

Estimation on Graphs from Relative Measurements

PRABIR BAROOAH and JOÃO P. HESPAHHA

DISTRIBUTED ALGORITHMS AND FUNDAMENTAL LIMITS

A sensor network is a collection of interconnected nodes that are deployed in a geographic area to perform monitoring tasks. Each node is equipped with sensing and computing capability. Sensor networks consisting of a large collection of nodes are currently under development or envisioned for the near future [1], [2]. Usually, each node can communicate with only a small subset of the remaining nodes. These communication constraints define a graph whose vertices are the nodes and whose edges are the communication links. In typical situations, a node may lack knowledge of certain attributes such as its own position in a global reference frame. However, nodes might be capable of measuring relative values with respect to nearby nodes. In this scenario, it is desirable to use relative measurements to estimate global attributes. We describe three scenarios that motivate these problems.

Consider the problem of localization, where a sensor does not know its position in a global coordinate system but can measure its position relative to a set of nearby nodes. These measurements can be obtained, for example, from range data and bearing (that is, angle) data (see Figure 1). In particular, two nearby sensors u and v located in a plane at positions p_u and p_v , respectively, have access to the measurement

$$\zeta_{u,v} = p_u - p_v + \varepsilon_{u,v},$$

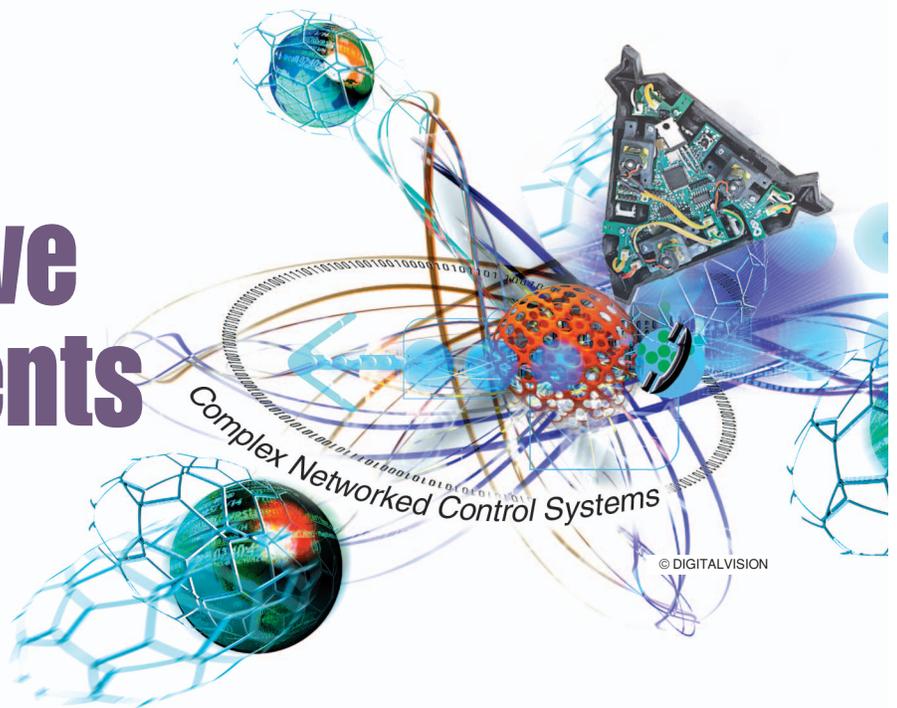
where $\varepsilon_{u,v}$ denotes measurement error. The problem of interest is to use the $\zeta_{u,v}$ s to estimate the positions of all the nodes in a common coordinate system whose origin is fixed arbitrarily at one of the nodes.

The second scenario involves the time-synchronization problem, in which the sensing nodes are part of a multi-hop communication network. Each node has a local clock, but each pair of clocks differs by a constant offset. However, nodes that communicate directly can estimate the difference between their local clocks by exchanging “hello” messages that are time stamped with local clock times. For example, suppose that nodes u and v can communicate directly with each other and have clock offsets t_u and t_v with respect to a reference clock. By passing messages back and forth, the nodes can measure the relative clock offset $t_u - t_v$ with the measurement

$$\zeta_{u,v} = t_u - t_v + \varepsilon_{u,v} \in \mathbb{R},$$

where $\varepsilon_{u,v}$ denotes measurement error (see Figure 2). The task is now to estimate the clock offsets with respect to the global time, which is defined to be the local time at some reference node. A variation of this problem is considered in [3].

The third scenario is a motion consensus problem, in which each agent wants to determine its velocity with respect to the velocity of a leader using only



measurements of its relative velocities with respect to nearby agents. These measurements can be obtained, for example, by using vision-based sensors. In particular, two nearby agents u and v moving with velocities \dot{p}_u and \dot{p}_v , respectively, have access to the measurement

$$\zeta_{u,v} = \dot{p}_u - \dot{p}_v + \varepsilon_{u,v},$$

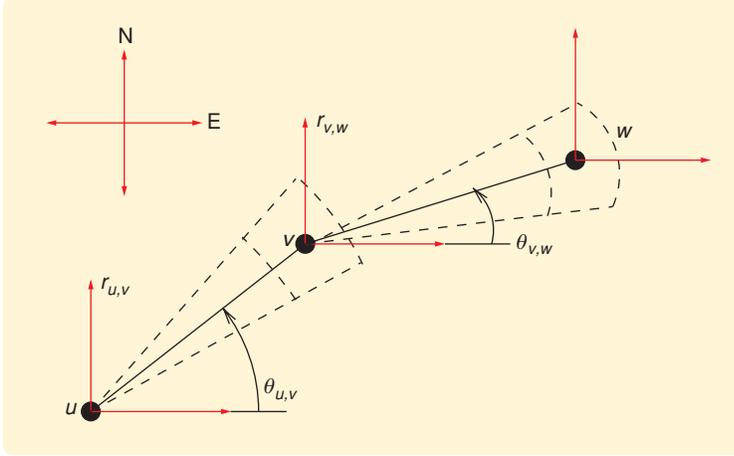


FIGURE 1 Relative position measurement in a Cartesian reference frame using range and bearing measurements. A local compass at each sensor is needed to measure bearing with respect to a common north. Noisy measurements of $r_{u,v}$ and $\theta_{u,v}$, range and bearing between a pair of sensors u and v are converted to noisy measurements of relative position in the $x-y$ plane as $\zeta_{u,v} = [r_{u,v} \cos \theta_{u,v}, r_{u,v} \sin \theta_{u,v}]^T$. The same procedure is performed for every pair of sensors that can measure their relative range and bearing. The task is to estimate the node positions from the relative position measurements.

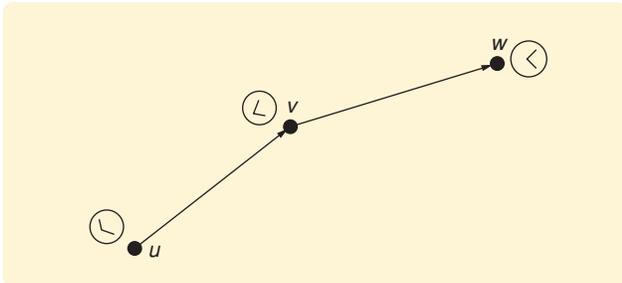


FIGURE 2 Measurement of differences in local times by exchanging time-stamped messages. Node u transmits a message, say, at global time t , while the local time of the transmitter u is $\tau_{tu} = t + t_u$. The receiver v receives this message at a later time, when its local clock reads $\tau_{rv} = t + t_v + \delta_{u,v}$, where $\delta_{u,v}$ is a random delay arising from the processing of the message at both u and v . Some time later, say at t' (global time), node v sends a message back to u , when its local time is $\tau'_{tv} = t' + t_v$. This message includes the values τ_{rv} and τ'_{tv} in the message body. Receiver u receives this message at local time $\tau'_{ru} = t' + t_u + \delta_{v,u}$, where the delay $\delta_{v,u}$ has the same mean as the delay $\delta_{u,v}$. Node u can now estimate the clock offsets as $\zeta_{u,v} = \frac{1}{2}[(\tau'_{ru} - \tau'_{tv}) - (\tau_{rv} - \tau_{tu})] = t_u - t_v + (\delta_{vu} - \delta_{uv})/2$. The error $\varepsilon_{u,v} := (\delta_{vu} - \delta_{uv})/2$ has zero mean as long as the (unidirectional) delays have the same expected value. The measured clock offset between u and v is now $\zeta_{u,v} = t_u - t_v + \varepsilon_{u,v}$, of the form (1). Similarly, the measurement of clock offsets between nodes v and w is $\zeta_{v,w} = t_v - t_w + \varepsilon_{v,w}$.

where $\varepsilon_{u,v}$ denotes measurement error. The task is to determine the velocity of each agent with respect to the leader based solely on the available relative velocities between pairs of neighboring agents.

GRAPH-INDUCED MEASUREMENT MODEL AND OPTIMAL ESTIMATE

The problems described above share the common objective of estimating the values of *node variables* x_1, \dots, x_n from noisy relative measurements of the form

$$\zeta_{u,v} = x_u - x_v + \varepsilon_{u,v}, \quad u, v \in \{1, \dots, n\}, \quad (1)$$

where $\varepsilon_{u,v}$ is a zero-mean noise vector, with associated covariance matrix $P_{u,v} = E[\varepsilon_{u,v} \varepsilon_{u,v}^T]$. Node variables are vector valued, and the dimension of the node variables is denoted by k . For example, in the time-synchronization problem, $k = 1$, while in the localization problem, k can be 2 or 3. The measurement noise is assumed to be spatially uncorrelated, that is, $E[\varepsilon_{u,v} \varepsilon_{p,q}^T] = 0$ if $u \neq p$ or $v \neq q$. This estimation problem can be naturally associated with the *measurement graph* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, which is a directed graph. For more details on directed graphs and the terminology associated with them, see “Graph-Theoretic Definitions.” The vertex set of the measurement graph consists of the set of nodes $\mathbf{V} := \{1, \dots, n\}$, while its edge set \mathbf{E} consists of all of the ordered pairs of nodes (u, v) for

which a noisy measurement of the form (1) is available.

By stacking together all of the measurements into a single vector \mathbf{z} , all node variables into one vector \mathbf{X} , and all of the measurement errors into a vector ε , we can express all of the measurement equations (1) in the compact form

$$\mathbf{z} = \mathcal{A}^T \mathbf{X} + \varepsilon, \quad (2)$$

where the matrix \mathcal{A} is uniquely determined by the graph \mathbf{G} . To construct \mathcal{A} , we start by defining the *incidence matrix* of \mathbf{G} , which is defined in “Graph-Theoretic Definitions.” A measurement graph and its incidence matrix are shown in “A Graph Estimation Example.” The matrix \mathcal{A} that appears in (2) is an expanded version of the incidence matrix A , defined by $\mathcal{A} := A \otimes I_k$, where I_k is the $k \times k$ identity matrix and \otimes denotes the Kronecker product. Essentially, every entry of A is replaced by a matrix of the form $a_{ue} I_k$ to construct the matrix \mathcal{A} (see “A Graph Estimation Example”).

With relative measurements alone, determining x_u is possible only up to an additive constant. To avoid this ambiguity, we assume that at least one of the nodes is used as a reference by all of the nodes, and therefore its node variable can be assumed known. When several node

variables are known, we can have several references. The task is to estimate all of the unknown node variables from the measurements and the reference variables.

By partitioning \mathbf{X} into a vector \mathbf{x} containing the unknown node variables and another vector \mathbf{x}_r containing the known reference node variables, we can rewrite (2) as $\mathbf{z} = \mathcal{A}_r^T \mathbf{x}_r + \mathcal{A}_b^T \mathbf{x} + \varepsilon$ or

$$\mathbf{z} - \mathcal{A}_r^T \mathbf{x}_r = \mathcal{A}_b^T \mathbf{x} + \varepsilon, \quad (3)$$

where \mathcal{A}_r contains the rows of \mathcal{A} corresponding to the reference nodes and \mathcal{A}_b contains the rows of \mathcal{A} corresponding to the unknown node variables.

Estimation of the unknown node variables in the vector \mathbf{x} based on the linear measurement model (2) is a classical estimation problem. Since ε is a random vector with zero mean and covariance matrix $\mathcal{P} := \mathbb{E}[\varepsilon\varepsilon^T]$, the least squares solution leads to the classical best linear unbiased estimator (BLUE) [6], given by

$$\begin{aligned} \hat{\mathbf{x}}^* &:= \mathcal{L}^{-1} \mathbf{b}, & \mathcal{L} &:= \mathcal{A}_b \mathcal{P}^{-1} \mathcal{A}_b^T, \\ \mathbf{b} &:= \mathcal{A}_b \mathcal{P}^{-1} (\mathbf{z} - \mathcal{A}_r^T \mathbf{x}_r). \end{aligned} \quad (4)$$

Among all linear estimators of \mathbf{x} , the BLUE has the smallest variance for the estimation error $\mathbf{x} - \hat{\mathbf{x}}^*$ [6]. The inverse of the matrix \mathcal{L} , which exists as long as the measurement graph is weakly connected, provides the covariance matrix of the estimation error [7], given by

$$\Sigma := \mathbb{E}[(\mathbf{x} - \hat{\mathbf{x}}^*)(\mathbf{x} - \hat{\mathbf{x}}^*)^T] = \mathcal{L}^{-1}. \quad (5)$$

A directed graph is *weakly connected* if it is connected ignoring the edge directions. A precise definition is given in “Graph-Theoretic Definitions.” The covariance matrix Σ_u for the estimation error of a particular node variable x_u appears in the corresponding $k \times k$ diagonal block of Σ . A measurement graph, along with the corresponding measurement equations (2) and (3) and the estimate (4), is shown in “A Graph Estimation Example.” Since the measurement errors are assumed uncorrelated with one another, the error covariance matrix is block diagonal. For this reason, the structure of the matrix \mathcal{L} is closely related to the structure of the graph Laplacian of \mathbf{G} . For details, see [8].

CHALLENGES IN ESTIMATION ON GRAPHS

The estimation problem defined above can be solved by first sending all measurements to one particular node, computing the optimal estimate using (4) in that node, and then distributing the estimates to the individual nodes. However, this centralized solution is undesirable for several reasons. First, it unduly burdens the nodes close to the central processor, since for a large ad-hoc network of wireless sensor nodes, sending all of the measurements requires multihop communication, and most of the data

transmitted to the central processor have to be routed through the nodes close to it. When the nodes operate on batteries with small energy budgets, this mode of operation greatly reduces the life of the nodes that carry out most of the communication. It should be noted that the primary source of energy consumption in wireless sensor networks is communication [2], while much less energy is consumed for computation [9]. Second, centralized computation is less robust to node and link failures over time. Multihop data transfer to a central node typically requires construction of a routing tree rooted at the central node. Failure of a node in one of the branches of the routing tree effectively cuts off communication from all of the nodes in the tree branch rooted at the faulty node. In addition, construction of a routing tree can be challenging when

Graph-Theoretic Definitions

A *directed graph* \mathbf{G} is a pair (\mathbf{V}, \mathbf{E}) consisting of a set \mathbf{V} of *vertices* or *nodes*, and a set \mathbf{E} of *edges*, where each edge $e \in \mathbf{E}$, is an ordered pair (u, v) of nodes $u, v \in \mathbf{V}$. The edge (u, v) is *directed toward* v and *away from* u , and is *incident on* u and v .

The *incidence matrix* of the directed graph (\mathbf{V}, \mathbf{E}) with n nodes and m edges the $n \times m$ matrix A with one row per node and one column per edge defined by $A := [a_{ue}]$, where a_{ue} is nonzero if and only if the edge $e \in \mathbf{E}$ is incident on the node $u \in \mathbf{V}$, and, when a_{ue} is nonzero, $a_{ue} = -1$ if the edge e is directed toward u and $a_{ue} = 1$ otherwise. Figure S1 of “A Measurement Graph Example” shows an example of a directed graph and its incidence matrix.

A directed graph is *weakly connected* if it is possible to go from every node to every other node by traversing the edges, not necessarily respecting the directions of the edges.

Given two graphs $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ and $\bar{\mathbf{G}} = (\bar{\mathbf{V}}, \bar{\mathbf{E}})$, we say that \mathbf{G} can be *embedded* in $\bar{\mathbf{G}}$, or, $\bar{\mathbf{G}}$ *embeds* \mathbf{G} , if the following conditions hold:

- 1) Every node $u \in \mathbf{V}$ of \mathbf{G} can be mapped to one node $\bar{u} \in \bar{\mathbf{V}}$ of $\bar{\mathbf{G}}$ such that no two nodes of \mathbf{G} are mapped to the same node of $\bar{\mathbf{G}}$.
- 2) For every edge $e \in \mathbf{E}$ between u and v in \mathbf{G} , there is an edge $\bar{e} \in \bar{\mathbf{E}}$ between \bar{u} and \bar{v} in $\bar{\mathbf{G}}$, where \bar{u} and \bar{v} are the nodes of $\bar{\mathbf{V}}$ that correspond to the nodes u and v of \mathbf{V} , respectively.

When \mathbf{G} can be embedded in $\bar{\mathbf{G}}$, we write $\mathbf{G} \subset \bar{\mathbf{G}}$. Figure 6 shows two graphs to illustrate the concept of graph embedding. Since edge directions play no role in the definition of embedding, they are not shown in the figure.

Given two nodes u and v of a graph \mathbf{G} , their *graphical distance*, denoted by $d_{\mathbf{G}}(u, v)$, is the minimal number of edges needed to be traversed in going from one node to the other. In this definition, we allow edges to be traversed in any direction, and therefore $d_{\mathbf{G}}(u, v) = d_{\mathbf{G}}(v, u)$.

communication links suffer from temporary failures or when nodes are mobile [10]. Third, a centralized computation renders the entire network susceptible to a catastrophe if the central processor fails. This discussion raises one of the key issues investigated in this article:

» **Question 1:** Is it possible to construct the optimal estimate (4) in a distributed fashion such that the computation and communication burden is shared equally by all of the nodes? If so, how much communication is required, and how robust is the distributed algorithm with respect to communication faults?

By a *distributed algorithm* we mean an algorithm in which every node carries out independent computations

to estimate its own variable but is allowed to periodically exchange messages with its neighbors. A neighborhood is defined by the edges of the measurement graph \mathbf{G} . In particular, two nodes u and v are allowed to communicate directly if and only if there is an edge between the corresponding vertices of \mathbf{G} (in any direction), which is to say that there is a relative measurement between x_u and x_v . We implicitly assume bidirectional communication between nodes.

In this article we show that it is indeed possible to design scalable distributed algorithms to compute optimal estimates that are robust to communication faults. However, one may wonder what are the *fundamental limitations*

A Graph Estimation Example

A measurement graph \mathbf{G} with four nodes and five edges, in which node 1 is the reference, is shown in Figure S1. The incidence matrix A is therefore a 4×5 matrix consisting of 0s, 1s, and -1 s. The matrix form (2) of the measurement equations (1) for this graph is

$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{bmatrix} = \underbrace{\begin{bmatrix} I & -I & 0 & 0 \\ I & -I & 0 & 0 \\ 0 & I & 0 & -I \\ 0 & I & -I & 0 \\ 0 & 0 & -I & I \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}}_{\boldsymbol{\varepsilon}},$$

where I is the $k \times k$ identity matrix. The four-node variables in the vector \mathbf{x} are related to the five measurements in the vector \mathbf{z} by the $4k \times 5k$ matrix \mathcal{A} , the expanded version of the incidence matrix. The measurement model (3) when node 1 is the reference with $x_1 = 0$ is

$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{bmatrix} = \underbrace{\begin{bmatrix} I \\ I \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} 0 \\ x_r \end{bmatrix}}_{\mathbf{x}_r} + \underbrace{\begin{bmatrix} -I & 0 & 0 \\ -I & 0 & 0 \\ I & 0 & -I \\ I & -I & 0 \\ 0 & -I & I \end{bmatrix}}_{A_b^T} \underbrace{\begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} + \boldsymbol{\varepsilon}.$$

The relationship between the three unknown node variables in the vector \mathbf{x} are related to the known quantities, that is, measurements \mathbf{z} and the reference variable x_1 , by the $3k \times 5k$ matrix \mathcal{A}_b . Since the graph \mathbf{G} is weakly connected, \mathcal{L} is invertible. The optimal estimate of the vector \mathbf{x} , the solution to (4), is given by $\hat{\mathbf{x}}^* = \mathcal{L}^{-1} \mathcal{A}_b \mathcal{P}^{-1} \bar{\mathbf{z}}$. Therefore, the optimal estimates (4) when all measurement covariance matrices are equal to the identity matrix are

$$\begin{bmatrix} \hat{x}_2^* \\ \hat{x}_3^* \\ \hat{x}_4^* \end{bmatrix} = \underbrace{\begin{bmatrix} 4I & -I & -I \\ -I & 2I & -I \\ -I & -I & 2I \end{bmatrix}^{-1}}_{\mathcal{L}} \underbrace{\begin{bmatrix} -I & -I & 0 & 0 & 0 \\ 0 & 0 & 0 & -I & -I \\ 0 & 0 & -I & 0 & I \end{bmatrix}}_{\mathcal{A}_b \mathcal{P}^{-1}} \times \underbrace{\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{bmatrix}}_{\mathbf{z} - A^T \mathbf{x}_r}.$$

Note the Laplacian-like structure of the matrix \mathcal{L} . The covariance matrices of the overall estimation error and of the individual node-variable errors are

$$\Sigma = \frac{1}{6} \underbrace{\begin{bmatrix} 3I & 3I & 3I \\ 3I & 7I & 5I \\ 3I & 5I & 7I \end{bmatrix}}_{\mathcal{L}^{-1}}, \quad \Sigma_2 = \frac{1}{2} I, \\ \Sigma_3 = \frac{7}{6} I, \quad \Sigma_4 = \frac{7}{6} I.$$

The covariance of the estimation error of node u is simply the $u - 1$ th diagonal block of the covariance matrix Σ .

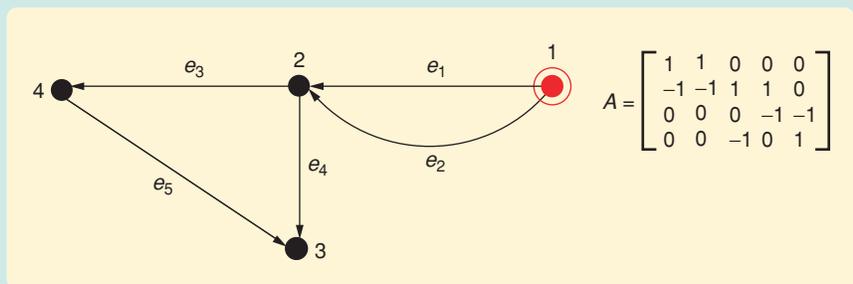


FIGURE S1 A measurement graph \mathbf{G} and its incidence matrix A . The row and column indices of A correspond to node and edge indices, respectively. The single positive entry in each column of A , namely, 1, indicates the start node of the corresponding edge in \mathbf{G} , while the single negative entry -1 indicates the end node.

in terms of accuracy for estimation problems defined in truly large graphs. Reasons for concern arise from estimation problems such as the one associated with the graph shown in Figure 3, which is a chain of nodes with node 1 as the reference and with a single edge $(u + 1, u)$ between consecutive nodes u and $u + 1$. For such a graph the optimal estimate of x_u is given by

$$\hat{x}_u = \zeta_{u,u-1} + \dots + \zeta_{3,2} + \zeta_{2,1} + x_1.$$

Since the variance of a sum of uncorrelated random variables is the sum of their variances, the variance of the optimal estimation error $\hat{x}_u - x_u$ increases linearly with u . Therefore, if the u th node is far from the reference node 1, then its estimate is quite poor. Although the estimation error depends on the values of the variances of the measurements, for this graph the variance of the optimal estimate of x_u grows linearly with u . This example motivates the second issue investigated in this article:

» **Question 2:** Do all graphs exhibit the property that the estimation error variance scales linearly with the distance to the reference nodes? If not, what error scaling laws can occur, and how can one infer these scaling laws from structural properties of the graph?

It seems reasonable that for every measurement graph the estimation error variance increases with the distance to the reference nodes. We show that the exact nature of the scaling of error variance with distance depends on intrinsic structural properties of the measurement graph and that some graphs exhibit scaling laws that are far better than the linear scaling associated with the graph shown in Figure 3. For a given maximal acceptable error, the number of nodes with acceptable estimation errors is large if the graph exhibits a slow increase of variance with distance but is small otherwise. These scaling laws therefore help us to design and deploy large networks for which accurate estimates are possible.

Roughly speaking, the structural properties of interest are related to the denseness of the graph but are not captured by naive measures of density such as node degree or node and edge density, which are commonly used in the sensor networks literature [11], [14], [15]. We describe a classification of graphs that determines how the variance grows with distance from the reference node. In particular, we show that, for a large class of graphs, the variance grows only logarithmically with distance. Most surprisingly, in certain graphs the node variance remains below a constant value, regardless of the distance from the node to the reference node.

In this article, we address the error scaling issue (Question 2) before delving into distributed estimation (Question 1).

ERROR SCALING OF THE OPTIMAL ESTIMATE

As a first step toward addressing the error-scaling issue, we show that the covariance of a node's optimal estimate

Large-scale sensor networks give rise to estimation problems that have a rich graphical structure.

is numerically equal to the *matrix-valued effective resistance* in an abstract electrical network that can be constructed from the measurement graph \mathbf{G} . This analogy with electrical networks is instrumental in deriving several results and builds useful intuition into the problem. We show that the matrix-valued effective resistance in a complicated graph can sometimes be bounded by the matrix-valued effective resistance in a simpler graph, in which we know how the resistance grows with distance. These simpler graphs are the lattice graphs. An answer to the question of variance scaling is thus obtained by exploiting the electrical analogy.

Analogies with electrical networks are used in [12] and [21] to construct solutions to various graph problems, notably those concerned with random walks in graphs. In [12], questions about random walks in certain infinite graphs are answered by bounding the effective resistance in those graphs with that in lattices. It turns out that a similar approach can be used to answer the question of estimation-error scaling once we establish the analogy between error covariance matrices and matrix-valued effective resistances.

Electrical Analogy

A resistive electrical network consists of an interconnection of purely resistive elements. Such interconnections are generally described by graphs whose nodes represent the connection points between resistors and whose edges correspond to the resistors. The *effective resistance* between two nodes in an electrical network is the potential drop between the nodes when a current source of 1 A is

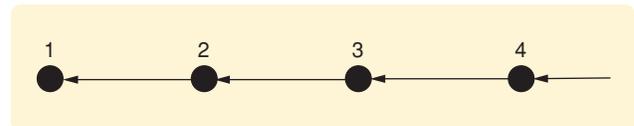


FIGURE 3 A graph where a node variable's optimal estimate has an error variance that grows linearly with the node's distance from the reference. In this case, since every edge represents a measurement with independent error, the optimal estimate of a node variable x_u is simply the sum of the measurements along the edges from the reference (node 1) to node u , while its variance is the sum of the error variances of those measurements. Assuming for simplicity that all of the measurement-error variances are equal, the variance of the estimation error $x_u - \hat{x}_u$ is proportional to the minimal number of edges that must be traversed in going from 1 to u , that is, the graphical distance between 1 and u .

connected across the two nodes (see Figure 4). The computation of effective resistances in an electrical network relies on Kirchoff's laws and Ohm's law.

To see the connection between electrical networks and the estimation problem, consider the measurement graph shown in Figure 5, where the unknown scalar variable x_2 needs to be estimated based on two noisy measurements with error variances σ_a^2 and σ_b^2 . The reference node variable $x_1 = 0$ is assumed known. Calculation using the BLUE covariance (5) shows that the variance σ_2^2 of the optimal estimate of x_2 is given by

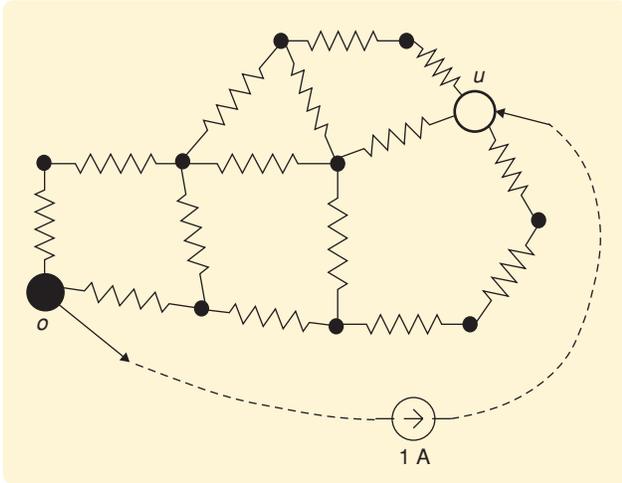


FIGURE 4 A resistive electrical network. A current of 1 A is injected at node u and extracted at the reference node o . The resulting potential difference $V_u - V_o$ is the effective resistance between u and o . For scalar-valued measurements, if every resistance value is set equal to the variance of the measurement associated with that edge, the effective resistance between u and o is numerically equal to the variance of the estimation error $x_u - \hat{x}_u$ obtained with o as the reference.

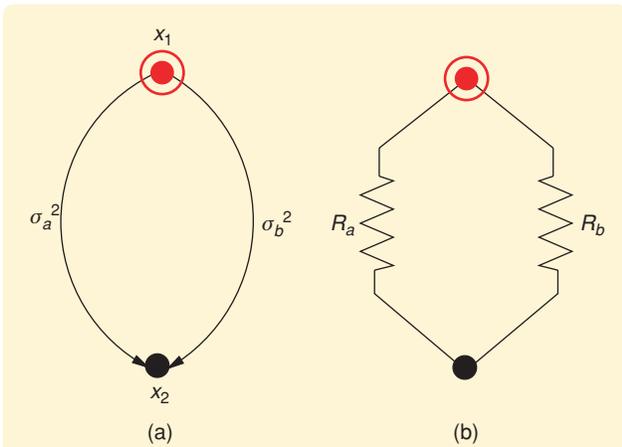


FIGURE 5 (a) measurement graph and (b) the corresponding resistive electrical network. Node 1 is the reference. The variance of the optimal estimate of x_2 has the same numerical value as the effective resistance between nodes 1 and 2 in the electrical network on the right when resistances are chosen to be equal to the measurement-error variances, that is, $R_a = \sigma_a^2$ and $R_b = \sigma_b^2$.

$$\frac{1}{\sigma_2^2} = \frac{1}{\sigma_a^2} + \frac{1}{\sigma_b^2}.$$

Suppose now that we construct a resistive network based on the measurement graph, by setting the edge resistances R_a, R_b equal to the variances σ_a^2, σ_b^2 of the corresponding measurement errors (see Figure 5). The effective resistance between nodes 1 and 2 of the electrical network is thus given by

$$\frac{1}{R_{12}^{\text{eff}}} = \frac{1}{R_a} + \frac{1}{R_b} = \frac{1}{\sigma_a^2} + \frac{1}{\sigma_b^2} = \frac{1}{\sigma_2^2}, \quad (6)$$

and therefore the variance σ_2^2 of the optimal estimate of x_2 is exactly equal to the effective resistance R_{12}^{eff} between this node and the reference node. This observation, which is made in [3] in the context of estimating scalar node variables, extends to arbitrary measurement graphs, not just the simple one shown in Figure 5.

Returning to the estimation problem in Figure 5, but with node variables that are k -vectors, a simple application of (5) shows that the $k \times k$ covariance matrix Σ_2 of the optimal estimate of the k -vector x_2 is now given by

$$\Sigma_2^{-1} = P_a^{-1} + P_b^{-1}, \quad (7)$$

where P_a and P_b are the $k \times k$ covariance matrices of the two measurements. Since this formula is similar to the effective resistance formula (6), we search for a more general electrical analogy.

We now consider an abstract *generalized electrical network* in which currents, potentials, and resistors are $k \times k$ matrices. For such networks, the generalized Kirchoff's current and voltage laws are defined in the usual way, except that currents are added as matrices and voltages are subtracted as matrices. Furthermore, the generalized Ohm's law takes the matrix form

$$V_e = R_e i_e,$$

where i_e is the generalized $k \times k$ matrix current flowing through the edge e of the electrical network, V_e is the generalized $k \times k$ matrix potential drop across the edge e , and R_e is the generalized resistance on that edge. Generalized resistances are symmetric positive-definite matrices.

Generalized electrical networks share many of the properties of scalar electrical networks. In particular, given the generalized current injected into a node and extracted at another node, the generalized Kirchoff's and Ohm's laws uniquely determine all generalized currents and generalized voltage drops on the edges of a generalized electrical network. Solving for the generalized currents and voltages allows us to define the *generalized effective resistance* between two nodes as the generalized potential difference between

the two nodes when a current source equal to the $k \times k$ identity matrix is connected across them. It turns out that (7) is precisely the formula for computing the generalized effective resistance for the parallel network of two generalized resistors. Moreover, the electrical analogy for scalar measurements [3] remains valid for vector measurements when one considers generalized electrical networks. This result, which is proved in [7], is stated below.

Theorem 1

Consider a weakly connected measurement graph \mathbf{G} with a single reference node o , and construct a generalized electric network with $k \times k$ edge resistors whose resistances equal the covariance matrices of the edge-measurement errors. Then, for every node u , the $k \times k$ covariance matrix Σ_u of the BLUE estimation error of x_u is equal to the generalized effective resistance between the node u and the reference node o .

The optimal estimates and coefficient matrices that multiply the measurements to construct these estimates also have interesting interpretations in terms of electrical quantities. For example, the coefficient matrices turn out to be the generalized currents on the edges [8]. It should be noted that although measurement graphs are directed because of the need to distinguish between a measurement of $x_u - x_v$ versus a measurement of $x_v - x_u$, the direction of edges is irrelevant as far as determining the error covariance. In the context of electrical networks, the edge directions determine the signs of the currents but are irrelevant in determining generalized effective resistances.

Rayleigh's Monotonicity Law

We now discuss several results for scalar electrical networks that can be adapted to generalized electrical networks. In view of Theorem 1, these results carry over to our graph estimation problem. The first such result is *Rayleigh's monotonicity law* [12], which states that if the edge resistances in a scalar electrical network are increased, then the effective resistance between each pair of nodes in the network cannot decrease. Conversely, a decrease in edge resistances cannot lead to an increase in effective resistances. The proof of Rayleigh's monotonicity law given in [12] can be extended to the case of generalized electrical networks to show that this monotonicity also holds for generalized effective resistances [7]. Theorem 1 therefore allows us to use Rayleigh's monotonicity law for error scaling in measurement graphs.

For the problems considered here, it is convenient to consider not only increases or decreases in edge resistances but also the removal and addition of new edges, for which we need the concept of graph embedding. Given two graphs $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ and $\bar{\mathbf{G}} = (\bar{\mathbf{V}}, \bar{\mathbf{E}})$, we say that \mathbf{G} can be embedded in

$\bar{\mathbf{G}}$, or alternatively that $\bar{\mathbf{G}}$ can embed \mathbf{G} , if, ignoring the edge directions, \mathbf{G} appears as a subgraph of $\bar{\mathbf{G}}$. Figure 6 illustrates the concept of graph embedding. A precise definition of embedding is given in "Graph-Theoretic Definitions." Since edge directions play no role in the definition of embedding, they are not shown in Figure 6.

The next theorem, which is taken from [7], shows that Rayleigh's monotonicity law also holds for generalized electrical networks.

Theorem 2

Consider two generalized electrical networks with graphs $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ and $\bar{\mathbf{G}} = (\bar{\mathbf{V}}, \bar{\mathbf{E}})$ and matrix $k \times k$ edge resistances $R_e, e \in \mathbf{E}$, and $\bar{R}_{\bar{e}}, \bar{e} \in \bar{\mathbf{E}}$, respectively, and assume that the following statements hold:

- 1) \mathbf{G} can be embedded in $\bar{\mathbf{G}}$, that is, $\mathbf{G} \subset \bar{\mathbf{G}}$.
- 2) For every edge $e \in \mathbf{E}$ of \mathbf{G} , $R_e \geq \bar{R}_{\bar{e}}$, where $\bar{e} \in \bar{\mathbf{E}}$ is the corresponding edge of $\bar{\mathbf{G}}$.

Then, for every pair of nodes $u, v \in \mathbf{V}$ of \mathbf{G} ,

$$R_{u,v}^{\text{eff}} \geq \bar{R}_{\bar{u},\bar{v}}^{\text{eff}},$$

where $R_{u,v}^{\text{eff}}$ denotes the generalized effective resistance between u and v in \mathbf{G} , and $\bar{R}_{\bar{u},\bar{v}}^{\text{eff}}$ denotes the generalized effective resistance between the corresponding nodes \bar{u} and \bar{v} in $\bar{\mathbf{G}}$.

In the statement of Theorem 2 and in the sequel, given two matrices A and B , $A \geq B$ means that $A - B$ is positive semidefinite.

In terms of the original estimation problem, Rayleigh's monotonicity law leads to the conclusion that if the error covariance of one or more measurements is reduced (that is, measurements are made more accurate), then for every node variable the optimal estimation error-covariance matrix decreases, that is, the estimate becomes more accurate. In addition, when additional measurements are introduced, the

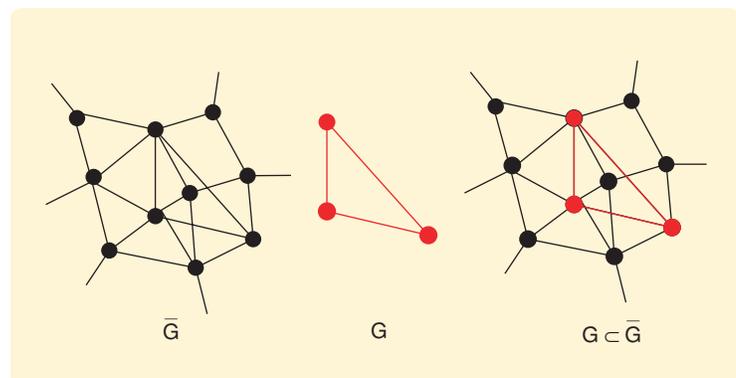


FIGURE 6 A graph-embedding example. Every node of the graph \mathbf{G} can be made to correspond in a one-to-one fashion with a subset of the nodes of $\bar{\mathbf{G}}$, and, whenever there is an edge between two nodes in \mathbf{G} , there is an edge between their corresponding nodes in $\bar{\mathbf{G}}$. Consequently, $\bar{\mathbf{G}}$ can embed \mathbf{G} , or, equivalently, \mathbf{G} can be embedded in $\bar{\mathbf{G}}$, which is denoted by $\mathbf{G} \subset \bar{\mathbf{G}}$.

resulting optimal estimates become more accurate, regardless of the noise levels in the additional measurements.

Lattices, Fuzzes, and their Generalized Effective Resistances

Scaling laws for the generalized effective resistance in graphs possessing special symmetry can be obtained easily. One such class consists of the lattice graphs (see Figure 7), which are described in “Lattices and Fuzzes.” The effective resistance in lattices with scalar resistors on every edge are studied in the literature [4], [5], and we show that similar results can be obtained for generalized lattice networks, in which edges have matrix-valued resistances. Lattices and a class of graphs derived from them, called lattice fuzzes, are useful for analyzing the scaling laws for generalized effective resistance in large graphs.

An *h-fuzz* of a graph G , where h is a positive integer, is a graph with the same set of nodes as G but with a larger set of edges [12]. In particular, the *h-fuzz* $G^{(h)}$ has an edge between every two nodes whose graphical distance in G is less than or equal to h . For the definition of graphical distance see “Graph-Theoretic Definitions.” More detail on fuzzes is provided in “Lattices and Fuzzes.” An *h-fuzz* of a graph has lower generalized effective resistances than the original graph because of Rayleigh’s monotonicity law. However, it is shown in [13] for scalar electrical networks that the effective resistance in the *h-fuzz* is lower than that in the original graph only by a constant factor. It is not difficult to see that the same result also holds for generalized networks, which is stated in the following lemma.

Lemma 1

Consider a weakly connected graph $G = (V, E)$ and let h be a positive integer. Construct two generalized electrical

networks, one by placing a matrix resistance R at every edge of a graph G and the other by placing the same matrix resistance R at every edge of its *h-fuzz* $G^{(h)}$. Then there exists $\alpha \in (0, 1]$ such that, for every pair of nodes u and v in V ,

$$\alpha R_{u,v}^{\text{eff}}(G) \leq R_{u,v}^{\text{eff}}(G^{(h)}) \leq R_{u,v}^{\text{eff}}(G),$$

where $R_{u,v}^{\text{eff}}(G)$ is the generalized effective resistance between u and v in G and $R_{u,v}^{\text{eff}}(G^{(h)})$ is the generalized effective resistance in $G^{(h)}$.

The following lemma from [7] establishes the generalized effective resistance of d -dimensional lattices and their fuzzes.

Lemma 2

Consider a generalized electrical network obtained by placing a generalized matrix resistance equal to R at every edge of the *h-fuzz* of the d -dimensional lattice, where h is a positive integer, $d \in \{1, 2, 3\}$, and R is a symmetric positive definite $k \times k$ matrix, where k is a positive integer. Then, for $d = 1, 2, 3$, there exist positive constants $\ell_d, \alpha_d, \beta_d$ such that the formulas in Table 1 hold for every pair of nodes u, v whose graphical distance from each other is larger than ℓ_d .

The result in Lemma 2 that in a one-dimensional (1D) lattice, the generalized effective resistance grows linearly with the distance between nodes can be deduced from the formula for the effective resistance of a series of resistors, which generalizes to generalized electrical networks. In two-dimensional (2D) lattices the generalized effective resistance grows with only the logarithm of the graphical distance and thus more slowly than in the 1D case. Far more surprising is what Lemma 2 says about

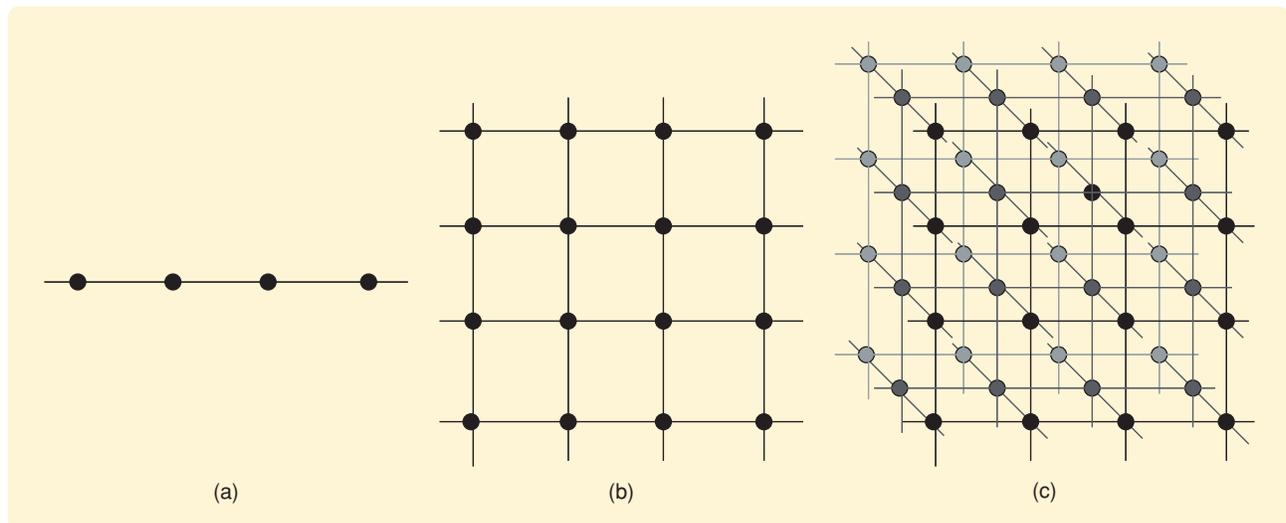


FIGURE 7 Lattices in Euclidean spaces. (a) 1D lattice Z_1 , (b) 2D lattice Z_2 , and (c) 3D lattice Z_3 . Due to their special symmetry, generalized effective resistances in lattices with unit resistance on each edge can be analytically derived. By embedding graphs in lattices or lattices in graphs, the effective resistances in graphs can be compared with those in lattices.

Lattices and Fuzzes

A d -dimensional lattice, denoted by Z_d , is a graph that has a vertex at every point in \mathbb{R}^d with integer coordinates as well as an edge between every pair of vertices whose Euclidean separation is one. Edge directions are arbitrary. Figure 7 shows 1D, 2D, and 3D lattices. Lattices have infinitely many nodes and edges and are therefore examples of *infinite graphs*. In practice, infinite graphs serve as proxies for large graphs, in the sense that, from the point of view of most nodes in a large finite graph, the graph appears to extend to infinity in all directions. Analysis of error-scaling laws in infinite graphs is easier than in finite graphs, since boundary effects can be neglected.

Given a graph G and an integer $h \geq 1$, the h -fuzz of G [12], denoted by $G^{(h)}$, is a graph that has an edge between u and v whenever the graphical distance between u and v is less than or equal to h . The directions of the edges in $G^{(h)}$ are arbitrary (see the comment following Theorem 1). Figure S2 shows a graph and its two-fuzz. Although the generalized effective resistance between two nodes in the h -fuzz is lower than that in the original graph, it is lower by a constant factor that does not depend on the distance between the nodes.

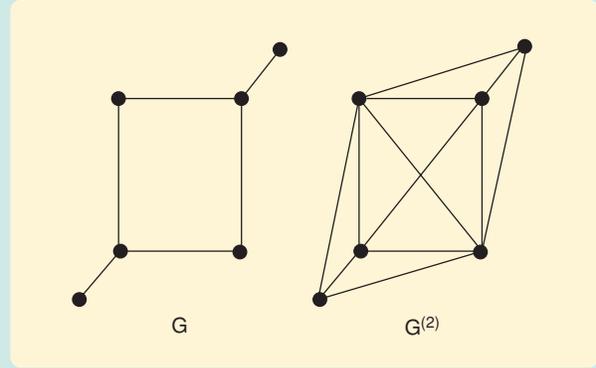


FIGURE S2 A graph G and its two-fuzz $G^{(2)}$. Every pair of nodes in G that are at a graphical distance of two have an edge between them in $G^{(2)}$. The graphical distances are reduced by a factor of h in going from a graph to its h -fuzz. Note that the edge directions are not shown since they are not important as long as we are interested only in the generalized effective resistance.

three-dimensional (3D) lattices. Specifically, in 3D, the generalized effective resistance is bounded by a constant even for arbitrarily large distances.

Error Scaling with Distance: Dense and Sparse Graphs

We now combine the tools developed so far to determine scaling laws for the estimation error variance arising from general classes of measurement graphs. Roughly speaking, our approach is the following: we determine the structural properties that a graph must satisfy so that it can either embed, or be embedded in, a lattice or the h -fuzz of a lattice. When a graph can be embedded in a lattice, Rayleigh's monotonicity law provides a lower bound on the generalized effective resistance of the graph in terms of the generalized effective resistance in the lattice, which is given by Lemma 2. Similarly, when a graph can embed a lattice, we obtain an upper bound on the generalized effective resistance.

Before we describe these concepts precisely, we ask ourselves whether there exist simple indicators of the relationship between graph structure and estimator accuracy that determine variance scaling. In the sensor networks literature, it is recognized that a higher density of nodes and edges usually leads to better estimation accuracy [11], [15]. In particular, the average number of nodes per unit area, or the average degree of a node (that is, its number of neighbors), is used to quantify the notion of denseness [11], [14], [15]. We have already seen that when one graph is embedded in another, the graph with the higher number of

edges has a lower generalized effective resistance, and, consequently, lower covariance of the estimation error. We therefore expect higher node and edge density to lead to better estimates. However, naive measures of density, such as node degree or the number of nodes and edges per unit area, turn out to be misleading predictors for how the estimation error variance scales with distance. We now present an example to motivate the search for deeper graph-structural properties to determine variance scaling with distance.

Counterexamples to Conventional Wisdom

The three graphs in Figure 8 illustrate the inadequacy of node degree as a measure of denseness. In particular, Figure 8 shows a 3-fuzz of a 1D lattice, a triangular lattice, and a 3D lattice. The generalized effective resistance

TABLE 1 Effective resistance for lattices and their fuzzes. The graphical distance between the nodes u and v in the lattice Z_d is denoted by $d_{Z_d}(u, v)$. Generalized effective resistance grows linearly with distance in the 1D lattice, logarithmically in the 2D lattice, and remains bounded by a constant independent of distance in the 3D lattice.

Graph	Generalized effective resistance $R_{u,v}^{\text{eff}}$ between u and v
 $Z_1^{(h)}$	$\alpha_1 d_{Z_1}(u, v) R \leq R_{u,v}^{\text{eff}} \leq \beta_1 d_{Z_1}(u, v) R$
 $Z_2^{(h)}$	$\alpha_2 \log(d_{Z_2}(u, v)) R \leq R_{u,v}^{\text{eff}} \leq \beta_2 \log(d_{Z_2}(u, v)) R$
 $Z_3^{(h)}$	$\alpha_3 R \leq R_{u,v}^{\text{eff}} \leq \beta_3 R$

scales linearly with distance in the 3-fuzz of the 1D lattice, scales logarithmically with distance in the triangular lattice, and is globally bounded with respect to distance in the 3D lattice, even though each of these graphs has the same degree, namely, six. The statements about the generalized effective resistances in the 3-fuzz of the 1D lattice and in the 3D lattice follow from Lemma 2. The growth of generalized effective resistance in the triangular lattice is discussed below.

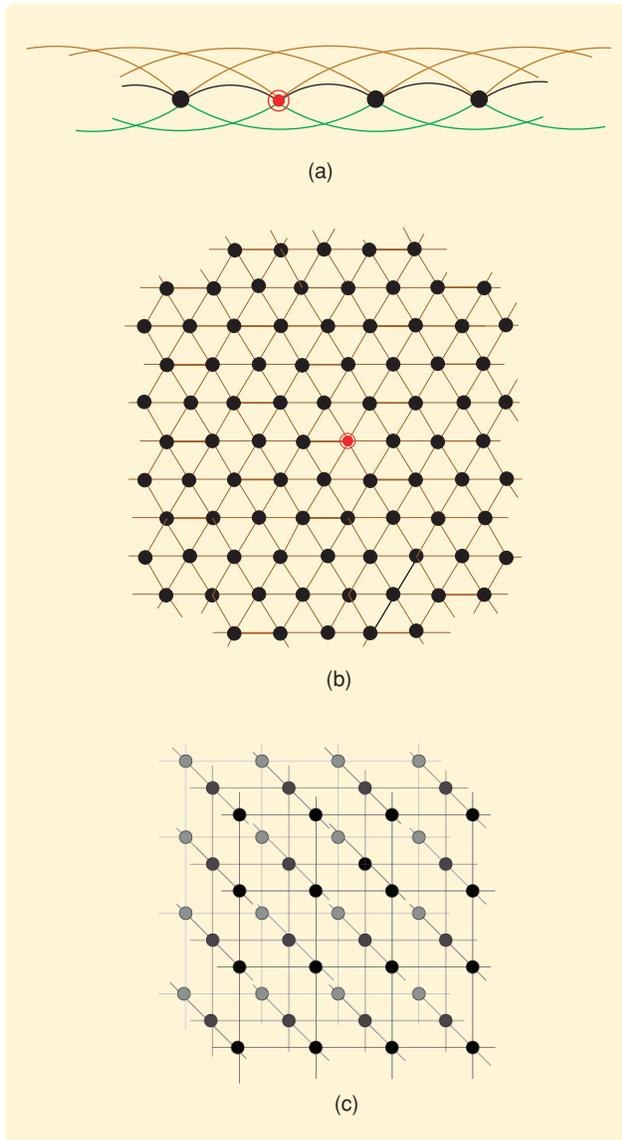


FIGURE 8 Measurement graphs. (a) Three-fuzz of a 1D lattice, (b) triangular lattice, and (c) 3D lattice. In all of these graphs, every node has the same degree, but the graphs have quite different variance growth rates with distance. The variance grows linearly with distance in the three-fuzz of the 1D lattice, as guaranteed by Lemma 2. On the other hand, in the triangular lattice the variance grows as the logarithm of distance, while, in the 3D lattice, the variance is bounded by a constant regardless of the distance between nodes. The last two statements follow from Theorem 4.

Graph Drawings

We now derive conditions for embedding a graph in a lattice, and vice versa, by looking at drawings of the graph. A drawing of a graph $G = (\mathbf{V}, \mathbf{E})$ is a mapping of its nodes to points in a Euclidean space \mathbb{R}^d , which can formally be described by a function $f: \mathbf{V} \rightarrow \mathbb{R}^d$. Figure 9 shows two different drawings of the same graph. From a graph-theoretic point of view, the two graphs are identical because they have the same nodes and edges. However, the two graphs are drawn differently. Further discussion on graph drawings and its relevance to sensor networks can be found in “Drawing Graphs.” Although every graph can be drawn in many different ways, we are interested in drawings that accurately reflect denseness and sparseness of the graph, which we call dense and sparse drawings. We show below that the existence of such drawings completely characterizes whether or not the graph (or a fuzz of it) can be embedded in a lattice, or vice versa.

For a particular drawing of a graph, the Euclidean distance between nodes induced by the drawing is the Euclidean distance between the nodes in the drawing. More precisely, given two nodes $u, v \in \mathbf{V}$, the *Euclidean distance between u and v induced by the drawing $f: \mathbf{V} \rightarrow \mathbb{R}^d$* is defined by

$$d_f(u, v) := \|f(v) - f(u)\|,$$

where $\|\cdot\|$ denotes the Euclidean norm on \mathbb{R}^d . Note that the Euclidean distance between two nodes depends on the drawing and can be different from the graphical distance. It is important to emphasize that the definition of drawing does not require edges to not intersect, and therefore every graph has a drawing in every Euclidean space.

For a sensor network, there is a *natural drawing* of its measurement graph obtained by associating each node to its position in 1D, 2D, or 3D Euclidean space. In reality, all sensor networks are situated in 3D space. However, it is often more natural to draw these networks on a 2D Euclidean

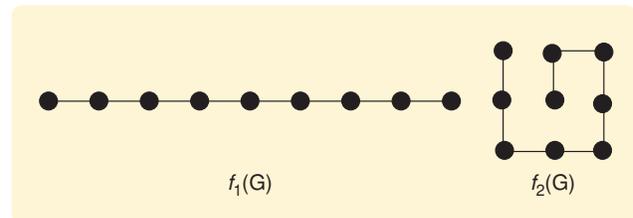


FIGURE 9 Two drawings of the same graph G . Drawing $f_1(G)$ is a 1D drawing, whereas $f_2(G)$ is a 2D drawing. These drawings show that the same graph may appear quite different when drawn in different Euclidean spaces. To determine the estimation error scaling laws in a graph, we want to compare the graph with the d -dimensional lattice, which is best done by comparing different drawings of the graph in \mathbb{R}^d with the natural drawing of the d -dimensional lattice. In fact, the notions of denseness and sparseness of graphs described in this article, which determine the scaling of estimation error, are based on graph drawings.

space if one dimension (for example, height) does not vary much from node to node, or is irrelevant. In yet another situation, such as shown in Figure 3, one can draw the graph in 1D since the nodes essentially form a chain even though the nodes are situated in 3D space. For natural drawings, the Euclidean distance induced by the drawing is, in general, a much more meaningful notion of distance than the graphical distance. In this article, we see that the Euclidean distance induced by appropriate drawings provides the relevant measure of distance for determining scaling laws for error variances.

Measures of Graph Denseness and Sparseness

For a drawing f and induced Euclidean distance d_f of a graph $G = (V, E)$, four parameters can be used to characterize graph denseness and sparseness. The term *minimum node distance* s denotes the minimum Euclidean distance between the drawing of two nodes defined by

$$s := \inf_{\substack{u, v \in V \\ v \neq u}} d_f(u, v).$$

The term *maximum connected range* r denotes the Euclidean length of the drawing of the longest edge

$$r := \sup_{(u, v) \in E} d_f(u, v).$$

The term *maximum uncovered diameter* γ denotes the diameter of the largest open ball that can be placed in \mathbb{R}^d with no drawing of a node inside it, that is,

$$\gamma := \sup\{\delta : \text{there exists } B_\delta \text{ such that } f(u) \notin B_\delta, \text{ for every } u \in V\},$$

where B_δ is a ball in \mathbb{R}^d with diameter δ . Finally, the *asymptotic distance ratio* ρ is the largest asymptotic ratio between the graphical and the Euclidean distance between two nodes defined by

$$\rho := \liminf_{n \rightarrow \infty} \left\{ \frac{d_f(u, v)}{d_G(u, v)} : u, v \in V \text{ and } d_G(u, v) \geq n \right\},$$

where $d_G(u, v)$ denotes the graphical distance between u and v in the graph G . Essentially ρ provides a lower bound for the ratio between the Euclidean and the graphical distance for nodes that are far apart. Figure 10 shows the drawing of a graph and the corresponding parameters s , r , γ , and ρ .

Dense Graphs

The drawing of a graph for which the maximum uncovered diameter γ is finite and the asymptotic distance ratio ρ is positive is a *dense drawing*. A graph G is *dense in* \mathbb{R}^d if there exists a dense drawing of G in \mathbb{R}^d . Intuitively, these

drawings are dense in the sense that the nodes can cover \mathbb{R}^d without leaving large holes between them, and the

Drawing Graphs

In graph theory, a graph is treated purely as a collection of nodes connected by edges, without any regard to the geometry determined by the nodes' locations. However, in sensor network problems there is an underlying geometry for the measurement graph since this graph is tightly related to the physical locations of the sensor nodes. For example, a pair of nodes from a sensor network typically has an edge if the two nodes are within some sensing or communication range of each other. Although this range can be defined in a complicated fashion (not just determined by the Euclidean distance), the geometric configuration of nodes in Euclidean space plays a key role in determining the measurement graph. The geometric features of a graph are best captured by its drawings, which are mappings of its nodes to points in Euclidean spaces.

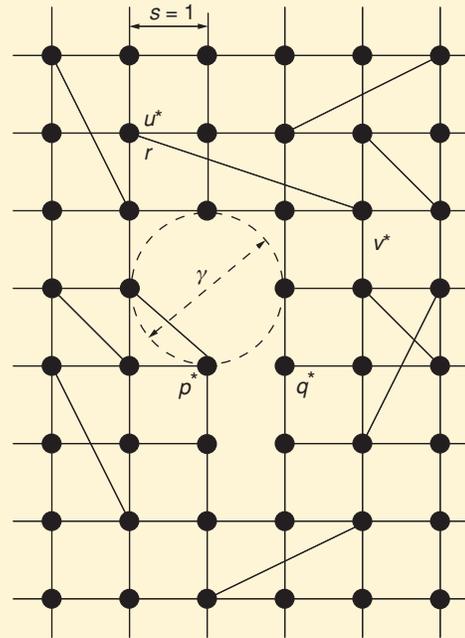


FIGURE 10 A drawing of a graph in 2D Euclidean space, and the corresponding denseness and sparseness parameters. Since the minimal distance between any two nodes is one, the minimum node distance is $s = 1$. Since the longest edge is between u^* and v^* , of length $\sqrt{10}$, the maximum connected range is $r = \sqrt{10}$. The diameter of the largest ball that can fit inside the drawing without enclosing any node is two, the maximum uncovered diameter is thus $\gamma = 2$. The minimal ratio between the Euclidean and graphical distances between a pair of nodes is achieved by the pair p^*, q^* ; hence the asymptotic distance ratio is $\rho = d_f(p^*, q^*)/d_G(p^*, q^*) = 1/5$.

graph has sufficiently many edges so that a small Euclidean distance between two nodes in the drawing guarantees a small graphical distance between them. In particular, it is shown in [8] that, for every dense drawing f of a graph \mathbf{G} , there exist finite constants α, β such that, for every pair of nodes u, v in \mathbf{G} ,

$$d_{\mathbf{G}}(u, v) \leq \alpha d_f(u, v) + \beta. \quad (8)$$

Using the natural drawing of a d -dimensional lattice, it follows that the d -dimensional lattice is dense in \mathbb{R}^d . One can also show that the d -dimensional lattice is not dense in $\mathbb{R}^{\bar{d}}$ for all $\bar{d} > d$. For example, no dense drawing of a 2D lattice in the 3D Euclidean space is possible.

Sparse Graphs

Graph drawings for which the minimum node distance s is positive and the maximum connected range r is finite are called *civilized drawings* [12]. Intuitively, one can draw the edges with finite lengths while maintaining a minimum distance between the nodes. We say that a graph \mathbf{G} is sparse in \mathbb{R}^d if it can be drawn in a civilized manner in the d -dimensional Euclidean space. For example, we can conclude from the natural drawing of a d -dimensional lattice that the d -dimensional lattice is sparse in \mathbb{R}^d . In fact, every h -fuzz of a d -dimensional lattice is sparse in \mathbb{R}^d . However, for all $\bar{d} < d$, no d -dimensional lattice can be drawn in a civilized way in $\mathbb{R}^{\bar{d}}$. For example, the 3D lattice is not sparse in \mathbb{R}^2 .

The notions of graph sparseness and denseness are interesting mainly for infinite graphs because every finite graph is sparse in all Euclidean spaces \mathbb{R}^d for $d \geq 1$, and no finite graph can be dense in any Euclidean space \mathbb{R}^d for $d \geq 1$. The reason is that a drawing of a finite graph that does not place nodes on top of each other necessarily has a positive minimum node distance and a finite maximum connected range (from which sparseness follows) and it is not possible to achieve a finite maximum uncovered diameter with a finite number of nodes (from which lack of denseness follows). However, infinite graphs serve as proxies for large graphs in the sense that, from the perspective of most nodes, a large graph appears to extend to infinity in all directions. Furthermore, it is often easier to examine scaling laws for the effective resistance in infinite graphs since boundary effects are weaker than in finite graphs. For this reason we derive results for infinite graphs and draw heuristic conclusions about finite graphs. As long as we are interested in nodes that are not close to the boundary, such conclusions drawn for infinite graphs are applicable to large graphs.

Sparseness, Denseness, and Embeddings

The notions of sparseness and denseness introduced above are useful because they characterize the classes of graphs that can embed or be embedded in lattices, for which

Lemma 2 provides the precise scaling laws for the generalized effective resistance.

Theorem 3

Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be a graph without multiple edges between the same pair of nodes.

- 1) \mathbf{G} is sparse in \mathbb{R}^d if and only if \mathbf{G} can be embedded in an h -fuzz of a d -dimensional lattice. Formally, \mathbf{G} is sparse in \mathbb{R}^d if and only if there exists a positive integer h such that $\mathbf{G} \subset \mathbf{Z}_d^{(h)}$.
- 2) \mathbf{G} is dense in \mathbb{R}^d if and only if i) the d -dimensional lattice can be embedded in an h -fuzz of \mathbf{G} for some positive integer h , and ii) every node of \mathbf{G} that is not mapped to a node of \mathbf{Z}_d is at a uniformly bounded graphical distance from a node that is mapped to \mathbf{Z}_d . More precisely, \mathbf{G} is dense in \mathbb{R}^d if and only if there exist positive integers h, c such that $\mathbf{G}^{(h)} \supset \mathbf{Z}_d$ and for every $u \in \mathbf{V}$ there is a $\bar{u} \in \mathbf{V}_{\text{lat}}(\mathbf{G})$ such that $d_{\mathbf{G}}(u, \bar{u}) \leq c$, where $\mathbf{V}_{\text{lat}}(\mathbf{G})$ denotes the set of nodes of \mathbf{G} that are mapped to nodes of \mathbf{Z}_d .

The first statement of Theorem 3 is essentially taken from [12], while the second statement is a consequence of results in [7] and [8]. The condition of “no multiple edges between two nodes” is not restrictive for the estimation problems because the generalized effective resistance between any two nodes in a graph does not change if we replace a set of multiple, parallel edges between two nodes by a single edge with a generalized resistance equal to the generalized effective resistance of those parallel edges.

Scaling Laws for the Estimation-Error Variance

We now characterize the scaling laws of the estimation error variance in terms of the denseness and sparseness properties of the measurement graph. The following theorem characterizes the scaling laws by combining the electrical analogy Theorem 1, Rayleigh’s monotonicity law, the lattice generalized effective resistance Lemma 2, and the lattice embedding Theorem 3. This result follows from the results established in [7] and their extensions in [8].

Theorem 4

Consider a measurement graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with a single reference node $o \in \mathbf{V}$, in which there exist symmetric positive-definite matrices P_{\min} and P_{\max} such that, for every $e \in \mathbf{E}$, the covariance matrices of the measurement errors satisfy $P_{\min} \leq P_e \leq P_{\max}$. Then, for $d = 1, 2, 3$, there exist constants $\ell_d, \alpha_d, \beta_d > 0$ such that the formulas in Table 2 hold for every node u whose Euclidean distance from the reference node o is larger than ℓ_d , whenever the graph is sparse or dense in the d -dimensional Euclidean space.

At this point it is easy to check that the triangular lattice in Figure 8 is both sparse and dense in 2D, which validates the statement in the section “Counterexamples to Conventional Wisdom” that the effective resistance in the triangular lattice grows as the logarithm of distance.

In practice, sensor networks are typically sparse and dense in appropriate Euclidean spaces, as seen by considering the natural drawing of the network. All natural drawings of sensor networks are sparse in 3D, since the only requirements for sparseness are that nodes not lie on top of each other and that edges have finite length. When a sensor network is deployed in a 2D domain or when the third physical dimension is irrelevant, again the natural drawing is likely to be sparse in 2D space for the same reasons. Furthermore, a dense graph in \mathbb{R}^2 is generated when nodes are deployed in the plane in such a way that every node communicates with all of its neighbors within a range twice as large as the diameter of the largest ball that contains no nodes. A similar construction can be used to generate a graph that is dense in \mathbb{R}^3 by deploying nodes in 3D space.

DISTRIBUTED COMPUTATION

We now answer the first question concerning the use of local information to compute the optimal estimate of the node variables in a distributed way. We show that this objective is indeed feasible and present two distributed asynchronous algorithms that achieve this goal. The algorithms are iterative, whereby every node starts with an arbitrary initial guess for its variable and successively improves its estimate by using the measurements on the edges incident on it as well as the estimates of its neighbors. The algorithms are guaranteed to converge to the optimal estimate as the number of iterations goes to infinity. Moreover, these algorithms are robust to link failures, and they converge to the optimal estimate even in the presence of faulty communication links, as long as certain mild conditions are satisfied.

The starting point for the construction of the algorithm is the recognition that the optimal estimate given by (4) is the unique solution to the system of linear equations

$$\mathcal{L}\hat{\mathbf{x}}^* = \mathbf{b}, \quad (9)$$

where \mathcal{L} and \mathbf{b} are defined in (4). We seek iterative algorithms to compute the solution to (9) subject the following constraints:

- 1) At every iteration, each node is allowed to broadcast a message to all of its one-hop neighbors.
- 2) Each node is allowed to perform computations involving only variables that are local to the node or that were previously obtained from its neighbors.

The “one-hop neighbors” of a node u is the set of nodes in the measurement graph \mathbf{G} with which u has an edge. By letting the nodes exchange information with their one-hop neighbors, we allow two nodes to receive each other’s messages if the measurement graph \mathbf{G} has an edge between them in either direction. In short, we implicitly assume bidirectional communication.

Jacobi Algorithm

Consider a node u with unknown node variable x_u and imagine for a moment that the node variables of all of the neighbors of u are exactly known and available to u . In this case, the node u can compute its optimal estimate by using the measurements between u and its one-hop neighbors. This estimation problem is no different from the original problem, except that it is defined over the much smaller graph $\mathbf{G}_u(1) = (\mathbf{V}_u(1), \mathbf{E}_u(1))$, whose nodes include u and its one-hop neighbors and whose edge set $\mathbf{E}_u(1)$ consists of only the edges between u and its one-hop neighbors. We call $\mathbf{G}_u(1)$ the *one-hop subgraph of \mathbf{G} centered at u* . Since we assume that the node variables of the neighbors of u are exactly known, all of these nodes should be understood as references. The *Jacobi algorithm* for computing the optimal estimates of the node variables is an iterative algorithm that operates as follows:

- 1) Each node $u \in \mathbf{V}$ picks an arbitrary initial estimate $\hat{x}_v^{(0)}$ for the node variables x_v of each of its one-hop neighbors $v \in \mathbf{V}_u(1)$. These estimates need not be consistent across different nodes.
- 2) At the i th iteration, each node $u \in \mathbf{V}$ assumes that its current estimate $\hat{x}_v^{(i)}$ for the node variable x_v of each of its neighbors $v \in \mathbf{V}_u(1)$ is correct and solves the corresponding estimation problem associated with the one-hop subgraph $\mathbf{G}_u(1)$. The corresponding

TABLE 2 Scaling laws for the covariance matrix of the estimation error for a measurement graph \mathbf{G} that is dense or sparse in d dimensions. $d_f(u, o)$ denotes the Euclidean distance between node u and the reference node o for the drawing f that establishes the graph’s sparseness or denseness.

Euclidean space		Covariance matrix Σ_u of the estimation error of x_u if f is a sparse drawing of \mathbf{G} in \mathbb{R}^d	Covariance matrix Σ_u of the estimation error of x_u if f is a dense drawing of \mathbf{G} in \mathbb{R}^d
 \mathbb{R}		$\alpha_1 d_f(u, o) P_{\min} \leq \Sigma_u$	$\Sigma_u \leq \beta_1 d_f(u, o) P_{\max}$
 \mathbb{R}^2		$\alpha_2 \log(d_f(u, o)) P_{\min} \leq \Sigma_u$	$\Sigma_u \leq \beta_2 \log(d_f(u, o)) P_{\max}$
 \mathbb{R}^3		$\alpha_3 P_{\min} \leq \Sigma_u$	$\Sigma_u \leq \beta_3 P_{\max}$

estimate $\hat{x}_u^{(i+1)}$ turns out to be the solution to the system of linear equations

$$\left(\sum_{e \in \mathbf{E}_u} P_e^{-1} \right) \hat{x}_u^{(i+1)} = \sum_{e \in \mathbf{E}_u} P_e^{-1} \left(\hat{x}_{v_e}^{(i)} + a_{ue} \zeta_e \right), \quad (10)$$

where v_e denotes the one-hop neighbor that shares the edge e with u , and a_{ue} is the (u, e) entry of the incidence matrix A . The node u then broadcasts the new estimate $\hat{x}_u^{(i+1)}$ to all of its neighbors.

- 3) At the end of the i th iteration, each node listens for the broadcasts from its one-hop neighbors, which are used to update the node-variable estimate $\hat{x}_v^{(i+1)}$ for each of its neighbors $v \in \mathbf{V}_u(1)$. Once all updates are received, a new iteration can start.

These iterations can be terminated at a node when the change in its recent estimate is seen to be lower than a pre-specified threshold value or a pre-specified maximum number of iterations are completed. "Jacobi Iteration" shows the relevant equations for one iteration of the Jacobi algorithms applied to the measurement graph shown in Figure S1 of "A Graph Estimation Example."

The iterative algorithm described by (10) can be viewed as the Jacobi algorithm for solving linear equations. Iterative techniques for solving linear equations

have a rich history, and a host of iterative methods have been developed for specific applications, each with its own particular advantages and disadvantages. For the optimal estimation problem, the Jacobi algorithm has several benefits, namely, it is scalable, it converges to the optimal estimate under mild conditions, and it is robust to temporary link failures [16]. However, its weakness lies in its slow convergence rate.

Overlapping Subgraph Estimator Algorithm

The overlapping subgraph estimator (OSE) algorithm achieves faster convergence than Jacobi, while retaining its scalability and robustness properties. The OSE algorithm can be thought of as an extension of the Jacobi algorithm, in which individual nodes utilize larger subgraphs to improve their estimates. To understand how, suppose that each node broadcasts to its one-hop neighbors not only its current estimate, but also all of the latest estimates that it received from its one-hop neighbors. In the absence of drops, at the i th iteration step each node has the estimates $\hat{x}_v^{(i)}$ for its one-hop neighbors as well as the (older) estimates $\hat{x}_v^{(i-1)}$ for its two-hop neighbors, that is, the nodes at a graphical distance equal to two.

Under this information exchange scheme, at the i th iteration each node u has estimates of all of the node

The Jacobi Iteration

To describe the Jacobi iterations in a measurement graph, we consider the measurement graph in Figure S1 of "A Graph Estimation Example." The one-hop subgraph $\mathbf{G}_4(1)$ of node 4 for the measurement graph in Figure S1 is shown in Figure S3. In this subgraph, the measurement model (3) for the only unknown variable x_4 , when x_2 and x_3 are taken as references, is

$$\underbrace{\begin{bmatrix} z_3 \\ z_5 \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}}_{\mathbf{A}_r^T} \underbrace{\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}}_{\mathbf{x}_r} + \underbrace{\begin{bmatrix} -I \\ I \end{bmatrix}}_{\mathbf{A}_b^T} x_4 + \underbrace{\begin{bmatrix} \varepsilon_2 \\ \varepsilon_5 \end{bmatrix}}_{\boldsymbol{\varepsilon}}.$$

The corresponding optimal estimate (4) when all measurement covariance matrices are equal to the identity matrix is given by

$$\underbrace{(\mathbf{A}_b \mathbf{A}_b^T)^{-1}}_{\mathbf{c}} \underbrace{\mathbf{A}_b (\mathbf{z} - \mathbf{A}_r^T \mathbf{x}_r)}_{\mathbf{b}} = \frac{1}{2} (x_2 - z_3 + x_3 + z_5).$$

The Jacobi iteration for node 4 is

$$\hat{x}_4^{(i+1)} = \frac{1}{2} (\hat{x}_2^{(i)} - z_3 + \hat{x}_3^{(i)} + z_5).$$

A similar construction based on the one-hop subgraphs centered at nodes 2 and 3 leads to update equations for estimates of x_2 and x_3 given by

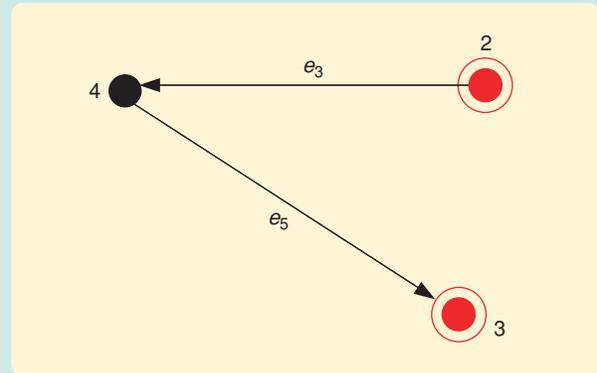


FIGURE S3 The one-hop subgraph $\mathbf{G}_4(1)$ for the measurement graph \mathbf{G} in Figure S1. In each time step of the Jacobi algorithm, node 4 estimates its own variable by solving the optimal estimation problem for this subgraph taking the current estimates of x_2 and x_3 as reference variables.

$$\begin{aligned} \hat{x}_2^{(i+1)} &= \frac{1}{4} (\hat{x}_4^{(i)} + \hat{x}_3^{(i)} + \zeta_3 + \zeta_4 - \zeta_1 - \zeta_2), \\ \hat{x}_3^{(i+1)} &= \frac{1}{2} (\hat{x}_2^{(i)} + \hat{x}_4^{(i)} - \zeta_4 - \zeta_5). \end{aligned}$$

The reference node, which is node 1, is assumed to be at the origin, and thus x_1 does not appear in the equations.

variables of the nodes in the set $\mathbf{V}_u(2)$ consisting of all of its one-hop and two-hop neighbors. In the OSE algorithm, each node updates its estimate using the *two-hop subgraph* $\mathbf{G}_u(2) = (\mathbf{V}_u(2), \mathbf{E}_u(2))$ centered at u , with edge set $\mathbf{E}_u(2)$ consisting of all of the edges of the original graph \mathbf{G} that connect elements of $\mathbf{V}_u(2)$. For this estimation problem, node u takes as references the node variables of its two-hop neighbors. The gain in convergence speed with respect to the Jacobi algorithm comes from the fact that the two-hop subgraph $\mathbf{G}_u(2)$ contains more edges than the one-hop subgraph $\mathbf{G}_u(1)$. The OSE algorithm can be summarized as follows:

- 1) Each node $u \in \mathbf{V}$ picks an arbitrary initial estimate $\hat{x}_v^{(-1)}$ of the node variable x_v of each of its two-hop neighbors $v \in \mathbf{V}_u(2) \setminus \mathbf{V}_u(1)$. These estimates need not be consistent across different nodes.
- 2) At the i th iteration, each node $u \in \mathbf{V}$ assumes that the estimates $\hat{x}_v^{(i-2)}$ of the node variables x_v of its two-hop neighbors that it received through its one-hop neighbors are correct and solves the corresponding optimal estimation problem associated with the two-hop subgraph $\mathbf{G}_u(2)$. In particular, each node u solves the linear equations $\mathcal{L}_{u,2}\mathbf{y}_u = \mathbf{b}_u$, where \mathbf{y}_u is a vector of node variables that correspond to the nodes in its one-hop subgraph $\mathbf{G}_u(1)$, and $\mathcal{L}_{u,2}, \mathbf{b}_u$ are defined for the subgraph $\mathbf{G}_u(2)$ as \mathcal{L}, \mathbf{b} are for \mathbf{G} in (9). After this computation, node u updates its estimate as $\hat{x}_u^{(i+1)} \leftarrow \lambda y_u + (1 - \lambda)\hat{x}_u^{(i)}$, where $0 < \lambda \leq 1$ is a prespecified design parameter and y_u is the variable in \mathbf{y}_u that corresponds to x_u . The new estimate $\hat{x}_u^{(i+1)}$ as well as the estimates $\hat{x}_v^{(i)}$ previously received from its one-hop neighbors $v \in \mathbf{V}_u(1)$ are then broadcast to all of its one-hop neighbors.
- 3) At the end of the i th iteration, each node u then listens for the broadcasts from its one-hop neighbors and uses them to update its estimates for the node variables of all of its two-hop neighbors. Once all updates are received a new iteration can start.

As in the case of the Jacobi algorithm, the termination criteria vary depending on the application, and nodes use measurements and covariances obtained initially for all future time. Figure 11 shows a two-hop subgraph used by the OSE algorithm.

The previous description assumes that communication is synchronous and that each node receives broadcasts from all of its neighbors. For the OSE algorithm to work under imperfect synchronization and link failures, a node may have to proceed to a new iteration step before receiving broadcast messages from all of its neighbors or after receiving multiple messages from the same neighbor. A timeout

mechanism can be used for this purpose, in which each node resets a timer as it broadcasts its most recent estimates. When this timer reaches a prespecified timeout value, the node initiates a new iteration, regardless of whether or not it received messages from all of its one-hop neighbors. If a message is not received from one of its neighbors, the node uses the data most recently received from that neighbor for the next iteration.

One can also design an *h-hop OSE* algorithm by letting every node utilize an *h-hop* subgraph centered at itself, where h is an (small) integer. The resulting algorithm is a straightforward extension of the two-hop OSE just described, except that at every iteration, individual nodes have to transmit to their neighbors larger amounts of data than in two-hop OSE, potentially requiring multiple packet transmissions at each iteration. In practice, this added communication cost limits the allowable value of h .

The following result establishes the correctness of the OSE algorithm [17].

Theorem 5

When the total number of consecutive iterations for which a node does not receive information from one of its neighbors is uniformly upper-bounded by a constant ℓ_f , the OSE algorithm is guaranteed to converge if all of the covariance matrices P_e , where $e \in \mathbf{E}$, are either all equal or are all diagonal (but not necessarily equal).

It should be noted that the requirement that the matrices P_e be equal or diagonal to establish convergence in the presence of drops can probably be relaxed. In the simulations described below, this assumption is violated but the algorithm is seen to converge.

Flagged Initialization

The performance of the basic Jacobi or OSE algorithms can be further improved by providing them with better initializations, which does not require more communication or computation. After the deployment of the network, the

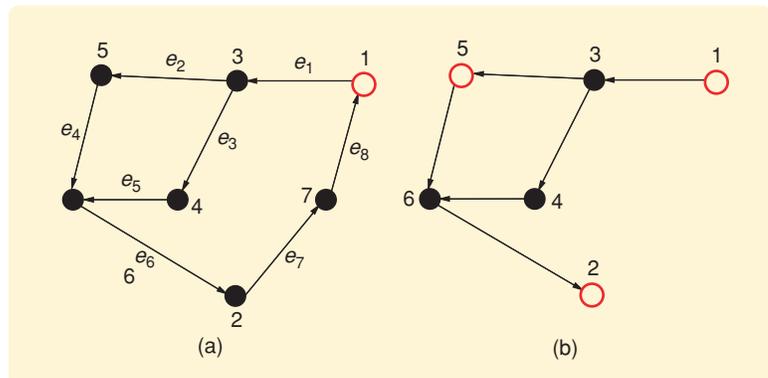


FIGURE 11 (a) A measurement graph \mathbf{G} with node 1 as reference and (b) a two-hop subgraph $\mathbf{G}_4(2)$ centered at node 4. While running the overlapping subgraph estimator algorithm, node 4 treats nodes 1, 5, and 2 as reference nodes in the subgraph $\mathbf{G}_4(2)$ and solves for the unknowns x_3, x_4 , and x_6 .

reference nodes initialize their estimates to their known values, but all other nodes initialize their estimates to ∞ , which serves as a flag to declare that these nodes do not have a good estimate of their variables. Subsequently, in its estimate updates, each node includes in its one- or two-hop subgraph only those nodes that have finite estimates. If none of their neighbors has a finite estimate, then the node keeps its estimate at ∞ . In the beginning, only the references have a finite estimate. In the first iteration, the one-hop neighbors of the references can compute finite estimates, whereas in the second iteration, the two-hop neighbors of the references can also obtain finite estimates and so forth until all nodes have finite estimates. Flagged initialization affects only the initial stage of the algorithms, and thus does not affect their convergence properties.

SIMULATIONS

In this section, we present numerical simulations to illustrate the performance of the OSE algorithm. In these simulations the node variables represent the physical position of sensors in the plane. All simulations refer to a network with 200 nodes that are randomly placed in the unit square (see Figure 12). Node 1, placed at the origin, is chosen as the single reference node. Pairs of nodes separated by a distance smaller than $r_{\max} := 0.11$ are allowed to have noisy measurements of each others' relative range and bearing (see Figure 1). The range measurements are corrupted with

zero-mean additive Gaussian noise with standard deviation $\sigma_r = 0.15 r_{\max}$, and the angle measurements are corrupted with zero-mean additive Gaussian noise with standard deviation $\sigma_\theta = 10^\circ$. Assuming that the range and bearing measurement errors are independent and have variances independent of distance, consider a noisy measurement (r, θ) of true range and angle (r_0, θ_0) . Then it can be shown that the covariance matrix of the measurement $\zeta_{u,v} = [r \cos \theta, r \sin \theta]^T$ is given approximately by

$$P_{u,v} = \begin{bmatrix} y_0^2 \sigma_\theta^2 + \sigma_r^2 \cos^2 \theta_0 & -x_0 y_0 \sigma_\theta^2 + \frac{\sigma_r^2}{2} \sin(2\theta_0) \\ -x_0 y_0 \sigma_\theta^2 + \frac{\sigma_r^2}{2} \sin(2\theta_0) & x_0^2 \sigma_\theta^2 + \sigma_r^2 \sin^2 \theta_0 \end{bmatrix},$$

where $x_0 = r_0 \cos \theta_0$ and $y_0 = r_0 \sin \theta_0$. Assuming that the scalars σ_r, σ_θ are provided a priori to the nodes, a node can estimate this covariance by using the measured r and θ in place of their unknown true values. Since the covariances are not diagonal and since distinct measurements have distinct covariances, this example does not satisfy the assumptions for which the OSE algorithm is guaranteed to converge. The locations estimated by the centralized optimal estimator are shown in Figure 12, together with the true locations.

Figure 13(a) compares the normalized error as a function of iteration number for the Jacobi and OSE algorithms. Two versions of the OSE are tested, namely, OSE two-hop and three-hop. The parameter λ for OSE is chosen arbitrarily as 0.9. The straight lines in the log-scaled graph reflect the exponential convergence of both algorithms as well as the faster convergence rate of the OSE algorithm compared to Jacobi. Figure 13 also shows the dramatic improvement achieved with the flagged initialization scheme. With flagged initialization, the two-hop OSE algorithm can estimate the node positions within 3% of the optimal estimate after only nine iterations. Figure 13(b) shows the performance of the two-hop OSE algorithm with flagged initialization under two different link-failure probabilities. Every link is made to fail independently with probability p_f . Not surprisingly, higher failure rates result in slower convergence.

OSE Versus Jacobi

In Figure 13, the OSE algorithm exhibits faster convergence than the Jacobi algorithms. However, faster convergence is achieved at the

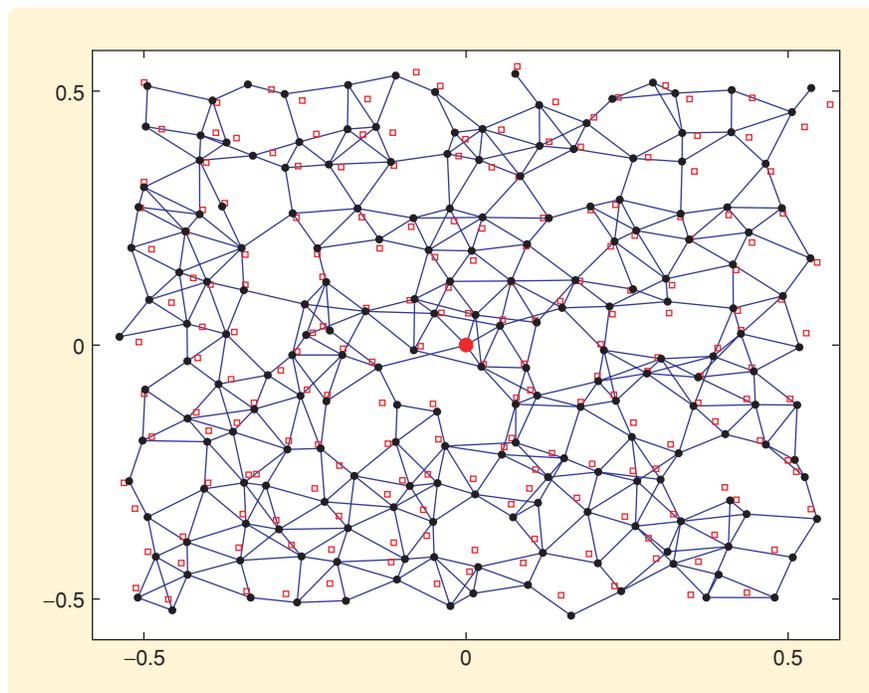


FIGURE 12 A sensor network with 200 nodes distributed randomly in a unit square area. The edges of the measurement graph are shown as line segments connecting the node positions, which are shown as black dots. Two nodes with an edge between them are provided with a measurement of their relative positions in the plane. The red squares are the positions estimated by the (centralized) optimal estimator. A single reference node is located at the origin.

expense of each node sending more data to its one-hop neighbors because each node broadcasts its own estimate as well as the estimates previously received from its one-hop neighbors. Hence the messages needed by OSE are d times longer than the messages required by Jacobi, where d denotes the node degree. One may then ask whether there is a significant advantage to using the OSE algorithm.

Energy consumption in wireless communication networks depends in a complicated way on radio hardware, underlying physical and medium access control layer protocols, network topology, and a host of other factors. Due to the overhead introduced by these factors, sending a short message offers no advantage in terms of energy consumption over sending a somewhat longer message [18]. In fact, transmitted energy per bit in a packet decreases monotonically up to the maximum payload [19]. One of the main findings in [20] is that, in highly contentious networks, transmitting large payloads is more energy efficient. Therefore, communication overhead generally favors the transmission of fewer long messages over many short ones. As a result, sending a packet may cost almost as much energy as sending a packet many times longer. In such cases, the OSE algorithm is advantageous compared to the Jacobi algorithm because OSE requires a smaller number of iterations—and therefore a smaller number of messages—compared to Jacobi to achieve a desired error tolerance, resulting in lower energy consumption and

increased network life. In [17], simulations with a simple model of energy consumption shows that OSE can reduce energy consumption by a factor of two or more compared to the Jacobi algorithm while achieving the same accuracy.

CONCLUSIONS

Large-scale sensor networks give rise to estimation problems that have a rich graphical structure. We studied one of these problems in terms of how such an estimate can be efficiently computed in a distributed manner as well as how the quality of an optimal estimate scales with the size of the network. Two distributed algorithms are presented to compute the optimal estimates that are scalable and robust to communication failures. In designing these algorithms, we found the literature on parallel computation to be a rich source of inspiration.

In answer to the second question, structural properties that dictate how variance scales with distance are determined. The answer to the variance-scaling question results in two classes of graphs, namely, dense and sparse, for which we can find upper and lower bounds on the variance growth with distance. The variance-scaling question was answered by exploiting the analogy between estimation error covariance and generalized effective resistance. The monograph by Doyle and Snell [12] in particular helped us immensely by bringing to our attention the notion of bounding effective resistance by embedding.

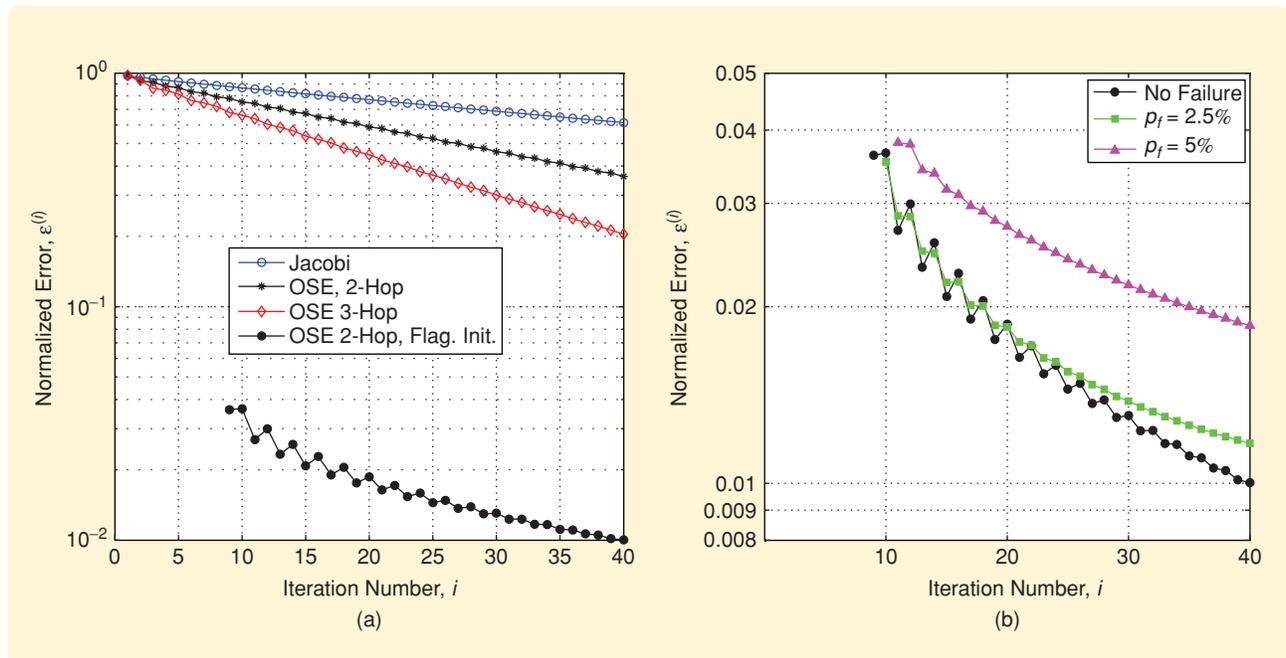


FIGURE 13 (a) Performance comparison between the Jacobi algorithm and the overlapping subgraph estimator (OSE) algorithm without link failures. The normalized error is defined as $\epsilon^{(i)} = \|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^*\| / \|\hat{\mathbf{x}}^*\|$, where $\hat{\mathbf{x}}^{(i)}$ is the vector of estimates at the i th iteration and $\hat{\mathbf{x}}^*$ is the optimal estimate. Except for the case with flagged initialization, all of the simulations are run with all initial estimates of node variables set to zero. For the flagged OSE, the normalized error can be defined only after iteration number 8 because until then not all nodes have valid (finite) estimates. (b) Performance of two-hop OSE with link failures. All simulations are run with flagged initialization. Two different failure probabilities are compared with the case of no failure. With higher probability of failure, performance degrades but the error is seen to decrease with iteration count even with large failure probabilities.

The study of generalized electrical networks with matrix-valued resistances appears to be useful for a wide variety of other problems, such as distributed control, defined on large graphs [22].

ACKNOWLEDGMENT

The authors would like to thank Prof. Edmond Jonckheere for helpful and stimulating discussions on dense and sparse graphs.

REFERENCES

- [1] A. Arora, R. Rammath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, C. Zhang, M. Nesterenko, R. Shah, S. Kulkarni, M. Aramugam, L. Wang, M. Gouda, Y. Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker, "Exscal: Elements of an extreme scale wireless sensor network," in *Proc. 11th IEEE Int. Conf. Embedded Real-Time Computing Systems Applications (RTCISA)*, 2005, pp. 102–108.
- [2] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.
- [3] R. Karp, J. Elson, D. Estrin, and S. Shenker, "Optimal and global time synchronization in sensor networks," Center for Embedded Networked Sensing, Los Angeles, CA, CENS Tech. Rep. #12, 2003 [Online]. Available: <http://citeseer.ist.psu.edu/article/karp03optimal.html>
- [4] G. Venezian, "On the resistance between two points on a grid," *Amer. J. Phys.*, vol. 62, no. 11, pp. 1000–1004, 1994.
- [5] J. Cserti, "Application of the lattice green's function for calculating the resistance of and infinite network of resistors," *Amer. J. Phys.*, vol. 68, no. 10, pp. 896–906, 2000.
- [6] J.M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [7] P. Barooah and J.P. Hespanha, "Estimation from relative measurements: Error bounds from electrical analogy," in *Proc. 2nd Int. Conf. Intelligent Sensing Information Processing (ICISIP)*, 2005, pp. 88–93.
- [8] P. Barooah and J.P. Hespanha, "Optimal estimation from relative measurements: Electrical analogy and error bounds," Univ. California, Santa Barbara, Tech. Rep., 2005 [Online]. Available: <http://www.ece.ucsb.edu/~hespanha/techrep.html>
- [9] J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Emerging challenges: Mobile networking for "smart dust"," *J. Commun. Networks*, vol. 2, pp. 188–196, Sept. 2000.
- [10] S. Mueller, R.P. Tsang, and D. Ghosal, "Multipath routing in mobile ad hoc networks: Issues and challenges," in *Performance Tools and Applications to Networked Systems*, M. Calzarossa and E. Gelenbe, Eds (LNCS, vol. 2965). Berlin: Springer-Verlag, 2004, pp. 209–234.
- [11] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, "The effects of ranging noise on multihop localization: An empirical study," in *Proc. 4th Int. Symp. Information Processing Sensor Networks (IPSN'05)*, 2005, pp. 73–80.
- [12] P.G. Doyle and J.L. Snell, *Random Walks and Electric Networks*. Math. Assoc. America, 1984.
- [13] P.G. Doyle. (1998). Application of Rayleigh's short-cut method to Polya's recurrence problem [Online]. Available: <http://math.dartmouth.edu/~doyle/docs/thesis/thesis.pdf>
- [14] D. Niculescu and B. Nath, "Error characteristics of ad hoc positioning systems (APS)," in *Proc. 5th ACM Int. Symp. Mobile Ad Hoc Networking Computing (MobiHoc)*, 2004, pp. 20–30.
- [15] A. Savvides, W. Garber, R. Moses, and M. b. Srivastava, "An analysis of error inducing parameters in multihop sensor node localization," *IEEE Trans. Mobile Computing*, vol. 4, no. 6, pp. 567–577, 2005.
- [16] P. Barooah and J.P. Hespanha, "Distributed optimal estimation from relative measurements," in *Proc. 3rd Int. Conf. Intelligent Sensing Information Processing (ICISIP)*, 2005, pp. 226–231.
- [17] P. Barooah, N.M. da Silva, and J.P. Hespanha, "Distributed optimal estimation from relative measurements for localization and time synchronization," P.B. Gibbons, T. Abdelzaher, J. Aspnes and R. Rao, Eds., in *Distributed Computing in Sensor Systems (DCOSS)*, (LNCS, vol. 4026), New York: Springer-Verlag, 2006, pp. 266–281.
- [18] A. Chandrakasan, R. Min, M. Bhardwaj, S. Cho, and A. Wang, "Power aware wireless microsensor systems," in *28th European Solid-State Circuits Conf. (ESSCIRC)*, Florence, Italy, 2002, pp. 47–54.
- [19] B. Bougard, F. Catthoor, D.C. Daly, A. Chandrakasan, and W. Dehaene, "Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: Modeling and improvement perspectives," in *Proc. Design, Automation Test Europe (DATE)*, vol. 1, 2005, pp. 196–201.
- [20] M.M. Carvalho, C.B. Margi, K. Obraczka, and J. Garcia-Luna-Aceves, "Modeling energy consumption in single-hop IEEE 802.11 ad hoc networks," in *Proc. 13th Int. Conf. Computer Communications Networks (ICCCN)*, 2004, pp. 367–372.
- [21] A.K. Chandra, P. Raghavan, W.L. Ruzzo, R. Smolensky, and P. Tiwari, "The electrical resistance of a graph captures its commute and cover times," in *Proc. 21st Annu. ACM Symp. Theory Computing*, 1989, pp. 574–586.
- [22] P. Barooah and J.P. Hespanha, "Graph effective resistances and distributed control: Spectral properties and applications," in *Proc. 45th IEEE Conf. Decision Control*, 2006, pp. 3479–3485.
- [23] C. Godsil and G. Royle, *Algebraic Graph Theory* (Graduate Texts in Mathematics). New York: Springer-Verlag, 2001.

AUTHOR INFORMATION

Prabir Barooah received his B.Tech. and M.S. degrees in mechanical engineering from the Indian Institute of Technology, Kanpur, in 1996 and the University of Delaware, Newark, in 1999, respectively. From 1999 to 2002 he worked at the United Technologies Research Center. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. He is a recipient of a NASA group achievement award (2003) and the best paper award at the Second International Conference on Intelligent Sensing and Information Processing (2005). His research interests include active combustion control, system identification, and distributed control and estimation in large-scale networks.

João P. Hespanha (hespanha@ece.ucsb.edu) received the Licenciatura in electrical and computer engineering from the Instituto Superior Técnico, Lisbon, Portugal in 1991 and the M.S. and Ph.D. degrees in electrical engineering and applied science from Yale University, New Haven, Connecticut, in 1994 and 1998, respectively. From 1999 to 2001, he was an assistant professor at the University of Southern California, Los Angeles. He is currently a professor with the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. His research interests include hybrid and switched systems, modeling and control of communication networks, distributed control over communication networks, vision in feedback control, stochastic modeling in biology, control of haptic devices, and game theory. He is the author of over 100 technical papers. He is the recipient of the Yale University's Henry Prentiss Becton Graduate Prize for exceptional achievement in research in Engineering and Applied Science, a National Science Foundation CAREER Award, the 2005 best paper award at the Second International Conference on Intelligent Sensing and Information Processing, the 2005 *Automatica* Theory/Methodology best paper prize, and the 2006 George S. Axelby Outstanding Paper Award. Since 2003, he has been an associate editor of *IEEE Transactions on Automatic Control*. He can be contacted at Room 5157, Engineering I, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560 USA.

