52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

# Distributed Online Topology Design for Network-level Disturbance Rejection

Airlie Chapman, Eric Schoof and Mehran Mesbahi

***Abstract*—In this paper, we examine a networked multi-agent system running the consensus protocol susceptible to misinformation from its environment. The influenced dynamics are modeled with leader-follower dynamics and the impact of the foreign input is measured through the open loop $\mathcal{H}_2$ norm of the network dynamics. To dampen the external disturbances a novel decentralized edge reweighting method is proposed. The method is composed of a decentralized conjugate gradient method coupled with a decentralized online optimization algorithm. The uncertainties of the effect of local rewiring and unknown environmental influences are demonstrated to be well-suited to the online regret framework. A simulation of the reweighting method is discussed and shown to have a small regret.**

## 1. Introduction

Consensus-type systems have become a popular area of research presenting a method for distributed information-sharing and controlled agreement for networked, multi-agent systems. Motivating examples include multi-vehicle control, formation control, swarming, and distributed estimation [1], [2], [3], [4]. One attraction of consensus that for large scale networks it that it can run anonymously, without individual ID tagging of agents. The downside is that there is susceptibility for agents to incorrectly identify agents and other objects in the network. These can be viewed as disturbances to the networked dynamics.

Control theory presents many mechanisms to reject or dampen disturbances in a dynamic system. In general they fall in two categories: active control such as dynamic feedback, and passive control such as structural damping. Typically in networked dynamics systems the network structure is considered a passive element. If feedback is unavailable or global network knowledge is insufficient to apply feedback, adapting the topology may in fact act as an active system via dynamic selection of interconnection (edge) weights. Further this is often a favorable solution when global information is limited and decisions are tentative. Edge reweighting is a conservative response with algorithms like consensus able to perform in the presence of variations in edge weights.

We consider a scenario where an agent $i$ can communicate with any other nearby agents $j \neq i$. This communication does not implicitly involve knowing the location of the peer agents. In the course of a mission, the agent might mistake an object in the environment for a peer agent. The agent would know there is a discrepancy in the data since it knows how many peers it has, and how many objects it can see. It might want to limit the spread of misinformation via network reweighting since its removal might render the network unconnected, especially if other agents are also making the same decisions.

To model such a scenario we consider the agent's false positives as foreign input signals and use the leader-follower dynamics [4] to model the system. The main tool for investigating the susceptibility of these influenced subset of agents is the open loop $\mathcal{H}_2$ norm of these leader-follower dynamics. In particular, the open loop $\mathcal{H}_2$ norm for the network can be employed as a measure to dynamically reweight the interaction topology, reducing the effect of the foreign inputs on the network. Increasing the open loop $\mathcal{H}_2$ norm tends to increase the receptiveness of the system to control.

Designing and adapting topologies to optimize for certain metrics has been addressed by several authors: Ghosh and Boyd [5] aimed to maximize the second smallest eigenvalue of the graph Laplacian; Zelazo and Mesbahi [6] examined the minimization of the network's $\mathcal{H}_2$ performance when noise is applied to the interaction links in the network; Wan *et al.* [7] considered maximizing the largest eigenvalue of the graph Laplacian. All aforementioned authors used centralized optimization techniques over *weighted* graphs. Kim and Mesbahi [8] used fading functions to approximately represent the on/off linkage relationship when searching for the maximum second-smallest eigenvalue of the graph Laplacian to increase the convergence properties of the network dynamics. Wu and Wang [9] have approached the same problem using genetic-algorithms. Intuitive methods for network reconfiguration have been designed to improve network resilience, for example, using thresholding methods to decide when to alter the topology [10]. Chapman and Mesbahi [11] proposed algorithms that performed local edge swap over tree graphs with a game-theoretic analysis to justify the algorithm's performance.

Unique to the aforementioned methods we propose a fully *distributed* reweighting method composed of two stages.

The first stage is the distributed estimation of the local effect of the foreign signals, achieved via the formulation of a distributed conjugate gradient method. In the 1950s, Hestens and Stiefel [12] formulated the linear conjugate gradient method as an iterative method to solve $A^{-1}b$. An attraction of this method is theoretical guaranteed convergence in less than $n$ iterations. In addition, it boasts small storage requirements. In fact, only samples of the range space of $A$ are required rather than knowledge of the complete $A$. We show that, if $A$ encodes the network structure, this feature makes it possible to form a distributed version of the algorithm with the aid of two consensus updates per timestep.

The second stage is the implementation of a distributed online algorithm to reweight the network. The appeal of

This research of the authors was supported by AFOSR grant FA9550-12-1-0203 and ONR grant N00014-12-1-1002. The authors are with the Department of Aeronautics and Astronautics, University of Washington, WA 98105. Emails: {airliec, eschoof, mesbahi}@uw.edu.

the online method is that, irrespective of the changes in the network unseen by a single agent, certain performance guarantees can still be made. The proposed online regret algorithm is a distributed version of a centralized algorithm proposed by Hazan *et al.* [13]. The algorithm is similar to gradient decent or incremental gradient methods surveyed by Bertsekas [14]. The attraction of the proposed algorithm is that it displays an $O(\log(T))$ regret, meaning that on average the algorithm performs as well as the best fixed case solution with the benefit of hindsight. Related online distributed optimization is the work by Yan *et al.* [15] examining a strongly convex cost function decomposable into smaller convex functions. Sundhar *et al.* [16] explored a stochastic subgradient showing convergence to an optimal solution with probability 1. Recently, Raginsky *et al.* [17] focused on a network of agents performing a subgradient-based sequential convex optimization scheme robust to the network structure.

The novelty of our distributed online formulation is that, unlike the aforementioned distributed algorithms, our cost function is not decomposable. We derive a method to instead decompose the gradient of the $\mathcal{H}_2$ norm with respect to the edges. Consequently, it is this formulation that allows our algorithm to be decentralized.

The paper is organized as follows. §2 contains the relevant background pertaining to graphs and online optimization. Consensus-based, leader-follower dynamics is described for the network dynamics in §3 as well as the main dynamics performance measure, the open loop $\mathcal{H}_2$ norm. The distributed conjugate gradient method is then presented in §4 as well as its counter-part, the distributed online gradient descent method. The proposed method is applied and the performance of the algorithm is examined in the regret framework. The paper is concluded with a few remarks in §5.

## 2. Background

### A. Notation

This section provides the models that will be used in this paper, including abbreviated descriptions of graphs and the consensus protocol in both its traditional and leader-follower settings. The following notation is used in this paper: $\|\cdot\|_2$ denotes the 2-norm; $\mathbf{tr}(\cdot)$ denotes the trace of a matrix; $|\cdot|$ denotes the cardinality of a set; $\mathbf{1}$ denotes the column vector of ones; $I$ denotes the identity matrix. For a column vector $v \in \mathbb{R}^p$, $v_i$ or $[v]_i$ denotes the $i$th element of $v$. The $ij$th element of matrix $M$ is $[M]_{ij}$. The term $e_i$ denotes the column vector which contains all zero entries except $[e_i]_i = 1$. For matrices $M, N \in \mathbb{R}^{n\times n}$, $N \preceq M$ denotes that $M-N$ is positive semidefinite. The Hadamard product of two matrices is denoted by $\circ$ where $[M \circ N]_{ij} = [M]_{ij} \cdot [N]_{ij}$.

### B. Graph

To represent the network topology, the communication points in the network are denoted as nodes, and edges are communication links between points. For a multi-agent network, the nodes represent agents and the edges correspond to inter-agent links, such as wireless communications or relative sensors. It is assumed that all communication links are bi-directional. These nodes and edges can therefore be considered as forming an undirected graph.

Abstractly, an undirected graph $\mathcal{G} = (V, E, W)$ is defined by a node set $V$ with cardinality $n$, the number of nodes in the graph, and an ordered edge set $E$ with cardinality $m$, the number of edges in the graph and an ordered weight set $W$, also represented by a diagonal matrix. The ordered edge set is comprised of pairs of nodes, where nodes $v_i$ and $v_j$ are adjacent if $\{v_i, v_j\} \in E \subseteq [V]^2$, the set of two-element subsets of $V$ with each edge having weight $w_{ij} \in W$. The neighborhood set $N(i)$ of node $i$ is composed of all agents in $V$ adjacent to $v_i$. The ordering of the edge set is encoded through the index mapping $\sigma(\cdot)$ such that $l = \sigma(ij)$ if and only if edge $l$ connects nodes $i$ and $j$. If clear, vectors pertaining to the edges such as the edge weights will be denoted as $w_{ij}$ and $w_l$, interchangeably. The *weight matrix* $W(\mathcal{G})$ is an $m \times m$ diagonal matrix with the weight of edge $l$ at position $[W(\mathcal{G})]_{ll}$. The *incidence matrix* $E(\mathcal{G})$ is a $n \times m$ composed of columns $a_{\sigma(ij)}$ which encodes the edge $\sigma(ij)$ with $a_{\sigma(ij)} = e_i - e_j$.[1] The *graph Laplacian* is defined as $L(\mathcal{G}) = E(\mathcal{G})W(\mathcal{G})E(\mathcal{G})^T \in \mathbb{R}^{n\times n}$. In expanded form, $L(\mathcal{G}) = \sum w_{ij} a_{ij} a_{ij}^T$.

### C. Online Convex Optimization

Online convex optimization is formulated as a game where, at each time $t$, a player selects a point $x_t$ in a convex set $\mathcal{P}$, referred to as an action. After the player has committed to the action a convex function $f_t(\cdot)$ is revealed to the player at which point a penalty of $f_t(x_t)$ is paid by the player. The objective of this game is to minimize the accumulative penalty. *Regret* is a common measure for the performance of this highly uncertain online system. Regret is defined as the difference between the cost of the sequence of actions taken by the player and the performance of the best single action $x^*$ taken at every time step if the sequence $\{f_t(\cdot)\}$ is known *a priori*. Hence, the regret of an algorithm with action sequence $\{x_t\}$ is $\mathcal{R}_T = \sum_{t=1}^{T} (f_t(x_t) - f_t(x^*))$. The objective of a good online algorithm is to achieve a guaranteed low regret; Specifically, one that guarantees a sublinear $\mathcal{R}_T$ or $\mathcal{R}_T/T \to 0$. The reasoning is that when $\mathcal{R}_T/T \to 0$, "on average" the algorithm performs as well as the best fixed action in hindsight.

## 3. Model and Measure

### A. Consensus Dynamics

A common objective for networked systems is to reach agreement on one or more of their states. For example, agreement on position achieves rendezvous or, if agreeing on a virtual position, formations can be acquired with known position offsets from the virtual position. Velocity agreement is another attractive property for formation maintenance. For distributed surveillance, bearing agreement is often desired.

[1] The assignment where $a_{\sigma(ij)} = e_j - e_i$, can also be applied.

If agreement is required over a network of agents without all-to-all communications, a distributed approach is necessary. A protocol which is particularly adept at distributed agreement is the consensus protocol, which is now detailed.

Consider $x_i(t) \in \mathbb{R}$ to be the $i$-th node's state at time $t$ on which agreement is required for all nodes. The continuous-time consensus dynamics is defined over the graph $\mathcal{G} = (V, E)$ as

$$\dot{x}_i(t) = \sum_{\{v_i, v_j\} \in E} (x_j(t) - x_i(t)) . \tag{3.1}$$

Thus, only the relative state of node $i$'s neighbors is required to perform an update to node $i$. The collective dynamics over states $x(t) \in \mathbb{R}^n$ is represented as $\dot{x}(t) = -L(\mathcal{G})x(t)$, with $L(\mathcal{G})$ being the graph Laplacian matrix of the underlying interaction topology, described in §2-B. For a connected graph $\mathcal{G}$ the network dynamics will converge to an agreement on the state, i.e., $x_1(t) = x_2(t) = \cdots = x_n(t) = \alpha$, for some constant $\alpha$, from all initial conditions [1].

### B. Leader-Follower Consensus Dynamics

One of the advantages of the consensus dynamics is that the additional agents can seamlessly integrate into the dynamics by entering the communication range of the network. The detriment is that incorrect identification of an agent adds an unwanted signal into the network. Subsequently, an agent incorrectly "follows" this unwanted signal. The dynamics governing this foreign signal takes the form of the popular leader-follower consensus dynamics [4], where the incorrectly identified agent plays the role of a leader, and the native agents in the network as followers.

To form the leader-follower consensus dynamics, the network graph $\mathcal{G} = (V, E)$ is extended to incorporate the foreign agents/signals into the graph. This is accomplished by considering the foreign agent set $\mathcal{R} = (F, R, \mathcal{E}_R)$, where $F$ is the $f$ element foreign node set and $\mathcal{E}_R \subseteq F \times R$ is the set of false edges attached to the network at the $r$ element native node set $R$. It is assumed that a foreign agent $f_j \in F$ is mis-observed at position $u_j(t) \in \mathbb{R}$ by only one native agent. Further, one native agent mistakes at most one foreign agent at a time.[2] The resulting leader-follower consensus system now assumes the form,

$$\dot{x}_i(t) = \sum_{\{v_i, v_j\} \in E} (x_j(t) - x_i(t)) + \sum_{\{r_i, f_j\} \in \mathcal{E}_R} (u_j(t) - x_i(t))$$

with the full dynamics

$$\dot{x}(t) = A(\mathcal{G}, \mathcal{R})x(t) + B(\mathcal{R})u(t), \tag{3.2}$$

where $B(\mathcal{R}) \in \mathbb{R}^{n \times r}$ with $[B(\mathcal{R})]_{ij} = 1$ when $\{r_i, f_j\} \in \mathcal{E}_R$ and $[B(\mathcal{R})]_{ij} = 0$ otherwise, and

$$A(\mathcal{G}, \mathcal{R}) = -(L(\mathcal{G}) + \sum_{i \in R} e_i e_i^T) \in \mathbb{R}^{n \times n}, \tag{3.3}$$

The matrix $A(\mathcal{G}, \mathcal{R})$ in (3.3) is the *Dirichlet matrix*, or grounded Laplacian [18], [19]. In the following discussion, only connected graphs are considered for which $A(\mathcal{G}, \mathcal{R})$ is always negative definite.

[2]This assumption can be easily relaxed to multiple foreign agents.

### C. Disturbance Rejection using the Open Loop $\mathcal{H}_2$ Norm

A measure of the effect that an input, represented by the matrix $B$, on a state dynamics $\dot{x}(t) = Ax(t) + Bu(t)$ is the open loop $\mathcal{H}_2$ norm of the system $\|G(s)\|_2$ where the full-state output state-space realization is $G(s) = (sI - A)^{-1} B$. The metric represents the amplification of the mapping of inputs to the full state outputs, i.e., $y(s) = x(s) = G(s)u(s)$. More precisely, $y(s)$ is the energy of the system at the states from the unit impulse input $u(t)$ when $x(0) = 0$. Consequently, in general, decreasing $\|G(s)\|_2$ has the effect of dampening disturbances in the system inputs through the matrix $B$.

A convenient method of parameterizing the norm $\|G(s)\|_2$ is using the trace of the controllability gramian defined as $P(A, B) := \int_0^\infty e^{A\tau} B B^T e^{A^T \tau} d\tau$. From this relationship, when $A$ is negative definite,

$$\begin{aligned} \|G(s)\|_2^2 &= \mathbf{tr}(\int_0^\infty e^{A\tau} B B^T e^{A^T \tau} d\tau) \\ &= \mathbf{tr}(\int_0^\infty B B^T e^{A^T \tau} e^{A\tau} d\tau) \\ &= \mathbf{tr}(B B^T \int_0^\infty e^{2A\tau} d\tau) \\ &= -\frac{1}{2} \mathbf{tr}(B^T A^{-1} B). \end{aligned} \tag{3.4}$$

From these observations, in general, inputs perturb the outputs more effectively as $\|G(s)\|_2$ increases. With this in mind, the focus of this work is to minimize $\|G(s)\|_2^2$.

The $\mathcal{H}_2$ norm of the system also has attractive graph theoretic properties linked through the concept of the effective resistance of the graph discussed in [11]. A consequence of this connection is that the weights on the edges can be considered as conductances in an equivalent electrical network realization of the graph [18]. Since the effective resistance always decreases with increasing conductances, $\|G(s)\|_2^2$ will decrease or remain constant with any increase in edge weights. This useful property leads to the following section on the reweighting of edges to decrease $\|G(s)\|_2$.

## 4. Distributed Online Topology Design

As many networked systems, such as UAV swarms, require only non-physical interconnections for their coordinated behavior, they have the advantage that their inter-vehicle coordination graph can be *reweighted*. This observation leads to an online method for improving network manageability, namely via a judicious topology reweighting. The goal is to adapt the network topology distributively and with only minimum local knowledge of the network topology so as to improve disturbance rejection. The metric for "good rejection" used in the following analysis is the open loop $\mathcal{H}_2$ norm of the network dynamics, described in §3-C.

The challenge of dynamic distributed reweighting is that agents are unable to coordinate with non-neighboring agents. Consequently, local edge reweightings in light of other agent's reweights can be detrimental. Furthermore, by the time information pertaining to the topological effects of a foreign agent has been received, the foreign agent may no

longer be present. Therefore, the online regret framework is ideal to address this problem. The changing dynamic structure manifests itself as a time-varying graph $\mathcal{G}_t$ with a corresponding time varying $L(\mathcal{G}_t)$. In turn, varying foreign agents appear as a changing foreign agent set $\mathcal{R}_t$. Therefore the state model (3.2) is

$$\dot{x}(t) = A_t x(t) + B_t u(t), \tag{4.1}$$

where $A_t = A(\mathcal{G}_t, \mathcal{R}_t)$ and $B_t = B(\mathcal{R}_t)$. The matrix $A_t$ can be represented as

$$A_t = -\sum_{(i,j)\in E} w_{ij} a_{ij} a_{ij}^T - \sum_{i\in R} e_i e_i^T, \tag{4.2}$$

which is invertible for all positive weights so long as $R$ is nonempty and $\mathcal{G} = (V, E, I)$ is connected.

Thus, our metric of interest becomes $\|G(s)\|_2^2 = -\frac{1}{2}\mathbf{tr}\left(B_t^T A_t^{-1} B_t\right)$. The unconstrained minimization of $\|G(s)\|_2$ will increase the weights on the network arbitrarily. In addition, large edge weights can have adverse effects on the network responsiveness. Consequently, we consider the following minimization so as to balance good rejection with the penalization of overly large weights,

$$f_t(W) = -\frac{1}{2}\mathbf{tr}\left(B_t^T A_t^{-1} B_t\right) + \frac{1}{2} h \mathbf{1}^T W^T W \mathbf{1},$$

where constant $h > 0$ and $W$ is the diagonal weight matrix. The arbitrary removal of weights is also unwanted as the graph can be rendered disconnected. Therefore no edges can be reduced below some positive vector $q_{\sigma(ij)}$, i.e., $w_{ij} \geq q_{\sigma(ij)}$. Additional constraints relating to maximum edge deviations can also be considered. These constraints are specified via vector $u_{\sigma(ij)} \in \mathbb{R}^m$ such that $w_{ij} \leq u_{\sigma(ij)}$, leading to the optimization problem

$$\begin{aligned} &\min f_t(W) \\ &\text{s.t. } \; q_{\sigma(ij)} \leq w_{ij} \leq u_{\sigma(ij)}. \end{aligned} \tag{4.3}$$

A convex constraint set denoted by $\mathcal{P}$ is defined via these linear constraints. The analysis of the derivative and hessian of the metric $\|G(s)\|_2^2$ with respect to the edge weights is relegated to the Appendix, with Proposition 1 proving its convexity via a positive semi-definite hessian. Hence noting that $\mathbf{1}^T W^T W \mathbf{1} = \sum w_{ij}^2 = 0$ if and only if $w_{ij} = 0$ for all $(i,j) \in E$, then $f_t(W)$ is a strictly convex function with hessian $\nabla^2 f_t(W) \succeq hI$.

We proceed to provide a decentralized version of the conjugate gradient method to form $\nabla \|G(s)\|_2^2$ distributively. The gradient is then applied to a distributed formulation of online gradient descent leading to a logarithmic regret bound for problem (4.3).

### A. Local Gradient via Distributed Conjugate Gradient

Necessary to the distributed gradient descent algorithm is the local evaluation of the gradient of $f_t(W)$ with respect to a weight $w_{ij}$. Examining the derivative result in Proposition 1 for $y^s := A(\mathcal{G}, \mathcal{R})^{-1} e_s$ for all $s \in R$ then

$$\frac{\partial \|G(s)\|_2^2}{\partial w_{ij}} = -\frac{1}{2}\sum_{s\in R}\left(y_i^s - y_j^s\right)^2.$$

Consequently, if every agent $i$ has access to $y_i^s$ for all $s \in R$ then neighboring agents $i$ and $j$ can negotiate and calculate the gradient with respect to the edge connecting them.

To this end we propose a distributed form of the linear conjugate gradient method that for each run $s$, provides $y_i^s$ to each agent $i$. The centralized version of the conjugate gradient method solves $A^{-1}b$, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. For $A = A(\mathcal{G}, \mathcal{R})$ and $b = e_p$, the method can be used to find $y^s$. An attraction of the algorithm is that updates only require $b$ and the evaluation of $Ay$ for a given $y \in \mathbb{R}^n$. For the case where $A$ is encoded in the graph structure, this is ideal whereby if each agent $i$ only has knowledge of $y_i$ then $[Ay]_i$ can be calculated by agent $i$ simply by querying its neighbors. Specifically, $[A(\mathcal{G}, \mathcal{R})y]_i = \sum_{j\in N(i)} w_{ij} y_j$.

Our distributed conjugate gradient is featured in Algorithm 1, with timesteps indicated by $k$. The estimate $y$, residues $r$ and conjugate $p$ are the main components of the algorithm updates in lines 11, 12, and 15, respectively. Our method takes the same form as the traditional conjugate gradient with the exception of two agreement variables $\tilde{p}$ and $\tilde{r}$ required per timestep, shown in lines 9 and 13, respectively. These are of the form $\frac{1}{n}\sum_{i\in V} r_i^2$ and $\frac{1}{n}\sum p_i b_i$, respectively, and so can be calculated distributively using a traditional information-based consensus model (3.1). Termination occurs when the average residue $\tilde{r}$ falls below a small threshold value $\epsilon$, which for our application is chosen as $\epsilon = 1e-6$. Convergence is typically fast with theoretical guarantees to converge in less than $n$ steps.

Interestingly, the convergence rates are strongly tied to the spectrum of $A$, and for our application the eigenvalues of $\mathcal{G}$. If the eigenvalues are clustered then the algorithm tends to be more performant. Networks that exhibit this trait are regular graphs and, in general, graphs with many symmetries [20]. In practice the $n$ step guarantee is not always met due to rounding errors and for our case errors in the consensus updates. There are a myriad of techniques to combat this; we found a restart criteria when $\tilde{r}$ increased significantly to be effective and easy to coordinate through the network. For more details on the intricacies of the conjugate gradient method we invite the reader to examine the manuscript [21] by Nocedal and Wright.

Figure 1(a) shows the residue for a sample run of the distributed conjugate gradient method on a 20 node graph with 4 foreign agents. The restart condition was triggered at $k = 7$ due to the increase in the value of $\tilde{r}$. The algorithm converged with $\epsilon = 1e-10$ in only 10 timesteps, much less than the theoretical $n = 20$.

### B. Online Algorithm

Given the online nature of problem (4.3) whereby a member of the arbitrary sequence of strictly convex functions $\{f_t(W)\}$ arrives at each timestep, an online convex algorithm is a natural choice. We present a distributed version of the online gradient descent algorithm formulated by Hazan *et al.* [13]. The algorithm is a variation of the traditional gradient descent algorithm and has a run time $O(n)$ per iteration. The distributed version of the algorithm is presented for our cost

**Algorithm 1:** Distributed Conjugate Gradient Method

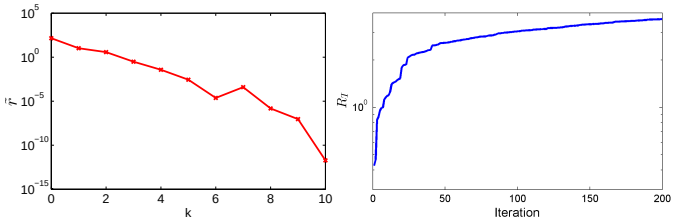
*Given* $y_i[0]$ *for each node* $i \in V$
**Initialize:**
$k = 0$
$r_i[0] = \sum_{j \in N(i)} a_{ij} x_j[0] - b_i$
$p_i[0] = -r_i[0]$
$\tilde{r}_i[0] \leftarrow$ Consensus on $r_i^2 \approx \frac{1}{n} r_i^T r_i$
**while** $|\tilde{r}_i[k]| > \epsilon$ **do**
  $b_i[k] = \sum_{j \in N(i)} a_{ij} p_j[k]$
  $\tilde{p}_i[k] \leftarrow$ Consensus on $p_i b_i \approx \frac{1}{n} p_i^T A p_i$
  $\alpha_i = \frac{\tilde{r}_i[k]}{\tilde{p}_i[k]}$
  $y_i[k+1] = y_i[k] + \alpha_i p_i[k]$
  $r_i[k+1] = r_i[k] + \alpha_i b_i[k]$
  $\tilde{r}_i[k+1] \leftarrow$ Consensus on $(r_i)^2 \approx \frac{1}{n} r_i^T r_i$
  $\beta_i = \frac{\tilde{r}_i[k+1]}{\tilde{r}_i[k]}$
  $p_i[k+1] = \beta_i p_i[k] - r_i[k+1]$
  $k = k + 1$
**end**

function $f_t(W)$ in Algorithm 2. At each time step $k+1$ the local gradient $g_{ij}[k]$ (line 4) of edge $w_{ij}$ at time step $k$ is revealed to agents $i, j$ which perform a gradient descent step in line 5. As the step may be infeasible, a projection $\prod_{\mathcal{P}}(\cdot)$ corrects the step by projecting to the closest point on the constraint set $\mathcal{P}$ under the 2-norm. For the linear constraints in problem (4.3) this can be accomplished distributively and cheaply. An attraction of this algorithm is that Hazan *et al.* [13] proved that if step size is chosen as $\eta[k] = \frac{1}{hk}$ the regret is $\mathcal{R}_T \leq O(\log(T))$, i.e., "on average" the algorithm performs as well as the best fixed strategy in hindsight.

**Algorithm 2:** Distributed Online Gradient Descent

*Given convex set* $\mathcal{P} \subset \mathbb{R}^n$ *and some* $w_{ij}[1] \in \mathcal{P}$, $\forall \{v_i, v_j\} \in E$
**Initialize:**
$k = 1$
**foreach** $\{v_i, v_j\} \in E$ **do**
  $g_{ij}[k] = -\frac{1}{2} \sum_{s \in R} \left(y_i^s[k] - y_j^s[k]\right)^2 + h w_{ij}[k]$
  $w_{ij}[k+1] = \prod_{\mathcal{P}} \left(w_{ij}[k] - \eta[k] g_{ij}[k]\right)$
  $k = k + 1$
  Here, $\prod_{\mathcal{P}}(z) = \arg\min_{x \in \mathcal{P}} \|x - z\|_2$.
**end**

The distributed gradient descent algorithm coupled with the distributed conjugate gradient method was applied to a random 50 node graph. Foreign agents at time-varying locations were detected by the agents marked with large circles. The edge weights were initialized uniformly as 0.25 and constrained such that $q_l = 0.05$ and $u_l = 1$ for all edges in $E$. The resultant graph after each gradient descent step is depicted in Figure 2. The first three timesteps depict the algorithm's response to a static foreign agent location. The notable characteristic, over these images, is the increasing of edge weight on those edges close to the foreign agents. This aligns with the conductance analog mentioned in §3-C, where additional conductance is added to edges in the neighborhood of the foreign agents. These edges are those most likely to dampen the effect of the foreign agents' signal.




(a) (b)

Figure 1. (a) Average residue per timestep of the conjugate gradient method on a 20 node graph. (b) Regret over time of Algorithm 2.

The time evolution of all the figures provides insight into the inner working of the online regret framework. Once a foreign agent location is revealed the algorithm is penalized for the absence of heavily weighted edges in its vicinity for example at timesteps $t = 0$ and 3, respectively. The algorithm recovers by timesteps $t = 2$ and 4, respectively. This process also provides the algorithm with a *memory* of the foreign agent's location notable from the residue edge weights from previous heavily weighted edges as seen at timestep $t = 4$.

The best static graph for this foreign agent evolution was calculated over 200 iterations. The resultant regret is depicted in Figure 1(b) emphasizing the performance agreement with the $\mathcal{O}(\log(T))$ bound found by Hazan *et al.* [13].

## 5. Conclusion

The paper presents a distributed method for the reweighting of network edges so as to dampen the inputs of external signals. The open loop $\mathcal{H}_2$ norm was presented as a metric to quantify the network's susceptibility to such signals. The reweighting algorithm involved the formulation of a distributed conjugate gradient method and a distributed online gradient descent method. The work presents a first foray into the realm of online topology design approaches with proven small regret. The online approach forms an attractive framework to highly uncertain optimization problems. Our future research aims to explore the application of the distributed online approach to the myriad of highly uncertain networked system problems.

## Appendix

**Proposition 1.** *The derivative with respect to* $w_{ij}$ *and Hessian of* $\|G(s)\|_2^2$ *from dynamics* (4.1) *are*

$$\frac{\partial \|G(s)\|_2^2}{\partial w_{ij}} = -\frac{1}{2} \sum_{p \in R} \left( \left[A_t^{-1} e_p\right]_i - \left[A_t^{-1} e_p\right]_j \right)^2$$

*and* $\nabla^2 \|G(s)\|_2^2 = -E^T A_t^{-1} B_t B_t^T A_t^{-T} E \circ E^T A_t^{-1} E$, *which is positive semidefinite.*

*Proof:* From (3.4), the $\mathcal{H}_2$ norm squared is

$$\begin{aligned} \|G(s)\|_2^2 &= -\frac{1}{2} \mathbf{tr}\left(B^T A^{-1} B\right) \\ &= -\frac{1}{2} \sum_{p \in R} e_p^T A^{-1} e_p = -\frac{1}{2} \sum_{p \in R} g_p, \end{aligned}$$

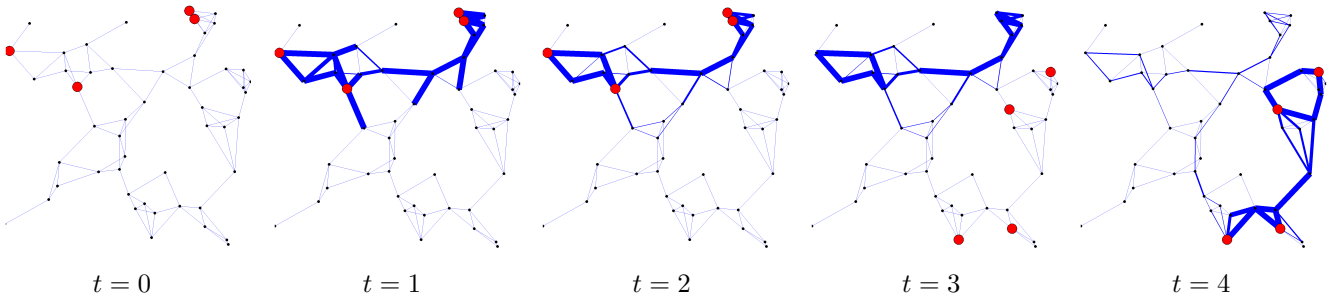



Figure 2. Distributed reweighting with a variable foreign node attachment set $R$, denoted with large circles.

where $g_p(A) = e_p^T A^{-1} e_p$. Further,

$$\begin{aligned}
\frac{\partial g_p(A)}{\partial w_{ij}} &= \mathbf{tr}\left[\left(\frac{\partial g_p(A)}{\partial A}\right)^T \frac{\partial A}{\partial w_{ij}}\right] \\
&= \mathbf{tr}\left[\left(-A^{-T} e_p e_p^T A^{-T}\right)^T \left(-a_{ij} a_{ij}^T\right)\right] \\
&= \mathbf{tr}\left[A^{-1} e_p e_p^T A^{-1} a_{ij} a_{ij}^T\right] \\
&= \mathbf{tr}\left[\left(a_{ij}^T A^{-1} e_p\right)\left(e_p^T A^{-1} a_{ij}\right)\right] \\
&= \left(a_{ij}^T A^{-1} e_p\right)^2 = \left(\left[A^{-1} e_p\right]_i - \left[A^{-1} e_p\right]_j\right)^2.
\end{aligned}$$

Hence,

$$\frac{\partial \|G(s)\|_2^2}{\partial w_{ij}} = -\frac{1}{2} \sum_{p \in R} \left(\left[A^{-1} e_p\right]_i - \left[A^{-1} e_p\right]_j\right)^2.$$

Examining the hessian of $g_p$,

$$\begin{aligned}
\frac{\partial g_p^2(A)}{\partial w_{ij} w_{kl}} &= \frac{\partial}{\partial w_{kl}} \left(a_{ij}^T A^{-1} e_p\right)^2 \\
&= \mathbf{tr}[\frac{\partial \left(\left(a_{ij}^T A^{-1} e_p\right)^2\right)^T}{\partial A} \frac{\partial A}{\partial w_{kl}}] \\
&= \mathbf{tr}[\left(2 a_{ij}^T A^{-1} e_p \left(A^{-T} a_{ij} e_p^T A^{-T}\right)\right)^T a_{kl} a_{kl}^T] \\
&= 2 a_{ij}^T A^{-1} e_p \mathbf{tr}[A^{-1} e_p a_{ij}^T A^{-1} a_{kl} a_{kl}^T] \\
&= 2\left(a_{ij}^T A^{-1} e_p\right)\left(a_{kl}^T A^{-1} e_p\right)\left(a_{ij}^T A^{-1} a_{kl}\right) \\
&= 2\left(a_{ij}^T A^{-1} e_p e_p^T A^{-1} a_{kl}\right)\left(a_{ij}^T A^{-1} a_{kl}\right) \\
&= 2\left[E^T A^{-1} e_p e_p^T A^{-T} E\right]_{(\sigma(ij),\sigma(kl))} \\
&\quad \cdot \left[E^T A^{-1} E\right]_{(\sigma(ij),\sigma(kl))}.
\end{aligned}$$

Consequently, $\nabla^2 g_p = 2 E^T A^{-1} e_p e_p^T A^{-T} E \circ E^T A^{-1} E$, and so

$$\nabla^2 \|G(s)\|_2^2 = -E^T A^{-1} B B^T A^{-T} E \circ E^T A^{-1} E.$$

As $E^T A^{-1} B B^T A^{-T} E = \left(B^T A^{-T} E\right)^T \left(B^T A^{-T} E\right) \succcurlyeq 0$ and $A$ is negative definite then $-E^T A^{-1} E \succcurlyeq 0$. Finally, the Hadamard product exhibits the property that the Hadamard product of two positive semi-definite matrices is positive semi-definite [22] and so the result follows. ∎

## References


[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[2] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.

[3] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[4] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.

[5] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *Proc. 45th IEEE Conference on Decision and Control*, 2006, pp. 6605–6611.

[6] D. Zelazo and M. Mesbahi, "Edge agreement: graph-theoretic performance bounds and passivity analysis," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 544–555, 2011.

[7] Y. Wan, S. Roy, and A. Saberi, "Network design problems for controlling virus spread," in *Proc. 46th IEEE Conference on Decision and Control*, 2007, pp. 3925–3932.

[8] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[9] Z. Wu and R. Wang, "The consensus in multi-agent system with speed-optimized network," *International Journal of Modern Physics B*, vol. 23, no. 10, pp. 2339–2348, 2009.

[10] G. Tyson, A. T. Lindsay, S. Simpson, and D. Hutchison, "Improving wireless sensor network resilience with the INTERSECTION framework," in *Proc. 2nd International Conference on Mobile Lightweight Wireless Systems, Critical Information Infrastructure Protection*, 2010.

[11] A. Chapman and M. Mesbahi, "Semi-autonomous consensus: network measures and adaptive trees," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 19–31, 2013.

[12] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[13] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, 2007.

[14] D. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," in *Lab. for Information and Decision Systems Report LIDS-P-2848*, 2010, pp. 1–38.

[15] F. Yan and S. Sundaram, "Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties," in *arXiv:1006.4039v3*, 2010, pp. 1–24.

[16] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[17] M. Raginsky, N. Kiarashi, and R. Willett, "Decentralized online convex programming with local information," in *Proc. American Control Conference*, 2011, pp. 5363–5369.

[18] P. Barooah and J. P. Hespanha, "Graph effective resistance and distributed control: spectral properties and applications," in *Proc. 45th IEEE Conference on Decision and Control*, 2006, pp. 3479–3485.

[19] S. Salsa, *Partial Differential Equations in Action: From Modelling to Theory*. New York: Springer, 2008.

[20] B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.

[21] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer, 2006.

[22] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1990.