Systems & Control Letters 67 (2014) 19–23

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Step-size sequence design for finite-time average consensus in secure wireless sensor networks

Alain Y. Kibangou *

GIPSA-Lab, CNRS, University Joseph Fourier, 11 Rue des mathématiques, Grenoble Campus, 38402 Saint Martin d'Hères Cedex, France

ARTICLE INFO

Article history:
Received 14 November 2012
Received in revised form
14 January 2014
Accepted 31 January 2014
Available online 4 March 2014

Keywords:
Distributed algorithms
Secure wireless sensor networks
Finite-time average consensus
Distance-regular graphs

ABSTRACT

This paper concerns the study of average consensus in wireless sensor networks with aim of providing a way to reach consensus in a finite number of steps. In particular, we investigate the design of consensus protocols when, for security reasons for instance, the underlying graph is constrained to be strongly regular or distance regular. The proposed design method is based on parameters of the intersection array characterizing the underlying graph. With this protocol, at execution time, average consensus is achieved in a number of steps equal to the diameter of the graph, i.e. the smallest possible number of steps to achieve consensus.

1. Introduction

Distributed estimation, by means of static or mobile wireless sensor networks, is a topic of great interest for researchers from control and signal processing communities. In the recent literature, several distributed algorithms have been proposed and most of them are based on the concept of consensus where the goal is to reach an agreement between the nodes of a given network. Consensus can be obtained using linear iteration schemes, where each node repeatedly updates its value as a linear combination of its own value and those received from its neighbors. By appropriately designing the weighting policy, one can guarantee that all nodes asymptotically converge to a common value [1]. However, in several applications, requiring an arbitrary long time to get the consensus value is unacceptable. That is the reason why designing protocols for reaching consensus in finite-time has been recently the subject of intensive research. In the discrete-time framework, for linear iteration schemes, design of finite-time consensus protocols follows mainly two ways: methods based on the minimal polynomial concept [2,3] and those using matrix factorization [4–6].

Consider a static network. In methods based on the minimal polynomial concept, the network dynamics is modeled as a linear time-invariant system, with the state matrix given by the consensus matrix. In [2], it was shown that each node can compute the consensus value as a linear combination of its D consecutive observations, with D being the degree of the minimal polynomial of the associated consensus matrix. For computing the weights involved in such a linear combination, the method in [2] is distributed but each node needs to store $N(N-1)$ samples, with N being the network size. This method has been improved in [3] where much less data storage is required. However, the computation of the rank of a given matrix and that of its kernel is needed. Therefore, significant computation resources are required.

The matrix factorization approach consists in factorizing the average consensus matrix $\frac{1}{N}\mathbf{J}_N$, where $\mathbf{J}_N$ stands for an $N \times N$ all ones matrix, as a product of matrices consistent with the underlying graph topology. The existence of solutions has been studied in [4,5]. It has been shown that the number of factor matrices is lower-bounded by the diameter of the graph and upper-bounded by two times the radius of the graph [5]. It was also conjectured that the lower-bound can always be reached, i.e. the number of steps can always be set equal to the diameter of the graph [5]. However, a counter example has been recently reported in [7] showing that this conjecture is not true. The matrix factorization problem is equivalent to a set of multivariate polynomial equations to be solved using Gröbner bases for instance. However, their retrieval has exponential time complexity with respect to the number of parameters. It is hence impractical for solving the matrix factorization problem even for some small graphs [5]. Therefore, other centralized methods have been proposed by either switching the topology [4] or varying the weights [5]. In [6,8], the space of solutions has been restricted to Laplacian based matrices

* Tel.: +33 4 76 82 64 51; fax: +33 4 76 82 63 88.
E-mail address: alain.kibangou@ujf-grenoble.fr.

http://dx.doi.org/10.1016/j.sysconle.2014.01.010
0167-6911/

and closed-form solutions based on Laplacian eigenvalues were introduced. In general, matrix factorization approaches require global information (network topology or Laplacian spectrum) at design time. For this reason, they are generally considered as centralized design methods. However, decentralized factorization methods can also be designed. For instance, the Laplacian eigenvalues can be computed in a decentralized way at design time using methods recently proposed in the literature [9–11]. It is worth noting that, at execution time, matrix factorization based consensus protocols do not require storage of past data.

To summarize, the minimum polynomial approach leads to demanding storage and computation resources at execution time while local and simple tools are needed at design time. In contrast, matrix factorization leads to demanding tasks (centralized or decentralized) in the design time, while execution time is as simple as for standard consensus algorithms. However, according to the available a priori knowledge on the underlying graph structure, a trade-off is to be found. By thoroughly analyzing the different layers of interaction of the nodes, such knowledge can be obtained. In this paper, we show how to design finite-time average consensus protocols when the topology, viewed from network layer perspective, is constrained to be distance-regular.

The paper is organized as follows: in Section 2, we give the network model considered in this paper and some properties of families of graphs under study and then formulate the main problem to be solved. In Section 3, the proposed solution is derived and illustrated before concluding the paper.

2. Problem setting

2.1. Network model

In the design of wireless sensor networks, confidentiality, authenticity, integrity, and availability are crucial security services to be guaranteed. Since all encryption and authentication operations involve keys, key establishment is the first step to set up a security infrastructure [12,13]. A key predistribution scheme (KPS) consists in installing keys in each node before deployment. Several deterministic techniques for designing KPS can be found in the literature [12,14].

Let $\mathcal{N} = \{\nu_1, \nu_2, \ldots, \nu_N\}$ be a set of nodes with cardinality $|\mathcal{N}| = N$. We consider a network model where two nodes ν_1 and ν_2 can construct a single-hop bi-directional secure communication if and only if the following two conditions are satisfied: (i) they are within each other's communication range, and (ii) they share a given common number of keys. From condition (i) we can draw an undirected graph $\mathcal{G}_p(\mathcal{N}, \mathcal{E}_{(i)})$ that is related to the *physical layer* of the communication system while (ii) gives rise to an undirected graph $\mathcal{G}_n(\mathcal{N}, \mathcal{E}_{(ii)})$ related to the *network layer*. In other words, $(\nu_1, \nu_2) \in \mathcal{E}_{(i)}$ if ν_1 and ν_2 are within communication range and $(\nu_1, \nu_2) \in \mathcal{E}_{(ii)}$ if ν_1 and ν_2 share at least a given number of common keys. The secure communication graph is then given by the intersection graph $\mathcal{G} = \mathcal{G}_p \cap \mathcal{G}_n$, i.e. a graph with the vertex set $\mathcal{N}$ and the edge set $\mathcal{E} = \mathcal{E}_{(i)} \cap \mathcal{E}_{(ii)}$. In this paper, we assume that KPS techniques are based on robust designs and therefore give rise to strongly regular graphs or distance regular graphs (Hamming graph, in particular) for the network layer graph. In addition, we consider a scenario where the communication range is appropriately controlled so that the intersection graph corresponds to the network layer graph. Note that all graphs considered in this paper are simple graphs.

2.2. Distance regular graphs

Definition 1. Consider a graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ with the vertex set $\mathcal{N} = \{\nu_1, \nu_2, \ldots, \nu_N\}$. It is said to be regular of degree (or valency) K, when every vertex is precisely adjacent to K vertices. Its diameter $D(\mathcal{G})$ is the maximum distance between any two vertices in $\mathcal{G}$ [15].

It is usual to capture interactions in a graph by means of the adjacency matrix $\mathbf{A}$, which is an $N \times N$ matrix with entries $\mathbf{A}_{ij} = 1$ if $(\nu_i, \nu_j) \in \mathcal{E}$ and $\mathbf{A}_{ij} = 0$ if $(\nu_i, \nu_j) \notin \mathcal{E}$.

Definition 2. A graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ with diameter $D(\mathcal{G})$ is said to be distance regular if there exist integers a_k, b_k, and c_k, $k = 0, 1, \ldots, D(\mathcal{G})$, such that for any two vertices ν_i and ν_j in $\mathcal{G}$ and distance $k = dist(\nu_i, \nu_j)$, there are exactly a_k neighbors of ν_j in $\mathcal{N}_k(\nu_i)$, b_k neighbors of ν_j in $\mathcal{N}_{k+1}(\nu_i)$, and c_k neighbors of ν_j in $\mathcal{N}_{k-1}(\nu_i)$ [15].

The parameters of a distance regular graph with valency K are linked as $a_i + b_i + c_i = K$, with $a_0 = c_0 = b_{D(\mathcal{G})} = 0$ and $c_1 = 1$ [15]. Therefore, it is usual to characterize a distance regular graph by its intersection array $\{b_0, b_1, \ldots, b_{D(\mathcal{G})-1}; c_1, c_2, \ldots, c_{D(\mathcal{G})}\}$, where $b_0 \geq b_1 \geq \cdots \geq b_{D(\mathcal{G})-1}$ and $0 < c_1 \leq c_2 \leq \cdots \leq c_{D(\mathcal{G})}$.

Examples of distance regular graphs are: a cycle with N vertices (its intersection array is $\{2, 1, 1, \ldots, 1; 1, 1, \ldots, 1\}$ if N is odd and $\{2, 1, 1, \ldots, 1; 1, \ldots, 1, 2\}$ else), a connected strongly regular graph SRG (N, K, a, c) (its intersection array is $\{K, K-a-1; 1, c\}$).

Let us define $\mathcal{G}_k(\mathcal{N}, \mathcal{E}_k)$, where $(\nu_i, \nu_j) \in \mathcal{E}_k$ if and only if $dist(\nu_i, \nu_j) = k$, and $\mathbf{A}_k$ its adjacency matrix. Whatever the graph, we can note that [15]:

$$\sum_{k=0}^{D(\mathcal{G})} \mathbf{A}_k = \mathbf{J}_N. \tag{1}$$

For distance regular graphs, we also have the following property that will be used in the sequel [15]:

$$\mathbf{A}\mathbf{A}_k = b_{k-1}\mathbf{A}_{k-1} + a_k\mathbf{A}_k + c_{k+1}\mathbf{A}_{k+1}, \tag{2}$$

where $\mathbf{A} = \mathbf{A}_1$ stands for the graph adjacency matrix.

2.3. Average consensus problem

Assume that each node $\nu_i \in \mathcal{N}$ has an initial scalar $x_i(0) \in \Re$. We are interested in the computation of the average of these initial values. For this purpose, a linear iteration scheme, with properly chosen parameters, allows achieving such a task in a distributed way. Therefore, at each iteration, node ν_i updates its value $x_i(t)$ as a linear combination of the data received from its neighbors with its own value:

$$x_i(t+1) = w_{ii}x_i(t) + \sum_{j|\nu_j \in \mathcal{N}_1(\nu_i)} w_{ij}x_j(t). \tag{3}$$

In a matrix form, we get: $\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t)$, where $\mathbf{x}(t) = \begin{pmatrix} x_1(t) & \cdots & x_N(t) \end{pmatrix}^T$, with the matrix $\mathbf{W}$ being consistent with the graph topology, i.e. the off-diagonal entries w_{ij} of $\mathbf{W}$ are nonzero iff $\nu_j \in \mathcal{N}_i$. Average consensus is then reached if all nodes converge to the average of their initial values. It is well known that the necessary and sufficient convergence condition can be stated as: $\mathbf{W}$ is doubly stochastic, i.e. $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$ and $\mathbf{W}\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is an all ones N-dimensional vector and admits 1 as a simple eigenvalue whereas the magnitude of the remaining eigenvalues is strictly lower than 1. Different consensus matrices $\mathbf{W}$ exhibiting the properties above can be found in the literature [1,16].

The convergence speed of the linear iteration scheme for reaching consensus is directly linked to the second largest eigenvalue of the consensus matrix $\mathbf{W}$. Therefore, for accelerating the convergence of the consensus algorithm, various authors have proposed to appropriately modify the spectrum of the consensus matrix by optimizing its entries [1]. Although consensus algorithms can be made fast enough, setting the stopping criteria is not trivial. In most consensus algorithms, there is no distributed way for each individual node to know if consensus has been reached within desired error margin, except for the work in [17]