

## Using Hierarchical Decomposition to Speed Up Average Consensus<sup>\*</sup>

Michael Epstein<sup>\*</sup> Kevin Lynch<sup>\*\*</sup> Karl H. Johansson<sup>\*\*\*</sup>  
Richard M. Murray<sup>\*</sup>

<sup>\*</sup> *California Institute of Technology, Pasadena, CA USA (e-mail:  
epstein,murray@caltech.edu).*

<sup>\*\*</sup> *Northwestern University, Evanston, IL USA (e-mail:  
kmlynch@northwestern.edu).*

<sup>\*\*\*</sup> *Royal Institute of Technology, Stockholm, Sweden (e-mail:  
kallej@kth.se).*

---

**Abstract:** We study the continuous-time consensus problem where nodes on a graph attempt to reach average consensus. We consider communication graphs that can be decomposed into a hierarchical structure and present a consensus scheme that exploits this hierarchical topology. The scheme consists of splitting the overall graph into layers of smaller connected subgraphs. Consensus is performed within the individual subgraphs starting with those of the lowest layer of the hierarchy and moving upwards. Certain “leader” nodes bridge the layers of the hierarchy. By exploiting the increased convergence speed of the smaller subgraphs, we show how this scheme can achieve faster overall convergence than the standard single-stage consensus algorithm running on the full graph topology. The result presents some fundamentals on how the communication architecture influences the global performance of a networked system. Analytical performance bounds are derived and simulations provided to illustrate the effectiveness of the scheme.

---

### 1. INTRODUCTION

Recent years have seen a large amount of research focused on issues relating to multi-agent and cooperative control (Olfati-Saber et al. (2007)). The task is generally to have a group of systems/agents collectively achieve a desired task in a decentralized fashion while making use of shared information. Examples of specific applications include those in the areas of; consensus (Olfati-Saber et al. (2007)), behavior of swarms (Xi et al. (2005)), multi-vehicle formation control (Fax and Murray (2004)), sensor fusion (Spanos et al. (2005)) and many others. In this paper we focus on the problem of consensus, i.e., having a group of agents reach agreement/consensus on a quantity of interest.

The average consensus problem, which we consider here, is to find a distributed algorithm such that a collection of agents reaches consensus on the average of their initial conditions. To do so the agents must communicate their values to other agents, but they can only communicate with some subset of the other nodes. Given certain conditions on the topology of this communication network, and using an update rule that changes their value in the direction of the aggregate value of the nodes they communicate with the average consensus can be achieved.

---

<sup>\*</sup> The work by K. H. Johansson was partially done during a stimulating sabbatical visit at Caltech. The work was partially supported by the Swedish Foundation for Strategic Research, the Swedish Research Council, the KTH ACCESS Linnaeus Center, and the European Commission through the HYCON Network of Excellence.

In addition to proving consensus can be reached, performance of the consensus algorithm has been a key area of research. One performance measure is in terms of the robustness to possible communication sharing impediments. Issues such as communication delays and changes in the communication topology over time have been examined, see Olfati-Saber and Murray (2004) and Ren and Beard (2005). Another performance measure that has been a key area of research and is the focus of the present work is the time to reach consensus.

Tools from graph theory (Godsil and Royle (2001)) have been used to represent the information sharing topology and aid in the analysis in consensus problems. The key factor in determining the time to convergence has been shown to be the second smallest eigenvalue, represented as  $\lambda_2$ , of the graph Laplacian (the matrix representing the evolution of the agent's values based on the communication topology). Yang et al. (2006) attempted to speed up the time to convergence, while trading-off robustness, by optimally choosing how much relative weight each node should give to the other nodal values in the update rule in order to maximize the ratio  $\lambda_2/\lambda_{\max}$ . A version for the discrete time case was given in Xiao and Boyd (2004). Introducing additional communication links into the network and creating small world networks, as was done in Olfati-Saber (2005), is another approach. The main idea of these approaches and others is the attempt to increase  $\lambda_2$ .

In this work we introduce a new approach to speed up convergence in consensus algorithms applicable to graphs that can be decomposed into a hierarchical graph. It consists of splitting the overall graph into layers of smaller connected

subgraphs. Consensus is performed within the individual subgraphs starting with those of the lowest layer of the hierarchy and moving upwards. Certain “leader” nodes bridge the layers of the hierarchy. By exploiting the larger  $\lambda_2$  values of the smaller subgraphs, this scheme can achieve faster overall convergence than the standard single-stage consensus algorithm running on the full graph topology. Furthermore, using consensus for the individual subgraphs endows them with the benefits associated with standard consensus algorithm, such as robustness to information perturbations, no need for a global planner within the subgraphs, etc. The contribution of this paper is to extend the basic understanding of consensus algorithms to situations when the system may have a hierarchical structure. This hierarchical structure is typical in layered communication networks, where some nodes are gateways between clusters of local nodes and the rest of the network.

The paper is organized as follows. Section 2 provides a quick review of graph theory and continuous-time consensus algorithms, and introduces the concept of hierarchical decomposition of graphs. The hierarchical consensus algorithm is described and analyzed in Section 3. Section 4 contains examples and simulations to illustrate the algorithm. Finally the paper ends with conclusions and a description of future work in Section 5.

## 2. GRAPHS AND CONTINUOUS-TIME CONSENSUS

### 2.1 Graph Theory

Consider a group of  $N$  agents with an underlying graph topology  $\mathcal{G} = \{V, E\}$ , where  $V = \{1, \dots, N\}$  is the set of nodes and the edge set  $E = \{(i, j)\} \subseteq V \times V$  is the node pairs  $(i, j)$  where node  $j$  sends information to node  $i$ . If the communication link between nodes  $i$  and  $j$  is bidirectional then both  $(i, j)$  and  $(j, i) \in E$ . Define the neighbor set of node  $i$  as  $\mathcal{N}_i = \{j \mid (i, j) \in E\}$ . The adjacency matrix is the matrix  $A = [a_{ij}]$  where  $a_{ij} = 1$  if  $(i, j) \in E$  otherwise  $a_{ij} = 0$ , it is usually assumed that  $a_{ii} = 0$ . Letting  $D = \text{diag}(d_i)$  with  $d_i = \sum_j a_{ij}$  we then

have the Laplacian matrix for graph  $\mathcal{G}$  given by

$$L = D - A. \quad (1)$$

The Laplacian matrix has been widely studied in the context of graph theory (Godsil and Royle (2001)) and it plays a central role in the analysis of consensus algorithms (Olfati-Saber et al. (2007)).

### 2.2 Hierarchical Graph Decomposition

The algorithm presented in this paper is applicable to graphs that can be hierarchically decomposed. In this section we provide a formal definition for a hierarchical decomposition and present some associated properties.

*Definition 1.* An  $M$ -layer hierarchical decomposition of a connected graph  $\mathcal{G} = \{V, E\}$  consists of a collection of subgraphs  $\mathcal{G}_j^i = \{V_j^i, E_j^i\}$ , with  $i = 1, \dots, M$ , of  $\mathcal{G}$ . The vertex set of  $\mathcal{G}_j^i$  is denoted by  $V_j^i \subseteq V$  and  $E_j^i = \{(m, n) \in E \mid m, n \in V_j^i\} \subseteq E$  denotes the edge set. Let  $S^i$  denote the number of subgraphs,  $\mathcal{G}_1^i, \mathcal{G}_2^i, \dots, \mathcal{G}_{S^i}^i$ , at layer  $i$ . Let

$\mathcal{V}^i = \bigcup_{j=1}^{S^i} V_j^i$  be the set of all nodes in layer  $i$ . The collection of subgraphs  $\mathcal{G}_j^i$  must satisfy the following properties:

- (1) Each  $\mathcal{G}_j^i$  is connected and  $|V_j^i| \geq 1$ .
- (2) There is only one top-layer graph, i.e.,  $S^M = 1$ .
- (3) The lowest-layer graphs contain all the nodes of the graph, i.e.,  $\mathcal{V}^1 = V$ .
- (4) The subgraphs at the same layer  $i$  are disjoint, i.e.,  $V_j^i \cap V_k^i = \emptyset$  for all  $j \neq k$  and  $j, k \in \{1, \dots, S^i\}$ .
- (5) For each  $\mathcal{G}_m^i, i < M$ , there exists exactly one parent subgraph  $\mathcal{G}_m^{i+1}$  that shares a single node, i.e., exactly one  $m \in \{1, \dots, S^{i+1}\}$  satisfies  $|V_j^i \cap V_m^{i+1}| = 1$ .

An example of a hierarchical decomposition is given in Fig. 1. Notice the hierarchical decomposition does not require that all available links be utilized since the links between nodes 2 and 3 as well as between nodes 5 and 6 are never used. Of course each graph does not necessarily have a unique hierarchical decomposition, and others could be created that utilize these links.

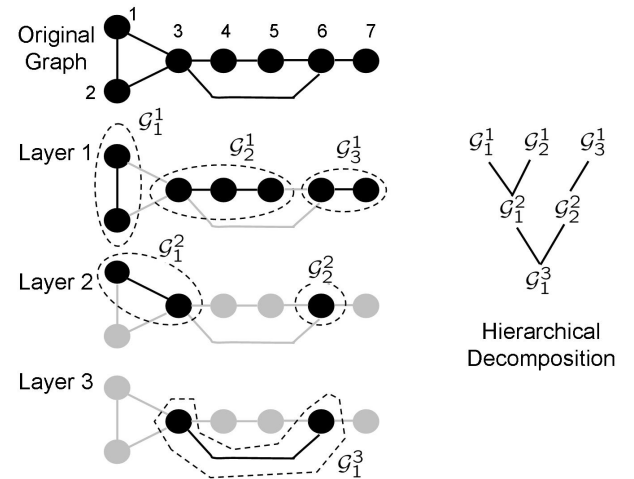


Fig. 1. Example hierarchical decomposition.

We introduce a few more definitions and properties associated to the subgraphs of a hierarchical decomposition. All nodes present in a given layer  $i < M$  have an associated *parent node* for that layer. If node  $k$  is in subgraph  $\mathcal{G}_j^i$ , then node  $k$ 's parent node is the one node in  $\mathcal{G}_j^i$  that is also in some subgraph of the next layer  $\mathcal{G}_m^{i+1}$ . Let  $p(k, i)$  denote the parent node of node  $k$  for layer  $i$ , i.e., if  $k \in V_j^i$  and its parent node is in  $V_m^{i+1}$  then  $p(k, i) = V_j^i \cap V_m^{i+1}$ . A node can be its own parent node. As an example, referring to the hierarchical decomposition in Fig. 1 we see that in layer 1 node 1 is the parent node for both itself and node 2, i.e.,  $p(1, 1) = p(2, 1) = 1$ .

For every layer  $i$  each node  $k$  also has an associated *leader node* denoted by  $\mathcal{L}_k^i$  whose definition is slightly different than its parent node. If a node  $k$  is in layer  $i$ ,  $k \in \mathcal{V}^i$ , then it is its own leader node  $\mathcal{L}_k^i = k$ . If the node is in layer  $i - m$  but is not in any of layers from  $i - m + 1, i - m + 2, \dots, i$ , i.e.,  $k \in \mathcal{V}^{i-m}$  but  $k \notin \bigcup_{l=0}^{m-1} \mathcal{V}^{i-l}$ , then its leader node is

the successive parent node from layer  $i - m$  to  $i - 1$ , i.e.,  $\mathcal{L}_k^i = p(p(\dots p(p(k, i - m), i - m - 1), \dots), i - 1)$ . The leader node will be shown to be important later on as each node will “follow” the value of their leader node. In Fig. 1 we see that node 2's leader nodes in layers 1, 2 and 3 are nodes 2, 1 and 3 respectively, i.e.,  $\mathcal{L}_2^1 = 2$ ,  $\mathcal{L}_2^2 = 1$  and  $\mathcal{L}_2^3 = 3$ .

The *total node set* for the subgraph  $\mathcal{G}_j^i$ , denoted  $\mathbb{V}_j^i$ , is the set of all nodes whose leader node is in subgraph  $\mathcal{G}_j^i$ , i.e.,  $\mathbb{V}_j^i = \{k \mid \mathcal{L}_k^i \in V_j^i\}$ . For the first layer we see  $\mathbb{V}_j^1 = V_j^1$  for all  $j = 1, \dots, S^1$ . For each layer  $i$  associate to each node  $k$  a *total neighbor set* that is given by  $\mathcal{N}_k^i = \mathbb{V}_j^i$  for  $\mathcal{L}_k^i \in V_j^i$ , where this set represents node  $k$  and all nodes that are connected to it across layers 1 through  $i$ . We let  $\mathcal{N}_k^0 = k$  and note that if node  $k$  is in layer  $i$  the nodes that “follow” it are those in the set  $\mathcal{N}_k^{i-1} \setminus k$ . Since all the subgraphs of any layer  $i$  are disjoint, then for any nodes  $k$  and  $j$  if there is any overlap in their total neighbor sets,  $\mathcal{N}_k^i \cap \mathcal{N}_j^i \neq \emptyset$ , the sets must be identical,  $\mathcal{N}_k^i = \mathcal{N}_j^i$ , and their leader nodes must be in the same subgraph of layer  $i$ . From this we can see that every total neighbor set  $\mathcal{N}_k^i$  is repeated  $|\mathcal{N}_k^i|$  times and there are a total of  $S^i$  unique neighbor sets for layer  $i$ . The union of the total neighbor sets at any layer equals  $V$ . We also have the following relationship

$$\mathcal{N}_k^{i+1} = \bigcup_{m \in V_j^{i+1}} \mathcal{N}_m^i, \text{ for } \mathcal{L}_k^{i+1} \in V_j^{i+1}. \quad (2)$$

Note in the final layer  $\mathbb{V}_1^M = \mathcal{N}_k^M = V$  for all  $k$ , meaning both the total node set and every total neighbor set in the final layer is the set containing all nodes of the graph. Now we refer to Fig. 1 to see the total node sets for layer 2 are  $\mathbb{V}_1^2 = \{1, 2, 3, 4, 5\}$  and  $\mathbb{V}_2^2 = \{6, 7\}$ . Focusing on node 2, the total neighbor sets are  $\mathcal{N}_2^1 = \{1, 2\}$ ,  $\mathcal{N}_2^2 = \{1, 2, 3, 4, 5\}$  and  $\mathcal{N}_2^3 = \{1, \dots, 7\}$ .

### 2.3 Continuous-Time Consensus

Denote the state of node  $i$  as  $x_i$ . The standard consensus algorithm consists of each agent's dynamics evolving according to

$$\dot{x}_i = \sum_{j=1}^N a_{ij} (x_j - x_i), \quad i = 1, \dots, N, \quad (3)$$

with  $a_{ij}$  defined by the adjacency matrix as given above. If we let the vector  $x = [x_1, \dots, x_N]^T$  then we can write the consensus dynamics of the group as

$$\dot{x} = -Lx, \quad (4)$$

where  $L$  is the graph Laplacian as defined above. Consensus is reached when all the nodal values are the same,  $x_1 = x_2 = \dots = x_N$ . The goal is average consensus, where the consensus value is the average of the initial conditions

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i(0). \text{ Let } \bar{\mathbf{x}} = \mathbf{1}_N^T \bar{x}, \text{ with } \mathbf{1}_N \text{ the row ones}$$

vector with dimension  $N$ , be the vector version of the scalar quantity  $\bar{x}$ . The consensus error is then defined as

$$e(t) = x(t) - \bar{\mathbf{x}}. \quad (5)$$

The standard consensus algorithm and certain variants of it have been widely studied over the years. Much focus has been given to requirements on the topology of

the network, and hence the Laplacian, for consensus to be achieved. Likewise the performance of the consensus algorithm, i.e., the time to reach consensus, is governed by the Laplacian. More specifically, following Olfati-Saber et al. (2007), with a connected graph and  $L$  symmetric (all links are bidirectional) all the eigenvalues lie on the real axis in the right half plane. The error convergence is then bounded by the second smallest eigenvalue  $\lambda_2$  of  $L$  according to

$$\|e(t)\|_2 \leq e^{-\lambda_2 t} \|e(0)\|_2, \quad (6)$$

so the consensus error goes to 0 as  $t \rightarrow \infty$ . Clearly larger values of  $\lambda_2$  cause the upper bound of the error to converge faster.

For use with the hierarchical scheme we introduce the term *starting value* to differentiate from initial conditions. The initial conditions  $x(0)$  are the values of the nodes at the very beginning of the consensus algorithm, i.e., at the lowest layer of the hierarchy. The starting values are the node values at the start of a new layer. Denote the start time of layer  $i$  as  $t_{i-1}^+$  and the end time as  $t_i^-$ , and for layer 1 we assume  $t_0^+ = 0$ . The starting values for layer  $i$  are thus  $x(t_{i-1}^+)$ . We then define the *subgraph average* as the average of the starting values of the subgraph nodes,

$$\bar{S}_j^i = \text{average}(\{x_k(t_{i-1}^+) \mid k \in V_j^i\}), \quad (7)$$

and the nodal subgraph error is

$$\tilde{e}_k(t) = \begin{cases} x_k(t) - \bar{S}_j^i & \text{for } k \in V_j^i \\ 0 & \text{if } k \notin V_j^i \end{cases} \quad (8)$$

for  $t \in [t_{i-1}^+, t_i^-]$ . The consensus error is still relative to the average of the initial conditions as in Eqn. (5).

In the next section we develop a variant of the standard consensus algorithm, applicable to graphs that can be hierarchically decomposed, to speed up the convergence time. It will take advantage of larger  $\lambda_2$  values for the smaller subgraphs compared to the  $\lambda_2$  of the full graph. For example, the full graph in Fig. 1 has  $\lambda_2 = 0.586$  while the smallest  $\lambda_2$  of all the subgraphs of the hierarchical decomposition is 1.

## 3. HIERARCHICAL CONSENSUS ALGORITHM DESCRIPTION AND ANALYSIS

In this section we will introduce the hierarchical consensus algorithm, then provide an analysis deriving a bound on the consensus error under this scheme and end with a discussion of the key features of the algorithm.

### 3.1 Hierarchical Consensus Algorithm

As opposed to the standard single-stage consensus algorithm that utilizes the full communication topology, the new consensus scheme aims to speed up the convergence by exploiting the hierarchical decomposition of the graph. The scheme consists of “disconnecting” the hierarchical graph into layers consisting of smaller disconnected subgraphs. Consensus is run within the subgraphs while moving up the hierarchy. The scheme does not introduce additional links to the topology, but may not utilize all available links depending on the decomposition.

The scheme works by running a consensus dynamics like Eqn. (3) within each subgraph, starting with the subgraphs in the lowest layer. The subgraphs will start converging towards the average of their starting values at that layer. The layer is complete when the nodes within that layer have all converged to within a specified tolerance  $\epsilon_s$  of their respective subgraph average, that is  $\|\tilde{e}\|_\infty \leq \epsilon_s$ . As we show later this stopping condition can be assured by enforcing a minimum layer time. After this stopping criterion is met the algorithm moves to the next higher layer of the hierarchy, repeating until the final layer is reached. The flow diagram for this scheme is shown in Fig. 2.

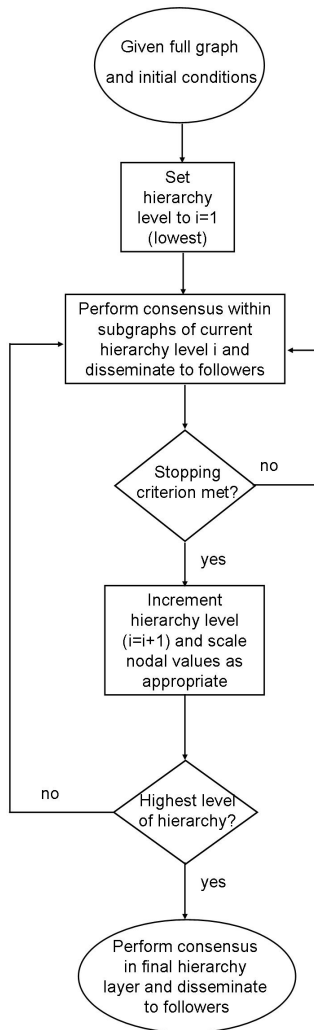


Fig. 2. Flow diagram for the hierarchical consensus scheme.

Subgraphs of consecutive layers are connected by the nodes that are present in both layers. These allow the information to flow between the levels in the hierarchy. As the algorithm moves to higher levels, these nodes must disseminate information to their follower nodes. We assume the leaders are able to relay their values down to their followers, still utilizing the communication topology but in a different mode than before. This allows all nodes to instantaneously assume the value of their leader node, i.e., for every layer  $i$  we get  $x_k(t) = x_{\mathcal{L}_k^i}(t)$ .

In designing the hierarchical scheme we want the consensus value that every subgraph converges towards to be equal to the average of the initial conditions of that subgraph's total node set. This means we want  $\bar{S}_j^i$  as defined in Eqn. (7) to be equal to average  $(\{x_k(0) \mid k \in \mathbb{V}_j^i\})$ . To assure this we simply rescale the starting values of the nodes within the subgraph. Combining this rescaling with the relay option that sets all follower node values to their leader's value, we have upon starting a new layer  $i$

$$x_k(t_{i-1}^+) = \alpha_{\mathcal{L}_k^i}^i x_{\mathcal{L}_k^i}(t_{i-1}^-) \quad (9)$$

$$\alpha_{\mathcal{L}_k^i}^i = \frac{|\mathbb{V}_j^i|^{-1}}{|\mathbb{V}_j^i|} \left| \mathcal{N}_{\mathcal{L}_k^i}^{i-1} \right|$$

for  $\mathcal{L}_k^i \in \mathbb{V}_j^i$  and we let  $\alpha_k^1 = 1$  for all  $k$ . It is important to note that in order for the nodes to compute  $\alpha_k^i$  they only need to know the local topology of the hierarchical graph, that is the number of nodes in their subgraph and the total number of followers for each of those nodes. The scaling factors are independent of the actual values of the nodes. If all nodes in a subgraph have the same number of total followers, the corresponding scale value is 1, i.e., if  $|\mathcal{N}_{\mathcal{L}_k^i}^{i-1}| = |\mathcal{N}_{\mathcal{L}_l^i}^{i-1}|$  for all  $k, l \in \mathbb{V}_j^i$  then it is easy to see  $|\mathbb{V}_j^i| = |\mathbb{V}_j^i| |\mathcal{N}_{\mathcal{L}_k^i}^{i-1}|$  and  $\alpha_k^i = 1$ . Self-similar graphs, such as that in Fig. 3, have this property.

### 3.2 Analysis

We want to bound the consensus error for the hierarchical scheme and compare this with the bound of the standard single-stage consensus algorithm utilizing the full graph topology that was given in Eqn. (6). As mentioned above a layer is considered completed as soon as  $\|\tilde{e}\|_\infty \leq \epsilon_s$ . We will first derive a bound for the consensus error of the hierarchical scheme, which will depend on the value of  $\epsilon_s$  and then we will show how to guarantee this tolerance can be met.

We will need to assume a bound on the initial conditions  $\|x(0)\|_\infty \leq \beta$ , to aid in the analysis of the rescaling. Next, define for each layer  $i$  the vectors  $\bar{\mathcal{N}}^i = [\bar{\mathcal{N}}_1^i, \bar{\mathcal{N}}_2^i, \dots, \bar{\mathcal{N}}_N^i]^T$  and  $\tilde{\mathcal{N}}^i = [\tilde{\mathcal{N}}_1^i, \tilde{\mathcal{N}}_2^i, \dots, \tilde{\mathcal{N}}_N^i]^T$  with

$$\bar{\mathcal{N}}_k^i = \text{average}(\{x_m(0) \mid m \in \mathcal{N}_k^i\}) \quad (10)$$

$$\tilde{\mathcal{N}}_k^i = \text{average}(\{x_m(t_{i-1}^+) \mid m \in \mathbb{V}_j^i\}) \quad (11)$$

for  $\mathcal{L}_k^i \in \mathbb{V}_j^i$ . Thus  $\bar{\mathcal{N}}_k^i$  represents the average of the initial conditions of all nodes connected to node  $k$  from layer 1 through layer  $i$  and  $\tilde{\mathcal{N}}_k^i$  represents the average of the starting values at layer  $i$  of all nodes in the subgraph containing node  $k$ 's leader node.

To bound the consensus error we start by noting that for each layer  $i$  we can write

$$\begin{aligned} \|e(t)\|_2 &= \|x(t) - \bar{x}\|_2 \\ &= \left\| x(t) + \bar{\mathcal{N}}^i - \bar{\mathcal{N}}^i + \tilde{\mathcal{N}}^i - \tilde{\mathcal{N}}^i - \bar{x} \right\|_2 \\ &\leq \left\| x(t) - \tilde{\mathcal{N}}^i \right\|_2 + \left\| \tilde{\mathcal{N}}^i - \bar{\mathcal{N}}^i \right\|_2 + \left\| \bar{\mathcal{N}}^i - \bar{x} \right\|_2 \end{aligned} \quad (12)$$

for any  $t \in [t_{i-1}^+, t_i^-]$ . The first term on the right hand side of Eqn. (12) represents the error of the nodes with respect to their subgraph averages. The second term is the difference between the total neighbor subgraph averages and the total neighbor initial condition averages. The final term represents the difference between the average of the initial conditions of the total neighbor sets for layer  $i$  and the initial conditions of all the nodes. We will now provide bounds for each of the terms on the right hand side of Eqn. (12).

*Lemma 2.* Using the hierarchy scheme with stopping tolerance  $\epsilon_s$  then for any layer  $i$  the difference between the total neighbor subgraph averages and the total neighbor initial condition averages can be bounded by

$$\left\| \tilde{\mathcal{N}}^i - \bar{\mathcal{N}}^i \right\|_2 \leq (i-1)\epsilon_s \sqrt{N}. \quad (13)$$

**Proof.** See the proof of Lemma 5.2 in Epstein (2007). ■

*Lemma 3.* For any layer  $i < M$  the difference between the average of the initial conditions of the total neighbor sets and the initial conditions of all the nodes is bounded by

$$\left\| \bar{\mathcal{N}}^i - \bar{\mathbf{x}} \right\|_2 \leq \|e(0)\|_2. \quad (14)$$

For the final layer we have by definition

$$\left\| \bar{\mathcal{N}}^M - \bar{\mathbf{x}} \right\|_2 = 0. \quad (15)$$

**Proof.** See the proof of Lemma 5.3 in Epstein (2007). ■

With these lemmas we have bound the last two terms on the right hand side of Eqn. (12). Now we seek to bound the first term. To do so we will use the following results which will bound the consensus error at the end and beginning of each layer.

*Lemma 4.* Given the stopping criterion  $\|\tilde{e}\|_\infty \leq \epsilon_s$ , at the end of every hierarchy level  $i < M$  the consensus error is bounded according to

$$\|e(t_i^-)\|_2 \leq \|e(0)\|_2 + i\epsilon_s \sqrt{N}. \quad (16)$$

As the final layer  $M$  does not terminate, i.e.,  $t_M^- \rightarrow \infty$ , we get the steady state consensus error for the hierarchy scheme is bounded by

$$\lim_{t \rightarrow \infty} \|e(t)\|_2 \leq (M-1)\epsilon_s \sqrt{N}. \quad (17)$$

**Proof.** See the proof of Lemma 5.4 in Epstein (2007). ■

*Lemma 5.* The consensus error at the beginning of every layer  $i$  can be bound by

$$\|e(t_{i-1}^+)\|_2 \leq \|e(0)\|_2 + \epsilon_s \sqrt{N} (2\|\tilde{\alpha}^i\|_\infty + i - 1) + \|\tilde{\alpha}^i - 1\|_\infty \sqrt{N}\beta, \quad (18)$$

where  $\tilde{\alpha}^i = [\alpha_k^i]_{k \in \mathcal{V}^i}$ .

**Proof.** See the proof of Lemma 5.5 in Epstein (2007). ■

Using the bound on the consensus error at the start of each layer we now provide a bound on the subgraph error at the start of each layer.

*Lemma 6.* At the start of any layer  $i$  the subgraph error  $\tilde{e}$  as defined in Eqn. (8) is bounded by

$$\|\tilde{e}(t_{i-1}^+)\|_2 \leq \tilde{e}_0^i$$

where

$$\tilde{e}_0^i = b^i \|e(0)\|_2 + \|\tilde{\alpha}^i - 1\|_\infty \beta \sqrt{N} + 2\epsilon_s \sqrt{N} (\|\tilde{\alpha}^i\|_\infty + i - 1) \quad (19)$$

with  $b^i = 2$  if  $i < M$  and  $b^M = 1$ .

**Proof.** See the proof of Lemma 5.6 in Epstein (2007). ■

We are now ready to determine a bound for the first term on the right hand side of Eqn. (12) as given below.

*Lemma 7.* Using the hierarchy scheme the subgraph error is bounded by

$$\|x(t) - \tilde{\mathcal{N}}^i\|_2 \leq \tilde{e}_0^i \left[ \sum_{j=1}^{S^i} \left( \max_{k \in \mathcal{V}_j^i} |\mathcal{N}_k^{i-1}| \right) e^{-2\lambda_2^{i,j}(t-t_{i-1}^+)} \right]^{\frac{1}{2}} \quad (20)$$

for  $t \in [t_{i-1}^+, t_i^-]$  with  $\tilde{e}_0^i$  as given in Eqn. (19) and  $\lambda_2^{i,j}$  representing the second smallest eigenvalue of the subgraph  $\mathcal{G}_j^i$ .

**Proof.** See the proof of Lemma 5.7 in Epstein (2007). ■

Now that we have provided bounds for all the terms on the right hand side of Eqn. (12) we are in a position to bound the consensus error.

*Theorem 8.* Using the hierarchical consensus scheme with layer stopping tolerance  $\|\tilde{e}(t_i^-)\|_\infty \leq \epsilon_s$ , the consensus error during each layer  $i$  can be bounded according to

$$\|e(t)\|_2 \leq \tilde{e}_0^i \left[ \sum_{j=1}^{S^i} \left( \max_{k \in \mathcal{V}_j^i} |\mathcal{N}_k^{i-1}| \right) e^{-2\lambda_2^{i,j}(t-t_{i-1}^+)} \right]^{\frac{1}{2}} + (i-1)\epsilon_s \sqrt{N} + (b^i - 1) \|e(0)\|_2 \quad (21)$$

with  $\tilde{e}_0^i$  as in Eqn. (19),  $b^i = 2$  if  $i < M$  and  $b^M = 1$ , and  $\lambda_2^{i,j}$  representing the second smallest eigenvalue of the subgraph  $\mathcal{G}_j^i$ .

**Proof.** This is a direct consequence of Eqn. (12) with Lemmas 2, 3 and 7. ■

As mentioned in the description of the hierarchical algorithm, each layer  $i < M$  terminates when  $\|\tilde{e}\|_\infty \leq \epsilon_s$ . Up to this point we simply assumed this stopping criterion was met before moving to the next layer. With the analysis above and assuming we know a bound on the initial error, we can assure the criterion is met by keeping the time within each layer to be larger than a certain value as show below.

*Lemma 9.* To assure the stopping criterion  $\|\tilde{e}(t_i^-)\|_\infty \leq \epsilon_s$  is met for each layer  $i < M$  of the hierarchical scheme we simply make sure the layer time (time spent in the layer), which is given by

$$T_i = t_i^- - t_{i-1}^+, \quad (22)$$

is large enough to satisfy the following inequality

$$\tilde{e}_0^i \sum_{j=1}^{S^i} e^{-\lambda_2^{i,j} T_i} \leq \epsilon_s, \quad j = 1, \dots, S^i \quad (23)$$

for every layer  $i$ .

**Proof.** From Lemma 6 we know the subgraph error is bounded by  $\tilde{e}_0^i$  at the start of layer  $i$ . Using the fact that

within any layer the subgraph error from each subgraph must be no greater than the total subgraph error from all those in the layer we can bound the starting subgraph error for each subgraph by  $\tilde{e}_0^i$  as well. Since each subgraph is connected the error of subgraph  $\mathcal{G}_j^i$  will converge at a rate no slower than  $\lambda_2^{i,j}$ . The norm of the total subgraph error for any layer is bound by the sum of the norm of the individual subgraph errors in that layer, hence we have

$$\|\tilde{e}(t)\|_2 \leq \tilde{e}_0^i \sum_{j=1}^{S^i} e^{-\lambda_2^{i,j}(t-t_{i-1}^+)} \quad (24)$$

for  $t \in [t_{i-1}^+, t_i^-]$ . From this we see that if Eqn. (23) is satisfied then  $\|\tilde{e}\|_2 \leq \epsilon_s \Rightarrow \|\tilde{e}(t_i^-)\|_\infty \leq \epsilon_s$ . ■

With the definition of layer times in Eqn. (22) and since  $t_0^+ = 0$  we see that

$$t_i = \sum_{j=1}^i T_j. \quad (25)$$

With the bound for  $\|e(t)\|_2$  determined we can compute when this bound will be lower than that of the standard single-stage consensus. In the hierarchical scheme only after the final layer begins will all the nodes be hierarchically connected. Therefore we look at the consensus error of the hierarchical scheme during the final layer and compare with the standard algorithm.

*Theorem 10.* The bound on the norm of the consensus error will be smaller for the hierarchical scheme than the standard single-stage consensus algorithm for all time  $t_{M-1}^+ + T$  with  $0 \leq T_* < T < T^* < \infty$  and satisfying the following inequality

$$e^{-\lambda_2 T} e^{-\lambda_2 t_{M-1}^+} \|e(0)\|_2 - e^{-\lambda_2^M T} \tilde{e}_0^M \left( \max_{k \in \mathcal{V}^M} |\mathcal{N}_k^{M-1}| \right) \geq (M-1)\epsilon_s \sqrt{N} \quad (26)$$

with  $\lambda_2^M$  the second smallest eigenvalue of the Laplacian of  $\mathcal{G}_1^M$ . That is to say the bound on the hierarchical scheme will be smaller than the bound of the single-stage consensus during the finite time interval  $t \in (t_{M-1}^+ + T_*, t_{M-1}^+ + T^*)$ . Note  $T_* < T < T^* < \infty$  is equivalent to Eqn. (26) being satisfied.

**Proof.** This is a straightforward comparison of the bound on the consensus error in the hierarchical scheme using Eqn. (21) with the final layer  $i = M$  and the bound on the standard consensus algorithm from Eqn. (6). ■

### 3.3 Discussion

We have derived a bound on the hierarchy consensus error and compared it to the standard single-stage consensus algorithm culminating with Theorem 10. The key factors that determine when the condition is met are: the stopping tolerance,  $\epsilon_s$ ; the layer times,  $T_i$ ; the speed of convergence of the full graph  $\lambda_2$  and subgraphs  $\lambda_2^{i,j}$ ; the initial error,  $\|e(0)\|_2$ , and the tightness of the known bound on this.

Notice that the term on the right hand side of the inequality in Eqn. (26) is the bound on the steady state consensus error of the hierarchy scheme which depends on the number of nodes, number of layers and stopping

criterion tolerance  $\epsilon_s$ . In fact  $\epsilon_s$  will play more of a role to follow. The first term on the left hand side of Eqn. (26) is the consensus error bound for the standard scheme. Notice that at the time of the start of the final layer of the hierarchy,  $t = t_{M-1}^+$  and  $T = 0$ , the second term will be greater than  $\|e(0)\|_2$ , and since  $t_{M-1}^+ > 0$  the condition of Eqn. (26) can not be satisfied at  $T = 0$ . As  $T$  increases, since we assume  $\lambda_2^M > \lambda_2$ , the second term goes to zero faster than the first. Thus there will be a time  $T = T_*$  such that the inequality is satisfied, and the larger the difference between the eigenvalues the faster the inequality is satisfied.

The inequality can also be satisfied faster the sooner the final layer starts, i.e., smaller values of  $t_{M-1}^+$ . Of course for the final layer to start all previous layers must have satisfied the stopping criterion. Thus we see the layer times  $T_i$  should be chosen as small as possible to satisfy Eqn. (23). Key in this is how conservative of a bound we can assume on  $\|e(0)\|_2$ . Since most likely the starting norm error is not known we instead use a bound for the value in Eqn. (23) as well as in evaluating the inequality in Eqn. (26). If the bound is conservative then the layer times will be longer than necessary and the inequality will turn out to be conservative.

The stopping tolerance  $\epsilon_s$  also determines the layer times, so at first one might decide not to make this value too small. Since the steady state error of the hierarchy scheme is proportional to  $\epsilon_s$ , and we want to have a small steady state error, there is a trade-off. Note that as  $T$  continues to increase the inequality in Eqn. (26) will once again fail to be satisfied after a certain point  $T = T^* > T_*$ . This occurs when the standard consensus error bound crosses the (nearly) steady state value of the hierarchical scheme consensus error bound. Finally it should be noted that using the analysis above one could chose a desired level of consensus error to be reached and could chose the appropriate parameters so that the hierarchical scheme reach this bound first.

## 4. EXAMPLE

To help illustrate the analysis above we present a simulation example below. Consider the 27 node hierarchical graph of Fig. 3. The overall graph has second smallest eigenvalue given by  $\lambda_2 = 0.1625$ , while all the subgraphs of the hierarchical decomposition are fully connected graphs with three nodes and thus have  $\lambda_2^{i,j} = 3$ . This means the smaller subgraphs converge nearly 18.5 times faster than the full graph. We picked the initial conditions uniformly distributed in the interval  $x_k(0) \in [0, \beta]$  with  $\beta = 1000$ . This means  $0 \leq \bar{x} \leq 1000$  so the initial error can be bound by  $\|e(0)\|_2 \leq 1000\sqrt{27}$ .

Simulation results are shown in Fig. 4 with stopping tolerance  $\epsilon_s = 10^{-5}$ . The actual initial error for this simulation was  $\|e(0)\|_2 \approx 281\sqrt{27}$ , yet the bounds and the layer times were computed using  $\|e(0)\|_2 \leq 1000\sqrt{27}$  as this is the assumed knowledge at design time. Notice the hierarchy bound is first lower than the full graph bound at roughly 16 seconds and stays below until 109 seconds where the error bound is around  $10^{-4}$ . The actual error in the hierarchy scheme goes below the full graph

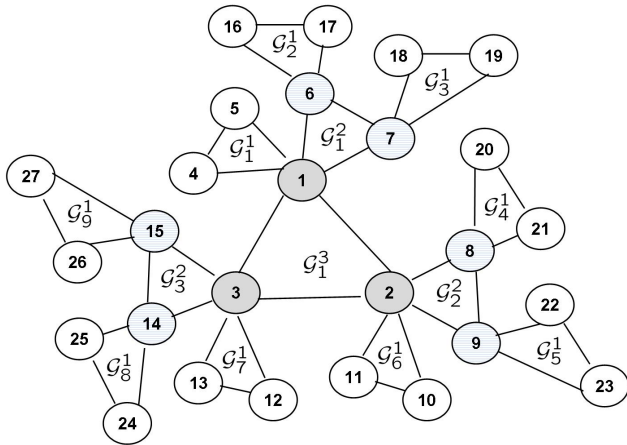


Fig. 3. Graph with hierarchical decomposition.

scheme at roughly 15.3 seconds and stays below until after 150 seconds, thus showing the actual performance is even better than the bounds indicate. In Fig. 5 we plot the upper-bound of  $\|e(t)\|_2$  for the hierarchy scheme for different values of  $\epsilon_s$ . Notice how the time at which the hierarchy bound is first below the standard scheme bound increases and the steady state value of the hierarchical scheme decreases as  $\epsilon_s$  decreases.

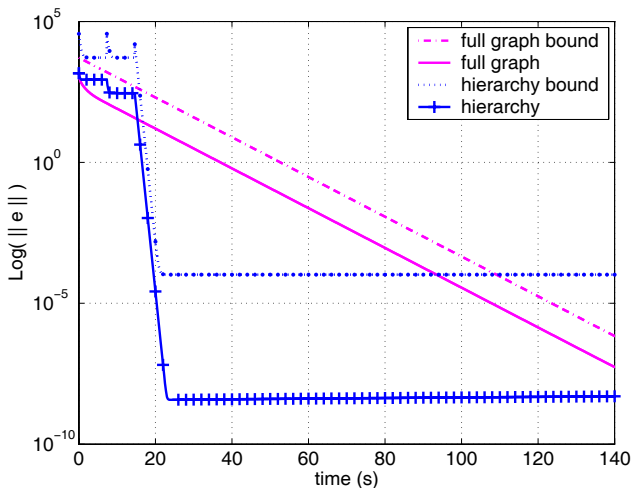


Fig. 4. Error results for  $\epsilon_s = 10^{-5}$  and  $\|e(0)\|_2 \approx 281\sqrt{27}$ .

## 5. CONCLUSIONS AND FUTURE WORK

In this work we introduced the hierarchical consensus scheme. We showed that by allowing subgraphs of a larger connected graph to converge separately and then joining the leaders of the subgraph to the larger graph the overall time to consensus can be reduced. We showed the key parameters that determine the performance of the scheme and used examples to show the effectiveness.

There are many avenues to pursue in the future. We only analyzed the broadcast feature for disseminating the information from leader nodes to followers. Naturally one could analyze the case where the follower nodes still run consensus treating the leaders' information as inputs to their layer. The algorithm could be adapted to the case

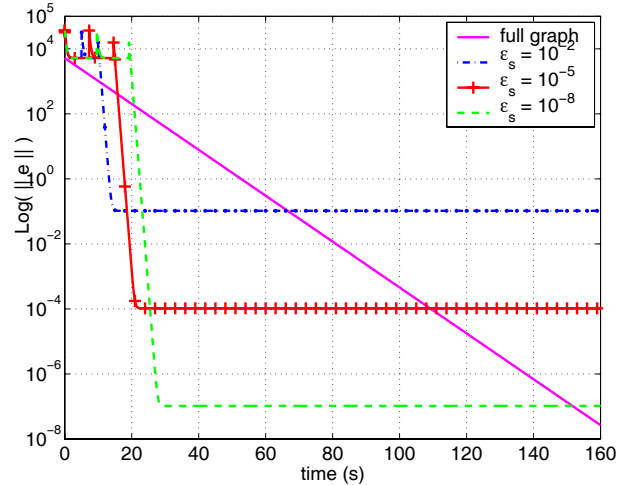


Fig. 5. Error bounds for different values of  $\epsilon_s$ .

with non-static input values at the nodes will be very interesting. In this paper we ignored the issue of non-unique hierarchical decompositions for a given graph. Determining a way to optimally choose how to hierarchically decompose a graph into layers and subgraphs would be very valuable to make the algorithm more applicable. We could also include various network effects such as delayed and dropped information in the analysis or consider discrete time consensus.

## REFERENCES

- M. Epstein. *Managing Information in Networked and Multi-Agent Control Systems*. PhD thesis, California Institute of Technology, 2007.
- J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.
- C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer-Verlag, 2001.
- R. Olfati-Saber. Ultrafast consensus in small-world networks. In *American Control Conference*, 2005.
- R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–2533, 2004.
- R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- W. Ren and R. W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 2005.
- D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Distributed sensor fusion using dynamic consensus. In *IFAC World Congress*, 2005.
- W. Xi, X. Tan, and J. S. Baras. A stochastic algorithm for self-organization of autonomous swarms. In *Conference on Decision and Control*, 2005.
- L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Controls Letters*, 53(1):65–78, 2004.
- P. Yang, R. Freeman, and K. M. Lynch. Optimal information propagation in sensor networks. In *International Conference on Robotics and Automation*, 2006.