Deterministic and Randomized Diffusion Based Iterative Generalized Hard Thresholding (DiFIGHT) for Distributed Recovery of Sparse Signals

Samrat Mukhopadhyay, *Member, IEEE*, and Mrityunjoy Chakraborty, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a distributed iterative hard thresholding algorithm, namely, DiFIGHT, for a network that uses diffusion as the means of intra-network collaboration. Subsequently, we present a modification of the proposed algorithm, namely, MoDiFIGHT, that has lesser communication complexity than DiFIGHT. We additionally propose four different strategies, namely, RP, RNP, RGP$_r$, and RGNP$_r$ that are used to randomly select a subset of nodes for taking part in DiFIGHT/MoDiFIGHT. This gives rise to further reduction in the mean number of communications during the run of the proposed distributed algorithms. We present theoretical estimates of the long run communication per unit time, both for DiFIGHT and MoDiFIGHT, with and without random selection of nodes. Also, we present theoretical analysis of the two proposed algorithms and provide provable bounds on their recovery performance with or without using the random node selection strategies. Finally we use numerical studies to show that both with and without random selections, the proposed algorithms exhibit performances far superior to the consensus based distributed IHT algorithm.**

***Index Terms*—Distributed estimation, diffusion network, iterative hard thresholding (IHT).**

I. Introduction

We consider a distributed sparse optimization problem, where there is a network of nodes, with each node $v \in \{1, 2, \ldots, L\}$ individually solving the following problem:

$$\min_{\boldsymbol{z}\in\mathbb{R}^n} f_v(\boldsymbol{z}) \text{ s.t. } \|\boldsymbol{z}\|_0 \leq K.$$

Here the functions f_v, $v = 1, 2, \ldots, L$ are cost functions which are generally chosen to satisfy some kind of restricted convexity assumptions, i.e., they are generally designed so that their curvatures have some specific properties. For example, in the distributed compressed sensing setting, a node v measures a K-sparse vector $\boldsymbol{x} \in \mathbb{R}^n$, and stores a m dimensional ($m < n$) measurement as $\boldsymbol{y}_v = \boldsymbol{\Phi}_v\boldsymbol{x} + \boldsymbol{e}_v$, where $\boldsymbol{e}_v$ is measurement noise and $\boldsymbol{\Phi}_v$ is a so-called $m \times n$ measurement matrix. A suitable cost function in this case is $f_v(\boldsymbol{z}) = \|\boldsymbol{y}_v - \boldsymbol{\Phi}_v\boldsymbol{z}\|_2^2$, and to impose conditions on its curvature, the matrix $\boldsymbol{\Phi}_v$ is assumed to satisfy some kind of restricted isometry property [1]. However, in a collaborative, or distributed setting, the nodes do not work alone and sparse recovery algorithms working at neighboring nodes update their estimates by exchanging information among themselves during the run of the algorithm. This information exchange through collaboration helps the true estimate to emerge, often in a faster or in other more advantageous ways compared to non-cooperative setting. The literature on distributed recovery is relatively recent, with some of the major developments being distributed basis pursuit [2], [3], distributed subgradient methods (DSM) [4], [5], distributed alternating direction method of multipliers (ADMM) [6], and distributed iterative soft thresholding algorithm (DISTA) [7]–[9]. All these algorithms, in one form or other, use the *consensus* optimization paradigm, where the nodes in a neighborhood cooperatively minimize a global cost function while minimizing their individual local cost functions. However, in the literature of adaptive networks, there is a different family of algorithms, namely, *diffusion*, which are studied extensively by Sayed *et al.* [10]–[15], and are shown to exhibit superior performances compared to consensus strategies [10], and also to all noncooperative strategies. These strategies can be traced back to the generalized distributed communication and processing based model for distributed computation, proposed by Tsitsiklis [16]. It is only recently that distributed sparse recovery algorithms have been designed to incorporate diffusion as the underlying mechanism. In [17], the authors proposed sparsity promoting diffusion LMS with l_0 and l_1 regularizers, enjoying lesser mean square deviation over the network. In [18], the authors considered diffusion with proximal projection that employ general non-differentiable regularizers with only bounded subgradients. In [19], [20], the sparsity promoting regularization problem in diffusion networks was considered in a multi-tasking context. In [5], Patterson *et al.*, in addition to a consensus based distributed hard thresholding (CB-DIHT) method, also designed

Manuscript received March 8, 2021; revised July 26, 2021 and October 9, 2021; accepted October 22, 2021. Date of publication November 4, 2021; date of current version January 12, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Paolo Di Lorenzo. *(Corresponding author: Mrityunjoy Chakraborty.)*

Samrat Mukhopadhyay is with the Department of Electronics Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad 826004, India (e-mail: samrat@iitism.ac.in).

Mrityunjoy Chakraborty is with the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721302, India (e-mail: mrityun@ece.iitkgp.ac.in).

Digital Object Identifier 10.1109/TSIPN.2021.3124362

a distributed hard thresholding (DIHT) algorithm which is reminiscent of the diffusion mechanism, where one parent node forms a spanning tree and collects estimates of the gradients of the functions from all the nodes in the network over several time steps. More recently, another distributed hard thresholding algorithm DiHaT was proposed and analyzed by Chouvardas *et al.* [21]. Also Zaki *et al.* [22] proposed and analyzed a greedy distributed algorithm called the network gradient pursuit (NGP). Further, Zaki *et al.* [23] analyzed the distributed hard thresholding pursuit (DHTP) algorithm, originally proposed by Chouvardas [21]. All these algorithms have diffusion as the underlying mechanism.

The above algorithms, however, do not consider the problem of insufficient computational, memory and bandwidth resources, even though networks are often required to operate under resource constrained environments. These problems were initially addressed for the general resource constrained optimization framework in [24], [25], [26], [27]. In the context of sparse distributed recovery, the above issues have been considered recently in [28] by suitably modifying consensus IHT algorithm. It is the goal of this paper to propose and analyze a distributed IHT algorithm that minimizes general convex functions by using *diffusion* as its underlying mechanism, and then to modify it to obtain algorithms where only a few nodes are selected per time step, resulting in significantly reduced communication complexity. Specifically:

- We propose a distributed IHT algorithm termed DiFIGHT that uses diffusion mechanism to minimize general convex functions available to individual nodes.
- We also propose a simple low complexity modification of the DiFIGHT algorithm, termed MoDiFIGHT that, unlike diffusion based methods, exchanges only a subset of the estimates and thus uses less communication bandwidth.
- We propose four strategies that are used to randomly select and activate only a subset of nodes at each time step, thus reducing communication overhead, and also give theoretical estimates on the long run communication overhead per unit time required by both DiFIGHT and MoDiFIGHT when these strategies are used.
- We analyze the algorithms with and without using the random node selection strategies and derive convergence conditions. In particular, we show that if all the nodes have a common K-sparse stationary point, say, $\boldsymbol{x}^\star$, then under the convergence conditions, the node iterates converge to $\boldsymbol{x}^\star$ exactly, whereas, if the nodes have different but close-by stationary points, node iterates in the steady state remain confined to a neighborhood of the stationary points.
- We numerically evaluate the performance of these different algorithms with and without random node selection strategies and establish superiority of diffusion mechanism over its consensus counterparts.

II. Notation

The following notations have been used throughout the paper: the number of nodes in the network is denoted by L and the dimension of the unknown vector associated with any of these nodes is denoted by n. The symbol 't' in superscript indicates transposition of matrices / vectors, $\mathcal{H}$ denotes the set of all the set of indices $\{1, 2, \ldots, n\}$ and $\mathbf{1}$ denotes a $L \times 1$ vector of 1's. For any $S \subseteq \mathcal{H}$, $\boldsymbol{x}_S$ denotes the vector $\boldsymbol{x}$ restricted to S, i.e., $\boldsymbol{x}_S$ consists of those entries of $\boldsymbol{x}$ that have indices belonging to S. The operator $H_K(\cdot)$ returns the *K-best approximation* of a vector, i.e., for any vector $\boldsymbol{x} \in \mathbb{R}^n, H_K(\boldsymbol{x}) = \arg\min_{\boldsymbol{z}\in\mathbb{R}^n:\|z\|_0\leq K} \|\boldsymbol{z} - \boldsymbol{x}\|_2$. Given a function $f(x_1, x_2, \ldots, x_n)$, the gradient vector ∇f is given by $\nabla f = [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}]^t$, and its Hessian matrix is given by Hess_f with $[\mathrm{Hess}_f]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$, $1 \leq i, j \leq n$. Furthermore, we denote by $\nabla_S f$, the restricted vector $(\nabla f)_S$, for any $S \subseteq \mathcal{H}$. Also, for any integer K $(1 \leq K \leq n)$, we denote by $\nabla^K f$, the vector ∇f restricted to the subset corresponding to its K magnitude-wise largest coordinates. For any matrix $\boldsymbol{A}$, we use $\|\boldsymbol{A}\|$ to denote the l_2 operator norm, defined as $\|\boldsymbol{A}\| = \max_{\boldsymbol{x}\neq\mathbf{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}$. For any real symmetric matrix $\boldsymbol{A}$, we denote by $\lambda_{\max}(\boldsymbol{A})$ and $\lambda_{\min}(\boldsymbol{A})$ to denote the maximum and minimum eigenvalues of $\boldsymbol{A}$ respectively and in this case, one can show that $\|\boldsymbol{A}\| = \max_{\boldsymbol{x}\neq\mathbf{0}} \frac{|\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x}|}{\boldsymbol{x}^t\boldsymbol{x}} = \max\{|\lambda_{\max}(\boldsymbol{A})|, |\lambda_{\min}(\boldsymbol{A})|\}$. The symmetric difference Δ, between two sets A, B, is defined as $A\Delta B := (A \setminus B) \cup (B \setminus A)$. For any two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^L$, the inequality $\boldsymbol{a} \preccurlyeq \boldsymbol{b}$ is used to denote that $a_u \leq b_u$, $\forall u = 1, 2, \ldots, L$. Finally, we use the symbol $a.s.$ to denote almost sure convergence.

TABLE I
Algorithm: DiFIGHT and MoDiFIGHT

Input: Number of nodes L, the combining matrix $\boldsymbol{A}$ such that $\boldsymbol{A}^t\mathbf{1} = \mathbf{1}$, sparsity level K; Initial estimates $\boldsymbol{x}_v^0$, $1 \leq v \leq L$; step sizes $\mu_v > 0$, $v = 1, 2, \cdots, L$; maximum number of iterations k_{it};

While $(k < k_{\mathrm{it}})$

For $v = 1$ to L

$$\boldsymbol{\psi}_v^{k+1} = \boldsymbol{x}_v^k - \mu_v \nabla f_v(\boldsymbol{x}_v^k)$$

$$\hat{\boldsymbol{\psi}}_v^{k+1} = \begin{cases} \boldsymbol{\psi}_v^{k+1}, & \text{DiFIGHT} \\ H_K\left(\boldsymbol{\psi}_v^{k+1}\right), & \text{MoDiFIGHT} \end{cases}$$

End For

For $v = 1$ to L

$$\boldsymbol{x}_v^{k+1} = H_K\left(\sum_{u=1}^L a_{uv}\hat{\boldsymbol{\psi}}_u^{k+1}\right)$$

End For

$k = k + 1$

End While

III. Diffusion Based Hard Thresholding

A. Deterministic Strategies

The DiFfusion based Iterative Generalized Hard Thresholding (DiFIGHT) and its low complexity modification, the Modified DiFfusion based Iterative Generalized Hard Thresholding (MoDiFIGHT) algorithm are described in Table I. There are L nodes in the network. The combining matrix $\boldsymbol{A} \in \mathbb{R}^{L\times L}$ specifies the connectivity between the different nodes in the

network. The $(u,v)^{\text{th}}$ entry of $\boldsymbol{A}$, denoted by $a_{uv} \in [0,1]$, is the weight of the outgoing edge from node u to node v. Furthermore, the matrix $\boldsymbol{A}$ is assumed to be left stochastic, i.e., $\boldsymbol{A}^t\mathbf{1} = \mathbf{1}$. For any node v in the graph represented by the combination matrix $\boldsymbol{A}$, the *neighborhood* of v is denoted by $\mathcal{N}_v$, defined as $\mathcal{N}_v = \{u \in \{1,\ldots,L\} : a_{uv} > 0\}$. The nodes belonging to $\mathcal{N}_v$ are called the neighbors of the node v. We assume that if u is a neighbor of v, v is also a neighbor of u, so that $a_{uv} > 0 \Leftrightarrow a_{vu} > 0$. Further, we assume that for each node v, $a_{vv} > 0$, i.e., each node is a neighbor to itself. Lastly, it is assumed that each node v has the function $f_v(\cdot)$ available with it.

B. Randomized Strategies

We also propose IHT based diffusion algorithms where all the nodes need not participate in the diffusion process at each time step. This absence of participation results in significant reduction in the amount of communication between the neighboring nodes of the network, that would otherwise be required while exchanging values of estimates and gradient vectors. Inspired by Ravazzi *et al.* [8], we propose four different strategies for selecting the participating nodes, as given below. Here, at each step of iteration, a subset of nodes G is chosen randomly and while the gradient based updation of the estimates is carried out at all the nodes belonging to G and their neighbors, the diffusion process is restricted only to the nodes belonging to G from the neighboring nodes.

1) *Random Persistence (RP):* In this strategy, at a time step k, only one node is selected at random according to a probability distribution $\{p_1 \ldots, p_L\}$ over the nodes in the network. The probability distribution satisfies $p_v > 0$ for each node v in the network, and $\sum_{v=1}^{L} p_v = 1$, implying that each node has a positive probability of getting selected at a time step. Thus the selected group is $G = \{v\}$.
2) *Random Neighborhood Persistence (RNP):* As in the RP strategy, in this strategy too, at a time step k, a node v is selected with probability p_v, where the probability distribution satisfies the same conditions as in the RP strategy. However, unlike the RP strategy, the neighborhood $\mathcal{N}_v$ of the selected node v is also selected for participation in the diffusion process. Thus the selected group is $G = \{v\} \cup \mathcal{N}_v$.
3) *Random Group Persistence of order r (RGP$_r$):* In this strategy, a group G of r nodes is selected according to a probability distribution $\{p_G\}$ over all possible $\binom{L}{r}$ groups of nodes of size r. Here the probability distribution is chosen such that $p_G > 0$ for all such groups, and $\sum_{G\in\mathcal{G}_r} p_G = 1$, where $\mathcal{G}_r$ is the collection of all subsets of $\{1,2,\ldots,L\}$ of size r. Here the selected group of nodes is G.
4) *Random Group Neighborhood Persistence of order r (RGNP$_r$):* In this strategy, a group of nodes $\widetilde{G}$ is chosen with probability $p_{\widetilde{G}}$ and $\widetilde{G}$ as well the union of their neighborhoods is selected. The probability distribution $p_{\widetilde{G}}$ is assumed to satisfy the same conditions as in the RGP$_r$ strategy. The selected group is $G = \widetilde{G} \cup_{u\in\widetilde{G}} \mathcal{N}_u$.

TABLE II
ALGORITHM: RANDOMIZED DIFIGHT AND MODIFIGHT

Input: Number of nodes L, the combining matrix $\boldsymbol{A}$ such that $\boldsymbol{A}^t\mathbf{1} = \mathbf{1}$, sparsity level K; Initial estimates $\boldsymbol{x}_v^0,\ 1 \le v \le L$; step sizes $\mu_v > 0,\ v = 1,2,\cdots,\ L$; maximum number of iterations k_{it};

While $(k < k_{\text{it}})$
  For $v = 1 : L$
  if $v \in G$ or $\mathcal{N}_v \cap G \neq \emptyset$
    $\boldsymbol{\psi}_v^{k+1} = \boldsymbol{x}_v^k - \mu_v \nabla f_v(\boldsymbol{x}_v^k)$
    $\hat{\boldsymbol{\psi}}_v^{k+1} = \begin{cases} \boldsymbol{\psi}_v^{k+1}, & \text{DiFIGHT} \\ H_K\left(\boldsymbol{\psi}_v^{k+1}\right), & \text{MoDiFIGHT} \end{cases}$
  end if
  End For
  For $v \in G$
    $\boldsymbol{x}_v^{k+1} = H_K\left(\sum_{u\in\mathcal{N}_v} a_{uv}\hat{\boldsymbol{\psi}}_u^{k+1}\right)$
  End For
  $\boldsymbol{x}_u^{k+1} = \boldsymbol{x}_u^k\ \forall u \notin G$
  $k = k + 1$
End While

Once a group is selected, the diffusion process is applied to all the nodes in the group. The resulting algorithms are described in Table II.

C. Discussion on Communication Complexities

We present a comparative discussion on the *communication complexities* of the different strategies, which is defined as the sum of the time averages of the transmissions and receptions performed by all the nodes in the network for a specific strategy. Note that the communication complexity of a strategy is simply the sum of the time averages of transmissions and receptions of individual nodes. Furthermore, the communication complexity depends on both the diffusion mechanism as well as the strategy of selection of group of participating nodes. Therefore, it is enough to evaluate the time average of communications (transmission and receptions) for some fixed node v for the different group selection strategies. We denote the time averages of the number of messages transmitted and received by the node by $T(v) = \lim_{t\to\infty} \frac{\sum_{k=1}^{t} T_k(v)}{t}$ and $R_v = \lim_{t\to\infty} \frac{\sum_{k=1}^{t} R_k(v)}{t}$ respectively, where $T_k(v), R_k(v)$ are the number of messages transmitted and received respectively, by the node v at time step k. In all these strategies, for each node v, $\{T_k(v)\}_{k\ge0}(\{R_k(v)\}_{k\ge0})$ depends only on the nodes selected at time slot k and is therefore independent of $T_j(v)(resp.R_j(v)),\ j \neq k$. Clearly, these random variables are independent, identically distributed (i.i.d.), which are also

bounded as $T_k(v) \leq d_v \leq n$. Therefore, using the strong law of large numbers (SLLN), the limits $T(v)$ and $R(v)$ exist for all nodes v. To carry out the analysis, we denote by d_v the degree of the node v, which is the number of neighbors of node v, excluding itself.

Before proceeding to find expressions for $T(v),\ R(v)$ for the different algorithms, we point out that the communication complexities of DiFIGHT and MoDiFIGHT are intrinsically distinct because of the fact that for an update to occur, in DiFIGHT, each transmitting node transmits n values, whereas, in MoDiFIGHT each transmitting node has to perform $2\,K$ transmissions, K for the support indices and K for the values corresponding to those indices. Therefore, if $K << n/2$, the number of communications in MoDiFIGHT can be much smaller than that of DiFIGHT.

We first analyze the communication complexities of the deterministic diffusion algorithms. In this case, all the nodes of the network are chosen at every time step, so that $T_k(v) = T(v),\ R_k(v) = R(v), \forall k \geq 0$. Clearly, in DiFIGHT $T(v) = R(v) = nd_v$, while in MoDiFIGHT, $T(v) = R(v) = 2Kd_v$.

Next we carry out the analysis of $T(v),\ R(v)$ for the randomized algorithms where we introduce the variable C to denote n (for DiFIGHT) and $2\,K$ (for MoDiFIGT). First let us consider the calculation of $R_k(v)$. Observe that $R_k(v) = Cd_vI_k(v)$, where $I_k(v)$ is an indicator random variable taking value 1 if $v \in G_k$ where G_k is the group of nodes selected at time k by the randomized algorithm (in which case v is referred to as a *participating node* at time k) and is 0 otherwise. Clearly, for a fixed v, the sequence $\{I_k(v)\}_{k\geq 0}$ is a sequence of i.i.d. random variables. Therefore, by SLLN,

$$\frac{R(v)}{Cd_v} = \lim_{t\to\infty} \frac{\sum_{k=0}^{t-1} I_k(v)}{t} = \mathbb{E}\left[I_0(v)\right] = \pi_v \ a.s., \quad (1)$$

where π_v is the probability that the node v is participating (at any time $k \geq 0$) and is called the *participation probability*.

To calculate $T_k(v)$, observe that $T_k(v)$ is equal to C times the number of nodes in the neighborhood of v (distinct from v) participating at time k. Therefore,

$$T_k(v) = C \sum_{u\in\mathcal{N}_v\setminus\{v\}} I_k(u)$$

$$\Rightarrow \frac{T(v)}{C} = \lim_{t\to\infty} \frac{\sum_{k=0}^{t-1} T_k(v)}{C\,t} = \sum_{u\in\mathcal{N}_v\setminus\{v\}} \pi_u \ a.s. \quad (2)$$

Therefore, Eqs. (1) and (2) together describe the time average of communications performed by a node v in the network.

We now evaluate π_v for the different randomized strategies proposed. We assume in the following that the probability of selection of a node v is p_v and the probability of selection of a group of nodes G is p_G. Clearly, for uniformly random selections $p_v = 1/L,\ p_G = 1/\binom{L}{|G|}$.

1) For the RP strategy, only one node can be selected at a time. Therefore, $\pi_v = p_v$. For uniformly random selection, $\pi_v = 1/L$.
2) For the RNP strategy the node v participates if either it is selected (w.p. p_v) or one of its neighbor is selected (w.p. $\sum_{u\in\mathcal{N}_v\setminus v} p_u$). Hence $\pi_v = \sum_{u\in\mathcal{N}_v} p_u$. For uniformly random selection $\pi_v = \frac{d_v+1}{L}$.
3) For the RGP$_r$ strategy, the node v participates if it belongs to a group of r nodes G which contains v. Since only one such group is selected, we have $\pi_v = \sum_{G:|G|=r,\ v\in G} p_G$. For uniformly random selection $\pi_v = \frac{\binom{L-1}{r-1}}{\binom{L}{r}} = \frac{r}{L}$.
4) For the RGNP$_r$ strategy, the node v participates if a group $\widetilde{G}$ of size r is selected such that v belongs to the neighborhood of the nodes in $\widetilde{G}$. Therefore, v participates if

$$v \in \cup_{u\in\widetilde{G}}\mathcal{N}_u \Leftrightarrow \widetilde{G} \cap \mathcal{N}_v \neq \emptyset$$
$$\Rightarrow \pi_v = 1 - \sum_{\substack{G:|G|=r,\\ G\cap\mathcal{N}_v=\emptyset}} p_G \quad (3)$$

For uniformly random selection, we have $\pi_v = 1 - \frac{\binom{L-(d_v+1)}{r}}{\binom{L}{r}}$. This, of course, assumes that $L - (d_v + 1) \geq r$. Otherwise, the node v is always present in the neighborhood of some node or other of the group chosen and thus always participates, i.e. $\pi_v = 1$, which is an example of a highly connected node.

In the next section, we carry out a convergence analysis for DiFIGHT and MoDiFIGHT, both with and without the random node selection protocol, where, for the randomized node selection strategies, we make use of the above figures of participation probabilities to derive the necessary convergence conditions.

IV. Convergence Analysis of Deterministic and Randomized DiFIGHT and MoDiFIGHT

In this section, we analyze the convergence of the different diffusion based algorithms proposed in Section III. In order to do this we analyze how the distance between a $K-$sparse vector $\boldsymbol{x}^\star$ and the iterates produced by the diffusion algorithms changes with each iteration. For the sake of our analysis, a few assumptions on the functions $f_v,\ 1 \leq v \leq L$ are necessary and are described below.

A. Preliminaries and Assumptions

Definition 1 (Restricted Positive Definite Hessian): Suppose that f is a twice continuously differentiable function with Hessian $\mathrm{Hess}_f(\cdot)$. Then f is said to have a Restricted Positive Definite Hessian (RPDH) of order s with constants $(\alpha_s,\ \beta_s)$ (such that $\beta_s \geq \alpha_s > 0$), or $(\alpha_s,\ \beta_s)$-RPDH if the following holds:

$$\alpha_s \left\|\boldsymbol{x}\right\|_2^2 \leq \boldsymbol{x}^t \mathrm{Hess}_f(\boldsymbol{u})\boldsymbol{x} \leq \beta_s \left\|\boldsymbol{x}\right\|_2^2 \quad (4)$$

for all vectors $\boldsymbol{x},\ \boldsymbol{u} \in \mathbb{R}^n$ such that $\|\boldsymbol{u}\|_0 \leq s,\ \|\boldsymbol{x}\|_0 \leq s$. Here α_s and β_s are the largest and smallest such numbers, respectively, which satisfy the inequality (4).

This property is just a variant of the Stable Restricted Hessian (SRH) property defined in [29], which bounds the curvature of f, when restricted to the union of all subspaces of sparse vectors of a given sparsity. To see the implication of the RPDH property, observe that the Hessian $\mathrm{Hess}_f(\boldsymbol{u})$ is a Hermitian

matrix $\forall \boldsymbol{u}$, so that it admits the unique eigen-decomposition $\boldsymbol{Q}(\boldsymbol{u})^t \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}(\boldsymbol{u})$, where $\boldsymbol{Q}(\boldsymbol{u})$ is an orthogonal matrix and $\boldsymbol{D}(\boldsymbol{u})$ is a diagonal matrix. Fix any set $S \subset \{1, 2, \ldots, n\}$ such that $|S| \leq s$. Then, it can be easily verified that the RPDH property implies that,

$$\min_{S \subset \mathcal{H}: |S| \leq s} \lambda_{\min} \left(\boldsymbol{Q}_S(\boldsymbol{u})^t \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}_S(\boldsymbol{u}) \right) \geq \alpha_s, \tag{5}$$

$$\max_{S \subset \mathcal{H}: |S| \leq s} \lambda_{\max} \left(\boldsymbol{Q}_S(\boldsymbol{u})^t \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}_S(\boldsymbol{u}) \right) \leq \beta_s. \tag{6}$$

We use the RPDH property to prove the following lemma:

Lemma 1: Let $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ are vectors in $\mathbb{R}^n$ with supports T_1, T_2, T_3 respectively, and let $T = T_1 \cup T_2 \cup T_3$. Let ρ be an arbitrary positive number. Also, let $\boldsymbol{g}(\boldsymbol{y}, \boldsymbol{z}) := \boldsymbol{y} - \boldsymbol{z} - \rho(\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{z}))$. Then,

1)

$$\langle \boldsymbol{x}, \boldsymbol{g}(\boldsymbol{y}, \boldsymbol{z}) \rangle \leq \rho'_{|T|} \|\boldsymbol{x}\|_2 \|\boldsymbol{y} - \boldsymbol{z}\|_2, \tag{7}$$

and,

2)

$$\left\| (g(\boldsymbol{y}, \boldsymbol{z}))_{T_1} \right\|_2 \leq \rho'_{|T|} \|\boldsymbol{y} - \boldsymbol{z}\|_2, \tag{8}$$

where $\rho'_{|T|} = (|1 - \rho\delta^{(1)}_{|T|}| + \rho\delta^{(2)}_{|T|})$, f satisfies the RPDH-$(\alpha_{|T|}, \beta_{|T|})$ property, and $\delta^{(1)}_{|T|} = \frac{\beta_{|T|} + \alpha_{|T|}}{2}$, $\delta^{(2)}_{|T|} = \frac{\beta_{|T|} - \alpha_{|T|}}{2}$.

Proof: The key observation for the proof is the following:

$$\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{z}) = \int_0^1 \mathrm{Hess}_f(\boldsymbol{u})(\boldsymbol{z} - \boldsymbol{y}) d\tau,$$

where, $\boldsymbol{u} = \boldsymbol{y} + \tau(\boldsymbol{z} - \boldsymbol{y})$.

To prove 1) we use the eigen-decomposition of $\mathrm{Hess}_f(\boldsymbol{u}) = \boldsymbol{Q}^t(\boldsymbol{u}) \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}(\boldsymbol{u})$ to write the inner product in question as $\int_0^1 \left\langle \boldsymbol{x}, (\boldsymbol{I} - \rho \boldsymbol{Q}^t(\boldsymbol{u}) \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}(\boldsymbol{u}))(\boldsymbol{y} - \boldsymbol{z}) \right\rangle d\tau = \int_0^1 \left\langle \boldsymbol{x}_T, (\boldsymbol{I}_T - \rho \boldsymbol{Q}_T^t(\boldsymbol{u}) \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}_T(\boldsymbol{u}))(\boldsymbol{y}_T - \boldsymbol{z}_T) \right\rangle d\tau$. Using Cauchy-Scwartz inequality, one obtains,

$$\begin{aligned}
&\left\langle \boldsymbol{x}_T, (\boldsymbol{I}_T - \rho \boldsymbol{Q}_T^t(\boldsymbol{u}) \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}_T(\boldsymbol{u}))(\boldsymbol{y}_T - \boldsymbol{z}_T) \right\rangle \\
&\leq \|\boldsymbol{x}_T\|_2 \left\| \boldsymbol{I}_T - \rho \boldsymbol{Q}_T^t(\boldsymbol{u}) \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}_T(\boldsymbol{u}) \right\| \|\boldsymbol{y}_T - \boldsymbol{z}_T\|_2 \\
&\overset{\psi}{\leq} \left(\left| 1 - \rho\delta^{(1)}_{|T|} \right| + \rho\delta^{(2)}_{|T|} \right) \|\boldsymbol{x}\|_2 \|\boldsymbol{y} - \boldsymbol{z}\|_2,
\end{aligned} \tag{9}$$

where in step ψ, we use the RPDH-$(\alpha_{|T|}, \beta_{|T|})$ property of f, or equivalently, the implications (5) and (6) for $s = |T|$ along with the observation that,

$$\begin{aligned}
&\left\| \boldsymbol{I}_T - \rho \boldsymbol{Q}_T^t(\boldsymbol{u}) \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}_T(\boldsymbol{u}) \right\| \\
&= \max \left\{ \left| 1 - \rho \lambda_{\max} \left(\boldsymbol{Q}_T^t(\boldsymbol{u}) \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}_T(\boldsymbol{u}) \right) \right|, \right. \\
&\left. \left| 1 - \rho \lambda_{\min} \left(\boldsymbol{Q}_T^t(\boldsymbol{u}) \boldsymbol{D}(\boldsymbol{u}) \boldsymbol{Q}_T(\boldsymbol{u}) \right) \right| \right\} \\
&\leq \max \left\{ \left| 1 - \rho\beta_{|T|} \right|, \left| 1 - \rho\alpha_{|T|} \right| \right\} \\
&= \begin{cases} 1 - \rho\alpha_{|T|} \text{ if } 0 < \rho \leq \frac{2}{\alpha_{|T|} + \beta_{|T|}} \\ \rho\beta_{|T|} - 1 \text{ if } \rho > \frac{2}{\alpha_{|T|} + \beta_{|T|}} \end{cases}
\end{aligned}$$

Since the RHS of the inequality (9) is independent of τ, the final inequality (7) follows immediately.

For the proof of inequality (8), first construct the vector $\boldsymbol{u} \in \mathbb{R}^n$ such that $\boldsymbol{u}_{T_1} = \boldsymbol{g}(\boldsymbol{y}, \boldsymbol{z})_{T_1}$, and $\boldsymbol{u}_{T_1^C} = \boldsymbol{0}_{T_1^C}$. Then, using the inequality (7), one obtains

$$\begin{aligned}
&\langle \boldsymbol{u}, \boldsymbol{g}(\boldsymbol{y}, \boldsymbol{z}) \rangle \leq \rho'_{|T|} \|\boldsymbol{u}\|_2 \|\boldsymbol{y} - \boldsymbol{z}\|_2 \\
&\Rightarrow \|\boldsymbol{u}_{T_1}\|_2^2 \leq \rho'_{|T|} \|\boldsymbol{u}_{T_1}\|_2 \|\boldsymbol{y} - \boldsymbol{z}\|_2,
\end{aligned}$$

which, after cancellation of $\|\boldsymbol{u}_{T_1}\|_2$ from both sides of the above inequality results in the inequality (8). ■

We will also require the following monotonicity property of ρ'_s for any integer $s \geq 1$:

Lemma 2 (Monotonicity of ρ'_s): For any positive integer s, let ρ'_s be defined as in Lemma 1 (with $|T|$ replaced by s). Then for positive integers $s_1 \leq s_2$, we have $\rho'_{s_1} \leq \rho'_{s_2}$.

Proof: First, note that one can express ρ'_s from Lemma 1 as

$$\rho'_s = \max\{1 - \rho\alpha_s, \rho\beta_s - 1\}. \tag{10}$$

Consider any pair of positive integers (s_1, s_2) such that $s_1 \leq s_2$. To use Eq. (10), first note that $\alpha_{s_1} \geq \alpha_{s_2}$ and $\beta_{s_1} \leq \beta_{s_2}$. This directly follow from the definition of α_s, β_s from inequality (4). Consequently,

$$\begin{aligned}
1 - \rho\alpha_{s_1} \leq 1 - \rho\alpha_{s_2} \leq \max\{1 - \rho\alpha_{s_2}, \rho\beta_{s_2} - 1\} = \rho'_{s_2}, \\
\rho\beta_{s_1} - 1 \leq \rho\beta_{s_2} - 1 \leq \max\{1 - \rho\alpha_{s_2}, \rho\beta_{s_2} - 1\} = \rho'_{s_2}.
\end{aligned} \tag{11}$$

The above two inequalities together imply the result. ■

Before proceeding to analyze the error sequence $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^\star\|_2$, we recall a few definitions from the theory of non-negative matrices [30].

Definition 2 (Non-negative matrix): A square matrix $\boldsymbol{X}$ is said to be non-negative if for every pair of indices i, j, $(\boldsymbol{X})_{ij} \geq 0$.

Definition 3 (Irreducible matrix): A square non-negative matrix $\boldsymbol{X}$ is said to be irreducible, if for any pair of indices i, j, $\exists$ a positive integer t_{ij} such that $(\boldsymbol{X}^{t_{ij}})_{ij} > 0$.

We also recall the following classical result from Perron-Frobenius theory [30], which is going to be useful in our analysis.

Theorem 1 (Perron-Frobenius [30]): Let $\boldsymbol{X} \in \mathbb{R}^{L \times L}$ be a non-negative irreducible matrix. Then, the following results hold:

1) $\exists r > 0$, such that r is an eigenvalue of $\boldsymbol{X}$, and $|\lambda| \leq r$, for any other eigenvalue λ of $\boldsymbol{X}$.
2) $r \in [\min_i \sum_j (\boldsymbol{X})_{ij}, \max_i \sum_j (\boldsymbol{X})_{ij}]$.
3) r has algebraic multiplicity 1, and has strictly positive right and left eigenvectors $\boldsymbol{u}, \boldsymbol{w}^t$ respectively.
4) If $r, \lambda_2, \lambda_3, \ldots, \lambda_s$ are the distinct eigenvalues of $\boldsymbol{X}$ with multiplicities $1, m_2, \ldots, m_s$, with $r > |\lambda_2| > \cdots > |\lambda_s|$, then,
   If $\lambda_2 \neq 0$, as $k \to \infty$, $\boldsymbol{X}^k = r^k \boldsymbol{u}\boldsymbol{w}^t + o(k^{m_2 - 1}|\lambda_2|^k)$.
   If $\lambda_2 = 0$, $\forall k \geq L - 1$, $\boldsymbol{X}^k = r^k \boldsymbol{u}\boldsymbol{w}^t$.

We will also use the following simple but useful lemma:

Lemma 3: Let $\mathcal{G}$ be an undirected connected graph, with an associated non-negative weight matrix $\boldsymbol{X}$. Then,

1) $\boldsymbol{X}^t$ is irreducible.
2) $\boldsymbol{D}_1 \boldsymbol{X} \boldsymbol{D}_2$ is irreducible for any two diagonal matrices $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ which have strictly positive diagonal entries.

3) $\boldsymbol{X}+\boldsymbol{M}$ is irreducible for any non-negative matrix $\boldsymbol{M}$.

Proof: A short proof is provided in Appendix A. ■

We will further use the following lemma that will be useful to find upper bounds on the norm of the error between the iterates produced by an algorithm, and the target vector.

Lemma 4: Let $\boldsymbol{B}\in\mathbb{R}^{L\times L},\ \boldsymbol{b}\in\mathbb{R}^{L}$ be a non-negative matrix and a non-negative vector, respectively. Let $(\boldsymbol{u}^k)_{k\geq 0}$ be a sequence of non-negative vectors in $\mathbb{R}^L$ such that

$$\boldsymbol{u}^{k+1}\preccurlyeq \boldsymbol{B}\boldsymbol{u}^k+\boldsymbol{b},\ k\geq 0.$$

Then, if the matrix $\boldsymbol{B}$ is stable,[1] then the sequence $\{\boldsymbol{u}^k\}_{k\geq 0}$ is a bounded sequence, satisfying

$$\boldsymbol{u}^k\preccurlyeq(\boldsymbol{I}-\boldsymbol{B})^{-1}(\boldsymbol{b}+\boldsymbol{u}^0),$$

for each $k>0$. Also, the sequence $\{\boldsymbol{u}^k\}_{k\geq 0}$ has at least one converging subsequence with limit $\boldsymbol{u}$ which satisfies the above bound.

Proof: The proof is given in Appendix B. ■

B. Notation Used in the Main Results

We now proceed to analyze the evolution of the distance between $\boldsymbol{x}^\star$ and the iterates produced by DiFIGHT. Before presenting the main results, we list the notation used hereafter in the paper:

- $\boldsymbol{x}_v^k$ denote the update of the v^{th} node at time step k, where $1\leq v\leq L$.
- $\boldsymbol{h}^k=[\|\boldsymbol{x}_1^k-\boldsymbol{x}^\star\|_2\ \ldots,\ \|\boldsymbol{x}_L^k-\boldsymbol{x}^\star\|_2]^t,\ \forall k\geq 0$.
- At any step $k\geq 0,\ 1\leq v\leq L$, Λ_v^k is the support set of $\boldsymbol{x}_v^k$. The set Λ denotes the support set of $\boldsymbol{x}^\star$.

Also, if for a node v, the function f_v satisfies $(\alpha_s,\ \beta_s)$-RPDH property, then, since the constants $\alpha_s,\ \beta_s$ depend on the node v, we will replace them in the rest of the paper by $\alpha_{v,s}$ and $\beta_{v,s}$ respectively.

- $\omega_v=|1-\mu_v\frac{\beta_{v,3K}+\alpha_{v,3K}}{2}|+\mu_v\frac{\beta_{v,3K}-\alpha_{v,3K}}{2},\quad 1\leq v\leq L$.
- $\boldsymbol{\Omega}=\mathtt{diag}(\omega_1,\ldots,\ \omega_L),\ \boldsymbol{M}=\mathtt{diag}(\mu_1,\ldots,\ \mu_L)$.
- $\boldsymbol{b}=[\|\nabla^{2K}f_1(\boldsymbol{x}^\star)\|_2,\ldots,\ \|\nabla^{2K}f_L(\boldsymbol{x}^\star)\|_2]^t$.

C. Main Results

1) Deterministic Algorithms: The main results for the deterministic DiFIGHT algorithm are stated in Theorem 2:

Theorem 2: Under the assumption that for each node $v=1,\ldots,L$, the function $\{f_v\}_{v=1}^L$ satisfies the $(\alpha_{v,3K},\ \beta_{v,3K})$-RPDH property, at any iteration k, the iterate produced by DiFIGHT as well as MoDiFIGHT satisfies the following inequality:

$$\boldsymbol{h}^{k+1}\preccurlyeq\alpha\boldsymbol{A}^t\boldsymbol{\Omega}\boldsymbol{h}^k+\alpha\boldsymbol{A}^t\boldsymbol{M}\boldsymbol{b},\tag{12}$$

where $\alpha=\sqrt{3}$ for DiFIGHT, and $\alpha=3$ for MoDiFIGHT.

Furthermore, if $\max_v\sum_{u=1}^L\omega_u a_{uv}<1/\alpha$, or, $\max_v\omega_v<1/\alpha$, then the matrix $\alpha\boldsymbol{A}^t\boldsymbol{\Omega}$ is stable and consequently, the following holds:

$$\boldsymbol{h}^k\preccurlyeq\left(\boldsymbol{I}-\alpha\boldsymbol{A}^t\boldsymbol{\Omega}\right)^{-1}(\alpha\boldsymbol{A}^t\boldsymbol{M}\boldsymbol{b}+\boldsymbol{h}^0).\tag{13}$$

[1]A matrix is said to be stable if it has spectral radius less than unity.

Proof: The proof is presented in Appendix C. ■

Eq.(12) implies that if all the node functions have a common $K-$sparse minimizer $\boldsymbol{x}^\star$ which implies $\boldsymbol{b}=\boldsymbol{0}$, the sequence $\{\boldsymbol{h}^k\}_{k\geq 0}$ converges to zero for a stable $\alpha\boldsymbol{A}^t\boldsymbol{\Omega}$ and thus at each node v, the estimate $\boldsymbol{x}_v^k$ converges absolutely to $\boldsymbol{x}^\star$. On the other hand, consider the case where the nodes have different but closeby stationary points. If $\boldsymbol{x}^\star$ is taken from a small neighborhood of the stationary points, then the vector $\boldsymbol{b}$, though not zero now, will have elements of very small magnitude and thus, (12) ensures that $||\boldsymbol{h}^k||_2$ will assume very small values in the steady state.

2) Randomized Algorithms: We now present the main results regarding the convergence of the randomized DiFIGHT and DIFHTP algorithms for different random selection strategies. For this purpose, we introduce the diagonal matrix $\boldsymbol{P}=\mathtt{diag}(\pi_1,\ldots,\ \pi_L)$ with diagonal entries $\pi_v,\ 1\leq v\leq L$ which were defined in Section III-C and evaluated explicitly for uniform distribution for selection of group of nodes.

Theorem 3: Under the same $(\alpha_{v,3K},\ \beta_{v,3K})$-RPDH condition on the functions $\{f_v\}_{v=1}^L$ as assumed in Theorem 2, for any of the random network persistence strategies, (described at the end of Section III-B), the iterates of the randomized DiFIGHT as well as randomized MoDiFIGHT satisfy the following inequalities at time step n:

$$\mathbb{E}\left[\boldsymbol{h}^{k+1}\right]\preccurlyeq(\boldsymbol{I}-\boldsymbol{P}+\alpha\boldsymbol{P}\boldsymbol{A}^t\boldsymbol{\Omega})\mathbb{E}\left[\boldsymbol{h}^k\right]+\alpha\boldsymbol{P}\boldsymbol{A}^t\boldsymbol{M}\boldsymbol{b},\tag{14}$$

where $\alpha=\sqrt{3}$ for DiFIGHT, and $\alpha=3$ for MoDiFIGHT. Furthermore, under the condition, $\max_v\sum_{u=1}^L a_{uv}\omega_u<1/\alpha$ or, $\max_v\omega_v<1/\alpha$, the following bound is satisfied:

$$\mathbb{E}\left[\boldsymbol{h}^k\right]\preccurlyeq\left(\boldsymbol{I}-\alpha\boldsymbol{A}^t\boldsymbol{\Omega}\right)^{-1}[\boldsymbol{P}^{-1}\mathbb{E}\left[\boldsymbol{h}^0\right]+\alpha\boldsymbol{A}^t\boldsymbol{M}\boldsymbol{b}].\tag{15}$$

Proof: The proof is supplied in Appendix D. ■

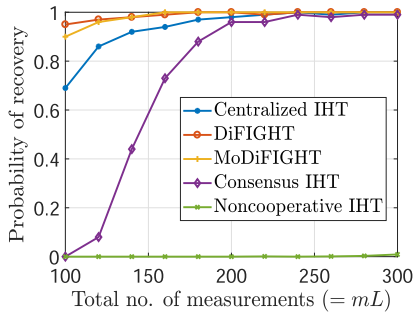
While the above theorem provides conditions for convergence in mean, if $\boldsymbol{x}^\star$ is a stationary point for all the functions $f_v,\ v=1,\ldots,L$, i.e., $\nabla f_v(\boldsymbol{x}^\star)=\boldsymbol{0},\ \forall v=1,\ldots,L$, then we have, $\boldsymbol{b}=\boldsymbol{0}$ and it is easy to show by using Markov inequality that $\sum_{k\geq 0}Prob(h_v^k>\epsilon)<\infty$ for any $\epsilon>0$, which implies that for $v=1,\ldots,L$, $\boldsymbol{x}_v^k\to\boldsymbol{x}^\star$ a.s.

V. Simulation Results

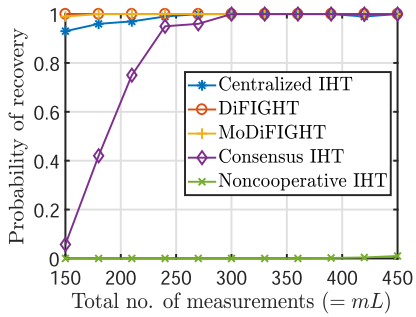
A. Simulation Setup

In this section, we carry out numerical study of the DiFIGHT and MoDiFIGHT algorithms along with the Consensus IHT algorithm, which is similar to DiFIGHT, with the only exception being that the nodes first exchange the estimates and then use their individual gradient vectors for the hard thresholding update. We also plot the performance of the non-cooperative IHT where all the nodes run their respective algorithms but do not communicate with each other, as well as the performance of the centralized IHT algorithm which is executed using the measurements available with all the nodes present in the network. In all the experiments, the unknown vector $\boldsymbol{x}^\star$ has a fixed dimension $n=200$, and sparsity $K=10$. The support of $\boldsymbol{x}^\star$ is constructed by elements following $\mathcal{N}(0,1)$ distribution. We consider networks with $L=10,\ 15$ nodes for our experiments. For each L, the network is generated using Erdős-Reyni model

[31] where there is a link between two nodes is with probability p, and not generated with probability $1-p$. $p = \frac{\ln L}{L}$ is selected to get a connected graph with high probability. The generated graph is checked for full connectivity using depth-first search algorithm, and the process is continued until a connected graph is obtained. The adjacency matrix of the graph thus obtained, is normalized to make it left stochastic, and is used as the combination matrix $\boldsymbol{A}$. For each node, the measurement model is taken to be the noiseless linear measurement model, where the node v has a measurement $\boldsymbol{y}_v$ available with it which is obtained from an unknown signal $\boldsymbol{x}^\star$ via the linear transformation $\boldsymbol{y}_v = \boldsymbol{\Phi}_v \boldsymbol{x}^\star$. The matrix $\boldsymbol{\Phi}_v$ is a $m \times n$ measurement matrix that is generated with entries sampled from i.i.d. $\mathcal{N}(0, 1/m)$ distribution. The node minimizes the cost function $f_v(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y}_v - \boldsymbol{\Phi}_v \boldsymbol{x}\|_2^2$. It is easy to check that the Hessian in this case is given by $\boldsymbol{\Phi}_v^t \boldsymbol{\Phi}_v$. The underlying assumption is that the matrix $\boldsymbol{\Phi}_v$ satisfies the so-called Restricted Isometry Property (RIP) [1] of order K with a Restricted Isometry Constant δ_K. It is then easily seen that the function f_v satisfies the RPDH property of order K, with $\alpha_K = 1 - \delta_K$ and $\beta_K = 1 + \delta_K$. All the algorithms are run for 100 experiments and in each experiment, independent copies of the target vector $\boldsymbol{x}^\star$, and the measurement matrix $\boldsymbol{\Phi}_v$ are generated. However, for a particular L, the underlying network is kept the same in all the experiments.



(a) $L = 10$



(b) $L = 15$

Fig. 1. Probability of recovery vs number of measurements using all nodes in the network.

B. Probability of Recovery Performance

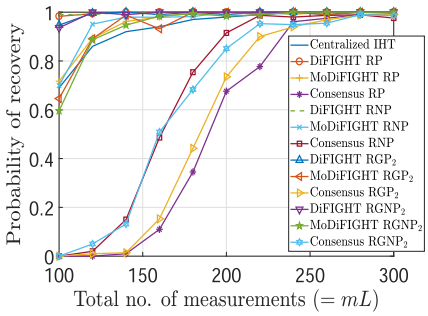
In this experiment, we plot the probability with which the different algorithms recover the unknown signal $\boldsymbol{x}^\star$, against the total number of measurements. The performance of the centralized IHT is evaluated first taking all these measurements. On the other hand, for the distributed algorithms, each node has access to considerably smaller number of measurements. For example, if the total number of measurements is 150 and the network size is $L = 10$, then each node of the network has access to only 15 measurements. To calculate the probability of recovery, we calculate the number of instances (out of the 100 instances) in which an algorithm has a "successful" recovery, where a successful recovery is quantified as follows: 1) for a centralized algorithm, we define the mean square deviation (MSD) of the algorithm to be $\frac{\|\hat{\boldsymbol{x}} - \boldsymbol{x}^\star\|_2^2}{\|\boldsymbol{x}^\star\|_2^2}$, where $\hat{\boldsymbol{x}}$ is the final estimate produced by the centralized algorithm at the end of a run. 2) For the distributed case, we define the MSD of an algorithm to be $\frac{\sum_{v=1}^{L} \|\hat{\boldsymbol{x}}_v - \boldsymbol{x}^\star\|_2^2}{L\|\boldsymbol{x}^\star\|_2^2}$, where $\{\hat{\boldsymbol{x}}_v\}_{1 \le v \le L}$ are the final estimates produced by all the nodes of the network at the end of a run of the algorithm. An instance or a run of an algorithm (centralized or distributed) is called successful if the MSD of the algorithm in that run satisfies $MSD < 10^{-4}$.

Performance of the Deterministic Algorithms: Fig. 1 compares the probability of recoveries of the different algorithms considered in this paper. From this figure, one can appreciate the substantial amount performance gain offered by the DiFIGHT and MoDiFIGHT algorithms over the consensus IHT algorithm and even over that of the centralized algorithm. This gain can be explained using the fact that these diffusion algorithm leverage the diversity offered by the different gradient vectors gathered from the neighborhood of a node. We also see that the distributed algorithms require very small number of measurements for successful recovery compared to the standalone algorithms, as is exemplified by the abysmal performance of the non-cooperative IHT algorithm. For example, from the Fig. 1(b), we see that all the distributed algorithms have recovery probability 1 after m crosses 20, whereas the recovery probability of the non-cooperative algorithm is almost 0 even when m is close to 30. We also observe that the performance of the DiFIGHT and MoDiFIGHT algorithms are very close, with the latter exhibiting slightly poorer performance than the former only for small m ($10 - 12$).

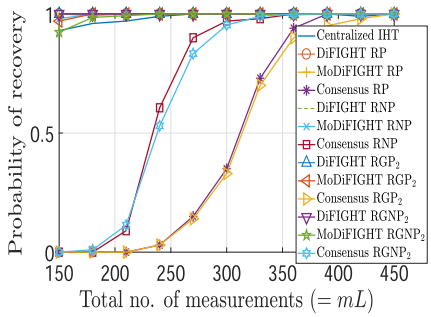
Performance Under the Randomized Strategies: Fig. 2 demonstrates the relative performances of the DiFIGHT, MoDiFIGHT and consensus IHT algorithms for the different randomized node selection strategies which were discussed in Section III-B. We have used group size 2 for the experiment. From the plots we observe that the RP strategy performs the worst among all the four strategies, which is expected as only one node at a time is selected in this strategy. The RGP$_r$ strategy is slightly better than the RP strategy as a few nodes are selected. But the best strategies are seen to be the RNP and RGP$_r$, as in both these strategies many neighboring nodes are selected at a time, which elevates the performance of the distributed algorithms, especially in case of dense networks.

C. Mean Square Deviation Performance

Here, we study the convergence behavior of the proposed algorithm vis-a-vis others, by plotting the learning curves, i.e.,

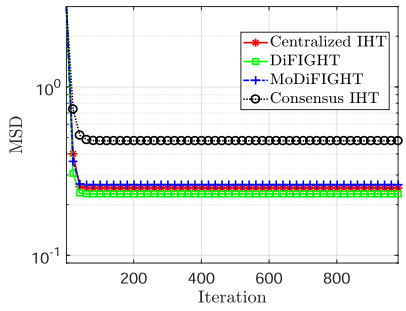


(a) $L = 10$

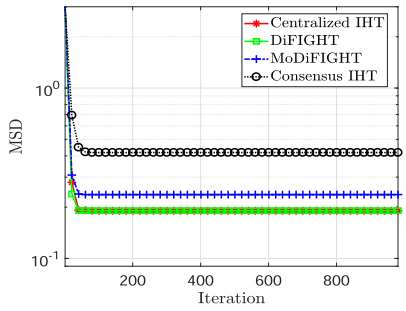


(b) $L = 15$

Fig. 2. Probability of recovery vs number of measurements for randomized node selection strategies.



(a) $L = 10$



(b) $L = 15$

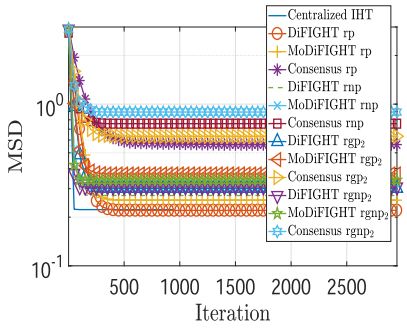
Fig. 3. Comparison of MSDs of deterministic DiFIGHT and MoDiFIGHT.

MSD-vs-iteration index plots. For this purpose, we have used noisy measurements in the nodes, i.e., for each node v in the network, the measurement obtained is given by $\boldsymbol{y}_v = \boldsymbol{\Phi}_v \boldsymbol{x}^\star + \boldsymbol{e}_v$. For our experiments, we have used the same measurement noise vector $\boldsymbol{e}$ for each $\boldsymbol{e}_v$, which is obtained by sampling from a Gaussian random vector with distribution $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_m)$, with $\sigma = 0.1$.

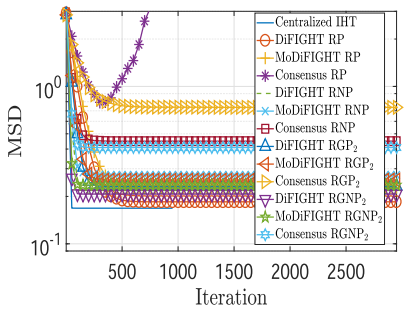
Deterministic Algorithms: In Fig. 3, we compare the MSD of the diffusion and consensus strategies with all the nodes participating at each time slot. It is easily seen from Fig. 3 that for both $L = 10, 15$, the diffusion based methods achieve smaller MSD than the consensus based method. Also, as L increases, the MSD of DiFIGHT becomes closer to that of the centralized IHT, whereas, the gap between the MSDs of MoDiFIGHT and DiFIGHT increase slightly.

Randomized Strategies: For the randomized strategies, the comparative MSD performance is illustrated in Fig. 4. We can observe that for both $L = 10$ and $L = 15$, the consensus based randomized strategies have poorer MSD performance than the DiFIGHT and MoDiFIGHT based counterparts. In fact, one can see that for $L = 15$, some of the consensus based extensions can even diverge, whereas, all the diffusion based extensions are seen to converge, with the final MSD being very close to the one obtained by running a centralized IHT algorithm with measurements available from all the nodes.

Comparison of Algorithms for Fixed and Time Varying Step Sizes: Time varying step sizes have been found to improve convergence behavior in distributed optimization [19], [32] under noisy condition. For comparative MSD assessment, we ran the proposed deterministic DiFIGHT, MoDiFIGHT, and their



(a) $L = 10$



(b) $L = 15$

Fig. 4. Comparison of MSDs of DiFIGHT and MoDiFIGHT for different randomized strategies in the presence of Gaussian measurement error.

randomized counterparts (RGNP_2 based), first with a fixed step size of $\mu_v = 0.2, v = 1, \ldots, L$, and then with a time varying step size $\mu_v(t) = \frac{0.2}{\sqrt{t}},\ v = 1, \ldots, L,\ t \geq 1$ [32]. The corresponding learning curves are shown in Fig. 5. From Fig. 5, it is, however, fair to conclude that the steady state MSD of the proposed

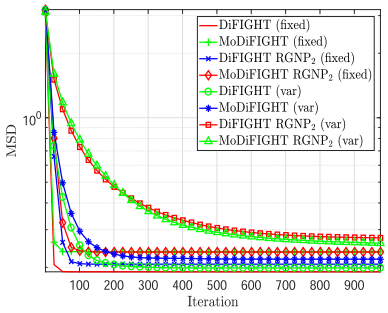


Fig. 5. Comparison of the MSDs of a few deterministic and randomized strategies with fixed and time varying step sizes ($\propto \frac{1}{\sqrt{t}}$). Here $L = 15$.

algorithms remain more or less at par as we move from fixed to time varying step sizes.

VI. Conclusion

In this paper, the conventional IHT algorithm is extended to the diffusion network scenario. First, a diffusion based iterated generalized hard thresholding (DiFIGHT) algorithm is proposed where local gradient descent updates are shared and combined at different nodes followed by hard thresholding. Subsequently, a low complexity modification of the DIFIGHT algorithm, called MoDiFIGHT is proposed, where only a sparse approximation of the gradient update is shared instead of the full update. To reduce communication complexities further, several random node activation strategies are proposed and analyzed where only a few of the nodes in the network can participate in the diffusion process at each iteration while the rest remain idle. Theoretical convergence results are presented for both the DiFIGHT and MoDiFIGHT, both with and without the random selection of nodes.

Appendix A
Proof of Lemma 3

The key observation is that as the graph $\mathcal{G}$ is connected, the associated weight matrix $\boldsymbol{X}$ is irreducible. We now prove the three claims as below:

1) Since $\boldsymbol{X}$ is irreducible, for any two indices u, v, $\exists$ a positive integer t_{uv}, such that $(\boldsymbol{X}^{t_{uv}})_{uv} > 0$. Denoting t_{uv} by t'_{vu}, this means, $((\boldsymbol{X}^{t})^{t'_{vu}})_{vu} = ((\boldsymbol{X}^{t'_{vu}})^{t})_{vu} = (\boldsymbol{X}^{t_{uv}})_{uv} > 0$ which proves irreducibility of $\boldsymbol{X}^t$.
2) Since each element of $\boldsymbol{D}_1$, $\boldsymbol{D}_2$ is strictly positive, $\exists \alpha > 0$ such that d_{1v}, $d_{2v} \geq \alpha$, $\forall v = 1, \ldots, L$. Therefore, the $(u,v)^{\text{th}}$ element of $\boldsymbol{D}_1\boldsymbol{X}\boldsymbol{D}_2$ is $d_{1,u}(\boldsymbol{X})_{uv}d_{2,v} \geq \alpha^2(\boldsymbol{X})_{uv}$. Since $\boldsymbol{X}$ is irreducible, for any $1 \leq u, v \leq L$, there exists a positive integer t_{uv} such that $(\boldsymbol{X}^{t_{uv}})_{uv} > 0$. Therefore, $((\boldsymbol{D}_1\boldsymbol{X}\boldsymbol{D}_2)^{t_{uv}})_{uv} \geq \alpha^2(\boldsymbol{X}^{t_{uv}})_{uv} > 0$, which establishes the claim.
3) Observe that for any $1 \leq u, v \leq L$, $(\boldsymbol{X} + \boldsymbol{M})_{uv} \geq (\boldsymbol{X})_{uv}$ and that $\boldsymbol{X}$ is irreducible. Therefore, $\boldsymbol{X} + \boldsymbol{M}$ is irreducible.

Appendix B
Proof of Lemma 4

First note that since the sequence $(\boldsymbol{u}^k)_{k\geq 0}$, as well as the matrix $\boldsymbol{B}$ and vector $\boldsymbol{b}$ are non-negative, we have, for each $k \geq 0$, $\boldsymbol{u}^{k+1} \preccurlyeq \boldsymbol{B}^{k+1}\boldsymbol{u}^0 + \sum_{j=0}^{k}\boldsymbol{B}^j\boldsymbol{b} \preccurlyeq \sum_{l=0}^{\infty}\boldsymbol{B}^l\boldsymbol{u}^0 + \sum_{j=0}^{\infty}\boldsymbol{B}^j\boldsymbol{b} = (\boldsymbol{I}-\boldsymbol{B})^{-1}(\boldsymbol{b}+\boldsymbol{u}^0)$, where we have used the fact that $\sum_{j=0}^{\infty}\boldsymbol{B}^j = (\boldsymbol{I}-\boldsymbol{B})^{-1}$ as the matrix $\boldsymbol{B}$ is stable. Thus the sequence $(\boldsymbol{u}^k)_{k\geq 0}$ is non-negative as well as upper bounded, which ensures, by the Bolzano-Weierstrass theorem [33] that the sequence $(\boldsymbol{u}^k)_{k\geq 0}$ has at least one converging subsequence, with limit, say, $\boldsymbol{u}$, i.e., there exists a strictly increasing sequence of non-negative integers $k_j \geq 0$, $j = 1, 2, \ldots$ such that $\lim_{j\to\infty}\boldsymbol{u}^{k_j} = \boldsymbol{u}$. Since the set $\{\boldsymbol{u}^k | k_j \geq 0, j = 1, 2, \ldots\}$ is upper bounded by $(\boldsymbol{I}-\boldsymbol{B})^{-1}(\boldsymbol{b}+\boldsymbol{u}^0)$, it follows trivially that $\boldsymbol{u} \preccurlyeq (\boldsymbol{I}-\boldsymbol{B})^{-1}(\boldsymbol{b}+\boldsymbol{u}^0)$, which concludes the proof.

Appendix C
Proof of Theorem 2

For the analysis of both DiFIGHT and MoDiFIGHT, we employ the technique of analysis of Theorem 3.5 of [34] and extend it to the distributed case.

We fix any node $v = 1, \ldots, L$. From the description of the DiFIGHT algorithm in Table I, using the expression for $\hat{\boldsymbol{\psi}}_u^{\,k+1}$ and writing $\boldsymbol{e}_u^k = \boldsymbol{x}_u^k - \boldsymbol{x}^\star - \mu_u(\nabla f_u(\boldsymbol{x}_u^K) - \nabla f_u(\boldsymbol{x}^\star))$, one can derive using triangle inequality that,

$$\begin{aligned}
&\left\|(\boldsymbol{x}_v^{k+1} - \boldsymbol{x}^\star)_{\Lambda_v^{k+1}}\right\|_2 \\
&\leq \sum_{u=1}^{L} a_{uv}\left\|\left(\boldsymbol{e}_u^k\right)_{\Lambda_v^{k+1}}\right\|_2 + \sum_{u=1}^{L} a_{uv}\mu_j\left\|\left(\nabla f_u(\boldsymbol{x}^\star)\right)_{\Lambda_v^{k+1}}\right\|_2 \\
&\leq \sum_{u=1}^{L} a_{uv}\left(\omega_u\left\|\boldsymbol{x}_u^k - \boldsymbol{x}^\star\right\|_2 + \mu_u\left\|\nabla^K f_u(\boldsymbol{x}^\star)\right\|_2\right) \\
&\leq \sum_{u=1}^{L} a_{uv}\left(\omega_u\left\|\boldsymbol{x}_u^k - \boldsymbol{x}^\star\right\|_2 + \mu_u b_u\right) \qquad (16)
\end{aligned}$$

where the last step uses the fact that $\|\nabla^K f_u(\boldsymbol{x}^\star)\|_2 \leq \|\nabla^{2K} f_u(\boldsymbol{x}^\star)\|_2 = b_u$ and the inequality (8) of Lemma 1 which uses the fact that $|\Lambda_v^k \cup \Lambda \cup \Lambda_v^{k+1}| \leq 3K$, along with Lemma 2. On the other hand, since the support of $\boldsymbol{x}_v^{k+1}$ is Λ_v^{k+1}, one obtains,

$$\begin{aligned}
&\left\|\left(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1}\right)_{\Lambda_v^{k+1}}\right\|_2 \geq \left\|\left(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1}\right)_{\Lambda}\right\|_2 \\
&\Leftrightarrow \left\|\left(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1}\right)_{\Lambda_v^{k+1}\setminus\Lambda}\right\|_2 \geq \left\|\left(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1}\right)_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2. \qquad (17)
\end{aligned}$$

Again, using the expression of $\hat{\boldsymbol{\psi}}_u^{\,k+1}$, it is easy to observe that

$$\begin{aligned}
&\left\|\left(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1}\right)_{\Lambda_v^{k+1}\setminus\Lambda}\right\|_2 \\
&\leq \sum_{u=1}^{L} a_{uv}\left\|\left(\boldsymbol{e}_u^k\right)_{\Lambda_v^{k+1}\setminus\Lambda}\right\|_2 + \sum_{u=1}^{L} a_{uv}\mu_u\left\|\left(\nabla f_u(\boldsymbol{x}^\star)\right)_{\Lambda_v^{k+1}\setminus\Lambda}\right\|, \qquad (18)
\end{aligned}$$

where the last step used triangle inequality and the fact that $(\boldsymbol{x}^\star)_{\Lambda_v^{k+1}\setminus\Lambda} = \mathbf{0}$. Similarly, using reverse triangle inequality, one can find,

$$\begin{aligned}
&\left\|\left(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1}\right)_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2 \\
&\geq \left\|(\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star)_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2 - \sum_{u=1}^{L} a_{uv}\left\|(\boldsymbol{e}_u^k)_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2 \\
&- \sum_{u=1}^{L} a_{uv}\mu_j\left\|(\nabla f_u(\boldsymbol{x}^\star))_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2. \qquad (19)
\end{aligned}$$

Therefore, from inequality (17) it follows that

$$\begin{aligned}
&\left\|(\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star)_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2 \\
&\leq \sum_{u=1}^{L} a_{uv}\left(\left\|(\boldsymbol{e}_u^k)_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2 + \left\|(\boldsymbol{e}_u^k)_{\Lambda_v^{k+1}\setminus\Lambda}\right\|_2\right) \\
&+ \sum_{u=1}^{L} a_{uv}\mu_j\left(\left\|(\nabla f_u(\boldsymbol{x}^\star))_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2 + \left\|(\nabla f_u(\boldsymbol{x}^\star))_{\Lambda_v^{k+1}\setminus\Lambda}\right\|_2\right) \\
&\leq \sqrt{2}\sum_{u=1}^{L} a_{uv}\left(\left\|(\boldsymbol{e}_u^k)_{\Lambda_v^{k+1}\Delta\Lambda}\right\|_2 + \mu_u\left\|(\nabla f_u(\boldsymbol{x}^\star))_{\Lambda_v^{k+1}\Delta\Lambda}\right\|_2\right) \\
&\leq \sqrt{2}\sum_{u=1}^{L} a_{uv}\left(\omega_u\left\|\boldsymbol{x}_u^k-\boldsymbol{x}^\star\right\|_2 + \mu_u\left\|\nabla^{2K} f_u(\boldsymbol{x}^\star)\right\|_2\right) \\
&= \sqrt{2}\sum_{u=1}^{L} a_{uv}\left(\omega_j\left\|\boldsymbol{x}_u^k-\boldsymbol{x}^\star\right\|_2 + \mu_u b_u\right) \qquad (20)
\end{aligned}$$

where the last step used inequality (8) of Lemma 1. Therefore, it follows that for all nodes $v = 1, \ldots, L$

$$\left\|\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star\right\|_2 \leq \sqrt{3}\sum_{u=1}^{L} a_{uv}\left(\omega_u\left\|\boldsymbol{x}_u^k-\boldsymbol{x}^\star\right\|_2 + \mu_u b_u\right). \qquad (21)$$

Collecting the above set of inequalities for all the nodes to form a vector inequality and recalling the definition of $\boldsymbol{h}^k$ from Section IV.B, we can then write (21) compactly as

$$\boldsymbol{h}^{k+1} \preccurlyeq \boldsymbol{H}\boldsymbol{h}^k + \boldsymbol{d}, \qquad (22)$$

where $\boldsymbol{H} = \sqrt{3}\boldsymbol{A}^t\boldsymbol{\Omega}$, and $\boldsymbol{d} = \sqrt{3}\boldsymbol{A}^t\boldsymbol{M}\boldsymbol{b}$, where $\boldsymbol{M}$, $\boldsymbol{b}$ are defined in Theorem 2. Now, using lemma 3, we find that $\boldsymbol{H}$ is an irreducible matrix. Then according to the Perron-Frobenius Theorem 1, the maximum eigenvalue (according to absolute value) of $\boldsymbol{H}$ satisfies $r \leq \sqrt{3}\max_u \sum_{v=1}^{L}\omega_u a_{uv}$. By imposing the restriction $\max_v \sum_{u=1}^{L}\omega_u a_{uv} < 1/\sqrt{3}$, we see from (4) of Theorem 1 that the matrix $\boldsymbol{H}$ is stable, and consequently, applying Lemma 4, one finds that $\boldsymbol{h}^k \preccurlyeq (\boldsymbol{I}-\boldsymbol{H})^{-1}[\boldsymbol{d}+\boldsymbol{h}^0] = (\boldsymbol{I}-\alpha\boldsymbol{A}^t\boldsymbol{\Omega})^{-1}[\alpha\boldsymbol{A}^t\boldsymbol{M}\boldsymbol{b}+\boldsymbol{h}^0]$, where $\alpha = \sqrt{3}$. Furthermore, $\max_v \omega_v < 1/\sqrt{3}$ ensures that $\max_v \sum_{u=1}^{L}\omega_u a_{uv} < 1/\sqrt{3}$, which is a weaker sufficient condition for the stability of matrix $\boldsymbol{H}$, that does not require the explicit knowledge of the combination matrix $\boldsymbol{A}$.

In order to derive an evolution inequality for the MoDiFIGHT algorithm, we first note that since $\hat{\boldsymbol{\psi}}_u^{\,k+1} = H_K(\boldsymbol{x}_u^k - \mu_j\nabla f_u(\boldsymbol{x}_u^k)),\ u = 1, 2, \ldots, L$, one can readily use the above analysis of the DiFIGHT algorithm, restricted to a single node, to derive the following inequality for all nodes $u = 1, \ldots, L$:

$$\left\|\hat{\boldsymbol{\psi}}_u^{\,k+1} - \boldsymbol{x}^\star\right\|_2 \leq \sqrt{3}\omega_u\left\|\boldsymbol{x}_u^k-\boldsymbol{x}^\star\right\|_2 + \sqrt{3}b_u. \qquad (23)$$

On the other hand, we have $\boldsymbol{x}_v^{k+1} = H_K(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1})$. This implies,

$$\begin{aligned}
\left\|\left(\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star\right)_{\Lambda_v^{k+1}}\right\|_2 &\leq \sum_{u=1}^{L} a_{uv}\left\|\left(\hat{\boldsymbol{\psi}}_u^{\,k+1}-\boldsymbol{x}^\star\right)_{\Lambda_v^{k+1}}\right\|_2 \\
&\leq \sum_{u=1}^{L} a_{uv}\left\|\hat{\boldsymbol{\psi}}_u^{\,k+1}-\boldsymbol{x}^\star\right\|_2. \qquad (24)
\end{aligned}$$

Next, for finding an upper bound of $\|(\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star)_{\Lambda\setminus\Lambda_v^{k+1}}\|_2$, using the definition of the H_K operator, we can write, $\|(\sum_{u=1}^{L} a_{uv}(\hat{\boldsymbol{\psi}}_u^{\,k+1}-\boldsymbol{x}^\star)_{\Lambda_v^{k+1}\setminus\Lambda}\|_2 \geq \|(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1})_{\Lambda\setminus\Lambda_v^{k+1}}\|_2$. Using the reverse triangle inequality, we further obtain, $\|(\sum_{u=1}^{L} a_{uv}\hat{\boldsymbol{\psi}}_u^{\,k+1})_{\Lambda\setminus\Lambda_v^{k+1}}\|_2 \geq \|(\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star)_{\Lambda\setminus\Lambda_v^{k+1}}\|_2 - \|(\sum_{u=1}^{L} a_{uv}(\hat{\boldsymbol{\psi}}_u^{\,k+1} - \boldsymbol{x}^\star)_{\Lambda\setminus\Lambda_v^{k+1}}\|_2$. This leads to

$$\begin{aligned}
&\left\|\left(\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star\right)_{\Lambda\setminus\Lambda_v^{k+1}}\right\|_2 \\
&\leq \sqrt{2}\sum_{u=1}^{L} a_{uv}\left\|\left(\hat{\boldsymbol{\psi}}_u^{\,k+1}-\boldsymbol{x}^\star\right)_{\Lambda_v^{k+1}\Delta\Lambda}\right\| \\
&\leq \sqrt{2}\sum_{u=1}^{L} a_{uv}\left\|\hat{\boldsymbol{\psi}}_u^{\,k+1}-\boldsymbol{x}^\star\right\|. \qquad (25)
\end{aligned}$$

Consequently, taking the contributions from the sets Λ_v^{k+1} and $\Lambda\setminus\Lambda_v^{k+1}$ from Eqs. (24), (25), we obtain,

$$\Rightarrow \left\|\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star\right\|_2 \leq \sqrt{3}\sum_{u=1}^{L} a_{uv}\left\|\hat{\boldsymbol{\psi}}_u^{\,k+1}-\boldsymbol{x}^\star\right\|. \qquad (26)$$

Therefore the inequalities (23) and (26) together yield the following main inequality governing the evolution of $\|\boldsymbol{x}_i^{k+1}-\boldsymbol{x}^\star\|_2$ for node v for MoDiFIGHT:

$$\begin{aligned}
&\left\|\boldsymbol{x}_v^{k+1}-\boldsymbol{x}^\star\right\|_2 \\
&\leq 3\sum_{u=1}^{L} a_{uv}\omega_j\left\|\boldsymbol{x}_u^k-\boldsymbol{x}^\star\right\|_2 + 3\sum_{u=1}^{L} a_{uv}\mu_u\left\|\nabla^{2K} f_u(\boldsymbol{x}^\star)\right\|_2 \qquad (27)
\end{aligned}$$

We note that the inequality (27) is essentially the same as the inequality (21), with the factor $\sqrt{3}$ on RHS replaced by a factor of 3. Thus, using similar analysis as presented in case of DiFIGHT after inequality (21), we arrive at the following vector inequality:

$$\boldsymbol{h}^{k+1} \preccurlyeq \sqrt{3}\boldsymbol{H}\boldsymbol{h}^k + \sqrt{3}\boldsymbol{d}, \qquad (28)$$

where $\boldsymbol{H}$, $\boldsymbol{d}$ are defined as in the analysis of the DiFIGHT algorithm. Clearly, a sufficient condition for the matrix $\sqrt{3}\boldsymbol{H}$ to be stable is $\max_v \sum_{u=1}^{L} a_{uv}\omega_u < 1/3$, or a weaker condition is $\max_v \omega_v < 1/3$. Under this, from Lemma 4, $\{\boldsymbol{h}^k\}_{k\geq 0}$ is a bounded sequence with $\boldsymbol{h}^k \preccurlyeq (\boldsymbol{I} - \sqrt{3}\boldsymbol{H})^{-1}[\sqrt{3}\boldsymbol{d} + \boldsymbol{h}^0] = (\boldsymbol{I} - \alpha \boldsymbol{A}^t \boldsymbol{\Omega})^{-1}[\alpha \boldsymbol{A}^t \boldsymbol{M}\boldsymbol{b} + \boldsymbol{h}^0]$, where $\alpha = 3$.

Appendix D
Proof of Theorem IV.3

To carry out the proof, let us first consider a node $v \in G_k$, where G_k is the group of nodes chosen at time k. For both the randomized DiFIGHT and MoDiFIGHT algorithms, derivation of the evolution of the norm of the error $\|\boldsymbol{x}_v^{k+1} - \boldsymbol{x}^\star\|_2$ in terms of $\|\boldsymbol{x}_v^{k} - \boldsymbol{x}^\star\|_2$ will be identical to that of their deterministic counterparts, that is either the inequality (21) or (27). Therefore, for any $v \in G_k$, one obtains,

$$h_v^{k+1} \leq \alpha \sum_{u=1}^{L} a_{uv} \left(\omega_u h_u^k + \mu_u b_u\right), \tag{29}$$

where $\alpha = \sqrt{3}$ or $\alpha = 3$, depending on whether DiFIGHT or MoDiFIGHT is used. However, the nodes not in G do not update their estimate, so that for $v \notin G_k$, one has

$$h_v^{k+1} = h_v^k. \tag{30}$$

Taking the inequalities (29) and (30) together, the following evolution inequality is obtained for the vector $\boldsymbol{h}^k$:

$$\boldsymbol{h}^{k+1} \preccurlyeq \boldsymbol{B}_k \boldsymbol{h}^k + \boldsymbol{c}_k, \tag{31}$$

where the vector $\boldsymbol{c}_k$ and the matrix $\boldsymbol{A}_k$, where $\boldsymbol{B}_k = \alpha \boldsymbol{A}_k^t \Omega$ with $\boldsymbol{A}_k = [\boldsymbol{a}_{k,1} \cdots \boldsymbol{a}_{k,L}]$, are determined as below:

$$c_{k,v} = \begin{cases} \alpha \boldsymbol{a}_v^t \boldsymbol{M}\boldsymbol{b}, & v \in G \\ 0, & v \notin G \end{cases} \tag{32}$$

$$\boldsymbol{a}_{k,v} = \begin{cases} \boldsymbol{a}_v, & v \in G \\ \frac{\boldsymbol{e}_v}{\alpha\omega_v}, & v \notin G \end{cases} \tag{33}$$

where $\boldsymbol{e}_v$ is the the column vector with all entries set to 0 except for the v^{th} entry which is set to 1. We will now use the compact inequality (31) to derive condition for stability of the mean of the sequence $\{\boldsymbol{h}^k\}$.

Taking expectation of both sides of the inequality (31) we find

$$\mathbb{E}\left[\boldsymbol{h}^{k+1}\right] \preccurlyeq \boldsymbol{B}\mathbb{E}\left[\boldsymbol{h}^k\right] + \boldsymbol{c} \tag{34}$$

$$\Rightarrow \mathbb{E}\left[\boldsymbol{h}^{k+1}\right] \preccurlyeq \boldsymbol{B}^{k+1}\mathbb{E}\left[\boldsymbol{h}^0\right] + \sum_{j=0}^{k} \boldsymbol{B}^j \boldsymbol{c} \tag{35}$$

where $\boldsymbol{B} = \mathbb{E}[\boldsymbol{B}_k]$ and $\boldsymbol{c} = \mathbb{E}[\boldsymbol{c}_k],\ k \geq 0$. It follows that the right hand side of the inequality (35) converges if the matrix $\boldsymbol{B}$ is stable. The matrix $\boldsymbol{B}$ and vector $\boldsymbol{c}$ have different forms for different strategies and for the different algorithms. We find them as below:

Let us first find $\mathbb{E}[\boldsymbol{b}_{k,v}]$, where $\boldsymbol{b}_{k,v}$ is the v^{th} column of the matrix $\boldsymbol{B}_k$. Since $\mathbb{E}[\boldsymbol{B}_k] = \alpha\mathbb{E}[\boldsymbol{A}_k^t]\boldsymbol{\Omega}$, we only require to find the expected value of $\boldsymbol{a}_{k,v}$. Note that the column $\boldsymbol{a}_{k,v}$ can take only two vector values, $\boldsymbol{a}_v$ and $\boldsymbol{e}_v/(\alpha\omega_v)$, depending on whether the node v participates or not in the diffusion process at the k^{th} time step. Therefore,

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{a}_{k,v}\right] &= \pi_v \boldsymbol{a}_v + (1 - \pi_v)\frac{\boldsymbol{e}_v}{\alpha\omega_v},\ v = 1, \ldots, L \\ &\Rightarrow \mathbb{E}\left[\boldsymbol{A}_k\right] = \boldsymbol{A}\boldsymbol{P} + \frac{(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{\Omega}^{-1}}{\alpha} \\ &\Rightarrow \boldsymbol{B} = \boldsymbol{I} - \boldsymbol{P} + \alpha \boldsymbol{P}\boldsymbol{A}^t\boldsymbol{\Omega} \end{aligned} \tag{36}$$

where $\boldsymbol{P} = \texttt{diag}(\pi_1, \ldots \pi_L)$. Note that the diagonal matrix $\boldsymbol{P}$ varies with different strategies and have diagonal entries π_v that can be found from the discussion of Section III.C.

In a similar manner one can find:

$$\mathbb{E}\left[c_{k,v}\right] = \alpha \boldsymbol{a}_v^t \boldsymbol{M}\boldsymbol{b}\pi_v \Rightarrow \mathbb{E}\left[\boldsymbol{c}_k\right] = \alpha \boldsymbol{P}\boldsymbol{A}^t\boldsymbol{M}\boldsymbol{b}. \tag{37}$$

Now, observe that as the network is connected, the matrix $\boldsymbol{A}$ is irreducible. Also, $\boldsymbol{P}$ and $\boldsymbol{\Omega}$, are non-negative diagonal matrices. Thus, using Lemma 3, and using the Perron-Frobenius theory, we can conclude that the matrix $\boldsymbol{B}$ can be ensured to be stable if the functions $\{f_v\}_{1\leq v\leq L}$ are chosen such that

$$\max_v \left(1 - \pi_v + \alpha\pi_v \sum_{u=1}^{L} a_{uv}\omega_u\right) < 1 \Leftrightarrow \max_v \sum_{u=1}^{L} a_{uv}\omega_u < \frac{1}{\alpha}. \tag{38}$$

Clearly, a weaker requirement to ensure stability of $\boldsymbol{B}$ is to choose the functions $\{f_v\}_{v=1}^{L}$ such that $(\max_v \omega_v) < 1/\alpha$. Then, using Lemma 4, from Equation (35), we have

$$\mathbb{E}\left[\boldsymbol{h}^k\right] \preccurlyeq (\boldsymbol{I} - \boldsymbol{B})^{-1}[\mathbb{E}\left[\boldsymbol{h}^0\right] + \boldsymbol{c}], \tag{39}$$

from which the result follows trivially.

References

[1] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Berlin, Germany: Springer, 2013.

[2] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, Apr. 2012.

[3] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

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

[5] S. Patterson, Y. C. Eldar, and I. Keidar, "Distributed compressed sensing for static and time-varying networks," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 4931–4946, Oct. 2014.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[7] C. Ravazzi, S. Fosson, and E. Magli, "Energy-saving gossip algorithm for compressed sensing in multi-agent systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 5060–5064.

[8] C. Ravazzi, S. M. Fosson, and E. Magli, "Distributed iterative thresholding for $l_0 l_1$-regularized linear inverse problems," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 2081–2100, Feb. 2015.

[9] P. D. Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

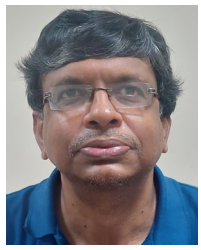
[10] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

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

[12] P. D. Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.
[13] A. H. Sayed, *Diffusion Adaptation Over Networks*, vol. 3, Academic Press, 2013.
[14] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks-part I: Transient analysis," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3487–3517, Jun. 2015.
[15] J. Chen, and A. H. Sayed, "On the learning behavior of adaptive networks-part II: Performance analysis," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3518–3548, Jun. 2015.
[16] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.
[17] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4480–4485, Aug. 2012.
[18] S. Vlaski and A. H. Sayed, "Proximal diffusion for stochastic costs with non-differentiable regularizers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 3352–3356.
[19] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2835–2850, Jun. 2016.
[20] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6329–6344, Dec. 2016.
[21] S. Chouvardas, G. Mileounis, N. Kalouptsidis, and S. Theodoridis, "Greedy sparsity-promoting algorithms for distributed learning," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1419–1432, Mar. 2015.
[22] A. Zaki, A. Venkitaraman, S. Chatterjee, and L. K. Rasmussen, "Greedy sparse learning over network," *IEEE Trans. Signal. Inf. Process. Netw.*, vol. 4, no. 3, pp. 424–435, Sep. 2018.
[23] A. Zaki, P. P. Mitra, L. K. Rasmussen, and S. Chatterjee, "Estimate exchange over network is good for distributed hard thresholding pursuit," *Signal Process.*, vol.156, pp. 1–11, 2019.
[24] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
[25] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.
[26] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.
[27] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks-part I: Modeling and stability analysis," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 811–826, Feb. 2015.
[28] C. Ravazzi, S. M. Fosson, and E. Magli, "Randomized algorithms for distributed nonlinear optimization under sparsity constraints," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1420–1434, Mar. 2016.
[29] S. Bahmani, B. Raj, and P. T. Boufounos, "Greedy sparsity-constrained optimization," *J. Mach. Learn. Res.*, vol. 14, pp. 807–841, Mar. 2013.
[30] E. Seneta, *Non-Negative Matrices and Markov Chains*. Berlin, Germany: Springer, 2006.
[31] P. Erdds and A. R&WI, "On random graphs I," *Publicationes Math. Debrecen*, vol. 6, pp. 290–297, 1959.
[32] D. Yuan and D. W. Ho, "Randomized gradient-free method for multiagent optimization over time-varying networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1342–1347, Aug. 2014.
[33] W. Rudin *et al.*, *Principles of Mathematical Analysis*, vol. 3, New York, NY, USA: McGraw-hill, 1964.
[34] S. Foucart, "Hard thresholding pursuit: An algorithm for compressive sensing," *SIAM J. Numer. Anal.*, vol. 49, no. 6, pp. 2543–2563, 2011.

Samrat Mukhopadhyay received the B.E. degree from the Indian Institute of Engineering Science and Technology (IIEST), Shibpur, India, in 2012, the M.E. degree from the Indian Institute of Science (IISC), Bangalore, India, in 2014, and the Ph.D. from the Indian Institute of Technology Kharagpur (IIT Kharagpur), Kharagpur, India, in 202.

He is currently an Assistant Professor with the Electronics Engineering Department of IIT(ISM), Dhanbad, India. His research interests include compressed sensing, adaptive filtering, high dimensional statistics, machine learning, and optimization.



Mrityunjoy Chakraborty (Senior Member, IEEE) received bachelor of engineering degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 1983, the master of technology degree in electrical engineering from the Indian Institute of Technology Kanpur, India, in 1985, and the Doctor of Philosophy degree in electrical engineering from the Indian Institute of Technology Delhi, India, in 1994. In 1994, he joined the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology Kharagpur, India, as a Lecturer and is currently a Professor (also the Chair) with the same Department. His research interests include digital and adaptive signal processing, VLSI signal processing, linear algebra, and compressive sensing.

During 2017—2020, he was a Senior Editorial Board (SEB) Member of the IEEE SIGNAL PROCESSING MAGAZINE. Earlier, he was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, PART I during 2004–2007 and 2010–2012, and PART II during 2008–2009, apart from being a SEB Member of the IEEE JOURNAL ON EMERGING TECHNIQUES IN CIRCUITS AND SYSTEMS during 2016–2017. During 2016—2018, he was the Chair of the DSP Technical Committee (TC) of the IEEE Circuits and Systems Society. He has also been the Guest Editor of the *EURASIP Journal on Advances in Signal Processing* and SPECIAL ISSUES OF IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, the DSP Track Co- Chair of ISCAS 2015–2021, the TPC Co-Chair of IEEE SIPS-2018, the Special Session Co- Chair of DSP-18, and the Gabor Track Chair of DSP-15. He is a Co-Founder of the Asia Pacific Signal and Information Processing Association (APSIPA), was a Member of the APSIPA BOG during 2013–2016, and was also the Chair of the APSIPA TC on Signal and Information Processing Theory and Methods. During 2012—2020, he was the General Chair of the National Conference on Communications.

Prof. Chakraborty is a Fellow of the National Academy of Sciences, India, and Indian National Academy of Engineering (INAE). Recently, he received the prestigious Chair Professorship of the INAE. During 2012–2013, he was selected as a Distinguished Lecturer of the APSIPA.