression (4) represents the maximum likelihood (ML) estimator of $\boldsymbol{x}$. Otherwise, if the noise probability density function (pdf) is unknown, (4) is still significant, as it represents the so called Best Linear Unbiased Estimator (BLUE) [11].

2) Maximum likelihood estimation: In general, if the noise has a known pdf, the ML estimate of $\boldsymbol{x}$ can be found as the solution of problem:

$$\begin{aligned}\widehat{\boldsymbol{x}} \;=\; &\arg\max_{\boldsymbol{x}} \;\; \log p(\boldsymbol{x};\boldsymbol{y}) \\ &\text{subject to } \boldsymbol{x} \in \mathcal{R}(\mathbf{U})\end{aligned} \tag{6}$$

where $p(\boldsymbol{x};\boldsymbol{y})$ is the joint pdf of the observations. Problem (6) is an instance of (2), and is convex only if the joint pdf is a log-concave function, otherwise (6) is generally nonconvex. In some circumstances, we can also exploit the intrinsic structure of the joint pdf $p(\boldsymbol{x};\boldsymbol{y})$, which can be factorized into functions of subsets of variables [3]. The most simple situation pertains to the case where the observations are statistically independent, but we may also consider more general correlation structures among the variables described by, e.g., Bayes networks or Markov random fields. This is the case, for example, when the sensors monitor a field of spatially correlated values, like a temperature or an atmospheric pressure field.

3) Interpolation: A fundamental task in signal processing is interpolation, which emerges whenever cost constraints limit the number of observations that can be taken. Given a vector $\boldsymbol{x}$, suppose we observe a subset of samples x_i, with $i \in \mathcal{T}$, where $\mathcal{T}$ represents the sampling set, whose cardinality is smaller than the dimension N of $\boldsymbol{x}$. The goal is to recover the whole vector $\boldsymbol{x}$ from this subset of observations. In formulas, suppose we observe the values:

$$y_i = d_i x_i \quad i = 1, \ldots, N, \tag{7}$$

with $d_i = 1$, if $i \in \mathcal{T}$ and $d_i = 0$ otherwise. Given the sampled vector $\boldsymbol{y}$, the goal is to reconstruct $\boldsymbol{x}$. This is possible if $\boldsymbol{x}$ has a structure. Suppose $\boldsymbol{x}$ satisfies (1). Combining (1) with (7), the interpolation task involves the solution of the system of linear equations

$$\boldsymbol{y} = \mathbf{D}\mathbf{U}\boldsymbol{s}, \tag{8}$$

where $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_N)$ is the sampling operator. If system (8) admits a unique solution, i.e., if $\mathrm{rank}(\mathbf{DU}) = \mathrm{rank}(\mathbf{U}) = r$, the signal interpolation task can be equivalently cast as the solution of the following optimization problem:

$$\begin{aligned}\widehat{\boldsymbol{x}} \;=\; &\arg\min_{\boldsymbol{x}} \;\; \|\mathbf{D}(\boldsymbol{y}-\boldsymbol{x})\|_2^2 \\ &\text{subject to } \boldsymbol{x} \in \mathcal{R}(\mathbf{U}).\end{aligned} \tag{9}$$

Again, (9) can be seen as an instance of (2). The formulation in (9) is totally general, and also includes interpolation of graph signals as a particular case. In such circumstance, the basis $\mathbf{U}$ can be chosen as a proper selection of eigenvectors of Laplacian or adjacency matrices describing the graph signal frequency support, see, e.g., [12], [13].

4) Outlier rejection via ℓ_1 minimization: Let us consider now the case where a subset $\mathcal{C}$ of nodes is strongly corrupted by noise or damaged. In such a case, we have

$$y_i = x_i + q_i v_i, \quad i = 1, \ldots, N, \tag{10}$$

with $q_i = 1$ for $i \in \mathcal{C}$, and $q_i = 0$ otherwise. We also assume that the noise vector $\boldsymbol{v} = \{v_i\}_{i=1}^N$ in (10) is arbitrary but bounded, i.e., $\|\boldsymbol{v}\|_1 < \infty$. The goal in this case is the exact recovery of the signal $\boldsymbol{x}$ irrespective of noise. This problem is also known as the Logan's phenomenon [14]. If the signal $\boldsymbol{x}$ belongs to a low-dimensional subspace and the indexes of the noisy samples are known, the solution is simple, as one could simply discard the noisy measurements and then running an interpolation method over the noise-free samples. The challenging situation occurs when the location of the noisy observations is not known. In such a case, we may resort to a constrained ℓ_1-norm minimization, which reads as:

$$\begin{aligned}\widehat{\boldsymbol{x}} \;=\; &\arg\min_{\boldsymbol{x}} \;\; \|\boldsymbol{y}-\boldsymbol{x}\|_1 \\ &\text{subject to } \boldsymbol{x} \in \mathcal{R}(\mathbf{U}).\end{aligned} \tag{11}$$

Problem (11) is an instance of (2), and its solution allows perfect recovery of the bandlimited signal $\boldsymbol{x}$ if the number of corrupted nodes is not too large, see, e.g., [14].

5) Bandlimited plus sparse signal recovery: In many real systems, the observed signal is often given by the superposition of a smooth, or bandlimited, component plus a sparse component. This is the case, for example, of traffic data observed over the backbone of large-scale telecommunication networks, given by the superposition of some unknown "clean" traffic, which is usually smooth due to network topology-induced correlations, and traffic anomalies that occur sporadically over time and space [15]. In such a case, the observation model is given by:

$$y_i = x_i + a_i + v_i, \quad i = 1, \ldots, N, \tag{12}$$

where $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ represents the bandlimited part, $\boldsymbol{a} = [a_1, \ldots, a_N]^T$ models the sparse component, and $\boldsymbol{v} = [v_1, \ldots, v_N]^T$ is the additive noise vector. Leveraging the bandlimitedness property of $\boldsymbol{x}$ and the sparsity of $\boldsymbol{a}$, the signal recovery problem can be formulated as the constrained, regularized least square fitting given by:

$$\begin{aligned}(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{a}}) \;=\; &\arg\min_{\boldsymbol{x},\boldsymbol{a}} \;\; \|\boldsymbol{y}-\boldsymbol{x}-\boldsymbol{a}\|^2 + \lambda\|\boldsymbol{a}\|_1 \\ &\text{subject to } \boldsymbol{x} \in \mathcal{R}(\mathbf{U})\end{aligned} \tag{13}$$

with $\lambda > 0$. Problem (13) is a particular case of (2), where only one of the two variables (i.e., $\boldsymbol{x}$) is constrained to lie in the subspace spanned by the columns of $\mathbf{U}$. Of course, Problem (13) can be generalized in several ways incorporating, e.g., sampling (see Example #3), or adaptivity over time resorting to recursive least squares (RLS) formulations.

Related Works on Distributed Optimization. In principle, problem (2) requires that all nodes send their measurements to a fusion center that solves the corresponding optimization. Nevertheless, there is an ample literature showing when and how the problem can be solved in distributed form, where each node exchanges local information with its neighbors only. To allow a distributed solution for (2), we need some preliminary assumption on the objective function. First of all, we can notice that the objective functions in (3), (9), (11), (13), all have a separable structure, i.e.,

$$f(\boldsymbol{x};\boldsymbol{y}) = \sum_{i=1}^{N} f_i(x_i; y_i). \tag{14}$$

The structure (14) is amenable for a distributed solution using consensus-based optimization algorithms [16]. More specifically, using (1), an iterative algorithm can be implemented to achieve a consensus on the global vector $\boldsymbol{s}$. In particular, denoting with $\boldsymbol{s}_i$ the local copy of the global variable $\boldsymbol{s}$ at node i, with $i = 1, \ldots, N$, problem (2) can be recast as the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{s}} \sum_{i=1}^{N} f_i(\boldsymbol{s}; y_i) \;=\; & \min_{\{\boldsymbol{s}_i\}_{i=1}^{N}} \sum_{i=1}^{N} f_i(\boldsymbol{s}_i; y_i) \\ & \text{subject to } \boldsymbol{s}_i = \boldsymbol{s}_j \quad \text{for all } i, j. \end{aligned} \tag{15}$$

This problem can be solved using a consensus optimization algorithm where, at each iteration, each node exchanges the current local copy $\boldsymbol{s}_i$ with its neighbors. At convergence, every node will have the same value for vector $\boldsymbol{s}_i$, i.e. $\boldsymbol{s}_i = \boldsymbol{s}$. Then, exploiting (1) and denoting with $\overline{\boldsymbol{u}}_i^T$ the i-th row of matrix $\mathbf{U}$, every node can compute its local component x_i by simply taking $x_i = \overline{\boldsymbol{u}}_i^T \boldsymbol{s}$. This only requires knowledge, at each node, of the local vector $\overline{\boldsymbol{u}}_i$. Distributed solution methods for *convex* instances of Problem (15) have been widely studied in the literature; they are usually either primal (sub)gradient-based methods or primal-dual schemes. Algorithms belonging to the former class include: i) consensus-based (sub)gradient schemes [17]–[19] along with its accelerated versions [20]–[22]; ii) the (sub)gradient push-method [23]; iii) the dual-average method [24]; and iv) distributed second-order-based schemes [25]. Algorithms for adaptation and learning tasks based on in-network diffusion techniques were proposed in [26]–[28]. The second class of distributed algorithms is that of dual-based techniques. Among them, we mention here only the renowned Alternating Direction Method of Multipliers (ADMM); see [16] for a survey. Distributed ADMM algorithms tailored for specific machine learning problems and parameter estimation in sensor networks were proposed in [29]–[31]. The literature related to *nonconvex* distributed optimization is much more scarce, but we may highlight some important works [32]–[40]. Interestingly, the very recent works in [38]–[40] moved beyond first order stationarity, giving also second-order guarantees on the final solution.

Although there are substantial differences between all the above mentioned approaches, these methods can be generically abstracted as combinations of local optimization steps followed by variable exchanges and averaging of information among neighbors. Since all these methods solve (15) by learning the r-dimensional parameter $\boldsymbol{s}$, they typically need to exchange at least r scalar parameters per iteration (some implementations require even more exchange of data [16], [34]). Furthermore, from a complexity point of view, the simpler (linear) methods require a number of computations that scales as $O(r)$ [18], [19], [26], whereas ADMM typically needs to solve an optimization problem at each iteration [if f is quadratic, ADMM has complexity $O(r^3)$] [16], [31]. Thus, for consensus-based optimization methods, both computational complexity and communication burden increases with the dimension r of the signal subspace.

Contributions. In this paper, differently from all the literature based on consensus-like approaches, we wish to find a solution of the general problem (2) using algorithms where each node, at each iteration, exchanges only a *scalar* variable with its neighbors, *irrespective of the dimension r of the signal subspace*. Eventually, each node reaches the desired value x_i directly, i.e., without the need of building local copies of vector $\boldsymbol{s}$ at each node. More specifically, at a first instance, we are interested in finding solutions of problem (2) using iterative algorithms that can be expressed in the form

$$x_i[k+1] = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x_j[k] - \mu[k] \partial f_i(x_i[k]) \tag{16}$$

where $\mu[k] > 0$ is a step-size sequence, $\partial f_i(x_i)$ denotes the (sub)gradient of $f_i(x_i; y_i)$ and $\mathcal{N}_i$ is the set of neighbors of node i. We will term this new method as Distributed Subspace Projected Optimization (DiSPO). Consensus methods represent a trivial case of (16) corresponding to the very simple scenario where the true vector $\boldsymbol{x}$ is constant or, in terms of (1), $\boldsymbol{x}$ is simply a scalar multiplying the vector $\mathbf{1}$ of all ones.

The crux of DiSPO is the interplay between distributed projections and optimization, which enables us to solve (2) directly in the signal domain so that each node i reconstructs the local value x_i that is compliant with the global subspace constraint $\boldsymbol{x} \in \mathcal{R}(\mathbf{U})$, without the need of reconstructing $\boldsymbol{s}$. This marks a sharp difference with consensus-based methods where all nodes tend to reach an agreement over the r-dimensional vector $\boldsymbol{s}$. Our algorithm is especially suitable for applications like sensor networks, whose goal is to reconstruct a spatial field, so that each node has only to retrieve the value of the filed in its position and not the overall field. Our method presents also clear advantages with respect to consensus methods in terms of communication and computations, because it only requires the exchange of scalar values at each iteration, instead of vectors of dimension r, and less computations at each iteration. This is also corroborated by numerical results showing that our schemes outperform classical distributed methods such as distributed gradient descent [18] and ADMM [16] in terms of practical convergence speed, when applied to instances of the general formulation in (2). Furthermore, under mild assumptions, *DiSPO is proved to converge in both convex and nonconvex scenarios*. As a second instance of our work, we introduce an extension of DiSPO that we call EDiSPO, which is still based on the exchange of scalar values, at each iteration, and enables the use of a constant step size, which leads to faster convergence properties. The price paid for the advantages of our algorithms with respect to consensus-based

approaches is that DiSPO requires a network connectivity that increases with the dimension of the signal subspace (cf. Proposition 2); and the weights w_{ij} depend on the signal subspace, so that they have to be pre-computed and stored in each node before the iterations start [cf. (22)-(23)].

Part of this work was recently presented in [41]; furthermore, in the same venue, another group of researchers came up with similar results in a totally independent fashion, see [42] and the extensions in [43], [44]. The works in [42]–[44] consider a stochastic setting assuming strongly convex and differentiable cost functions, proposing a diffusion adaptation algorithm with provable mean-square performance. Differently from these works, in this paper we consider a deterministic scenario, which however allows more general nondifferentiable, nonconvex cost functions, thus proposing an aggregation plus innovation type of algorithm with provable convergence properties. In this work we extend our preliminary results in [41] adding several important contributions (not contained in [42]–[44]), which can be summarized as:

1) We extend the theory of distributed subspace projection methods initially proposed in [45] and [46] and we highlight an interesting interplay between network connectivity and signal subspace dimension; more specifically, we derive necessary conditions showing that the connectivity of the network has to increase with the dimension r of the signal subspace to guarantee the feasibility of the design problem;
2) Hinging on the initial results of [41], we develop DiSPO, the first algorithmic framework based on distributed subspace projections, aimed at solving the class of problems (2) in a distributed fashion;
3) We provide a detailed convergence analysis for DiSPO, which proves its convergent behavior for different properties [e.g., (non)differentiability, (non)convexity, etc.] of the objective function in (2), giving also convergence rates to the optimal solutions;
4) We propose an Exact DiSPO (EDiSPO) method that provides exact (and fast) convergence to stationary solutions of (2) by exploiting gradient information at two previous instants, while using constant step-size rules;
5) We test the proposed DiSPO/EDiSPO frameworks over realistic data generated using the ray-tracing method of the Wireless InSite Prediction Software [47], and over the real Abilene IP traffic dataset.

Outline. The paper is organized as follows. Section II describes the main theoretical aspects of distributed projections over networks, including optimal design of sparse projection matrices and necessary conditions for the optimization problem feasibility. Section III contains the main theoretical results of the paper: we start with a constructive description of the DiSPO algorithm; then, we introduce formally its convergence properties (cf. Sec. III.A); finally, we extend DiSPO to EDiSPO, which enables exact convergence to stationary solutions of (2) using constant step-sizes (cf. III.B). Section IV assesses the performance of the proposed schemes numerically and compares our methods with other distributed algorithms, considering two practical problems in distributed signal processing. Finally, Section V draws some conclusions.

Notation. Scalar, vector, and matrix variables are indicated by plain letters a, bold lowercase letters $\boldsymbol{a}$, and bold uppercase letters $\mathbf{A}$, respectively. a_{ij} is the (i,j)-th element of $\mathbf{A}$, $\mathbf{I}$ is the identity matrix, and $\mathbf{1}_N$ ($\mathbf{0}_N$) is the $N \times 1$ vector of all ones (zeros). The rank of a matrix is denoted by $\mathrm{rank}(\mathbf{A})$; the matrix spectral norm is given by $\|\mathbf{A}\|_2$ or by the spectral radius $\rho(\mathbf{A})$. $\mathcal{R}(\mathbf{A})$ and $\mathrm{Null}(\mathbf{A})$ denote the range and the nullspace of matrix $\mathbf{A}$, respectively. $f(\boldsymbol{x};\boldsymbol{y})$ denotes a function of the variable $\boldsymbol{x}$, with $\boldsymbol{y}$ being a given parameter. Other notation is defined along the paper, whenever it is needed.

II. Distributed Subspace Projections

In this section we recall and extend the theory related to distributed subspace projections [45], [46], which will form the basic block for the development of the proposed distributed methods. The operation performed in (4) corresponds to the orthogonal projection of the observation vector $\boldsymbol{y}$ onto the subspace spanned by the columns of $\mathbf{U}$. Assuming, without any loss of generality (w.l.o.g.), the columns of $\mathbf{U}$ to be orthonormal, the projector simplifies in

$$\widehat{\boldsymbol{x}} \;=\; \mathbf{U}\mathbf{U}^T\boldsymbol{y} \;=\; \mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}. \tag{17}$$

The aim of this section is to set up a distributed procedure where each node initializes a state variable with the local measurement, let us say $x_i[0] = y_i$, and then it evolves by interacting with nearby nodes in order to compute (17). The nodes are interconnected through a communication network described by the weight matrix $\mathbf{W} = \{w_{ij}\}_{i,j=1}^N \in \mathbb{R}^{N\times N}$, whose sparsity pattern describes its topology, i.e., $w_{ij} = 0$ if nodes i and j do not share a link. Then, $w_{ij} \neq 0$ if $j \in \mathcal{N}_i \cup \{i\}$, and $w_{ij} = 0$ otherwise. Denoting by $\boldsymbol{x}[k]$ the N-size state vector containing the states of all the nodes at iteration k, we let the network state evolve according to the following dynamical system:

$$\boldsymbol{x}[k+1] \;=\; \mathbf{W}\,\boldsymbol{x}[k], \;\; \text{with} \;\; \boldsymbol{x}[0] = \boldsymbol{y}. \tag{18}$$

Clearly, (18) is a distributed procedure because, thanks to the sparsity of $\mathbf{W}$, at each iteration each node interacts (directly) only with its neighborhood. Useful distributed procedures for filtering over graphs were also recently proposed in [48]–[50]. Now, given the interaction mechanism (18), our problem is twofold: 1) guarantee that system (18) converges to the desired vector (17); 2) find the sparse matrix $\mathbf{W}$, under a topological constraint, so that the convergence time is minimized. We will analyze these two key points in the following subsections.

A. Convergence Properties

The dynamical system (18) converges to the desired orthogonal projection of the observation vector $\boldsymbol{y}$ onto $\mathcal{R}(\mathbf{U})$, for any given $\boldsymbol{y} \in \mathbb{R}^N$, if and only if

$$\lim_{k\to\infty} \boldsymbol{x}[k] \;=\; \lim_{k\to\infty} \mathbf{W}^k\boldsymbol{y} \;=\; \mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}, \tag{19}$$

or, equivalently,

$$\lim_{k\to\infty} \mathbf{W}^k \;=\; \mathcal{P}_{\mathcal{R}(\mathbf{U})}. \tag{20}$$

Resorting to basic algebraic properties of discrete-time systems, it is possible to derive immediately some features that matrix $\mathbf{W}$ has to satisfy in order to guarantee (20), as illustrated in the following proposition.

Proposition 1: For any non-null vector $\boldsymbol{y} \in \mathbb{R}^N$, the dynamical system (18) admits a unique globally stable solution given by $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$ if and only if:

$$\text{(C1)} \qquad \mathbf{W}\mathcal{P}_{\mathcal{R}(\mathbf{U})} = \mathcal{P}_{\mathcal{R}(\mathbf{U})}$$
$$\text{(C2)} \qquad \mathcal{P}_{\mathcal{R}(\mathbf{U})}\mathbf{W} = \mathcal{P}_{\mathcal{R}(\mathbf{U})}$$
$$\text{(C3)} \qquad \rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})}) \leq \beta < 1$$

with $\rho(\cdot)$ denoting the spectral radius operator.

Proof. **Sufficiency:** From (18), we have

$$\begin{aligned} \|\boldsymbol{x}[k] - \mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}\|_2 &\overset{(a)}{\leq} \|\mathbf{W}^k - \mathcal{P}_{\mathcal{R}(\mathbf{U})}\|_2 \|\boldsymbol{y}\|_2 \\ &\overset{(b)}{=} \|(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})^k\|_2 \|\boldsymbol{y}\|_2 \\ &\overset{(c)}{=} \left(\rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})\right)^k \|\boldsymbol{y}\|_2 \end{aligned} \tag{21}$$

where in (a) we exploited (19) and the Cauchy-Schwartz inequality; in (b) we exploited the fact that, under (C1) and (C2), we have $\mathbf{W}^k - \mathcal{P}_{\mathcal{R}(\mathbf{U})} = (\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})^k$; finally, (c) follows from the fact that the ℓ_2 norm of a Hermitian matrix coincides with its spectral radius. From (21), if (C3) holds, for any finite vector $\boldsymbol{y} \in \mathbb{R}^N$, the error $\boldsymbol{x}[k] - \mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$ vanishes asymptotically as $k \to \infty$. **Necessity:** (C1) is necessary to guarantee that $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$ is an invariant quantity for the dynamical system (18), i.e., during its evolution (18) keeps the component $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$ of $\boldsymbol{y}$ unaltered. (C2) is necessary to make $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$ a fixed point of matrix $\mathbf{W}$ and thus a potential accumulation point for $\{\boldsymbol{x}[k]\}_k$. Thus, if $\boldsymbol{y}$ is non-null and system (18) converges, (C1) and (C2) are necessary conditions to converge to $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$. Finally, (C3) is necessary to have convergence to the subspace $\mathcal{R}(\mathbf{U})$, since it imposes that all the modes associated to the eigenvectors orthogonal to $\mathcal{R}(\mathbf{U})$ be asymptotically vanishing. ■

As a final remark, conditions (C1)-(C3) contain, as a special case, the convergence conditions of the average consensus algorithm, when $r = 1$ and $\mathbf{U} = \frac{1}{\sqrt{N}}\mathbf{1}_N$. In such a case, (C1)-(C3) can be restated as: the graph associated to the network described by $\mathbf{W}$ must be strongly connected and balanced.

B. Design of the Weight Matrix

In this section we formulate an optimization problem to design a weight matrix $\mathbf{W} = \{w_{ij}\}_{i,j=1}^N \in \mathbb{R}^{N\times N}$ that asymptotically projects a vector onto the desired subspace $\mathcal{R}(\mathbf{U})$ [i.e., (20) holds] with maximum convergence rate, while having a sparsity pattern imposed by a given communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the vertex and edge sets, respectively. This problem was already tackled in [45], where the same design objective considered here was pursued, but assuming a particular structure of the weight matrix given by $\mathbf{W} = \mathbf{I} - \varepsilon\overline{\mathbf{L}}$. The method proposed in [45] proceeded by formulating a nonconvex problem aimed at jointly optimizing the parameter ε and matrix $\overline{\mathbf{L}}$. Here, we generalize the approach in [45], considering matrices $\mathbf{W}$ not necessarily adhering to the previous model. Thus, following the approach of [51], but extending it to enable general subspace projections, we consider the optimization problem:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})}) \\ \text{subject to} \quad & \mathbf{W}\mathcal{P}_{\mathcal{R}(\mathbf{U})} = \mathcal{P}_{\mathcal{R}(\mathbf{U})} \\ & \mathbf{W} = \mathbf{W}^T \\ & \rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})}) \leq \beta < 1 \\ & w_{ij} = 0 \quad \text{for all } (i,j) \notin \mathcal{E}. \end{aligned} \tag{22}$$

As shown in (21), the minimization of the objective function in (22) aims at maximizing the convergence rate of the distributed subspace projection operator. The first three constraints in (22) impose the conditions (C1), (C2) and (C3) on the symmetric matrix $\mathbf{W}$, in order to guarantee convergence to the desired subspace. Finally, given the set $\mathcal{E}$ of edges of the communication graph, the last constraint in (22) imposes a sparsity pattern to matrix $\mathbf{W}$ that reflects the network topology. Notice that, differently from [45], problem (22) is convex and the weight matrix does not require to be built using the model $\mathbf{W} = \mathbf{I} - \varepsilon\overline{\mathbf{L}}$. In fact, since $\rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})$ is a convex function of $\mathbf{W}$ [16], it is easy to check convexity of (22), which can be equivalently recast as a semidefinite program (SDP) by introducing a scalar variable γ to bound $\rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})$, as:

$$\begin{aligned} \min_{\gamma,\mathbf{W}} \quad & \gamma \\ \text{subject to} \quad & -\gamma\mathbf{I} \preceq \mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})} \preceq \gamma\mathbf{I} \\ & \mathbf{W}\mathcal{P}_{\mathcal{R}(\mathbf{U})} = \mathcal{P}_{\mathcal{R}(\mathbf{U})} \\ & \mathbf{W} = \mathbf{W}^T \\ & 0 \leq \gamma \leq \beta < 1 \\ & w_{ij} = 0 \quad \text{for all } (i,j) \notin \mathcal{E} \end{aligned} \Bigg\} \triangleq \mathcal{W} \tag{23}$$

where the symbol $\preceq$ denotes matrix inequality, i.e., $\mathbf{X} \preceq \mathbf{Y}$ means $\mathbf{Y} - \mathbf{X}$ is positive semidefinite. Thus, the global optimal solution of (23) can be found efficiently using standard SDP solvers, which are very efficient at least for small or medium size problems [52]. Finally, since $\mathbf{W} \in \mathbb{R}^{N\times N}$ is a symmetric matrix, one can resort to its half-vectorization form, i.e., $\mathbf{w} = \text{vech}(\mathbf{W}) \in \mathbb{R}^{N(N+1)/2}$ (which contains only the lower triangular part of $\mathbf{W}$), and recast problem (23) in terms of $\mathbf{w}$, thus removing redundant unknowns due to symmetry.

Necessary conditions for feasibility of $\mathcal{W}$. In what follows, we derive necessary conditions for the existence of a solution of problem (23). We start by recasting the set $\mathcal{W}$ in (23) as:

$$\mathcal{W} = \begin{cases} \text{(a)} & \rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})}) \leq \beta < 1 \\ \text{(b)} & \mathbf{W}\mathcal{P}_{\mathcal{R}(\mathbf{U})} = \mathcal{P}_{\mathcal{R}(\mathbf{U})} \\ \text{(c)} & \mathbf{W} = \mathbf{W}^T \\ \text{(d)} & w_{ij} = 0 \quad \text{for all } (i,j) \notin \mathcal{E}. \end{cases} \tag{24}$$

Now, following the arguments in Appendix A and letting $K = |\mathcal{E}|$ be the number of edges in the graph, conditions (b)-(d) in (24) can be rewritten in the following compact form:

$$\mathbf{F}\boldsymbol{z} = \boldsymbol{b}, \tag{25}$$

where $\boldsymbol{z} \in \mathbb{R}^{N+K}$ is the vector of variables collecting the non-null elements of $\mathbf{W}$; whereas $\boldsymbol{b} = \text{vec}(\mathbf{U}) \in \mathbb{R}^{rN}$,

$\mathbf{F} \triangleq \mathbf{BME} \in \mathbb{R}^{rN\times(N+K)}$, with $\mathbf{B} \triangleq \mathbf{U}^T \otimes \mathbf{I}_N$, $\mathbf{M} \in \mathbb{R}^{N^2\times N(N+1)/2}$ being the duplication matrix, and $\mathbf{E} \in \mathbb{R}^{N(N+1)/2\times(N+K)}$ having on each column all zeros entries, except for the row index corresponding to the i-th non-null element of $\mathbf{w}$ (which is set equal to 1) (cf. Appendix A). Let also define $\overline{\mathbf{F}} = \overline{\mathbf{B}}\mathbf{ME} \in \mathbb{R}^{(N-r)N\times(N+K)}$, with $\overline{\mathbf{B}} \triangleq \mathbf{U}_{\bar{r}}^T \otimes \mathbf{I}_N$, where $\mathbf{U}_{\bar{r}} \in \mathbb{R}^{N\times N-r}$ has columns that span the subspace orthogonal to $\mathcal{R}(\mathbf{U})$ (cf. Appendix B). Then, resorting to basic theory for linear systems of equations [53], in the following proposition, we state necessary conditions for the feasibility of $\mathcal{W}$.

Proposition 2: Given the optimization set $\mathcal{W}$ in (24), the following statements hold true:

i) The set of possible solutions is given by $\mathbf{W} = \mathbf{I} - \mathbf{L}$, where $\mathbf{L}$ is any (non-trivial) matrix whose vectorized form $\boldsymbol{\ell}$ is such that $\mathbf{F}\boldsymbol{\ell} = \mathbf{0}$;

ii) $\mathcal{W}$ is feasible if the following necessary conditions are both satisfied:

$$\text{rank}(\mathbf{F}) < N + K \quad \text{and} \quad \text{rank}(\overline{\mathbf{F}}) = N + K; \tag{26}$$

iii) if $\text{rank}(\mathbf{F}) = rN$, (26) implies that the average node degree $\bar{d} := 2K/N$ has to respect the inequality

$$\bar{d} > 2(r-1).$$

Proof. See Appendix B. ∎

The previous proposition, although providing only necessary conditions for the set $\mathcal{W}$ to be feasible, unveils an interesting interplay between signal subspace dimension and network connectivity: It shows that the average degree has to increase with the dimension of the signal subspace.

III. Distributed Subspace Projected Optimization

In this section, we derive a distributed solution method for the class of problems in (2). To this end, letting $\boldsymbol{x}[k]$ be the guess of variable $\boldsymbol{x}$ at time k, a possible starting point might be to implement a (centralized) projected gradient descent algorithm to solve (2), which reads as:

$$\boldsymbol{x}[k+1] = \mathcal{P}_{\mathcal{R}(\mathbf{U})}\big[\boldsymbol{x}[k] - \mu[k]\partial f(\boldsymbol{x}[k])\big], \tag{27}$$

where $\mu[k] > 0$ is a step-size sequence, and $\partial f(\boldsymbol{x})$ denotes the (sub)gradient of $f(\boldsymbol{x};\boldsymbol{y})$[1]. The straightforward implementation of (27) requires a centralized mechanism. The challenging question now is how to implement (27) using a decentralized approach, where each node does not have access to the full matrix $\mathbf{U}$ and it can only share information with its neighbors. To find a distributed solution of (2), we exploit the convergence properties of the projection matrix $\mathbf{W}$ that we designed in the previous section. In particular, note that, from conditions (C1) and (C2), the constraint $\boldsymbol{x} \in \mathcal{R}(\mathbf{U})$ in (2) can be equivalently recast as $\boldsymbol{x} \in \text{Null}(\mathbf{I} - \mathbf{W})$, or, equivalently, $(\mathbf{I} - \mathbf{W})\boldsymbol{x} = \mathbf{0}$. Then, to derive a distributed implementation, we proceed as in penalty optimization methods [54], converting the constrained optimization in (2) into a sequence of penalized unconstrained problems, which at time k write as:[2]

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x};\boldsymbol{y}) + \frac{1}{2\mu[k]}\,\boldsymbol{x}^T(\mathbf{I}-\mathbf{W})\boldsymbol{x} \tag{28}$$

where $\{\mu[k]\}_k$ is a positive non-increasing sequence of scalar parameters, which helps us to force the constraint $\boldsymbol{x} \in \text{Null}(\mathbf{I} - \mathbf{W})$, as $k \to \infty$. In our implementation, at each iteration k, we use one step of (sub)gradient algorithm applied to (28), where $\{\mu[k]\}_k$ takes the role of a step-size sequence, thus obtaining the following recursive rule:

$$\boldsymbol{x}[k+1] \;=\; \mathbf{W}\,\boldsymbol{x}[k] - \mu[k]\partial f(\boldsymbol{x}[k]). \tag{29}$$

Now, assuming the objective function to be separable as in (14), each element of the (sub)gradient in (29) depends only on its corresponding variable, i.e., $\partial f(\boldsymbol{x}[k];\boldsymbol{y}) = [\partial f_1(x_1[k]),\ldots,\partial f_N(x_N[k])]^T$. Exploiting this property and, thanks to the sparsity of matrix $\mathbf{W}$, the recursion (29) is amenable for the distributed implementation illustrated in Algorithm 1, where each node interacts (directly) only with its neighbors. From now on, we shall refer to Algorithm 1 as Distributed Subspace Projected Optimization (DiSPO). DiSPO requires that each node i combines its local estimate $x_i[k]$ with those of its spatial neighbors, i.e., $\{x_j[k]\}_{j\in\mathcal{N}_i}$, using the weighting coefficients $\{w_{ij}\}$. Then, (sub)gradient information of the local loss function is exploited in order to drive the algorithm toward the optimal solution of (2). Algorithm 1 has very low complexity: it requires only $|\mathcal{N}_i|+1$ scalar multiplications and sums per iteration. From a communication point of view, each node needs to exchange only one scalar parameter with its neighbors per iteration. As previously mentioned, this makes a sharp difference with respect to consensus-based methods, whose per-node computational and communication burdens typically increase with the dimension r of the signal subspace. This improvement is obtained thanks to the exploitation of in-network subspace projections, which represent the building block of the proposed DiSPO algorithm.

Algorithm 1: Distributed Subspace Projected Optimization

Data: $x_i[0]$ chosen at random for all i; $\{w_{ij}\}_{i,j}$ satisfying (C1)-(C3); step-size sequence $\{\mu[k]\}_k$. Then, for each time $k \geq 0$ and for each node $i = 1,\ldots,N$, repeat:

$$x_i[k+1] = \sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}x_j[k] - \mu[k]\partial f_i(x_i[k]) \tag{30}$$

A. Convergence Analysis

In this section, we illustrate the convergence properties of the proposed DiSPO Algorithm. Our goal is to develop an algorithm that converges to stationary solutions of Problem (2) while being implementable in the above distributed setting.

Proposition 3: A point $\boldsymbol{x}^* \in \mathcal{R}(\mathbf{U})$ is a stationary solution of Problem (2) if a (sub)gradient $\partial f(\boldsymbol{x}^*)$ exists such that

$$\mathcal{P}_{\mathcal{R}(\mathbf{U})}\partial f(\boldsymbol{x}^*) = \mathbf{0}. \tag{31}$$

[1] To simplify the notation, from now on we omit the dependency of ∂f from the known vector parameter $\boldsymbol{y}$.

[2] If (C1)-(C3) hold, $\rho(\mathbf{W}) = 1$ and $\mathbf{I} - \mathbf{W}$ is positive semidefinite.

Proof. From the minimum principle, any stationary solution $\boldsymbol{x}^*$ of (2) must satisfy

$$\boldsymbol{x}^* = \mathcal{P}_{\mathcal{R}(\mathbf{U})}[\boldsymbol{x}^* - \partial f(\boldsymbol{x}^*)]. \tag{32}$$

Since $\mathcal{P}_{\mathcal{R}(\mathbf{U})}$ is a matrix multiplication and $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{x}^* = \boldsymbol{x}^*$, equation (32) leads immediately to (31). ■

Let $\mathcal{S}$ be the set of stationary solutions of (2). We consider the following assumptions on problem (2), which will characterize the solution set $\mathcal{S}$.

Assumption A [On function f in (2)]: f is continuous and it satisfies a proper combination of the following properties, where (A1-1) and (A1-2) have to be considered as alternative, i.e., f satisfies either (A1-1) or (A1-2). (A2) can be used in combination with (A1-1) or (A1-2) [cf. Theorems 4 and 5]. (A3) holds true in any case.

(A1-1) f is a nondifferentiable, convex function;

(A1-2) f is a differentiable, (possibly) nonconvex function, with Lipschitz continuous gradient, i.e.,

$$\|\partial f(\boldsymbol{x}) - \partial f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \text{ for all } \boldsymbol{x}, \boldsymbol{y};$$

(A2) f has bounded (sub)gradients, i.e., there exists $G > 0$ such that $\|\partial f(\boldsymbol{x})\| \leq G$ for all $\boldsymbol{x}$;

(A3) f is proper (i.e., not everywhere infinite) and coercive, i.e., $\lim_{\|\boldsymbol{x}\|\to\infty} f(\boldsymbol{x}) = +\infty$.

Assumption A is standard and satisfied by many practical problems. Under (A1-1), $\mathcal{S}$ is the set of globally optimal solutions of (2); otherwise, if (A1-2) holds, $\mathcal{S}$ is the set of stationary points of (2). (A2) is a technical assumption typically used in several papers to prove convergence of distributed optimization algorithms, see, e.g., [17]–[19], [33], [34], [36]. Assumption (A3), together with the continuity of f, guarantees the existence of a global minimizer of problem (2). Finally, in this paper, we consider two alternative choices for the step-size sequence $\{\mu[k]\}_k$ in Algorithm 1, which are illustrated in the following assumption.

Assumption B [On the step-size]: The step-size sequence $\{\mu[k]\}_k$ is chosen as:

(B1) a constant, i.e., $\mu[k] = \mu > 0$ for all k;

(B2) a diminishing sequence, see, e.g., [55], chosen such that $\mu[k] > 0$, for all k,

$$\sum_{k=0}^{\infty} \mu[k] = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \mu[k]^2 < \infty.$$

We are now ready to illustrate the convergence properties of DiSPO, which are summarized in the two following Theorems.

Theorem 4: Let $\{\boldsymbol{x}[k]\}_k$ be the sequence generated by Algorithm 1, and $\{\overline{\boldsymbol{x}}[k]\}_k \triangleq \{\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{x}[k]\}_k$ be its projection onto the subspace $\mathcal{R}(\mathbf{U})$; let also $\{\boldsymbol{x}_\perp[k]\}_k = \{\boldsymbol{x}[k] - \overline{\boldsymbol{x}}[k]\}_k$. Suppose that conditions (A2), (A3) and (C1)-(C3) hold. Then, the following results hold.

(a) `[subspace projection]`: Under (B1), the sequence $\{\boldsymbol{x}_\perp[k]\}_k$ satisfies:

$$\lim_{k\to\infty} \|\boldsymbol{x}_\perp[k]\| = O(\mu); \tag{33}$$

if (B2) holds, the sequence $\{\boldsymbol{x}[k]\}_k$ asymptotically converges to the subspace $\mathcal{R}(\mathbf{U})$, i.e.,

$$\lim_{k\to\infty} \|\boldsymbol{x}_\perp[k]\| = 0; \tag{34}$$

(b) `[Convergence for nondifferentiable convex functions]`: Under (A1-1), let $\boldsymbol{x}^* \in \mathcal{S}$ be a a global optimum of (2); let $f^* = f(\boldsymbol{x}^*)$ and $f^{best}[k] = \inf_{n=1,\ldots,k} f(\boldsymbol{x}[n])$. Under (B1), we have:

$$f^{best}[k] - f^* \leq \left(\|\overline{\boldsymbol{x}}[0] - \boldsymbol{x}^*\|^2 + \frac{2G^2\mu^2}{1-\beta}(k+1) + \frac{2G\|\boldsymbol{x}_\perp[0]\|\mu}{1-\beta} + G^2(k+1)\mu^2 \right) \frac{1}{2(k+1)\mu}, \tag{35}$$

and, consequently,

$$\lim_{k\to\infty} f^{best}[k] - f^* = \mathcal{O}(\mu), \tag{36}$$

with convergence rate $\mathcal{O}\left(\frac{1}{k+1}\right)$; under (B2), we get:

$$\lim_{k\to\infty} f^{best}[k] = f^*; \tag{37}$$

Finally, if $\mu[k] = \dfrac{1}{\sqrt{k+1}}$, we obtain:

$$f^{best}[k] - f^* \leq \frac{C_2 + \frac{G^2(3-\beta)}{2(1-\beta)}\log(k+1)}{\sqrt{k+1}}, \tag{38}$$

where $C_2 < \infty$ is a given constant, i.e., (37) hold with a convergence rate $\mathcal{O}\left(\frac{\log(k+1)}{\sqrt{k+1}}\right)$.

Proof. See Appendix D. ■

The next Theorem presents the results for the differentiable, (possibly) nonconvex case. To this aim, let us introduce the Lyapunov potential function [cf. (28)]:

$$J(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{y}) + \frac{1}{2\mu}\,\boldsymbol{x}^T(\mathbf{I} - \mathbf{W})\boldsymbol{x}, \tag{39}$$

and the performance metric

$$g[k] = \|\partial f(\overline{\boldsymbol{x}}[k])\|^2_{\mathcal{P}_{\mathcal{R}(\mathbf{U})}}, \tag{40}$$

which quantifies proximity to a stationary solution of (2) [cf. (31)]. Also, let $g^{best}[k] = \inf_{n=1,\ldots,k} g[n]$. The eigenvalues of $\mathbf{W}$ are ordered as $1 = \lambda_1(\mathbf{W}) \geq \lambda_2(\mathbf{W}) \geq \ldots \geq \lambda_N(\mathbf{W}) > -1$.

Theorem 5: `[Convergence for differentiable nonconvex functions]`. Suppose that conditions (A1-2), (A3), and (C1)-(C3) hold. Then, the following results hold. If $0 < \mu < L^{-1}(1 + \lambda_N(\mathbf{W}))$, the sequence $\{\boldsymbol{x}[k]\}_k$ converges to a stationary point of (39), i.e.,

$$\lim_{k\to\infty} \partial J(\boldsymbol{x}[k]) = \mathbf{0}. \tag{41}$$

Let also (A2) hold true. Then, under (B1), we have:

$$g^{best}[k] \leq \left(f(\overline{\boldsymbol{x}}[0]) + \frac{LG\|\boldsymbol{x}_\perp[0]\|\mu}{1-\beta} + \frac{L}{2}(k+1)G^2\mu^2 + \frac{LG^2\mu^2}{1-\beta}(k+1) \right) \frac{1}{(k+1)\mu}, \tag{42}$$

and, consequently,

$$\lim_{k\to\infty} g^{best}[k] = \mathcal{O}(\mu), \tag{43}$$

with convergence rate $\mathcal{O}\left(\frac{1}{k+1}\right)$; if $\mu[k] = \frac{1}{\sqrt{k+1}}$, we get:

$$g^{best}[k] \leq \frac{C_4 + \frac{LG^2(3-\beta)}{2(1-\beta)}\log(k+1)}{\sqrt{k+1}}, \tag{44}$$

where $C_4 < \infty$ is a given constant, i.e., $\lim_{k\to\infty} g^{best}[k] = 0$ with a convergence rate $\mathcal{O}\left(\frac{\log(k+1)}{\sqrt{k+1}}\right)$; finally, if (B2) holds,

$$\lim_{k\to\infty} g[k] = 0, \tag{45}$$

i.e., $\{\overline{\boldsymbol{x}}[k]\}_k$ converges to a stationary solution of (2).

Proof. See Appendix E. ∎

Remark: Interestingly, the convergence rates obtained for the convex and nonconvex cases have very similar expressions, which depend on the network topology, the subspace constraint $\mathcal{R}(\mathbf{U})$, the properties of the objective function, and the adopted step-size [cf. (35) with (42), and (38) with (44)]. To the best of our knowledge, this is a novel result that does not have a direct counterpart in the literature of distributed consensus optimization, see, e.g., [36, Table 1]. In general, using a fixed step-size, DiSPO converges to a point in the $O(\mu)$-neighborhood of a solution to (2) [cf. (43)]. On the other hand, properly letting $\mu[k]$ to vanish asymptotically [cf. (B2)], it is possible to enable exact convergence, i.e., $\{\boldsymbol{x}[k]\}_k$ converges to the exact solution [cf. (34), (45)]. However, reducing $\mu[k]$ causes slower convergence, as it will be shown numerically in Section IV. In the next section, we will show how to modify DiSPO to achieve exact convergence using constant step-sizes.

B. Exact Distributed Subspace Projected Optimization

As illustrated in Theorem 5, [cf. (41)], with a fixed (and sufficiently small) step-size, DiSPO has *inexact convergence*, i.e., the sequence $\{\boldsymbol{x}[k]\}_k$ converges to the set of stationary points of the Lyapunov function J in (39), which in general does not coincide with the solution set $\mathcal{S}^*$ of (2). To see this, let $\boldsymbol{x}^\infty$ be the limit of $\{\boldsymbol{x}[k]\}_k$ as $k \to \infty$, and consider

$$\partial J(\boldsymbol{x}^\infty) = \partial f(\boldsymbol{x}^\infty) + (\mathbf{I} - \mathbf{W})\boldsymbol{x}^\infty. \tag{46}$$

Left multiplying (46) by $\mathcal{P}_{\mathcal{R}(\mathbf{U})}$, since $\mathcal{P}_{\mathcal{R}(\mathbf{U})}(\mathbf{I} - \mathbf{W}) = \mathbf{0}$, we obtain:

$$\mathcal{P}_{\mathcal{R}(\mathbf{U})}\partial f(\boldsymbol{x}^\infty) = \mathbf{0}, \tag{47}$$

which is the optimality condition of problem (2) stated in Proposition 3 [cf. (31)]. However, since for fixed step-sizes $\boldsymbol{x}^\infty$ is not guaranteed to belong to $\mathcal{R}(\mathbf{U})$ [cf. (33)], condition (47) alone does not guarantee optimality of $\boldsymbol{x}^\infty$.

In the sequel, proceeding as in [21], we will modify the DiSPO method in order to guarantee *exact convergence* also when using a constant step-size. Let us consider the DiSPO updating rule over two consecutive iterations:

$$\boldsymbol{x}[k+2] \;=\; \mathbf{W}\,\boldsymbol{x}[k+1] - \mu\partial f(\boldsymbol{x}[k+1]) \tag{48}$$

$$\boldsymbol{x}[k+1] \;=\; \widetilde{\mathbf{W}}\,\boldsymbol{x}[k] - \mu\partial f(\boldsymbol{x}[k]) \tag{49}$$

Algorithm 2: Exact Distributed Subspace Projected Optimization

Data: $x_i[0]$ chosen at random for all i; $\{w_{ij}\}_{i,j}$ satisfying (C1)-(C3); $\{\widetilde{w}_{ij}\}_{i,j}$ as in (50); (small) step-size $\mu > 0$; set

$$x_i[1] = \sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}x_j[0] - \mu\partial f_i(x_i[0])$$

for all i. Then, for each time $k \geq 1$ and for each node $i = 1,\dots,N$, repeat:

$$x_i[k+2] = x_i[k+1] + \sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}x_j[k+1] \tag{52}$$
$$- \sum_{j\in\mathcal{N}_i\cup\{i\}} \widetilde{w}_{ij}x_j[k] - \mu\left[\partial f_i(x_i[k+1]) - \partial f_i(x_i[k])\right]$$

where (48) uses the mixing matrix $\mathbf{W}$, while (49) uses

$$\widetilde{\mathbf{W}} = \frac{\mathbf{I} + \mathbf{W}}{2}. \tag{50}$$

The Exact Distributed Subspace Optimization (EDiSPO) rule is obtained subtracting (49) from (48):

$$\boldsymbol{x}[k+2] \;=\; (\mathbf{I} + \mathbf{W})\,\boldsymbol{x}[k+1] - \widetilde{\mathbf{W}}\boldsymbol{x}[k] \\ - \mu\big[\partial f(\boldsymbol{x}[k+1]) - \partial f(\boldsymbol{x}[k])\big]. \tag{51}$$

Due to the sparsity of matrices $\mathbf{W}$ and $\widetilde{\mathbf{W}}$ and the separability of function f [cf. (14)], the EDiSPO recursion (51) enjoys the distributed implementation illustrated in Algorithm 2. EDiSPO requires that each node i combines its local estimate with those of its spatial neighbors at two consecutive times $k+1$ and k using a different set of weights, i.e., $\{w_{ij}\}_{i,j}$ and $\{\widetilde{w}_{ij}\}_{i,j}$, respectively. Then, the difference of (sub)gradient information of the local loss function at two consecutive times is exploited in order to enable exact convergence of the algorithm toward the optimal solution of (2). Algorithm 2 has very low complexity: it requires only $2(|\mathcal{N}_i| + 1)$ scalar multiplications and sums per iteration. From a communication point of view, each node needs to exchange only one scalar parameter with its neighbors per iteration.

On the convergence of EDiSPO. We will now illustrate the exact convergence property of EDiSPO by showing that, if Algorithm 2 converges, then its limit point must be a stationary solution of problem (2). To this end, let us assume that $\{\boldsymbol{x}[k]\}_k$ generated by Algorithm 2 converges to $\boldsymbol{x}^\infty$ as $k \to \infty$. Let us also assume that ∂f is continuous. Then, considering the EDiSPO rule (51) at convergence, using (50), and multiplying by 2 both sides of the equation, we have:

$$2(\widetilde{\mathbf{W}} - \mathbf{W})\,\boldsymbol{x}^\infty \;=\; (\mathbf{I} - \mathbf{W})\,\boldsymbol{x}^\infty \;=\; \mathbf{0}. \tag{53}$$

Therefore, $\boldsymbol{x}^\infty \in \mathrm{Null}(\mathbf{I} - \mathbf{W})$ or, equivalently, $\boldsymbol{x}^\infty \in \mathcal{R}(\mathbf{U})$ [cf. (C1) and (C2)]. Now, using (51) and then applying telescopic cancelation, we obtain:

$$\boldsymbol{x}[k+2] \;=\; \mathbf{W}\boldsymbol{x}[k+1] - \mu\partial f(\boldsymbol{x}[k+1]) \\ + \sum_{l=0}^{k}(\mathbf{W} - \widetilde{\mathbf{W}})\boldsymbol{x}[l]. \tag{54}$$

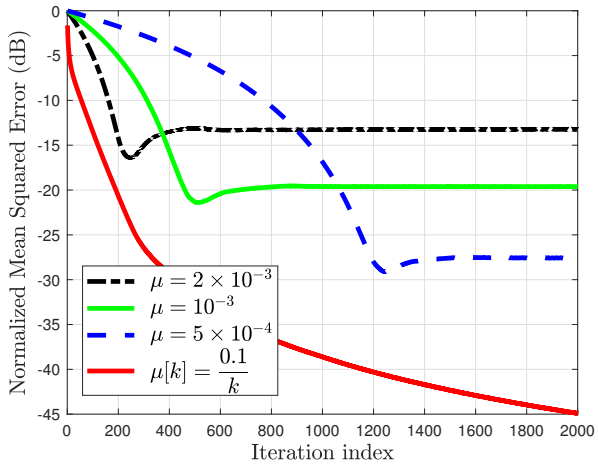

Fig. 1: Normalized Mean Squared Error versus iteration index, for different selection of the step-size.

At convergence, since $\lim_{k\to\infty} \boldsymbol{x}[k] = \boldsymbol{x}^\infty$ and $\mathbf{W}\boldsymbol{x}^\infty = \boldsymbol{x}^\infty$, from (54) we get:

$$\mu \partial f(\boldsymbol{x}^\infty) = -\sum_{l=0}^{k} (\widetilde{\mathbf{W}} - \mathbf{W})\boldsymbol{x}[l]. \tag{55}$$

Left multiplying by $\mathcal{P}_{\mathcal{R}(\mathbf{U})}$ on both sides of (55), and exploiting (50) and (C2), since $\mathcal{P}_{\mathcal{R}(\mathbf{U})}(\widetilde{\mathbf{W}} - \mathbf{W}) = \frac{1}{2}\mathcal{P}_{\mathcal{R}(\mathbf{U})}(\mathbf{I} - \mathbf{W}) = \mathbf{0}$, we finally obtain $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\partial f(\boldsymbol{x}^\infty) = \mathbf{0}$, which is the optimality condition of problem (2) stated in Proposition 1. To summarize, if $\text{Null}(\mathbf{I} - \mathbf{W}) = \mathcal{R}(\mathbf{U})$ and $\widetilde{\mathbf{W}}$ is chosen as in (50), if the sequence generated by Algorithm 2 converges to a point $\boldsymbol{x}^\infty$, then $\boldsymbol{x}^\infty$ belongs to $\mathcal{R}(\mathbf{U})$ and is a stationary point of problem (2). A detailed convergence analysis of EDiSPO is beyond the aim of this paper and will be considered in a future publication. Nevertheless, the excellent convergence properties of EDiSPO will be illustrated numerically in Sec. IV.

IV. Applications and Numerical Results

In this section, we customize the proposed framework to specific distributed signal processing tasks, encompassing signal recovery in the presence of strong impulsive noise and interpolation of graph signals. Numerical results confirm the theoretical findings, and illustrate that DiSPO and EDiSPO compare favorably with respect to other distributed schemes. The Matlab codes used for simulations are available online [3].

Example #1 - Distributed Signal Recovery in the Presence of Outliers. Let us consider a sensor network composed of $N = 60$ nodes randomly scattered over a 2D domain. The observation is composed of a smooth signal $\boldsymbol{x}^o$, modeled as in (1), where the columns of $\mathbf{U}$ are the 2D Fourier low frequency components (we used $r = 5$ components), plus a very strong impulsive noise that affects a (randomly chosen) subset of $|\mathcal{C}| = 20$ nodes. The signal-to-noise ratio in the noisy nodes is set equal to -20 dB, whereas in the other nodes the noise is negligible. Under such a setting, the algorithm in (11) is able to recover the useful signal perfectly, *irrespective of the noise power*, using a centralized approach [14], [56]. Here, we provide a *distributed* solution, exploiting the theory developed before, based only on the exchange of *scalar* data between neighbor nodes. We consider a communication topology built using a small world random graph model, having an average node degree equal to 10, and a rewiring probability given by 0.2. The graph weighting matrix $\mathbf{W}$ has been selected by solving (23) with $\beta = 0.99$. In Fig. 1, we illustrate the behavior of the normalized mean square error (NMSE), i.e., $\|\boldsymbol{x}[k] - \boldsymbol{x}^o\|^2/\|\boldsymbol{x}^o\|^2$ averaged over 100 independent realizations, versus the iteration index, obtained by the DiSPO algorithm considering different choices for the step-size sequence $\{\mu[k]\}_k$. As we can notice from Fig. 1, using a constant step-size, the algorithm converges to a final solution with an error that decreases as we select a smaller step value, even though this comes at the cost of a slower convergence time. On the other hand, if we use the diminishing step-size rule $\mu[k] = 0.1/k$ [which satisfies (B2)], from Fig. 1 we can observe how the algorithm keeps learning over time, thus asymptotically converging to the true signal $\boldsymbol{x}^o$, which represents the optimal solution of the centralized problem (11). These numerical results are in line with the theoretical findings of Theorem 4 in (36) and (37), respectively.

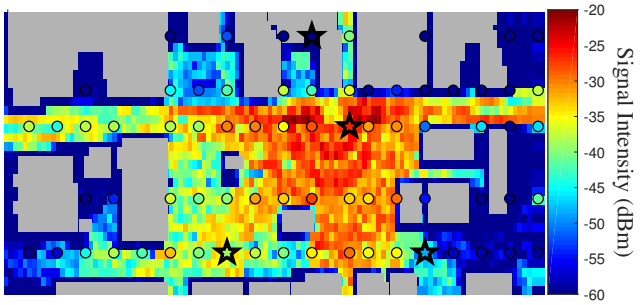

Fig. 2: EM field generated by one radio transmitter over a small area in Ottawa. Gray spots represent buildings.

Example #2 - Distributed Interpolation of Graph Signals. In this example, we considered the electromagnetic (EM) field generated using the ray-tracing tool of the Wireless InSite Prediction Software [47] applied to a small area of Ottawa, Canada, illuminated by one radio base station. The resulting EM spatial power distribution (in dBm) is illustrated in Fig. 2. We also assume the presence of $N = 76$ sensor nodes that are scattered over the area in the way depicted in Fig. 2. The goal of the sensor network is to recover the intensity of the EM signal at each node via interpolation from a small number of collected measurements. To this aim, we exploit the theory of sampling and recovery of signals defined over graphs [13]. Thus, considering the sensor nodes as vertices of a graph, we build edges that encode similarities among the signals at different locations, i.e., the weight a_{ij} associated with the ij-th element of the adjacency matrix of the graph is given by $w_{ij} = \exp\{-[(y_i - y_j)^2]/(2\sigma^2)\}$, with $\sigma = 5$, where y_i is the measurement collected at node i. From now on, we will refer to this graph as the processing graph. The EM signal turns out to be very smooth over the processing graph. In fact, about 99.9% of the signal energy is concentrated over the first four frequencies of the Graph Fourier Transform (GFT) computed

[3] https://sites.google.com/site/paolodilorenzohp/publications

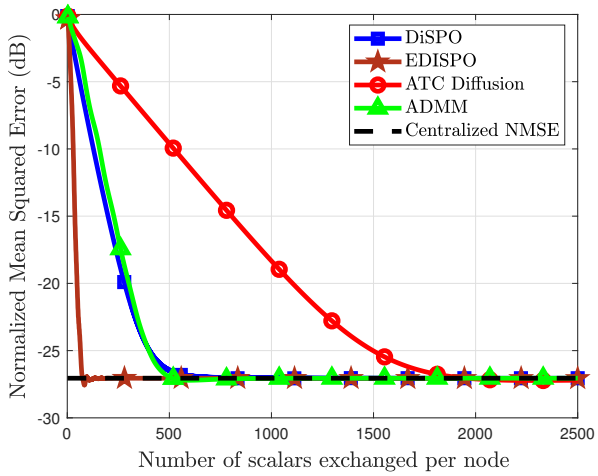

Fig. 3: Normalized mean squared error versus number of scalars exchanged per iteration, for different algorithms.

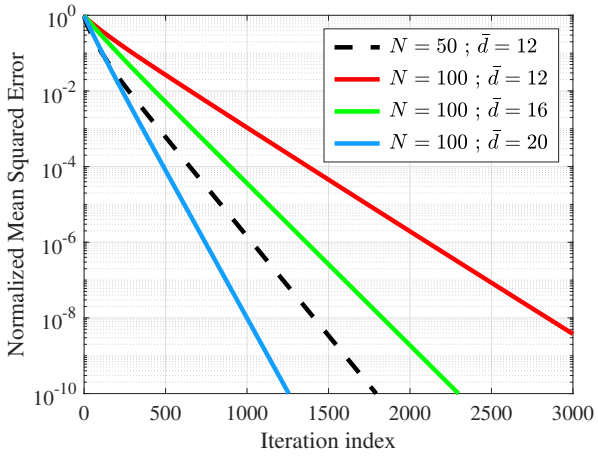

Fig. 4: Normalized mean squared error versus iteration index, for different network size N and average degree $\bar{d}$.

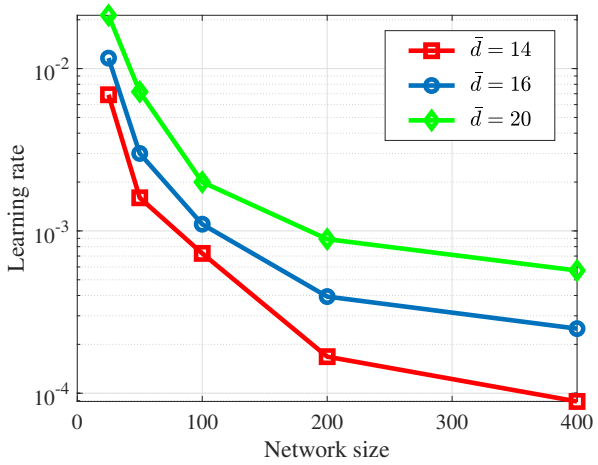

Fig. 5: Learning rate versus network size, for different values of average degree $\bar{d}$.

using the eigenvectors of the Laplacian matrix [12]. Then, the network goal translates in solving the optimization problem in (9), where $\mathbf{U}$ is composed of the first four eigenvectors of the Laplacian matrix of the processing graph, whereas the sampling operator $\mathbf{D}$ is built selecting only four samples according to the MinPinv strategy proposed in [13]. The four selected nodes are depicted as stars in Fig. 2.

To find a distributed solution to problem (9), we set up a communication graph among the nodes using a k-nearest neighbor graph, with $k = 15$. In Fig. 3, we illustrate the behavior of the NMSE, i.e., $\mathbb{E}\{\|\boldsymbol{x}[k] - \boldsymbol{x}^o\|^2/\|\boldsymbol{x}^o\|^2\}$, with $\boldsymbol{x}^o$ denoting the true graph signal, versus the number of scalars exchanged per iteration by each node, considering four different algorithms: the DiSPO algorithm in (30); the ATC diffusion algorithm proposed in [57] for distributed learning of graph signals; the ADMM algorithm from [31] aimed to solve (9) recast as a consensus optimization problem [cf. (15)]; the EDiSPO strategy in (52). The curves are averaged over 100 independent simulations, considering an additive Gaussian observation noise with variance equal to -50 dBm. In our simulations we have observed that other distributed gradient-based consensus strategies have similar performance to ATC diffusion, and we did not report the results. Furthermore, we have chosen the ADMM algorithm from [31], among all the available distributed ADMM implementations, because it is the one with less exchange of parameters among nodes per iteration. The parameters of the selected four algorithms are described in the sequel. The weight matrix $\mathbf{W}$ of DiSPO and EDiSPO algorithms was selected solving (23) with $\beta = 0.99$. The step-sizes for DiSPO and EDiSPO have been set equal to $\mu = 0.43$ and $\mu = 1.1$, respectively. For the ATC Diffusion algorithm we used a Metropolis weighting matrix and a constant step-size equal to 15. The ADMM implementation used a regularization parameter $c = 10^{-3}$. The rationale underlying our parameter selection has been the maximization of the empirical convergence speed for all methods. In Fig. 3, the horizontal dashed line represents the optimal NMSE obtained solving problem (9) in a centralized fashion, as a benchmark to evaluate the performance of distributed approaches. As we can notice from Fig. 3, all algorithms converge to the centralized solution. For DiSPO, this comes from (41), when we add the strongly convex structure of the objective in (9). Furthermore, DiSPO largely outperforms the ATC diffusion method from [57], while showing similar performance with ADMM. Remarkably, DiSPO achieves such performance with a very low complexity per iteration (cf. Algorithm 1), which is much smaller than $O(r^3)$ required by ADMM in such implementation. Finally, we can notice how EDiSPO largely outperforms all other methods, while having a computational complexity per iteration similar to the DiSPO algorithm. This very fast convergence behavior is due to the exploitation of gradient information at two previous steps, which leads to the exact convergence of the method.

Example #3 - The effect of network size and connectivity. In this paragraph, we illustrate the scalability of the proposed methods with respect to network size and connectivity. The observation is composed of a smooth signal $\boldsymbol{x}^o$, modeled as in (7), where the columns of $\mathbf{U}$ are the 2D Fourier low frequency components (we used $r = 4$ components); the

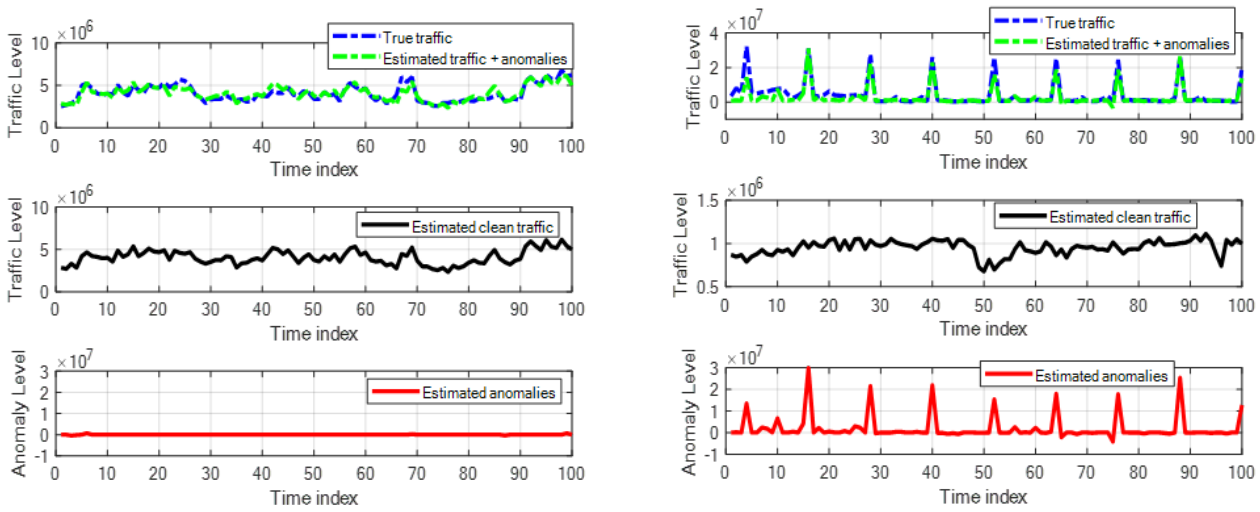

Fig. 6: Temporal behavior of the true signal, and the estimated nominal and anomalous components, considering a link with no anomalies (left) and one with anomalies (right).

sampling operator $\mathbf{D}$ is chosen according to the MinPinv strategy proposed in [13], where the number of samples is equal to the size r of the subspace spanned by the columns of $\mathbf{U}$. Under such a setting, the solution of (9) enables to interpolate exactly $\boldsymbol{x}^o$ over the unobserved values. The network composed of N nodes randomly scattered over a 2D domain. To find a distributed solution to problem (9), we set up a communication graph among the nodes using a small world random graph model, having an average node degree equal to $\bar{d}$, and a rewiring probability given by 0.25. Thus, in Fig. 4 we illustrate the behavior of the NMSE achieved by the DiSPO algorithm versus the iteration index, for different network sizes N and average degrees $\bar{d}$. The curves are averaged over 100 independent graph realizations; and the constant step-size is chosen empirically to maximize the learning rate for each pair N and $\bar{d}$. As we can notice from Fig. 4, the algorithm shows linear convergence. As expected, fixing $\bar{d}$, the convergence speed diminishes by increasing the network size; whereas, interestingly, fixing N, the algorithm shows faster behavior by increasing the average connectivity $\bar{d}$. This behavior is further illustrated in Fig. 5, which illustrates the learning rate (i.e., the absolute value of the slope of the lines in Fig. 4) of the DiSPO algorithm, averaged over 100 independent graph realizations, with respect to the network size N, for different values of $\bar{d}$. As we can notice, from Fig. 5, the learning rate tends to diminish by increasing N; a fact that can be properly counteracted by increasing the average network connectivity.

Example #4 - Tracking and Anomalography of Traffic Data in IP Networks. We consider the Abilene dataset [58], which contains aggregate flow of traffic data (i.e., the link counts) based on measurements of origin-destination (OD) flows on the Abilene Internet 2 network. Overall, the network is composed of 11 nodes, 41 links, and 121 OD flows. In this case, the observed process is defined over the edges of the graph, and is composed by the superposition of a smooth component (i.e., the nominal traffic data) and a sparse component (i.e., the anomalies) as in (12). Thus, the network goal is to distributively learn, track, and separate the two components of the (partially) observed graph process, by solving an online version of problem (13) with streaming data. We assume that the nominal graph process can be considered (approximatively) band-limited over the first six eigenvectors of the edge Laplacian matrix. The communication graph was selected as a k-nearest neighbor graph, with $k = 12$, and the weight matrix $\mathbf{W}$ was selected solving (22) with $\beta = 0.99$. Then, we apply the proposed EDiSPO method (with $\mu = 0.01$), assuming that each link is sampled at each time with a probability $p_i = 0.3$, for all $i \in \mathcal{E}$. In Fig. 6, we illustrate the temporal behavior of the true signal, and the estimated nominal and anomalous components, over two links of the network, considering absence of anomalies on the left side, and presence of anomalies on the right side. As we can notice from Fig. 6, the method is capable to recover and track the dynamic signal in a totally distributed fashion, while separating efficiently the nominal and anomalous components.

V. Conclusions

In this paper we have introduced DiSPO, a novel algorithmic framework for distributed optimization and processing of subspace-constrained signals in multi-agent networks. DiSPO exploits (sub)gradient optimization techniques while leveraging distributed projections as a mechanism to enforce subspace constraints in a cooperative and distributed fashion. A detailed theoretical analysis has been carried out to illustrate the convergence properties of DiSPO, and to pave the way towards the EDiSPO strategy, which enables faster convergence to the optimal solutions. Then, we have customized our framework to specific signal processing tasks over networks, thus illustrating how the proposed methods compare favorably with respect to existing distributed algorithms. The key feature of the proposed methods is that the nodes need to exchange only a scalar parameter with their neighbors, irrespective of the size

of the unknown parameter vector to be estimated. The price paid for this advantage is that the number of neighbors tends to increase with the size of the underlying parameter vector. In this work, we have still assumed a centralized computation of the weights w_{ij} to be sent to all nodes to enable their distributed computations. However, this is the only information to be sent to all nodes. The nodes do not need to know the dictionary $\mathbf{U}$. A future interesting development will be to derive decentralized mechanisms to get the mixing matrix $\mathbf{W}$.

APPENDIX A
LINEAR SYSTEM ASSOCIATED WITH (b)–(d) IN (24)

Let us start from equation (b) in (24). Right multiplying both sides by $\mathbf{U}$, and using the equality $\mathbf{U}^T\mathbf{U} = \mathbf{I}_r$, we get

$$\mathbf{W}\mathbf{U} = \mathbf{U}. \tag{56}$$

Since from (c) $\mathbf{W} \in \mathbb{R}^{N\times N}$ is symmetric, we can exploit its half-vectorization form, i.e., $\mathbf{w} = \text{vech}(\mathbf{W})$, which contains only the $N(N+1)/2$ entries on and below the main diagonal of $\mathbf{W}$. In particular, defining $\mathbf{M} \in \mathbb{R}^{N^2\times N(N+1)/2}$ as the duplication matrix, we have

$$\text{vec}(\mathbf{W}) = \mathbf{M}\mathbf{w}. \tag{57}$$

Now, from (56) and (57), we obtain:

$$\text{vec}(\mathbf{W}\mathbf{U}) = (\mathbf{U}^T \otimes \mathbf{I}_N)\text{vec}(\mathbf{W}) = \mathbf{B}\mathbf{M}\mathbf{w}, \tag{58}$$

where $\mathbf{B} \triangleq \mathbf{U}^T \otimes \mathbf{I}_N$. We now incorporate condition (d) in (24). Note that, letting $K = |\mathcal{E}|$ be the number of edges in the communication graph, the number of nonzero elements in $\mathbf{w}$ reduces to $N + K$, considering also the N non-zero diagonal elements of $\mathbf{W}$. Hence, introducing the matrix $\mathbf{E} \in \mathbb{R}^{N(N+1)/2\times(N+K)}$, having on each column all zeros entries, except for the row index corresponding to the i-th non-null element of $\mathbf{w}$ (which is set equal to 1), we obtain:

$$\mathbf{w} = \mathbf{E}\boldsymbol{z}, \tag{59}$$

where $\boldsymbol{z} \in \mathbb{R}^{N+K}$ contains the non-null elements of $\mathbf{w}$. Thus, using (59) in (58), and exploiting (56), equations (b)-(d) in (24) can be recast in compact form as:

$$\mathbf{F}\boldsymbol{z} = \boldsymbol{b}, \tag{60}$$

with $\mathbf{F} \triangleq \mathbf{B}\mathbf{M}\mathbf{E} \in \mathbb{R}^{rN\times(N+K)}$ and $\boldsymbol{b} = \text{vec}(\mathbf{U})$.

APPENDIX B
PROOF OF PROPOSITION 2

Point i): We begin observing that equation (b) in system (24) reduces, after right multiplying both sides by $\mathbf{U}$, to the form $\mathbf{W}\mathbf{U} = \mathbf{U}$ [cf. (56)]. This last equation admits $\mathbf{W} = \mathbf{I}$ as a possible solution. We can associate with this solution its vector form in (59), i.e., the $N+K$-dimensional vector $\boldsymbol{z}_I$ such that $\mathbf{w} = \text{vech}(\mathbf{W}) = \text{vech}(\mathbf{I}) = \mathbf{E}\boldsymbol{z}_I$. Then, $\boldsymbol{z}_I$ is one of the solutions of (25) [cf. (60)]. Then, the solution set of $\mathbf{F}\boldsymbol{z} = \boldsymbol{b}$ is given by $\boldsymbol{z}_I - \text{Null}(\mathbf{F})$, where $\text{Null}(\mathbf{F})$ denotes the nullspace of $\mathbf{F}$, i.e., the solution set of the homogeneous system $\mathbf{F}\boldsymbol{z} = \mathbf{0}$ [53, Th. 2.6.3]. This implies that the set of feasible solutions in $\mathcal{W}$ can be generally expressed as $\mathbf{W} = \mathbf{I}-\mathbf{L}$, with $\mathbf{L}$ denoting any matrix whose vectorization form, say $\boldsymbol{\ell}$, is solution of system $\mathbf{F}\boldsymbol{\ell} = \mathbf{0}$. Also, since $\mathbf{W} = \mathbf{I}$ does not satisfy (a) in (24), $\mathbf{L}$ cannot be the null matrix $\mathbf{0}$. This proves point i).

Point ii): The first condition in (26) comes from observing that system $\mathbf{F}\boldsymbol{z} = \mathbf{0}$ admits at least one (not-trivial) solution if and only if the rank of $\mathbf{F}$ is less than the number of variables, i.e., $\text{rank}(\mathbf{F}) < N + K$. To prove the second condition in (26), we need to ensure that condition (a) in (24) holds true. To this aim, let us introduce the matrix $\mathbf{U}_{\bar{r}} \in \mathbb{R}^{N\times N-r}$, whose columns span the subspace orthogonal to $\mathcal{R}(\mathbf{U})$. Now, a necessary condition for having (a) in (24) is that $\mathbf{W}\mathbf{U}_{\bar{r}} \neq \mathbf{U}_{\bar{r}}$ or, equivalently, $\mathbf{L}\mathbf{U}_{\bar{r}} \neq \mathbf{0}$ by using the fact that $\mathbf{W} = \mathbf{I}-\mathbf{L}$, as proved in point i). It is then fundamental to investigate under which conditions the system $\mathbf{L}\mathbf{U}_{\bar{r}} = \mathbf{0}$ admits no solution. Following the same steps described in Appendix A to derive (25), system $\mathbf{L}\mathbf{U}_{\bar{r}} = \mathbf{0}$ reduces to the form $\overline{\mathbf{F}}\boldsymbol{z} = \mathbf{0}$, where $\overline{\mathbf{F}} \in \mathbb{R}^{(N-r)N\times(N+K)}$ writes as:

$$\overline{\mathbf{F}} = \overline{\mathbf{B}}\mathbf{M}\mathbf{E}, \tag{61}$$

with $\overline{\mathbf{B}} \triangleq \mathbf{U}_{\bar{r}}^T \otimes \mathbf{I}_N$, whereas $\mathbf{M}$ and $\mathbf{E}$ are defined in Appendix A. Then, a necessary and sufficient condition for $\overline{\mathbf{F}}\boldsymbol{z} = \mathbf{0}$ to admit no (not-trivial) solution is that $\text{rank}(\overline{\mathbf{F}}) = N + K$. This proves the second condition in (26), and also point ii).

Point iii): Assuming $\text{rank}(\mathbf{F}) = rN$, since from point ii) we have $\text{rank}(\mathbf{F}) < N + K$, we easily get $K > N(r-1)$. Thus, the average node degree $\bar{d} := 2K/N$ must satisfy $\bar{d} > 2(r-1)$. This concludes the proof of iii).

APPENDIX C
SOME USEFUL LEMMAS

Lemma 6: Let $0 < \gamma < 1$, and let $\{\alpha[k]\}$ and $\{\nu[k]\}$ be two positive scalar sequences. Then, the following hold:

(a) If $\lim_{k\to\infty} \alpha[k] = 0$, then

$$\lim_{k\to\infty} \sum_{l=1}^{k} \gamma^{k-l}\alpha[l] = 0. \tag{62}$$

(b) If $\sum_{k=1}^{\infty} \alpha[k]^2 < \infty$ and $\sum_{k=1}^{\infty} \nu[k]^2 < \infty$, then

$$\lim_{k\to\infty} \sum_{n=1}^{k}\sum_{l=1}^{n} \gamma^{n-l}\alpha[n]\nu[l] < \infty. \tag{63}$$

Proof. (a) can be found in [19]; (b) was proved in [34]. ■

Lemma 7: Let $\{Y[k]\}$, $\{X[k]\}$, and $\{Z[k]\}$ be three scalar sequences such that $X[k] \geq 0$ for all k. Suppose that

$$Y[k+1] \leq Y[k] - X[k] + Z[k], \quad \text{for all } k,$$

and $\sum_{k=1}^{\infty} Z[k] < \infty$. Then, either $Y[k] \to -\infty$ or else $\{Y[k]\}$ converges to a finite value, and $\sum_{k=1}^{\infty} X[k] < \infty$.

Proof. The proof can be found in [59, Lemma 1]. ■

APPENDIX D
PROOF OF THEOREM 4

Point (a): Let $\mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp} = \mathbf{I} - \mathcal{P}_{\mathcal{R}(\mathbf{U})}$ be the projector onto the subspace orthogonal to $\mathcal{R}(\mathbf{U})$. We will now study the temporal evolution of

$$\boldsymbol{x}_\perp[k] = \mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}\boldsymbol{x}[k] = \boldsymbol{x}[k] - \underbrace{\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{x}[k]}_{\overline{\boldsymbol{x}}[k]}, \tag{64}$$

i.e., the component of $\boldsymbol{x}[k]$ that lies in the subspace orthogonal to $\mathcal{R}(\mathbf{U})$. To this aim, multiplying (29) from the left side by $\mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}$, and letting $\boldsymbol{c}[k] = \mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}\partial f(\boldsymbol{x}[k])$, we obtain:

$$\mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}\boldsymbol{x}[k+1] \;=\; \mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}\mathbf{W}\,\boldsymbol{x}[k] - \mu[k]\boldsymbol{c}[k]. \tag{65}$$

Now, since $\mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}\mathbf{W} = \mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}\mathbf{W}\mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}$ if (C1) and (C2) hold, eq. (65) becomes:

$$\begin{aligned}\boldsymbol{x}_\perp[k+1] \;&=\; \mathcal{P}_{\mathcal{R}(\mathbf{U})^\perp}\mathbf{W}\,\boldsymbol{x}_\perp[k] - \mu[k]\boldsymbol{c}[k]\\ &=\; (\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})\boldsymbol{x}_\perp[k] - \mu[k]\boldsymbol{c}[k].\end{aligned} \tag{66}$$

Iterating recursion (66), we have:

$$\begin{aligned}\boldsymbol{x}_\perp[k] \;&=\; (\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})^k\boldsymbol{x}_\perp[0]\\ &\quad - \sum_{l=1}^{k}(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})^{k-l}\mu[l-1]\boldsymbol{c}[l-1].\end{aligned} \tag{67}$$

Now, taking the norm of (67) and using (A2), (C3), we obtain:

$$\|\boldsymbol{x}_\perp[k]\| \;\leq\; \beta^k\|\boldsymbol{x}_\perp[0]\| + G\sum_{l=1}^{k}\beta^{k-l}\mu[l-1]. \tag{68}$$

Then, if (B1) holds, taking the limit of (68), since $\beta^k \to 0$ as $k \to \infty$ and $\lim_{k\to\infty}\sum_{l=1}^{k}\beta^{k-l} = \dfrac{1}{1-\beta}$, we have:

$$\lim_{k\to\infty}\|\boldsymbol{x}_\perp[k]\| \;\leq\; \frac{G\mu}{1-\beta} = O(\mu), \tag{69}$$

which proves (33) [cf. (64)]. On the other side, if (B2) holds, invoking Lemma 6(a) [cf. (62)], from (68) we conclude that:

$$\lim_{k\to\infty}\|\boldsymbol{x}_\perp[k]\| \;=\; 0, \tag{70}$$

thus proving also (34), and completing the proof of point (a).

Point (b): We start analyzing the temporal evolution of $\overline{\boldsymbol{x}}[k]$, i.e., the component of $\boldsymbol{x}[k]$ that lies in $\mathcal{R}(\mathbf{U})$. Thus, multiplying (29) from the left side by $\mathcal{P}_{\mathcal{R}(\mathbf{U})}$, and using (C2), we obtain:

$$\overline{\boldsymbol{x}}[k+1] = \overline{\boldsymbol{x}}[k] - \mu[k]\mathcal{P}_{\mathcal{R}(\mathbf{U})}\partial f(\boldsymbol{x}[k]). \tag{71}$$

Now, exploiting recursion (71), the Euclidean distance of the sequence $\overline{\boldsymbol{x}}[k+1]$ to the optimal solution set $\mathcal{S}$ reads as:

$$\begin{aligned}\|\overline{\boldsymbol{x}}[k+1] - \boldsymbol{x}^*\|^2 &= \|\overline{\boldsymbol{x}}[k] - \boldsymbol{x}^*\|^2 + \mu[k]^2\|\mathcal{P}_{\mathcal{R}(\mathbf{U})}\partial f(\boldsymbol{x}[k])\|^2\\ &\quad - 2\mu[k]\partial f(\boldsymbol{x}[k])^T(\overline{\boldsymbol{x}}[k] - \boldsymbol{x}^*),\end{aligned} \tag{72}$$

where we exploited the fact that both $\overline{\boldsymbol{x}}[k]$ and $\boldsymbol{x}^*$ lie in the subspace $\mathcal{R}(\mathbf{U})$. Now, summing and subtracting properly the vector $\boldsymbol{x}[k]$, we derive a bound on the third term on the RHS of (72). In particular, we have:

$$\begin{aligned}&-\partial f(\boldsymbol{x}[k])^T(\overline{\boldsymbol{x}}[k] - \boldsymbol{x}^* + \boldsymbol{x}[k] - \boldsymbol{x}[k])\\ &\quad= -\partial f(\boldsymbol{x}[k])^T(\boldsymbol{x}[k] - \boldsymbol{x}^*) + \partial f(\boldsymbol{x}[k])^T(\boldsymbol{x}[k] - \overline{\boldsymbol{x}}[k])\\ &\quad\overset{(a)}{\leq} -(f(\boldsymbol{x}[k]) - f^*) + G\|\boldsymbol{x}[k] - \overline{\boldsymbol{x}}[k]\|\\ &\quad\overset{(b)}{\leq} -(f(\boldsymbol{x}[k]) - f^*) + \beta^k G\|\boldsymbol{x}_\perp[0]\|\\ &\qquad + G^2\sum_{l=1}^{k}\beta^{k-l}\mu[l-1]\end{aligned} \tag{73}$$

where (a) follows from the convexity of f [cf. (A1-1)] and (A2); and in (b) we used (68) [cf. (64)]. Then, exploiting (73) in (72), and using (A2), we obtain:

$$\begin{aligned}\|\overline{\boldsymbol{x}}[k+1] - \boldsymbol{x}^*\|^2 \;&\leq\; \|\overline{\boldsymbol{x}}[k] - \boldsymbol{x}^*\|^2\\ &\quad - 2\mu[k](f(\boldsymbol{x}[k]) - f^*) + 2G^2\mu[k]\sum_{l=1}^{k}\beta^{k-l}\mu[l-1]\\ &\quad + 2G\|\boldsymbol{x}_\perp[0]\|\beta^k\mu[k] + G^2\mu[k]^2.\end{aligned} \tag{74}$$

Applying recursively inequality (74), we get:

$$\begin{aligned}&\|\overline{\boldsymbol{x}}[k+1] - \boldsymbol{x}^*\|^2 \;\leq\; \|\overline{\boldsymbol{x}}[0] - \boldsymbol{x}^*\|^2\\ &- 2\sum_{n=0}^{k}\mu[n](f(\boldsymbol{x}[n]) - f^*) + 2G^2\sum_{n=0}^{k}\sum_{l=1}^{n}\beta^{n-l}\mu[n]\mu[l-1]\\ &+ 2G\|\boldsymbol{x}_\perp[0]\|\sum_{n=0}^{k}\beta^n\mu[n] + G^2\sum_{n=0}^{k}\mu[n]^2.\end{aligned} \tag{75}$$

Now, since $\|\overline{\boldsymbol{x}}[k+1] - \boldsymbol{x}^*\|^2 \geq 0$ for all k, we have:

$$\begin{aligned}&2\sum_{n=0}^{k}\mu[n](f(\boldsymbol{x}[n]) - f^*)\\ &\qquad\leq\; \|\overline{\boldsymbol{x}}[0] - \boldsymbol{x}^*\|^2 + 2G^2\sum_{n=0}^{k}\sum_{l=1}^{n}\beta^{n-l}\mu[n]\mu[l-1]\\ &\qquad + 2G\|\boldsymbol{x}_\perp[0]\|\sum_{n=0}^{k}\beta^n\mu[n] + G^2\sum_{n=0}^{k}\mu[n]^2.\end{aligned} \tag{76}$$

Finally, exploiting

$$\sum_{n=0}^{k}\mu[n](f(\boldsymbol{x}[n]) - f^*) \geq \left(\sum_{n=0}^{k}\mu[n]\right)(f^{best}[k] - f^*),$$

in (76), we obtain:

$$\begin{aligned}f^{best}[k] - f^* \leq \Bigg(&\|\overline{\boldsymbol{x}}[0] - \boldsymbol{x}^*\|^2 + 2G^2\sum_{n=0}^{k}\sum_{l=1}^{n}\beta^{n-l}\mu[n]\mu[l-1]\\ &+ 2G\|\boldsymbol{x}_\perp[0]\|\sum_{n=0}^{k}\beta^n\mu[n] + G^2\sum_{n=0}^{k}\mu[n]^2\Bigg)\frac{1}{\left(2\sum_{n=0}^{k}\mu[n]\right)}.\end{aligned} \tag{77}$$

Then, if (B1) holds, exploiting $\sum_{n=0}^{k}\beta^n \leq 1/(1-\beta)$ for all k, from (77) we obtain:

$$\begin{aligned}f^{best}[k] - f^* \leq \Bigg(&\|\overline{\boldsymbol{x}}[0] - \boldsymbol{x}^*\|^2 + \frac{2G^2\mu^2}{1-\beta}(k+1)\\ &+ \frac{2G\|\boldsymbol{x}_\perp[0]\|\mu}{1-\beta} + G^2(k+1)\mu^2\Bigg)\frac{1}{2(k+1)\mu},\end{aligned} \tag{78}$$

which proves (35). Thus, from (78), it is immediate to see that

$$\lim_{k\to\infty} f^{best}[k] - f^* \leq \frac{G^2(3-\beta)}{2(1-\beta)}\mu \;=\; O(\mu), \tag{79}$$

with convergence rate $O\left(\frac{1}{k+1}\right)$, which proves (36). On the other side, under (B2), taking the limit as $k \to \infty$, it is immediate to check that the numerator of (77) is bounded [cf. (63), (B2)], while the denominator tends to infinity [cf.

(B2)]. Since by definition $f^{best}[k] \geq f^*$, and the argument above proves that $\lim_{k\to\infty} f^{best}[k] \leq f^*$, it must be that under (B2) we have (37).

Finally, to prove (38), we bound (77) as:

$$f^{best}[k] - f^* \leq \left(C_1 + \frac{G^2(3-\beta)}{2(1-\beta)}\sum_{n=0}^{k}\mu[n]^2\right)\frac{1}{\left(\sum_{n=0}^{k}\mu[n]\right)} \quad (80)$$

that follows from setting

$$\frac{1}{2}\|\overline{\boldsymbol{x}}[0]-\boldsymbol{x}^*\|^2 + \frac{G\|\boldsymbol{x}_\perp[0]\|\mu[0]}{1-\beta} \leq C_1 < \infty,$$

$$\sum_{n=0}^{k}\sum_{l=1}^{n}\beta^{n-l}\mu[n]\mu[l-1] \leq \frac{1}{1-\beta}\sum_{n=0}^{k}\mu[n]^2, \quad (81)$$

which exploit (62), and the inequality $\mu[n]\mu[l-1] \leq (\mu[n]^2 + \mu[l-1]^2)/2$. Now, if we set $\mu[k] = \frac{1}{\sqrt{k+1}}$, exploiting in (80) the inequalities

$$\sum_{n=0}^{k}\mu[n] = \sum_{n=0}^{k}\frac{1}{\sqrt{n+1}} \geq \int_0^{k+1}\frac{du}{\sqrt{u+1}} \geq \sqrt{k+1} \quad (82)$$

$$\sum_{n=0}^{k}\mu^2[n] = \sum_{s=1}^{k+1}\frac{1}{s} \leq 1 + \int_1^{k+1}\frac{du}{u} = 1+\log(k+1) \quad (83)$$

we obtain:

$$f^{best}[k] - f^* \leq \frac{C_2 + \frac{G^2(3-\beta)}{2(1-\beta)}\log(k+1)}{\sqrt{k+1}}, \quad (84)$$

with $C_2 = C_1 + \frac{G^2(3-\beta)}{2(1-\beta)} < \infty$, which proves (38) and establishes the convergence rate given by $\mathcal{O}\left(\frac{\log(k+1)}{\sqrt{k+1}}\right)$. This completes the proof of point (b) and Theorem 4.

Appendix E
Proof of Theorem 5

First, we prove (41). Under (A1-2), invoking the descent lemma on J in (39), and using (29) with fixed step-size sequence $\mu[k] = \mu$ for all k[4], we get:

$$J(\boldsymbol{x}[k+1]) \leq J(\boldsymbol{x}[k]) - \left(\mu - \frac{L_J}{2}\mu^2\right)\|\partial J(\boldsymbol{x}[k])\|^2 \quad (85)$$

where $L_J = L + \mu^{-1}(1-\lambda_N(\mathbf{W}))$ denotes the Lipshitz constant of ∂J. Let

$$\eta = \mu\left(1 - \frac{L_J}{2}\mu\right) = \frac{\mu}{2}(1+\lambda_N(\mathbf{W}) - \mu L), \quad (86)$$

which is greater than zero if $0 < \mu < \frac{1+\lambda_N(\mathbf{W})}{L}$. Now, applying recursively (85), since under (A3) J is bounded from below (w.l.o.g. $J(\boldsymbol{x}[k+1]) \geq 0$ for all k), we obtain:

$$0 \leq J(\boldsymbol{x}[k+1]) \;\leq\; J(\boldsymbol{x}[0]) - \eta\sum_{l=0}^{k}\|\partial J(\boldsymbol{x}[l])\|^2. \quad (87)$$

[4]Note that (29) is a step of gradient descent applied to function J.

Since (87) is true for all k, we get:

$$\lim_{k\to\infty}\sum_{l=0}^{k}\|\partial J(\boldsymbol{x}[l])\|^2 \leq \frac{1}{\eta}J(\boldsymbol{x}[0]) < \infty, \quad (88)$$

which implies that $\lim_{k\to\infty}\partial J(\boldsymbol{x}[k]) = \mathbf{0}$, thus proving (41).

We now prove (42) and (43). Under (A1-2), invoking the descent lemma on f and using (71), we get:

$$f(\overline{\boldsymbol{x}}[k+1]) \;\leq\; f(\overline{\boldsymbol{x}}[k]) - \mu[k]\partial f(\overline{\boldsymbol{x}}[k])^T\mathcal{P}_{\mathcal{R}(\mathbf{U})}\partial f(\boldsymbol{x}[k]) + \frac{L}{2}\mu[k]^2\|\mathcal{P}_{\mathcal{R}(\mathbf{U})}\partial f(\boldsymbol{x}[k])\|^2. \quad (89)$$

Now, summing and subtracting the vector $\partial f(\overline{\boldsymbol{x}}[k])$ properly in the second term on the RHS of (89), we obtain:

$$\begin{aligned} f(\overline{\boldsymbol{x}}[k+1]) &\overset{(a)}{\leq} f(\overline{\boldsymbol{x}}[k]) - \mu[k]\|\partial f(\overline{\boldsymbol{x}}[k])\|^2_{\mathcal{P}_{\mathcal{R}(\mathbf{U})}} + \frac{L}{2}G^2\mu[k]^2 \\ &\quad - \mu[k]\partial f(\overline{\boldsymbol{x}}[k])^T\mathcal{P}_{\mathcal{R}(\mathbf{U})}(\partial f(\boldsymbol{x}[k]) - \partial f(\overline{\boldsymbol{x}}[k])) \\ &\overset{(b)}{\leq} f(\overline{\boldsymbol{x}}[k]) - \mu[k]g[k] + \frac{L}{2}G^2\mu[k]^2 + \mu[k]LG\|\boldsymbol{x}[k]-\overline{\boldsymbol{x}}[k]\| \\ &\overset{(c)}{\leq} f(\overline{\boldsymbol{x}}[k]) - \mu[k]g[k] + \frac{L}{2}G^2\mu[k]^2 + LG\|\boldsymbol{x}_\perp[0]\|\beta^k\mu[k] \\ &\quad + LG^2\mu[k]\sum_{l=1}^{k}\beta^{k-l}\mu[l-1] \end{aligned} \quad (90)$$

where in (a) we used (A2); (b) comes from (A1-2), (A2), and (40); and in (c) we used (68) [cf. (64)]. Now, applying recursively (90), since under (A3) f is bounded from below (w.l.o.g., we consider $f(\overline{\boldsymbol{x}}[k+1]) \geq 0$ for all k), we obtain:

$$\begin{aligned} \sum_{n=0}^{k}\mu[n]g[n] \;&\leq\; f(\overline{\boldsymbol{x}}[0]) + LG\|\boldsymbol{x}_\perp[0]\|\sum_{n=0}^{k}\beta^n\mu[n] \\ &\quad + \frac{L}{2}G^2\sum_{n=0}^{k}\mu[n]^2 + LG^2\sum_{n=0}^{k}\sum_{l=1}^{n}\beta^{n-l}\mu[n]\mu[l-1]. \end{aligned} \quad (91)$$

Using $\sum_{n=0}^{k}\mu[n]g[n] \geq \left(\sum_{n=0}^{k}\mu[n]\right)g^{best}[k]$ in (91), we get:

$$\begin{aligned} g^{best}[k] \leq \Bigg(&f(\overline{\boldsymbol{x}}[0]) + LG\|\boldsymbol{x}_\perp[0]\|\sum_{n=0}^{k}\beta^n\mu[n] + \frac{L}{2}G^2\sum_{n=0}^{k}\mu[n]^2 \\ &+ LG^2\sum_{n=0}^{k}\sum_{l=1}^{n}\beta^{n-l}\mu[n]\mu[l-1]\Bigg)\frac{1}{\left(\sum_{n=0}^{k}\mu[n]\right)}. \end{aligned} \quad (92)$$

Then, if (B1) holds, exploiting $\sum_{n=0}^{k}\beta^n \leq 1/(1-\beta)$ for all k, from (92) we obtain:

$$\begin{aligned} g^{best}[k] \leq \Bigg(&f(\overline{\boldsymbol{x}}[0]) + \frac{LG\|\boldsymbol{x}_\perp[0]\|\mu}{1-\beta} + \frac{L}{2}(k+1)G^2\mu^2 \\ &+ \frac{LG^2\mu^2}{1-\beta}(k+1)\Bigg)\frac{1}{(k+1)\mu}, \end{aligned} \quad (93)$$

which proves (42). Thus, from (93), it is easy to see that

$$\lim_{k\to\infty} g^{best}[k] \leq \frac{LG^2(3-\beta)}{2(1-\beta)}\mu = O(\mu), \quad (94)$$

with convergence rate $O\left(\frac{1}{k+1}\right)$, thus proving also (43). To prove (44), we bound (93) as:

$$g^{best}[k] \leq \left(C_3 + \frac{LG^2(3-\beta)}{2(1-\beta)}\sum_{n=0}^{k}\mu[n]^2\right)\frac{1}{\left(\sum_{n=0}^{k}\mu[n]\right)} \quad (95)$$

where we exploited (81), and set

$$f(\overline{\boldsymbol{x}}[0]) + \frac{LG\|\boldsymbol{x}_\perp[0]\|\mu[0]}{1-\beta} \leq C_3 < \infty.$$

Now, if $\mu[k] = \frac{1}{\sqrt{k+1}}$, exploiting in (95) the inequalities (82) and (83), we obtain:

$$g^{best}[k] \leq \frac{C_4 + \frac{LG^2(3-\beta)}{2(1-\beta)}\log(k+1)}{\sqrt{k+1}}, \quad (96)$$

with $C_4 = C_3 + \frac{LG^2(3-\beta)}{2(1-\beta)} < \infty$, which proves (44) and establishes the convergence rate given by $\mathcal{O}\left(\frac{\log(k+1)}{\sqrt{k+1}}\right)$. Finally, to prove (45), we apply Lemma 7 to the recursive inequality (90) using the following identifications:

$$Y[k] = f(\overline{\boldsymbol{x}}[k]); \qquad X[k] = \mu[k]g[k]; \quad (97)$$

$$Z[k] = \frac{L}{2}G^2\mu[k]^2 + LG\|\boldsymbol{x}_\perp[0]\|\beta^k\mu[k] + LG^2\mu[k]\sum_{l=1}^{k}\beta^{k-l}\mu[l-1]. \quad (98)$$

It is straightforward to check that $X[k] \geq 0$ for all k [cf. (B2), (40)], and that $\sum_{k=1}^{\infty} Z[k] < \infty$ [cf. (B2), (C3), $\|\boldsymbol{x}_\perp[0]\| < \infty$, and (63)]. Since $f(\overline{\boldsymbol{x}}[k])$ is coercive [cf. (A3)], it follows from Lemma 7 that $f(\overline{\boldsymbol{x}}[k])$ converges to a finite value, and $\sum_{k=1}^{\infty}\mu[k]g[k] < \infty$, which from (40), using (B2), leads to

$$\lim_{k\to\infty}\inf_{k}\ \|\partial f(\overline{\boldsymbol{x}}[k])\|_{\mathcal{P}_{\mathcal{R}(\mathbf{U})}} = 0. \quad (99)$$

Using similar arguments as in [60, p.1887], we can also prove:

$$\lim_{k\to\infty}\sup_{k}\|\partial f(\overline{\boldsymbol{x}}[k])\|_{\mathcal{P}_{\mathcal{R}(\mathbf{U})}} = 0. \quad (100)$$

In conclusion, from (99) and (100), we must have:

$$\lim_{k\to\infty}\|\partial f(\overline{\boldsymbol{x}}[k])\|_{\mathcal{P}_{\mathcal{R}(\mathbf{U})}} = 0, \quad (101)$$

i.e., the sequence $\{\overline{\boldsymbol{x}}[k]\}_k$ converges to a stationary point of (2) [cf. (31)]. This concludes the proof of Theorem 5.

REFERENCES

[1] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Proc. Mag.*, pp. 56–69, 2007.

[2] A. H. Sayed *et al.*, "Adaptation, learning, and optimization over networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[3] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, *Distributed Detection and Estimation in Wireless Sensor Networks*. Academic Press Library in Signal Processing, 2014, vol. 2, pp. 329–408.

[4] S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta, "Distributed data mining in peer-to-peer networks," *IEEE internet computing*, vol. 10, no. 4, pp. 18–26, 2006.

[5] A. Lazarevic and Z. Obradovic, "Boosting algorithms for parallel and distributed learning," *Distributed and Parallel Databases*, vol. 11, no. 2, pp. 203–229, 2002.

[6] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while Computing: Distributed Mobile Cloud Computing over 5G Heterogeneous Networks," *IEEE Sig. Proc. Mag.*, vol. 31, no. 6, 2014.

[7] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Trans. on Knowl. and Data Eng.*, vol. 26, no. 1, pp. 97–107, 2014.

[8] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb 2014.

[9] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of signal processing*. Cambridge University Press, 2014.

[10] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[11] S. M. Kay, "Fundamentals of statistical signal processing, volume i: estimation theory," 1993.

[12] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. on Signal Proc.*, vol. 61, pp. 1644–1656, 2013.

[13] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.

[14] D. L. Donoho and B. F. Logan, "Signal recovery and the large sieve," *SIAM Journal on Appl. Math.*, vol. 52, no. 2, pp. 577–591, 1992.

[15] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics:(statistical) learning tools for our era of data deluge," *IEEE Signal Proc. Mag.*, vol. 31, no. 5, pp. 18–31, 2014.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

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

[18] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[19] A. Nedić, A. Ozdaglar, and P. Parillo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[20] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[21] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[22] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, "A dual approach for optimal algorithms in distributed optimization over networks," *arXiv preprint arXiv:1809.00710*, 2018.

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

[24] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. on Aut. Control*, vol. 57, no. 3, pp. 592–606, March 2012.

[25] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato, "Newton-Raphson consensus for distributed convex optimization," in *IEEE Conf. on Dec. and Control*, Orlando, Dec. 2011, pp. 5917–5922.

[26] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, August 2012.

[27] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, pp. 1035–1048, March 2010.

[28] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. on Signal Processing*, vol. 61, no. 6, pp. 1419–1433, March 2013.

[29] S. Kar, J. M. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, 2012.

[30] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. of Machine Learning Research*, vol. 11, pp. 1663–1707, Jan. 2010.

[31] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," in *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, 2016, pp. 461–497.

[32] M. Zhu and S. Martínez, "An approximate dual subgradient algorithm for distributed non-convex constrained optimization," *IEEE Trans. on Automatic Control*, vol. 58, no. 6, pp. 1534–1539, June 2013.

[33] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Trans. on Automatic Control*, vol. 58, no. 2, pp. 391–405, Feb. 2013.

[34] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[35] M. Hong, D. Hajinezhad, and M.-M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *Proc. of ICML*, 2017, pp. 1529–1538.

[36] J. Zeng and W. Yin, "On nonconvex decentralized gradient descent," *IEEE Trans. on Signal Processing*, vol. 66, no. 11, pp. 2834–2848, 2018.

[37] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *preprint arXiv:1809.01106*, 2018.

[38] A. Daneshmand, G. Scutari, and V. Kungurtsev, "Second-order guarantees of distributed gradient algorithms," *arXiv preprint arXiv:1809.08694*, 2018.

[39] B. Swenson, S. Kar, H. V. Poor, and J. Moura, "Annealing for distributed global optimization," *arXiv preprint arXiv:1903.07258*, 2019.

[40] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments–part ii: Polynomial escape from saddle-points," *arXiv preprint arXiv:1907.01849*, 2019.

[41] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti, "Distributed signal recovery based on in-network subspace projections," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5242–5246.

[42] R. Nassif, S. Vlaski, and A. H. Sayed, "Distributed inference over networks under subspace constraints," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2019, pp. 5232–5236.

[43] ——, "Adaptation and learning over networks under subspace constraints - part i: Stability analysis," *arXiv preprint arXiv:1905.08750*, 2019.

[44] ——, "Adaptation and learning over networks under subspace constraints - part ii: Performance analysis," *arXiv preprint arXiv:1906.12250*, 2019.

[45] S. Barbarossa, G. Scutari, and T. Battisti, "Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, 2009, pp. 2893–2896.

[46] ——, "Cooperative sensing for cognitive radio using decentralized projection algorithms," in *2009 IEEE 10th Workshop on Signal Processing Advances in Wireless Communications*. IEEE, 2009, pp. 116–120.

[47] "Wireless insite prediction software," *https://www.remcom.com/wireless-insite-em-propagation-software/*.

[48] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4117–4131, 2017.

[49] M. Coutino, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *arXiv preprint arXiv:1808.03004*, 2017.

[50] T. Weerasinghe, D. Romero, C. Asensio-Marco, and B. Beferull-Lozano, "Fast distributed subspace projection via graph filters," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, 2018, pp. 4639–4643.

[51] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[52] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[53] D. S. Bernstein, *Matrix mathematics: Theory, facts, and formulas with application to linear systems theory*. Princeton University Press Princeton, 2005, vol. 41.

[54] G. Di Pillo and L. Grippo, "Exact penalty functions in constrained optimization," *SIAM Journal on control and optimization*, vol. 27, no. 6, pp. 1333–1360, 1989.

[55] A. H. Sayed, *Adaptive filters*. John Wiley & Sons, 2011.

[56] D. L. Donoho and P. Stark, "Uncertainty principles and signal recovery," *SIAM Jour. on Appl. Math.*, vol. 49, no. 3, pp. 906–931, 1989.

[57] P. Di Lorenzo, P. Banelli, S. Barbarossa, and S. Sardellitti, "Distributed adaptive learning of graph signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4193–4208, 2017.

[58] "Abilene dataset," *http://math.bu.edu/people/kolaczyk/datasets.html*.

[59] D. P. Bertsekas and J. N. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM Journal on Optimization*, vol. 10, no. 3, pp. 627–642, 2000.

[60] A. Daneshmand, F. Facchinei, V. Kungurtsev, and G. Scutari, "Hybrid random/deterministic parallel algorithms for nonconvex big data optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3914–3929, Aug. 2015.

Paolo Di Lorenzo (S'10-M'13-SM'19) received the M.Sc. and the Ph.D. degrees in Electrical Engineering from Sapienza University of Rome, Rome, Italy, in 2008 and 2012, respectively. In 2010, he held a visiting research appointment with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA, USA. From May 2015 to February 2018, he was an Assistant Professor with the Department of Engineering, University of Perugia, Perugia, Italy. He is currently an Assistant Professor with the Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome. He has participated in the FP7 European research projects FREEDOM, on femtocell networks; SIMTISYS, on moving target detection and imaging using a constellation of satellites; and TROPIC, on communication, computation, and storage over collaborative femtocells. His research interests include signal processing theory and methods, distributed optimization, mobile edge computing, machine learning, and graph signal processing. He is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. He was the recipient of the three best student paper awards, respectively, at IEEE SPAWC10, EURASIP EUSIPCO11, and IEEE CAMSAP11. He is also the recipient of the 2012 GTTI (Italian national group on telecommunications and information theory) award for the Best Ph.D. thesis in communication engineering.

Sergio Barbarossa (S'84-M'88-F'12) received his MS and Ph.D. EE degree from the Sapienza University of Rome, where he is now a Full Professor. He has held visiting positions at the Environmental Research Institute of Michigan ('88), Univ. of Virginia ('95, '97), and Univ. of Minnesota ('99). He is an IEEE Fellow, EURASIP Fellow, and he has been an IEEE Distinguished Lecturer. He received the 2000 and 2014 IEEE Best Paper Awards from the IEEE Signal Processing Society and the 2010 Technical Achievements Award from the EURASIP. He is the coauthor of papers that received the Best Student Paper Award at ICASSP 2006, SPAWC 2010, EUSIPCO 2011, and CAMSAP 2011. He has been the scientific coordinator of several EU projects on wireless sensor networks, small cell networks, and distributed mobile cloud computing. He is now the technical manager of the H2020 Europe/Japan project 5G-MiEdge. His current research interests are in the area of graph signal processing, distributed optimization, millimeter wave communications, mobile edge computing and machine learning. From 1997 to 2003, he was a member of the IEEE Technical Committee for Signal Processing in Communications. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1998-2000 and 2004-2006), the IEEE SIGNAL PROCESSING MAGAZINE, and the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. He has been the General Chairman of the IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2003 and the Technical Co-Chair of SPAWC, 2013. He has been the Guest Editor for Special Issues on the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, EURASIP Journal of Applied Signal Processing, EURASIP Journal on Wireless Communications and Networking, the IEEE SIGNAL PROCESSING MAGAZINE, and the IEEE SELECTED TOPICS ON SIGNAL PROCESSING.

Stefania Sardellitti (M'12) received the M.Sc. degree in Electronic Engineering from the University of Rome "La Sapienza," Italy, in 1998 and the Ph.D. degree in Electrical and Information Engineering from the University of Cassino, Italy, in 2005. From 2005 to 2019 she was an appointed professor of digital communications at the University of Cassino, Italy. Currently, she is assistant professor with the Department of Information Engineering, Electronics and Telecommunications, University of Rome, Sapienza, Italy. She received the 2014 IEEE Best Paper Award from the IEEE Signal Processing Society. She has participated in the European projects WINSOC, FREEDOM, TROPIC, and in the H2020 Europe/Japan project 5G-MiEdge. Her current research interests are in the area of graph signal processing, mobile edge computing, wireless sensor networks and distributed optimization.