Estimation on Graphs from Relative Measurements

Distributed algorithms and fundamental limits

Prabir Barooah João P. Hespanha

INTRODUCTION

Sensor networks are collections of interconnected nodes equipped with sensing and computing capability that are deployed in a geographic area to perform some form of monitoring tasks. Networks consisting of a large collection of such nodes are currently under development or envisioned for the near future [1, 2]. Usually a node can communicate with only a small subset of other nodes. These constraints on the communications define a graph whose vertices are the nodes and the edges are the communication links. In many such situations, nodes lack knowledge of certain global attributes, for example, their own positions in a global reference frame. However, nodes might be capable of measuring the relative values of such attributes with respect to nearby nodes. In such a scenario, it would be useful if the nodes could estimate their global attributes from these relative measurements. The scenarios described below provide motivation for these problems.

a) Localization: A large number of sensors is deployed in a region of space. Sensors do not know their positions in a global coordinate system, but every sensor can measure its relative position with respect to a set of nearby sensors. These measurements could be obtained, for example, from range and bearing (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

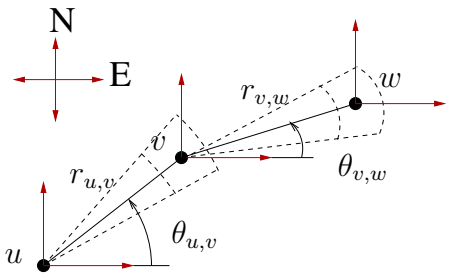


Fig. 1. Relative position measurement in a Cartesian reference frame from range and bearing measurement. A local compass at each sensor is needed to measure bearing with respect to a common "North." Noisy measurements of range and bearing, $r_{u,v}$ and $\theta_{u,v}$, between a pair of sensors u and 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 is done 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.

measurement

$$\zeta_{u,v} = p_u - p_v + \epsilon_{u,v} \in \mathbb{R}^2,$$

where $\epsilon_{u,v}$ denotes some inevitable measurement error. The problem of interest is to estimate positions of all sensors in a common coordinate system based on this type of noisy relative position measurements. One of the sensor is assumed to be at the origin to remove the ambiguity that arises out of all sensors having only measurements of relative positions.

b) Time-Synchronization: A group of sensing nodes are part of a multi-hop communication network. Each node has a local clock but the times measured by the clocks differ by constant values, called clock offsets. However, nodes that communicate directly can estimate the difference between their local clocks, typically by exchanging "hello" messages that are time-stamped with local clock times. Consider a pair of nodes u and v that can communicate directly with each other and that 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 some error

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

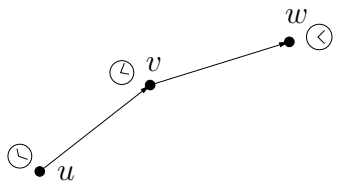


Fig. 2. Measurement of differences in local times by exchanging time-stamped messages. Node u transmits a message, say, at global time t. The transmitter u's local time is $\tau_{tu} = t + t_u$. The receiver v receives this message at some later time, when its local clock reads $\tau_{rv} = t + t_v + \delta_{u,v}$, 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 $\tau'_{tv} = t' + t_v$. It includes the values τ_{rv} and τ'_{tv} in the message body. The receiver u receives this message at local time $\tau'_{ru} = t' + t_u + \delta_{uv}$, where the delay δ_{vu} has the same mean as the delay δ_{uv}. Node u can now estimate the clock offsets as $\zeta_{u,v} = \frac{1}{2}\left((\tau'_{ru} - \tau'_{tv}) - (\tau_{rv} - \tau_{tu})\right) = t_u - t_v + (\delta_{vu} - \delta_{uv})/2$. The error $\epsilon_{u,v} := (\delta_{vu} - \delta_{uv})/2$ has zero mean as long as the (unidirectional) delays have the same expected value. The measured clock offset between u and v is now $\zeta_{u,v} = t_u - t_v + \epsilon_{u,v}$, of the form (1). Similarly, the measurement of clock offsets between nodes v and w is $\zeta_{v,w} = t_v - t_w + \epsilon_{v,w}$.

where $\epsilon_{u,v}$ denotes the measurement error (see Figure 2). The task is now to estimate the clock offsets with respect to the global time as accurately as possible. The global time could simply be the local time at some reference node. This problem is considered in [3], though the measurements are modeled in a slightly different manner.

c) Motion Consensus: Several mobile agents move in space. Every agent would like to determine its velocity with respect to the velocity of a "leader," but the agents can only measure their relative velocities with respect to nearby agents. These measurements could be obtained, for example, 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

$$\zeta_{u,v} = \dot{p}_u - \dot{p}_v + \epsilon_{u,v} \in \mathbb{R}^3,$$

where $\epsilon_{u,v}$ 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.

The problems described above share the common objective of estimating the values of a number of *node variables*

$$x_1,\ x_2,\ \ldots\ x_n \in \mathbb{R}^k, \quad k \geq 1$$

from noisy "relative" measurements of the form

$$\zeta_{u,v} = x_u - x_v + \epsilon_{u,v}, \quad u, v \in \{1, 2, \ldots, n\}, \tag{1}$$

where the $\epsilon_{u,v}$'s are zero-mean noise vectors with associated covariance matrices $P_{u,v} = \mathrm{E}[\epsilon_{u,v}\epsilon_{u,v}^T]$. The measurements are assumed to be uncorrelated with one another, that is, $\mathrm{E}[\epsilon_{u,v}\epsilon_{p,q}^T] = 0$ unless $u = p$ and $v = q$. This estimation problem can be naturally associated with the directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, which we call the *measurement graph*. Its *vertex set* is the node set $\mathbf{V} := \{1, 2, \ldots, n\}$ and its *edge set* $\mathbf{E}$ consists of all 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 $\boldsymbol{\epsilon}$, we can express all of the measurement equations (1) in the compact form

$$\mathbf{z} = \mathcal{A}^T\mathbf{X} + \boldsymbol{\epsilon}, \tag{2}$$

where $\mathcal{A}$ is uniquely determined by the graph $\mathbf{G}$. To construct $\mathcal{A}$, we start by defining the *incidence matrix of* $\mathbf{G}$, which is an $n \times m$ 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 nonzero $a_{ue} = -1$ if the edge e is directed toward u and $a_{ue} = 1$ otherwise. An edge $e = (u, v)$ is *incident* on the nodes u and v. Figure 3 shows an example of a measurement graph and its incidence matrix. 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 $\otimes$ denote the Kronecker product. Essentially, every entry of A is replaced by a matrix $a_{ue}I_k$ to form the matrix $\mathcal{A}$ (see Figure 3).

Just with relative measurements, determining the x_u's is only possible up to an additive constant. To avoid this ambiguity, we assume that at least one of the nodes is used as a *reference* and therefore its node variable can be assumed known. In general, several node variables may be known and therefore we may have several references. The task is to estimate all of the unknown node variables from the measurements.

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

$$\mathbf{z} - \mathcal{A}_r^T\mathbf{x}_r = \mathcal{A}_b^T\mathbf{x} + \boldsymbol{\epsilon}, \tag{3}$$

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.

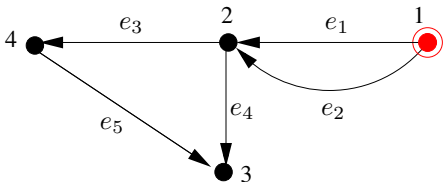
The estimation of the unknown node-variables in the vector $\mathbf{x}$ based on the linear measurement model (3) is a classical estimation problem. The least-squares solution to this problem is generally attributed to Gauss [4] but the American mathematician Adrain [5] seems to have arrived at the same results independently [6]. When $\boldsymbol{\epsilon}$ is a random vector with zero mean and covariance matrix $\mathscr{P} := \mathrm{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T]$, the least squares solution leads to the classical Best Linear Unbiased Estimator (BLUE), given by

$$\hat{\mathbf{x}}^* := \mathcal{L}^{-1}\mathbf{b}, \qquad \mathcal{L} := \mathcal{A}_b\mathscr{P}^{-1}\mathcal{A}_b^T, \qquad \mathbf{b} := \mathcal{A}_b\mathscr{P}^{-1}(\mathbf{z} - \mathcal{A}_r^T\mathbf{x}_r). \tag{4}$$

Among all linear estimators of $\mathbf{x}$, the BLU estimator has the smallest variance for the estimation error $\mathbf{x} - \hat{\mathbf{x}}^*$ [7]. The inverse of the matrix $\mathcal{L}$ exists as long as the measurement graph is weakly connected [8] and provides the covariance matrix of the estimation error:

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

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. 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 $\mathbf{\Sigma}$. Figure 3 shows a simple measurement graph along with the corresponding measurement equations (2)–(3) and estimate (4) defined above. Since the measurement errors are assumed uncorrelated with one-another, the error covariance matrix $\mathscr{P}$ happens to be a block diagonal matrix. For this reason, the matrix $\mathcal{L}$ has a structure that is closely related to the graph Laplacian of $\mathbf{G}$. However, this connection is not explored here; the interested reader is referred to [8] for more details.



$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

A measurement Graph $\mathbf{G}$ and its incidence matrix A (row and column indices of A correspond to node and edge indices, respectively). Matrix form (2) of the measurement equations (1) for this graph:

$$\underbrace{\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} I & -I & 0 & 0 \\ I & -I & 0 & 0 \\ 0 & I & 0 & -I \\ 0 & I & -I & 0 \\ 0 & 0 & -I & I \end{bmatrix}}_{\mathcal{A}^T} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{X}} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

Measurement model (3) when node 1 is the reference with $x_1 = 0$:

$$\underbrace{\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} I \\ I \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\mathcal{A}_r^T} \underbrace{0}_{\mathbf{x}_r} + \underbrace{\begin{bmatrix} -I & 0 & 0 \\ -I & 0 & 0 \\ I & 0 & -I \\ I & -I & 0 \\ 0 & -I & I \end{bmatrix}}_{\mathcal{A}_b^T} \underbrace{\begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\mathbf{x}} + \boldsymbol{\epsilon}$$

Optimal estimates (4) when all measurement covariance matrices are equal to the identity matrix:

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

Covariance matrices of the overall estimation error and of the individual node-variable errors:

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

Fig. 3. A measurement graph $\mathbf{G}$ with 4 nodes and 5 edges. Node 1 is the reference. The incidence matrix A is therefore a 4×5 matrix consisting of 0s, 1s and -1s. The 4 node variables (in the vector $\mathbf{X}$) are related to the 5 measurements (in the vector $\mathbf{z}$) by the $4k \times 5k$ matrix $\mathcal{A}$, the expanded version of the incidence matrix. The relationship between the 3 *unknown* node variables in the vector $\mathbf{x}$ are related to the known quantities (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 the equation (4), is given by $\hat{\mathbf{x}}^* = \mathcal{L}^{-1}\mathcal{A}_b\mathscr{P}^{-1}\bar{\mathbf{z}}$. Note the Laplacian-like structure of the matrix $\mathcal{L}$. The covariance of the estimation error of a node u is simply the $u-1$th diagonal block of the covariance matrix $\boldsymbol{\Sigma}$.

The estimation problem defined above could 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 because it introduces a single point of failure and does not balance the communication/computational load among nodes. For a large ad-hoc network of wireless sensor nodes, sending all of the measurements requires multi-hop communication and unduly burdens the nodes close to the central processor. A high communication load reduces the life of a sensor node that operates on batteries and typically has a small energy budget. Reducing the life of individual nodes in turn reduces the operational life of the network [2]. In addition, a centralized computation is less robust to dynamic changes in the topology resulting from node and link failures over time. Therefore a distributed algorithm that can compute the optimal estimates while using only local communication is advantageous in terms of scalability, robustness and network life.

The preceeding 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 computation and communication burden is equally shared by all of the nodes? And 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, but is allowed to periodically exchange messages with its neighbors. In this article, the concept of "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. In other words, we are implicitly assuming 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 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 simple graph shown in Figure 4. It is a chain of nodes with node 1 as the reference

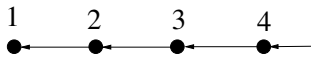


Fig. 4. A graph where a node variable's optimal estimate has an error variance that grows linearly with the node's distance from the reference. In this case, since every edge represents a measurement with independent error, the optimal estimate of a node variable x_u is simply the sum of the measurements along the edges from the reference (node 1) to node u, and its variance is the sum of the error variances of those measurements. Assuming for simplicity that all measurement error variances are equal, the variance of the estimation error $x_u - \hat{x}_u$ is proportional to the minimum number of edges one has to traverse in going from 1 to u – the "graphical distance" between 1 and u.

and with a single edge $(u+1, u)$, $u \geq 1$ between consecutive nodes u and $u+1$. Without much difficulty, one can show that for such a graph the optimal estimate of x_u is given by

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

and since each measurement introduces an additive error, the variance of the optimal estimation error $\hat{x}_u - x_u$ increases linearly with u. Therefore, if u is "far" from the reference node 1 then its estimate is necessarily be quite poor. Although the precise estimation error depends on the exact values of the variances of the measurements, for this graph the variance of the optimal estimate of x_u grows essentially 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 grows linearly with the "distance" to the reference nodes? If not, what types of scaling laws*

for error are possible, and 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 far better than the linear scaling that we encountered in the graph shown in Figure 4. For a given maximum acceptable error, the number of nodes with acceptable estimation errors is going to be large if the graph exhibits a slow increase of variance with distance, but small otherwise. These scaling laws therefore determine how large a graph can be in practice. The answer to question 2 also helps us in designing and deploying large networks for which accurate estimates are possible.

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/edge density that are commonly used in the sensor networks literature. We describe a classification of graphs that determines how the variance grows with distance from the reference node. There are graphs where variance grows linearly with distance, as in the example of Figure 4. But there are also a large class of graphs where it grows only logarithmically with distance. Most surprisingly, in certain graphs it can even stay below a constant value, no matter the distance.

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 error scaling issue we show that the variance of a node's optimal estimate is numerically equal to an abstract *matrix-valued effective resistance* in

an appropriately defined 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 also builds an invaluable intuition into this problem. We show that the matrix-valued effective resistance in a complicated graph can sometimes be bounded by the effective resistance in a "nicer" graph, in which we know how the resistance grows with distance. These nice graphs are the well known lattice graphs. An answer to the question of variance scaling is thus obtained by exploiting the electrical analogy.

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 themselves. The *effective resistance* between two nodes in an electrical network is then defined as the potential drop between the two nodes when a current source with intensity equal to 1 Ampere is connected across the two nodes (see Figure 5). For a general network, the computation of effective resistances relies on the usual Kirchoff's and Ohm's laws.

To see the connection between electrical networks and our estimation problem, consider the simple measurement graph shown in Figure 6, where a single *scalar unknown variable* x_2 needs to be estimates 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. A simple calculation using the BLU covariance formula (5) shows that the variance σ_2^2 of the optimal estimate of x_2 is given by the formula

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

Suppose now that we construct a resistive network based on the measurement graph, by assigning the edge-resistances R_a, R_b numerically equal to the variances σ_a^2, σ_b^2 of the corresponding

measurement errors (see Figure 6). It is an elementary result in electrical circuits that that the effective resistance between nodes 1 and 2 of the electrical network is given by

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

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 extends to *arbitrary measurement graphs* (not just the simple one shown in Figure 6) and is made in [3] in the context of estimating *scalar node variables.*

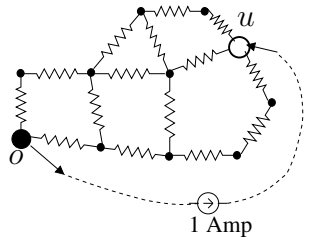


Fig. 5. A resistive electrical network. 1 Ampere current is injected at node u and extracted at the reference node o. The resulting potential difference $V_u - V_o$ is the effective resistance between u and o. When one deals with scalar valued measurements, if every resistance value is set equal to the 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 $x_u - \hat{x}_u$ (with o as the reference). In this article we demonstrate how this analogy can be extended – and suitably exploited – when variables and measurements are vector-valued quantities.

Going back to the estimation problem in Figure 6, but with node variables that are k-vectors, still 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

$$\Sigma_2^{-1} = P_a^{-1} + P_b^{-1}, \tag{6}$$

where now P_a and P_b are the $k \times k$ covariance matrices of the two measurements. This formula looks tantalizingly similar to the effective resistance formula shown before and prompted us to search for a more general electrical analogy.

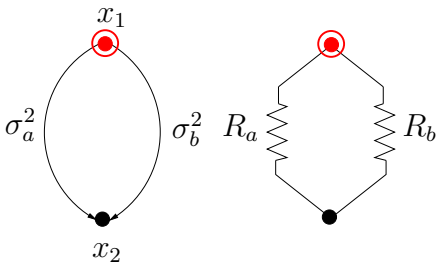


Fig. 6. A measurement graph (left) and the corresponding resistive electrical network (right). Node 1 is the reference. The variance of the optimal estimate of x_2 has the same numerical values as the effective resistance between nodes 1 and 2 in the electrical network on the right when resistances are assigned equal to the measurement error variances: $R_a = \sigma_a^2$, $R_b = \sigma_b^2$.

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

$$V_e = R_e i_e,$$

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

The generalized electrical networks so defined share many of the properties of "regular" electrical networks. In particular, given the current injected into a node and extracted at another, the generalized Kirchoff's and Ohm's laws uniquely define all currents and voltage drops on the edges of a generalize electrical network. Solving for the currents and voltages allows us to define the *generalized effective resistance* between two nodes as the 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 (6) is precisely the formula to compute the generalize effective resistance for the parallel network of two generalized resistors. Moreover, the electrical analogy, discovered in [3]

for scalar measurements still holds for vector measurements when one considers generalized electrical networks. This result is proved in [8], and is stated below.

Theorem 1. *Consider a measurement graph* $\mathbf{G}$ *with a single reference node* o *and construct a generalized electric network with* $k \times k$ *edge-resistors that are numerically equal to the covariance matrices of the edge-measurement errors. For every node* u*, the* $k \times k$ *covariance matrix* Σ_u *of the 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 the coefficient matrices that multiply the measurements to construct these estimates also have interesting interpretations in terms of electrical quantities [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 error covariance. In the context of electrical networks, the directions of edges are important to determine the "signs" of the currents, but are irrelevant in determining effective resistances.

In the remaining part of this section, we discuss several results that were previously known for "regular" electrical networks and that can be adapted to generalized electrical networks. In view of Theorem 1, these results then carry over to our graph estimation problem.

Rayleigh's Monotonicity Law

Rayleigh's Monotonicity Law states that if the edge-resistances in a (regular) electrical network are increased, then the effective resistance between every pair of nodes in the network can only increase. Conversely, a decrease in edge-resistances can only lead to a decrease in effective resistance. In this article, we show that Theorem 1 allows us to use a generalized version of Rayleigh's Monotonicity Law to the problem of variance scaling in measurement graphs.

Rayleigh's Monotonicity Law was considered so evidently true that no proof was deemed necessary. Indeed, James Clark Maxwell wrote in his *Treatise on Electricity and Magnetism* (p. 427, [9]):

> *If the specific resistance of any portion of the conductor be changed, that of the remainder being unchanged, the resistance of the whole conductor will be increased if that of the portion is increased, and diminished if that of the portion is diminished. This principle may be regarded as self-evident ...*

Fortunately, at least Doyle and Snell did not regarded Rayleigh's Monotonicity Law as self evident and published its proof in [10]. It turns out that their proof can be extended to generalized electrical networks.

For the problems considered here, it is convenient to consider not only increases or decreases in edge-resistances but also removing an edge altogether and adding new edges. For scalar resistances removing an edge would correspond to making a resistance equal to infinity, but for matrix resistances one needs to be more careful because there are multiple ways in which a matrix can be taken to infinity. The statement of a version of Rayleigh's Monotonicity Law that allow us to consider edge removal requires the concept of "graph embedding." Given two graphs

$$\mathbf{G} = (\mathbf{V}, \mathbf{E}), \qquad\qquad \bar{\mathbf{G}} = (\bar{\mathbf{V}}, \bar{\mathbf{E}})$$

we say that $\mathbf{G}$ *can be embedded in* $\bar{\mathbf{G}}$, or alternatively that $\bar{\mathbf{G}}$ *embeds* $\mathbf{G}$ if the two 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 $\bar{\mathbf{G}}$, but 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 7 shows two graphs to illustrate the concept of graph embedding.

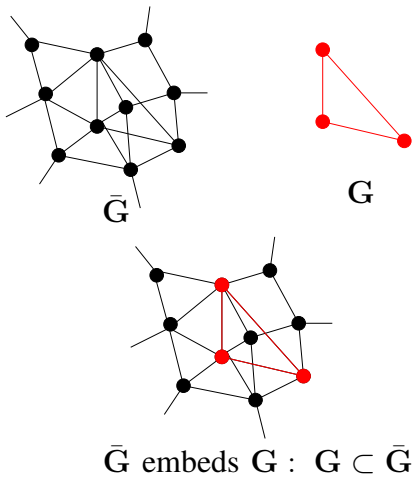


Fig. 7. A graph embedding example. Every node of the graph $\mathbf{G}$ can be made to correspond in a one-to-one fashion with a subset of the nodes of $\mathbf{G}$, and if there is an edge between two nodes in $\mathbf{G}$, there is an edge between their corresponding nodes in $\bar{\mathbf{G}}$. So we say $\bar{\mathbf{G}}$ embeds $\mathbf{G}$, or equivalently, $\mathbf{G}$ can be embedded in $\bar{\mathbf{G}}$. In mathematical notation, $\mathbf{G} \subset \bar{\mathbf{G}}$. One can also think of $\mathbf{G}$ being a "subgraph" of $\bar{\mathbf{G}}$.

It is proved in [8] that Rayleigh's Monotonicity Law holds even in the case of 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*

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* $\mathbf{G}$ *we have that* $R_e \geq \bar{R}_{\bar{e}}$, *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* $\mathbf{G}$, *we have that*

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

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

In the statement of Theorem 2 and below, given two symmetric matrices A and B, we write $A \geq B$ and say "B is smaller than A" to mean that $A - B$ is a positive semi-definite matrix.

In terms of the original estimation problem, Rayleigh's Generalized 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 can only decrease (that is, the estimate becomes more accurate). In addition, if new measurements are introduced, the new optimal estimate necessarily becomes more accurate (even if the new measurements are very noisy). Maxwell would probably say that these statements "may be regarded as self-evident." Perhaps only control theorists bother to prove them.

Lattices, Fuzzes and their Effective Resistances

Lattice graphs (see Figure 9) are familiar examples of regular degree graphs, that is, graphs in which every node has the same degree, and perhaps require no introduction. The effective resistance in lattices (with scalar resistors on every edge) have been studied in the literature, and we show that similar results can be obtained for *generalized* lattice networks. Lattices and a class of graphs derived from them called lattice fuzzes are especially useful in studying the scaling laws of effective resistance in large graphs.

To define the concept of a fuzz of a graph [10], we introduce the notion of graphical distance. Given two nodes u and v of a graph $\mathbf{G}$, their *graphical distance*, denoted by $d_{\mathbf{G}}(u, v)$ is the minimum number of edges one has to traverse in going from one node to the other. In this definition, we allow edges to be traversed in any direction and therefore $d_{\mathbf{G}}(u, v) = d_{\mathbf{G}}(v, u)$.

Given a graph $\mathbf{G}$ and an integer $h \geq 1$, *the h-fuzz of* $\mathbf{G}$, denoted by $\mathbf{G}^{(h)}$, is a graph with the same set of nodes as $\mathbf{G}$ but with a larger set of edges. In particular, $\mathbf{G}^{(h)}$ has an edge between u and v whenever the graphical distance between u and v is less than or equal to h [10]. The directions of the "new" edges are arbitrary (see the comment following Theorem 1). Figure 8 shows a graph and its 2-fuzz.

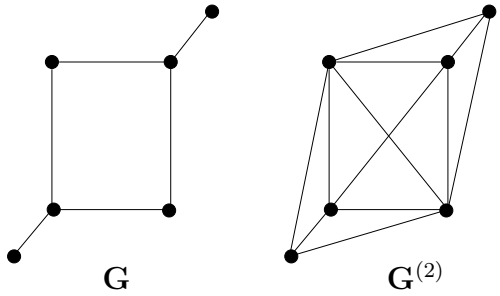


Fig. 8. A graph $\mathbf{G}$ and its 2-fuzz $\mathbf{G}^{(2)}$. Every pair of nodes in $\mathbf{G}$ that are at a graphical distance of 2 have an edge between them in $\mathbf{G}^{(2)}$. The graphical distance is cut by a factor of h in going from a graph to its h-fuzz. Note that the edge directions are not shown as they are not important as long as we are interested only in the effective resistance.

An h-fuzz clearly has a lower effective resistance than the original graph because of Rayleigh's Generalized Monotonicity Law, but it is lower only by a constant factor. In particular, we have the following result that was first derived for "regular" electrical networks in [11], and then shown to hold for generalized networks in [8].

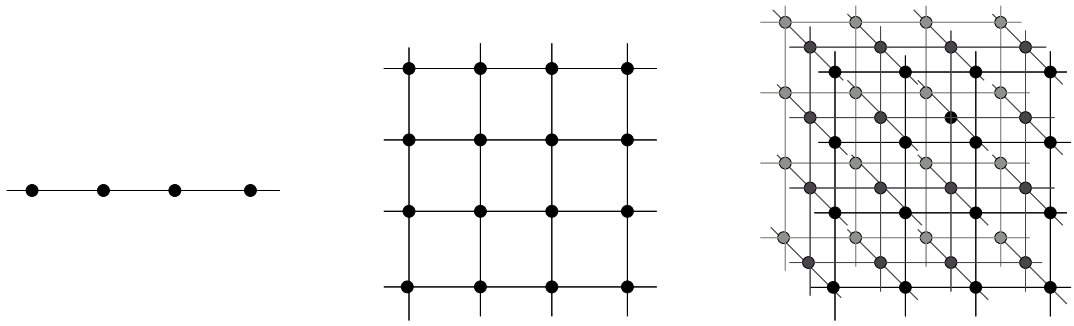
Lemma 1.

Consider a weakly connected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ 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 $\mathbf{G}$ and the other by placing the same matrix resistance R at every edge of its h-fuzz $\mathbf{G}^{(h)}$. For every pair of nodes u and v in $\mathbf{V}$,

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

where $R^{\text{eff}}_{u,v}(\mathbf{G})$ *is the effective resistance between* u *and* v *in* $\mathbf{G}$ *and* $R^{\text{eff}}_{u,v}(\mathbf{G}^{(h)})$ *is the effective resistance in* $\mathbf{G}^{(h)}$ *and* $\alpha \in (0, 1]$ *is a positive constant that does not depend on* u *and* v.

A *d-dimensional lattice*, denoted by $\mathbf{Z}_d$ is a graph that has a vertex at every point in $\mathbb{R}^d$ with integer coordinates and an edge between every two vertices with an Euclidean distance between them equal to one. Edge directions are arbitrary. Figure 9 shows 1-, 2-, and 3-dimensional 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.



(a) 1-dimensional lattice $\mathbf{Z}_1$

(b) 2-dimensional lattice $\mathbf{Z}_2$

(c) 3-dimensional lattice $\mathbf{Z}_3$

Fig. 9. Lattices

The next lemma from [8] establishes the generalized effective resistance of d-dimensional lattices and their fuzzes.

Lemma 2 (Lattice Effective Resistance).

Consider a generalized electrical network obtained by placing generalized matrix resistances equal to R *at the edges 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. There exist constants*

TABLE I

EFFECTIVE RESISTANCE FOR LATTICES AND THEIR FUZZES. $d_{\mathbf{Z}_d}(\cdot)$ REPRESENTS THE GRAPHICAL DISTANCE IN THE LATTICE $\mathbf{Z}_d$

Graph	Effective resistance $R^{\text{eff}}_{u,v}$ between u and v
 $\mathbf{Z}_1^{(h)}$	$\alpha_1 d_{\mathbf{Z}_1}(u,v)R \leq R^{\text{eff}}_{u,v} \leq \beta_1 d_{\mathbf{Z}_1}(u,v)R$
 $\mathbf{Z}_2^{(h)}$	$\alpha_2 \log\big(d_{\mathbf{Z}_2}(u,v)\big)R \leq R^{\text{eff}}_{u,v} \leq \beta_2 \log\big(d_{\mathbf{Z}_2}(u,v)\big)R$
 $\mathbf{Z}_3^{(h)}$	$\alpha_3 R \leq R^{\text{eff}}_{u,v} \leq \beta_3 R$

$\ell, \alpha_i, \beta_i > 0$ *such that the formulas in Table I hold for every pair of nodes* u, v *at a graphical distance larger than* ℓ*.*

The fact that in a 1-dimensional lattice the effective resistance grows linearly with the distance between nodes can be trivially deduced from the well known formula for the effective resistance of a series of resistors (which generalizes to generalized electrical networks). In two-dimensional lattices the effective resistance only grows with the logarithm of the graphical distance and therefore the effective resistance grows slowly with the distance between nodes. Far more surprising is the fact that in three-dimensional latices the effective resistance is actually bounded by a constant even when the distance is arbitrarily large.

Error scaling with Distance: Dense and Sparse Graphs

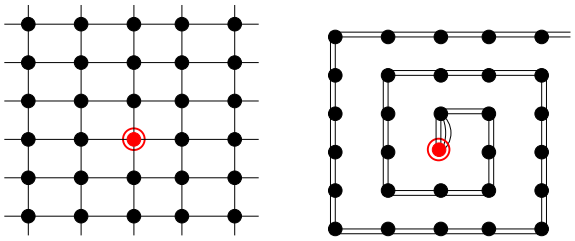
We now show how to combine the tools developed so far to determine the scaling laws of the estimation error variance for general classes of measurement graphs. Roughly speaking,

our approach is the following: we determine what structural properties must a graph satisfy so that it can either embed, or be embedded in a lattice or the h-fuzz of a lattice. These conditions come from looking at different "drawings" of the graph. When a graph can be embedded in a lattice, Rayleigh's Generalized Monotonicity Law gives us a lower bound on the generalized effective resistance of the graph in terms of the effective resistance in the lattice. We already know the effective resistance in the lattice as a function of distance, from the lattice effective resistance Lemma 2. When a graph embeds a lattice, we get an upper bound.

Before we go into describing these concepts precisely, we might ask ourselves if there are no simple indicators of the relationship between graph structure and estimator accuracy that might be used to answer the question of variance scaling without going into the somewhat complex route we have outlined above. In fact, in the sensor networks literature, it is recognized that higher "density" of nodes/edges usually leads to better estimation accuracy. Usually, the average number of nodes in an unit of area or the average degree of a node (that is, its number of neighbors) are used to quantify the notion of denseness [12, 13]. We have already seen that when one graph can be embedded in another, the one with the higher number of edges have a lower effective resistance, and consequently, lower estimator variance. One could therefore expect that a higher density of edges and nodes in the measurement graph should lead to better estimates. However, "naive" measures of node and edge density turn out to be misleading predictors for how the *estimation error variance scales with distance*. We now present a few examples to motivate the search for deeper graph-structural properties that determine how variance scales with distance.

Counterexamples to Conventional Wisdom

Consider the two graphs $\mathbf{G}_1$ and $\mathbf{G}_2$ in Figure 10 and imagine that these two graphs represent the measurement graphs of two sensor networks deployed in 2-d Euclidean space, and that we draw the graphs so that the positions of the vertices in the paper matches precisely the physical locations of the sensors in the plane where they are deployed (perhaps up to some



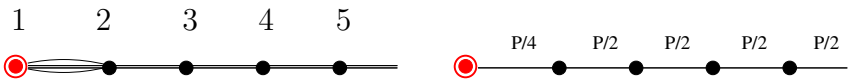
(a) $\mathbf{G}_1$ (b) $\mathbf{G}_2$

Fig. 10. Two measurement graphs drawn in 2-dimension with the same number of nodes and edges per unit area and with the same node degree of 4 (except for a single node of $\mathbf{G}_2$ that has a higher degree of 6). The nodes in the center shown in red are the reference nodes. By all conventional measures of "dense-ness" such as node degree, number of nodes/edges per unit area, these two graphs are quite similar. One would expect the estimation error to scale with distance in the same manner in both the graphs. However, as the discussion in this article shows, it is not the case.

scaling). For simplicity we assume that all measurements are affected by noise with the same covariance matrix P. For both graphs the number of nodes per unit area is exactly the same and the degree of every node in both the graphs is four (except for a single node of $\mathbf{G}_2$ that has a higher degree of 6). So "naive" notions of denseness would classify the graphs as similar.

The graph $\mathbf{G}_1$ is a 2-dimensional lattice so from the Electrical Analogy Theorem 1 and the lattice effective resistance Lemma 2, we conclude that the estimation error variance grows asymptotically with the logarithm of the graphical distance from the node to the reference. The exact same statement is true if we replace "graphical distance" by "Euclidean distance" because this drawing of the graph has the property that the graphical distance is always between the Euclidean distance and $\sqrt{2}$ times the Euclidean distance.

To study the estimation error variance of $\mathbf{G}_2$, we redraw this graph as in Figure 11(a). It is now easy to recognize that $\mathbf{G}_2$ is also equivalent to an 1-dimensional lattice shown in Figure 11(b), with the noise covariance matrix equal to $P/2$ at every edge, except the first one that for which the covariance matrix is $P/4$. This equivalence follows from the fact that the



(a) A 1-dimensional drawing of $\mathbf{G}_2$ shown in Figure 10

(b) A graph that is equivalent to the one in (a) in terms of effective resistances.

Fig. 11. (a) A 1-dimensional drawing of the graph $\mathbf{G}_2$ in Figure 10. This graph drawing is simply obtained by "unspiralling" the graph $\mathbf{G}_2$. (b) A 1-dimensional drawing of a graph that has the same effective resistance (between every pair of nodes) as the graph in (a). The graph in (b) is obtained by replacing parallel matrix-resistances between a pair of nodes by a single equivalent matrix-resistance. For example, the four edges between node 1 and 2, each of resistance P, is replaced by a single edge of resistance $P/4$. The resulting graph, when drawn in 1-dimension, is recognized to be a 1-dimensional lattice.

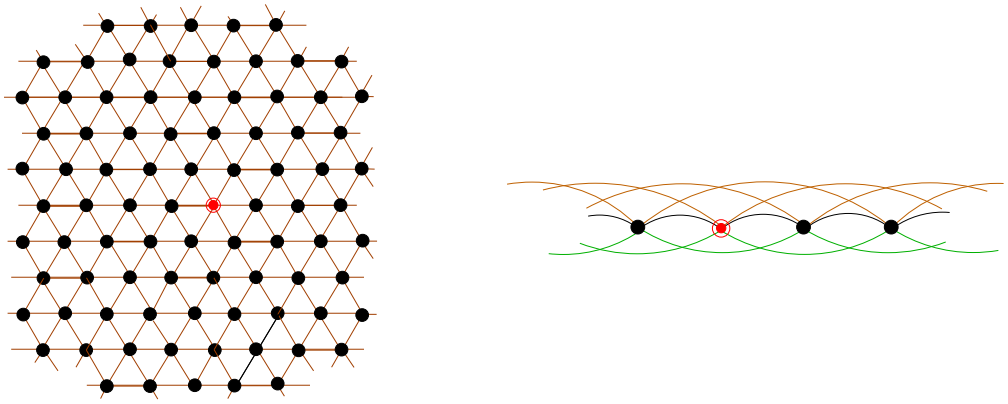
parallel of n equal matrix resistors R has an effective resistance of R/n. As a result, in $\mathbf{G}_2$ the estimation error variance of a node grows linearly with its graphical distance to the reference.

Two conclusions can be drawn from this example:

1) Two graphs with the same node degree and node/edge densities can exhibit fundamentally different scaling laws of error variance with distance.
2) The difference between the graphs become apparent when we *draw* them appropriately.

The two graphs in Figure 12 offer another – and perhaps more interesting – example of the inadequacy of node degree as a measure of denseness. It shows a triangular lattice and a 3-fuzz of a 1-dimensional lattice. The effective resistance in the 3-fuzz of the 1-dimensional lattice grows linearly with distance, whereas in the triangular lattice it grows only with the logarithm of distance, in spite of both graphs having the *same node degree of* 6. At this point the reader has to take our word for the stated growth of the effective resistance in triangular lattices, but the tools needed to prove this fact are forthcoming.

Graph Drawings



(a) A triangular 2-dimensional lattice

(b) A 3-fuzz of a 1-dimensional lattice

Fig. 12. Two different measurement graphs — a triangular 2-dimensional lattice and a 3-fuzz of a 1-dimensional lattice. Both graphs have the same node degree for every node but quite different variance growth rates with distance. Variance grows linearly with distance in the 3-fuzz of the 1-dimensional lattice (see lattice effective resistance Lemma 2). On the other hand, variance grows as logarithm of distance in the triangular lattice, though we have to wait till Theorem 4 to see why.

In graph theory, the branch of mathematics dealing with the study of graphs, 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 because generally 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/communication range" of each other. In general, this range may be defined in a complex fashion (not just determined by the Euclidean distance), but still the geometric configuration of nodes in Euclidean space plays a key role in determining the measurement graph.

Graph drawings are used to capture the geometry of graphs in Euclidean space. Figure 13 shows two different "pictures" of the same graph. From a graph theoretic point of view, the two graphs are the same because they have the same nodes and edges. However, the two graphs are

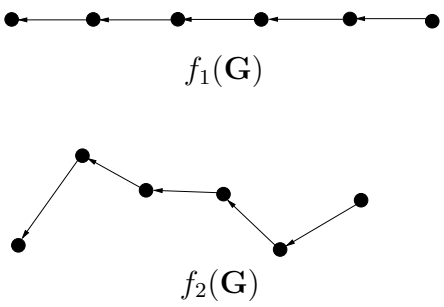


Fig. 13. Two different drawings, f_1 and f_2 of the same graph $\mathbf{G}$. Drawing $f_1(\mathbf{G})$ is a 1-dimensional drawing, whereas $f_2(\mathbf{G})$ is a 2-dimensional drawing.

"drawn" differently. A drawing of a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is simply a mapping of its nodes to points in some Euclidean space, which can formally be described by a function $f : \mathbf{V} \to \mathbb{R}^d,\ d \geq 1$. A drawing is also sometimes called a *representation* of a graph [14].

For a particular drawing f of a graph, we can define an Euclidean distance between nodes, which is simply the distance in that Euclidean space between the drawings of the nodes. In particular, given two nodes $u, v \in \mathbf{V}$ the *Euclidean distance between u and v induced by the drawing $f : \mathbf{V} \to \mathbb{R}^d$* is defined by

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

where $\|\cdot\|$ denoted the usual Euclidean norm in d-space. Note that Euclidean distances depend on the drawing and can be completely different from graphical distances. The two drawings of the same graph found in Figure 10(b) and Figure 11(a) should make this point quite clear. 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. In fact, it has infinitely many drawings.

For a sensor network, there is a *natural drawing* of its measurement graph that is obtained by associating each node to its position in 1-, 2- or 3-dimensional Euclidean space. In reality, all

sensor networks are situated in 3-dimensional space. However, sometimes it maybe more natural to draw them on a 2-dimensional Euclidean space if one dimension (for example, height) does not vary much from node to node, or is somehow irrelevant. In yet another situation, such as the one shown in Figure 4, one could draw the graph in 1-dimension since the nodes essential form a chain even though nodes are situated in 3-dimensional 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 are going to see that the Euclidean distance induced by appropriate drawings provide the right measure of distance to determine scaling laws of error variances.

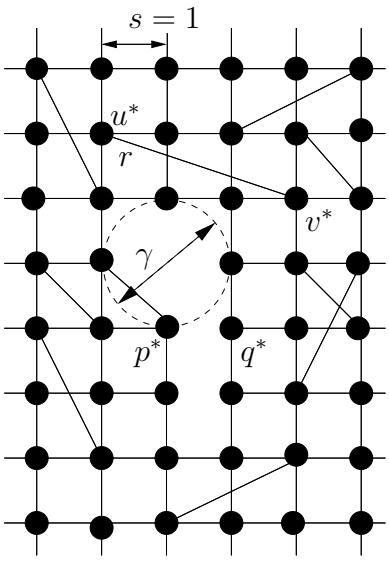


Fig. 14. Drawing of a graph in 2-dimension and the corresponding denseness/sparseness parameters: $s = 1$, $r = d_f(u^*, v^*) = \sqrt{10}$, $\gamma = 2$, and $\rho = d_f(p^*, q^*)/d_{\mathbf{G}}(p^*, q^*) = 1/5$.

Measures of Graph Denseness/Sparseness

For a particular drawing f and induced Euclidean distance d_f of a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, four parameters can be used to characterize graph denseness/sparseness. The term *minimum node*

distance denotes the minimum Euclidean distance between the drawing of two nodes

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

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

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

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

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

where the existential quantification spans over the balls B_δ in $\mathbb{R}^d$ with diameter δ. Finally, the term *asymptotic distance scaling* denotes the largest asymptotic ratio between the graphical and the Euclidean distance between two nodes as

$$\rho := \lim_{n \to \infty} \inf \Big\{ \frac{d_f(u, v)}{d_{\mathbf{G}}(u, v)} : u, v \in \mathbf{V} \text{ and } d_{\mathbf{G}}(u, v) \geq n \Big\}.$$

Essentially ρ provides a lower bound for the ratio between the Euclidean and the graphical distance for nodes that are far apart. Figure 14 shows the drawing of a graph and the four corresponding parameters s, r, γ, and ρ.

Dense Graphs

The drawing of a graph for which the maximum uncovered diameter is finite ($\gamma < \infty$) and the asymptotic distance scaling is positive ($\rho > 0$) is called a *dense drawing*. We say that a $\mathbf{G}$ *is dense in* $\mathbb{R}^d$ if there exists a dense drawing of the graph 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 still having sufficiently many edges so that a small Euclidean distance between two nodes in the

drawing guarantees a small graphical distance between them. In particular, for dense drawings there are always finite constants α, β for which

$$d_{\mathbf{G}}(u, v) \leq \alpha \, d_f(u, v) + \beta, \tag{7}$$

for every pair of nodes $u, v \in \mathbf{V}$. This fact is proved in [8]. Using the natural drawing of a d-dimensional lattice, one concludes that this graph is dense in $\mathbb{R}^d$. One can also show that a d-dimensional lattice can never be dense in $\mathbb{R}^{\bar{d}}$ with $\bar{d} > d$. For example, every drawing of a 2-dimensional lattice in the 3-dimensional Euclidean space is not dense.

Sparse Graphs

Graph drawings for which the minimum node distance is positive ($s > 0$) and the maximum connected range is finite ($r < \infty$) are called *civilized drawings*. This definition is essentially a refinement of the one given in [10], with the quantities r and s made to assume precise values. Intuitively, these drawings are "sparse" in the sense that one can keep the edges with finite lengths, without cramping all nodes on top of each other. We say that a graph $\mathbf{G}$ *is sparse in* $\mathbb{R}^d$ if *it can be drawn in a civilized manner in* d*-dimensional Euclidean space*. For example, we can conclude from the natural drawing of a d-dimensional lattice that this graph is sparse in $\mathbb{R}^d$. In fact, every h-fuzz of a d-dimensional lattice is still sparse in $\mathbb{R}^d$. However, a d-dimensional lattice can never be drawn in a civilized way in $\mathbb{R}^{\bar{d}}$ with $\bar{d} < d$. In particular, every drawing of a 3-dimensional lattice in the 2-dimensional Euclidean space is never civilized. A 3-dimensional lattice is therefore not sparse in $\mathbb{R}^2$.

The notions of graph "sparseness" and "denseness" are mostly interesting for infinite graph because every finite graph is sparse in all Euclidean spaces $\mathbb{R}^d$ for every $d \geq 1$ and no finite graph can ever be dense in any Euclidean space $\mathbb{R}^d$ for every $d \geq 1$. 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, in practice infinite graphs serve as proxies for large graphs that, from the perspective of most nodes, "appear to extend in all directions as far as the eye can see." So conclusions drawn for sparse/dense infinite graphs hold for large graphs, at least far from the graph boundaries.

Sparseness, Denseness, and Embeddings

The notions of sparseness and denseness introduced above are useful because they provide a complete characterization for the classes of graphs that can embed or be embedded in lattices, for which the lattice effective resistance Lemma 2 provides the precise scaling laws for the 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,*

$$\mathbf{G} \text{ is sparse in } \mathbb{R}^d \quad \Leftrightarrow \quad \text{there exists } h < \infty : \mathbf{G} \subset \mathbf{Z}_d^{(h)}$$

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* $\mathbf{Z}_d$*. Formally,*

$$\begin{aligned}\mathbf{G} \text{ is dense in } \mathbb{R}^d \quad \Leftrightarrow \quad & \text{there exist } h, c < \infty : \mathbf{G}^{(h)} \supset \mathbf{Z}_d \quad \& \\ & \text{for every } u \in \mathbf{V} \text{ there is a } \bar{u} \in \mathbf{V}_{\mathrm{lat}}(\mathbf{G}) : d_{\mathbf{G}}(u, \bar{u}) \leq c,\end{aligned}$$

where $\mathbf{V}_{\mathrm{lat}}(\mathbf{G})$ *denotes the set of nodes of* $\mathbf{G}$ *that are mapped to a node of* $\mathbf{Z}_d$*.*

The first statement of the lemma is essentially taken from [10] and the second statement is a consequence of results in [8]. The condition of "no multiple edges between two nodes" is not restrictive for our problems because the effective resistance between any two nodes in a graph does not change if we can replace a set of multiple, parallel edges between two nodes by a single edge with a resistance equal to the effective resistance of those parallel edges [8].

Scaling Laws for the Estimation Error Variance

We are now finally ready to characterize the scaling laws of the estimation error variance in terms of the density/sparseness properties of the measurement graph. The following theorem from [8] precisely characterizes the scaling laws by combining the Electrical Analogy Theorem 1, Rayleigh's Generalized Monotonicity Law, the lattice effective resistance Lemma 2, and the lattice embedding Theorem 3.

Theorem 4. *Consider a measurement graph* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ *with a single reference node* $o \in \mathbf{V}$ *and covariance matrices for the measurement errors* P_e, $e \in \mathbf{E}$ *that satisfy* $P_{\min} \leq P_e \leq P_{\max}$, *for every* $e \in \mathbf{E}$, *for some symmetric positive definite matrices* $P_{\min}$, $P_{\max}$. *There exist constants* $\ell, \alpha_i, \beta_i > 0$ *such that the formulas in Table II hold for every node* u *at an Euclidean distance to the reference node* o *larger than* ℓ.

Graphs that are both sparse and dense in some Euclidean space $\mathbb{R}^d$ form a particularly nice family since the scaling laws of variance with distance in such graphs are known exactly. At this point the reader may wish to check that the 2-dimensional triangular lattice in Figure 12 is both sparse and dense in 2-D, which validates the statement we made earlier that the effective resistance in it grows as the logarithm of distance.

It might be asked whether it is common for sensor networks to be sparse and/or dense in some Euclidean space $\mathbb{R}^d$. The answer happens to be "very much so", and is often seen by considering the natural drawing of the network. Recall that a natural drawing of a sensor

TABLE II

COVARIANCE MATRICES OF THE ESTIMATION ERROR FOR GRAPHS THAT ARE DENSE OR SPARSE. $d_f(u,o)$ DENOTES THE EUCLIDEAN DISTANCE BETWEEN NODE u AND THE REFERENCE NODE o, FOR A DRAWING f THAT ESTABLISHES THE GRAPH'S SPARSENESS/DENSENESS.

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

network is obtained by associating each node to its physical position in $1, 2$ or 3-dimensional Euclidean space. All natural drawings of sensor networks are likely to be sparse in 3-dimensional space, since the only requirements for sparseness are that nodes not lie on top of each other and edges be of finite length. When a sensor network is deployed in a 2-dimensional domain or when the third physical dimension is irrelevant, again the natural drawing is likely to be sparse in 2-dimensional space for the same reasons. It is slightly harder for a graph to satisfy the denseness requirements. Formally, a graph has to be infinite to be dense. However, what matters in practice are the properties of the graph "not too close to the boundary". Thus, a large-scale sensor network satisfies the denseness requirements, as long as there are no big holes between nodes and sufficiently many interconnections between them. A commonly encountered deployment model of sensor networks consists of nodes that are Poisson distributed random points in 2-dimensional space with an edge between every pair of nodes that are within a certain range. Graphs produced by this model likely to be dense in 2-dimensional for a sufficiently large range (when compared to the intensity of the Poisson process). Underwater sensors filling

a 3-dimensional volume using a similar model are likely to be dense in 3-dimensional space.

DISTRIBUTED COMPUTATION

We now answer the first question raised in this article: Is it possible to compute the optimal estimate the node variables in a distributed way using only local information? We show that it is indeed feasible, and present a distributed asynchronous algorithm to achieve this goal. The algorithm is iterative, whereby every node starts with an arbitrary initial guess for its variable and successively improves it by using the measurements on the edges incident on it and the estimates of its neighbors. The algorithm is guaranteed to converge to the optimal estimate as the number of iterations go to infinity. Moreover, it is robust to link failures and converges 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:

$$\mathcal{L}\hat{\mathbf{x}}^* = \mathbf{b}, \tag{8}$$

where $\mathcal{L}$ and $\mathbf{b}$ are defined in (4). We seek for iterative algorithms to compute the solution to (8) subject the constraints:

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

The concept of "1-hop neighbor" is determined by the measurement graph $\mathbf{G}$, in the sense that two nodes receive each other messages if the graph $\mathbf{G}$ has an edge between them (in either direction). In short, we implicitly assume bi-directional communication.

Jacobi algorithm

Consider a node u with unknown node variable x_u and imagine for a moment that the node variables for all neighbors of u are exactly known and available to u. In this case, the node u could compute its optimal estimate by simply using the measurements between u and its 1-hop neighbors. This estimation problem is fundamentally no different that 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 1-hops neighbors and the edge set $\mathbf{E}_u(1)$ consists of only the edges between u and its 1-hops neighbors. We call $\mathbf{G}_u(1)$ the *1-hop subgraph of* $\mathbf{G}$ *centered at* u. Since we are assuming that the node variables of the neighbors of u are exactly known, all of these nodes should be understood as references.

The *Jacobi algorithm* is an iterative algorithm that operates as follows:

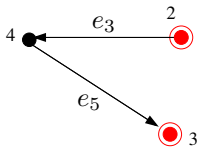
1) Each node $u \in \mathbf{V}$ picks arbitrary initial estimates $\hat{x}_v^{(0)}$, $v \in \mathbf{V}_u(1)$ for the node variables of all its 1-hop neighbors. These estimates do not necessarily have to be consistent across the different nodes.
2) At the ith iteration, each node $u \in \mathbf{V}$ assumes that its current estimates $\hat{x}_v^{(i)}$, $v \in \mathbf{V}_u(1)$ for the node variables of its neighbors are correct and solves the corresponding estimation problem associated with the 1-hop subgraph $\mathbf{G}_u(1)$. The corresponding estimate $\hat{x}_u^{(i+1)}$ turns out to be the solution to the system of linear equations

$$\Big(\sum_{e \in \mathbf{E}_u} P_e^{-1} \Big) \hat{x}_u^{(i+1)} = \sum_{e \in \mathbf{E}_u} P_e^{-1} \big(\hat{x}_{v_e}^{(i)} + a_{ue} \zeta_e \big), \tag{9}$$

where v_e denotes the 1-hop neighbor that shares the edge e with u, and a_{ue} is the (u, e)th element of the incidence matrix A. The node u then broadcasts the new estimate $\hat{x}_u^{(i+1)}$ to all its neighbors.
3) Each node then listens for the broadcasts from its 1-hop neighbors, and uses them to update its estimates $\hat{x}_v^{(i+1)}$, $v \in \mathbf{V}_u(1)$ for the node variables of its neighbors. Once all updates are received a new iteration can start.

The algorithm can be terminated at a node when the change in its recent estimate is seen to be lower than a certain pre-specified threshold value, or when a certain maximum number of iterations are completed. Figure 15 shows the relevant equations for one iteration of the Jacobi algorithms, applied to the measurement graph $\mathbf{G}$ shown in Figure 3.

The 1-hop subgraph $G_4(1)$ for the measurement graph in Figure 3.



$$G_4(1) = (\{2,3,4\}, \{e_3, e_5\})$$

Measurement model (3) for the only unknown variable x_4, when x_2 and x_3 are taken as references:

$$\underbrace{\begin{bmatrix} z_3 \\ z_5 \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}}_{\mathcal{A}_r^T} \underbrace{\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}}_{\mathbf{x}_r} + \underbrace{\begin{bmatrix} -I \\ I \end{bmatrix}}_{\mathcal{A}_b^T} \underbrace{x_4}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \epsilon_2 \\ \epsilon_5 \end{bmatrix}}_{\boldsymbol{\epsilon}}$$

Corresponding optimal estimate (4) when all measurement covariance matrices are equal to the identity matrix:

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

Jacobi iteration for node 4:

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

A similar construction based on the 1-hop subgraphs centered at nodes 2 and 3, leads to the following update equations for x_2 and x_3's estimates in the measurement graph of Figure 3:

$$\hat{x}_2^{(i+1)} = \frac{1}{4}(\hat{x}_4^{(i)} + \hat{x}_3^{(i)} + \zeta_3 + \zeta_4 - \zeta_1 - \zeta_2)$$

$$\hat{x}_3^{(i+1)} = \frac{1}{2}(\hat{x}_2^{(i)} + \hat{x}_4^{(i)} - \zeta_4 - \zeta_5)$$

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

Fig. 15. Equations for one iteration of the Jacobi algorithms by node 4 in the measurement graph $\mathbf{G}$ shown in Figure 3.

The iterative algorithm described by equation (9) can be viewed as the Jacobi algorithm to solve linear equation. Iterative techniques for solving linear equations have a rich history and a host of iterative methods have been developed for different applications, each with its own particular advantages and disadvantages. For the optimal estimation problem, the Jacobi algorithm has several benefits: it is simple, scalable, it converges to the optimal estimate under mild conditions, and is even robust with respect to temporary link failures [15]. However, its weakness lies in a relatively slow convergence rate.

Overlapping Subgraph Estimator (OSE) Algorithm

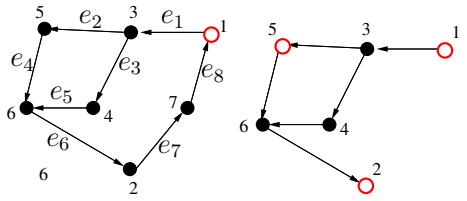
The Overlapping Subgraph Estimator (OSE) algorithm achieves faster convergence than Jacobi, while retaining its scalability and robustness properties. The OSE algorithm can be thought of as an extension of the Jacobi algorithm, in which individual nodes utilize larger subgraphs to improve their estimates. To understand how, suppose that each node broadcasts to its neighbors, not only its current estimate, but also all of the latest estimates that he got from his neighbors. In practice, we have a simple two-hop communication scheme and, in the absence of drops, at the ith iteration step, each node has the estimates $\hat{x}_v^{(i)}$ for its 1-hop neighbors and the (older) estimates $\hat{x}_v^{(i-1)}$ for its 2-hop neighbors (that is, the nodes at a graphical distance equal to two).

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

1) Each node $u \in \mathbf{V}$ picks arbitrary initial estimates $\hat{x}_v^{(-1)}$, $v \in \mathbf{V}_u(2) \setminus \mathbf{V}_u(1)$ for the node variables of all its 2-hop neighbors. These estimates do not necessarily have to be consistent across the different nodes.
2) At the ith iteration, each node $u \in \mathbf{V}$ assumes that the estimates $\hat{x}_v^{(i-2)}$, $v \in \mathbf{V}_u(2) \setminus \mathbf{V}_u(1)$ (that it received through its 1-hop neighbors) are correct and solves the corresponding optimal estimation problem associated with the 2-hop subgraph $\mathbf{G}_u(2)$. In particular, it 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 1-hop subgraph $\mathbf{G}_u(1)$, and $\mathcal{L}_{u,2}, \mathbf{b}_u$ are defined for the subgraph $G_u(2)$ as $\mathcal{L}, \mathbf{b}$ are for $\mathbf{G}$ in (8). 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 pre-specified 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 $\hat{x}_v^{(i)}$, $v \in \mathbf{V}_u(1)$ previously received from its 1-hop neighbors are then broadcasted to all its 1-hop neighbors.
3) Each node u then listens for the broadcasts from its neighbors, and uses them to update its estimates for the node variables of all of its 2-hop neighbors $\mathbf{V}_u(2) \setminus \mathbf{V}_u(1)$. Once all updates are received a new iteration can start.

As in the case of Jacobi, the termination criteria varies depending on the application, and nodes use measurements and covariances obtained initially for all future time. Figure 16 shows a 2-hop subgraph used by the OSE algorithm.

In the previous description of the OSE algorithm, we have assumed that communication is essentially synchronous and that all nodes always received broadcasts from all of their 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 its neighbors or after receiving multiple messages from the same neighbor. A timeout mechanism can be used for this purpose: each node resets a timer as it broadcasts its most recent estimates and when this timer reaches a pre-specified timeout value, the node initiates a new iteration, regardless of



(a) $\mathbf{G}$ (b) $\mathbf{G}_4(2)$

Fig. 16. (a) A measurement graph $\mathbf{G}$ with node 1 as reference, and (b) a 2-hop subgraph $\mathbf{G}_4(2)$ centered at node 4. While running the OSE algorithm, node 4 treats $1, 5$ and 2 as reference nodes in the subgraph $\mathbf{G}_4(2)$ and solves for the unknowns x_3, x_4 and x_6.

whether or not it received messages from all its 1-hop neighbors. If a message is not received from one of its neighbors, it uses "old data" for the computations of the new iteration; and if it received several messages from the same neighbors, it only uses the most recent data.

One could also design a *h-hop OSE* algorithm by letting every node utilize a h-hop subgraph centered at itself, where h is some (small) integer. The resulting algorithm would be a straightforward extension of the 2-hop OSE just described, except that at every iteration, individual nodes have to transmit to their neighbors larger amounts of data than in 2-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.

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 both when the covariance matrices P_e, $e \in \mathbf{E}$ are all equal or when they are all diagonal (but not necessarily equal).*

The proof of this theorem makes use of results from the literature on parallel computing, by exploring the connection between the OSE algorithm and an iterative method for solving linear

equations that we call Asynchronous Filtered Weighted Additive Schwarz (AFWAS) method [16]. The AFWAS is closely related to the Weighted Additive Schwarz method reported in [17]. 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 somehow relaxed. In fact, it appear to be quite difficult to make the algorithm not converge to the correct solution. In the simulations we have done (which we are going to describe in a moment), this assumption is violated but the algorithm always converges.

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 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 have no good estimate of their variables. Subsequently, in their estimate updates each node only includes in their 1- or 2-hop subgraph 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 1-hop neighbors of the references are able to compute finite estimates, in the second iteration the 2-hop neighbors of the references are also able to obtain finite estimates and so forth until all nodes have finite estimates. Since the flagged initialization only affects the initial stage of the algorithms, it does not affect their convergence properties.

Simulations

In this section, we present a few numerical simulations to study 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 an unit square (see Figure 17). 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 $\sigma_r = 15\% r_{\max}$ and the angle measurements are corrupted with zero-mean additive Gaussian noise with standard deviation $\sigma_\theta = 10^o$. When the range and bearing measurement errors are independent and have variances independent of distance, for a noisy measurement (r, θ) of true range and angle (r_o, θ_o), a little algebra shows that the covariance matrix of the measurement error $\zeta_{u,v} = [r\cos\theta, \;\; r\sin\theta]^T$ is given approximately by:

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

where $x_o = r_o\cos\theta_o$ and $y_o = r_o\sin\theta_o$. Assuming that the two 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. It is clear that the covariances are not diagonal and different measurements have different covariances, so 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 17, together with the true locations.

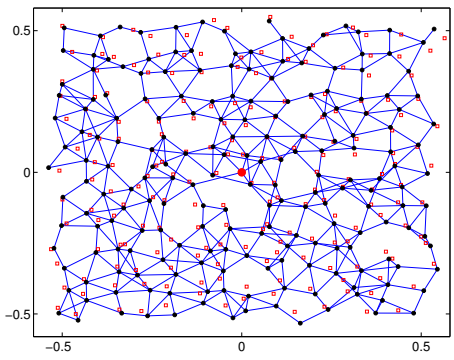


Fig. 17. A sensor network with 200 nodes randomly distributed in a unit square area. The edges of the measurement graph are shown as line segments connecting the true nodes positions, which are shown as black circles. Two nodes with an edge between them have a measurement of their relative positions in the plane. The red squares are the positions estimated by the (centralized) optimal estimator. A single reference node is located at the origin.

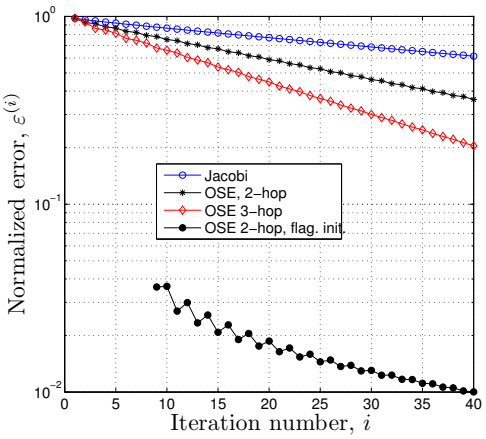


Fig. 18. Performance comparison of Jacobi and OSE. Normalized error is defined as $\varepsilon^{(i)} = \frac{\|\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^*\|}{\|\hat{\mathbf{x}}^*\|}$ where $\hat{\mathbf{x}}^{(i)}$ is the vector of estimates at the i-th iteration and $\hat{\mathbf{x}}^*$ is the optimal estimate. Except for the case with flagged initialization, simulations are run with all initial estimates of node variables set to 0. For the flagged OSE, the normalized error can only be defined after iteration number 8 because until then not all nodes have valid (finite) estimates.

Figure 18 compares the normalized error as a function of iteration number for the Jacobi and OSE algorithms. Two versions of the OSE are tested - OSE 2-hop and 3-hop. The parameter λ for OSE is chosen somewhat arbitrarily as 0.9. The straight lines in the log-scaled graph show the exponential converge of both algorithms, and the faster converge rate of the OSE algorithm. This figure also shows a dramatic improvement achieved with the flagged initialization scheme. With flagged initialization, the 2-hop OSE algorithm is able to estimate the node positions within 3% of the optimal estimate after only 9 iterations. Figure 19 shows the performance of the 2-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

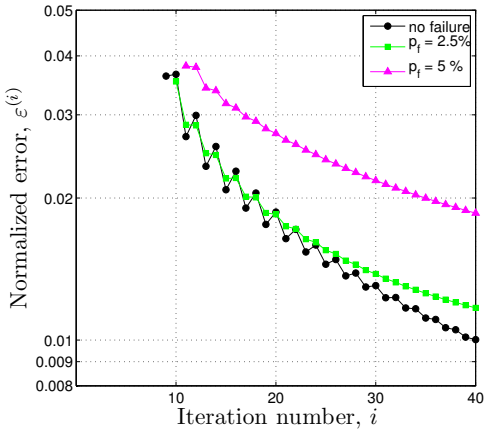


Fig. 19. Normalized error of 2-hop OSE as a function of iteration number in the presence of link failures, where the normalized error $\varepsilon^{(i)}$ has been defined in Figure 18. Two different failure probabilities are compared with the case of no failure. With higher failure probability performance degrades, but the error is seen to decrease with iteration count even with large failure probabilities.

The OSE algorithm exhibits a faster convergence rate than Jacobi, as evident from the simulations. However, faster convergence is achieved at the expense of each node sending more data to its 1-hop neighbors, because each node broadcasts its own estimate as well as the estimates that he previously received from its 1-hop neighbors. Hence the messages needed by OSE are d times longer than the ones required by Jacobi, where d denotes the node degree. One may then ask whether there is any significant advantage to using the OSE algorithm.

Energy consumption in wireless communication has a complex dependence on radio hardware, underlying physical and medium access control layer protocols, network topology and a host of other factors. Due to the overheads introduced by these factors, sending a short message offers no advantage in terms of energy consumption over sending a somewhat longer message (see [18]). In fact, transmitted energy per bit in a packet *decreases monotonically* upto 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, and sending a

packet d times longer may cost negligible additional energy. In such cases, the faster convergence of OSE yields the greatest benefits because this algorithm requires a smaller number of iterations (and therefore a smaller number of messages) to achieve a desired error tolerance, resulting in lower energy consumption and increased network life.

CONCLUSION

Large-scale sensor networks give rise to estimation problems that have a rich graphical structure. We studied one of these problems in terms of (1) how such estimate can be efficiently computed in a distributed manner and (2) 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 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, dense and sparse, for which we can find upper and lower bounds on the variance growth with distance.

The tools used to investigate estimation error scaling rely on the analogy between estimation error covariance and matrix-valued effective resistance, and the monotonicity of the matrix resistances with respect to embeddings in lattices. Analogy with electrical networks have been used to construct elegant solutions to a range of problems, notably those concerned with random walks in graphs [10, 21]. The monograph by Doyle and Snell [10] in particular helped us immensely by bringing this way of thinking to our attention. The tools we have developed for generalized electrical networks with matrix-valued resistances appear to be applicable to study a wide variety of other problems defined on large graphs [22]. We also found the literature on distributed computation to be an especially rich source of inspiration for the design of distributed algorithms for sensor networks.

REFERENCES

[1] A. Arora, R. Ramnath, 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 *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp. 102–108, 2005.

[2] D. Estrin, D. Culler, K. Pister and G. Sukhatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, vol. 1, no. 1: 59–69, 2002.

[3] R. Karp, J. Elson, D. Estrin and S. Shenker. Optimal and global time synchronization in sensornets. Tech. rep., Center for Embedded Networked Sensing,Univ. of California, Los Angeles, 2003.

[4] C. F. Gauss. Theoria motus corporum coelestium in sectionibus conicis solem ambientum (the theory of the motion of heavenly bodies moving around the sun in conic sections). Hamburg, Germany, 1809. URL `http://134.76.163.65/agora_docs/137784TABLE_OF_CONTENTS.html`.

[5] R. Adrain. Research concerning the probabilities of the errors which happen in making observations. *The Analyst*, vol. I: 93–109, 1808. Article XIV.

[6] B. Hayes. Science on the farther shore. *American Scientist*, vol. 90, no. 6: 499–502, 2002.

[7] J. M. Mendel. *Lessons in Estimation Theory for Signal Processing, Communications and Control*. Prentice Hall P T R, 1995.

[8] P. Barooah and J. P. Hespanha. Optimal estimation from relative measurements: Electrical analogy and error bounds. Tech. rep., University of California, Santa Barbara, 2003. URL `http://www.ccec.ece.ucsb.edu/~pbarooah/publications/TR1.html`.

[9] J. C. Maxwell. *A Treatise on Electricity and Magnetism*. Clarendon, 1918, reprinted by

Dover, New York, 3rd edn., 1954.

[10] P. G. Doyle and J. L. Snell. Random walks and electric networks. Math. Assoc. of America, 1984.

[11] P. G. Doyle. Application of Rayleigh's short-cut method to Polya's recurrence problem. online, 1998. URL http://math.dartmouth.edu/~doyle/docs/thesis/thesis/.

[12] D. Niculescu and B. Nath. Error characteristics of ad hoc positioning systems (APS). In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2004.

[13] A. Savvides, W. Garber, , R. Moses and M. b. Srivastava. An analysis of error inducing parameters in multihop sensor node localization. *IEEE Transactions on Mobile Computing*, vol. 4, no. 6: 567–577, 2005.

[14] C. Godsil and G. Royle. *Algebraic Graph Theory*. Graduate Texts in Mathematics. Springer, 2001.

[15] P. Barooah and J. P. Hespanha. Distributed optimal estimation from relative measurements. In *Proceedings of the 3rd International Conference on Intelligent Sensing and Information Processing (ICISIP)*, pp. 226–231, Bangalore, India, 2005.

[16] P. Barooah, N. M. da Silva and J. P. Hespanha. Distributed optimal estimation from relative measurements in sensor networks: Applications to localizationa and time synchronization. Tech. rep., University of California, Santa Barbara, 2006. URL http://www.ccec.ece.ucsb.edu/~pbarooah/publications/TR3.html.

[17] A. Frommer, H. Schwandt and D. B. Szyld. Asynchronous weighted additive Schwarz methods. *Electronic Transactions on Numerical Analysis*, vol. 5: 48–61, 1997.

[18] R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Sinha, A. Wang and A. Chandrakasan. Low-power wireless sensor networks. In *Keynote Paper, 28th European Solid-State Circuits Conference (ESSCIRC)*, Florence, Italy, 2002.

[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 *Proceedings of Design, Automation and Test in Europe (DATE)*, vol. 1, pp. 196–201, 2005.

[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 *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN)*, pp. 367 – 372, 2004.

[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. of the 21st Annual ACM Symposium on Theory of Computing*, pp. 574–586, 1989.

[22] P. Barooah and J. P. Hespanha. Graph effective resistances and distributed control: Part I — spectral properties and applications, 2006. Submitted, available at `http://www.ece.ucsb.edu/~hespanha/published.html`.

Author Bios

Prabir Barooah received his B.Tech and M.S. degrees in Mechanical Engineering from the Indian Technology, Kanpur, India in 1996 and University of Delaware, Newark, DE in 1999, respectively. From 1999 to 2002 he worked at United Technologies Research Center, and in January 2003 joined the Electrical and Computer Engineering Ph. D. program at University of California, Santa Barbara. He is a recepient of the NASA group achievement award and the best paper award at the 2nd Int. Conf. on Intelligent Sensing and Information Processing. His research interests include active combustion control, system identification, and distributed control and estimation in large-scale networks.

João P. Hespanha 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. He currently holds an Associate Professor position with the Department of Electrical and Computer Engineering, the University of California, Santa Barbara. From 1999 to 2001, he was an Assistant Professor at the University of Southern California, Los Angeles. Dr. Hespanha is the associate director for the Center for Control, Dynamical-systems, and Computation (CCDC) and an executive committee member for the Institute for Collaborative Biotechnologies (ICB), an Army sponsored University Affiliated Research Center (UARC).

His research interests include hybrid and switched systems; the modeling and control of communication networks; distributed control over communication networks (also known as networked control systems); the use of vision in feedback control; stochastic modeling in biology; the control of haptic devices, and game theory. He is the author of over one hundred technical papers and the PI and co-PI in several federally funded projects.

Dr. Hespanha 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 Automatica Theory/Methodology best paper prize, and the best paper award at the 2nd Int. Conf. on Intelligent Sensing and Information Processing. Since 2003, he has been an Associate Editor of the IEEE Transactions on Automatic Control. More information about Dr. Hespanha's research can be found at http://www.ece.ucsb.edu/ hespanha

Contact Information:
Joã P. Hespanha
Addres: Room 5157, Engineering I
Dept. of Electrical & Computer Eng.
University of California
Santa Barbara, CA 93106-9560, USA
email: hespanha@ece.ucsb.edu
Tel: +1(805) 893 7042
Fax: +1(805) 893 3262