2010 American Control Conference
Marriott Waterfront, Baltimore, MD, USA
June 30-July 02, 2010

On the Time Complexity of Information Dissemination via Linear Iterative Strategies

Shreyas Sundaram and Christoforos N. Hadjicostis

Abstract— **Given an arbitrary network of interconnected nodes, each with an initial value, we study the number of time-steps required for some (or all) of the nodes to gather all of the initial values via a linear iterative strategy. At each time-step in this strategy, each node in the network transmits a weighted linear combination of its previous transmission and the most recent transmissions of its neighbors. We show that for almost any choice of real-valued weights in the linear iteration (i.e., for all but a set of measure zero), the number of time-steps required for any node to accumulate all of the initial values is upper-bounded by the size of the largest tree in a certain subgraph of the network; we use this fact to show that the linear iterative strategy is time-optimal for information dissemination in certain networks. In the process of deriving our results, we also obtain a characterization of the observability index for a class of linear structured systems.**

I. Introduction

A key requirement in distributed systems and networks is to disseminate information from some or all of the nodes in the network to the other nodes. Various algorithms to achieve this have been developed by the computer science, communication, and control communities over the past few decades [1], [2], [3], [4]. A particular strategy that has attracted significant attention in the control systems community is that of *linear iterations*; in this strategy, each node in the network repeatedly updates its value to be a weighted linear combination of its own value and those of its neighbors (e.g., see [5] and the references therein). These works have revealed that if the network topology satisfies certain conditions, the weights for the linear iteration can be chosen so that all of the nodes asymptotically converge to the same value (usually a weighted linear combination of the initial values of the nodes); in this case, the nodes are said to reach *asymptotic consensus*. Recently, it was shown in [6] that this linear iterative strategy can actually be applied to the more general data aggregation problem, allowing any node in networks with time-invariant topologies to obtain all of the initial node values in a *finite* number of time-steps – this number was shown to be upper-bounded by $N - \deg_i$ for node x_i in the network, where N is the total number of nodes, and $\deg_i$ is the number of neighbors of node x_i. This bound was derived in [6] via a direct application of techniques from observability theory.

Linear iterative strategies can also be viewed as a form of *network coding*, a topic that has been extensively studied by the communications community over the past few years [3]. The works [7] and [8] studied the average number of time-steps required by a node to gather all of the data with a *gossip*-based algorithm, whereby every node in the network periodically sends a random linear combination of the messages that it has previously received to a randomly chosen neighbor. The paper [9] studied a network with a source node connected to multiple receivers via unreliable links, and showed that allowing the source node to send random linear combinations of the source packets allows the receivers to recover all of the packets in fewer time-steps than having the source node send a particular packet at each time-step.

There is a great deal of analysis of the time-complexity for other algorithms for information dissemination under varying assumptions on the underlying topology and communications modality [10]. Tree-based schemes, in particular, have received a great deal of attention for data aggregation, partly due to their simplicity of analysis and implementation. For example, [11], [12] consider the use of trees to either broadcast or aggregate data, under the assumption that the network is operating in a wireless environment where *collisions* are possible (i.e., only one neighbor of any given node is allowed to transmit at any given time-step). Optimal scheduling of transmissions in such cases is known to be a NP-hard problem [12], and tree-based schemes are shown to offer good approximations to the optimal solution.

In contrast to the above works, this paper studies the time-complexity of aggregating information in networks where multiple nodes are allowed to simultaneously exchange information, and when collisions are not an issue (e.g., as would be the case in wireless networks operating under a multiple-access protocol). We perform a careful analysis of linear iterative strategies for such networks, and prove that linear strategies are at least as fast as tree-based schemes for aggregating information (strictly faster in some cases). Furthermore, for certain networks, we show that no other strategy can outperform the linear iterative strategy in terms of the number of time-steps required to accumulate all of the initial values. In the process of obtaining this result,

This material is based upon work supported in part by the National Science Foundation under NSF ITR Award 0426831 and NSF CNS Award 0834409. The research leading to these results has also received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreements INFSO-ICT-223844 and PIRG02-GA-2007-224877. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NSF or EC.

S. Sundaram is with the GRASP Laboratory and the Department of Electrical and Systems Engineering, University of Pennsylvania, USA. E-mail: ssund@seas.upenn.edu. C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, and also with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. E-mail: chadjic@ucy.ac.cy.

we derive an upper bound on the observability index of *structured linear systems*, which is of independent interest.

II. Notation and Background on Graph Theory

We use $\mathbf{e}_{i,l}$ to denote the column vector of length l with a "1" in its i-th position and "0" elsewhere. The symbol $\mathbf{1}_l$ denotes the column vector of length l with all entries equal to "1", and $\mathbf{I}_N$ denotes the $N \times N$ identity matrix. The transpose of matrix $\mathbf{A}$ is denoted by $\mathbf{A}'$. We denote the cardinality of a set $\mathcal{S}$ by $|\mathcal{S}|$, and for two sets $\mathcal{A}$ and $\mathcal{B}$, we use $\mathcal{A} \setminus \mathcal{B}$ to indicate all elements of $\mathcal{A}$ that are not in $\mathcal{B}$.

A graph is an ordered pair $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where $\mathcal{X} = \{x_1, \ldots, x_N\}$ is a set of vertices, and $\mathcal{E}$ is a set of ordered pairs of different vertices, called directed edges. If $(x_i, x_j) \in \mathcal{E} \Leftrightarrow (x_j, x_i) \in \mathcal{E}$, the graph is said to be undirected. The nodes in the set $\mathcal{N}_i = \{x_j | (x_j, x_i) \in \mathcal{E}\}$ are said to be neighbors of node x_i, and the in-degree of node x_i is denoted by $\deg_i = |\mathcal{N}_i|$. A *subgraph* of $\mathcal{G}$ is a graph $\mathcal{H} = \{\bar{\mathcal{X}}, \bar{\mathcal{E}}\}$, with $\bar{\mathcal{X}} \subseteq \mathcal{X}$ and $\bar{\mathcal{E}} \subseteq \mathcal{E}$ (where all edges in $\bar{\mathcal{E}}$ are between vertices in $\bar{\mathcal{X}}$).

A *path* P from vertex x_{i_0} to vertex x_{i_t} is a sequence of vertices $x_{i_0} x_{i_1} \cdots x_{i_t}$ such that $(x_{i_j}, x_{i_{j+1}}) \in \mathcal{E}$ for $0 \leq j \leq t-1$, and no vertex appears more than once in the sequence. The nonnegative integer t is called the length of the path. The *distance* between node x_j and node x_i in a graph is the length of the shortest path between node x_j and node x_i in the graph. The *eccentricity* of node x_i is the distance from the node that is farthest away from x_i in the graph. A path is called a *cycle* if its start vertex and end vertex are the same, and no other vertex appears more than once in the path.

A graph is called *acyclic* if it contains no cycles. A graph $\mathcal{G}$ is a *spanning tree rooted at* x_i if it is an acyclic graph where every node in the graph has a path to x_i, and every node except x_i has an outgoing edge to exactly one node. Similarly, a graph is a *spanning forest rooted at* $\mathcal{R} = \{x_{i_1}, x_{i_2}, \ldots, x_{i_p}\}$ if it is a disjoint union of a set of trees, each of which is rooted at one of the vertices in $\mathcal{R}$. An example of a spanning tree rooted at x_1 is shown in Fig. 1.a, and an example of a spanning forest rooted at $\mathcal{R} = \{x_1, x_2, x_3\}$ is shown in Fig. 1.b. A graph is *strongly-connected* if there is a path from every node to every other node. Further background on graph theory can be found in standard texts, such as [13].

III. Motivating Example

Consider again the network $\mathcal{G}$ shown in Fig. 1.a; each node in this network possesses a single value from a field[1] $\mathbb{F}$, and node x_1 needs to obtain all of these values. Each node in the network is allowed to transmit a single value from the field $\mathbb{F}$ at each time-step. Under these conditions, note that x_1 can obtain at most one new value at each time-step from each of its neighbors. Since the values of both x_2 and x_3 have to pass through x_2, it will take exactly two time-steps for x_1 to receive those values. In parallel, x_1 is also receiving values from x_4 and x_7; it will take exactly three time-steps for x_1 to receive the values of x_4, x_5, x_6, and exactly four time-steps to receive the values of x_7, x_8, x_9 and x_{10}. Thus, x_1 can receive all values in the network after exactly four time-steps. This example suggests that one bottleneck in the network is related to the number of values that must pass through any given neighbor of x_1. To state this in a form that will be easier for us to analyze, we remove certain edges from the network to form a spanning forest rooted at $\{x_1\} \cup \mathcal{N}_1$ (shown in Fig. 1.b). Each tree in this forest contains all of the values that must pass through a given neighbor of x_1. The largest tree in this forest has four nodes, which is equal to the number of time-steps required for x_1 to obtain all of the values.[2]

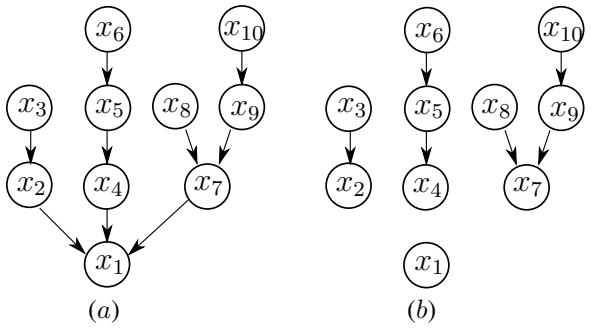

Fig. 1. (a) Spanning tree rooted at x_1. (b) Spanning forest rooted at $\{x_1, x_2, x_4, x_7\}$. Note that this spanning forest is also a subgraph of the graph in (a).

Now consider the network $\mathcal{G}$ shown in Fig. 2.a, which is no longer a simple spanning tree rooted at x_1. To determine the number of time-steps it will take for x_1 to obtain all of the values in this network, note that the forest shown in Fig. 1.b is a subgraph of $\mathcal{G}$, and thus it is definitely possible for x_1 to obtain the values of all nodes in four time-steps. However, one can actually do better by noting that $\mathcal{G}$ also contains the spanning forest rooted at $\{x_1\} \cup \mathcal{N}_1$ shown in Fig. 2.b, which only has three nodes in any tree. Thus, x_1 can receive the values of all nodes in three time-steps (e.g., by following the routing scheme specified by the spanning forest in Fig. 2.b); one cannot do any better in this network, since the eccentricity of node x_1 is three.

The above examples show that the amount of time required by a node to obtain all of the values is upper bounded by the size of the largest tree in a subgraph of the network that is a spanning forest rooted at that node and its neighbors. The bound becomes tighter if one can find the "best" such subgraph – this concept will play a recurring role in this

[1] In this paper, we will focus on the case where nodes are allowed to manipulate real numbers, and perform our analysis over the field of complex numbers. However, the analysis also carries over to finite fields after suitable modifications; this is useful when the nodes are only allowed to transmit a finite number of bits at each time-step (e.g., due to bandwidth constraints). The extension of linear iterative strategies to such cases is described in [14], but we will forego the details here in the interest of conciseness.

[2] Note that the analysis for this network holds regardless of the actual algorithm that is used for information dissemination (i.e., nodes can simply schedule and forward incoming values, or they can perform more complicated operations). However, in general networks, tree-based schemes represent a special case of scheduling and routing; each node sends its value toward the root along the path that exists between that node and the root.

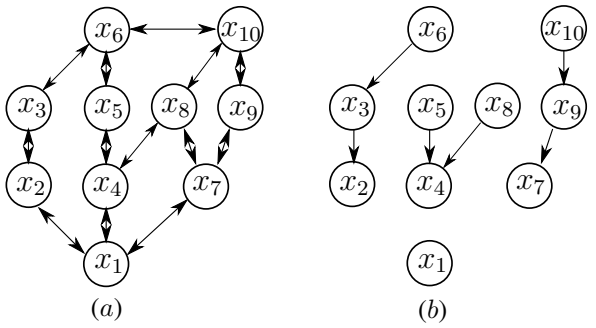

Fig. 2. (a) Node x_1 needs to receive the values of all of the nodes in this network. (b) A subgraph of the original network that is a spanning forest rooted at x_1, x_2, x_4 and x_7, with only three nodes in the largest tree.

paper, and so we will use the following definition throughout the paper.

Definition 1: Let $\mathcal{G}$ denote the (directed) graph of a fixed strongly-connected network, and for any set $\mathcal{R} \subset \mathcal{X}$, consider a subgraph $\mathcal{H}$ of $\mathcal{G}$ that is a spanning forest rooted at $\mathcal{R}$, with the property that the size of the largest tree in $\mathcal{H}$ is minimal over all possible spanning forests rooted at $\mathcal{R}$. We call $\mathcal{H}$ an *optimal spanning forest rooted at* $\mathcal{R}$. When $\mathcal{R} = \{x_i\} \cup \mathcal{N}_i$ for some node $x_i \in \mathcal{X}$, we simply say that $\mathcal{H}$ is an *optimal spanning forest for* x_i. □

Note that there are several challenges presented by the above analysis. First, given an arbitrary network, it is not clear how to efficiently find the optimal spanning forest for a given node. Second, even if the optimal forest could be found, the case where multiple nodes in the network have to receive all of the values would need a different analysis (since by removing edges to create an optimal forest for a certain node, we could be preventing some other node from receiving all of the values). Third, while the above tree-based characterization provides an *upper bound* on the number of time-steps required for any node to gather the data, it is unclear at this point in the narrative whether it is possible to do better (we will show later that we can, in fact, do better).

In this paper, we will demonstrate that linear iterative strategies provide a novel, simple and effective solution to these problems. Specifically, we will show that these strategies do not require any complicated analysis or manipulation of the network topology, and disseminate information at least as quickly as trees, simultaneously for all of the nodes in the network. This will reveal that linear iterative strategies are simple and powerful methods for disseminating information rapidly in networks.

IV. Problem Formulation

Consider a network modeled by the directed strongly-connected[3] graph $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where $\mathcal{X} = \{x_1, \ldots, x_N\}$ is the set of nodes and directed edge $(x_j, x_i) \in \mathcal{E}$ if node x_i can receive information directly from node x_j. Each node x_i has some initial value $x_i[0]$ that is potentially required by other nodes. We study a linear iterative strategy to disseminate these values through the network; specifically, at each time-step k, each node x_i updates its value as

$$x_i[k+1] = w_{ii}x_i[k] + \sum_{j \in \mathcal{N}_i} w_{ij}x_j[k],$$

where the w_{ij}'s are a set of weights. For ease of analysis, we aggregate the values of all nodes at time-step k into the vector $\mathbf{x}[k] = \begin{bmatrix} x_1[k] & x_2[k] & \cdots & x_N[k] \end{bmatrix}'$, so that

$$\mathbf{x}[k+1] = \mathbf{W}\mathbf{x}[k] \tag{1}$$

for $k = 0, 1, \ldots$, where the (i, j) entry of $\mathbf{W}$ is the weight w_{ij} if $x_j \in \mathcal{N}_i$, and zero otherwise. For the above strategy, we will demonstrate the following key result in this paper.

Theorem 1: Let $\mathcal{G}$ denote the graph of a fixed strongly-connected network, and for each $x_i \in \mathcal{X}$, consider a subgraph $\mathcal{H}_i$ that is an optimal spanning forest for x_i. Let D_i denote the size of the largest tree in $\mathcal{H}_i$. Then for almost any choice of real-valued weight matrix (with the constraint that $w_{ij} = 0$ if $j \notin \mathcal{N}_i$), each node x_i can obtain all of the initial values after running the linear iteration (1) for at most D_i time-steps. □

In the above theorem, the phrase "almost any" means that the set of weights for which the theorem does not hold has Lebesgue measure zero. Theorem 1 (which we will prove later in the paper) reveals that linear iterative strategies essentially bypass the problem of finding an optimal spanning forest in graphs – one can simply choose weights at random, and the linear iterative strategy will allow all nodes to simultaneously receive all of the values, and furthermore, each node x_i will do so in at most D_i time-steps (where D_i is the size of the largest tree in the optimal spanning forest for x_i).

Remark 1: A lower bound on the number of time-steps required by node x_i to obtain all of the data is given by the eccentricity of node x_i (since it takes one time-step for a value to propagate along an edge). □

V. Data Accumulation Via the Linear Iterative Strategy

Let $\mathbf{y}_i[k]$ denote the vector of outputs (node values) that node x_i receives at the k–th time-step. Specifically, since node x_i has access to its own value as well as the values of its neighbors, we can write $\mathbf{y}_i[k] = \mathbf{C}_i\mathbf{x}[k]$, $1 \leq i \leq N$, where $\mathbf{C}_i$ is the $(\deg_i + 1) \times N$ matrix with a single "1" in each row denoting the positions of the state-vector $\mathbf{x}[k]$ that are available to node x_i (i.e., these positions correspond to the nodes that are neighbors of node x_i, along with node x_i itself). Since $\mathbf{x}[k] = \mathbf{W}^k\mathbf{x}[0]$, the set of all outputs seen by node x_i over $L+1$ time-steps is given by

$$\underbrace{\begin{bmatrix} \mathbf{y}_i[0] \\ \mathbf{y}_i[1] \\ \vdots \\ \mathbf{y}_i[L] \end{bmatrix}}_{\mathbf{y}_i[0:L]} = \underbrace{\begin{bmatrix} \mathbf{C}_i \\ \mathbf{C}_i\mathbf{W} \\ \vdots \\ \mathbf{C}_i\mathbf{W}^L \end{bmatrix}}_{\mathcal{O}_{i,L}} \mathbf{x}[0] \ . \tag{2}$$

When $L = N - 1$, the matrix $\mathcal{O}_{i,L}$ in the above equation is the *observability matrix* for the pair $(\mathbf{W}, \mathbf{C}_i)$ [15]; in this

[3]This assumption is made in the interest of clarity, but the results can be extended to more general graphs without too much difficulty.

paper, we will use the term *observability matrix* to refer to $\mathcal{O}_{i,L}$ for any L. If $\text{rank}(\mathcal{O}_{i,L}) = N$, the pair $(\mathbf{W}, \mathbf{C}_i)$ is said to be *observable*, and node x_i can determine the entire initial value vector $\mathbf{x}[0]$ from the outputs of the system.

An important feature of the observability matrix is that there exists an integer ν_i, termed the *observability index*, such that the rank of the matrix $\mathcal{O}_{i,L}$ monotonically increases with L until $L = \nu_i - 1$, at which point it stops increasing. To see this, note that if $\text{rank}(\mathcal{O}_{i,L}) = \text{rank}(\mathcal{O}_{i,L-1})$ for some L, then it must be the case that $\mathbf{C}_i\mathbf{W}^L = \mathbf{K}\mathcal{O}_{i,L-1}$ for some matrix $\mathbf{K}$. Then,

$$\mathbf{C}_i\mathbf{W}^{L+1} = \mathbf{C}_i\mathbf{W}^L\mathbf{W} = \mathbf{K}\mathcal{O}_{i,L-1}\mathbf{W} = \begin{bmatrix}\mathbf{0} & \mathbf{K}\end{bmatrix}\mathcal{O}_{i,L},$$

and thus $\text{rank}(\mathcal{O}_{i,L+1}) = \text{rank}(\mathcal{O}_{i,L})$. This means that every new set of rows of the form $\mathbf{C}_i\mathbf{W}^k$ must increase the rank of $\mathcal{O}_{i,k-1}$ by at least one until the observability matrix reaches its maximum rank.

Based on the above discussion, we see that we can recast our analysis of the time-complexity of data accumulation via the linear iterative strategy as a weight matrix design problem, where the objective is to choose the weight matrix $\mathbf{W}$ (subject to the constraint that $w_{ij} = 0$ if $x_j \notin \mathcal{N}_i$) to maximize the rank of the observability matrix for each node in the fewest number of time-steps. To do this, we will start in the next section by characterizing the observability index for a class of linear *structured* systems.

VI. Observability Index of Structured Linear Systems

A linear system of the form

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k], \;\; \mathbf{y}[k] = \mathbf{C}\mathbf{x}[k]$$

with state vector $\mathbf{x} \in \mathbb{R}^N$, input $\mathbf{u} \in \mathbb{R}^m$, output $\mathbf{y} \in \mathbb{R}^p$ and system matrices $\mathbf{A} \in \mathbb{R}^{N\times N}, \mathbf{B} \in \mathbb{R}^{N\times m}, \mathbf{C} \in \mathbb{R}^{p\times N}$ is said to be *structured* if each entry in the system matrices is either identically zero, or an independent free parameter. A structured linear system is said to have a certain property (such as observability, controllability, etc.) if that property holds for some (real-valued) choice of free parameters. In fact, structural properties are generic (i.e., if the property holds for some choice of free parameters, it will hold for almost any choice of free parameters) [16].

In this section, we will derive a characterization of the *generic observability index* of a given structured system, which we will define to be the observability index that is attained for almost any real-valued choice of free parameters. We will start by investigating the observability of matrix pairs of the form $(\mathbf{A}, \mathbf{e}'_{1,N})$, where $\mathbf{A}$ is an $N \times N$ matrix, and $\mathbf{e}'_{1,N}$ is a row-vector of length N with a 1 in its first position and zeros elsewhere. Matrix $\mathbf{A}$ may be *structured* (i.e., every entry of $\mathbf{A}$ is either zero, or an independent free parameter), or it may be numerically specified. As commonly done in the study of structured systems [16], our analysis will be based on a graph representation $\mathcal{H}$ of matrix $\mathbf{A}$, which we obtain as follows. The vertex set of $\mathcal{H}$ is $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$, and the edge set is given by $\mathcal{E} = \{(x_j, x_i) \mid \mathbf{A}_{ij} \neq 0\}$. The weight on edge (x_j, x_i) is set to the value of $\mathbf{A}_{ij}$ (this can be a free parameter if $\mathbf{A}$ a structured matrix). Note that if $\mathbf{A}$ is the weight matrix for a linear iteration, $\mathcal{H}$ is simply the graph of the network $\mathcal{G}$ augmented with a self-loop on node x_i if $\mathbf{A}_{ii} \neq 0$.

The following theorem from [14] shows that the matrix pair $(\mathbf{A}, \mathbf{e}'_{1,N})$ will be observable under certain conditions.

Theorem 2 ([14]): Consider the matrix pair $(\mathbf{A}, \mathbf{e}'_{1,N})$, where $\mathbf{A}$ is an $N \times N$ matrix with elements from a field $\mathbb{F}$ of size at least N. Suppose that the following two conditions hold:

- The graph $\mathcal{H}$ associated with $\mathbf{A}$ is a spanning tree rooted at x_1, augmented with self-loops on every node.
- The weights on the self-loops are different elements of $\mathbb{F}$ for every node, and the weights on the edges between different nodes are equal to 1.

Then the pair $(\mathbf{A}, \mathbf{e}'_{1,N})$ is observable over the field $\mathbb{F}$. □

We now generalize the above theorem to the case where the graph of the system is a spanning forest (i.e., it consists of disjoint trees rooted at certain nodes).

Theorem 3: Consider the matrix pair $(\mathbf{A}, \mathbf{C})$, where $\mathbf{A}$ is an $N \times N$ matrix with elements from a field $\mathbb{F}$, and $\mathbf{C}$ is a $p \times N$ matrix of the form $\mathbf{C} = \begin{bmatrix}\mathbf{e}_{i_1,N} & \mathbf{e}_{i_2,N} & \cdots & \mathbf{e}_{i_p,N}\end{bmatrix}'$. Suppose the graph $\mathcal{H}$ associated with the matrix $\mathbf{A}$ satisfies the following two conditions:

- The graph $\mathcal{H}$ is a spanning forest rooted at $\{x_{i_1}, x_{i_2}, \ldots, x_{i_p}\}$, with self-loops on every node.
- No two nodes in the same tree have the same weight on their self-loops, and the weights on the edges between different nodes are equal to 1.

Let D denote the maximum number of nodes in any tree in $\mathcal{H}$. Then, the pair $(\mathbf{A}, \mathbf{C})$ is observable with observability index equal to D. □

Proof: Let $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_p$ denote the trees in $\mathcal{H}$, and let r_i denote the number of nodes in tree $\mathcal{T}_i$ (so that $N = r_1 + r_2 + \cdots + r_p$). Since the graph associated with $\mathbf{A}$ is a spanning forest rooted at $\{x_{i_1}, x_{i_2}, \ldots, x_{i_p}\}$, there exists a numbering of the nodes such that the pair $(\mathbf{A}, \mathbf{C})$ has the form

$$\left(\begin{bmatrix}\mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2 & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_p\end{bmatrix}, \begin{bmatrix}\mathbf{e}'_{1,r_1} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{e}'_{1,r_2} & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{e}'_{1,r_p}\end{bmatrix}\right),$$

where the $r_i \times r_i$ matrix $\mathbf{A}_i$ corresponds to the tree $\mathcal{T}_i$. If we denote the observability matrix for the pair $(\mathbf{A}, \mathbf{C})$ over D time-steps by $\mathcal{O}_{D-1}$, and the observability matrix for the pair $(\mathbf{A}_i, \mathbf{e}'_{1,r_i})$ as $\mathcal{O}_{i,D-1}$, it is easy to see that $\text{rank}(\mathcal{O}_{D-1}) = \sum_{i=1}^{p} \text{rank}(\mathcal{O}_{i,D-1})$. Since each matrix $\mathbf{A}_i$ is a spanning tree rooted at the first node in $\mathbf{A}_i$, and this matrix satisfies the conditions in Theorem 2, we see that the pair $(\mathbf{A}_i, \mathbf{e}'_{1,r_i})$ will be observable; specifically, the matrix $\mathcal{O}_{i,r_i-1}$ will have rank equal to r_i. Since D is the maximum value of all the r_i's, the above expression for the rank of the observability matrix becomes $\text{rank}(\mathcal{O}_{D-1}) = \sum_{i=1}^{p} r_i = N$, which concludes the proof of the theorem. ■

Corollary 1: Consider the matrix pair $(\mathbf{A}, \mathbf{C})$, where $\mathbf{A}$ is an $N \times N$ structured matrix, and $\mathbf{C}$ is a $p \times N$ matrix of the form $\mathbf{C} = \begin{bmatrix} \mathbf{e}_{i_1,N} & \mathbf{e}_{i_2,N} & \cdots & \mathbf{e}_{i_p,N} \end{bmatrix}'$. Suppose the graph $\mathcal{H}$ associated with the matrix $\mathbf{A}$ contains a path from every node to at least one node in the set $\{x_{i_1}, x_{i_2}, \ldots, x_{i_p}\}$, and furthermore, every node has a self-loop (i.e., the diagonal elements of $\mathbf{A}$ are free parameters). Let $\bar{\mathcal{H}}$ be a subgraph of $\mathcal{H}$ that is an optimal spanning forest rooted at $\{x_{i_1}, x_{i_2}, \ldots, x_{i_p}\}$. Let D denote the size of the largest tree in $\bar{\mathcal{H}}$. Then if $\mathbb{F}$ has size at least D, there exists a choice of parameters from $\mathbb{F}$ such that the observability matrix corresponding to the pair $(\mathbf{A}, \mathbf{C})$ has rank N over that field, with observability index equal to D.

The proof of the above corollary is readily obtained by setting the values of all parameters corresponding to edges that are not in $\bar{\mathcal{H}}$ to zero, and then choosing the weights for edges in $\bar{\mathcal{H}}$ to satisfy Theorem 3. Thus, one can explicitly choose parameters from a field of size D or greater in order to make the pair $(\mathbf{A}, \mathbf{C})$ observable, with observability index D. However, we will also be interested in characterizing the *generic* observability index of a given matrix pair, and this is the subject of the following theorem.

Theorem 4: Consider the matrix pair $(\mathbf{A}, \mathbf{C})$, where $\mathbf{A}$ is an $N \times N$ structured matrix, and $\mathbf{C}$ is a $p \times N$ matrix of the form $\mathbf{C} = \begin{bmatrix} \mathbf{e}_{i_1,N} & \mathbf{e}_{i_2,N} & \cdots & \mathbf{e}_{i_p,N} \end{bmatrix}'$. Suppose the graph $\mathcal{H}$ associated with the matrix $\mathbf{A}$ contains a path from every node to at least one node in the set $\{x_{i_1}, x_{i_2}, \ldots, x_{i_p}\}$, and furthermore, every node has a self-loop (i.e., the diagonal elements of $\mathbf{A}$ are free parameters). Let $\bar{\mathcal{H}}$ be a subgraph of $\mathcal{H}$ that is an optimal spanning forest rooted at $\{x_{i_1}, x_{i_2}, \ldots, x_{i_p}\}$. Let D denote the size of the largest tree in $\bar{\mathcal{H}}$. Then for almost any real-valued choice of free parameters in $\mathbf{A}$, the pair $(\mathbf{A}, \mathbf{C})$ will be observable and the observability index will be upper bounded by D. □

Proof: Let the free parameters of matrix $\mathbf{A}$ be given by $\lambda_1, \lambda_2, \ldots, \lambda_l \in \mathbb{R}$. When convenient, we will aggregate these parameters into a vector $\lambda \in \mathbb{R}^l$. With this notation, the matrix $\mathbf{A}$ will also be denoted as $\mathbf{A}(\lambda)$ to explicitly show its dependence on the free parameters. Any particular choice of the free parameters will be denoted by λ^*, with corresponding numerical matrix $\mathbf{A}(\lambda^*)$.

If the graph of matrix $\mathbf{A}$ satisfies the conditions in the theorem, then we know from Corollary 1 that there exists a choice of parameters $\lambda^* \in \mathbb{R}^l$ such that the observability matrix $\mathcal{O}(\lambda^*)_{D-1}$ for the pair $(\mathbf{A}(\lambda^*), \mathbf{C})$ has rank N. This means that $\mathcal{O}(\lambda^*)_{D-1}$ contains an $N \times N$ submatrix (denoted by $\mathbf{Z}(\lambda^*)$) whose determinant will be nonzero. Next, consider the matrix $\mathbf{Z}(\lambda)$ (which is obtained by reverting the special choice of parameters λ^* back to the original symbolic parameters). The determinant of $\mathbf{Z}(\lambda)$ will therefore be a nonzero polynomial in these parameters (since this polynomial is nonzero after a specific choice of parameters). Now, note that the set of parameters $(\lambda_1^*, \lambda_2^*, \ldots, \lambda_l^*)$ for which $\det \mathbf{Z}(\lambda^*) = 0$ forms an *algebraic variety*,[4] which has measure zero in the space $\mathbb{R}^l$ [16]. Thus, for almost any real-valued choice of parameters, the observability matrix has full rank after at most D time-steps, and the observability index is upper bounded by D. ■

[4]An algebraic variety is the set of points in a space that are the common roots of a given set of polynomials.

VII. Designing the Weight Matrix

Noting that the matrix $\mathbf{W}$ in (1) is a structured matrix (since each entry is either identically zero, or an independent free parameter), we can now use Theorem 4 to prove Theorem 1 (introduced at the end of Section III).

Proof: [Theorem 1] Consider the graph $\mathcal{H}$ associated with the matrix $\mathbf{W}$. In this case, $\mathcal{H}$ is obtained by taking the graph of the network $\mathcal{G}$ and adding a self-loop to every node to correspond to the free parameters w_{jj} on the diagonal of the matrix $\mathbf{W}$. Every node in $\mathcal{H}$ has a path to $\mathcal{N}_i$ (since the network $\mathbf{G}$ is strongly-connected), and furthermore, every node has a self-loop, which satisfies the conditions in Theorem 4. This implies that, with probability 1, the observability matrix for the pair $(\mathbf{W}, \mathbf{C}_i)$ will have full column rank and the observability index will be at most D_i. Furthermore, since this holds for any node x_i with probability 1, it will hold simultaneously for all nodes with probability 1. Thus each node x_i can recover $\mathbf{x}[0]$ from the outputs of the system over at most D_i time-steps. ■

VIII. Time-Optimality of Linear Iterative Strategies for Information Dissemination

Note that Theorem 1 only provides an upper bound on the number of time-steps required to disseminate information via a linear iterative strategy. Specifically, it says that linear iterative strategies perform at least as well as any tree-based scheme for information dissemination. The following example shows that there are circumstances where linear iterative strategies strictly outperform tree-based schemes.

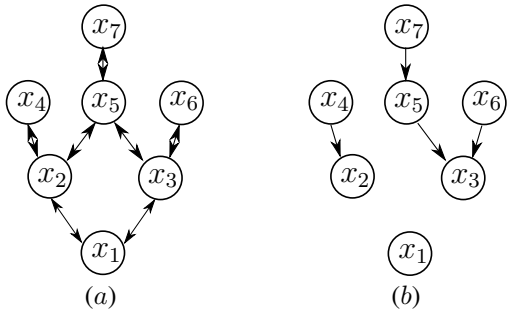

Fig. 3. (a) Network for information dissemination. (b) A subgraph of the original network that is an optimal spanning forest rooted at x_1, x_2 and x_3.

Example 1: Consider the strongly-connected network shown in Fig. 3.a. Theorem 1 indicates that for almost any real-valued choice of weights, a linear iterative strategy will allow each node x_i to obtain all of the initial values in the network after at most D_i time-steps, where D_i is the size of the largest tree in the optimal spanning forest rooted at $\{x_i\} \cup \mathcal{N}_i$. For example, if we consider x_1, it is easy to verify that the spanning forest that minimizes the size of the largest tree is the one in Fig. 3.b (this forest is not unique).

The largest tree in this forest has $D_1 = 4$ nodes, and thus a tree-based (routing) approach would require four time-steps in order for x_1 to accumulate all of the data.

Now let us consider a linear iterative strategy with weights chosen independently from a uniform distribution in the range $[a, b]$ (for any $b > a$). One can readily verify (by picking such a set of weights) that the observability matrix for the pair $(\mathbf{W}, \mathbf{C}_1)$ (where $\mathbf{C}_1 = \begin{bmatrix}\mathbf{I}_3 & \mathbf{0}\end{bmatrix}$) becomes full rank after three time-steps, Thus node x_1 can actually obtain all of the initial values in the network after using a linear iterative strategy for just *three* time-steps (i.e., as described in Section V). Note that this is the best possible result for this network, since the eccentricity of node x_1 is three. The linear strategy has outperformed the tree-based scheme in this example. □

While the above example shows that there are cases where the tree-based upper bound in Theorem 1 is not tight, a full characterization of all networks for which this is true is an avenue for future research. On the other hand, our development does provide us with a way to prove time-optimality of linear iterative strategies in certain networks (i.e., in these networks, no scheme can perform faster than the linear iterative strategy for gathering all of the data).

Theorem 5: Let $\mathcal{G}$ denote the (directed) graph of a fixed strongly-connected network where there is a unique path between any two nodes. Then, the linear iterative strategy is time-optimal for information dissemination in this network. □

Proof: Consider node x_i, and let x_j be any other node. Since there is only one path between any two nodes in the network, x_j's value must pass through a unique neighbor of x_i in order to get to x_i. For each $x_l \in \mathcal{N}_i$, let $D_{i,l}$ be the number of values that have to pass through x_l in order to get to x_i. Thus, *any algorithm* will require at least $D_i = \max_{x_l \in \mathcal{N}_i} D_{i,l}$ time-steps in order to convey all of the data to x_i (since each neighbor of x_i can only transmit one value to x_i at each time-step). This network has a unique spanning forest rooted at $\{x_i\} \cup \mathcal{N}_i$: simply take the edges of the forest to be those on the unique paths from each node to the corresponding unique neighbor of x_i. The largest tree in this spanning forest has D_i nodes. Theorem 1 indicates that the linear iterative strategy will take at most D_i time-steps, and since any algorithm will require at least D_i time-steps, the linear iterative strategy is time-optimal for this network. ■

Theorem 6: Let $\mathcal{G}$ denote the graph of a fixed undirected ring network. Then, the linear iterative strategy is time-optimal for information dissemination in this network. □

Proof: Consider node x_i, with neighbors x_{i-1} and x_{i+1} (if these indices are larger than N or smaller than 1, we can just have them wrap around). We form a spanning forest rooted at $\{x_i, x_{i-1}, x_{i+1}\}$ as follows. Take the tree rooted at x_{i-1} to be a path consisting of all nodes in $\mathcal{X} \setminus x_i$ that are strictly closer to x_{i-1} than to x_{i+1}. Similarly, take the tree rooted at x_{i+1} to be a path consisting of all nodes in $\mathcal{X} \setminus x_i$ that are closer to x_{i+1} than to x_{i-1} (including any node that is equidistant to both). Since the network is a ring, the trees rooted at x_{i-1} and x_{i+1} will have exactly $\lfloor \frac{N-1}{2} \rfloor$ nodes and $\lceil \frac{N-1}{2} \rceil$ nodes, respectively (both of these trees will be paths). Thus, $D_i = \lceil \frac{N-1}{2} \rceil$. However, it is easy to verify that the eccentricity of node x_i is precisely $\lceil \frac{N-1}{2} \rceil$, and thus the linear iterative strategy is time-optimal in this network. ■

IX. Summary

We showed that for almost any choice of real-valued weights in the linear strategy, the number of time-steps required for any node to gather all of the initial values is upper bounded by the size of the largest tree in a certain subgraph of the network. This means that linear iterative strategies perform at least as well as any tree-based scheme for information dissemination, and in some cases, perform faster (as we showed through an example). Furthermore, our upper bound is tight in certain networks, implying that linear iterative strategies are time-optimal for those networks.

References

[1] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4–5, pp. 436–463, 2005.

[2] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., 1996.

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[4] A. Giridhar and P. R. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 98–107, Apr. 2006.

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

[6] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *IEEE Journal on Selected Areas in Communications: Special Issue on Control and Communications*, vol. 26, no. 4, pp. 650–660, May 2008.

[7] S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2486–2507, June 2006.

[8] D. Mosk-Aoyama and D. Shah, "Information dissemination via network coding," in *Proceedings of the 2006 IEEE International Symposium on Information Theory*, 2006, pp. 1748–1752.

[9] A. Eryilmaz, A. Ozdaglar, and M. Medard, "On the delay and throughput gains of coding in unreliable networks," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5511–5524, Dec. 2008.

[10] J. Hromkovic, R. Klasing, A. Pelc, P. Ruzicka, and W. Unger, *Dissemination of Information in Communication Networks*. Berlin, Germany: Springer-Verlag, 2005.

[11] I. Chlamtac and S. Kutten, "Tree-based broadcasting in multihop radio networks," *IEEE Transactions on Computers*, vol. 36, no. 10, pp. 1209–1223, Oct. 1987.

[12] Z. Zhu and X. Hu, "Improved algorithm for minimum data aggregation time problem in wireless sensor networks," *Journal of Systems Science and Complexity*, vol. 21, no. 4, pp. 626–636, Dec. 2008.

[13] D. B. West, *Introduction to Graph Theory*. Prentice-Hall Inc., Upper Saddle River, New Jersey, 2001.

[14] S. Sundaram and C. N. Hadjicostis, "Information dissemination in networks via linear iterative strategies over finite fields," in *Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009, pp. 3781–3786.

[15] C.-T. Chen, *Linear System Theory and Design*. Holt, Rinehart and Winston, 1984.

[16] J.-M. Dion, C. Commault, and J. van der Woude, "Generic properties and control of linear structured systems: a survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, July 2003.