Leader Selection in Multi-Agent Systems for Smooth Convergence via Fast Mixing

Andrew Clark, Basel Alomair, Linda Bushnell, and Radha Poovendran

Abstract— **In a leader-follower multi-agent system (MAS), a set of leader nodes receive state updates directly from the network operator. The follower nodes then compute their states based on the inputs from the leader nodes. In this paper, we study the problem of selecting a set of leader nodes in order to minimize the time required for the distributed coordination law used by the MAS to converge. We first represent the convergence time of a MAS in terms of the mixing time of a random walk on the underlying network graph. We then study two leader selection problems as convex optimization problems of fast mixing. First, we formulate the problem of selecting a fixed number of leaders in order to minimize the convergence time. We then study the problem of finding the minimum-size set of leaders in order to satisfy a constraint on the convergence time. We develop leader selection algorithms based on supergradient descent methods for static network topologies as well as a MAS experiencing random link failures and a MAS that switches between predefined topologies. We compare our leader selection algorithms with random and degree-based leader selection for both static and dynamic networks through simulation study. From the simulation comparisons, we note that the convergence rate of fast mixing is faster than that of degree-based methods. We also note that the fast mixing requires smallest number of leaders to achieve a given bound on the convergence time.**

I. Introduction

A multi-agent system (MAS) consists of a group of agents, each of which controls its own state based on the state information of its neighbors. Distributed protocols are used to exchange and propagate this state information among agents in order to accomplish global, network-wide tasks, even in cases where the network topology changes over time [1]. Applications of MAS range from autonomous aerial and ground vehicles [2] to biological networks [3].

In many of these applications, an external entity must steer the agent states towards a desired point; for example, in a formation of aerial vehicles, an operator may change the heading and altitude of the agents in order to track a specific target [4]. A common approach is to control a small subset of agents, designated as leaders. Since the follower agents update their states based on the observed states of the leaders, the leader agents act as external control inputs that influence the followers. By choosing the number of leader agents to be much smaller than the number of followers, a leader-follower approach enables scalable and efficient control of large multi-agent systems.

Effective operation of a leader-follower system requires the followers to receive inputs from the leaders and converge to their desired states based on those inputs. Furthermore, in order to ensure that the follower agents quickly complete their assigned task, the system must be designed so that the follower agents' states converge as rapidly as possible to their desired values. Recent studies have shown that the time required for the follower agents to converge, denoted as *convergence time*, is affected by the choice of leaders [5], [6]. While these studies quantify the impact of a given set of leader agents on the convergence time, they do not provide a tractable method for choosing leaders. In fact, to the best of our knowledge, there is currently no analytical approach to selecting leaders in order to minimize the convergence time.

In this paper, we present a framework for selecting a set of leader agents in order to minimize the convergence time of the system, defined as the time required for the follower agents to converge to within ϵ of their desired final states. We make the following specific contributions:

- We study the problem of leader selection to minimize convergence time by first proving that the convergence time is equal to the mixing time of a random walk on the underlying graph, defined as the time required for a random walk to reach its stationary distribution.
- We formulate convex relaxations of two leader selection problems, namely (i) the problem of selecting up to k leaders in order to minimize the convergence time, and (ii) the problem of selecting the minimum number of leaders in order to achieve a given bound on the convergence error, in static networks as well as dynamic networks that switch between predefined topologies.
- Using supergradient descent methods, we provide efficient algorithms for approximating the optimal leader set based on our convex relaxation framework. We also show that our convex relaxation gives a lower bound on the achievable convergence time.
- We demonstrate our results through simulation study and compare with other leader selection algorithms, including random leader selection, selection of high- and average-degree nodes as leaders, and a greedy selection heuristic.

The rest of this paper is organized as follows. In Section II, we review the related work on leader selection and convergence rate in MAS. In Section III, our system model and convergence rate metrics are presented. In Section IV, we formulate the problem of leader selection for smooth convergence in MAS with a static network topology. In Section V, we study leader selection in dynamic networks. Section VI presents simulation results. Section VII concludes the paper.

A. Clark, L. Bushnell, and R. Poovendran are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA. awclark@uw.edu, lb2@uw.edu, rp3@uw.edu

B. Alomair is with the Computer Research Institute, King Abdulaziz City for Science and Technology, Riyadh, Saudi Arabia. alomair@uw.edu

II. Related Work

Leader selection in multi-agent systems has been studied under a variety of design criteria. In [7], the problem of selecting the minimum number of leaders in order to achieve controllability was studied within a structural controllability framework [8]. Leader selection in order to minimize errors in the follower agent states due to link noise has been approached using continuous [9], [10] and discrete [11] optimization methods. In [12], selecting leaders in order to minimize the errors experienced while the follower states converge was studied using supermodular optimization techniques. In [13], a framework for selecting leaders based on joint consideration of controllability and performance criteria, such as errors due to link noise, was presented.

Minimizing the convergence time has been considered in other contexts, such as design of gossip algorithms [14] and fast-mixing Markov chains [15]. These methods assume a homogeneous network, in which there are no leader nodes, and determine the optimal choice of link weights in order to minimize the convergence time.

When a set of leader nodes is given, the effect of the leader set on the convergence time has been analyzed using the spectrum of the associated graph Laplacian [5], [6]. These studies, however, are applicable for quantifying the performance of a MAS for a fixed set of leaders, rather than choosing which nodes should act as leaders in order to minimize convergence time.

III. Model and Preliminaries

In this section, the system model is given and the convergence time for the MAS is formally defined. We then define the mixing time of a random walk on a graph and prove the equivalence between convergence time and mixing time.

A. System Model

A MAS consisting of n agents is considered, with each agent given a unique index in $V = \{1, \ldots, n\}$. The set of communication links between agents is denoted E; the neighbor set $N(i)$ consists of the set of agents such that $(i,j) \in E$. The network topology $G = (V, E)$ may be static, or may change over time due to link failures or switching between topologies.

Agent i has a time-varying state variable $x_i(t) \in \mathbb{R}$. A subset of agents, denoted S, are designated as leaders, and receive their states directly from the network operator. The remaining agents update their states according to $x_i(t) = -\sum_{j \in N(i)} W_{ij}(x_i(t) - x_j(t))$. We assume that $W_{ij} = W_{ji}$ for all $(i,j) \in E$. Letting $\mathbf{x}_F(t)$ and $\mathbf{x}_L(t)$ denote the vectors of follower and leader states, respectively, the state dynamics are given by $[\dot{\mathbf{x}}_F(t)\ \dot{\mathbf{x}}_L(t)]^T = -L[\mathbf{x}_F(t)\ \mathbf{x}_L(t)]^T$, where L is a real-valued matrix defined by $L_{ij} = -W_{ij}$ for $(i,j) \in E$, $L_{ii} = \sum_{j \in N(i)} W_{ij}$ for all $i \in V \setminus S$, and $L_{ij} = 0$ when $i \in S$.

B. Definition of Convergence Time

The following lemma, first proved in [4], gives a sufficient condition for convergence of the follower dynamics described in Section III-A to a desired value $\mathbf{x}^* \in \mathbb{R}^{n-|S|}$.

Lemma 1: Let the desired state $\mathbf{x}^*$ be given by $\mathbf{x}^* = \hat{\mathbf{x}} + x_0\mathbf{1}$, where $\hat{\mathbf{x}}$ is a known offset and x_0 is an unknown reference state, and suppose that there exists a path from each node in V to at least one leader node. Then $\lim_{t\rightarrow\infty} ||\mathbf{x}_t - \mathbf{x}^*|| = 0$.

We define the time required for the follower states to converge as follows.

Definition 1: The convergence time associated with leader set S is given by

$$t_\epsilon(S) \triangleq \max_{||\mathbf{x}(0)||=1} \min\{t : ||\mathbf{x}(t) - \mathbf{x}^*||_2 < \epsilon\}.$$

In words, the convergence time is equal to the time required for the follower agent states to reach within ϵ of their desired values, where the maximum is taken over all initial follower states.

C. Random Walks and Mixing Time

We first define the distance between two probability distributions [15].

Definition 2: Let μ and ν be probability distributions on the same underlying space Ω. The distance $||\mu - \nu||_2$ is defined by $||\mu - \nu||_2 = \frac{1}{2}\sum_{i \in \Omega} |\mu(i) - \nu(i)|^2$.

We consider a random walk on the set of nodes V with transition matrix P, where $P_{ij} > 0$ if there is an edge $(i,j) \in E$. A probability distribution π is a stationary distribution for the random walk with transition matrix P if $\pi P = \pi$. The time required for a random walk to reach its stationary distribution is denoted the mixing time.

Definition 3 (Mixing time): The mixing time of a random walk on G with transition matrix P and space of stationary distributions Π is equal to

$$t_{mix}(\epsilon) \triangleq \min_t \{\max_\mu \{\min_{\pi \in \Pi} ||\mu P^t - \pi||_2 < \epsilon\}\},$$

where μ is a probability distribution.

Formally, the mixing time is equal to the time required for the distribution of the random walk to reach any stationary distribution. While the mixing time is typically defined for random walks that have a unique stationary distribution, Definition 3 applies to any random walk with at least one stationary distribution.

D. Equivalence of Mixing and Convergence Times

The following theorem shows that the convergence time of Definition 1 is equivalent to the mixing time of a random walk. As a preliminary, it was noted in [4] that for any $\delta > 0$, $e^{-L\delta}$ is a stochastic matrix.

Theorem 1: Let $P = e^{-L^T\delta}$, where $\delta > 0$ and $t = k\delta$. Then the convergence time of Definition 1 is equivalent to the mixing time of the random walk with transition matrix P.

Proof: We have

$$\begin{aligned}\max_{\mathbf{x}(0)} ||\mathbf{x}(t) - \mathbf{x}^*||_2 &= \max_{\mathbf{x}(0)} \min_{i \in S} ||e^{-Lt}\mathbf{x}(0) - e_i^T\mathbf{x}(0)||_2 \\ &= \max_{\mathbf{x}(0)} \min_{i \in S} ||\mathbf{x}(0)^T(e^{-L^Tt} - e_i)||_2 \quad (1)\end{aligned}$$

Now, the random walk defined by matrix P has an absorbing state at each leader node; hence the set of stationary distributions is the span of $\{e_i\}_{i \in S}$. Thus, after normalizing $\mathbf{x}(0)$ to ensure that it represents a probability distribution, (1) is equivalent to $\max_{\mu} \min_{\pi \in \Pi} ||\mu P^k - \pi||$. Since these error metrics are equivalent, the times until each of them is within ϵ are also equal. ■

IV. Leader Selection in Static Networks

In this section, the problem of selecting leader agents in order to minimize convergence time is studied for the case of static network topology. We first formulate the problem of selecting a fixed set of up to k leaders in order to minimize convergence time and present leader selection algorithms. We then study the problem of selecting the minimum number of leaders in order to achieve a given bound on the convergence time.

A. Problem Formulation – Fixed Number of Leaders

The error experienced by the follower agents at time t is given by $\max_{\mathbf{x}(0)} ||e^{-Lt}\mathbf{x}(0)||_2$, which in turn is equal to $\max_{\mathbf{x}(0)} \mathbf{x}(0)^T e^{-(L^T+L)t}\mathbf{x}(0)$. This maximum is achieved when $\mathbf{x}(0)$ is chosen to be equal to the eigenvector corresponding to the largest eigenvalue of $e^{-(L+L^T)}$ [16]. This eigenvalue is in turn defined by the smallest eigenvalue of $L+L^T$. Hence minimizing the convergence time is equivalent to maximizing the smallest eigenvalue of $L+L^T$. We denote the smallest eigenvalue of $L+L^T$ as the *convergence rate*.

This optimization is carried out by choosing the leader set S, which affects the matrix L. Letting $\tilde{L}$ denote the matrix L when the leader set is empty ($S = \emptyset$), the matrix L corresponding to leader set S is obtained by setting $L_{ij} = 0$ for all $i \in S$, $j \in V$.

Let Y denote the diagonal matrix with $Y_{ii} = 0$ if $i \in S$ and $Y_{ii} = 1$ otherwise. Then $L = Y\tilde{L}$, so that $L + L^T = Y\tilde{L} + \tilde{L}Y$, where we use the fact that $\tilde{L}$ is a symmetric matrix. The requirement that only k nodes can act as leaders is equivalent to a constraint that $\mathbf{tr}(Y) \geq n-k$. Hence the problem of selecting up to k leaders in order to minimize the convergence time is given by

$$\begin{array}{lll} \underset{Y}{\text{maximize}} & \min_{\mathbf{x}(0)} \mathbf{x}(0)^T(Y\tilde{L} + \tilde{L}Y)\mathbf{x}(0) & \\ \text{s.t.} & Y_{ij} = 0 \ \forall i \neq j & \\ & Y_{ii} \in \{0,1\} \ \forall i & \\ & \mathbf{tr}(Y) \geq n-k & \end{array} \quad (2)$$

This problem can be cast in an equivalent form as

$$\begin{array}{ll} \underset{Y,s}{\text{minimize}} & s \\ \text{s.t.} & -sI \leq Y\tilde{L} + \tilde{L}Y \leq sI \\ & Y_{ij} = 0 \ \forall i \neq j \\ & Y_{ii} \in \{0,1\} \ \forall i \\ & \mathbf{tr}(Y) \geq n-k \end{array} \quad (3)$$

Due to the constraint $Y_{ii} \in \{0,1\}$, the above problem is a mixed integer program, and hence may not be solvable in polynomial time. A relaxed version of the problem is given by

$$\begin{array}{lll} \underset{Y,s}{\text{minimize}} & s & \\ \text{s.t.} & -sI \leq Y\tilde{L} + \tilde{L}Y \leq sI & (4a) \\ & Y_{ij} = 0 \ \forall i \neq j & (4b) \\ & 0 \leq Y_{ii} \leq 1 \ \forall i & (4c) \\ & \mathbf{tr}(Y) \geq n-k & (4d) \end{array} \quad (4)$$

Lemma 2: The problem formulation (4) is convex.

Proof: Convexity follows from the fact that the objective function s, as well as the constraints (4a)–(4d), are linear in Y and t. ■

B. Leader Selection Algorithms

We present two algorithms for solving (2). In the first algorithm, problem (4) is solved using semidefinite programming techniques [17], yielding a matrix denoted $\hat{Y}$. The diagonal elements of $\hat{Y}$ are then ordered from largest to smallest, with the k smallest indices of $\hat{Y}$ chosen as the leader set.

The second algorithm leverages the convex formulation of (4) using a supergradient descent approach. The following lemma is needed first.

Lemma 3: Let L be a matrix corresponding to a leader set S, and let $\mathbf{v}$ be the eigenvector of L corresponding to the smallest eigenvalue of L. Then $\mathbf{v}\mathbf{v}^T$ is a supergradient of the function $\min_{\mathbf{x}(0)} \mathbf{x}(0)^T(L+L^T)\mathbf{x}(0)$.

Proof: Provided in the appendix. ■

A supergradient descent-based algorithm can then be developed based on $\mathbf{v}$ from Lemma 3. Under an unconstrained supergradient descent algorithm, at each iteration of the algorithm, a supergradient L' is found and L is updated to $L - \tau L'$ for some $\tau > 0$.

In this case, since the formulation of (2) requires L to be of the form $Y\tilde{L} + \tilde{L}Y$, where Y is diagonal and has integer entries, the matrix $L - \tau \mathbf{v}\mathbf{v}^T$ may not be feasible because there may be multiple nonzero entries of $\mathbf{v}$. As an approximation, we instead let i denote the index such that $\mathbf{v}_i$ is maximal, and define $\tilde{\mathbf{v}}$ by $\tilde{\mathbf{v}}_i = 1$ and $\tilde{\mathbf{v}}_i = 0$. Updating $Y = Y + \tilde{\mathbf{v}}\tilde{\mathbf{v}}^T$ then results in a feasible solution to (2). This is equivalent to adding the node with largest eigenvector component to the leader set at each iteration.

The algorithm is defined formally as follows. Let S_j denote the set of leaders at the j-th iteration of the algorithm, with $S_0 = \emptyset$, and initialize $Y_0 = I_n$ (where I_n denotes the nxn identity matrix) and $L_0 = \tilde{L}$. At the j-th iteration, let $\mathbf{v}_j$ denote the eigenvector corresponding to the largest eigenvalue of L_j, and let $\tilde{\mathbf{v}}_j$ be defined as above. The set Y_j is updated according to $Y_j = Y_{j-1} + \tilde{\mathbf{v}}\tilde{\mathbf{v}}^T$, while L_j is updated to $L_j = Y_jL_j + L_jY_j$. The algorithm terminates after k iterations and returns Y_k. A pseudocode description of the algorithm is given as **Choose-k-leaders** below.

C. Problem Formulation – Minimum-Size Set of Leaders

The problem of choosing the minimum-size set of leaders in order to ensure that the convergence time is below a given threshold t^* is formulated as follows.

Algorithm Choose-k-leaders: Algorithm for selecting a set of k leaders that minimizes convergence time

```
Input: Graph topology $G = (V, E)$
Initial matrix $\tilde{L}$
Maximum number of leaders $k$
Output: Set of leaders $S$
Initialization: $S_0 \leftarrow \emptyset$, $Y_0 \leftarrow I_n$, $L_0 \leftarrow \tilde{L}$ , $j \leftarrow 0$
while $j < k$
  $\mathbf{v}_j \leftarrow$ eigenvector of $L_j$
    corresponding to largest eigenvalue
  $w_j \leftarrow \arg\max_{i \in V} (\mathbf{v}_j)_i$
  $j \leftarrow j + 1$
  $S_j \leftarrow S_{j-1} \cup \{w_j\}$
  $Y_j \leftarrow Y_{j-1} + e_{w_j} e_{w_j}^T$
  $L_j \leftarrow Y_j \tilde{L} + \tilde{L} Y_j$
end while
$S \leftarrow S_k$, return $S$
```

Algorithm Choose-minimum-leaders: Algorithm for selecting the minimum number of leaders to achieve a convergence time bound

```
Input: Graph topology $G = (V, E)$
Initial matrix $\tilde{L}$
Bound on error, $\epsilon$, time threshold $t^*$
Output: Set of leaders $S$
Initialization: $S \leftarrow \emptyset$, $Y \leftarrow I_n$, $L \leftarrow \tilde{L}$
while $\max_{\mathbf{x}(0)} \mathbf{x}(0)^T e^{-(L+L^T)t^*} \mathbf{x}(0) > \epsilon$
  $\mathbf{v} \leftarrow$ eigenvector corresponding to
    largest eigenvalue of $e^{-(L+L^T)^*}$
  $w \leftarrow \arg\max_{i \in V} (\mathbf{v})_i$
  $S \leftarrow S \cup \{w\}$
  $Y \leftarrow Y + e_w e_w^T$
  $L \leftarrow Y \tilde{L} + \tilde{L} Y$
end while
return $S$
```

$$\begin{array}{lc} \underset{Y}{\text{maximize}} & \mathbf{tr}(Y) \\ \text{s.t.} & \max_{\mathbf{x}(0)} \mathbf{x}(0)^T e^{-(L+L^T)t^*} \mathbf{x}(0) \leq \epsilon \\ & Y_{ij} = 0 \ \forall i \neq j \\ & Y_{ii} \in \{0, 1\} \ \forall i \in V \end{array} \tag{5}$$

Since the trace of Y is equal to the number of follower nodes, maximizing $\mathbf{tr}(Y)$ is equivalent to minimizing the number of leaders. The first constraint guarantees that, regardless of the initial state of the system, the error in the follower node states is within the bound ϵ by time t^*, which in turn implies that the convergence time is no more than t^*. The remaining constraints ensure that Y is a 0-1 diagonal matrix, as in (2).

A supergradient-based algorithm for approximating the solution to (5) is given as follows. At each iteration, the eigenvector $\mathbf{v}$ corresponding to the largest eigenvalue of $e^{-(L+L^T)t}$ is computed. The index i such that $\mathbf{v}_i$ is maximized is chosen and added to the leader set, with $Y = Y + e_i e_i^T$. A pseudocode description is given as algorithm **Choose-minimum-leaders**.

V. Leader Selection in Dynamic Networks

Multi-agent systems may experience changes in network topology due to failures in the communication links between agents, or may switch between predefined topologies. In this section, we extend our leader selection formulation to incorporate both of these cases and present efficient leader selection algorithms.

A. Leader Selection Under Random Failures

Under a random failure model, each link $(i, j) \in E$ fails independently with probability p. Let $\mathbf{G}$ denote the (random) graph topology under this failure model. The expected convergence time for leader set S is given by $\mathbf{E}_{\mathbf{G}}(t_{conv}(S))$. Here $\mathbf{E}_{\mathbf{G}}(\cdot)$ denotes expectation of the argument over the set of network topologies, represented by random variable $\mathbf{G}$. Minimizing the expected convergence time is equivalent to maximizing the expected convergence rate. Hence the problem of selecting up to k leaders in order to maximize the convergence rate can be formulated as

$$\begin{array}{lc} \underset{Y}{\max} & \mathbf{E}_{\mathbf{G}} \left(\min_{\mathbf{x}(0)} \mathbf{x}(0)^T (Y \tilde{L} + \tilde{L} Y) \mathbf{x}(0) \right) \\ \text{s.t.} & \mathbf{tr}(Y) \geq n - k \\ & Y_{ii} \in \{0, 1\} \ \forall i \in V \\ & Y_{ij} = 0 \ \forall i \neq j \end{array} \tag{6}$$

where the objective function corresponds to the expected value of the convergence rate, defined in Section IV, over the possible network topologies.

Since $\min_{\mathbf{x}(0)} \mathbf{x}(0)^T (Y \tilde{L} + \tilde{L} Y) \mathbf{x}(0)$ is a convex function of Y, representing the convergence rate of the MAS, the objective function of (6) is a weighted sum of convex functions and hence is convex. A supergradient approximation algorithm for (6) is defined as follows. As a first step, a supergradient of the objective function is identified.

Lemma 4: Let $\mathbf{v}_{\mathbf{G}}$ denote the eigenvector corresponding to the largest eigenvalue of $\mathbf{E}(\mathbf{L}_{\mathbf{G}})$. Then $\mathbf{v}_{\mathbf{G}}$ is a supergradient of $\mathbf{E}_{\mathbf{G}} \left(\min_{\mathbf{x}(0)} \mathbf{x}(0)^T (L + L^T) \mathbf{x}(0) \right)$.

Proof: A proof is given in the appendix. ■

Based on Lemma 4, an algorithm for solving (6) is given as follows. The leader set S is initialized to $S_0 = \emptyset$ and $Y_0 = I_n$. At the j-th iteration, the eigenvector $\mathbf{E}_{\mathbf{G}}(\mathbf{v}_{\mathbf{G}})$ is computed. The maximal component, denoted i_j, of $\mathbf{v}_{\mathbf{G}}$ is then found and added to S_j, with $Y_{i_j i_j} = 0$. The algorithm terminates after k iterations.

The dual problem of selecting the minimum number of leaders such that the expected value of t_{conv} is below a given threshold t^* can likewise be formulated as

$$\begin{array}{lc} \underset{Y}{\text{maximize}} & \mathbf{tr}(Y) \\ \text{s.t.} & Y_{ii} \in \{0, 1\} \ \forall i, Y_{ij} = 0 \ \forall i \neq j \\ & \mathbf{E}_{\mathbf{G}} \left(\max_{\mathbf{x}(0)} \{ \mathbf{x}(0)^T e^{-(L+L^T)t^*} \mathbf{x}(0) \} \right) < \epsilon \end{array} \tag{7}$$

Problem (7) can be approximately solved using the supergradient identified in Lemma 4. At each iteration, the eigenvector $\mathbf{v}_G$ is found for the largest eigenvalue of $e^{-(L(G)+L(G)^T)t}$ for each realization of G. The vector $\mathbf{v_G}$ is then taken as the expectation of $\mathbf{v}_G$. The largest component of $\mathbf{v_G}$ is then added to the leader set.

B. Leader Selection Under Switching Topologies

In MAS with switching topologies, the network topology switches among a set of topologies $\mathcal{G} = \{G_1, \ldots, G_N\}$, with corresponding weight matrices $L_1, \ldots, L_N$. In this case, the leaders are selected in order to ensure that the worst-case convergence time among all the possible leader sets is minimized. Let $u(Y)$ denote the worst-case convergence time, so that

$$u(Y) \triangleq \min_{l=1,\ldots,N} \min_{\mathbf{x}(0)} \{\mathbf{x}(0)^T(Y\tilde{L} + \tilde{L}Y)\mathbf{x}(0)\}.$$

First, the problem of selecting up to k leaders in order to minimize the worst-case convergence time is given by

$$\begin{array}{ll} \underset{Y}{\text{minimize}} & u(Y) \\ \text{s.t.} & \mathbf{tr}(Y) \geq n-k \\ & Y_{ii} \in \{0,1\}\ \forall i,\ Y_{ij} = 0\ \forall i \neq j \end{array} \quad (8)$$

Problem (8) can be converted to an optimization problem of similar to (3), and hence relaxed to a convex optimization problem analogous to (4). A supergradient for $u(Y)$ is given by the following lemma.

Lemma 5: The function $u(Y)$ has supergradient equal to $\mathbf{v}\mathbf{v}^T$, where $\mathbf{v}$ denotes the eigenvector corresponding to the smallest eigenvalue of any of the matrices $L_1, \ldots, L_N$.

Proof: The proof is given in the appendix. ■

Based on this lemma, an algorithm for approximating the solution to (8) is as follows. The leader set S is initially empty. At each iteration, the supergradient $\mathbf{v}$ is computed as in Lemma 5. The component i^* satisfying $i^* = \arg\max\{\mathbf{v}_i : i \in V\}$ is selected and added to the leader set; the algorithm then terminates after k iterations. A pseudocode description of the algorithm is given as **k-leaders-switching**.

The problem of selecting the minimum-size set of leaders in order to ensure that the convergence time for each topology in $\mathcal{G}$ is below a threshold t^* is given as

$$\begin{array}{ll} \underset{Y}{\text{maximize}} & \mathbf{tr}(Y) \\ \text{s.t.} & u(Y) < \epsilon \\ & Y_{ii} \in \{0,1\}\ \forall i \in V\ Y_{ij} = 0\ \forall i \neq j \end{array} \quad (9)$$

An approximate algorithm for computing (9), given as **Leaders-switching-bound**, is as follows. First, at each iteration, $\mathbf{v}$ is determined according to Lemma 5. The node i^* is then selected such that $\mathbf{v}_i$ is maximized. The set S is updated to $S \cup \{i^*\}$. The algorithm terminates when the objective function of (9) is less than ϵ.

Algorithm k-leaders-switching: Algorithm for selecting a set of k leaders to minimize the worst-case convergence time

```
Input: Set of topologies $\mathcal{G} = \{G_1, \ldots, G_N\}$
Weight matrices $\tilde{L}_1, \ldots, \tilde{L}_N$
Maximum number of leaders, $k$
Output: Set of leaders $S$
Initialization: $S_0 \leftarrow \emptyset$, $Y_0 \leftarrow I_n$,
$L_{l,0} \leftarrow \tilde{L}_l$, $l = 1, \ldots, M$, $j \leftarrow 0$
while $j < k$
  $\mathbf{v}_j \leftarrow$ eigenvector corresponding to largest eigenvalue
    of $L_1, \ldots, L_N$
  $w_j \leftarrow \arg\max_{i \in V} (\mathbf{v}_j)_i$
  $j \leftarrow j + 1$
  $S_j \leftarrow S \cup \{w_j\}$
  $Y_j \leftarrow Y_j + e_{w_j} e_{w_j}^T$
  $L_{l,j} \leftarrow Y_j \tilde{L} + \tilde{L} Y_j\ \forall l = 1, \ldots, N$
end while
$S \leftarrow S_k$, return $S$
```

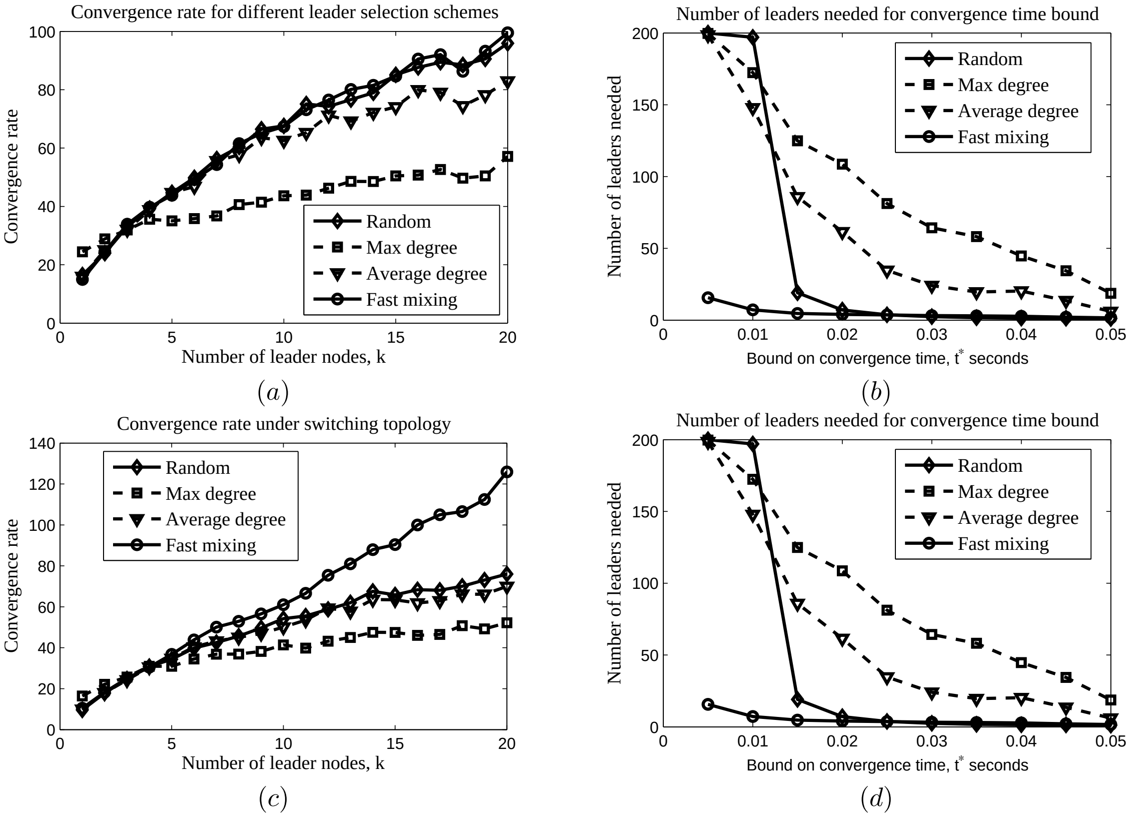
Algorithm Leaders-switching-bound: Algorithm for selecting the minimum number of leaders in order to ensure that $t_{conv} \leq t^*$ for each network topology in $\mathcal{G}$

```
Input: Set of topologies $\mathcal{G} = \{G_1, \ldots, G_N\}$
Weight matrices $\tilde{L}_1, \ldots, \tilde{L}_N$
Bound on error, $\epsilon$, time threshold $t^*$
Output: Set of leaders $S$
Initialization: $S_0 \leftarrow \emptyset$, $Y_0 \leftarrow I_n$
$L_{l,0} \leftarrow \tilde{L}_l$, $l = 1, \ldots, M$
while $\max_{\mathbf{x}(0)} \mathbf{x}(0)^T e^{-(L+L^T t)} \mathbf{x}(0) \geq \epsilon$
  $\mathbf{v} \leftarrow$ eigenvector corresponding to largest eigenvalue of
    $L_1, \ldots, L_N$
  $w \leftarrow \arg\max_{i \in V} (\mathbf{v}_j)_i$
  $j \leftarrow j + 1$
  $S \leftarrow S \cup \{w\}$
  $Y \leftarrow Y + e_w e_w^T$
  $L_l \leftarrow Y\tilde{L} + \tilde{L}Y\ \forall l = 1, \ldots, N$
end while
return $S$
```

VI. Simulation Study

A MAS consisting of $n = 100$ agents was simulated using Matlab. It was assumed that agents are deployed uniformly at random over a square region of dimensions 1000m x 1000m. Agents i and j shared a link (i,j) if they were within $r =$ 300m of each other. Link weights L_{ij} for the follower agents were chosen uniformly at random within the interval $[0, 50]$, with $L_{ij} = L_{ji}$.

We simulated two cases of MAS. In the first case, a static network with fixed topology was assumed. In the second case, a network that switched between three independently chosen topologies was considered. For both cases, we simulated leader selection using fast mixing, as well as selection of random nodes, nodes with high degree, and nodes with average degree as leaders. Each data point represents an



(a) (b) (c) (d)

Fig. 1. Simulation study of leader selection for smooth convergence. (a) In a static network, the convergence rate is consistently higher for our fast mixing approach. Note that maximum degree-based selection has the lowest convergence rate. (b) The fast mixing method requires fewer leaders than random or degree-based selection in order to achieve an error of $\epsilon = 1$ for different values of t^*. (c) When the network switches between three randomly generated topologies, the fast mixing method outperforms random and degree-based leader selection algorithms as the number of possible leaders increases. (d) The fast mixing approach requires fewer leaders than the random and degree-based algorithms under switching topologies in order to achieve a given bound on the convergence time.

average over 150 trials.

Figure 1(a) compares the convergence rate $\max_{\mathbf{x}(0)} \mathbf{x}(0)^T(L + L^T)\mathbf{x}(0)$ of our fast mixing-based approach and the three other selection algorithms. We observe that our mixing-based approach slightly outperforms the other schemes, while random leader selection also provides strong performance. Selection of nodes based on average degree performs significantly worse than both random and fast mixing algorithms, while selection of maximum degree nodes provides the slowest convergence rate.

The problem of selecting the minimum number of leaders in order to meet a given bound on the convergence time is demonstrated in Figure 1(b), with $\epsilon = 1$. The fast mixing approach typically requires fewer than 10 leaders to guarantee a given convergence time, compared to the remaining schemes which sometimes require all nodes to act as leaders in order to guarantee a given convergence time.

The fast mixing approach also improves the performance of networks with switching topologies (Figure 1(c)), especially when the number of leaders increases. For example, when 20 nodes can act as leaders, the convergence rate for the fast mixing algorithm is more than twice that of the next-largest algorithm, random leader selection. As in the static case, random leader selection provides a faster convergence rate on average than either degree-based method. Similarly, fewer leaders are required to meet a bound on the convergence time using fast mixing (Figure 1(d)). For example, in order to ensure that the convergence time is below 0.05, only 6 leaders are needed under the fast mixing approach, while 12 leaders are needed under random selection and 24 under maximum degree-based selection.

VII. Conclusions

In this paper, we studied leader selection in multi-agent systems in order to minimize the time required for the follower agents to converge to their desired states. We first proved that the convergence time is equivalent to the mixing time of a random walk on the underlying graph. We then formulated two leader selection problems: first, selection of a fixed set of up to k leaders in order to minimize convergence time, and second, selection of the minimum number of leaders in order to achieve a bound on the convergence time. For both problems, we derived efficient algorithms for leader selection, based on supergradient descent, for MAS with static topologies, as well as MAS with random link failures and switching between predefined topologies. We compared our fast mixing approach to other leader selection algorithms, including random leader selection, selection of high degree nodes as leaders, and selection of average degree nodes as leaders, through simulation study.

In our future work, we will investigate leader selection in networks where the topology varies arbitrarily in time due to agent mobility. We will also study joint minimization of convergence time based on both leader selection and selection of the link weights between agents.

References

[1] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, 2004.

[2] D. Stipanovic, G. Inalhan, R. Teo, and C. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.

[3] J. Acebron, L. Bonilla, C. Vicente, F. Ritort, and R. Spigler, "The kuramoto model: A simple paradigm for synchronization phenomena," *Reviews of modern physics*, vol. 77, no. 1, p. 137, 2005.

[4] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988–1001, 2003.

[5] F. Pasqualetti, S. Martini, and A. Bicchi, "Steering a leader-follower team via linear consensus," *Hybrid Systems: Computation and Control*, pp. 642–645, 2008.

[6] A. Rahmani, J. Meng, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2010.

[7] Y. Liu, J. Slotine, and A. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[8] C. Lin, "Structural controllability," *Automatic Control, IEEE Transactions on*, vol. 19, no. 3, pp. 201–208, 1974.

[9] F. Lin, M. Fardad, and M. Jovanovic, "Algorithms for leader selection in large dynamical networks: Noise-corrupted leaders," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[10] M. Fardad, F. Lin, and M. Jovanovic, "Algorithms for leader selection in large dynamical networks: Noise-free leaders," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[11] A. Clark and R. Poovendran, "A submodular optimization framework for leader selection in linear multi-agent systems," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[12] A. Clark, L. Bushnell, and R. Poovendran, "Leader selection for minimizing convergence error in leader-follower systems: A supermodular optimization approach," *Submitted to 10th IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2012.

[13] ——, "On leader selection for performance and controllability in multi-agent systems," *Submitted to the 51st IEEE Conference on Decision and Control (CDC)*, 2012.

[14] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, 2006.

[15] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM review*, pp. 667–689, 2004.

[16] R. Horn and C. Johnson, *Matrix analysis*. Cambridge Univ Pr, 1990.

[17] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

Appendix

In order to prove Lemma 3, we first recall the definition of supergradient.

Definition 4: Let $f : S^n \rightarrow \mathbf{R}$ be a real-valued function taking positive semidefinite matrices as input. Then a matrix $Z \in \mathbf{S}^n$ is a supergradient of f at point $X_0 \in \mathbf{S}^n$ if, for every matrix $Q \in \mathbf{S}^n$, $f(X_0 + Q) \leq f(X_0) + \mathbf{tr}(X_0 Q)$.

Proof: [Proof of Lemma 3] The goal is to show that $\mathbf{v}\mathbf{v}^T$ is a subgradient of $f(L) = \frac{1}{2}\min_{\mathbf{x}(0)} \mathbf{x}(0)^T(L + L^T)\mathbf{x}(0)$. We have

$$\begin{aligned} f(L_0 + Q) &= \min_{\mathbf{x}(0)} \mathbf{x}(0)^T(L_0 + Q + L_0^T + Q)\mathbf{x}(0) \\ &\leq \mathbf{v}^T(L_0 + L_0^T)\mathbf{v} + \mathbf{v}^T(Q)\mathbf{v} \\ &= f(Y_0) + \mathbf{tr}(\mathbf{v}^T Q \mathbf{v}) \\ &= f(Y_0) + \mathbf{tr}(Q\mathbf{v}\mathbf{v}^T) \end{aligned}$$

$\blacksquare$

Proof: [Proof of Lemma 4] The function $f(L)$ in this case is given by

$$f(L) \triangleq \mathbf{E}_{\mathbf{G}}\left(\min_{\mathbf{x}(0)} \mathbf{x}(0)^T(L + L^T)\mathbf{x}(0)\right) \qquad (10)$$

This yields

$$\begin{aligned} f(L_0 + Q) &= \mathbf{E}_{\mathbf{G}}\left(\min_{\mathbf{x}(0)} \mathbf{x}(0)^T(L_0 + L_0^T)\mathbf{x}(0)\right) \\ &\leq \mathbf{E}_{\mathbf{G}}(\mathbf{v}_{\mathbf{G}}^T(L_0 + L_0^T)\mathbf{v}_{\mathbf{G}}) + \mathbf{E}_{\mathbf{G}}(\mathbf{v}_{\mathbf{G}}^T Q \mathbf{v}_{\mathbf{G}}) \\ &= f(L_0) + \mathbf{tr}(Q\mathbf{v}_{\mathbf{G}}\mathbf{v}_{\mathbf{G}}^T) \end{aligned}$$

$\blacksquare$

Proof: [Proof of Lemma 5] Let $f(L)$ be defined by

$$f(L) = \min_{l=1,\ldots,N} \min_{\mathbf{x}(0)} \mathbf{x}(0)^T(L + L^T)\mathbf{x}(0).$$

Then

$$\begin{aligned} f(L_0 + Q) &= \min_{l=1,\ldots,N} \min_{\mathbf{x}(0)} \mathbf{x}(0)^T(L + L^T + Q)\mathbf{x}(0) \\ &\leq \mathbf{v}^T(L + L^T)\mathbf{v} + \mathbf{v}^T Q \mathbf{v} \\ &= f(L_0) + \mathbf{tr}(Qvv^T) \end{aligned}$$

as desired. $\blacksquare$