Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Distributed Identification of the Most Critical Node for Average Consensus ⋆

Hao Liu, **Xianghui Cao**∗, **Jianping He**, **Peng Cheng**, **Jiming Chen**, and **Youxian Sun**

State Key Lab. of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China. Emails: liu_hao@zju.edu.cn, {xhcao, jphe, pcheng, jmchen, yxsun}@iipc.zju.edu.cn
∗ *Corresponding author*

Abstract: In present computer networks, cyber attacks, such as compromising and energy depleting attacks, can cause malfunction or even failure of nodes, and can be significantly harmful to the convergence property of the average consensus algorithms. In this paper, we aim to distributively find the most critical nodes in the sense that removing it causes the largest destruction to the converging speed of an average consensus algorithm among all the nodes. The algebraic connectivity is used to assess the destruction and thus the importance of a node. We design two methods to estimate the algebraic connectivity and analyze their bounds for the estimation error. Based on this, we propose a fully distributed algorithm for the nodes to iteratively find the most critical one. Simulation results demonstrate the effectiveness of our algorithm.

Keywords: critical nodes, algebraic connectivity, Fiedler vector, distributed algorithm

1. INTRODUCTION

As a class of important distributed computing methods, the average consensus algorithm has gained intensive attentions recently (Olfati-Saber et al. (2007) and references therein). It requires only limited communication and computation resources for each node, with robustness against unreliable communications and topology changes. Such features have made the algorithm a powerful tool for a variety of applications particularly in large-scale networks, such as formation control/flocking in multi-agent systems, distributed time synchronization in wireless sensor networks, load balancing in communication networks, and time synchronization in oscillators. (Olfati-Saber et al. (2007), He et al. (2013), He et al. (2014), etc).

Relying on network communications among the nodes, the average consensus algorithm is usually subject to cyber attacks, which raises the critical problem of secure consensus(Khanafer et al. (2012)). For example, an adversary can apply denial of service (DoS) attacks to break down some communication links so to prevent the network from converging to consensus(Ganeriwal et al. (2005)). In this regard, there have been many works studying both the adversary strategies and secure mechanisms to protect the consensus algorithms (Pasqualetti et al. (2007), Khanafer et al. (2012) Ghosh and Boyd (2006)).

Another type of malicious attack is directly executed on the nodes by adversaries. For example, in battery powered networks, e.g., wireless sensor networks, an adversary can physically cause a node failure by injecting an excessive number of packets to deplete the energy of this node. If the adversary is powerful enough or there are multiple adversaries that break the network-wide connectivity by attacking some nodes, it is trivial to see that consensus is no longer achievable. In this paper, we consider the non-trivial case that the adversary can only attack one node at a time. In this case, although consensus may still be guaranteed, the performance of the consensus algorithm in terms of converging speed could be varied.

We study the importance of the nodes and design an efficient algorithm to find the most important node. Given global network connectivity information, a smart adversary may choose the most critical node to attack in order to maximize the time to consensus across the residual network. In fact, the algebraic connectivity of a network, is the smallest nonzero eigenvalue of state transition matrix of consensus, and thus characterizes the convergence speed (Xiao and Boyd (2004), Olshevsky and Tsitsiklis (2009)). A secure consensus algorithm should be able to protect the converging performance from such a smart adversary. Therefore, our goal is to identify the most critical node that its removal causes the biggest destruction in terms of longest converging time (or lowest converging speed).

Since the algebraic connectivity is a subtle property of the network topology, it is challenging, if not impossible, to distributively find the most critical node based on the algebraic connectivity without knowledge of the whole network topology. In the literature, maximizing the adjacency matrix spectral radius of the residual network after a node removal has been studied in Restrepo et al. (2006), Milanese et al. (2010). In Watanabe and Masuda

⋆ The work was supported in part by NSFC under Grant 61222305 and 61203036, the 863 High-Tech project under Grant 2011AA040101-1, the SRFDP under Grant 20120101110139, and National Program for Special Support of Top-Notch Young Professionals.

(2010), an estimation method has been proposed to estimate the algebraic connectivity of the residual network after deleting a node based on *Fiedler* vector of the original graph (Fiedler (1973)). Masuda et al. (2013) propose a semi-definite programming based approach to optimize the algebraic connectivity of the residual network after node removals. Nevertheless, the above two approaches are substantially centralized that require global information of the network topology.

In this paper, we propose a novel distributed approach to identify the most critical node whose removal causes the largest decrease of the algebraic connectivity. Specifically, our contributions can be summarized as follows. We formulate the node importance problem as a combinatorial optimization problem and derive an efficient method to estimate it. Based on matrix perturbation theory and matrix norm theory, we develop a theoretical bound for the estimation error. We propose a distributed algorithm, which runs on each node, to identify the most critical node. Through simulations, we show that our algorithm achieves the similar results as the exhaustive searching based optimal solution.

The remainder of this paper is organized as follows. The problem of interest is formulated in Section 2. Section 3 presents the estimation method and derives the estimation error bound. Section 4 describes the distributed algorithm in details. Section 5 presents simulation results and Section 6 concludes this paper.

2. PRELIMINARIES AND PROBLEM FORMULATION

Consider an undirected network described by the graph $G = (V, E)$ with nodes set $V = \{1, 2, ..., n\}$ and edges E representing the communication links. Denote the set of neighbors of node i by $N_i = \{j : (i, j) \in E\}$. $d_i = |N_i|$ is the degree of the node i. The network topology is characterized by a symmetric adjacency matrix $A = [a_{ij}]$, where $a_{ii} = 0$, $a_{ij} = 1$ if $(i, j) \in E$ and $a_{ij} = 0$ otherwise. In the following, $\mathbf{1}$ stands for the vector with all its elements equal to 1, e_i represents the vector with its ith element equals to 1 and the others equal to zero. $\|\cdot\|$ is the 2-norm operator.

2.1 The average consensus algorithm

The general form of an average consensus problem has the following dynamics(Olfati-Saber et al. (2007)):

$$\dot{x}(t) = -Lx(t) \tag{1}$$

where $L = [l_{ij}]$ is the *Laplacian* matrix of the network with $l_{ii} = \sum_j a_{ij}$, and $l_{ij} = -a_{ij}$ for any $i \neq j$. Apparently, the convergence analysis of the consensus problem reduces to spectral analysis of the *Laplacian* matrix of the network topology. L has a simple eigenvalue zero and all the other eigenvalues are nonnegative *iff* the network is connected (Xiao and Boyd (2004)). With a little abuse of notation, we use $\lambda(G)$ to represent corresponding eigenvalues of L. Without loss of generality, assume that $\lambda_1(G) = 0 < \lambda_2(G) \leq \cdots \leq \lambda_n(G)$, where $\lambda_i(\cdot)$ represents the *i-th* smallest eigenvalue of a matrix. $\lambda_2(G)$ is called the algebraic connectivity of the network. λ_2 plays a vital role in the convergence of consensus and network robustness (Fiedler (1973), Xiao and Boyd (2004)). For convenience, we denote $\alpha(G) = \lambda_2(G)$. The corresponding eigenvector is called the *Fiedler* vector. Let μ be the normalized *Fiedler* vector, i.e., $L\mu = \alpha\mu$ and $\|\mu\| = 1$. Similar to Bertrand and Moonen (2012), we make the following assumption.

Assumption 1. λ_2 is a simple eigenvalue of the *Laplacian* matrix L, which guarantees the uniqueness of the *Fiedler* vector μ.

Let G_i denote the graph originated from G with node i and all its incident edges removed. Define an n-dimensional matrix L_i as the *Laplacian* matrix associated with node i, where $l_{ii} = d_i$, for all $j \in N_i$ $l_{jj} = 1, l_{ij} = l_{ji} = -1$, and all the other entries are 0. We specify the n-dimensional matrix L^i as the *Laplacian* matrix of G_i except that the elements in *i-th* row and *i-th* column are all zero. Note that all the matrices defined above are n-dimensional. From the definition of L^i, it is easy to find that the algebraic connectivity of the network after deleting node i is the third smallest eigenvalue of L^i, i.e., $\lambda_3(L^i)$.

2.2 ***Problem formulation***

In this paper, we aim to identify the most critical node whose removal causes the lowest convergence speed (i.e., the largest decrease in algebraic connectivity) among all the nodes in V. That is, we want to find node i that solves the following optimization problem

$$\begin{aligned} \min \quad & \lambda_3(L^i) \\ s.t \quad & L^i = L - L_i, i \in V \end{aligned} \tag{2}$$

Given the *Laplacian* matrix L, an exhaustive search based method can find the optimal node by comparing the algebraic connectivity of the n residual graphs. In addition, SDP (semi-definite programming) method in Masuda et al. (2013) can be also applied to solve this problem. However, both approaches are centralized and not scalable. In this paper, we aim to find a distributed method to the above problem. Below we first propose an efficient method to estimate node importance by using the matrix perturbation theory.

3. NODE IMPORTANCE ESTIMATION

We first give basic matrix perturbation theory to analyze eigenvalue change under node removal. Then, we present two different methods to estimate the *Fiedler* eigenvector after perturbation, and derive an approximation value of algebraic connectivity descent. The accuracy of the approximation mainly depends on the estimation accuracy of *Fiedler* vector. We bound the *Fiedler* vector angle error, based on which we propose an upper bound for the estimation error.

3.1 ***Eigenvalue perturbation analysis***

Considering a symmetric matrix A with a pair of nontrivial eigenvalues λ and μ, we have

$$A\mu = \lambda\mu, \quad \mu^T A = \mu^T \lambda$$

Suppose that A is perturbed with a matrix ΔA and the eigenpair varies with $\Delta\lambda$ and $\Delta\mu$, respectively, i.e.,

$$(A + \Delta A)(\mu + \Delta\mu) = (\lambda + \Delta\lambda)(\mu + \Delta\mu),$$

Let $\nu = \mu + \Delta\mu$, following the similar deduction in Hultgren (2011), we get a new eigenvalue perturbation equation as

$$\Delta\lambda = \frac{\mu^T \Delta A(\mu + \Delta\mu)}{\mu^T(\mu + \Delta\mu)} = \frac{\mu^T \Delta A \nu}{\mu^T \nu} \tag{3}$$

From (3), it is easy to estimate the eigenvalue after perturbation using the original eigenvalue and the corresponding eigenvectors. Since we can formulate node removal as a matrix perturbation, the above analysis lays a foundation for our evaluation of node importance.

3.2 ***Different estimation methods***

For system (1), removing node i yields the perturbation matrix $\Delta L = -L_i$. Thus from (3), we get variation of the algebraic connectivity after removing node i as

$$\Delta\alpha_i = \frac{\sum_{j \in N_i} (\nu_j - \nu_i)(\mu_i - \mu_j)}{\sum_{k=1}^{N} \mu_k \nu_k} \tag{4}$$

where μ_i and ν_i stand for the *ith* element of μ and ν, respectively, and ν represents the eigenvector corresponding to the new *Fiedler* vector. From (4) we know that in order to calculated $\Delta\alpha_i$ distributively, each node needs to calculate μ and the new Fiedler vector ν based on local information exchanges, which, however, is extremely hard. In the following, we propose to estimate the vector ν, which is a first step to the design of our fully distributed algorithm in Section 4.

Lemma 1. (Mohar and Alavi (1991)). For an arbitrary eigenpair μ and λ of the *Laplacian* matrix L, the following conditions hold:

- $\mathbf{1}^T \mu = 0$;
- $(\lambda - d_i)\mu_i = \sum_{j \in N_i} \mu_j$, for all $i \in V$;
- $\nu_i = 0$, for an eigenvector ν of L^i.

From matrix perturbation theory (Nayfeh (2008)), we know that the eigenvector of a matrix with a small (in the sense of some kind of matrix norm) perturbation doesn't change much. Then, in order to estimate $\Delta\alpha_i$ in (4), there are two estimation methods for the unknown ν.

Method 1. $\nu = \mu$

Substituting $\nu = \mu$ into equation (4), we get:

$$\Delta\alpha_i \approx \frac{\mu^T \Delta L \mu}{\mu^T \mu} = -\sum_{j \in N_i} (\mu_i - \mu_j)^2 \tag{5}$$

The intuition behind $\nu = \mu$ is that the perturbation is so small that $\Delta\mu$ is close to 0. This kind of estimation resembles the criterion in Ghosh and Boyd (2006), where $-(\mu_i - \mu_j)^2$ is used to evaluate the importance of an edge on algebraic connectivity. Consequently, the importance of a node is determined by the sum of importance of its incident edges.

Method 2. $\nu = \mu - \mu_i e_i$

This method was first proposed in Watanabe and Masuda (2010) utilizing the fact that $\nu_i = 0$. Based on equation (4), we get the eigenvalue reduction:

$$\Delta\alpha_i \approx \frac{\mu^T \Delta L(\mu - \mu_i e_i)}{\mu^T(\mu - \mu_i e_i)} = \frac{\sum_{j \in N_i} \mu_j(\mu_i - \mu_j)}{1 - \mu_i^2} \tag{6}$$

Among the two estimation methods above, we have taken the advantage of properties of *Fiedler* vector Lemma 1 shows. However, it might be difficult to compare the relative optimality of these methods because different methods gain fluctuant effects with respect to different topologies and even distinct nodes. As one dimension to evaluate the accuracy of the estimation methods above, we try to bound the angle $\angle(\mu, \nu)$ and $\angle(\mu - \mu_i e_i, \nu)$, since the difference between μ and ν heavily depends on their angle.

3.3 ***Characterizing the estimation error bound***

Below we bound the estimation error of ν and $\Delta\alpha_i$ accordingly. We first give some property of L_i.

Lemma 2. $\|L_i\| = d_i + 1$ holds for all $i \in V$.

Proof. Note that L_i is symmetric and it is easy to see that the eigenvalue sequence of L_i is $\{\overbrace{0, ..., 0}^{n-d_i}, \underbrace{1, ..., 1}_{d_i - 1}, d_i + 1\}$. Applying the 2-norm to matrix L_i, we get, $\|L_i\| = d_i + 1$.

Without loss of generality, we assume $\|\mu\| = 1$ and $\|\nu\| = 1$ in the rest of this section. Let α and $\tilde{\alpha}$ denote the algebraic connectivity of L and $L + \Delta L$ respectively.

Lemma 3. (Mohar and Alavi (1991), Fiedler (1973)). Removing a node from original network G, we have: $\max\{0, \alpha - 1\} \le \tilde{\alpha} \le \lambda_3$.

Lemma 3 gives a compact upper and lower bounds for the algebraic connectivity using the current two eigenvalues, which provides a reference for our expected descent in algebraic connectivity and will be used in the simulation part.

Theorem 1. The angle between μ and ν satisfies:

$$\sin\angle(\mu, \nu) \le \frac{\min\{\sqrt{d_i + 1}, \sqrt{n - d_i - 1}\}}{\min\{\tilde{\alpha}, \lambda_3 - \tilde{\alpha}\}}$$

Proof. Let Y be a matrix with columns consisting of vectors pairwise orthonormal and are all orthonormal with μ. Vector ν can be normalized as,

$$1 = \|\nu\| = \left\| \begin{bmatrix} \mu^T \\ Y^T \end{bmatrix} \nu \right\| = \sqrt{(\mu^T \nu)^2 + (Y^T \nu)^2}$$

Note that $cos\angle(\mu, \nu) = \mu^T \nu$ since $\|\mu\| = \|\nu\| = 1$. Thus, $\sin\angle(\mu, \nu) = \|Y^T \nu\|$. Since $(L + \Delta L)\nu = \tilde{\alpha}\nu$, we have $\tilde{\alpha}\nu - L\nu = \Delta L \nu$ and hence Y^T, $Y^T \tilde{\alpha}\nu - Y^T L\nu = Y^T \Delta L \nu$. In the sense that L is real and symmetric, we certainly get n orthogonal eigenvectors. If we choose all the other eigenvectors of L except μ to form Y, we have

$$\begin{bmatrix} \mu^T \\ Y^T \end{bmatrix} L \begin{bmatrix} \mu & Y \end{bmatrix} = \begin{bmatrix} \alpha & 0 & \cdots \\ 0 & \lambda_1 & 0 \\ \vdots & 0 & \ddots \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & [M] \end{bmatrix}$$

In the meanwhile, we have the following conditions hold, $Y^TY = I, Y^TLY = M$; then $Y^TLYY^T = MY^T = Y^TL$. Thus we get, $Y^T\tilde{\alpha}\nu - MY^T\nu = Y^T\Delta L\nu$; then $Y^T\nu = (\tilde{\alpha}I - M)^{-1}Y^T\Delta L\nu$. Apparently,

$$\left\|Y^T\nu\right\| \le \left\|(\tilde{\alpha}I - M)^{-1}\right\| \|\Delta L\nu\|$$

From the structure of ΔL, we get

$$(\Delta L\nu)_j = \begin{cases} \nu_j, & j \in N_i \\ 0, & j \notin N_i,\ and\ j \neq i \\ -\sum\limits_{j\in N_i} \nu_j, & j = i \end{cases}$$

Then, one can prove that

$$\|\Delta L\nu\| < \min\{\sqrt{d_i+1}, \sqrt{n-d_i-1}\}$$

Thus

$$\sin\angle(\mu,\nu) < \left\|(\tilde{\alpha}I - M)^{-1}\right\| \cdot \min\{\sqrt{d_i+1}, \sqrt{n-d_i-1}\}$$

In the meanwhile,

$$\left\|(\tilde{\alpha}I - M)^{-1}\right\| = \max_{i\neq 2}\{\frac{1}{|\tilde{\alpha}-\lambda_i|}\} = \max\{\frac{1}{\tilde{\alpha}}, \frac{1}{\lambda_3 - \tilde{\alpha}}\}$$

Finally

$$\sin\angle(\mu,\nu) \le \frac{\min\{\sqrt{d_i+1}, \sqrt{n-d_i-1}\}}{\min\{\tilde{\alpha}, \lambda_3-\tilde{\alpha}\}},$$

which completes the proof.

Now we are ready to bound the angle $\angle(\mu - \mu_i e_i, \nu)$ of the second estimation method.

***Corollary** 1.* $\cos\angle(\mu - \mu_i e_i, \nu) = \frac{\cos\angle(\mu,\nu)}{\sqrt{1-{\mu_i}^2}}$.

Proof. we have the following equality:

$$\cos\angle(\mu - \mu_i e_i, \nu) = \frac{(\mu-\mu_i e_i)^T\nu}{\|\mu-\mu_i e_i\|\,\|\nu\|} = \frac{\mu^T\nu - \mu_i {e_i}^T\nu}{\sqrt{1-{\mu_i}^2}}$$

Because ${e_i}^T\nu = \nu_i = 0$,

$$\cos\angle(\mu - \mu_i e_i, \nu) = \frac{\cos\angle(\mu,\nu)}{\sqrt{1-{\mu_i}^2}}$$

which completes the proof.

***Remark** 1.* From Corollary 1, some other bounds can be obtained easily, e.g., $\cos\angle(\mu,\nu) \le \sqrt{1-{\mu_i}^2}$, and $\sin\angle(\mu,\nu) \ge |\mu_i|$, which means that the larger μ_i is, the bigger the angle deviates from 0.

***Remark** 2.* Because $\sqrt{1-\mu_i^2} \le 1$, Corollary 1 shows an intuitive explanation that the second estimation outperforms the first one.

The above angle error only reflects the estimation error of algebraic connectivity indirectly. In the following, we directly bound the relative estimation error of the eigenvalue. Denote the estimation error of $\Delta\alpha_i$ as δ_i. Then,

$$\begin{aligned}\delta_i &= |\mu^T\Delta L_i \cdot (\frac{\nu}{\mu^T\nu} - \frac{\omega}{\mu^T\omega})| \\ &= \left|\sum_{j\in N_i}(\mu_i-\mu_j)(\frac{\nu_j-\nu_i}{\mu^T\nu} + \frac{\omega_i-\omega_j}{\mu^T\omega})\right| \qquad (7)\end{aligned}$$

where ω represents the estimation of ν.

***Theorem** 2.* If the estimation vector ω satisfies $\omega = \nu + k\gamma$, where k is an arbitrary real number and γ is orthonormal with μ (e.g., $\gamma = 1$), then $\delta_i = 0$.

Proof. If $\omega = \nu + k\gamma$, then $\mu^T(\omega - \nu) = k\mu^T\gamma = 0$, thus $\mu^T\nu = \mu^T\omega$. Since we have $\nu_i = 0$, finally we get $\delta_i = 0$.

Considering the estimation method in (4), i.e., $\omega = \mu - \mu_i e_i$, we have the following theorem.

***Theorem** 3.* An upper bound for δ_i is:

$$\delta_i \le \min\{\bar{\delta}_{1,i}, \bar{\delta}_{2,i}, \bar{\delta}_{3,i}\}$$

where
$\bar{\delta}_{1,i} = |\lambda_3 - \lambda_2 - \sum_{j\in N_i}\frac{\mu_j(\mu_i-\mu_j)}{1-{\mu_i}^2}|$,
$\bar{\delta}_{2,i} = |\max\{-1, -\lambda_2\} - \sum_{j\in N_i}\frac{\mu_j(\mu_i-\mu_j)}{1-{\mu_i}^2}|$,
and $\bar{\delta}_{3,i}$ is described below in (9).

Proof. From Lemma 3 we get $\max\{-1, -\lambda_2\} \le \Delta\alpha_i \le \lambda_3 - \lambda_2$, and from the definition of δ_i, the following equation holds:

$$\delta_i = |\Delta\alpha_i - \sum_{j\in N_i}\frac{(\omega_j-\omega_i)(\mu_i-\mu_j)}{\mu^T\omega}| \qquad (8)$$

Substituting the bound of $\Delta\alpha_i$ into (8) yields the bounds $\bar{\delta}_{1,i}$ and $\bar{\delta}_{2,i}$.

Based on the **Method** 2 in(equation (6)) and that $\mu^T\omega = \mu^T(\mu - \mu_i e_i) = 1 - \mu_i^2$ (according to Corollary 1), (7) becomes

$$\delta_i = \left|\sum_{j\in N_i}(\mu_i-\mu_j)(\frac{\nu_j}{\cos\angle(\mu,\nu)} - \frac{\mu_j}{1-{\mu_i}^2})\right| \qquad (9)$$

Based on Theorem 1, we can obtain a pair of upper and lower bounds for $\cos\angle(\mu,\nu)$. Using these bounds, we can further obtain another upper bound, denoted by $\bar{\delta}_{3,i}$ of δ_i based on (9). The explicit expression for $\bar{\delta}_{3,i}$ is straightforward and omitted here to save space.

Based on Theorem 3, the estimation error of algebraic connectivity after removing one node is upper-bounded. If only for relative node importance comparison, the estimation in (6) provides an efficient replacement for the unknown vector ν.

4. DISTRIBUTED ALGORITHM DESIGN

In this section, we propose a distributed algorithm to identify the most critical node. In Bertrand and Moonen (2012), the authors proposed a distributed algorithm to compute the *Fiedler* vector based on power iteration. With this method, node i can finally has access to the *i-th* element of the *Fiedler* vector. Unfortunately, the converged vector is not necessarily normalized and the vector's length is unknown to every node. If we take a new parameter S to denote the square of length of the converged *Fiedler* vector, we immediately obtain an intuitive distributed algorithm to find the most critical node: after the convergence of the vector, we can apply an average-consensus to estimate S and a parallel minimum-consensus to identify the most important node. The major problem with this method is that the time needed to reach the final results is too long: it tends to be the addition of two sequential infinite process, i.e., the power iteration process and the average-consensus process. To obtain a more practical implementation, we need to modify the criteria.

Since μ is not normalized, based on the above second estimation method, we transform equation (6) into $\Delta\alpha_i \approx \frac{\sum_{j\in N_i} \mu_j(\mu_i-\mu_j)}{S-{\mu_i}^2}$. Note that the numerator of the criteria is totally node-specific while the denominator requires the global information of S. However, the square differences μ_i^2 among nodes are usually small compared to S, especially in large densely connected networks. Thus node importance $\Delta\alpha_i$ can be reduced into

$$\Delta\beta_i = \sum_{j\in N_i} \mu_j(\mu_i - \mu_j) \tag{10}$$

Theorem 4. For a pair of nodes i, k, the two criteria based on $\Delta\alpha$ and $\Delta\beta$ yields the same importance order of the two nodes *iff*

$$S \geq \frac{\Delta\beta_i {\mu_k}^2 - \Delta\beta_k {\mu_i}^2}{\Delta\beta_i - \Delta\beta_k}$$

Note that $\Delta\beta_i$ can be calculated in a distributed way after we calculate *Fiedler* vector according to Bertrand and Moonen (2012). Furthermore, as we only care about the correct direction of *Fiedler* vector, we can modify the algorithm to identify the most critical node in a parallel way. Algorithm 1 shows the details of the proposed distributed method. Algorithm 1 generates a number of

Algorithm 1: Distributed critical node identification

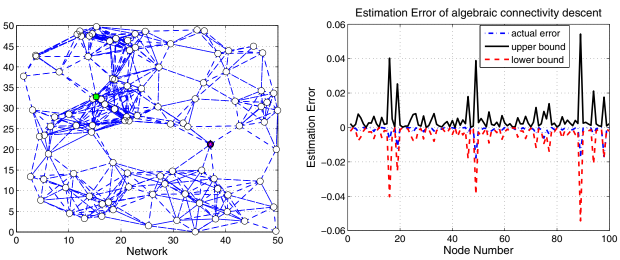
Initialization: every node i stores a flag bit $f_i = 1$, set $t \leftarrow 0$;
if $(t \mod D) = 0$ **then**
  each node checks whether $f_i = 1$ or not;
  Node(s) with $f_i = 1$ is (are) the most critical;
else
  at all nodes, set $f_i = 1$;
  Calculate $\Delta\beta_i$ according to current μ_i and $\mu_j, j \in N_i$;
  Each node i transmits $\Delta\beta_i$ to its neighbors in N_i and computes:
  $$\Delta\beta_i(t+1) = \min\{\Delta\beta_j(t), \Delta\beta_i(t)\}, j \in N_i$$
end
if $\Delta\beta_i(t+1) \neq \Delta\beta_i(t)$ **then**
  set $f_i = 0$;
  $\Delta\beta_i(t+1) \leftarrow \Delta\beta_i(t)$, $t \leftarrow t+1$;
end
Return to step 2;

critical node(s) every D period if D is larger than the diameter of the network. For simplicity, we can just set $D = N - 1$. Upon the convergence of the *Fiedler* vector calculation process, our algorithm will yield a converged optimal node. The critical nodes found by Algorithm 1 will quickly converge in D steps.

Although we can exhaustively search the optimal node, such method requires network-scale times of eigenvalue calculations, i.e., its time complexity is $O(n^4)$. In contrast, Algorithm 1 only needs to compute the original eigenspace once, i.e., its time complexity is $O(n^3)$.

5. SIMULATION RESULTS

We evaluate the proposed method over a network with topology as shown in Fig 1(a). There are $n = 100$ nodes with communication range $d = 10$ in a square area of 50×50. Fig 1(a) also shows that optimal nodes produced by, exhaustion (red dot), centralized (blue diamond) and distributed estimation (pink pentagon) methods overlap, while that by the degree-based(green square) algorithm is quite different. The estimation error of (what) among the network nodes is displayed in Fig 1(b). We can see that



(a) Identifying result (b) Error $(\tilde{\alpha} - \alpha) - \Delta\alpha$

Fig. 1. Random network with $n = 100, d = 10, 50 \times 50$.

the estimation error is very small in the order of (10^{-2}) compared to the magnitudes of algebraic connectivity descent(which may be the scale of 1 according to Lemma 3). The reason why we don't use the relative estimation error is that we don't distinguish between nodes whose removal lead to very small algebraic connectivity descent. Thus the estimation proved to be an effective criteria to evaluate node importance.

Under the same circumstance of Fig 1(a), we calculate four normalized criteria(i.e., degree, exhaustion for algebraic connectivity after removal of each node, centralized criteria in equation (6), distributed criteria of (10))of each node.For degree vector, we normalize degree vector d as

$$normalized\_d_i = \frac{max(d) - d_i}{max(d) - min(d)}, i = 1, ..., n$$

For other three measurements,

$$normalized\_c_i = \frac{c_i - min(c)}{max(c) - min(c)}, i = 1, ..., n$$

Normalized criteria of four methods are plot in Fig 2. It

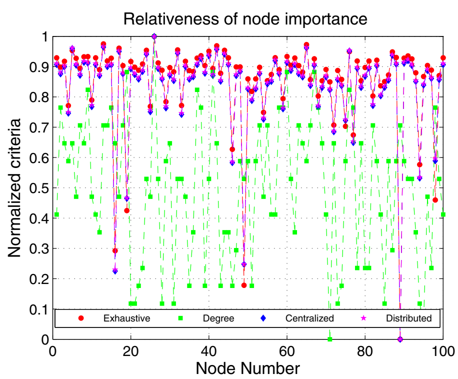
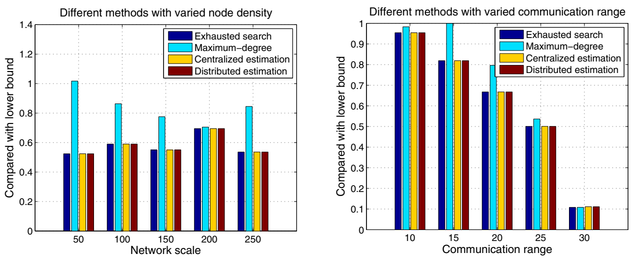


Fig. 2. Relative importance of nodes.

shows that the lines for centralized/distributed criteria almost parallel with the exhaustion one, which means their estimation method reflects the relative importance of nodes at all the nodes. The most critical node tends out to be node 89. However, the degree-based criterion fails to characterize node importance. For example, for nodes $20 \sim 23$, the order of node importance as obtained based on the degree-based criterion is inverse to that obtained by the other three methods.

To further evaluate our algorithm a number of different random networks have been generated with diverse node density and communication range. The results are shown in bar graphs where the height represents the algebraic connectivity after removal of optimal node identified by those four methods. Note that we only care about the relative optimality, so we subtract the lower bound $(\alpha-1)$(see Lemma 3) for convenience. In Fig 3(a), we have compared four methods with fixed communication range $d = 20$ and different numbers of nodes $n = 50, 100, 150, 200, 250$.



(a) Varied node density (b) Varied communication range

Fig. 3. Different methods of exhaustion,maximum-degree,centralized,distributed from left to right.

In all cases in Fig 3(a), both centralized and distributed methods show the same results with exhausted search, which indicates that they identify they always identify the same optimal node. In contrast, the nodes identified by the maximum-degree node are often not optimal. We also carry out simulations with varied communication ranges(i.e. varied connectivity) with $d = 10, 15, 20, 25, 30$ and fixed $n = 200$ in Fig 3(b),which again advocate the effectiveness of the proposed distributed algorithm. In addition, horizontal comparison shows that the higher connectivity a network has, the closer the optimal solution to the lower bound.

6. CONCLUSION AND FUTURE WORK

We have studied the problem of evaluating node importance in a network running the average consensus algorithm and formulate it as a combinatorial optimization problem. To identify the optimal node, a centralized criteria has been proposed and an efficient estimation method to approximate the algebraic connectivity descent has been given. In addition, the bounds for the estimation error has been discussed. A distributed algorithm to identify the most critical node has been proposed. Extensive simulations based on various network topologies show that our algorithm finds the optimal node. The proposed method can be useful for consensus designers to protect consensus convergence property against node attacks. In our future work, we will consider the scenario of removing multiple nodes.

REFERENCES

Bertrand, A. and Moonen, M. (2012). Distributed computation of the fiedler vector with application to topology inference in ad hoc networks. *Signal Processing*, 93(5), 1106–1117.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2), 298–305.

Ganeriwal, S., Čapkun, S., Han, C.C., and Srivastava, M.B. (2005). Secure time synchronization service for sensor networks. In *Proceedings of the 4th ACM workshop on Wireless security*, 97–106.

Ghosh, A. and Boyd, S. (2006). Growing well-connected graphs. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 6605–6611.

He, J., Chen, J., Cheng, P., and Cao, X. (2014). Secure time synchronization in wireless sensor networks: A maximum consensus-based approach. *IEEE Transactions on Parallel and Distributed Systems*, 25(4), 1055–1065.

He, J., Cheng, P., Shi, L., and Chen, J. (2013). Sats: Secure average-consensus-based time synchronization in wireless sensor networks. *IEEE Transactions on Signal Processing*, 61(24), 6387–6400.

Hultgren, N. (2011). Centrality and network analysis: A perturbative approach to dynamical importance. In *Undergraduate Hornors Theses and Student papers, Duke University.*

Khanafer, A., Touri, B., and Basar, T. (2012). Consensus in the presence of an adversary. In *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, 276–281.

Masuda, N., Fujie, T., and Murota, K. (2013). Application of semidefinite programming to maximize the spectral gap produced by node removal. In *Complex Networks IV*, 155–163.

Milanese, A., Sun, J., and Nishikawa, T. (2010). Approximating spectral impact of structural perturbations in large networks. *Physical Review E*, 81(4).

Mohar, B. and Alavi, Y. (1991). The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2, 871–898.

Nayfeh, A.H. (2008). *Perturbation methods*. John Wiley & Sons.

Olfati-Saber, R., Fax, J.A., and Murray, R.M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.

Olshevsky, A. and Tsitsiklis, J.N. (2009). Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1), 33–55.

Pasqualetti, F., Bicchi, A., and Bullo, F. (2007). Distributed intrusion detection for secure consensus computations. In *Proceedings of the 46th IEEE Conference on Decision and Control*, 5594–5599.

Restrepo, J.G., Ott, E., and Hunt, B.R. (2006). Characterizing the dynamical importance of network nodes and links. *Physical review letters*, 97(9).

Watanabe, T. and Masuda, N. (2010). Enhancing the spectral gap of networks by node removal. *Physical Review E*, 82(4).

Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1), 65–78.