Multi-hop Diffusion LMS for Energy-constrained Distributed Estimation

Wuhua Hu, *Member, IEEE*, and Wee Peng Tay, *Senior Member, IEEE*

***Abstract*—We propose a multi-hop diffusion strategy for a sensor network to perform distributed least mean-squares (LMS) estimation under local and network-wide energy constraints. At each iteration of the strategy, each node can combine intermediate parameter estimates from nodes other than its physical neighbors via a multi-hop relay path. We propose a rule to select combination weights for the multi-hop neighbors, which can balance between the transient and the steady-state network mean-square deviations (MSDs). We study two classes of networks: simple networks with a unique transmission path from one node to another, and arbitrary networks utilizing diffusion consultations over at most two hops. We propose a method to optimize each node's information neighborhood subject to local energy budgets and a network-wide energy budget for each diffusion iteration. This optimization requires the network topology, and the noise and data variance profiles of each node, and is performed offline before the diffusion process. In addition, we develop a fully distributed and adaptive algorithm that approximately optimizes the information neighborhood of each node with only local energy budget constraints in the case where diffusion consultations are performed over at most a predefined number of hops. Numerical results suggest that our proposed multi-hop diffusion strategy achieves the same steady-state MSD as the existing one-hop adapt-then-combine diffusion algorithm but with a lower energy budget.**

***Index Terms*—Multi-hop diffusion adaptation, distributed estimation, combination weights, energy constraints, convergence rate, mean-square deviation, sensor networks**

I. Introduction

Distributed estimation arises in a wide range of contexts, including sensor networks [1]–[3], smart grids [4], [5], machine learning [6], [7], and biological networks [8]–[10]. Several useful distributed solutions have been developed for this purpose, such as consensus strategies [2], [11]–[13], incremental strategies [14], [15], and diffusion strategies [16]–[19]. The diffusion strategies are particularly attractive because they are scalable, robust, fully-distributed, and endow networks with real-time adaptation and learning abilities [19]. They have superior stability ranges and transient performance compared to the consensus strategies when constant step-sizes are necessary to enable continuous adaptation under varying network conditions [20]. The mean-square stability of diffusion has also been shown to be insensitive to topological changes caused by asynchronous cooperation among the network nodes [21].

This work was supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grants MOE2013-T2-2-006 and MOE2014-T2-1-028. W. Hu and W. P. Tay are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore; Emails: {hwh, wptay}@ntu.edu.sg

In each iteration of a diffusion strategy, each node obtains intermediate parameter estimates from its neighbors,[1] which are those nodes within communication range of itself. We call these neighboring nodes the *physical neighbors* of the node. The communication cost per iteration of each node in a static network is thus fixed, and the total communication cost can be large if the diffusion algorithm converges slowly. To reduce the number of communication links, [22] and [23] limit each node to selecting only one of its neighbors for consultation based on the neighbors' current mean-square deviation (MSD) estimates and a variance-product metric, respectively. Simulations showed that in the steady state these two strategies outperform the probabilistic or gossip alternatives where a single neighbor is randomly selected for consultation at each iteration [24]–[26]. The reference [27] proposed a heuristic algorithm to discount physical neighbors with large excess mean-square errors, while [28], [29] suggested diffusing only a part of the intermediate estimate vector at each iteration so as to reduce the amount of information exchanged, and consequently the communication cost. A game theoretic approach with provable stability was proposed in [30] for each node to learn in a distributed manner whether to diffuse its estimate based on a utility function that captures the trade-off between its contribution and energy expenditure. A similar idea was also presented in [31] with numerical validation.

When designing or upgrading a cooperative sensor network, the strategies in the aforementioned literature are unable to account for predefined node energy budgets even though they are more energy-efficient overall. This prevents energy efficiency planning even if we have knowledge about the network operating environment (which may be inferred periodically from historical data). Moreover, as these strategies only allow a node to exchange information with its physical neighbors, this limits the estimation performance that a network can achieve. For example, consider an undirected network, as shown in Fig. 1, in which all edges have the same communication length, and the data model, except for the node noise variances, is the same as in the example in Section VI-A. Relying on one-hop communications, the traditional adapt-then-combine (ATC) diffusion strategy [19] invokes 8 broadcasts[2] per iteration and results in an average steady-state network MSD of -49.0 dB. In contrast, if we allow two-hop consultations, then the steady-state network MSD can be improved by 2

[1] If a node receives and incorporates an intermediate parameter estimate from another node into its own estimate, we say that the former node consults the latter node.

[2] In this paper, a broadcast means communication of a node with all of its directly reachable neighbors as defined in Section II.

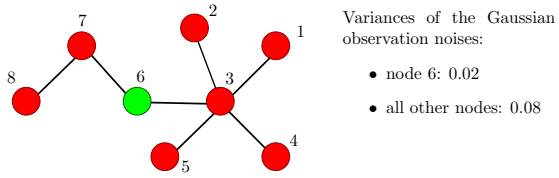


Fig. 1: A toy example to motivate the use of multi-hop diffusion. Other data used is referred to Section VI-A.

dB with the same number of broadcasts, or kept the same with only 4 broadcasts by letting nodes 3, 6, and 7 broadcast their intermediate estimates, and node 3 relay the intermediate estimate from node 6 to nodes 1, 2, 4, and 5 at every iteration. In the latter case, the communication cost per iteration is *halved* compared to the ATC diffusion strategy, but the steady-state network MSD remains the same since "high quality" information from node 6 is diffused to more nodes than in the ATC strategy. Although the implementation complexity is somewhat increased (node 3 needs to be programmed to rebroadcast what it receives from node 6), this example shows that to achieve an optimal network MSD-communication cost trade-off requires the use of multi-hop diffusions.

In this paper, we consider diffusion estimation with local and network-wide energy budgets per iteration, and the use of *multi-hop* consultations, i.e., a node that is not a physical neighbor can transmit its intermediate estimate to another node via a relay path in the network within the same iteration step. This is in sharp contrast to all aforementioned literature, which considers only *single-hop* consultations in each iteration. Our main contributions are the following:

(i) We generalize the concept of single-hop diffusion from physical neighbors to multi-hop diffusion from a set of *information neighbors*. In particular, we propose a multi-hop version of the ATC diffusion algorithm, which we call mATC. We formulate and apply mATC to a distributed estimation problem with local and network-wide energy constraints.

(ii) For a given set of information neighbors, we provide a rule to select combination weights for mATC that optimizes an approximate trade-off between the convergence rate of the algorithm and the steady-state network MSD.

(iii) Given the network topology and data and noise variance profiles of each node, we show how to select information neighbors to minimize an upper bound of the steady-state network MSD, subject to local and network-wide energy budgets per iteration, in two network classes: simple networks with a unique transmission path from one node to another, and arbitrary networks utilizing diffusion consultations over at most two hops. We formulate the problem as an offline centralized mixed integer linear program (MILP), and show that the selection is invariant to homogeneous scaling of node observation noise variances.

(iv) Our MILP requires knowledge of the network topology, and data and noise variance profiles of each node, which may be impractical in some applications. To overcome these requirements, we develop an approximate distributed and adaptive optimization algorithm to select the information neighbors for arbitrary networks utilizing diffusion consultations over at most h hops, in the absence of a network-wide energy budget constraint.

The concept of multi-hop diffusion unifies non-cooperative, distributed diffusion and centralized estimation strategies into a single framework, and allows us to study the trade-offs amongst these strategies easily. Our proposed strategy has the advantage that it achieves good trade-offs between estimation accuracy and predefined *hard* energy budgets, which the standard diffusion strategy or the approaches in [22], [23], [27], [30] cannot incorporate. This is also different from [26], which considers *average* energy budget constraints, and single-hop diffusions. As wireless sensor networks with renewable energy sources become more popular in applications, energy constraints need to be accounted for explicitly in the estimation algorithm [32]. We also note that multi-hop diffusion is different from geographic gossip [33] (or path averaging gossip [34]), which relies on randomized pair-wise (or relay path-wise) cooperation that exploits geographic knowledge (but not data and noise variance profiles) of the network to achieve more efficient *average consensus*.

Different from one-hop information transmission in the standard diffusion strategy, our proposed multi-hop diffusion strategy requires information relaying. If each multi-hop relaying is to be completed within each diffusion iteration, as is assumed in our analysis, this may require the nodes to take observations at a slower rate due to a longer communication delay at each iteration. The extra delay, however, can be minimized if one can perform the relaying over multiple diffusion iterations, so that intermediate estimates of information neighbors more than one hop away are combined only in a later diffusion iteration (we call this *asynchronous* mATC). Our simulation results in Section VI demonstrate that asynchronous mATC has similar MSD performance as mATC.

A major drawback of our proposed MILP solution in contribution (iii) is the need for a centralized optimizer and knowledge of the network topology, and data and noise variance profiles. This typically holds only in applications in which the network topology is static (e.g., in sensor networks used for structural health monitoring [35], [36]), and in which sensors' data and noise variance profiles do not change frequently. Each node in the network monitors its empirical data and noise variances, and trigger the centralized optimizer to re-calibrate the network whenever a significant change in variance profile is detected. Note however that even if the network is not re-calibrated, the parameter estimation procedure does not diverge as a result, but there is a loss in energy efficiency. Our proposed distributed and adaptive procedure in contribution (iv) avoids these issues, but works only in applications in which there are no network-wide energy budget constraints. This last requirement can be somewhat mitigated by imposing sufficiently tight local energy budgets at each node in the network design stage.

This paper is an extension of our conference paper [37], which assumes a simple network with a unique transmission path from one node to another. We have extended the results to cover a network with an arbitrary topology while restricting

the information neighbors to be within two hops away from a node. We also derive valid inequalities by exploiting the problem structures to enhance the MILP solution processes in both network cases. To deal with large-scale networks, we further present an efficient procedure for obtaining an approximate solution. In addition, to overcome the limitation of searching for a centralized solution, we develop a real-time algorithm that yields an approximate, distributed and adaptive solution.

The rest of this paper is organized as follows. In Section II, we introduce our data model and notations, and formulate the energy-constrained distributed optimization problem. In Section III, we introduce the concept of multi-hop diffusion adaptation. In Section IV, we propose a combination weight to optimize an approximate trade-off between the convergence rate and the steady-state network MSD, and in Section V, we show how to choose an approximately optimal set of information neighbors for every node in two classes of networks using an offline optimization, and also general networks with only local energy budget constraints in an adaptive procedure. Numerical results and conclusions follow in Sections VI and VII, respectively.

Notations. The notation $\mathbb{R}_{\geq 0}$ denotes the space of non-negative real numbers, $|\mathcal{N}|$ denotes the cardinality of a discrete set $\mathcal{N}$, $\mathbf{1}_N$ represents a vector of size N with all entries equal to one, I_N is an $N \times N$ identity matrix, A^T is the transpose of the matrix A, and $\lambda_{max}(A)$ and $\rho(A)$ are the largest eigenvalue and the largest absolute eigenvalue of the matrix A, respectively. The operation $A \otimes B$ denotes the Kronecker product of the two matrices A and B. The relation $A \succeq$ (or $\preceq$)B means that the matrix $A - B$ is positive (or negative) semi-definite, and similarly $A \succ$ (or $\prec$)B means the matrix $A - B$ is positive (or negative) definite. The notation col$\{\cdot\}$ denotes a column vector in which its arguments are stacked on top of each other, diag$\{\cdot\}$ denotes a diagonal matrix constructed from its arguments. We use boldface letters to denote random quantities (e.g., $\boldsymbol{x}$) and normal letters to denote their realizations or deterministic quantities (e.g., x). The symbol $\mathbb{E}\boldsymbol{x}$ denotes the expectation of the random variable $\boldsymbol{x}$, and "s.t." is abbreviation for "subject to".

II. Problem Formulation

We adopt the same notations as in [19], [20] for our problem formulation. Consider a network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, ..., N\}$ is the set of nodes, and $\mathcal{E}$ is the set of communication links between nodes.[3] Node l is said to be a *physical neighbor* of node k if either $(l, k) \in \mathcal{E}$ or $l = k$, and is said to be within the *multi-hop neighborhood* of node k if there is a path in $\mathcal{G}$ from node l to node k. Let the physical neighborhood of node k be $\mathcal{N}_{k\leftarrow}$, and its multi-hop neighborhood be $\overline{\mathcal{N}}_{k\leftarrow}$. We have $\mathcal{N}_{k\leftarrow} \subseteq \overline{\mathcal{N}}_{k\leftarrow}$.

On the other hand, we say that node l is within the *reachable neighborhood* of node $k \neq l$ if there is a path in $\mathcal{G}$ from node k to node l. We say that node l is directly reachable from node k if $(k, l) \in \mathcal{E}$. We let $\mathcal{N}_{k\rightarrow}$ be the set of directly reachable neighbors of node k, and $\overline{\mathcal{N}}_{k\rightarrow}$ be the reachable neighborhood of node k. We have $\mathcal{N}_{k\rightarrow} \subseteq \overline{\mathcal{N}}_{k\rightarrow}$. The various types of neighbors are illustrated in Fig. 2.

At every iteration i, each node k is able to observe realizations $\{d_k(i),\, u_{k,i}\}$ of a scalar random process $\boldsymbol{d}_k(i)$ and a $1 \times M$ vector random process $\boldsymbol{u}_{k,i}$ with a positive definite covariance matrix, $R_{u,k} = \mathbb{E}\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i} \succ 0$. The random processes $\{d_k(i),\, u_{k,i}\}$ are related via the linear regression model [19]:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}\omega^o + \boldsymbol{v}_k(i),$$

where ω^o is an $M \times 1$ parameter to be estimated, and $\boldsymbol{v}_k(i)$ is measurement noise with variance $\sigma_{v,k}^2$, and assumed to be temporally white and spatially independent, i.e.,

$$\mathbb{E}\boldsymbol{v}_k^*(i)\boldsymbol{v}_l(j) = \sigma_{v,k}^2\delta_{kl}\delta_{ij},$$

where δ_{kl} is the Kronecker delta function. The regression data $\boldsymbol{u}_{k,i}$ are likewise assumed to be temporally white and spatially independent. The noise $\boldsymbol{v}_k(i)$ and the regressors $\boldsymbol{u}_{l,j}$ are assumed to be independent of each other for all $\{k,\, l,\, i,\, j\}$. All random processes are assumed to be zero mean. The above data model has been frequently used in the parameter estimation literature [19], and are useful in studies of various adaptive filters [38].

The objective of the network is to estimate ω^o in a distributed and iterative way subject to certain energy constraints. During the iterative estimation process, the energy cost of node k per iteration consists of sensing cost, computing cost and communication cost (incurred to disseminate or relay intermediate estimates to the physical neighbors of a node[4]). While the sensing and computing costs are almost the same for all nodes, the communication cost depends on the information that is disseminated or relayed by a node in every iteration and forms the major cost incurred in the estimation process. For simplicity, we ignore the sensing and computing costs and use the terms "energy cost" and "communication cost" interchangeably throughout the paper. Denote the communication cost per iteration of a node k as c_k^{cm}. The nodes estimate ω^o by solving a constrained least mean-squares (LMS) problem:

$$\begin{aligned} \text{(P0)} \quad & \min_{\omega} \sum_{k\in\mathcal{N}} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\omega|^2 \\ & \text{s.t., } c_k^{cm} \leq c_k,\ \forall k \in \mathcal{N}, \\ & \sum_{k\in\mathcal{N}} c_k^{cm} \leq c, \end{aligned} \tag{1}$$

where c_k and c are the node- and network-wide energy budgets imposed in each iteration, respectively.

The ATC diffusion strategy solves (P0) without the energy constraints by using the following update equations [18], [19]:

$$\begin{aligned} \boldsymbol{\psi}_{k,i} &= \boldsymbol{\omega}_{k,i-1} + \mu_k\boldsymbol{u}_{k,i}^*[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\omega}_{k,i-1}], \\ \boldsymbol{\omega}_{k,i} &= \sum_{l\in\mathcal{N}_{k\leftarrow}} a_{lk}\boldsymbol{\psi}_{l,i}, \end{aligned} \tag{2}$$

[3] An undirected graph is treated as a directed graph by replacing each undirected edge with two edges of opposite directions.

[4] We assume that the energy cost of receiving an estimate is negligible compared to that of transmitting an estimate.

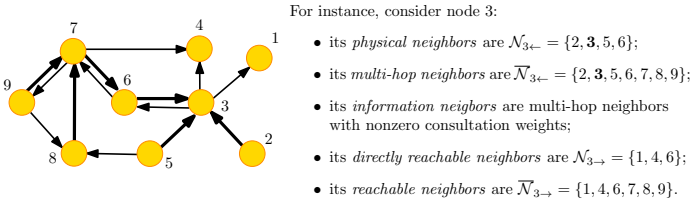


Fig. 2: The different types of neighbors of a node.

where μ_k is a positive step-size parameter, and a_{lk} are combination weights satisfying

$$a_{lk} \geq 0,\ A^T \mathbf{1}_N = \mathbf{1}_N,\ \text{and}\ a_{lk} = 0\ \text{if}\ l \notin \mathcal{N}_{k\leftarrow}.$$

Here, $A = (a_{lk})_{N\times N}$ is the combination weight matrix. The strategy consists of two steps, the *adaptation* step and the *consultation* (also known as the combination or diffusion) step. In the adaptation step, each node adapts its local estimate to an intermediate estimate $\boldsymbol{\psi}_{k,i}$ by using the new data available, and the consultation step combines the intermediate estimates from the physical neighborhood of a node through a weighted sum to obtain a local estimate $\boldsymbol{\omega}_{k,i}$ for the current iteration. In this paper, we consider only the ATC form of diffusion since it outperforms other alternative diffusion strategies under mild technical conditions [20].

The ATC strategy however does not have the flexibility to take into account the *hard* energy constraints in (1), because for a given network, the ATC strategy invokes a fixed communication cost at every node in each iteration. Although an ATC variant proposed in [26] uses controlled probabilistic on/off links at each node to satisfy an average (and hence *soft*) energy constraint in each iteration, it does not handle hard energy budget constraints. This motivates us to consider a flexible diffusion strategy, which allows multi-hop consultations under predefined energy budgets.

III. Multi-hop Diffusion Adaptation

In this section, we extend the ATC strategy by allowing a node to consult any node in its multi-hop neighborhood. The resulting mATC strategy uses the following update equations,

$$\begin{aligned}\boldsymbol{\psi}_{k,i} &= \boldsymbol{\omega}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^*[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\omega}_{k,i-1}],\\ \boldsymbol{\omega}_{k,i} &= \sum_{l\in\overline{\mathcal{N}}_{k\leftarrow}} a_{lk}\boldsymbol{\psi}_{l,i},\end{aligned} \tag{3}$$

where the combination weights satisfy

$$a_{lk} \geq 0,\ A^T \mathbf{1}_N = \mathbf{1}_N,\ \text{and}\ a_{lk} = 0\ \text{if}\ l \notin \overline{\mathcal{N}}_{k\leftarrow}. \tag{4}$$

The only difference between mATC and ATC is in the combination step: the node k consults its multi-hop neighbors $\overline{\mathcal{N}}_{k\leftarrow}$, which include the physical neighbors $\mathcal{N}_{k\leftarrow}$ as a subset. If $a_{lk} > 0$, we say that node l is an *information neighbor* of node k (cf. Fig. 2 for an illustration).

This simple modification to the ATC strategy generalizes the diffusion concept to cover centralized estimation at one extreme, and non-cooperative estimation at the other extreme. This unifies the centralized, non-cooperative, and distributed strategies into a single framework, which allows us to study the trade-offs amongst them easily.

The mATC strategy inherits all stability and performance results of the ATC strategy because the generalization introduced in the combination matrix A does not affect the analysis. Specifically, the network estimation is mean stable for any choice of A *if and only if* $\mu_k < \frac{2}{\lambda_{\max}(R_{u,k})}$. The same condition holds for mean-square stability if the step sizes $\{\mu_k\}_{k\in\mathcal{N}}$ are sufficiently small. Interested readers are referred to [18]–[20] for proofs of these stability results. Here we only summarize the mean-square performance results of the mATC strategy as these will be used in the sequel.

Denote the estimation error vector of an arbitrary node k at iteration i as $\tilde{\boldsymbol{\omega}}_{k,i} \triangleq \omega^o - \boldsymbol{\omega}_{k,i}$. Collect all error vectors and step-sizes across the network into a block vector and block matrix in

$$\begin{aligned}\tilde{\boldsymbol{\omega}}_i &\triangleq \text{col}\{\tilde{\boldsymbol{\omega}}_{1,i}, \tilde{\boldsymbol{\omega}}_{2,i}, ..., \tilde{\boldsymbol{\omega}}_{N,i}\},\\ \mathcal{M} &\triangleq \text{diag}\{\mu_1 I_M, \mu_2 I_M, ..., \mu_N I_M\},\end{aligned}$$

and let $\mathcal{A} \triangleq A \otimes I_M$. We further define the block diagonal matrix $\mathcal{R}$ and the $N \times N$ block matrix $\mathcal{B}$ with blocks of size $M \times M$ each, as follows:

$$\begin{aligned}\mathcal{R} &\triangleq \mathbb{E}\text{diag}\{\boldsymbol{u}_{1,i}^*\boldsymbol{u}_{1,i}, \boldsymbol{u}_{2,i}^*\boldsymbol{u}_{2,i}, ..., \boldsymbol{u}_{N,i}^*\boldsymbol{u}_{N,i}\},\\ \mathcal{B} &\triangleq \mathcal{A}^T(I_{NM} - \mathcal{M}\mathcal{R}).\end{aligned}$$

Then, the mean network error evolves as $\mathbb{E}\tilde{\boldsymbol{\omega}}_i = \mathcal{B}\mathbb{E}\tilde{\boldsymbol{\omega}}_{i-1}$. For any Hermitian nonnegative-definite weighting matrix Σ, we have the following approximation up to first order in μ_k:

$$\begin{aligned}\mathbb{E}\,\|\tilde{\boldsymbol{\omega}}_i\|_\Sigma^2 &\approx \text{Tr}\left(\mathcal{B}^{*i+1}\Sigma\mathcal{B}^{i+1}\Omega_{-1}\right) + \sum_{j=0}^{i}\text{Tr}\left(\mathcal{B}^{*j}\Sigma\mathcal{B}^j\mathcal{Y}\right)\\ &= \mathbb{E}\,\|\tilde{\boldsymbol{\omega}}_{i-1}\|_\Sigma^2 + \text{Tr}\left(\mathcal{B}^{*i}\Sigma\mathcal{B}^i\mathcal{Y}\right)\\ &\quad - \text{Tr}\left(\left(\mathcal{B}^{*i}\Sigma\mathcal{B}^i - \mathcal{B}^{*i+1}\Sigma\mathcal{B}^{i+1}\right)\Omega_{-1}\right),\end{aligned} \tag{5}$$

where $\|\tilde{\boldsymbol{\omega}}_i\|_\Sigma^2 \triangleq \tilde{\boldsymbol{\omega}}_i^*\Sigma\tilde{\boldsymbol{\omega}}_i$, $\Omega_{-1} \triangleq \mathbb{E}\tilde{\boldsymbol{\omega}}_{-1}\tilde{\boldsymbol{\omega}}_{-1}^*$ with $\tilde{\boldsymbol{\omega}}_{-1}$ being the initial estimation error, and

$$\begin{aligned}\mathcal{Y} &\triangleq \mathcal{A}^T\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A},\\ \mathcal{S} &\triangleq \text{diag}\{\sigma_{v,1}^2 R_{u,1}, \sigma_{v,2}^2 R_{u,2}, ..., \sigma_{v,N}^2 R_{u,N}\}.\end{aligned}$$

The recursive relation (5) can be used to compute the theoretical transient and steady-state network MSDs.

By specifying Σ as $\frac{1}{N}I_{NM}$, the above variance $\mathbb{E}\,\|\tilde{\boldsymbol{\omega}}_i\|_\Sigma^2$ gives the MSD of the network estimate $\boldsymbol{\omega}_i$, which is an average MSD across the network at the ith iteration, i.e., $\text{MSD}_i \triangleq \frac{1}{N}\sum_{k\in\mathcal{N}}\mathbb{E}\,\|\tilde{\boldsymbol{\omega}}_{k,i}\|^2$. In particular, as $i \to \infty$ the steady-state network MSD is obtained from (5) as

$$\text{MSD}_\infty \approx \frac{1}{N}\sum_{j=0}^{\infty}\text{Tr}\left(\mathcal{B}^j\mathcal{Y}\mathcal{B}^{*j}\right). \tag{6}$$

The node and network MSDs are controlled by the quantities $\mathcal{B}$ and $\mathcal{Y}$, both of which are dependent on the combination matrix A. Selection of the combinations weights $a_{l,k}$ can be done in two steps:

Step 1: Given an arbitrary set of information neighbors for each node to consult, we derive analytical forms of the combination weights that optimize the network performance.

Step 2: Given the analytical combination weights derived in Step 1, we optimize the information neighbor set to be consulted by each node such that the network MSD is minimized subject to predefined energy budget constraints.

The two steps together determine which nodes are consulted by each node and to what extent it is weighted if a consultation happens. The next section presents a combination rule for determining the weights in Step 1, while we discuss Step 2 in Section V.

IV. Selecting the Combination Weights

In this section, we aim to select the combination weight matrix A to optimize the steady-state network MSD in (6), given arbitrary information neighbors of each node in the network. The optimization, however, does not admit an analytical solution and has to be solved numerically in general, which prevents finding an adaptive solution under varying network conditions. To keep the adaptation ability of the network, we make a compromise by seeking for an analytical solution that approximately minimizes an *upper bound* of the steady-state network MSD, given by

$$\begin{aligned}\mathrm{MSD}_\infty &\approx \frac{1}{N}\sum_{j=0}^{\infty}\mathrm{Tr}\left(\mathcal{B}^j\mathcal{Y}\mathcal{B}^{*j}\right) \le \frac{\lambda_{\max}(\mathcal{Y})}{N}\sum_{j=0}^{\infty}\mathrm{Tr}\left(\mathcal{B}^j\mathcal{B}^{*j}\right)\\ &\le M\lambda_{\max}(\mathcal{Y})\sum_{j=0}^{\infty}\lambda_{\max}^j(\mathcal{B}\mathcal{B}^*) = \frac{M\lambda_{\max}(\mathcal{Y})}{1-\lambda_{\max}(\mathcal{B}\mathcal{B}^*)} \triangleq \overline{\mathrm{MSD}}_\infty.\end{aligned}$$

The first inequality above uses the positive semi-definiteness of the matrix $\mathcal{Y}$ and the last equality is due to the fact that $\lambda_{\max}(\mathcal{B}\mathcal{B}^*) < 1$ is necessary to ensure mean and mean-square stability [18]–[20].

The upper bound $\overline{\mathrm{MSD}}_\infty$ can be minimized by minimizing an auxiliary variable β so that $\overline{\mathrm{MSD}}_\infty \le \beta$. This inequality is equivalent to $\beta \ge 0$ and the eigenvalue constraint $\lambda_{\max}(\mathcal{Y})/\beta + \lambda_{\max}(\mathcal{B}\mathcal{B}^*) \le 1$. This approach however does not admit a closed-form solution. To obtain an explicit solution for the combination matrix A, we approximate and decompose the equivalent problem into two subproblems: Firstly, we solve for an approximate solution of β by strengthening the eigenvalue constraint into $\lambda_{\max}\left(\mathcal{Y}/\beta + \mathcal{B}\mathcal{B}^*\right) \le 1$. We can solve for an approximate solution of β using the following semi-definite program (SDP), which is derived in Appendix A:

$$\begin{aligned}&\beta^o = \arg\min_{\beta}\beta \quad \text{s.t.}\\ &\begin{bmatrix}\beta(\mathcal{M}\mathcal{S}\mathcal{M})^{-1}+\tilde{Q} & \tilde{Q}(\mathbf{1}_N\otimes I_M)\\ (\mathbf{1}_N^T\otimes I_M)\tilde{Q} & (\mathbf{1}_N^T\otimes I_M)[\tilde{Q}-I_{NM}](\mathbf{1}_N\otimes I_M)\end{bmatrix}\succeq 0,\end{aligned} \tag{7}$$

where $\tilde{Q} \triangleq (I_{NM}-\mathcal{M}\mathcal{R})^{-2} \succ 0$, which is independent of the combination weight matrix A to be optimized. The SDP is convex and hence readily solvable by standard SDP solvers.

Secondly, given the solution of β, we derive an analytical solution of the weight matrix A by minimizing $\mathrm{Tr}(\mathcal{Y})/\beta + \mathrm{Tr}(\mathcal{B}\mathcal{B}^*)$, which is an upper bound of the original eigenvalue constraint. This leads to the optimization problem in (8).

Theorem 1. *Suppose that the information neighbor set for each node $k \in \mathcal{N}$ is $\overline{\mathcal{N}}'_{k\leftarrow} \subseteq \overline{\mathcal{N}}_{k\leftarrow}$, β^o is the solution of the SDP (7), and $\alpha^o \triangleq (\beta^o+1)^{-1}$. The combination weights that solve the following optimization problem*

$$\begin{aligned}&\min_A\ \alpha^o\mathrm{Tr}(\mathcal{Y}) + (1-\alpha^o)\mathrm{Tr}(\mathcal{B}\mathcal{B}^*)\\ &\text{s.t. } a_{lk}\ge 0,\ A^T\mathbf{1}_N = \mathbf{1}_N,\ \text{and } a_{lk}=0 \text{ if } l\notin\overline{\mathcal{N}}'_{k\leftarrow}\end{aligned} \tag{8}$$

are given as follows:

$$a_{lk} = \begin{cases}\dfrac{\gamma_l^{-2}}{\sum_{j\in\overline{\mathcal{N}}'_{k\leftarrow}}\gamma_j^{-2}}, & \text{if } l\in\overline{\mathcal{N}}'_{k\leftarrow},\\ 0, & \text{otherwise},\end{cases} \tag{9}$$

where the composite variance γ_l^2 is defined by

$$\gamma_l^2 \triangleq \alpha^o\mu_l^2\cdot\sigma_{v,l}^2\cdot\mathrm{Tr}(R_{u,l}) + (1-\alpha^o)\mathrm{Tr}\left((I_M-\mu_lR_{u,l})^2\right). \tag{10}$$

Proof: Substituting the expression of $\mathcal{Y}$ into the objective function of problem (8), we have

$$\begin{aligned}&\alpha^o\mathrm{Tr}(\mathcal{Y}) + (1-\alpha^o)\mathrm{Tr}(\mathcal{B}\mathcal{B}^*)\\ &= \alpha^o\sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}}a_{lk}^2\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l})\\ &\quad + (1-\alpha^o)\sum_{k\in\mathcal{N}}\sum_{l\in\mathcal{N}}a_{lk}^2\mathrm{Tr}\left((I_M-\mu_lR_{u,l})^2\right).\end{aligned}$$

Therefore problem (8) can be decoupled into N separate optimization problems of the form:

$$\begin{aligned}&\min_A\ \sum_{l\in\mathcal{N}}a_{lk}^2\left[\alpha^o\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l}) + (1-\alpha^o)\mathrm{Tr}\left((I_M-\mu_lR_{u,l})^2\right)\right]\\ &\text{s.t. } a_{lk}\ge 0,\ A^T\mathbf{1}_N=\mathbf{1}_N,\ \text{and } a_{lk}=0 \text{ if } l\notin\overline{\mathcal{N}}'_{k\leftarrow},\end{aligned}$$

from which (9) follows, and the proof is complete. ∎

We call the closed-form solution of the combination weights given in (9) as the "balancing rule", since it optimizes a trade-off between the diffusion convergence rate (measured through $\mathrm{Tr}(\mathcal{B}\mathcal{B}^*)$), and the steady-state network MSD (measured through $\mathrm{Tr}(\mathcal{Y})$). This is further explained as follows.

We observe that the steady-state network MSD can be further upper bounded by

$$\mathrm{MSD}_\infty \le \begin{cases}\dfrac{M\lambda_{\max}(\mathcal{Y})}{1-r_1^2} \le \dfrac{M\mathrm{Tr}(\mathcal{Y})}{1-r_1^2} \triangleq \overline{\mathrm{MSD}}_\infty^a\\ \dfrac{Mr_2}{1-\lambda_{\max}(\mathcal{B}\mathcal{B}^*)} \le \dfrac{Mr_2}{1-\mathrm{Tr}(\mathcal{B}\mathcal{B}^*)} \triangleq \overline{\mathrm{MSD}}_\infty^b\end{cases} \tag{11}$$

where the two positive scalars r_1 and r_2 are given by

$$r_1 \triangleq \rho(I_{NM}-\mathcal{M}\mathcal{R}),\ r_2 \triangleq \lambda_{\max}(\mathcal{M}\mathcal{S}\mathcal{M}).$$

These two upper bounds can be shown to be minimized by $\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l})$ and $\mathrm{Tr}\left((I_M-\mu_lR_{u,l})^2\right)$, respectively. Therefore the composite variance γ_l^2 is a summation of the minimizers of the two looser upper bounds in (11). Observing that $\mathrm{Tr}(\mathcal{B}\mathcal{B}^*)$ is an upper bound of $\lambda_{\max}(\mathcal{B}\mathcal{B}^*)$, we can alternatively interpret the term $\mathrm{Tr}\left((I_M-\mu_lR_{u,l})^2\right)$ as an approximate minimizer of the transient network MSD. The balancing rule can then be interpreted as balancing between minimizing the steady-state and the transient network MSDs, with the balance tuned by varying the coefficient $\alpha^o \in [0,1]$.

To apply the balancing rule (9), each node k needs to know the composite variances $\{\gamma_l^2\}_{l\in\overline{\mathcal{N}}'_{k\leftarrow}}$, which depend on

the two components $\{\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l}),\ \mathrm{Tr}\left((I_M-\mu_l R_{u,l})^2\right)\}$ of each selected information neighbors. Without knowing them a priori, each node needs to gather their estimates from its information neighbors and use them to update the combination rule adaptively. The desired estimates can be obtained as moving averages of their realizations using real-time data (a similar method was used in [39] to obtain the adaptive relative-variance rule):

$$\begin{aligned}\hat{\boldsymbol{\gamma}}_{l,1}^2(i)&=(1-\nu_l)\hat{\boldsymbol{\gamma}}_{l,1}^2(i-1)+\nu_l\left\|\boldsymbol{\psi}_{l,i}-\boldsymbol{\omega}_{l,i-1}\right\|^2,\\ \hat{\boldsymbol{R}}_{u,l}(i)&=(1-\nu_l)\hat{\boldsymbol{R}}_{u,l}(i-1)+\nu_l\boldsymbol{u}_{l,i}^*\boldsymbol{u}_{l,i},\\ \hat{\boldsymbol{\gamma}}_{l,2}^2(i)&=\mathrm{Tr}\left((I_M-\mu_l\hat{\boldsymbol{R}}_{u,l}(i))^2\right),\\ \hat{\boldsymbol{\gamma}}_{l}^2(i)&=\hat{\alpha}^o\hat{\boldsymbol{\gamma}}_{l,1}^2(i)+(1-\hat{\alpha}^o)\hat{\boldsymbol{\gamma}}_{l,2}^2(i),\end{aligned}\tag{12}$$

where the symbol $\hat{}$ indicates an adaptive estimate and $\hat{\boldsymbol{\gamma}}_{l,1}^2$ and $\hat{\boldsymbol{\gamma}}_{l,2}^2$ are the estimates of the aforementioned two components of $\hat{\boldsymbol{\gamma}}_{l}^2$. The quantity $\nu_l\in(0,1)$ is a chosen discount factor, and $\hat{\alpha}^o\in[0,1]$ is a balancing coefficient usually chosen close to one. The balancing coefficient can also be obtained through the SDP in Theorem 1 by replacing the noise variances with empirical estimates. This however requires sending the empirical estimates to a central processor, which can be costly for the network, and is therefore done only infrequently. By the adaptation equation in (3), we can verify that $\mathbb{E}\left\|\boldsymbol{\psi}_{l,i}-\boldsymbol{\omega}_{l,i-1}\right\|^2=\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l})$, and we arrive at an adaptive implementation of the balancing rule:

$$\boldsymbol{a}_{lk}(i)=\frac{\hat{\boldsymbol{\gamma}}_l^{-2}(i)}{\sum_{j\in\overline{\mathcal{N}}'_{k\leftarrow}}\hat{\boldsymbol{\gamma}}_j^{-2}(i)}.\tag{13}$$

Note that the estimates $(\boldsymbol{\psi}_l(i),\hat{\boldsymbol{\gamma}}_l^2(i))$ of each information neighbor are transmitted to node k before it applies the rule.

Remark 1. The balancing rule reduces to the relative-variance rule of [39] if the coefficient α^o is set to 1. This implies that the relative-variance rule minimizes the bound $\overline{\mathrm{MSD}}_\infty^a$. The balancing rule is also related to the two-phase rules proposed in [40], where separate combination rules adopted for the transient and the steady-state phases approximately minimize $\lambda_{\max}(\mathcal{B}\mathcal{B}^*)$ and $\mathrm{Tr}(\mathcal{Y})$, respectively. In that case, a switching point between the two-phase rules needs to be estimated online by, e.g., using the technique developed in [41].

V. Selecting the Information Neighbors

Given the closed-form combination weights derived in the last section, we now proceed to optimize the information neighbor set of each node so that the upper bound of the steady-state network MSD, given as $\alpha^o\overline{\mathrm{MSD}}_\infty^a+(1-\alpha^o)\overline{\mathrm{MSD}}_\infty^b$, is minimized under predefined energy constraints. With the combination weights given in (9), the cost function in (8) becomes an explicit function of the composite variances $\{\gamma_k\}_{k\in\mathcal{N}}$, which can be further optimized by selecting appropriate information neighbors for each node under the energy budget constraints. This yields, after some algebraic manipulations, the following optimization problem:

$$\begin{aligned}(\mathrm{P1})\quad &\min_{\{\overline{\mathcal{N}}'_{k\leftarrow}:\overline{\mathcal{N}}'_{k\leftarrow}\subseteq\overline{\mathcal{N}}_{k\leftarrow}\}_{k\in\mathcal{N}}}\ \sum_{k\in\mathcal{N}}\frac{1}{\sum_{l\in\overline{\mathcal{N}}'_{k\leftarrow}}\gamma_l^{-2}}\\ &\text{s.t. } c_k^{cm}(\overline{\mathcal{N}}_{k\leftarrow},\{\overline{\mathcal{N}}'_{l\leftarrow}\}_{l\in\overline{\mathcal{N}}_{k\rightarrow}})\le c_k,\ \forall k\in\mathcal{N},\\ &\sum_{k\in\mathcal{N}}c_k^{cm}(\overline{\mathcal{N}}_{k\leftarrow},\{\overline{\mathcal{N}}'_{l\leftarrow}\}_{l\in\overline{\mathcal{N}}_{k\rightarrow}})\le c,\end{aligned}$$

where $c_k^{cm}(\overline{\mathcal{N}}_{k\leftarrow},\{\overline{\mathcal{N}}'_{l\leftarrow}\}_{l\in\overline{\mathcal{N}}_{k\rightarrow}})$ indicates that the communication cost of node k in one iteration depends on its multi-hop neighbors and reachable neighbors since it may be required to relay estimates from its multi-hop neighbors to its reachable neighbors.

Problem (P1) is intractable in its current form because of the unknown information neighbor sets $\{\overline{\mathcal{N}}'_{k\leftarrow}\}_{k\in\mathcal{N}}$ and the implicit communication costs $c_k^{cm}(\overline{\mathcal{N}}_{k\leftarrow},\{\overline{\mathcal{N}}'_{l\leftarrow}\}_{l\in\overline{\mathcal{N}}_{k\rightarrow}})$ for all $k\in\mathcal{N}$. To obtain a tractable form of (P1), we first introduce binary variables to represent the sets $\{\overline{\mathcal{N}}'_{k\leftarrow}\}_{k\in\mathcal{N}}$, and then model the in-network communications to get an explicit expression for $c_k^{cm}(\overline{\mathcal{N}}_{k\leftarrow},\{\overline{\mathcal{N}}'_{l\leftarrow}\}_{l\in\overline{\mathcal{N}}_{k\rightarrow}})$. This turns out to be very complex if the network has an arbitrary topology where there are multiple paths from one node to another. To reduce the complexity and make (P1) tractable, we consider two special cases separately:

- Case 1 (Simple topology): For any pair of nodes, there is at most one directed simple path connecting them.
- Case 2 (Two-hop consultations): The network has an arbitrary topology but the information neighbors of every node are restricted to be within two hops away.

In the sequel, we derive an explicit form of (P1) for each of the two cases as an MILP, which is solved offline before running mATC on the network. We also show that the general problem (P1) admits an approximate and distributed solution that can be obtained online if only local energy budget is imposed on each node.

A. Explicit formulation as an MILP

In this subsection, we introduce binary and auxiliary continuous decision variables to reformulate problem (P1) into an MILP that is solvable using standard solvers.

Before reformulating (P1), we first derive an explicit expression for the communication cost $c_k^{cm}(\overline{\mathcal{N}}_{k\leftarrow},\{\overline{\mathcal{N}}'_{l\leftarrow}\}_{l\in\overline{\mathcal{N}}_{k\rightarrow}})$ by introducing two classes of binary variables. The first class of binary variables are the selection variables $\delta_{lk}\in\{0,1\}$, for all $l\in\overline{\mathcal{N}}_{k\leftarrow}$ and $k\in\mathcal{N}$, where $\delta_{lk}=1$ if and only if node l is selected to be an information neighbor of node k. We note that $a_{lk}=0$ if and only if $\delta_{lk}=0$. The second class of binary variables are the relay variables $\pi_{lk}\in\{0,1\}$, for all $l\in\overline{\mathcal{N}}_{k\leftarrow}$ and $k\in\mathcal{N}$, where $\pi_{lk}=1$ if and only if node k relays the information originating from node l. We have $\pi_{kk}=1$ if and only if node k broadcasts its own intermediate estimate. We make the following assumptions.

Assumption V.1. *Every broadcast conveys information from a single node, and incurs a communication cost (which may be different for different nodes). All nodes having the broadcast*

node as a physical neighbor receives the information being broadcast.

Assumption V.2. *At every iteration, each node relays the same piece of information at most once.*

With the above two assumptions, the communication cost, $c_k^{cm}(\overline{\mathcal{N}}_{k\leftarrow}, \{\overline{\mathcal{N}}'_{l\leftarrow}\}_{l\in\overline{\mathcal{N}}_{k\rightarrow}})$, of node k in a single iteration is equal to the number of intermediate estimates it needs to relay and diffuse, multiplied by the energy cost incurred in each broadcast, i.e.,

$$c_k^{cm}(\overline{\mathcal{N}}_{k\leftarrow}, \{\overline{\mathcal{N}}'_{l\leftarrow}\}_{l\in\overline{\mathcal{N}}_{k\rightarrow}}) = c_k^{cm,0} \sum_{l\in\overline{\mathcal{N}}_{k\leftarrow}} \pi_{lk}, \forall k \in \mathcal{N}, \tag{14}$$

where $c_k^{cm,0}$ is the constant cost per broadcast by node k. Note that π_{lk} is a function of $\{\overline{\mathcal{N}}'_{j\leftarrow} : j \in \overline{\mathcal{N}}_{k\rightarrow} \cap \overline{\mathcal{N}}_{l\rightarrow}\}$.

We now investigate the relationships amongst the relay variables π_{lk} and the selection variables δ_{lj} and then reformulate (P1) into an MILP, under the two special cases alluded to above.

1) Case 1 (Simple topology): In this case, the unique directed path from a node l to a reachable neighbor j is characterized by a set of binary constants, $\{\eta_{lj,k}\}_{k\in\overline{\mathcal{N}}_{j\leftarrow}\setminus\{j\}}$, where $\eta_{lj,k} = 1$ if and only if node k is on the path from node l to node j. Then, the selection variables δ_{lj} and the relay variables π_{lk} are related to each other as follows:

$$\pi_{lk} = \min\left\{1, \sum_{j\in\overline{\mathcal{N}}_{l\rightarrow}\setminus\{k\}} \eta_{lj,k}\delta_{lj}\right\}, \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N}, \tag{15}$$

which implies that node k relays node l's information if and only if node l is consulted by some node j, and node k ($\neq j$) is on the directed path from node l to node j. The bound by 1 in the relation (15) is due to Assumption V.2, in which we assume that node l's estimate is relayed at most once by node k in each iteration. The above relation (15) can be further written in the linear form (23) ahead.

The objective function of (P1) can be transformed into a linear function by introducing a couple of linear constraints. Using the selection variables δ_{lk}, we express the objective function equivalently as $\sum_{k\in\mathcal{N}} \left(\sum_{l\in\overline{\mathcal{N}}_{k\leftarrow}} \delta_{lk}\gamma_l^{-2}\right)^{-1}$, where the candidate information neighbor set $\overline{\mathcal{N}}'_{k\leftarrow}$ is replaced by the multi-hop neighbor set $\overline{\mathcal{N}}_{k\leftarrow}$ with the help of selection variables. We then introduce auxiliary optimization variables

$$z_k = \left(\sum_{l\in\overline{\mathcal{N}}_{k\leftarrow}} \delta_{lk}\gamma_l^{-2}\right)^{-1}, \text{ and } p_{lk} = z_k\delta_{lk}, \tag{16}$$

which allows us to rewrite the objective as $\sum_{k\in\mathcal{N}} z_k$, and to perform McCormick linearization [42] on the bilinear constraints $p_{lk} = z_k\delta_{lk}$ without loss of optimality.

Consequently, (P1) is equivalent to an MILP problem defined in (17)-(29), which is called (P2) hereafter. The data and variables of problem (P2) are referred to Table I, which also contains those used in the later problem (P3) for Case 2. Constraints (18)-(22) arise from the linearization of the nonlinear objective of (P1). Constraints (23) describe the relation between the relay and the selection variables. Constraints (24) and (25) characterize the energy costs and their budgets for the distributed estimation in a single iteration.

TABLE I: Symbols used in formulating (P2) and (P3)

Problem data	
$\mathcal{N}$	full node set
$\mathcal{N}_{k\leftarrow}$	the set of all physical (i.e., one-hop) neighbors of node k
$\overline{\mathcal{N}}_{k\leftarrow}$	the set of all multi-hop neighbors of node k
$\overline{\mathcal{N}}^2_{k\leftarrow}$	the set of all two-hop neighbors of node k
$\mathcal{N}_{k\rightarrow}$	the set of all directly reachable neighbors of node k
$\overline{\mathcal{N}}_{k\rightarrow}$	the set of all reachable neighbors of node k
$\mathcal{P}_k$	the set of relay customers of node k
$\mathcal{P}^k$	the set of relay servers of node k
$\sigma_{v,k}$	the variance of measurement noise, equal to $\mathbb{E}\boldsymbol{v}_k^*(i)\boldsymbol{v}_k(i)$
$R_{u,k}$	the input covariance matrix, equal to $\mathbb{E}\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}$
μ_k	the step size parameter of node k implementing mATC in (3)
γ_k	the composite variance of node k defined in (10)
$\eta_{lj,k}$	binary indicator of whether node k is on the transmission path from node l to node j
$\underline{z}_k$	lower bound of z_k computed as $(\sum_{l\in\overline{\mathcal{N}}_{k\leftarrow}} \gamma_l^{-2})^{-1}$
$\bar{z}_k$	upper bound of z_k computed as $(\min_{l\in\overline{\mathcal{N}}_{k\leftarrow}} \gamma_l^{-2})^{-1}$
$c_k^{cm,0}$	the constant energy cost per broadcast by node k
c_k	the energy budget available per iteration for node k
c	the energy budget available per iteration for the network
Problem variables	
δ_{lk}	binary indicator of whether l is consulted by node k
π_{lk}	binary indicator of whether l's info is relayed by node k
p_{lk}, z_k	real auxiliary variables, see (16)

$$\text{(P2)} \ \min \sum_{k\in\mathcal{N}} z_k \tag{17}$$

$$\text{s.t.} \sum_{l\in\overline{\mathcal{N}}_{k\leftarrow}} \gamma_l^{-2} p_{lk} = 1, \ \forall k \in \mathcal{N} \tag{18}$$

$$p_{lk} \le \bar{z}_k\delta_{lk}, \ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N} \tag{19}$$

$$p_{lk} \ge \underline{z}_k\delta_{lk}, \ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N} \tag{20}$$

$$p_{lk} \ge z_k + \bar{z}_k(\delta_{lk} - 1), \ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N} \tag{21}$$

$$p_{lk} \le z_k + \underline{z}_k(\delta_{lk} - 1), \ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N} \tag{22}$$

$$\pi_{lk} \le \sum_{j\in\overline{\mathcal{N}}_{l\rightarrow}\setminus\{k\}} \eta_{lj,k}\delta_{lj} \le |\overline{\mathcal{N}}_{l\rightarrow}|\pi_{lk}, \ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N} \tag{23}$$

$$\sum_{l\in\overline{\mathcal{N}}_{k\leftarrow}} c_k^{cm,0}\pi_{lk} \le c_k, \ \forall k \in \mathcal{N} \tag{24}$$

$$\sum_{k\in\mathcal{N}} \sum_{l\in\overline{\mathcal{N}}_{k\leftarrow}} c_k^{cm,0}\pi_{lk} \le c \tag{25}$$

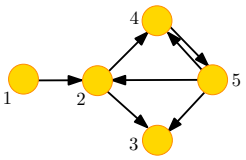
$$\delta_{lk} \in \{0, 1\}, \ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N} \tag{26}$$

$$\pi_{lk} \in \{0, 1\}, \ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N} \tag{27}$$

$$p_{lk} \in \mathbb{R}_{\ge 0}, \ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N} \tag{28}$$

$$z_k \in \mathbb{R}_{\ge 0}, \ \forall k \in \mathcal{N}. \tag{29}$$

Note that problem (P2) is solved offline before the mATC diffusion procedure. It requires a centralized processor to have prior knowledge of the network topology, and data and noise variance profiles of every node, which restricts the frequency that this optimization can be performed. This restriction is however alleviated to some extent by the following result.



For instance, consider node 2:

- its *relay customers* are $\mathcal{P}_2 = \{1, \mathbf{2}\} \subset \mathcal{N}_{2\leftarrow} = \{1, \mathbf{2}, 5\}$;
- its *relay servers* are $\mathcal{P}^2 = \{4\} \subset \mathcal{N}_{2\rightarrow} = \{3, 4\}$.

Fig. 3: The relay customer and server sets of a node. Node 5 is not a relay customer of node 2 because it can reach all its reachable neighbors without node 2's help. Node 3 is not a relay server of node 2 because none of its reachable neighbors needs node 3's help to obtain information from node 2.

Lemma 1. *The optimal solution* $\{(\delta_{lk}, \pi_{lk}),\ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N}\}$ *for (P2) is invariant to a homogeneous scaling of the composite variances* γ_l *for all* $l \in \mathcal{N}$.

Proof: Let the optimal solution of (P2) be $\{(\delta^\star_{lk}, \pi^\star_{lk}, p^\star_{lk}, z^\star_k),\ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N}\}$. If γ_l, for every $l \in \mathcal{N}$, is scaled by a constant α to $\alpha\gamma_l$, then it is easy to verify that the optimal solution becomes $\{(\delta^\star_{lk}, \pi^\star_{lk}, \alpha^2 p^\star_{lk}, \alpha^2 z^\star_k),\ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N}\}$, so that $\{(\delta_{lk}, \pi_{lk}),\ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N}\}$ remains unchanged. The lemma is now proved. ■

Lemma 1 shows that if changes in the data and noise variances happen uniformly over all network nodes, then the optimal information neighbor configuration remains unchanged. In general, the offline or centralized optimization in (P2) is performed infrequently, and is based on historical estimates of the data and noise variance profiles maintained by every node in the network.

2) Case 2 (Two-hop consultations): In this case, we do not make any assumptions about the network topology but restrict the selection of information neighbors to amongst the multi-hop neighbors that are at most two hops away from each node. We denote the set of neighbors of node k within two hops away as $\overline{\mathcal{N}}^2_{k\leftarrow}$.

With the above restriction, each node is only able to relay information originating from its physical neighbors. We use $\mathcal{P}_k$ to denote the set of physical neighbors of node k that have reachable neighbors who may need node k to relay information to. We call these the *relay customers* of node k. On the other hand, suppose that $k \to l \to m$ is a directed path so that node m is reachable but not directly reachable from node k. Then, we say that node l is a *relay server* of node k. Let $\mathcal{P}^k$ denote the set of relay servers of node k. We have

$$\mathcal{P}_k = \{k\} \cup \{l \in \mathcal{N}_{k\leftarrow} : \mathcal{N}_{k\rightarrow} \nsubseteq \mathcal{N}_{l\rightarrow}\} \subseteq \mathcal{N}_{k\leftarrow},$$
$$\mathcal{P}^k = \{l \in \mathcal{N}_{k\rightarrow} : \mathcal{N}_{l\rightarrow} \nsubseteq \mathcal{N}_{k\rightarrow}\} \subseteq \mathcal{N}_{k\rightarrow},$$

which are illustrated in Fig. 3. Then, it is sufficient to define the relay variables $\pi_{lk} \in \{0, 1\}$ for all $l \in \mathcal{P}_k$, $k \in \mathcal{N}$, and the selection variables $\delta_{lk} \in \{0, 1\}$ for all $l \in \overline{\mathcal{N}}^2_{k\leftarrow}$, $k \in \mathcal{N}$.

Next we model the relations between these two sets of variables. We note that, if node k consults node $l \neq k$, then firstly node l must broadcast its information and secondly, at least one of its relay servers must relay the information to node k if it is two hops away from node k. These observations are represented mathematically by

$$\delta_{lk} \leq \pi_{ll},\ \forall l \in \overline{\mathcal{N}}^2_{k\leftarrow}\backslash\{k\},\ k \in \mathcal{N}, \tag{30}$$
$$\delta_{lk} \leq \sum_{j \in \mathcal{P}^l \cap \mathcal{N}_{k\leftarrow}\backslash\{k\}} \pi_{lj},\ \forall l \in \overline{\mathcal{N}}^2_{k\leftarrow}\backslash\mathcal{N}_{k\leftarrow},\ k \in \mathcal{N}. \tag{31}$$

The first inequality excludes $l = k$ because the inequality does not hold in that case. The second inequality is defined for nodes l and k that are two hops away from each other, because it is trivially true due to the first inequality if the two nodes are one hop away from each other. These two sets of inequalities give a complete description of the relations between the relay variables and the selection variables.

Consequently, the information neighbor selection problem (P1) can again be reformulated into an MILP in the form of (P2) with some changes: the constraints (23) are replaced with the new ones in (30)-(31), the information neighbor set $\overline{\mathcal{N}}_{k\leftarrow}$ is replaced with $\overline{\mathcal{N}}^2_{k\leftarrow}$ everywhere, and furthermore, the relay variables π_{lk} are defined only for all $l \in \mathcal{P}_k$, $k \in \mathcal{N}$. We name the new formulation as (P3), to distinguish it from (P2).

Lemma 1 similarly holds for the MILP (P3). We also remark that (P3) is applicable to a network having the simple topology assumed in (P2), in which case the constraints (23) (applied to the two-hop neighborhoods) can be treated as valid inequalities for (P3) to enhance the solution process.

3) Valid inequalities to enhance the MILPs: As combinatorial problems, problems (P2) and (P3) are NP-hard in general. A typical way to speed up the solution process is by exploiting valid inequalities (i.e., constraints that maintain the optimal solution), which reduce the search space and enhance the branch-and-cut algorithm used by MILP solvers [43], [44]. The following property of (P2) and (P3) at optimality is useful for deriving such valid inequalities.

Lemma 2. *Given a feasible solution of (P2) or (P3), a better solution can be obtained if more information neighbors can be included without violating the energy budgets* (24)-(25).

Proof: The conclusion is clear from the objective function of (P1), which is decreasing in the size of the information neighbor set for each node. Since the objectives of (P1) and (P2) (or (P3)) are equivalent, the conclusion follows immediately. ■

Corollary 1. *The following equalities and inequalities are valid for (P2) or (P3):*

$$\textit{(P2) and (P3):}\ \delta_{kk} = 1,\ \forall k \in \mathcal{N}, \tag{32}$$
$$\textit{(P2):}\ \pi_{lk} \leq \delta_{lj},\ \forall l \in \overline{\mathcal{N}}_{k\leftarrow}\backslash\{j\},\ k \in \mathcal{N}_{j\leftarrow},\ j \in \mathcal{N}, \tag{33}$$
$$\textit{(P3):}\ \pi_{lk} \leq \delta_{lj},\ \forall l \in \mathcal{P}_k\backslash\{j\},\ k \in \mathcal{N}_{j\leftarrow},\ j \in \mathcal{N}. \tag{34}$$

Proof: The equalities (32) are because for optimality, a node always consults itself, which is evident from Lemma 2. The inequalities (33) (or (34)) mean if a node relays a piece of information, then every node that has the node as a physical neighbor must use the information in order to achieve optimality. This is proved as follows. By Assumption V.1, if a node relays a piece of information, then every node that has the node as a physical neighbor will receive the information;

then by Lemma 2, every node receiving the information must use it to ensure optimality. This completes the proof. ■

An additional set of valid inequalities holds for (P2), which states that if node k relays information originating from node l ($\neq k$), then any predecessor of node k on the unique transmission path must relay the same information towards k. Mathematically this means the following:

Lemma 3. *The following inequalities hold for (P2):*

$$\pi_{lk} \leq \pi_{lj},\ \forall j \in \{i \in \overline{\mathcal{N}}_{k\leftarrow} : \eta_{lk,i} = 1\},\ l \in \overline{\mathcal{N}}_{k\leftarrow} \backslash \{k\},\ k \in \mathcal{N}. \tag{35}$$

Proof: We prove the lemma by contradiction. Let there be a pair of nodes (k, j) in the defined set with $1 = \pi_{lk} > \pi_{lj} = 0$. This implies that there is a simple path that transmits the information from node l to node k while not traversing node j. Since by the definition, there is another path that transmits the same information to node k while traversing node j, this contradicts with the topological condition in Case 1 that there is a unique transmission path from node l to node k. This completes the proof. ■

Within the two-hop neighborhoods, similar inequalities $\pi_{lk} \leq \pi_{ll},\ \forall l \in \mathcal{P}_k \backslash \{k\},\ k \in \mathcal{N}$, are valid for (P3). They are, however, implied by the valid inequalities (34) and the model inequalities (30)-(31), and hence not treated as independent valid inequalities for (P3).

The valid inequalities (32)-(35) are added to (P2) or (P3) to reduce the search space and hence speed up the solution process. However, the computational time may still be prohibitively long if the network has a large size and each node has many reachable neighbors. In that case, we provide in the following subsection, approximate solutions for (P2) and (P3) that can be obtained efficiently by solving linear programs.

B. Approximate solutions via relaxations

In this subsection, we present a linear programming (LP) approach to solving the MILPs (P2) and (P3) approximately. A typical way of finding an approximate solution for (P2) or (P3) (including all valid inequalities in Corollary 1 and Lemma 3)[5] is by relaxing the binary variables as continuous variables taking values in $[0, 1]$, and then solving the resulting LP. The continuous solution is then translated back into binaries via thresholding. The choice of the threshold needs to ensure that the binary solution obtained satisfies the *hard* energy budget constraints. In the following, we propose a procedure to determine the threshold iteratively for both (P2) and (P3).

A common procedure to translate the solutions of relaxed (P2) and (P3) is summarized in Algorithm 1. The procedure decreases the threshold gradually (from the maximal value of 1) in lines 4 to 10 until it cannot be smaller without violating the network-wide energy budget. Then, in lines 11 to 16, the relay variables are adjusted so that local node energy budgets are satisfied. The validity of Algorithm 1 is proven in the following theorem.

[5]It is useful to include all valid inequalities into the relaxed problem, because they maintain the structural information of the original problem which may otherwise be lost after relaxation.

Algorithm 1 An approximate LP solution for (P2) or (P3)

1: For each $k \in \mathcal{N}$, let $\mathcal{S}_k = \overline{\mathcal{N}}_{k\leftarrow}$ and $\mathcal{S}_k = \overline{\mathcal{N}}^2_{k\leftarrow}$ for (P2) and (P3), respectively.
2: Relax all binary variables in (P2) or (P3) to continuous variables in $[0, 1]$. Solve the resulting LP to obtain $\{(\tilde{\pi}_{lk}, \tilde{\delta}_{lk}) : l \in \mathcal{S}_k, k \in \mathcal{N}\}$ as the optimal relay and selection variables for the relaxed LP.
3: Initialize the threshold $t_{\text{thres}} = 1$, the threshold decrement size $\epsilon = \min\{|\tilde{\pi}_{ij} - \tilde{\pi}_{i'j'}| : i \in \mathcal{S}_j, i' \in \mathcal{S}_{j'}, j, j' \in \mathcal{N}\}$, and set $\text{EXIT}_{\text{flag}} = 0$.
4: **while** $\text{EXIT}_{\text{flag}} = 0$ **do**
5: For all $l \in \mathcal{S}_k$ and $k \in \mathcal{N}$, let $\pi_{lk} = 1$ if $\tilde{\pi}_{lk} \geq t_{\text{thres}}$ and $\pi_{lk} = 0$ otherwise.
6: **while** network-wide energy budget (25) not satisfied **do**
7: Find $(l^*, k^*) = \arg\min_{l,k}\{\tilde{\pi}_{lk} : \pi_{lk} = 1\}$. Set $\pi_{l^*k^*} = 0$ and $\text{EXIT}_{\text{flag}} = 1$.
8: **end while**
9: Update the threshold as $t_{\text{thres}} \leftarrow t_{\text{thres}} - \epsilon$.
10: **end while**
11: **for** each $k \in \mathcal{N}$ **do**
12: **if** local energy budget (24) not satisfied for node k **then**
13: Find $l^* = \arg\min_l\{\tilde{\pi}_{lk} : \pi_{lk} = 1\}$. Set $\pi_{l^*k} = 0$.
14: For (P2), set $\pi_{l^*j} = 0$, for all nodes j reachable from node l^* via node k. For (P3), if $l^* = k$, set $\pi_{l^*j} = 0$, for all nodes $j \in \mathcal{N}_{k\rightarrow}$.
15: **end if**
16: **end for**
17: Determine values of $\{\delta_{lk}\}_{l\in\mathcal{S}_k,k\in\mathcal{N}}$ from $\{\pi_{lk}\}_{l\in\mathcal{S}_k,k\in\mathcal{N}}$.
18: Return $\{(\pi_{lk}, \delta_{lk}) : l \in \mathcal{S}_k, k \in \mathcal{N}\}$ as the approximately optimal solution of the binary relay and selection variables for the original MILP.

Theorem 2. *Algorithm 1 returns a feasible solution for the relay and selection variables of (P2) and (P3).*

Proof: See Appendix B. ■

Remark 2. The optimization formulation (P2) or (P3) provides a flexible platform to include additional constraints that may appear in various applications. For example, in (P2), we can easily restrict the information neighbors of a particular node to be within a given number of hops. For another example, the mATC strategy determined by (P3) can lead to improvements over various modified ATC strategies [18], [22], [23] by simply imposing constraints on the hops for gathering consultations or on the number of active links associated with each node.

C. Heuristic distributed solution satisfying local energy budgets

In this subsection, we present an online distributed algorithm to heuristically solve problem (P1) without the network-wide energy budget, for an arbitrary network in which consultations are restricted to at most h hops. In this case, by Lemma 2 every node will try to use up its energy budget in each iteration to diffuse intermediate estimates, such that the global cost $\sum_{k\in\mathcal{N}}(\sum_{l\in\overline{\mathcal{N}}^{h'}_{k\leftarrow}} \gamma_l^{-2})^{-1}$ is minimized, where $\overline{\mathcal{N}}^{h'}_{k\leftarrow}$ is a subset of multi-hop neighbors at most h hops away from node

k. Since each node k contributes to a cost that is decreasing in the size of its information neighborhood $\overline{\mathcal{N}}_{k\leftarrow}^{h'}$ and increasing in the composite variance γ_l^2 of each information neighbor l, heuristically, we see that each node should broadcast its current intermediate estimate and empirical estimate of its composite variance at each iteration, as well as information from its h-hop neighborhood corresponding to nodes with the smallest empirical composite variances, up to its local energy budget. The distributed and adaptive mATC diffusion algorithm is described formally in Algorithm 2.

Algorithm 2 Approximate, distributed and adaptive solution for (P1) with only local energy budget constraints

1: Each node $k \in \mathcal{N}$ initializes the estimates $\hat{\gamma}_{k,1}(-1)$, $\boldsymbol{w}_{k,-1}$ and $\hat{\boldsymbol{R}}_{u,k}(-1)$ (cf. (12)) as a zero scalar, $M \times 1$ vector and $M \times M$ matrix, respectively, and sets $i = 0$.
2: Each node $k \in \mathcal{N}$ uses its own measurement $(\boldsymbol{u}_{k,i}, \boldsymbol{d}_k(i))$ to update $\boldsymbol{\psi}_{k,i}$ and $\hat{\gamma}_k^2(i)$ by (3) and (12), respectively.
3: If its energy budget permits, each node $k \in \mathcal{N}$ broadcasts its estimates $(\boldsymbol{\psi}_{k,i}, \hat{\gamma}_k^2(i))$ to its directly reachable neighbors, and then chooses to rebroadcast $(\boldsymbol{\psi}_{l,i}, \hat{\gamma}_l^2(i))$ if $\hat{\gamma}_l^2(i-1)$ is among the smallest empirical composite variances it received in the $(i-1)$-th iteration. This process continues for a predefined period of time proportional to the maximum number of hops of consultation, h.
4: Each node $k \in \mathcal{N}$ uses the empirical composite variances received in Step 3 to compute the combination weights using (13), and then combines the received intermediate estimates by (3) to get its own estimate $\boldsymbol{w}_{k,i}$.
5: Each node completes the i-th iteration, sets $i \leftarrow i+1$ and goes to repeat the above procedure from Step 2.

In Algorithm 2, the maximum number of consultation hops h controls the length of time consumed to diffuse information over the network in each iteration. By applying the procedure of Algorithm 2, each node is able to combine intermediate estimates from a subset of its h-hop neighborhood adaptively. This holds even if the parameter vector and the data and noise statistics change slowly in time. We evaluate the performance of this algorithm in Section VI-B.

D. An approximate asynchronous implementation

If each multi-hop relaying is to be completed within each diffusion iteration, as is assumed in our analysis, this may require the nodes to take observations at a slower rate due to a longer communication delay at each iteration. To avoid this additional delay, one can perform the relaying over multiple diffusion iterations, so that intermediate estimates of multi-hop information neighbors are combined only in later diffusion iterations. We call this *asynchronous* mATC. The asynchronous mATC avoids waiting to receive the multi-hop information, and allows the mATC strategy to perform each combination step in the same time scale as the ATC strategy. Our simulation results in Section VI demonstrate that asynchronous mATC has similar MSD performance as mATC.

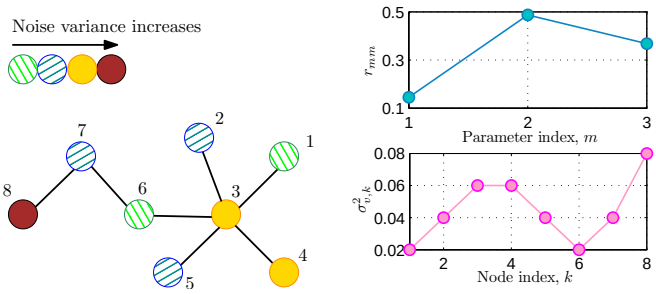


Fig. 4: Network topology and data and noise profiles of every node. The number next to a node denotes the node index.

VI. Simulation Results

We illustrate the application of the mATC strategy to a tree and general graph network, and compare its performance with those of the non-cooperative, ATC diffusion and the centralized strategies. In the simulations, the step sizes and the discount factors are set as $\mu_k = 0.08$ and $\nu_k = 0.05$, respectively, for all $k \in \mathcal{N}$, and the numerical results are averaged over 1000 instances unless otherwise indicated.

A. A simple tree network

An undirected tree network is shown in Fig. 4 together with the noise and data profiles of each node. The quantities r_{mm} are randomly generated such that $\sum_{m=1}^{M} r_{mm} = 1$ and the regression covariance matrix $R_{u,k}$ is diagonal with entries equal to $r_{mm} \cdot 100\sigma_{v,k}^2$. Each node aims to estimate a 3×1 vector ω^o with every entry equal to $1/\sqrt{3}$. We impose a maximum number n_b of broadcasts allowed in the network per iteration as the network-wide energy constraint.

We illustrate the flexibility and usefulness of the proposed balancing rule in Theorem 1 for assigning the combination weights. We set n_b to 8, the same as that invoked by the ATC strategy. Both the ATC and the mATC strategies adopt the balancing rule with the balancing coefficient α^o equal to the optimal value 0.9978, or the extreme value of 0 or 1. The simulation results under the non-adaptive and adaptive implementations are shown in Fig. 5(a) and Fig. 5(b), respectively. We observe that while the balancing rule with $\alpha^o = 0$ gives the best transient network MSD, the rule with $\alpha^o = 1$ gives the minimum steady-state network MSD. In comparison, the optimal balancing rule using $\alpha^o = 0.9978$ finds a good balance between these two extremes. The results also show that the mATC strategy outperforms the ATC strategy in the steady state when α^o is optimal or equal to 1.

We next investigate the trade-off between the network performance and the energy budget available per iteration. We define the *convergence rate* as the quotient of the decrease in the network MSD till 90% of its steady-state value divided by the number of iterations to achieve that decrease. When the mATC adopts the optimal balancing rule, the theoretical (based on the recursive equation (5)) and numerical results are shown in Fig. 6. We observe a sharp increase in the convergence rate when the strategy transits from a non-cooperative mode (with $n_b = 0$) to the cooperative modes (with $n_b \geq 1$). In the

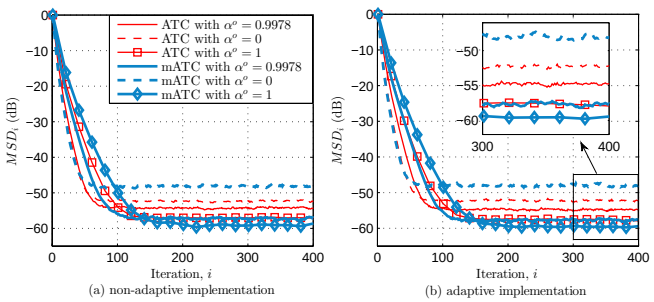


Fig. 5: MSD performance of the ATC and the mATC diffusion strategies implementing the balancing rule of Theorem 1.

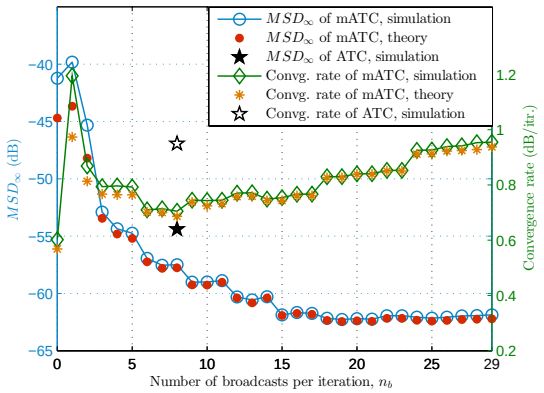


Fig. 6: The steady-state value and the convergence rate of the network MSD as functions of the number of broadcasts per iteration in the network (where 0 and 29 broadcasts correspond to non-cooperative and centralized estimation, respectively).

cooperative modes, the steady-state network MSD decreases almost monotonically as the number of broadcasts increases, while the convergence rate first decreases and then increases. (The zig-zag variations in the curves are due to the limitations of minimizing the *upper bound* of the steady-state network MSD, and not a simulation artifact.) Once sufficient number of broadcasts are invoked, the marginal benefit brought by more consultations is small to the steady-state network MSD, but still notable to the convergence rate.

A specific solution of the information neighbor configuration is shown in Fig. 7, which invokes 3 or 37.5% less broadcasts in each iteration compared to the ATC strategy. The figure also shows results obtained with an *asynchronous* mATC strategy, in which intermediate estimates from two-hop information neighbors arrive and are combined only in the next iteration. The estimation performance turns out to be close to the synchronous mATC strategy.

B. An arbitrary network

We randomly generated an undirected network with 20 nodes within a 10×10 square area, as shown in Fig. 8 together with the noise and data power profiles, where the data power $\mathrm{Tr}(R_{u,k})$ is equally distributed over the parameter components of each node k. Each node aims to estimate a 2×1 vector ω^o with every entry equal to $1/\sqrt{2}$. To mimic a

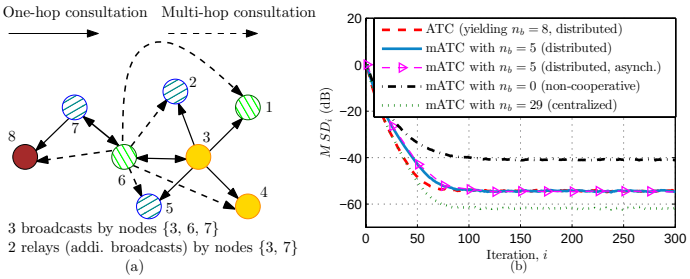


Fig. 7: Comparison of the ATC and mATC diffusion strategies adopting the optimal balancing rule: (a) consultations in each iteration with the mATC strategy limited to 5 broadcasts per iteration; (b) network MSD curves obtained from simulations.

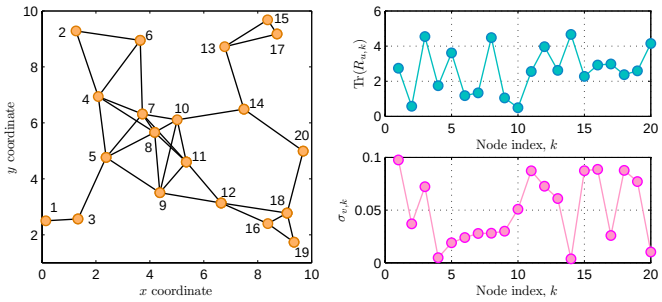


Fig. 8: The topology of a random network, and the data and noise profiles at each node. The number next to a node is the node index.

heterogenous environment, we assume that the communication cost incurred is proportional to the square distance of a node to its farthest directly reachable neighbor, i.e., the communication cost coefficient of node k in (14) is given by

$$c_k^{cm,0} = c^0 \times \left(\max\{\mathrm{dist}(k,l) : l \in \mathcal{N}_{k\rightarrow}\}\right)^2,$$

where c^0 is a given scalar and $\mathrm{dist}(k,l)$ is the distance between node k and node l. Without loss of generality, we set $c^0 = 1$, and we study the energy-performance trade-off when the network is subject to a network-wide energy budget and the information neighbors of every node are restricted to be within two hops away.

Fig. 9 shows the energy-performance trade-off when the mATC diffusion strategy adopts the balancing rule of Theorem 1. We show the performances of both the exact solution found using the MILP (P3) and the approximate solution found through relaxations using Algorithm 1. We observe that the steady-sate network MSD decreases almost monotonically with the energy budget, while the convergence rate of the network MSD first increases as cooperation is enabled, and then fluctuates and becomes steady as more energy budget is available. The corresponding changes in the information neighbor configuration are illustrated in Fig. 10(a)–10(c). The performance gain turns out to be very small if the energy budget is large enough (larger than 250 in this case). The approximate solutions give similar energy-performance trade-offs, but have uniformly worse steady-state network MSDs.

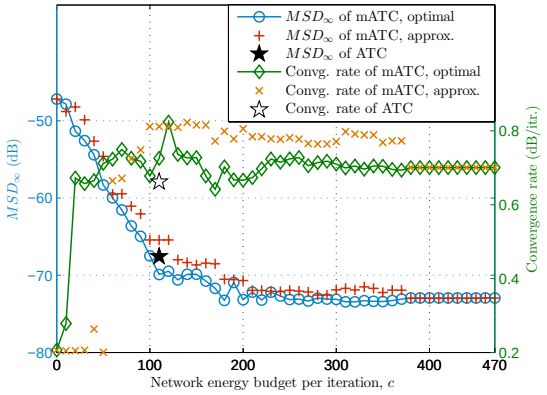
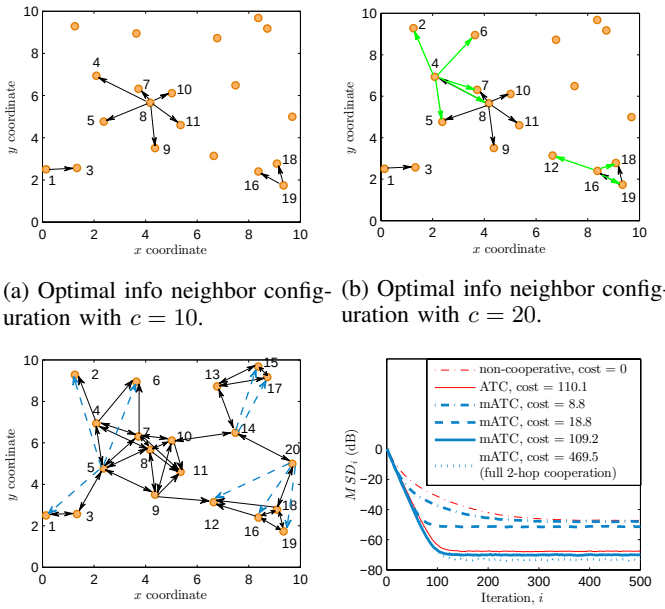


Fig. 9: The energy-performance trade-offs and convergence rates of mATC and ATC strategies.

In addition to ATC, we also compare the mATC strategy with the game theoretic combine-then-adapt diffusion strategy proposed in [30]. We call this the CTA-game strategy for short. This strategy requires the setting of a couple of parameters, and we refer the reader to [30] for the full details. Specifically, the local and global utility gains $K_{l,1}$ and K_g are chosen to be 1, the local energy price $K_{l,2}$ is given a value in the range of [0, 0.5] for every sensor (the network steady-state MSD and the convergence rate are found to remain the same once $K_{l,2} \geq 0.5$ in this case), the "exploration" factor δ is set to 0.1, and the other parameters γ_l and γ_g are fixed as 1. For the CTA-game strategy, we average the simulations results over 300 random instances, each using 2000 iterations.

In Fig. 11, we compare the CTA-game strategy with two versions of the mATC strategy: one implemented using full knowledge of the noise and input data statistics, and another in an adaptive manner per Algorithm 2 (where the local energy budgets correspond to the configurations that yield the optimal performance-energy trade-off curve shown in Fig. 9). We also compare against an adaptive implementation of an energy-constrained ATC strategy, which we call cATC, in which each node diffuses only its own intermediate estimate if its energy budget permits. We see that both versions of mATC require much less total energy to converge to 90% of the same steady-state network MSD as the CTA-game strategy. For example, for a steady-state MSD of -62 dB, the adaptive mATC strategy consumes about 0.6×10^4 units of energy. In contrast, the CTA-game strategy consumes about 3.6×10^4 units of energy, which is six times of that used by the adaptive mATC. The higher total energy consumption of the CTA-game strategy is found to be caused by its slow convergence, which does not change much even if the parameters are tuned in the given ranges. The simulation results also indicate that the adaptive mATC diffusion strategy requires less energy for the network to converge to the same MSD, as compared to the adaptive cATC strategy.

We also examine the adaptability of the mATC strategy in response to changes in the parameter w^o, the measurement noise variances and the locally available energy budgets. We assume that the network starts with the configuration shown



(a) Optimal info neighbor configuration with $c = 10$.

(b) Optimal info neighbor configuration with $c = 20$.

(c) Optimal info neighbor configuration with $c = 110$.

(d) Network MSD curves for different energy costs.

Fig. 10: Three optimal configurations of information neighbors and related network MSD curves obtained from simulations. The arrows in (a)–(c) indicate the diffusion directions. The thicker green arrows in (b) indicate new diffusions relative to those in (a), and the light blue dashed arrows in (c) represent two-hop diffusions that require neighbors' relay.

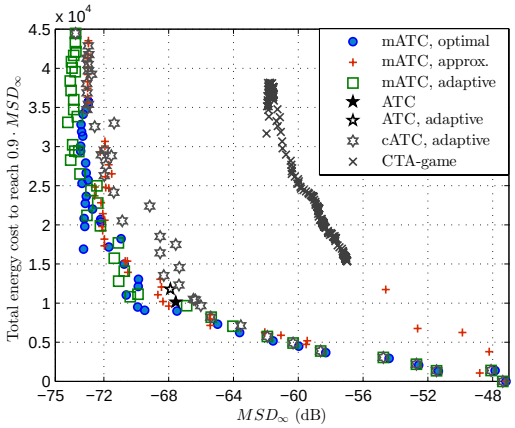


Fig. 11: Average total energy cost required for the network to converge to 90% of its steady-state MSD value.

in Fig. 10(c), where each node has the exact local energy budget to support the transmissions shown. We implement an asynchronous version of the adaptive mATC strategy in Algorithm 2 (i.e., intermediate estimates from two-hop neighbors arrive and are combined only in the next diffusion iteration), and the adaptive cATC strategy. After 1000 iterations, the parameter vector w^o changes from the original $[1/\sqrt{2}, 1/\sqrt{2}]^T$ to the new $[1/2, \sqrt{3}/2]^T$ while the measurement noise variance $\sigma^2_{v,k}$ is doubled for all $k \in \mathcal{N}$. After 2000 iterations, the local energy budgets available to the nodes 3, 4, 13 and 18

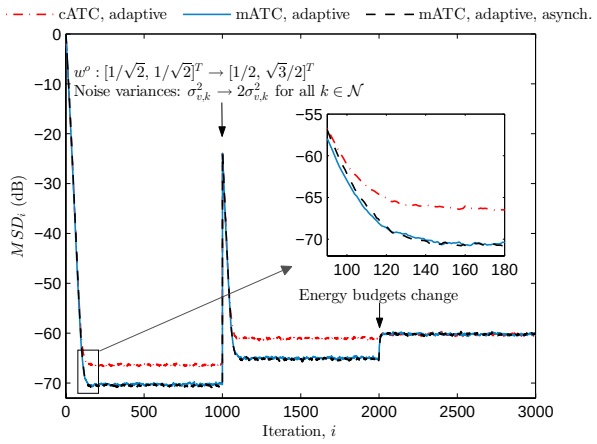


Fig. 12: Performances of the cATC and mATC diffusion strategies in response to changes in network conditions.

all decrease to support only one broadcast instead of two broadcasts per iteration, and the local energy budgets available to nodes 7 and 8 are set to zero. We observe from Fig. 12 that ATC, mATC, and the asynchronous mATC are all able to adapt to the changes in network conditions, and that both mATC and asynchronous mATC achieves steady-state MSDs better than cATC (except after the last change when mATC becomes equivalent to ATC), thanks to their selective multi-hop consultations that fully exploit the local energy budgets.

VII. Conclusion

In this paper, we have considered the use of multi-hop diffusion that allows nodes to exchange intermediate parameter estimates with their selected information neighbors instead of just the physical neighbors. For two classes of networks, we propose an MILP to select the information neighbors together with the relay nodes for each node, which approximately optimizes a trade-off between the available energy budgets for each iteration, and the steady-state network MSD performance. For arbitrary networks in which there are only local energy budget constraints, and consultations constrained to within a fixed number of hops, we propose a distributed and adaptive algorithm to select the information neighbors. Simulation results suggest that our proposed methods achieve better MSD performance than the traditional diffusion strategy, while having the same or lower communication cost.

Our current optimization procedure for networks with a network-wide energy budget requires knowledge of the network topology as well as the data and noise variance profiles of every node. This implies that the optimization can only be performed at a centralized processor, and only infrequently. It would be of future research interest to develop distributed optimization techniques like those in [45] to perform online adaptive optimization as the network conditions vary over time, and to study the frequency at which such optimization needs to be run, in order to maintain a reasonable level of optimality.

Appendix A
Derivation of the SDP in (7) for minimizing β

Some well-known matrix results used by the derivation are first given in the following lemma, in which (a) is proved by using the singular value decomposition technique and the proofs of (c)-(d) can be found in pages 399, 19 and 473 of [46], respectively.

Lemma 4.

(a) For any matrix $X \in \mathbb{R}^{n\times n}$, $X^TX \preceq I_n$ if and only if $XX^T \preceq I_n$.

(b) If $A \in \mathbb{R}^{n\times n}$ is positive semi-definite, then for any $C \in \mathbb{R}^{n\times n}$, C^*AC is positive semi-definite.

(c) If matrices X, Y and $X+Y$ are invertible, then $(X+Y)^{-1} = X^{-1} - X^{-1}(Y^{-1}+X^{-1})^{-1}X^{-1}$.

(d) Suppose that a Hermitian matrix is partitioned as $\begin{bmatrix} A & B \\ B^* & C \end{bmatrix}$, where A and C are positive definite. This matrix is positive semi-definite if and only if $C - B^*A^{-1}B$ is positive semi-definite.

Let $P_\beta \triangleq \mathcal{M}\mathcal{S}\mathcal{M}/\beta$ and $Q \triangleq (I_{NM} - \mathcal{M}\mathcal{R})^2$, which are positive definite matrices; and let $\phi_\beta \triangleq \sqrt{P_\beta + Q}$, which is a positive definite matrix, and so $\phi_\beta^2 = P_\beta + Q$. The derivation proceeds as follows (other constraints on the combination weight matrix A and the constraint $\beta > 0$ are not shown):

$$\begin{aligned}
&\min_{A,\,\beta} \beta, \ \text{s.t. } \lambda_{\max}(\mathcal{Y}/\beta + \mathcal{B}\mathcal{B}^*) \le 1, \ A^T\mathbf{1}_N = \mathbf{1}_N \\
&\Leftrightarrow \min_{A,\,\beta} \beta, \ \text{s.t. } \frac{\mathcal{Y}}{\beta} + \mathcal{B}\mathcal{B}^* \preceq I_{NM}, \ A^T\mathbf{1}_N = \mathbf{1}_N \\
&\Leftrightarrow \min_{A,\,\beta} \beta, \ \text{s.t. } \mathcal{A}^T\phi_\beta^2\mathcal{A} \preceq I_{NM}, \ , \ A^T\mathbf{1}_N = \mathbf{1}_N \\
&\overset{\text{Lemma 4(a)}}{\Longleftrightarrow} \min_{A,\,\beta} \beta, \ \text{s.t. } \phi_\beta\mathcal{A}\mathcal{A}^T\phi_\beta \preceq I_{NM}, \ A^T\mathbf{1}_N = \mathbf{1}_N \\
&\overset{\text{Lemma 4(b)}}{\Longleftrightarrow} \min_{A,\,\beta} \beta, \ \text{s.t. } \mathcal{A}\mathcal{A}^T \preceq \phi_\beta^{-2}, \ A^T\mathbf{1}_N = \mathbf{1}_N
\end{aligned}$$

By left multiplying both sides of the SDP constraint with $\mathbf{1}_N^T \otimes I_M$ and right multiplying with $\mathbf{1}_N \otimes I_M$, the last optimization implies that an approximate solution of β can be solved from:

$$\begin{aligned}
&\min_{\beta} \beta, \ \text{s.t. } (\mathbf{1}_N^T \otimes I_M)(\mathbf{1}_N \otimes I_M) \\
&\qquad \preceq (\mathbf{1}_N^T \otimes I_M)(Q+P_\beta)^{-1}(\mathbf{1}_N \otimes I_M) \\
&\overset{\text{Lemma 4(c)}}{\Longleftrightarrow} \min_{\beta} \beta, \ \text{s.t. } (\mathbf{1}_N^T \otimes I_M)(\mathbf{1}_N \otimes I_M) \\
&\quad \preceq (\mathbf{1}_N^T \otimes I_M)[Q^{-1} - Q^{-1}(P_\beta^{-1}+Q^{-1})^{-1}Q^{-1}](\mathbf{1}_N \otimes I_M) \\
&\Leftrightarrow \min_{\beta} \beta, \ \text{s.t. } (\mathbf{1}_N^T \otimes I_M)(Q^{-1} - I_{NM})(\mathbf{1}_N \otimes I_M) \\
&\quad - (\mathbf{1}_N^T \otimes I_M)Q^{-1}(P_\beta^{-1}+Q^{-1})^{-1}Q^{-1}(\mathbf{1}_N \otimes I_M) \succeq 0 \\
&\overset{\text{Lemma 4(d)}}{\Longleftrightarrow} \min_{\beta} \beta, \ \text{s.t.} \\
&\begin{bmatrix} P_\beta^{-1}+Q^{-1} & Q^{-1}(\mathbf{1}_N \otimes I_M) \\ (\mathbf{1}_N^T \otimes I_M)Q^{-1} & (\mathbf{1}_N^T \otimes I_M)(Q^{-1} - I_{NM})(\mathbf{1}_N \otimes I_M) \end{bmatrix} \succeq 0.
\end{aligned}$$

The last optimization problem is an SDP, and can be solved by standard solvers to get an approximate solution of β for the original problem.

APPENDIX B
PROOF OF THEOREM 2

We first prove the theorem for (P2). The proof relies on the following lemma, which can be shown by using the valid inequalities (35), and is omitted for brevity.

Lemma 5. *Let the optimal relay and selection variables of the relaxed LP corresponding to (P2) (including the valid inequalities in (35)) be* $\{(\tilde{\pi}_{lk}, \tilde{\delta}_{lk}) : l \in \overline{\mathcal{N}}_{k\leftarrow}, k \in \mathcal{N}\}$. *For any relay path from a node* l *to another node* j *such that* $\tilde{\delta}_{lj} = 1$, *and any pair of nodes* (k_1, k_2), *where* k_2 *is a successor of* k_1 *on the relay path, we have* $\tilde{\pi}_{lk_1} \geq \tilde{\pi}_{lk_2}$.

To prove that the solution of the relay and selection variables returned by Algorithm 1 is feasible for (P2), we need to show that: (i) the relay variables, obtained after thresholding those from the relaxed LP, result in valid transmission paths (i.e., every path is able to deliver the information as desired); and (ii) the local and network-wide energy budgets are satisfied by each node and the whole network, respectively. These are proved as follows.

Observe that line 5 of Algorithm 1 returns relay variables feasible for (P2) without energy budget constraints. This is because the solution is able to indicate every information relay path without ambiguity as shown by Lemma 5.

Lines 6–8 choose relays in non-decreasing order of their corresponding relay variable values, and removes them so as to satisfy the network-wide energy budget constraint. By Lemma 5, the operation only removes tails of certain relay paths sequentially until the network-wide energy budget constraint is satisfied. Therefore, the remaining part of the relay path is still valid.

In lines 12–15, selected relays for each node $k \in \mathcal{N}$ are removed in non-decreasing order of their $\tilde{\pi}_{lk}$ values in order to reduce node k's energy consumption to not more than c_k so that (24) is satisfied. If node k stops relaying information from node l^*, then we also set $\pi_{l^*j} = 0$ for all nodes j that used to obtain node l^*'s information through node k. This ensures that the relay path remains valid.

Since a feasible solution of the relay variables uniquely determines a feasible solution of the selection variables, the solution of the selection variables obtained from line 5 is feasible for (P2).

The theorem for (P3) can be proved in a similar way, because the result in Lemma 5 remains true for the relaxed (P3) with the valid inequalities (34). The proof is now complete.

REFERENCES

[1] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*. Berkeley, California: ACM, Apr. 2004, pp. 20–27.

[2] S. Kar and J. M. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2009.

[3] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," *arXiv preprint arXiv:1307.1448*, 2013.

[4] L. Xie, D.-H. Choi, S. Kar, and H. V. Poor, "Fully distributed state estimation for wide-area monitoring systems," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1154–1169, 2012.

[5] X. Li and A. Scaglione, "Robust decentralized state estimation and tracking for power systems via network gossiping," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1184–1194, 2013.

[6] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "A collaborative training algorithm for distributed learning," *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1856–1871, 2009.

[7] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 97–123, 2011.

[8] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems*, vol. 22, no. 3, pp. 52–67, 2002.

[9] S. Barbarossa and G. Scutari, "Bio-inspired sensor network design," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, 2007.

[10] F. S. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2038–2051, 2011.

[11] J. N. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Transactions on Automatic Control*, vol. 29, no. 1, pp. 42–50, 1984.

[12] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

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

[14] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM Journal on Optimization*, vol. 7, no. 4, pp. 913–926, 1997.

[15] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[16] ——, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.

[17] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[18] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[19] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Academic Press, Elsevier, 2014, vol. 3, pp. 323–454.

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

[21] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks—Part I: modeling and stability analysis," *arXiv preprint arXiv:1312.5434*, 2013.

[22] O. Rortveit, J. Husoy, and A. H. Sayed, "Diffusion LMS with communication constraints," in *the 44th Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, CA, Nov. 2010.

[23] X. Zhao and A. H. Sayed, "Single-link diffusion strategies over adaptive networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012.

[24] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[25] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies." in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, Apr 2008, pp. 3285–3288.

[26] N. Takahashi and I. Yamada, "Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. Dallas, Texas, USA: IEEE, 2010, pp. 3518–3521.

[27] S. Xu, R. C. de Lamare, and H. V. Poor, "Adaptive link selection strategies for distributed estimation in diffusion wireless networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

[28] S. Werner, T. Riihonen, and Y.-F. Huang, "Energy-efficient distributed parameter estimation with partial updates," in *International Conference on Green Circuits and Systems (ICGCS)*. Shanghai, China: IEEE, Jun 2010, pp. 36–40.

[29] R. Arablouei, S. Werner, Y. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, 2014.
[30] O. Namvar Gharehshiran, V. Krishnamurthy, and G. Yin, "Distributed energy-aware diffusion least mean squares: Game-theoretic learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 821–836, 2013.
[31] R. Arroyo-Valles, S. Maleki, and G. Leus, "A censoring strategy for decentralized estimation in energy-constrained adaptive diffusion networks," in *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Darmstadtium Darmstadt, Germany, 2013.
[32] M. Nokleby, W. U. Bajwa, R. Calderbank, and B. Aazhang, "Toward resource-optimal consensus over the wireless medium," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, p. 284?295, 2013.
[33] A. D. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient averaging for sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1205–1216, 2008.
[34] F. Bénézit, A. G. Dimakis, P. Thiran, and M. Vetterli, "Gossip along the way: Order-optimal consensus through randomized path averaging," in *Proceedings of the Allerton Conference on Communication, Control, and Computing*, Allerton, IL, 2007.
[35] T. J. Johnson, R. L. Brown, D. E. Adams, and M. Schiefer, "Distributed structural health monitoring with a smart sensor array," *Mechanical Systems and Signal Processing*, vol. 18, no. 3, pp. 555–572, 2004.
[36] X. Liu, J. Cao, W.-Z. Song, and S. Tang, "Distributed sensing for high quality structural health monitoring using wireless sensor networks," in *IEEE 33rd Real-Time Systems Symposium*, San Juan, PR, USA, 2012.
[37] W. Hu and W. P. Tay, "Generalized diffusion adaptation for energy-constrained distributed estimation," in *17th International Conference on Information Fusion*, Salamanca, Spain, July 2014.
[38] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ: John Wiley & Sons, 2008.
[39] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, 2012.
[40] C.-K. Yu and A. Sayed, "A strategy for adjusting combination weights over adaptive networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.
[41] J. Fernandez-Bes, J. Arenas-García, and A. H. Sayed, "Adjustment of combination weights over adaptive diffusion networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014.
[42] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems," *Mathematical Programming*, vol. 10, no. 1, pp. 147–175, 1976.
[43] B. Golden, S. Raghavan, and E. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer Verlag, 2008, vol. 43.
[44] S. Burer and A. Saxena, "The MILP road to MIQCP," in *Mixed Integer Nonlinear Programming*. Springer, 2012, pp. 373–405.
[45] Z. J. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Transations on Signal Processing*, vol. 62, no. 15, pp. 3924–3938, 2014.
[46] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York: Cambridge University Press, 1990.

Wuhua Hu (S'09 M'13) received the B.Eng. degree in Automation in 2005 and the M.Eng. degree in Detecting Technique and Automation Device in 2007 from Tianjin University, Tianjin, China. He earned the Ph.D. degree in Communication Engineering from Nanyang Technological University, Singapore, in 2012. He is currently a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, prior to which he held a position of Assistant Professor in the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. His research interests lie in modeling, control and optimization of dynamical systems and signal processing, with applications focused on power and energy systems, and industry processes.

Wee Peng Tay (S'06 M'08 SM'14) received the B.S. degree in Electrical Engineering and Mathematics, and the M.S. degree in Electrical Engineering from Stanford University, Stanford, CA, USA, in 2002. He received the Ph.D. degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. He is currently an Assistant Professor in the School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore. His research interests include distributed detection and estimation, distributed signal processing, sensor networks, social networks, information theory, and applied probability.

Dr. Tay received the Singapore Technologies Scholarship in 1998, the Stanford University President's Award in 1999, the Frederick Emmons Terman Engineering Scholastic Award in 2002, and the Tan Chin Tuan Exchange Fellowship in 2015. He is the coauthor of the best student paper award at the 46th Asilomar conference on Signals, Systems, and Computers. He currently serves on the MLSP TC of the IEEE Signal Processing Society, and as the chair of DSNIG in IEEE MMTC. He has also served as a technical program committee member for various international conferences.