Think Globally, Act Locally: On the Reshaping of Information Landscapes

Andreas Loukas Marco Zuniga Matthias Woehrle Marco Cattani Koen Langendoen

Embedded Software Group
Delft University of Technology, The Netherlands
{a.loukas, m.a.zunigazamalloa, m.woehrle, m.cattani, k.g.langendoen}@tudelft.nl

ABSTRACT

In large-scale resource-constrained systems, such as wireless sensor networks, global objectives should be ideally achieved through inexpensive local interactions. A technique satisfying these requirements is *information potentials*, in which distributed functions disseminate information about the process monitored by the network. Information potentials are usually computed through local aggregation or gossiping. These methods however, do not consider the topological properties of the network, such as node density, which could be exploited to enhance the performance of the system.

This paper proposes a novel aggregation method with which a potential becomes sensitive to the network topology. Our method introduces the notion of *affinity spaces*, which allow us to uncover the deep connections between the aggregation scope (the radius of the extended neighborhood whose information is aggregated) and the network's Laplacian (which captures the topology of the connectivity graph). Our study provides two additional contributions: (*i*) It characterizes the convergence of information potentials for static and dynamic networks. Our analysis captures the impact of key parameters, such as node density, time-varying information, as well as of the addition (or removal) of links and nodes. (*ii*) It shows that information potentials are decomposed into wave-like eigenfunctions that depend on the aggregation scope. This result has important implications, for example it prevents greedy routing techniques from getting stuck by eliminating local-maxima. Simulations and experimental evaluation show that our main findings hold under realistic conditions, with unstable links and message loss.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks*

Keywords

Aggregation; Diffusion; Information Discovery; Unimodality

IPSN'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.
Copyright 2013 ACM 978-1-4503-1959-1/13/04 ...$15.00.

1. INTRODUCTION

This paper focuses on two problems pertaining to sensor networks: *information aggregation and information discovery.* These problems appear when sensor nodes need to act locally in response to a process that is monitored globally. The monitored processes may relate to the intrinsic properties of the network operation, e. g., density, channel congestion, algorithm faults, load, or to properties related to the physical environment, e. g., monitoring the climate, traffic congestion or the movement of crowds. To avoid centralized solutions where nodes report their measurements to a single point, the sensor network community has explored the concept of information potentials [15, 17, 19, 21]. Potentials are formed through the exchange of local information and provide gradients that achieve objective-specific goals, such as routing through areas with low traffic congestion.

It is known that the computation of *useful* information potentials exposes a fundamental dilemma. How much information should a node take into account? How wide should the aggregation scope be? More information leads to more informed decisions, but it comes at the cost of increased message complexity. Our work moves beyond this dilemma and proposes that *the quality of information potentials depends not only on the extent of the aggregation scope, but also on the connectivity properties of the network.* Hence, the construction of information potentials should be tailored towards the specific characteristic of the network topology.

Hitherto, two distributed approaches have been used for aggregating information in the vicinity of nodes. The first approach exploits the (possible) spatial correlation of information processes, and requires nodes to aggregate information at a decaying function of distance [10, 13, 18]. This approach requires location information and it is mainly aimed at *physical* processes with strong spatial correlations, such as temperature. The distance-based approach does not perform well for processes related to the network operation (for example, load-balanced routing), because the information of these processes depend more on the network topology than on the inter-node distances. The second approach overcomes the limitations of the first by using constrained averaging; nodes aggregate all values within their k-hop vicinity. This approach is location-independent, but it does not distinguish the relative significance of information. This means that a neighbor with degree one has the same relevance as a high-degree neighbor. Additionally, all information within k-hops is aggregated uniformly, irrespectively of whether it is one or

k hops away. Within this context, our paper provides three main contributions.

Contribution 1: an aggregation method that is sensitive to topological features (Section 3). We introduce a subtle yet important change to traditional constrained averaging algorithms. This change allows us to model information dissemination as a random walk starting at a given node, and then, moving (randomly) for k steps. The distribution of the potential destinations of these random walks captures the underlying connectivity of the graph. Similar to constrained averaging, our approach requires no location information, it is simple, and decentralized. Yet, it is also sensitive to the relative significance of information, as well as to the network topology. The latter leads to a key advantage: our algorithm can be modeled by connectivity matrices, which allows us (*i*) to use standard linear algebra techniques to analyze the convergence of this new way of forming information potentials, and (*ii*) to use spectral graph theory to demonstrate how information potentials can reshape the landscapes of information processes.

Contribution 2: an analysis of the convergence of information potentials on static and dynamic networks (Sections 4 and 5). With respect to convergence analysis, our study provides the following findings:

- The algorithm converges exponentially and the convergence is faster in dense regular networks.
- There is a critical aggregation scope under which network dynamics hardly matter. Above this threshold, higher dynamics require narrower aggregation scopes for faster convergence.

In practice, these findings provide some *dos and don'ts* for information potential deployments. Avoid concave shapes (dents), void regions and radio transceivers with high sensitivity variance because they increase degree heterogeneity (irregularity). Adding nodes to a deployment helps more if they are uniformly distributed. Prior to a deployment, analyze the dynamics of the network, which might be caused by the movement of people, the failure/addition of nodes, etc.; if the dynamics are high, it is better to avoid using very wide aggregation scopes. If the dynamics are low, the extent of the aggregation scope does not play a major role, i. e., feel free to explore the trade-off between amount-of-information and message complexity.

Contribution 3: a framework to shape the landscapes of information potentials (Section 6). Information potentials resemble landscapes with valleys and peaks; Figure 1 depicts an example of our method. By borrowing results from spectral graph theory, we reveal deep connections between these landscapes and the eigenfunctions of the normalized Laplacian:

- The eigenfunctions of the normalized Laplacian are wave-like and characterize the spatial variation of the monitored process.
- The slope of our aggregation function shapes the potential landscape. In particular, there exists a critical slope under which the potential becomes unimodal, i. e., it features a single extremum.

To understand *the practical importance* of these results, consider an information potential that needs to smoothen the peaks formed by nodes with abnormally high (or low) values with respect to their neighbors. The spatial variation of information is captured by the eigenfunctions. The higher the variation, the higher the frequency; and our aggregation method filters eigenfunctions based on their frequency. In effect, this means that an information landscape can be smoothened to any desired level: the higher the slope, the smoother the landscape. Beyond a threshold, the continuous elimination of higher-frequency eigenfunctions leads to the elimination of local extrema until the point where a single maxima is present.

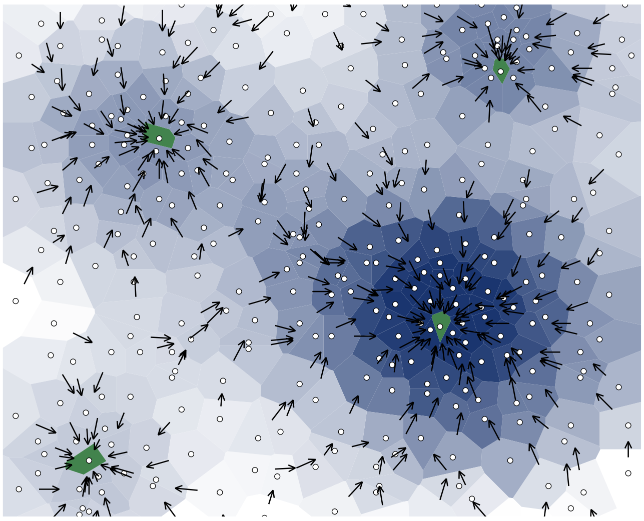

Figure 1: An information potential landscape which considers node density. Nodes within green cells have the maximum potential (highest relative density) among all of their neighbors. The arrows reveal the potential gradient.

Section 7 validates some of our analysis by simulations and experiments on a 100+ node testbed.

Applications of information potentials. The quintessential application of information potentials in sensor networks is greedy routing [19]. Still, the advent of smart mobile embedded devices, e. g., smartphones, makes information potentials also relevant to other practical applications, such as traffic management and crowd management.

Greedy routing. In greedy routing, nodes use gradients to forward information greedily. In these scenarios, the presence of local maxima is undesirable as it prevents packets from arriving at the intended destination. Our work eliminates this issue by providing unimodality (Contribution 3). It is important to remark that the work of Lin et al. [15] also provides unimodality. However, their approach, which is based on harmonic functions, relies on a larger set of assumptions. A detailed discussion of the similarities and differences of our approach with this and other studies is presented in Section 8.

In a more general context, information potentials can be thought of as distributed primitives that (*i*) facilitate the discovery of areas of high (low) level of information, and (*ii*) provide mobile entities an efficient way of navigating towards these areas. The mobile entities can be packets, people, robots, cars, etc. In a preliminary version of this study [16], we showed through simulations how our algorithm is used for rendezvous coordination, where a swarm of mobile entities identify the nodes with the largest potentials, and the rest of the swarm moves towards their closest high-potential node in real time. Below, we describe two applications that can benefit from our work on information potentials.

Crowd management. Our interest in information potential algorithms was sparked by their ability to support the management of large crowds in open air festivals. In these type of festivals, crowds of hundreds of thousands gather in city centers within confined spaces. For safety reasons, the crowds should not exceed densities above a given margin[1]. As part of two larger projects involving several institutions (D2S2 and EWiDS), we focus on the facilitation of crowd management through self-monitoring. We aim at providing attendees with coin-size wearable devices, each equipped with sensors, actuators (light), and a wireless transceiver. In this way, information potentials will be able to monitor the density of people in real time and warn attendees if their surrounding density exceeds safe levels. The attendees will also have access to the gradients formed by the potentials to move to areas with lower densities.

Traffic management. Parking management is a highly important topic in public policy. As reported by Shou [20], on average 30% of traffic in downtown areas is due to vehicles looking for parking. Cities such as San Francisco and London are installing wireless sensors on individual parking spots that report information to a central server to help drivers in finding available parking[2]. Information potentials are a low-cost efficient distributed alternative for the current centralized methods. Neighbouring meters could exchange their occupancy information and guide cars towards free parking areas by communicating with the drivers' smartphones via bluetooth. Information potentials have the added advantage of *aggregating* the information of nearby free parking spots into "more/less crowded" regions, as opposed to tracking individual spaces.

Overall, we believe that information potentials have an important role to play in the future of SmartCities. As more and more wireless devices are embedded in our daily surroundings and more data is harvested from them, a central problem will be how to guide the various mobile entities towards the areas that they are mostly interested in. This work describes an alternative that is simple, decentralized, and generic enough such that it can be shaped to suit the needs of the application at hand.

2. MODEL AND NOTATION

Network model. We model a network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set of cardinality n and $\mathcal{E}$ is the edge set. Two nodes are neighbors $i \sim j$, if $(i,j) \in \mathcal{E}$. The neighborhood of a node is captured by the neighbor set $\mathcal{V}_i = \{j : i \sim j\}$ and the adjacent edge set $\mathcal{E}_i = \{(i,j) : i \sim j\}$, with cardinality n_i and m_i, respectively. Node density is $d_i = n_i + 1$, while the max and min densities for all $i \in \mathcal{V}$ are d_{max} and d_{min}.

Matrix notation. We describe $\mathcal{G}$ through its $n \times n$ adjacency matrix A, where $A_{ij} = 1$ if $i \sim j$ or $i = j$, and $A_{ij} = 0$ otherwise. Matrix A is symmetric. We let D be the diagonal node density matrix with $D_{ii} = d_i \; \forall i \in V$ and zero otherwise. With A and D, we define the row-stochastic transition matrix $P = D^{-1}A$, the elements of which describe the transition probability of random particle moving randomly on $\mathcal{G}$. The eigenvalues of P are denoted by μ_k. The transition matrix is similar to the well known normalized Laplacian $\mathcal{L}$ [7]. For the spectral analysis, we use a modified Laplacian matrix that considers self-loops: $\mathcal{L}_{ij} = 1 - 1/d_i$ if $i = j$, $\mathcal{L}_{ij} = 1/\sqrt{d_i d_j}$ if $i \sim j$, and $\mathcal{L}_{ij} = 0$ otherwise.

3. INFORMATION POTENTIALS

This section describes our method of computing information potentials and explains how a potential captures information with respect to the topology of the network.

3.1 Distributed computation

In a nutshell, an information potential is a function that maps local information to a value more meaningful within the global network context. For instance, the local information may be the traffic load of the node, and the potential could be a relative value that states how high (or low) this traffic is with respect to other areas in the network – to facilitate, for instance, load balanced routing. Formally, let $x, y : \mathcal{V} \to \mathbb{R}$ be functions over $\mathcal{V}$ that assign a real value to each node. We call function x the *process information*, as $x(i)$ is a value that is sensed or computed by each node $i \in \mathcal{V}$ and represents the quantity or quality of a process in the vicinity of i. Function y is the information potential derived from x.

The computation of an information potential, described in Algorithm 1, is iterative. For the duration of each round, nodes $i \in \mathcal{V}$ exchange their potential $y(i)$ with their neighbors (lines 5 and 7). Since nodes do not exchange $x(i)$, they affect their neighbors indirectly by changing the average neighborhood potential. At the end of a round, nodes update their potential to the weighted sum of their local information process $x(i)$ and of the average over the most recent potential values, including their own (line 11). Intuitively, nodes behave like anchors, each pulling the neighborhood average towards its own $x(i)$ with a force that is proportional to the difference between $x(i)$ and the average. The force also depends on a parameter which we call *inhibiting factor*, and lies in $0 < \varphi \leq 1$. In the final step (line 12), all received values are discarded and the round ends. The algorithm converges when all forces are balanced. Note that the algorithm includes no termination condition; it runs indefinitely, continuously adapting to any network or information dynamics. If no dynamics are expected, termination is locally decided by comparing the difference of the potential at consecutive rounds against some error threshold.

The reuse of the information $x(i)$ throughout the computation (line 11) differentiates our work from: (*i*) *average consensus*, in which y is initialized to x and, at each round t, nodes simply average their values, $y_{t+1} = Py_t$ [7], as well as from (*ii*) *broadcast consensus* [1], in which each node i

[1] In 2010, a crowd rush in a popular electronic dance festival in Germany ended up with 21 deaths and more than 500 injured people.

[2] The system for San Francisco is SFpark (`spark.org`) and for London is the Bay Sensor Technology (`www.westminster.gov.uk/services/transportandstreets/parking/bay-sensor-technology`)

Algorithm 1 Potential computation

Require: Factor $\varphi \in (0, 1]$ and i a unique node id.

```
1:  initialize
2:      $y(i) \leftarrow x(i)$
3:      $\mathcal{S}(i) \leftarrow \{y(i)\}$                                ▷ state set
4:  event OnTransmit
5:      broadcast $\{i, y(i)\}$ to all neighbors
6:  event OnReceive($j, y(j)$)
7:      $\mathcal{S}(i) \leftarrow \mathcal{S}(i) \cup \{y(j)\}$            ▷ keep only latest values
8:  event OnRoundEnd
9:      update $x(i)$
10:     $d(i) \leftarrow |\mathcal{S}(i)|$                                  ▷ local density
11:     $y(i) \leftarrow (1-\varphi) \sum_{j \in \mathcal{S}(i)} \frac{y(j)}{d(i)} + \varphi\, x(i)$   ▷ compute
12:     $\mathcal{S}(i) \leftarrow \{y(i)\}$                                ▷ clear state
```

computes a weighted average after every received broadcast from $j \sim i$, $y_0(i) = x(i)$ and $y_{t+1}(i) = \gamma y_t(i) + (1-\gamma) y_t(j)$. As we show in the following, this simple alteration of average consensus gives rise to very interesting properties which relate to affinity spaces (Section 3.2) and the spectrum of the graph Laplacian (Section 6), improves the convergence compared to deterministic consensus (Section 4), and increases the algorithm's resilience to message loss (Section 7).

3.2 Algorithmic analysis

We will now describe how this simple distributed algorithm captures the topology of the network. For the sake of clarity and conciseness, our analysis assumes that (i) nodes operate in synchronous rounds, (ii) at the end of which nodes have received at least one message from each of their neighbors. In Section 7 we show that, *in practice, neither assumption is necessary for the correct and timely operation of the algorithm.* We will also assume, for now, that the information x is constant, i.e., line 9 in the algorithm does not have any valid effect at this point. This assumption will be lifted in Section 4.2, which studies the convergence under time-varying x_t. Formula (1) rewrites Algorithm 1 in an iterative matrix form, where y_t is the y vector after t iterations.

$$y_{t+1} = (1-\varphi) P y_t + \varphi\, x \tag{1}$$

At round t, the potential y_t is given by

$$y_t = (\psi P)^t y_0 + \varphi \sum_{k=0}^{t-1} (\psi P)^k x, \tag{2}$$

where, for brevity, we set $\psi = 1 - \varphi$. When t grows to infinity, the potential y has the following closed-form expression:

$$y = \lim_{t \to \infty} y_t = \varphi \sum_{k=0}^{\infty} (\psi P)^k x \tag{3}$$

Note that Formula (2) converges under all cases. As t grows larger, $(\psi P)^t$ approaches asymptotically zero because $\psi < 1$ and P^t is a row-stochastic matrix that converges to the stationary distribution of a uniform random walk on $\mathcal{G}$. The decay of $(\psi P)^t$ removes any influence of the initial state y_0 on the potential y. In Sections 4 and 5 we analyze the convergence rate for information processes in detail, the rest of this section provides more insights on the potential itself.

Affinity spaces. In contrast to definitions of affinity that result from other metrics of distance, such as euclidean distance [13], our method aggregates information based on a new type of affinity that is very sensitive to the topology of a network. As we will observe, this topological sensitivity is achieved because nodes aggregate information in a sort of "random-walk" manner.

Formula (3) expresses the potential as an infinite sum that, at each round k, changes information x with a weighting factor of $(\psi P)^k$. P^k_{ij} expresses the probability of a randomly moving particle starting from node i and reaching node j in k steps. The better the connectivity between i and j and the shorter the path, the higher the probability. In other words, instead of averaging the information within the k-th range of a node, our method assigns higher significance to the information residing in nearby nodes (close connectivity) and in nodes with higher centrality (better connectivity). In most graphs, centrality is an important metric that captures the "importance" of the information.

The inhibiting factor φ determines the aggregation scope: (i) When $\varphi = 1$, there is no exchange of information and hence the network topology plays no role. As Formula (1) shows, the potential has the same value as the process information ($y = x$). (ii) When $\varphi \to 0^+$, the aggregation scope is global and the network topology plays its greatest role. The iterative formulation reduces to the well studied average consensus $y_{t+1} = P\, y_t$ [7]. As $t \to \infty$, the potential gets closer to $\mathbb{1}\pi^\top x$, where $\mathbb{1}$ is the $n \times 1$ vector with all elements equal to one and π is the stationary distribution of the transition matrix. Within these extremes, the network has ample flexibility to shape information potentials according to the requirements of the application. In Section 6, we will analyze this characteristic in more detail and its impact on greedy search techniques. Note that, since the geometric series ψ^k converges to $^1/_\varphi$, the multiplication with φ serves as a normalization.

It is important to highlight that while the analysis considers the global connectivity matrix P, *the algorithm only requires communication with 1-hop neighbors.* The advantage of our simple and distributed algorithm is that it entails such global behavior inherently.

4. CONVERGENCE IN STATIC GRAPHS

In this section we bound the rate of convergence in static graphs for invariant and time-varying information processes. We limit our analysis to $\varphi \in (0, 1)$ as for $\varphi = 1$ the algorithm converges instantaneously. We will first present our theoretical analysis, and then, discuss its implications. In our study, we define the ℓ_2-distance of an information potential y_t as $\|y - y_t\|$, that is, its distance to the steady state.

4.1 Time-invariant information

We first consider the case when the information x stays constant over time and the graph $\mathcal{G}$ is static. In the next sections, we will remove these constraints.

Theorem 1. *After t rounds, the ℓ_2-distance of an information potential on a static graph $\mathcal{G}$ with information process x is bounded by $\varepsilon_t \le e^{-\varphi t}(c^2 + c)\,\|x\|$, where $c = \frac{n}{2m} d_{max} + \xi \mu_2$ and $\xi = \sqrt{d_{max}/d_{min}}$.*

PROOF. The ℓ_2-distance at the t-th round is

$$\begin{aligned}\varepsilon_t &= \|y - y_t\| \\ &= \|\,\varphi\sum_{k=0}^{\infty}(\psi P)^k x - \varphi\sum_{k=0}^{t-1}(\psi P)^k x - (\psi P)^t y_0\| \\ &= \big\|(\psi P)^t\Big(\varphi\sum_{k=0}^{\infty}(\psi P)^k x - y_0\Big)\big\|. \end{aligned}\tag{4}$$

Using a known bound on $P^t x$ [7], we get that for $t \geq 0$ and an arbitrary vector x,

$$\begin{aligned}\|(\psi P)^t x\| &\leq \Big(\psi^t \frac{n}{2m} d_{max} + (\psi\mu_2)^t\,\xi\Big)\,\|x\| \\ &\leq \psi^t c\,\|x\|. \end{aligned}\tag{5}$$

Above, $m = |\mathcal{E}|$ is the number of edges considering self-loops, μ_2 is the second eigenvalue of the transition matrix P and $\xi = \sqrt{d_{max}/d_{min}}$ quantifies the degree irregularity of $\mathcal{G}$. In the last step, $(\psi\mu_2)^t \leq \psi^t\mu_2$ because $\mu_2 < 1$ and $0 \leq \psi < 1$. This loosens the error bound when $\mu_2 > 0$, but allows to estimate the necessary number of rounds t until the algorithm converges ε-close to the stable state. One can achieve a tighter bound if an estimate of t is not required. Substituting Inequality (5) into Formula (4) we have that

$$\varepsilon_t \leq \psi^t c\,\big\|\varphi\sum_{k=0}^{\infty}(\psi P)^k x - y_0\big\|. \tag{6}$$

From Formula (2), we observe that the choice of the initial state y_0 is irrelevant to the stable state y. Nevertheless, a reasonable step is to set $y_0 = x$. The normed difference in Inequality (6) then is simplified to

$$\begin{aligned}\big\|\varphi\sum_{k=0}^{\infty}(\psi P)^k x - x\big\| &\leq \varphi\sum_{k=0}^{\infty}\|(\psi P)^k x\| + \|x\| \\ &\leq (\varphi\sum_{k=0}^{\infty}\psi^k c + 1)\|x\| \\ &= (c+1)\,\|x\|. \end{aligned}\tag{7}$$

The substitution of (7) into (6) concludes our proof.

$$\varepsilon_t < \psi^t c\,(c+1)\,\|x\| \leq e^{-\varphi t}(c^2 + c)\,\|x\|$$

□

A direct consequence of Theorem 1 is that the necessary number of rounds until the algorithm converges ε-close to y is given by

$$t \geq \varphi^{-1}\log\frac{(c^2+c)\|x\|}{\varepsilon}. \tag{8}$$

4.2 Time-varying information

During the lifetime of some practical applications, the information process x may change over time. For example, a sensor network monitoring the presence of some type of animals will change its measurements when the animal moves. We proceed to examine the behavior of such time-varying information processes.

LEMMA 1. *Let an information potential be at steady state y, while its underlying information process x changes to $\hat{x}$, with $\|\hat{x} - x\| \leq \delta_x$. After $\tau > 0$ steps, the ℓ_2-distance of the potential to the new stable state $\hat{y}$ is bounded by $\varepsilon_\tau \leq e^{-\varphi\tau} c\,\delta_x$.*

PROOF. Without loss of generality, assuming that the change from x to $\hat{x}$ occurs at time t, the ℓ_2-distance to $\hat{y}$ at round τ is bounded by

$$\begin{aligned}\varepsilon_\tau &= \|\,\hat{y} - y_{t+\tau}\| \\ &= \|\,\varphi\sum_{k=0}^{\infty}(\psi P)^k\hat{x} - \varphi\sum_{k=0}^{\tau-1}(\psi P)^k\hat{x} - (\psi P)^\tau y\| \\ &= \varphi\,\|\sum_{k=\tau}^{\infty}(\psi P)^k\hat{x} - \sum_{k=\tau}^{\infty}(\psi P)^k x\| \\ &\leq \varphi\sum_{k=\tau}^{\infty}\big\|(\psi P)^k(\hat{x} - x)\big\| \leq e^{-\varphi\tau} c\,\delta_x. \end{aligned}$$

□

A direct consequence of Lemma 1 is that the minimum τ for which the algorithm manages to converge ε close to the new stable state $\hat{y}$ is given by

$$\tau \geq \varphi^{-1}\log\frac{c\,\delta_x}{\varepsilon}. \tag{9}$$

4.3 Analysis Insights

Theorem 1 and Lemma 1 provide us with four important insights: (i) *The convergence error decreases exponentially.* The inhibiting factor determines the rate of convergence. Smaller values of φ aggregate the values over an exponentially larger subgraphs and as a consequence exhibit slower convergence. (ii) *For $\varphi > 0$, the proposed algorithm converges faster than average consensus* [7]. The common ratio of the geometric series which upper bounds the convergence error decreases from μ_2 (average consensus) to $(1-\varphi)\mu_2$ (proposed algorithm). For $\varphi = 0$, the two mechanisms are exactly the same. (iii) *Information dynamics proportionally increase the ℓ_2-error.* This effect should be taken into account when choosing the value of the inhibiting factor. (iv) *Convergence is faster in dense, degree regular graphs.* Constant c captures the influence of the network topology to the convergence. Through c we derive that convergence is faster for dense graphs (n/m is the inverse of the graph density), as well as for graphs with small node density variations (quantified by ξ).

5. CONVERGENCE IN DYNAMIC GRAPHS

In this section, we study the algorithmic behavior in the context of graphs that change over time. We model graph dynamics as a sequence of edge and node operations: *Edge operations* describe the addition or deletion of edges between pairs of nodes. *Node operations* model nodes joining or leaving the graph. Through edge and node operations we capture a wide range of network dynamics, such as node and link failures, as well as the dynamics of open networks where the network is subject to mobility and churn. Due to the complexity of the problem, we assume time-invariant information.

We will first derive a bound on the convergence error given any change in the graph. The bound, which is stated in Theorem 2, is general enough to hold for any possible graph dynamics. On the down side, the bound depends on the specifics of the graph dynamics $\delta_{\mathcal{G}}$ in question. We gain further insight by characterizing $\delta_{\mathcal{G}}$ for edge and node operations in Lemmas 2 and 3, respectively.

Theorem 2. *For any dynamic graph $\mathcal{G}_t$ which varies with steps $\tau > 0$ and is bounded, $\|P_{t+\tau} - P_t\| \le \delta_{\mathcal{G}}$, the ℓ_2-distance of the potential to the stable state, just before any consecutive variation of $\mathcal{G}_t$, is bounded by*

$$\varepsilon_{t+\tau} \le e^{-\varphi(t+1)} c_{t+\tau}\Big(\delta_{\mathcal{G}}\,\frac{1-\varphi}{\varphi}c_{t+\tau}\,c_t + \frac{\varphi}{1-\varphi}\Big)\|x\|,$$

where c_t quantifies the connectivity properties of $\mathcal{G}_t$.

Proof. Consider that the algorithm has already converged to a stable state y on a graph $\mathcal{G}$. An adversary then changes the graph to $\hat{\mathcal{G}}$. In the following we annotate symbols that relate to the new graph $\hat{G}$ with a hat; as such $\hat{P}$ is the new transition matrix, $\hat{n}$ is the number of nodes in $\hat{\mathcal{V}}$, and so on. As in previous proofs, we capture convergence error through the ℓ_2-distance between the state after t iterations and the new stable state $\hat{y}$. By substituting the analytic expression of y into Formula (4) we get

$$\varepsilon_t = \|\hat{y} - y_t\| = \Big\|\varphi\,(\psi\hat{P})^t\Big(\sum_{k=1}^{\infty}\psi^k\big(\hat{P}^k - P^k\big)\,x - x\Big)\Big\|.$$

Using Inequality (5) and after some algebraic manipulation we have that

$$\begin{aligned}
\varepsilon_t &\le \psi^t\varphi\,\hat{c}\,\Big(\Big\|\sum_{k=1}^{\infty}(\psi\hat{P})^k x - \sum_{k=1}^{\infty}(\psi P)^k x\Big\| + \|x\|\Big)\\
&= \psi^t\varphi\,\hat{c}\,\Big(\Big\|(\psi\hat{P} - \psi P)\sum_{k=1}^{\infty}(\psi\hat{P})^k\sum_{l=k+1}^{\infty}(\psi P)^{l-k-1}x\Big\| + \|x\|\Big)\\
&\le \psi^{t+1}\varphi\,\hat{c}\,\Big(\delta_{\mathcal{G}}\sum_{k=1}^{\infty}(\psi^k\hat{c})\sum_{l=0}^{\infty}(\psi^l c) + \psi^{-1}\Big)\,\|x\|\\
&\le e^{-\varphi(t+1)}\hat{c}\,\Big(\delta_{\mathcal{G}}\,\frac{1-\varphi}{\varphi}\hat{c}\,c + \frac{\varphi}{1-\varphi}\Big)\|x\|,
\end{aligned}$$

which concludes our proof. □

Solving for t we find that the least number of rounds until $\varepsilon_t \le \varepsilon$ is at most

$$t \ge \varphi^{-1}\log\frac{\hat{c}\,\Big(\delta_{\mathcal{G}}\frac{1-\varphi}{\varphi}\hat{c}\,c + \varphi\Big)\|x\|}{\varepsilon} - 1. \tag{10}$$

Theorem 2 draws a relation between the influence of graph dynamics in convergence time and the value of φ. Depending on whether $(1-\varphi)/\varphi < \varphi/(1-\varphi)$ and thus $\varphi > 1/2$, the theorem distinguishes two regions of convergence: (*i*) $\varphi > 1/2$ and as φ grows larger, the graph dynamics ($\delta_{\mathcal{G}}$) become irrelevant. The convergence becomes independent of the relation between G and $\hat{G}$; convergence depends solely on the new graph $\hat{G}$. (*ii*) $\varphi < 1/2$ and as φ gets closer to 0, the rate of convergence becomes slow enough such that the graph dynamics do matter. In this region, convergence depends heavily on the nature of edge operations performed.

We proceed to examine how edge and node operations influence convergence by computing δ_p for each case.

5.1 Edge operations

An adversary adds or deletes edges $\mathcal{E}_+$ and $\mathcal{E}_-$, respectively to the graph, such that either $\hat{\mathcal{E}} = \mathcal{E} \cup \mathcal{E}_+$ (edge addition), or $\hat{\mathcal{E}} = \mathcal{E} \setminus \mathcal{E}_-$ (edge deletion). We also place the constrain that the symmetric edge set difference, $\hat{\mathcal{E}} \ominus \mathcal{E} = \mathcal{E}_+ \cup \mathcal{E}_-$, contains at most one edge (i,j) for each node i in $\mathcal{V}$. The constraint demands that the adversary performs at most one edge operation in the vicinity of each node. Multiple edge alterations on the same node are modeled as consecutive operations. As expected, self-loops cannot be deleted, that is $d_i \ge 1$ for all i in $\mathcal{V}$. We prove the following bound.

Lemma 2. *For any graphs $\hat{\mathcal{G}}$, $\mathcal{G}$, with identical node sets, $\mathcal{V} = \hat{\mathcal{V}}$, and edge sets $\hat{\mathcal{E}} \ne \mathcal{E}$ that have at most one edge difference in the vicinity of each node, $|(i,j) \in \hat{\mathcal{E}} \ominus \mathcal{E}| \le 1$ for all $i \in \mathcal{V}$,*

$$\delta_{\mathcal{G}} = \frac{1}{d_{min}} + \frac{\sigma(A)+1}{\min\limits_{i\in\mathcal{V}_\pm}\{\hat{d}_i d_i\}},$$

with $\mathcal{V}_\pm = \{i \in \mathcal{V} : |(i,j) \in \hat{\mathcal{E}} \ominus \mathcal{E}| = 1$ for some $j \in \mathcal{V}\}$ the set of nodes that have different neighbors in $\hat{\mathcal{G}}$ and $\mathcal{G}$, $\hat{d}_i, d_i$ their respective densities in each graph, and $\sigma(A)$ the largest singular value of A.

Proof. The adjacency matrix of $\hat{\mathcal{G}}$ can be written as

$$\hat{A} = A + \sum_{(i,j)\in\mathcal{E}_+}\big(E_{ij} + E_{ji}\big) - \sum_{(i,j)\in\mathcal{E}_-}\big(E_{ij} + E_{ji}\big),$$

where matrix E_{ij} has only element (i,j) equal to one and the rest zero. The inverse density matrix of $\hat{G}$ can in turn be written as

$$\hat{D}^{-1} = D^{-1} + \sum_{(i,j)\in\mathcal{E}_+}\big(a_iE_{ii} + a_jE_{jj}\big) - \sum_{(i,j)\in\mathcal{E}_-}\big(b_iE_{ii} + b_jE_{jj}\big),$$

where $a_i = \frac{1}{d_i(d_i+1)}$ and $b_i = \frac{1}{d_i(d_i-1)}$. Matrix E_{ij} has the useful property of $\|E_{ij}\| = 1$ for all $i,j \in \mathcal{V}$. Due to $\hat{\mathcal{E}} \ominus \mathcal{E}$ containing at most one edge (i,j) for each node i in $\mathcal{V}$ and because $\hat{A} - A$ is a symmetric projection matrix,

$$\|\hat{A} - A\| = 1. \tag{11}$$

As in Lemma 2, $\mathcal{V}_\pm = \{i \in \mathcal{V} : |(i,j) \in \hat{\mathcal{E}} \ominus \mathcal{E}| = 1$ for some $j \in \mathcal{V}\}$ is the set of nodes that were affected by an edge operation. Matrix $\hat{D}^{-1} - D^{-1}$ is diagonal and its norm is the maximum diagonal element in absolute value.

$$\|\hat{D}^{-1} - D^{-1}\| = \min_{i\in\mathcal{V}_\pm}\{\hat{d}_i d_i\}^{-1} \tag{12}$$

Observe that $\hat{d}_i d_i \ge d_{min}(d_{min} - 1) \ge 2$, for all $i \in \mathcal{V}_\pm$. The first equality is satisfied when one of the endpoints of a deleted edge was connected to the node with the minimum density, while the second equality iff $d_{min} = 2$. We re-write the ℓ_2-distance between the two random walk matrices as

$$\begin{aligned}
\|\hat{P} - P\| &= \|\hat{D}^{-1}\hat{A} - D^{-1}A\|\\
&\le \Big(\|D^{-1}\| + \|\hat{D}^{-1} - D^{-1}\|\Big)\|\hat{A} - A\| + \ldots\\
&\quad + \|\hat{D}^{-1} - D^{-1}\|\,\|A\|.
\end{aligned}$$

The required bound is derived if we substitute Formulas (11) and (12) into $\|\hat{P} - P\|$.

$$\delta_{\mathcal{G}} = \|D^{-1}\| + \frac{\|A\|+1}{\min\limits_{i\in\mathcal{V}_\pm}\{\hat{d}_i d_i\}} = \frac{1}{d_{min}} + \frac{\sigma(A)+1}{\min\limits_{i\in\mathcal{V}_\pm}\{\hat{d}_i d_i\}}$$

Above, $\|A\| = \sigma(A)$ is the largest singular value of the adjacency matrix.

□

Let us reflect on the influence of edge operations on the convergence: (i) *Edge operations affect well connected graphs to a greater extend.* That is due to $\sigma(A) \leq n$, with the equality satisfied for fully connected networks. While in sparse networks the effects of dynamics tend to be isolated, in dense networks there is a higher likelihood that any single change affects more nodes. (ii) *Nevertheless, networks with few links mitigate the effects of network dynamics slower.* Our method compensates for edges added or deleted in dense areas of the network (i. e., $d_i >> d_{min}$ for $i \in \mathcal{V}_{\pm}$) faster than in areas where the network is sparsely connected. The effect is understood by the property of information to diffuse faster in dense than in sparse areas. (iii) *Convergence also depends on the density irregularity.* The algorithm exhibits faster convergence when running on regular networks, where $\xi = \hat{\xi} = 1$. (iv) *Last, even though edge operations affect multiple edges, the error bound is independent of the exact number of affected edges.* The error depends instead solely on the edge that connects to the least dense node.

5.2 Node operations

Node operations are operations on the set $\mathcal{V}$ of graph nodes. In node additions (deletions), an adversary adds (deletes) nodes $\mathcal{V}_+$ ($\mathcal{V}_-$), such that $\hat{\mathcal{V}} = \mathcal{V} \cup \mathcal{V}_+$ ($\hat{\mathcal{V}} = \mathcal{V} \backslash \mathcal{V}_-$). We provide an upper bound of $\delta_{\mathcal{G}}$ for the simultaneous addition and deletion of nodes.

Lemma 3. *Let $\hat{\mathcal{G}}$ be the graph that results from the addition and deletion of sets $\mathcal{V}_+$ and $\mathcal{V}_-$ from $\mathcal{G}$, respectively. Given that (i) at most one edge joins each node in the graph intersection to the symmetric graph difference, $|(i,j) \in \hat{\mathcal{E}} \ominus \mathcal{E}| \leq 1$ for all $i \in \hat{\mathcal{V}} \cap \mathcal{V}$, and (ii) that no two added nodes are adjacent, $(i,j) \notin \hat{\mathcal{G}}$ for all $i,j \in \mathcal{V}_+$, then*

$$\delta_{\mathcal{G}} = \sum_{i \in \hat{\mathcal{V}} \ominus \mathcal{V}} \frac{\sqrt{d_i}}{d_{min}} + \max_{i \in \hat{\mathcal{V}} \ominus \mathcal{V}} \left\{ \frac{d_i - 1}{d_i} \right\} (\sigma(A) + \sum_{i \in \hat{\mathcal{V}} \ominus \mathcal{V}} \sqrt{d_i}),$$

where $\sigma(A)$ is the largest singular value of A.

Proof. In the computation of $\delta_{\mathcal{G}}$ we cannot reuse our previous results. Deleting a node is equivalent to deleting all edges between the deleted node and its neighbors. The number of deleted edges can therefore be larger than one, which violates our constraint that at most one edge is deleted from each node. To avoid a loose bound, we do not model node deletion as a sequence of edge deletions. Instead, we redefine the constraint for node operations to allow multiple edge deletions if the edges are adjacent to a deleted node. More formally, at most one edge should join any node in the graph intersection to the symmetric graph difference. This constraint definition applies to both additions and deletions. An added node can simultaneously connect to multiple nodes, given that the nodes it connects to are not affected by other operations. For simplicity, we also assume that added nodes cannot be adjacent to each other. We now express $\hat{A}$ and $\hat{D}^{-1}$ as a function of A and D^{-1}, respectively.

$$\hat{A} = A + \sum_{i \in \mathcal{V}_+} \sum_{j \sim i} (E_{ij} + E_{ji}) - \sum_{i \in \mathcal{V}_-} \sum_{j \sim i} (E_{ij} + E_{ji})$$

and

$$\hat{D}^{-1} = D^{-1} + \sum_{i \in \mathcal{V}_+} \frac{d_i - 1}{d_i} E_{ii} + \sum_{i \in \mathcal{V}_+} \sum_{j \sim i} a_j E_{jj} + \dots + \sum_{i \in \mathcal{V}_-} \frac{1 - d_i}{d_i} E_{ii} - \sum_{i \in \mathcal{V}_-} \sum_{j \sim i} b_j E_{jj}.$$

Naturally, a deleted node can not be simultaneously added or vice versa. Also, added nodes cannot be adjacent to deleted nodes. Last, the constraint guarantees that at most one edge changes for any of the nodes that are not added or deleted. Therefore, at most one term is added or subtracted to each diagonal element. Given that $\hat{D}^{-1} - D^{-1}$ is diagonal and because $|\frac{d_i - 1}{d_i}| = |\frac{1 - d_i}{d_i}| \geq \max\{a_j, b_j\}$ for all $i \in \hat{\mathcal{V}} \ominus \mathcal{V}$ and $j \sim i$,

$$\left\| \hat{D}^{-1} - D^{-1} \right\| = \max_{i \in \hat{\mathcal{V}} \ominus \mathcal{V}} \left\{ \frac{d_i - 1}{d_i} \right\}. \tag{13}$$

For similar reasons, matrix $\hat{A} - A$ is symmetric with non-zero elements equal to one.

$$\left\| \hat{A} - A \right\| \leq \sum_{i \in \hat{\mathcal{V}} \ominus \mathcal{V}} \sqrt{d_i} \tag{14}$$

Substituting Inequalities (13) and (14) into the definition of $\delta_{\mathcal{G}}$ and factoring the result we get the desired bound. □

The above bound provides three insights on how node operations affect convergence: (i) *Added and deleted nodes $i \in \hat{\mathcal{V}} \ominus \mathcal{V}$ introduce a convergence error that is proportional to the square root of the number of nodes they connect to.* On the left, $\sqrt{d_i}$ is normalized to the minimum network density. On the right, $\sqrt{d_i}$ is weighted by the maximum value of $(d_i - 1)/d_i$ which tends to one as $d_i \to \infty$. (ii) *It is the most connected node involved in a node operation which effects convergence the most.* Since the right term incurs the most significant error, we deduce that the convergence error after a node operation depends to a large extend on the density of the most connected node of the operation; *whether the node was deleted or added is of no significance.* (iii) *It is density irregularity that affects convergence the most, not absolute node density.* This seems to be an inherent property of information potentials as it consistently arises in all the convergence bounds we have derived so far (cf. Sections 4 and 5). In contrast to regular spaces such as continuous domains and regular graphs, the information diffusion in irregularly shaped spaces incurs a price.

6. SPECTRAL PROPERTIES

Thus far we have analyzed the impact of various network dynamics on the convergence of information potentials. In this section, we show that these potentials are not solely determined by the network's properties, but they can be shaped into different landscapes using the inhibiting factor φ. First, we employ results from spectral graph theory to showcase the connection between information potentials and the eigenfunctions of the network's Laplacian, and then, we show how the inhibiting factor can "filter" these eigenfunctions to shape the information landscape.

6.1 The spectral form

Let us start with some basic definitions. A scalar λ is an *eigenvalue* of a matrix B if there exists a non-zero vector u

such that $Bu = \lambda u$. Vector u is called a (right) *eigenfunction* and the pair (λ, u) is called an *eigenpair*. The collection of all eigenpairs is often referred to as the *spectrum* of B, where the eigenvalues are ordered with increasing magnitude. Our main result ties information potentials to the spectrum of the normalized Laplacian, which is described in Section 2.

THEOREM 3. (SPECTRAL FORM) *Let (λ, u) be eigenpairs of the normalized Laplacian $\mathcal{L}$ of a graph $\mathcal{G}$. For any information process x, the information potential y is*

$$y = D^{-1/2} \sum_{k=1}^{n} w_k \langle D^{1/2}\, x, u_k \rangle\, u_k, \tag{15}$$

where $w_k = \frac{\varphi}{1-\psi(1-\lambda_k)}$.

PROOF. We exploit the spectral relations between the transition matrix P and the Laplacian $\mathcal{L}$. The reader can refer to the text by Biyikoglu et al. [3] for more details on the topic. Let (μ, v) be eigenpairs of the transition matrix P. In connected graphs, P has a unique largest eigenvalue $\mu_1 = 1$ and all other eigenvalues μ_k, with $k \in \mathbb{N}$ and $k \leq n$, have smaller magnitude. Since $\psi < 1$, matrix ψP has eigenvalues that are strictly smaller than one and we re-write Formula (3) as

$$y = \varphi\, (I - \psi P)^{-1} x. \tag{16}$$

As shown next, $(I - \psi P)$ has the same eigenfunctions as P and its eigenvalues are equal to $1 - \psi\mu$,

$$Pv = \mu v$$
$$(I - \psi P)v = (1 - \psi\mu)v.$$

It is well known that invertible matrices have the same eigenfunctions as their inverse and eigenvalues that are the reciprocal of the eigenvalues of their inverse. Formula (16) can therefore be re-written through the spectral expansion of the inverse of $1 - \psi P$ as

$$y = \sum_{k=1}^{n} \frac{\varphi}{1 - \psi\mu_k} v_k v_k^{\top} x, \tag{17}$$

where $v_k^{\top}$ is the k-th left eigenfunction of P. From [3], we know that $v_k = D^{-1/2} u_k$, $v_k^{\top} = u_k^{\top} D^{1/2}$ and $1 - \lambda_k = \mu_k$. Substituting these equalities into Formula (17) concludes our proof. □

6.2 Landscape architecture

In this section we interpret the results from Theorem 3 and describe how the inhibiting factor controls the shape of the information landscape. Before describing the theoretical framework behind our method, we start with an intuitive explanation.

Intuitive explanation. In simple terms, Theorem 3 states that an information potential is a weighted sum of the eigenfunctions of the normalized Laplacian. Setting aside their discrete and irregular shape, Laplacian eigenfunctions are analogous to waves. They possess peaks and valleys, and are characterized by a spatial frequency which is determined by their rank (i. e., higher rank means more peaks). The projection of the information x on an eigenfunction captures the components of x which are characterized by that specific spatial variation; the higher the eigenfunction's rank, the higher the spatial variation of x that it captures. The weight $0 \leq w_k \leq 1$, which depends on the inhibiting factor, decreases the significance of eigenfunctions with high spatial variation (w_1 is always equal to one and the other weights decrease monotonically). Hence, the inhibiting factor can be seen as a low-pass filter that attenuates phenomena of high spatial frequencies. By fine-tuning this parameter, our method smoothens the landscape of an information potential and, as a consequence, reduces the number of local extrema.

Formal explanation. Consider the n-dimensional bi-orthonormal eigenspace formed by the eigenfunctions u_k of $\mathcal{L}$. Theorem 3 rewrites the potential as a weighted projection of the density-normalized information on each of u_k. Weights $0 \leq w_k \leq 1$ decrease the significance of projections on non-principal[3] eigenfunctions.

To capture the spatial variation of eigenfunctions, we rely on the concept of nodal domains from spectral graph theory [3, 7]. Recall that an eigenfunction assigns a positive or negative value to each node in a graph. Nodal domains (also called *sign graphs*) induce a partition of the graph into maximal induced subgraphs on which a function does not change its sign. In other words, consider a graph where the real values of nodes are mapped to their positive or negative signs, then group connected nodes with the same sign into subgraphs. The number of subgraphs represent the number of nodal domains. Intuitively, the larger the number of nodal domains, the higher the variance of a function. Based on whether the subgraph also includes nodes with zero value, nodal domains are further characterized as weak or strong.

The *discrete nodal domain theorem* [8] gives an upper bound on the number of nodal domains of the eigenfunctions of a generalized Laplacian. Specifically for the normalized Laplacian of a connected graph, the theorem states that any eigenfunction u_k corresponding to the k-th eigenvalue λ_k with multiplicity r has at most k weak nodal domains and at most $k + r - 1$ strong nodal domains. Eigenfunctions with higher eigenvalues are likely to have more nodal domains. Intuitively, attenuating the projections of an information process into eigenfunctions with several nodal domains should lead to less local extrema. Below we proof Lemma 4, which provides an upper bound on the number of eigenfunction extrema. The lemma interprets positive and negative nodal domains as the peaks and valleys of eigenfunctions. It then bounds the number of extrema by showing that a positive (negative) nodal domain has at most one maximum (minimum).

LEMMA 4. *Any eigenfunction u_k corresponding to the k-th eigenvalue λ_k of the normalized Laplacian of a connected graph has at most k extrema.*

PROOF. The proof proceeds by method of contradiction. An eigenfunction of a graph Laplacian cannot have a non-negative local minimum or a non-positive local maximum [11, 12]. Without loss of generality, assume a negative nodal domain and suppose there are two local minima residing in the domain. This necessitates the existence of a negative local maximum between the two minima, which contradicts the first observation. Therefore, a single minimum (maximum) must exist at every negative (positive) nodal domain. Since the number of weak nodal domains of u_k (zeros are irrelevant) is bounded by k by the discrete nodal domain theorem, the number of extrema of u_k is at most k. □

[3] Principal is the eigenfunction that corresponds to λ_1.

We use Lemma 4 to draw some useful observations about the shape of information potential landscapes. We saw that a potential is a composition of information projections on n eigenfunctions (Theorem 3). The rank k of an eigenfunction determines the maximum number of extrema it contains (Lemma 4). A such, projections of the information on high order eigenfunctions describe information components of progressively higher spatial frequency. As the inhibiting factor decreases so do the weights of non-principal eigenfunctions (i.e., rank > 1). By decreasing the significance of projections on non-principal eigenfunctions, the inhibiting factor removes phenomena of high spatial variation and reveals spatial patterns of lower frequency.

Potentials reshape an information landscape in more subtle ways than simply eliminating its extrema. As shown in Section 3.2, the potential of a node depends on the values of all the nodes in the node's affinity space, as well as on the underlying connectivity. Therefore, it is possible that a node that is an extremum of y, but not of x. For example, an already unimodal x will remain unimodal. The position of the extremum however will not necessarily be the same. The extremum of x is the node with the maximum value, which is not always equivalent to the node with the highest information y in its surroundings. This phenomenon is captured in our evaluation.

6.3 The unimodality property

An important outcome of our spectral analysis is that, for any graph $\mathcal{G}$ and information process x, a critical value φ exists for which the potential is *unimodal*. The unimodality property proves useful for gradient search. In unimodal search spaces, greedy search cannot get stuck in local optima. By achieving unimodality in networks of multiple information sources (i. e., each node has its own information), this part of our study in essence extends the work by Lin et al. [15].

Formally, a function defined on a graph is unimodal if it has a single extremum and, for each node, at least one path to the extremum exists on which the values of the function are monotonic. Depending on the monotonicity, the unimodality is further characterized as *weak* or *strong*. As φ becomes smaller, non-principal weights decrease the influence of information of high spatial variation and thus eliminate local extrema. Due to this trend, a value of φ exists below which a potential possesses a single, global maximum (or minimum). Since Lemma 4 does not guarantee that two neighboring nodes in the same domain do not have the same value, the unimodality of potentials is in the general case weak.

In practice, unimodal potentials are not always desirable; one has to pay in terms of convergence time for the increased search scope. In very large networks, the critical value can be too small to allow for any adaptivity and landscapes with multiple extrema incur lower computation cost. Despite its sub-optimality, greedy gradient search in multimodal potentials yields more spatially relevant results as the extrema exhibit a higher spatial correlation to node location. Nevertheless, unimodal potentials are particularly useful in specific tasks, such as robot coordination. Our previous paper demonstrated how unimodal potentials solve the problem of multi-agent rendezvous [16].

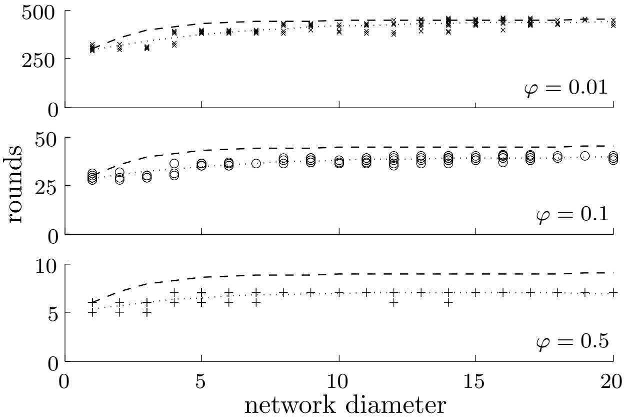

Figure 2: Number of rounds until the convergence error becomes smaller than 0.05 versus network diameter. Simulation results are depicted with markers connected by dotted lines and analytical results with dashed lines.

7. EVALUATION

Our analysis provides several insights. Due to page constraints, we can not present evaluations for all of them. Hence, we focus on the four which are the most important. We perform a controlled evaluation based on simulations (i) to evaluate the convergence of our algorithm under different settings and to validate the analytical results of Section 4.1, and (ii) to challenge our theoretical assumptions on synchronous execution and no data losses (Section 3.2). We perform a short testbed evaluation (iii) to show that the simple distributed algorithm can be implemented in resource-constrained devices, (iv) to exemplify that it is resilient to normal operating conditions such as link variability and node failures, and (v) to showcase its capability to shape the information landscapes into a unimodal potential (Section 6.3). Our evaluation is only meant to demonstrate the feasibility of our approach. A robust protocol implementation would need to consider the specific requirements of the application at hand.

7.1 Model validation

We used the COOJA simulator, a widespread tool for wireless network simulation. We chose the information values arbitrarily, by setting $x(i) = 0$ for some random $i \in \mathcal{V}$ and $x(\mathcal{V} \setminus i) = 1$, otherwise. The nodes were deployed uniformly at random and the unit-disk model was used to establish connectivity. To capture a wide range of connectivity properties, we tested four network sizes: 10, 50, 100 and 150 nodes, with four different transmission ranges. We evaluated 14 instances for each $\langle$*size*, *range*$\rangle$ tuple, resulting in 224 different networks. Overall, the networks had average degrees between 3.4 and 17.24, diameters between 1 and 20, and clustering coefficients between 0.27 and 1. We evaluated three representative values of the inhibiting factor, i. e., $\varphi = 0.01$, 0.1, and 0.5, respectively.

Asynchrony. Our first set of experiments show that synchrony is not a critical assumption; the convergence bounds in Section 4.1 still hold for *asynchronous* simulations. Figure 2 summarizes our experiments. The markers represent

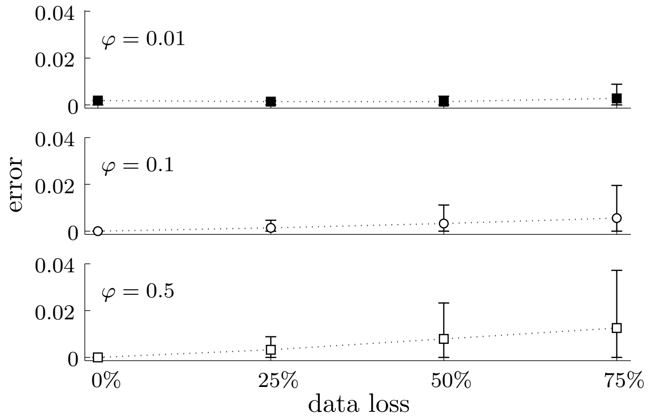

Figure 3: Error after convergence versus percentage of unknown neighboring values to the total neighbors at the end of each round. The algorithmic robustness increases for small φ.

simulation results and the dashed lines represent the convergence bounds from Section 4.1. Each marker represents a $\langle$*topology*, $\varphi\rangle$ tuple, and for each tuple, we record the round when $\|y - y_t\| \leq 0.05$, i. e., when it converges. The iterative calculation of y_t used our distributed algorithm, while the ground-truth y was calculated assuming an oracle's view of the network. To test our method under the worst possible circumstances, we intentionally bootstrap nodes to a value that is far from the stable state, $y_0 = 0 \ll x, y$. It is important to highlight the trade-offs of the inhibiting factor φ. Our method is better suited for large-scale networks. The number of rounds to reach convergence does not increase significantly with the network's diameter (for any value of φ). On the other hand, small values of φ are not suitable for small networks. Observe that, even in fully connected networks (1-hop diameter), our method incurs an overhead that is on the order of $1/\varphi$, which translates to more than 250 rounds for $\varphi = 0.01$. Setting the inhibiting factor to very small values only makes sense for large networks or in cases of high information variation. For small networks, high values of φ or simpler aggregation schemes may be preferable.

Robustness to data loss. Observe that some markers in Figure 2 are over the bounds. The observed difference suggests that message loss, i. e., our second assumption, has an effect on the algorithmic operation. Message losses require more rounds than expected to reach convergence. Figure 3 distills the results of a set of experiments which characterize the effect of imperfect knowledge. In the experiment, nodes disregarded a specific, randomly selected, percentage of their received values $\mathcal{S}_i$ just before recomputing their potential (line 11 of Algorithm 1). Before measuring the normalized potential error of a given topology, we waited for a sufficient number of rounds until the algorithm had converged. For each $\langle\varphi$, *data loss*$\rangle$ tuple, we summarize the errors across time from all 224 topologies, by the corresponding median and 68.2% confidence interval. Even under severe loss, the algorithm exhibits an error that is smaller than 0.04. The algorithm is robust because it is not based on algorithmic primitives that are sensitive to message loss, e. g., mass conservation, or to partial knowledge, e. g., counting. Instead, it employs averaging as a statistical measure that, in non-skewed distributions, approximates well the central tendency, even with small sample sizes. Comparing the three subfigures we also notice that, for large percentages of data loss ($\varphi = 0.01$), the topmost subfigure reports a median error that is approximately two times smaller that the one in the bottommost subfigure ($\varphi = 0.5$). This phenomenon stems from the tendency of small φ to limit the magnitude of change between consecutive computation rounds. As such, the sensitivity of the algorithm to the high variation effects that accompany data loss decreases with φ.

7.2 Empirical results

This section evaluates our method in a wireless testbed of 105 nodes. The testbed is deployed in the ceiling of our floor in TU Delft. The devices are equipped with a MSP430 micro-controller and a CC1101 radio chip, with the transmission power set to −30 dBm. The algorithm was implemented on Contiki OS. For medium access control, we used NullMAC, a simple random-access MAC protocol with carrier-sense capabilities that is part of the standard Contiki OS. Each computation round lasted for 0.5 seconds, over which each node exchanged an average of 3.5 messages. An instance of the connectivity graph is shown in Figure 4(a). As also revealed by previous investigations [23], the connectivity was highly variable over time due to the well-known volatility of low-power wireless links. Figure 4(b) depicts the nodes' degree (density). The darker the Voronoi cell color (the larger the disc), the higher the density. The maxima are shown in green cells and the red arrows represent greedy searches that reach their respective local maxima. Due to the non-uniform coverage of radio transceivers, nearby nodes may not have a link, while far away nodes may – this is typical in testbeds and real-world deployments. This implies that nodes in adjacent Voronoi cells are not necessarily neighbors (in terms of distance) and the reader should refrain from interpreting Voronoi diagrams as continuous fields.

To demonstrate that our method eliminates phenomena of high variation, such as noise and false extrema, (i) we corrupted the information process by adding noise to each node's degree. The noise was uniformly distributed between [0, d_{min}]. Also, (ii) we introduced a false maximum at a randomly chosen "faulty" node – colored orange. Figures 4(c–d) show the computed information potentials for $\varphi = 0.50$ and 0.01, respectively. Each experiment was run for ten minutes, during which we experienced a high variability of links and some (normal) node failures between the evaluations of $\varphi = 0.50$ and 0.01. This test highlights the resilience of our method to real network phenomena. Observe how our method progressively eliminates local maxima, thus making the potential unimodal. Figure 4(c) eliminates one of the four local maxima and Figure 4(d) has a single extrema. Two of the three search queries get stuck in local maxima in Figures 4(b–c), while in Figure 4(d) all queries reach the global maximum. Observe also that the position of the global maximum is not the same across all figures, which confirms our analysis in Section 6.2. Our method uncovers the node with the largest information in its vicinity, which is not necessarily the one with the largest information (discovering paths towards a maximum value can be easily solved using a max-consensus algorithm and a distance gradient).

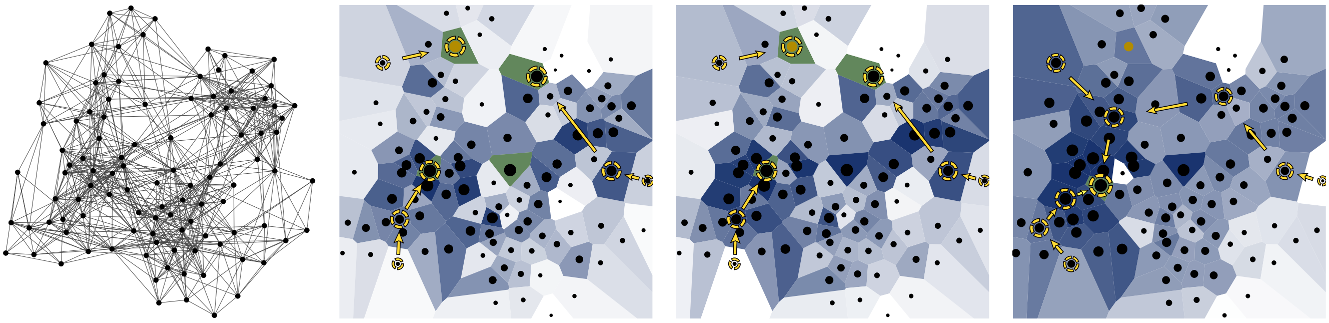
(a) testbed topology (b) information process (c) $\varphi = 0.50$, 3 maxima (d) $\varphi = 0.01$, 1 maximum

Figure 4: Information potentials (c–d) and information process (b) on a testbed of 105 nodes (a). Function maxima reside in green Voronoi cells. The orange node is an artificially injected maximum. As the aggregation scope widens, our method progressively eliminates local extrema.

8. RELATED WORK

The computation of potential functions is inspired by the natural process of *chemotaxis*, in which cells respond to the concentrations of chemicals in their environment [2], and has been exploited to achieve the coordination of swarms [17]. In the following, we group related work into three subcategories. First, We discuss the connections to vicinity-based aggregation methods. Second, we focus on unimodality and thus on information discovery. Last, we classify and relate our work with respect to the family of consensus algorithms.

Vicinity-based aggregation. In sensor networks, potential functions are information specific and vicinity based; nodes consider surrounding information with a significance that decays with distance [10, 13, 18]. Gao et al. [10], use quad-trees to store data, such that each node is aware of the data in its vicinity. In their seminal paper [13], Kempe et al. propose *spatial gossiping* algorithms, in which any two nodes gossip with probabilities that decrease polynomially to their distance. Sarkar et al. [18] extend spatial gossiping to compute multi-resolution representations of information. Their algorithm computes information aggregates over exponentially enlarging areas centered at each node. All of these approaches however use physical distance to define information affinity. Our method defines affinity using the random movement of particles on the network. As a result, it is *sensitive to the network topology and independent of any knowledge of physical location or distance.*

Information discovery. A number of recent works have also employed information potentials as mediums of discovery [10, 15, 19, 21]. Skraba et al. [21] use potential gradients to sweep a sensor network. Lin et al. [15] construct smooth harmonic gradients towards node subsets, called sources, such that local forwarding guarantees their discovery. Their method achieves the absence of local extrema by keeping the information of sources constant, fixing the values of boundary nodes to zero, and averaging in between. The number of extrema however is equal to the number of sources. In comparison, our method guarantees unimodality irrespectively of the number of sources. It does not require knowledge of the network boundary. It is also more flexible as it aggregates values that stem from a real-valued monitored process, as compared to the boolean distinction between sources and non-sources. Sarkar et al. [19] design query mechanisms for general information fields, which support the use of more advanced operations, such as iso-contour queries and value restricted routing. Their approach is complementary to ours, as it does not concern the landscape formation but advanced methods of information discovery.

Consensus algorithms. From an algorithmic point of view, this paper proposes a variant of the well known consensus algorithms [1, 9]. In a strict sense however, our algorithm does not solve the consensus problem; the stable state of a consensus algorithm is $\alpha\mathbb{1}$, where $\alpha \in \mathbb{R}$ is usually the average or the maximum of the information. In our case, each node converges to a distinct value; the collection of values form landscapes that support information discovery. However, Khan et al. recently proposed a wider family of consensus algorithms, referred to as *higher dimensional consensus algorithms*, under which our method can be classified [14]. We also have to note that, unlike most consensus and gossip algorithms that are used in sensor networks [1, 4], our method is not randomized and does not sacrifice accuracy in the presence of communication loss. Throughout the computation, information works as an anchor that steers the network towards the correct stable state. As witnessed by our evaluation, message loss mainly increases the variance of the stable state and has little effect on the mean error.

Due to its connection to the graph Laplacian, our method is also related to data clustering algorithms, such as mean shift clustering [5] and spectral clustering [22]. Choa et al. [6] independently proposed a similar approach to compute the modes of a graph. Even though their paper concerns the processing of images, it is a special case of our method as they consider only the case of $x_i = d_i$ and do not identify the property of unimodality. A preliminary version of this work was presented in [16]. The previous study focused on the analysis of the unimodality property and on its application to the multi-agent rendezvous problem. We enhanced the spectral graph analysis (Section 6) , and provide the following new contributions: the convergence bounds for static (Section 4) and dynamic (Section 5) graphs, and the algorithmic evaluation (Section 7).

9. CONCLUSIONS

Hitherto, information potentials in sensor networks have been studied without taking into consideration the topology of the network. In this study, we introduced a novel aggregation method that overcomes this limitation. By *anchoring* the information potential of each node to their *original* value – as opposed to letting the information potential *evolve "freely"* at each iteration, our method inherently considers network connectivity in the construction of information potentials. It is also simple, decentralized, robust to communication loss, and adaptive to network dynamics. We exploit linear algebra and spectral graph theory to gain deep insights into the impact of the network topology. We show that potentials are composed out of discrete waves that capture information phenomena of increasing spatial variation; our method reshapes the information landscape by attenuating information of high variation (e. g., noise). The elimination of local maxima has important implications for greedy search methods. Our analysis includes valuable guidelines for practical deployments: (*i*) dense regular networks provide faster convergence rates; (*ii*) expected information and network dynamics can be used to derive appropriate aggregation scopes.

Acknowledgements. Andreas Loukas was supported by the Dutch Technology Foundation STW and the Technology Program of the Ministry of Economic Affairs, Agriculture and Innovation (D2S2 project). Marco Cattani was supported by the Dutch national program COMMIT (project P09 EWiDS). Furthermore, we like to thank the anonymous reviewers and our shepherd Mingyan Liu for providing us detailed feedback on the draft version of this paper.

10. REFERENCES

[1] T. Aysal, A. Sarwate, and A. Dimakis. Reaching consensus in wireless networks with probabilistic broadcast. In *Annual Allerton Conference on Communication, Control, and Computing*, 2009.

[2] O. Babaoglu and et.al. Design patterns from biology for distributed computing. *Trans. Autonomous Adaptive Systems*, 2006.

[3] T. Biyikoglu, J. Leydold, and P. F. Stadler. *Laplacian eigenvectors of graphs: Perron-Frobenius and Faber-Krahn Type Theorems.* Springer, 2007.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 2006.

[5] Y. Cheng. Mean shift, mode seeking, and clustering. *Trans. on Pattern Analysis and Machine Intelligence*, 1995.

[6] M. Cho and K. M. Lee. Mode-seeking on graphs via random walks. *CVPR*, 2012.

[7] F. Chung. *Spectral graph theory.* 1997.

[8] E. Davies, G. Gladwell, J. Leydold, and P. Stadler. Discrete nodal domain theorems. *Linear Algebra and its Applications*, 2001.

[9] F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *IEEE Selected Areas in Communications*, 2008.

[10] J. Gao, L. J. Guibas, J. Hershberger, and L. Zhang. Fractionally cascaded information in a sensor network. In *IPSN*, 2004.

[11] G. M. L. Gladwell and H. Zhu. Courant's nodal line theorem and its discrete counterparts. *Quarterly Journal of Mechanics and Applied Mathematics*, 2002.

[12] L. Grover. Local search and the local structure of NP-complete problems. *Operation Research Letters*, 1992.

[13] D. Kempe, J. Kleinberg, and A. Demers. Spatial gossip and resource location protocols. In *STOC*, 2001.

[14] U. Khan, S. Kar, and J. Moura. Higher dimensional consensus: Learning in large-scale networks. *Trans. on Signal Processing*, 2010.

[15] H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L. J. Guibas. Composable information gradients in wireless sensor networks. In *IPSN*, 2008.

[16] A. Loukas, M. Woehrle, P. Glatz, and K. Langendoen. On distributed computation of information potentials. In *FOMC*, 2012.

[17] P. Ogren, E. Fiorelli, and N. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *Trans. on Automatic Control*, 2004.

[18] R. Sarkar, X. Zhu, and J. Gao. Hierarchical spatial gossip for multiresolution representations in sensor networks. *ToSN*, 2012.

[19] R. Sarkar, X. Zhu, J. Gao, L. J. Guibas, and J. S. B. Mitchell. Iso-contour queries and gradient descent with guaranteed delivery in sensor networks. In *INFOCOM*, 2009.

[20] D. Shoup. Cruising for parking. *Transport Policy*, 13(6):479–486, 2006.

[21] P. Skraba, Q. Fang, A. Nguyen, and L. Guibas. Sweeps over wireless sensor networks. In *IPSN*, 2006.

[22] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 2007.

[23] M. Woehrle, M. Bor, and K. Langendoen. 868 MHz: a noiseless environment, but no free lunch for protocol design. In *INSS*, 2012.