Effects of Graph Topology on Performance of Distributed Algorithms for Networked Control and Sensing

John S. Baras and Pedram Hovareshti

Abstract— **We consider distributed collaborative control and sensing as they frequently arise in networked control systems. The algorithms are constrained to use local information. We show by experiments that the performance of such distributed, local information based algorithms, can depend dramatically on the structure of the underlying topology (connectivity pattern) of the agents. We investigate the speed of convergence, accuracy, robustness and resiliency of such algorithms including consensus problems. We consider several graphs that can be used to represent collaborative control and communication patterns. We first show that small world topologies offer several advantages from a perspective of a favorable tradeoff between performance of collaborative behaviors vs costs of collaborative behaviors (or equivalently constraints for collaboration). Second, we show that a two level hierarchy consisting of carefully located and controlled 'leaders' at the higher level and the rest of the agents at the lower level, can provide a very efficient communication pattern with substantial improvement of performance. We close with a description of the possible topologies for this two tier structure and their performance.**

I. Introduction

In recent years the study of distributed algorithms for jointly achieving a common goal has received much attention in the control community. Collaborative control of groups of robots, flocking schemes, gossip algorithms and cooperation in sensor coverage are examples of different applications of distributed algorithms. The agents are provided with simple sets of decision making algorithms or dynamics, such that each agent take an action using its local information. The emergence of a desired global behavior is the goal of such schemes. The effectiveness of these schemes depends on three important factors: 1) The speed of convergence; 2) Robustness to failure of agents/connections; 3) Energy/communication efficiency. All of these factors cannot be necessarily achieved at once. On the contrary there is a trade off in achieving these objectives. An important point to notice is that the speed of convergence and robustness depend on both the structure of the network and the dynamics of the agents.

The subject of dynamic systems on graphs has also gained attention in other communities. Following the popular small world model of Watts and Strogatz and the preferential attaching model of Barabasi and Albert, many efforts have been dedicated to construct a scientific framework for studying networks and the processes running on them. The main interest of the ongoing research in this community is to understand the formation and function of real world networks. The networks under study come from different branches of science and include the World Wide Web, social networks, and biological systems.

The two research areas explained above have some overlap. Although the results in network science are usually asymptotic, they can provide control researchers with useful insight. As an example, Durrett [6] studies mixing times of random walks on different types of small world and random graphs. This is closely connected to the convergence rate of average consensus schemes considered in the control literature.

We address some structural properties of distributed schemes running on networks. We first show that the performance of distributed, local information based algorithms can be highly dependent on the structure of the underlying graph. In the case of consensus problems, we show that small world topologies offer a favorable tradeoff between performance (convergence speed) versus cost of collaboration (connectivity cost). We then generalize the concept of social leaders introduced by Blondel et al. [1] to classify the agents of more importance in a network. The two level hierarchy can provide a very efficient communication pattern. We utilize a distributed fixed point scheme to provide nodes with global information on which leaders they have more access to.

II. The importance of being well connected

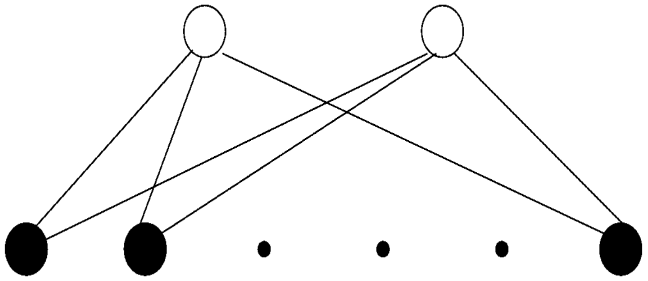
In this section we study the importance of being well connected in distributed algorithms. The first two examples given in this section demonstrate how being well connected can serve agents' selfish purposes. The next examples show how being well connected helps the overall network to be able to perform distributed algorithms in a faster and more robust manner.

A. Local majority voting

We start with a classic example by Peleg [14] which shows that in voting schemes a small number of well connected nodes can determine the outcome of the process. Consider n citizens each living on a vertex of a graph. Each citizen has an opinion about voting Yes or No on a controversial subject. However, citizens observe a rule by which they first ask privately their neighbors opinions. Each person then casts their vote. They will cast "Yes"(resp. "No") if the majority of their neighbors -including themselves- are "Yes-voters"(resp. "No-voters"). The question is what is the minimum number of "No-voters" that can guarantee a "No"

"Research partially supported by the U.S. Army Research Office under Award No DAAD 190110494."

The authors (baras@isr.umd.edu, hovaresp@isr.umd.edu) are with the Institute for Systems Research, University of Maryland College Park, MD 20742, USA.



c

Fig. 1. Two “No-voters” are enough to control the local majority polls.

outcome. As shown in Figure 1 the answer is 2. Every one of the $n-2$ “Yes- voters” should change its vote, because of having two “No-voters” in their neighborhood.

It is worthwhile to notice that all the “Yes-voters” observe a 2 to 1 majority of “No-voters” in their neighborhood. However, each “No-voter” observes a huge majority of “Yes-voters” in their neighborhood. Now, consider that the nodes follow the polling rule iteratively. In this case, each node will oscillate between Yes and No. However, if the “No-Voters” fail to observe the rule, the iteration will converge and all the nodes will vote No after the first iteration.

Peleg also shows that for large n, a negligible minority of $2\sqrt{n}$ “No-voters” can force all the voters to decide to vote No in just one iteration. This can be achieved by a clique of well connected “No-voters” who are attached to groups of badly connected “Yes-voters” as in Figure 2. In this case, by following the rules the “No-voters” can force the “Yes-voters” to change their vote while maintaining their own No votes. Therefore the iterative scheme will converge to an “all-No” configuration in just one step.

B. Consensus schemes

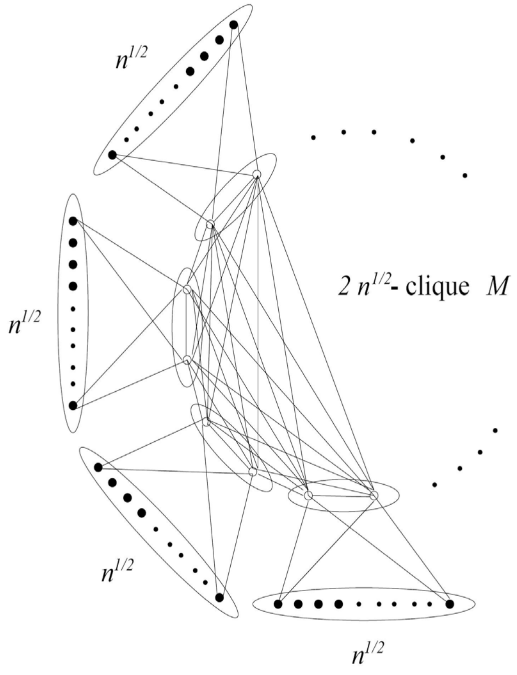
In Vicsek’s model for leaderless coordination [12] [17], at each time instant each agent’s state variable is updated using a local rule based on the average of its own state variable plus the state variables of its neighbors at that time. The local neighborhoods are time dependent. Each agent’s dynamic can be represented as:

$$\theta_i(t+1) = <\theta_i(t)> = \frac{1}{1+n_i(t)}[\theta_i(t) + \sum_{j\in N_i(t)} \theta_j(t)] \qquad (1)$$

Here $N_i(t)$ denotes the set of neighbors of agent i at time t and $n_i(t)$ denotes the cardinality of this set. The dynamics of the system can be written in matrix form. Let $\mathbb{G}_u$ be the set of possible graphs on n vertices. Let P be a suitably defined set that indexes the set $\mathbb{G}_u$. For each $G_p \in \mathbb{G}_u$ define a corresponding F-matrix as:

$$F_p = (I+D_p)^{-1}(A_p+I) \qquad (2)$$

where A_p is the adjacency matrix of the graph G_p and D_p is the diagonal matrix whose ith diagonal element is the degree of vertex i. The F matrices are a class of stochastic matrices and convergence of consensus protocols depends on properties of their infinite products. In this way linear consensus schemes are closely related to Markov chains and random walks on graphs with self loops. Different connectivity assumptions (symmetric vs. asymmetric neighborhoods) as well as different topology assumptions (fixed vs. changing) result in different sufficient conditions for convergence of consensus problems which can be found in [12], [7], [2] and references therein. Consider the fixed topology case. Vicsek’s model in this case is related to a random walk on a fixed graph with self loops. If the graph is connected, then $\lim_{t\to\infty} F_p^t = \mathbf{1}^T$, where $\pi^T F_p = \pi^T$. Therefore, all the agents’ state variable will converge to $x_\infty = \sum_{i=1}^n \pi_i x_i(0)$. It can be easily verified that $\pi_i = \frac{n_i+1}{2l+n}$, where l equals the number of edges in the graph. Therefore each agent’s contribution to the consensus is proportional to the number of neighbors it has. Therefore, for example in Figure 1, the ratio of the contribution of white agents to the contribution of black agents is of order $O(n)$.



c

Fig. 2. Two “No-voters” are enough to control the local majority polls.

C. Robustness

In order to be functional, distributed algorithms need to be robust to agent and link failures. In [8], Gupta et al. address the problem of formulation of robustness in distributed algorithms using a cost function approach. [8] defines three failure modes: Mode 1 in which agents cease functioning and communication, Mode 2 in which agents propagate constant arbitrary values, and Mode 3 in which agents propagate varying arbitrary values. It is shown that consensus problems, for example, are robust to $l-1$ node failures of Mode 1, in an $l-$connected graph. A classic theorem in graph theory states that the minimum number of vertices separating two independent sets of nodes is equal to the maximum number of disjoint paths between them [5]. This implies that well-connectedness will improve the robustness of distributed algorithms.

D. Fast convergence

A very important issue in distributed algorithms is the speed of convergence. In this section we consider discrete time consensus problems arising from Vicsek's model. The faster the consensus reached, the better the performance of the protocol. Since the applications that use consensus protocols involve many agents, it is necessary for all of them to converge quickly. The convergence rate is a function of the topology of the underlying graphs. This problem is actually closely connected to the asymptotic behavior of Markov chains. In fact if we consider a fixed topology, the convergence rate of the consensus protocol is nothing but the convergence rate to the stationary distribution of the Markov chain corresponding to the stochastic matrix F. Consider the system:

$$\theta(t+1) = F_{\sigma(t)}\theta(t) \tag{3}$$

as before where $F_p = (I+D_p)^{-1}A_p$ are stochastic matrices with nonzero diagonal elements. In the case of a fixed graph topology, the second largest eigenvalue modulus (SLEM) of the corresponding F matrix determines the convergence speed. This is because,

$$\theta(\infty) - \theta(t) = (F^{\infty} - F^{t})\theta(0) \tag{4}$$

Since F is a primitive stochastic matrix, according to the Perron-Frobenius theorem [16], $\lambda_1 = 1$ is a simple eigenvalue with a right eigenvector $\mathbf{1}$ and a left eigenvector π such that $\mathbf{1}^T\pi = 1$, $F^{\infty} = \mathbf{1}\pi^T$ and if $\lambda_2, \lambda_3, ..., \lambda_r$ are the other eigenvalues of F ordered in a way such that $\lambda_1 = 1 > |\lambda_2| \geq |\lambda_3| \geq ... \geq |\lambda_r|$, and m_2 is the algebraic multiplicity of λ_2, then

$$F^t = F^{\infty} + O(t^{m_2-1}|\lambda_2|^t) = \mathbf{1}\pi^T + O(t^{m_2-1}|\lambda_2|^t) \tag{5}$$

where $O(f(t))$ represents a function of t such that there exists $\alpha, \beta \in R$, with $0 < \alpha \leq \beta < \infty$, such that $\alpha f(t) \leq O(f(t)) \leq \beta f(t)$ for all t sufficiently large. This shows that the convergence of the consensus protocol is geometric, with relative speed equal to SLEM. We denote $\mu = 1 - SLEM(G)$ as the spectral gap of a graph, so graphs with higher spectral gaps converge more quickly. If the matrix F is symmetric, its SLEM can be written as the norm of its restriction to the subspace orthogonal to $\mathbf{1}$ [19]. However, the F matrices are not symmetric in general. In fact although the underlying graph structure is symmetrical, the weights that each node applies to another node are determined by its own degree. In general the SLEM of F matrices are not easily computable.

The ergodic coefficient [16], [3] is defined by:

$$\tau_1(F) = max_{x \in W, ||x||_1 = 1} ||x' F|| \tag{6}$$

in which W is the orthogonal subspace to $\mathbf{1}$.

This means that the matrix F contracts the subspace W by at least $\tau_1(F)$ at each iteration. If we denote the ith row of F by f_i, the ergodic coefficient can be written as:

$$\tau_1(F) = \frac{1}{2} max_{i \neq j} ||f_i - f_j|| \tag{7}$$

Also for any two stochastic matrices F_1 and F_2, $\tau_1(F_1F_2) \leq \tau_1(F_1)\tau_1(F_2)$. [9]

The ergodic coefficient provides a tractable upper bound for the SLEM and if the F matrices are ergodic, it provides a computable geometric rate to steady state in many cases. For Σ a finite set of $n \times n$ stochastic matrices, $\tau_1(\Sigma)$ is defined as $max_{F \in \Sigma}(F)$.

For the general case where topology changes are also included, Blondel *et al* [2] showed that the joint spectral radius of a set of matrices derived from F matrices determines the convergence speed. For Σ a finite set of $n \times n$ matrices, their joint spectral radius is defined as:

$$\rho = \limsup_{t \to \infty} \max_{A_1, \ldots, A_t \in \Sigma} ||A_t \ldots A_1||^{1/t} \tag{8}$$

Calculation of the joint spectral radius of a set of matrices is a mathematically hard problem and is not tractable for large sets of matrices. Using ergodic coefficients of blocks of matrices as in [9] can provide us with geometric rates. However, it is worthwhile to notice that graphs with well-connected nodes guarantee fast convergence. This is a direct result of the Cheeger inequality which relates the spectral gap of an F matrix to the conductance of the corresponding graph [3]. Switching over such topologies will also result in good convergence speed.

Since agents usually have energy constraints, the number of agents with which they communicate is limited. Therefore an important design issue is to find topologies which satisfy certain performance provided that the number of the links each agent can establish is less than an upper bound. In the next section we study the small world topologies and show their advantages from a perspective of favorable trade off between cost of communication versus speed of convergence.

III. Convergence in "small world" graphs

Watts and Strogatz [18] introduced and studied a simple tunable model that can explain behavior of many real world complex networks. Their "small world" model takes a regular lattice and replaces the original edges by random "shortcut" ones with some probability $0 \leq \phi \leq 1$. It is conjectured that dynamical systems coupled in this way would display enhanced signal propagation and global coordination, compared to regular lattices of the same size. The intuition is that the short paths between distant parts of the network cause high speed spreading of information which may result in fast global coordination. Olfati-Saber [15] studied continuous time consensus protocols on small world networks and proposed some conjectures. In this study, we use a variant of the Newman-Moore-Watts [13] improved form of the $\phi-$model originally proposed by Watts and Strogatz. The model starts with a ring of n nodes, each connected by undirected nodes to its nearest neighbors to a range k. Shortcut links are added

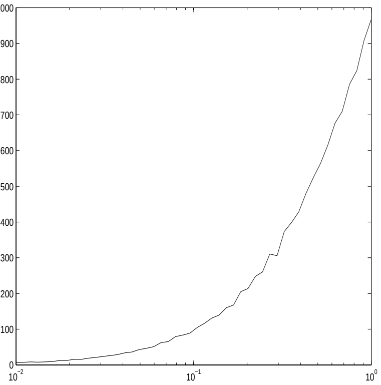


Fig. 3. Spectral gap gain for $(n,k) = (500,3)$

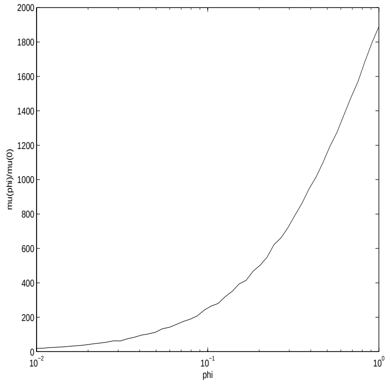


Fig. 4. Spectral gap gain for $(n,k) = (1000,5)$

-rather than rewired- between randomly selected pairs of nodes, with probability ϕ per link on the underlying lattice; thus there are typically $nk\phi$ shortcuts. Here we actually force the number of shortcuts to be equal to $nk\phi$ (comparable to the Watts $\phi-$model.) This reflects the limitation on the resources of networks. In our study, we have considered different initial rings $(n,k) = (100,2), (200,3)$, $(500,3), (1000,5)$, generated 20 samples of small world graphs $G(\phi)$ for 50 different ϕ values chosen in a logarithmic scale between 0.01 and 1. Picking these choices of (n,k) is done for comparison purposes with the results of [15]. In the figures 3 and 4, we have depicted the gain in spectral gap of the resulting small world graphs with respect to the spectral gap of the base lattice. We just illustrate the results of cases $(500,3)$ and $(1000,5)$. The others follow a similar pattern. Some important observations and comments follow:

1) In the low range of ϕ $(0 < \phi < 0.01)$ there is no spectral gap gain observed and the SLEM is almost constant and a drastic increase in the spectral gap is observed around $\phi = 0.1$.
2) Simulations show that "small world graphs" possess good convergence properties as far as consensus protocols are concerned.

The results show that adding $nk\phi$ shortcuts to a $1-d$ lattice dramatically improves the convergence properties of consensus schemes for $\phi \approx 0.1$. This is a trade off between the performance of collaborative behavior versus the cost of collaboration. For example in a $(500,3)$ lattice, by adding randomly 150 edges, we can on average increase the spectral gap approximately by a factor of 100.

Our aim is both to understand how this has happened and find a more clever way of adding edges so that after adding 150 edges to a $(500,3)$ lattice we get much larger increase of the spectral gap.

To formulate this problem, we consider a dynamic graph which evolves in time starting from a 1-d lattice $G_0 = C(n,k)$. Let's denote the complete graph on n vertices by K_n. Also, denote the complement of a graph $G = (V,E)$ -which is the graph with the same vertex set but whose edge set consists of the edges not present in G - by $\bar{G}$. So, $E(\bar{G}) = E(K_n) \setminus E(G)$.

If we denote the operation of adding an edge to a graph by A, the dynamic graph evolution can be written as:

$$\begin{array}{ll} G(t+1) = A(G(t),u(t)) & t = 0,1,2,\ldots,nk\phi - 1 \\ u(t) = e(t+1) & e(t+1) \in E(\bar{G}(t)) \\ G(0) = G_0 & \end{array} \qquad (9)$$

So, now the problem to solve is:

$$\min_{e(1),\ldots,e(n) \in E(\bar{G}(0)),\ldots,E(\bar{G}(n-1))} \max\left[\lambda_2(F(nk\phi)), -\lambda_N(F(nk\phi))\right]$$
$$subject \quad to: \quad (9) \qquad (10)$$

where $F(nk\phi) = D(G(nk\phi))^{-1}A(G(nk\phi))$. We will now mention some observations which are useful to build a framework for studying the above problem.

A. Spectral analysis

The choice of $G_0 = C(n,k)$ to be a regular 1-d lattice with self loops means that (possibly after re-labeling vertices) the adjacency matrix of the graph can be written as a circulant matrix:

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & . & . & . & a_n \\ a_n & a_1 & a_2 & . & . & . & a_{n-1} \\ a_{n-1} & a_n & a_1 & . & . & . & a_{n-2} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ a_2 & a_3 & . & . & . & . & a_1 \end{pmatrix} = circ[a_1, a_2, \ldots a_n] \qquad (11)$$

in which:

$$a \triangleq [a_1, a_2, \ldots a_n] = [\underbrace{1,\ldots,1}_{k+1} \quad \underbrace{0,\ldots,0,}_{n-2k-1} \quad \underbrace{1,\ldots,1}_{k}] \qquad (12)$$

Circulant matrices have a special structure which provides them with special properties. All entries in a given diagonal are the same. Each row is determined by its previous row by a shift to the right (modulo n). Consider the $n \times n$ permutation matrix, $\Pi = circ[0 \quad 1 \quad 0 \quad \ldots \quad 0]$. Then for any circulant matrix we can write:

$A = circ[a_1, a_2, \ldots, a_n] = a_1 I + a_2 \Pi + \ldots + a_n \Pi^{n-1}$. For a vector $a = [a_1, a_2, \ldots, a_n]$, the polynomial $p_a(z) = a_1 + a_2 z + a_3 z^2 + \ldots a_n z^n$ is called the representer of the circulant. The following theorem based on [4] states how to calculate the eigenvalues of circulants.

Theorem 1: [4] Let $\omega = e^{\frac{2\pi\sqrt{-1}}{n}}$ be the nth root of unity. The eigenvalues of $A = circ[a_1, a_2, ..., a_n]$ are given by $\lambda_i = p_a(\omega^{i-1})$, where $i = 1, 2, ...n$.

The main result considering the spectral properties of G_0 follows.

Proposition 1: The corresponding F matrix of $G_0 = C(n,k)$ is circulant. Furthermore, its SLEM has multiplicity at least 2.

Sketch of Proof: Since $G_0 = C(n,k)$ is $2k+1$-regular (including the self loop), $F = D^{-1}A = \frac{1}{2k+1}A$. So F is circulant $F = circ(\frac{1}{2k+1}a)$, where a is as in (12). The representer of this circulant is

$$p_a(z) = \frac{1}{2k+1}(1+z+...+z^{k-1}+z^k+z^{n-k}+z^{n-k+1}+...+z^{n-1}) \tag{13}$$

So, the eigenvlues of this matrix are $\lambda_i = p_a(\omega^{i-1})$. It is easy to show that $\lambda_1 = 1$ and moreover it is a simple eigenvalue because the underlying graph is connected. Since for integers A and B, $\omega^{An+B} = \omega^B$, it follows that $\lambda_2 = \lambda_n$, $\lambda_3 = \lambda_{n-1}$ and so on. In the case that n is odd apart from $\lambda_1 = 1$, all eigenvalues come in pairs. In the case that n is even, it can be shown that $\lambda_{\frac{n}{2}+1}$ is the only eigenvalue apart from -1 which can be single, however direct calculation shows that it is equal to $\frac{(-1)^k}{2k+1}$ which is clearly less than $\lambda_2 = \lambda_n$. A simple geometric argument shows that $SLEM = \lambda_2 = \lambda_n = \frac{1}{2k+1}[1+2Re(\omega)+2Re(2\omega)+...+2Re(k\omega)] < 1$ and $\lambda_i \leq \lambda_2$ for $i \in 2, ..., n-1$. This shows that for the case where $k \ll n$, which are the cases we are more interested in, as $n \longrightarrow \infty$ two of the non-unity eigenvalues approach 1. This describes the slow convergence of consensus protocols when diameter is large.

B. A "mean field" explanation for fast convergence in Small World networks

Our simulation results show that adding a small number of well chosen links to a ring-structured graph should result in high convergence rate. However analytical verification of this result is difficult due to the non-symmetric structure of F matrices. Here we try to justify our result using a "mean field" approach and perturbation analysis. In the ϕ model of small world and its variants, a regular lattice is considered and m shortcuts are added randomly where m is equal to a proportion ϕ of the lattice's initial edges. In this analysis, following [10] we reflect the effect of shortcuts by adding "small " nonzero positive numbers for the entries of the F which correspond to non-adjacent nodes of the lattice. Since the Small World model is a probabilistic model, this will take care of the small probability of a shortcut between any two nonadjacent nodes. Although by adding uniform perturbations the topology of the graph is not respected, we anticipate that the analysis gives insight on random communication patterns for small world networks. We state the result for the case where the base lattice is a ring but the result can be extended to $C(n,k)$ for other ks.

We follow the perturbation approach to small world networks proposed by Higham [10]. Consider the base lattice to have a ring topology on n nodes, $G(n,2)$ and the corresponding F matrix F_0. This can be also viewed as a random walk with self loops. This is similar to a particular case of our base circulant matrix F_0. Therefore the base matrix is:

$$F_0 = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & 0 & ... & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & ... & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & ... & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \frac{1}{3} & 0 & 0 & ... & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \tag{14}$$

We know that:

Corollary 1: $SLEM(F_0) = \frac{1}{3}[1+2cos(\frac{2\pi}{n})]$. Furthermore it has multiplicity at least 2.

Now we perturbs the nonzero entries of the matrix F_0 by $\varepsilon = \frac{K}{n^\alpha}$ for fixed $K > 0$ and $\alpha > 1$ in the limit $N \longrightarrow \infty$, to get the perturbed matrix F_ε:

$$F_\varepsilon = \begin{pmatrix} \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \varepsilon & \varepsilon & ... & \varepsilon & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} \\ \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \varepsilon & ... & \varepsilon & \varepsilon \\ \varepsilon & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & ... & \varepsilon & \varepsilon \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \varepsilon & \varepsilon & \varepsilon & ... & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} & \frac{1}{3}-\frac{(n-3)\varepsilon}{3} \end{pmatrix} \tag{15}$$

We call the "shortcuts" created this way $\varepsilon-$shortcuts. F_ε is also a circulant matrix. The representer of this circulant is

$$\begin{aligned} p_a(z) = \frac{1}{3} - \frac{(n-3)\varepsilon}{3} + (\frac{1}{3} - \frac{(n-3)\varepsilon}{3})z + \\ \varepsilon z^2 + \varepsilon z^3 + ... + \varepsilon z^{n-2} + (\frac{1}{3} - \frac{(n-3)\varepsilon}{3})z^{n-1} \end{aligned} \tag{16}$$

So, the eigenvalues of this matrix are $\lambda_i(\varepsilon) = p_a(\omega^{i-1})$. For this matrix the largest eigenvalue is 1. Using a similar argument the SLEM can be calculated to be equal to:

$$\lambda_2(\varepsilon) = (\frac{1}{3} - \frac{n}{3}\varepsilon)(1 + 2cos\frac{2\pi}{n}) \tag{17}$$

Thus we can state the following proposition:

Proposition 2: Let $\varepsilon = \frac{K}{n^\alpha}$, $\alpha \geq 1$.

- For $\alpha > 3$, the effect of ε-shortcuts on convergence rate is negligible. $\alpha = 3$ is the onset of the effectiveness of shortcuts.
- For $\alpha = 2$, the shortcuts are dominantly decreasing SLEM.
- For $\alpha = 1$, almost all of the nodes communicate effectively and thus the SLEM is very small.

Proof:

For large n we can write:

$$\lambda_2(F_\varepsilon) = \frac{1}{3} + \frac{2}{3}cos\frac{2\pi}{n} - \frac{n\varepsilon}{3} - \frac{2n}{3}cos\frac{2\pi}{n} \Rightarrow$$

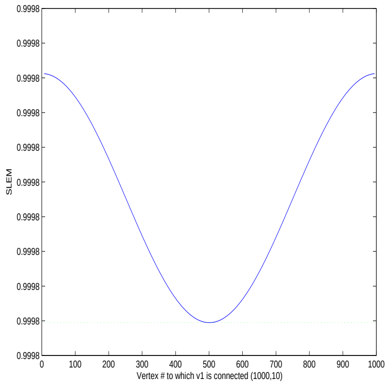


Fig. 5. Adding a shortcut (1000,5). The dotted line tangent to curve shows SLEM before adding edge

$$\lambda_2(F_\varepsilon) = 1 - \frac{4\pi^2}{3n^2} + o(\frac{1}{n^4}) - n\varepsilon + \frac{4\pi^2\varepsilon}{3n} + o(\frac{1}{n^3}) \qquad (18)$$

The first three terms are the contributions of the base lattice and the rest are the contributions of the perturbation. Comparing this to the SLEM of the base lattice

$$\lambda_2(F_0) = \frac{1}{3}(1 + 2cos\frac{2\pi}{n}) = 1 - \frac{4\pi^2}{3n^2} + o(\frac{1}{n^3}) \qquad (19)$$

yields the following results.

For the base lattice, the spectral gap decreases as fast as n^2. If ε is $o(n^\alpha), \alpha > 3$, then terms coming from the lattice are dominant, and therefore the shortcuts do not affect the spectral gap. For $\alpha = 3$ the terms regarding the shortcuts will be of the same degree as the terms from the base and for k large enough, the SLEM starts decreasing from the corresponding lattice SLEM. For $\alpha = 2$ the terms regarding the shortcuts are dominant and the SLEM is considerably decreased compared to the base lattice. Only for the case of $\alpha = 1$ the spectral gap does not vanish as $n \rightarrow \infty$.

As observed above $\varepsilon-$shortcuts are loosely analogous to the shortcuts in the ϕ-model.

C. Simulation results

We ran a set of simulations with different objectives based on (9). A counter intuitive result is that the SLEM does not monotonically change with addition of edges. Specifically, in cases when n is even, adding an edge will increase SLEM except for the case where a vertex is connected to the farthest vertex from it, that is i is connected to $i + n/2$ (modulo 2). In this case one of the multiplicities of the SLEM is lessened but the other multiplicity is not changed. Figures 5 and 6 illustrate this effect. The dotted line tangent to the curves show the SLEM of the original curves. The more distant the two joined vertices, the less increase in SLEM. Adding two shortcuts can however decrease the SLEM. It is worthwhile to mention that in all of our simulations, for a given n, shortcuts that reduced the diameter of the graph more, resulted in higher spectral gap. For example, for the case of adding 2 shortcuts to $G_0 = C(16, 2)$, Figure 6 shows the optimal topology. The analysis of this conjecture is the subject of future work.

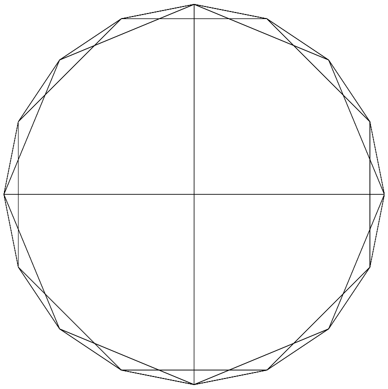


Fig. 6. The optimal topology; adding 2 shortcuts to $C(16, 2)$

IV. Hierarchical scheme

In this section we show that a very efficient communication pattern with substantial improvement in performance is possible by a two level hierarchical scheme. The idea here is that selecting a few well connected and controlled agents which are well protected should enhance the speed of convergence of distributed schemes like consensus algorithms.

Given n agents, suppose we can divide them into $O(\sqrt{n})$ groups each having $O(\sqrt{n})$ members. For each group suppose that we can select a "leader". The leaders should be able to have two properties, they should be well connected to the members of their group, and they should also be able to communicate with other leaders when necessary. If the distributed algorithm is carried out at each group separately and the leaders communicate on a higher level, the agents can enjoy faster convergence rates; the reduction of the size of each group from n to $\sqrt{n}$ results in faster intergroup convergence whereas the ease of communication between the leaders upon demand results in overall fast convergence.

We illustrate this by an example. Consider the agents of the graph in Figure 2 performing consensus algorithm. Suppose the agents previously known as "no-voters" are picked as leaders and the rest of the agents as regular agents. There are $\sqrt{n}$ groups of agents. Each group has two leaders. We usually select one leader for each group but selecting more leaders is allowable. The leaders of groups may communicate at some point on the clique, so if the consensus is reached inside a group, as soon as the leaders communicate, the consensus will be reached among all groups. The degree of leaders is $O(\sqrt{n})$, the degree of the regular nodes is a small number (here 3) and the average degree of all of the agents is also a small number. So, the communication is very efficient. Similarly we can consider topologies in which the leader has less/more intra-group connectivity and more/less inter -group connectivity.

The class of examples mentioned above assume perfect information about all the agents in the network. We now provide a semi-distributed method which can categorize the agents as “leader” or “regular”. Furthermore, the method assigns each regular agent with an influence vector which indicates which leader has more influence on it. This provides the nodes with some global picture of the network.

A. Distributed exploration of the graph structure

As shown in the previous sections, the structure of a graph plays a crucial role in the properties of a distributed algorithm that is running on it. Given a graph topology, individual nodes have only local knowledge about the graph structure, which includes information about their neighboring nodes. If any node wants to either improve its own performance - as in the local voting scheme- or a global performance measure -such as robustness- it needs to have more information about the global picture of the network. This information can be used by the node to refine its choice of neighbors in order to improve its performance.

The most complete measure of global graph structure is the adjacency matrix. Since each node has limited memory, energy, and computational capacity, they cannot store and process the adjacency matrix. Our goal is to devise a scheme to provide each node with a small vector that satisfies the following criteria:

1) Includes compact global information on how the node is located with respect to the other nodes.
2) It can be disseminated in a distributed manner achievable and requires minimal centralized action.

We propose a two stage algorithm for this purpose. Apart from a single data transmission and reception to a central authority by each node, the algorithm is carried out in a decentralized manner. In the first stage nodes will collaborate to find their social degree [1]. This is a local measure of how well connected a node is. Once the nodes find out their social degree, they will transmit it to a central authority which determines k “social leaders” of the graph, which are the nodes who are better connected than the rest. The central authority then broadcasts the list of k social leaders to all of the nodes. In the second stage of the algorithm, each node uses a simple iterative scheme to maintain a vector of size k which determines the influence of each social leader on it. We call this vector the *influence vector* associated to each node. In this section we define the proper concepts and give precise description of the algorithm.

B. Social degrees and leader nodes

To find the leaders or the agents with the highest influence we adopt a framework proposed by Blondel et al. [1] and use a slight generalization of their model to find the leader nodes. In [1] the authors define the social degree of a node as the number of the cycles of length 3 passing through that node. They also define a node to be a social leader provided that its social degree is greater or equal to its neighbors. Here we give a generalization of these definitions.

Definition 1: social degree: The number of the neighbors of a node is defined to be its Social degree of order 2. Social degree of order k $(k > 2)$ is defined to be the number of cycles of length k passing through a node. The social degree of order k of node v_i is denoted by $SD^{(k)}(v_i)$.

Definition 2: leader nodes: A node is called a leader node of order k if its social degree of order k is greater than or equal to that of its neighbors.

Notice that each node can determine its social degree of orders 2 and 3 by a simple query from its neighbors. Since determining higher order degree requires more effort, we use the orders 2 and 3 for our present application.

In the first stage of the algorithm, each node computes its social degree of order 2 and 3. It also queries the social degrees of its neighbors. Upon comparing its social degrees with its neighbors, if a node is found to be a leader of order 2 or 3, it transmits its degrees to the central authority. Upon receiving these data from the leader nodes, the central authority selects M nodes with the highest social degree in the following manner and gives an arbitrary order to them.

For given $\alpha > 0$ and $\beta = 1-\alpha$ and for all leader nodes l_i, $i = 1,2,...,K < N$, the central authority computes the leaders’ *social scores* as $SC(l_i) = \alpha SD^{(2)}(l_i) + \beta SD^{(3)}(l_i)$. Notice that the choice of α and β determines the preference between leaders in “star-like” neighborhood versus leaders of better connected neighborhoods. The central authority then selects the M leaders with the highest scores and give an arbitrary order to them and transmits their assigned order to them.

Once a selected leader is assigned its order $1 \le i \le M$ it will maintain the constant vector $e_i \in R^M$. This is the unit vector with 1 in its ith entry.

C. Determination of the influence vector

Our objective in this part is to associate with each of the regular nodes a vector that determines how well it is related to each of the leaders and how it is influenced by them. The amount of influence that a leader has on a local node is not only determined by their distance but also by the number of paths between them. We provide a definition for the influence vector based on the properties of random walks on graphs.

Definition 3: Consider a graph with M leaders and $n-M$ regular nodes. Consider a random walk on this graph starting from an arbitrary regular node i. The influence of leader node l_k $(k = 1,...,M)$ on any regular node i is the probability that a random walk that starts from i hits l_k before it hits any other leader node.

Given the leaders and their ordering, here we first describe the algorithm to determine the influence vector. Then we will show why it converges and why the algorithm outputs valid vectors as influence vectors.

We denote the influence vector of node i by $x_i \in R^M$. By $x_i^k(t)$ we mean the kth entry of the influence vector of node i evaluated at time t.

1) Algorithm: The influence vector of leader l_i is first assigned to be the unit vector $x_i = e_i$. These M vectors do not vary. For all regular nodes i, x_i is initialized randomly distributed uniformly on $[0,1]^M$. At each iteration time $t+1$ the leaders do not change their influence vectors, whereas each regular node updates its influence vector entry-wise using the following rule.

$$x_i^k(t+1) = \frac{1}{1+n_i(t)}[x_i^k(t) + \sum_{j\in N_i(t)} x_j^k(t)] \tag{20}$$

For $k = 1,2,...,n-M+1$.

The following theorem shows the convergence properties of the above scheme.

Theorem 2: If the underlying graph is connected, the iteration (20) converges to a set of unique vectors. Furthermore, $lim_{t\to\infty}x_i^k(t)$ is equal to the probability that a random walk starting at node i hits the leader node l_k before any other leader node.

Proof: The proof of convergence is similar to the proof in [11]. The particular form of the solution arises because the procedure solves a discrete version of the Dirichlet problem on the graph. We follow the proof of Bremaud [3]. Relabel the nodes, such that D={1,2,...n-M} denote the regular nodes and $\partial D = \{n-M+1,...,n\}$ denote the leader nodes, where $l_i = N-m+i$. For all $k = l_1,...,l_M$, define a function ϕ^i on the graph such that $\phi^k(l_j) = \delta(k,j)$, where δ is Dirac's Delta function.

Let $P = (I+D)^{-1}(A+D)$, where A is the adjacency matrix of the graph G, and D is the diagonal matrix with i'th diagonal element equal to the degree of node i (number of its neighbors). x_i^k converges to a value that satisfies the following equation

$$\begin{array}{ll} x_i^k = (Px^k)_i & i = 1,2,...,n-M \\ x_i^k = \phi^k(i) & i = n-M+1,...,n \end{array} \tag{21}$$

Note that P is a stochastic matrix and the equation is valid for the first N_R components of x. Let $\{Z_n^k\}_{n\geq 0}$ be a homogeneous Markov chain with state space $V = \{1,2,...,M\}$. Let T be the hitting time of ∂D. For each state $i \in V$ define:

$$h_i^k = E[\phi(Z_T^k)|Z_0^k = i] \tag{22}$$

Since the underlying graph G is connected, P is irreducible. Also $\forall i \in V,\ p_{ii} > 0$, which means the chain is aperiodic. The number of states is finite and therefore the chain is positive recurrent and $P(T < \infty|Z_0^k = i) = 1$.

By definition $h^k = \phi^k$ on ∂D and $x^k = \phi^k$ on D. By first step analysis:

$h_i^k = \sum_{j\in V} p_{ij}h_j^k$ on D. So:

$$h^k = \begin{cases} Px^k & on \quad D \\ \phi^k & on \quad \partial D \end{cases} \tag{23}$$

Therefore $h = x$ on the graph G. The proof of uniqueness of the solutions also follows from [3]. Notice that ϕ_i^k is defined so that h_i^k is equal to the probability of hitting the leader node l_k before the other leader nodes.

V. Conclusions

We showed in different contexts that the performance of distributed collaborative schemes can depend dramatically on the structure of the underlying topology. We showed that small world topologies offer several advantages from a perspective of a favorable tradeoff between performance of collaborative behaviors vs costs of collaborative behaviors. A two level hierarchy consisting of carefully located and controlled 'leaders' at the higher level and the rest of the agents at the lower level, was shown to provide a very efficient communication pattern with substantial improvement of performance. A semi-distributed method of finding proper "leader" nodes and measuring the influence of them on the regular ones in large networks was proposed.

References

[1] V. Blondel, C. de Kerchove, E. Huens, and P. Van Dooren. Social leaders in graphs. *Submitted to POSTA06*, 2006.

[2] V. Blondel, J. Hendrix, A.Olshevsky, and J. Tsitsiklis. Convergence in multiagent coordination, consensus and flocking.

[3] P. Bremaud. *Markov chains, Gibbs Fields, monte Carlo Simulations and Queues*. Springer Verlag, 2nd edition, 1999.

[4] P. J. Davis. *Circulant Matrices*. Wiley, 1979.

[5] R. Diestel. *Graph Theory*. Springer-Verlag, 2000.

[6] R. Durrett. *Random Graph Dynamics*. Cambridge University Press, 2007.

[7] Lei Fang and Panos Antsaklis. On communication requirements for multi-agents consensus seeking. *Proceedings of Workshop NESC05: University of Notre Dame*, 2005.

[8] V. Gupta, C. Langbort, and R. M. Murray. On the robustness of distributed algorithms. *Proceedings of the IEEE Conference on Decision and Control*, 2006.

[9] D. J. Hartfiel. *Nonhomogeneous Matrix Products*. World Scientific, 2002.

[10] D. J. Higham. A matrix perturbation view of the small world phenomenon. *Siam Journal on Matrix Analysis and Applications*, 25(2):429–44, 2003.

[11] P. Hovareshti and J.S. Baras. Decentralized control of networks of dynamic agents with invariant nodes: A probabilistic view. *ISR technical report: ISR TR 2004-39, Institute for Systems Research, University of Maryland, College Park*, 2004.

[12] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[13] M. E. J. Newman, C. Moore, and D. J. Watts. Mean-field solution of the small-world network model. *Phys. Rev. Lett.*, 84:3201–3204, 2000.

[14] D. Peleg. Local majority voting, small coalitions and controlling monopolies in graphs: A review. *Proceedings 3rd Colloquium on Structural information and Communication Complexity*, 1996.

[15] R. Olfati Saber. Ultrafast consensus in small-world networks. Preprint.

[16] E. Seneta. *Nonnegative Matrices and Markov Chains*. Springer, 2nd edition, 1981.

[17] T. Vicsek, A. Czirok, E. Ben Jakob, I. cohen, and O.Schochet. Novel type of phase transitions in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1299, 1995.

[18] D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[19] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Proceedings of 42th IEEE Conference on Decision and Control*, 2003.