A Submodular Optimization Framework for Leader Selection in Linear Multi-Agent Systems

Andrew Clark and Radha Poovendran
Network Security Lab (NSL), EE Dept., University of Washington, Seattle, WA, 98195, USA

***Abstract*— A broad class of multi-agent systems are leader-follower systems, in which the states of a set of leader agents are used to influence the states of the remaining agents. In this paper, we study the problem of choosing specific agents that will act as leaders in order to optimize system performance. We show that for a diverse set of networked multi-agent systems, including systems with constant and time-varying topologies, the leader selection problem can be studied using a submodular optimization framework. We further show that the problems of choosing a predefined number of leader nodes, as well as choosing both the number and specific nodes to satisfy a performance requirement, can be formulated within the proposed submodular optimization framework. We derive analytical performance bounds for the proposed solutions for linear multi-agent systems with static as well as time-varying network topologies. Numerical illustration is provided for the proposed approach.**

I. Introduction

A wide variety of processes, including robotic navigation and tracking [1], formation control of UAVs [2], and distribution estimation and synchronization in sensor networks [3], can be modeled as multi-agent systems. Typical multi-agent systems consist of autonomous agents that are interconnected, with each agent influencing the states and dynamics of the surrounding agents.

A broad class of multi-agent systems can be further classified as leader-follower systems, in which a set of leader nodes are used to control the states of the remaining nodes. Examples include localization in sensor networks based on relative distance to GPS-equipped anchor nodes, as well as unmanned vehicle formations in which some vehicles are controlled remotely. In such systems, the states of the leader nodes are treated as inputs, while each follower node's state is a weighted average of its neighbors' states (the *linear agreement protocol*) [4]. Through the agreement protocol, inputs from the leader nodes propagate through the system and influence the states of the follower nodes.

Propagation of leader input may be disrupted by a number of factors. First, since the agreement protocol relies on each node measuring the states of its neighbors, the protocol can be affected by measurement errors. Such errors may lead to application-specific failures, such as deviations from the expected formation or inaccurate distributed estimates. Secondly, since multi-agent systems may consist of mobile agents communicating over lossy links, they are prone to changes in network topology. Topology changes arise from *link failures*, in which certain links become unavailable over time, *switching topologies* [4], in which the network switches between predefined topologies, and *arbitrary time-varying topologies*, in which the network topology evolves stochastically over time.

Leader agents are traditionally selected based on majority voting or node degree-based approaches [5]. However, these heuristics may not be sufficient to ensure controllability or optimal performance of the system.

In [6], control-theoretic analysis of leader-follower systems was introduced. In a control framework, the follower states are treated as the plant, while the leader states provide external input. Under this model, the dynamics of the plant are governed by the graph Laplacian of the subgraph formed by the follower nodes. In [6], it was shown that, for a given set of leaders, the follower system is controllable if and only if the eigenvalues of the Laplacian for the graph consisting of the followers are distinct. It was also shown that choosing highly connected nodes as leaders may result in an uncontrollable system.

In the presence of noisy links, agent states will deviate from their desired operating points. In [7], the authors derive the error covariance associated with a given node's state as a function of the network topology and the error distribution of each link. The error covariance of a node's state is shown to be related to the graph effective resistance between that node and the leader nodes. The results of [6], [7] show that the controllability and noise rejection properties of a system depend on the choice of leader nodes.

In this paper, we study the problem of optimal leader selection in linear multi-agent systems with noisy links. We show that, for both static and time-varying topologies, the problem of selecting leaders to minimize the total error covariance of the system can be posed as a submodular optimization framework.

We make the following contributions:

- We consider the problem of *optimal leader selection* in linear multi-agent systems with noisy links and map the problem to a framework based on *submodular optimization*.
- In order to show that the leader selection problem is submodular, we observe that the effective resistance of a graph is proportional to the commute time of a random walk on the graph, which we prove is a supermodular function. This in turn implies that effective resistance is supermodular. We then use the known result that the error covariance of a multi-agent system is equal to its effective resistance to show that the error covariance is supermodular.

- We formulate two types of leader selection problems within our optimization framework and show that these problems are dual to one another: (a) when a fixed number of leader nodes must be chosen to minimize error, and (b) when the system must achieve a given error bound using the smallest possible number of leaders.
- We show that a broad class of system types can be analyzed under our framework, including (a) networks with static topologies, (b) networks experiencing random link outages, (c) networks that switch between a finite, predefined set of topologies, and (d) networks with arbitrary time-varying topologies. We present efficient solution algorithms for each case.
- We derive analytical bounds on the optimality of the results returned by our algorithms, and also give an empirical comparison of our scheme with random and degree-based leader selection algorithms.

This paper is organized as follows. Section II presents assumptions and necessary background. Section III introduces the submodular optimization framework for the case of a static network topology. Section IV discusses leader selection in time-varying multi-agent systems. Section V contains an experimental evaluation of our approach. Section VI provides our conclusions and directions for future work.

TABLE I

NOTATION USED IN THIS PAPER

Notation	Definition
n	Number of nodes
V	Set of nodes
E	Set of links
m	Number of links
$N(i)$	Set of neighbors of node i
A	Incidence matrix of graph $G = (V, E)$
A_f	Incidence matrix of follower nodes
A_l	Incidence matrix of leader nodes
x_i	State of node with index i
S	Set of leader nodes
k	Number of leader nodes
x_l	Vector of leader node states
x_f	Vector of follower node states
ϵ_{ij}	Measurement error for link (i, j)
r	Vector of relative measurements between nodes
$R(i, j)$	Variance of link error ϵ_{ij}
$R(S, u)$	Variance of overall measurement error for node u when leader set is S
$R(S)$	$\sum_{u \in V\setminus S} R(S, u)$
$R(S\|G)$	$\sum_{u \in V\setminus S} R(S, u)$ for topology G
α	Maximum tolerated measurement error

II. BACKGROUND AND PRELIMINARIES

In this section, the system model is defined. The connection between the assignment of leader nodes and the robustness of the network to measurement noise is explored. Background on submodular functions is also given.

A. System model

A multi-agent system of n agents with graph structure $G = (V, E)$ is considered, where $V = \{1, \ldots, n\}$ is the set of the agents. The pair (i, j) is in the link set E if agent i's state influences the dynamics of agent j. It is assumed that links are undirected[1]. The set $N(i)$ of i's *neighbors* consists of the nodes j with $(i, j) \in E$. Let A denote the incidence matrix of G, i.e. the $n \times m$ matrix where the e-th column has a 1 in the i-th row, a -1 in the j-th row, and zeros elsewhere, when $e = (i, j)$.

It is assumed that a subset S of agents acts as *leaders*. All non-leader nodes are *followers*. The incidence matrix can therefore be decomposed into A_f, consisting of the rows corresponding to the follower nodes, and A_l, consisting of the rows corresponding to the leader nodes.

B. System dynamics

Each agent with index i has a corresponding state x_i. The follower nodes are assumed to have state dynamics

$$\dot{x_i} = \sum_{j \in N(i)} (x_i - x_j) \tag{1}$$

referred to as the *agreement protocol*.

The agreement dynamics defined in (1) rely on agent i having an accurate measurement of $r_{ij} = (x_i - x_j)$. Suppose that the measurements r_{ij} are corrupted by additive noise ϵ_{ij} with known covariance $R(i, j)$, so that

$$r_{ij} = x_i - x_j + \epsilon_{ij} \tag{2}$$

The corresponding matrix equation is

$$r = Ax + \epsilon \tag{3}$$

where r and ϵ are vectors in $\mathbf{R}^m$ describing the relative measurements and noise, respectively. Letting x_f denote the vector of follower node states and x_l denote the vector of leader node states, and using the decomposition of A into leader and follower components, (3) becomes

$$r = A_f x_f + A_l x_l + \epsilon \tag{4}$$

It is assumed that the states x_l are known and that the follower nodes rely on a linear estimator to find the correct values of x_f. The following theorem in [7] describes the error of the resulting estimate in terms of the Laplacian of the graph.

Theorem 1: Let $L_f = A_f W A_f^T$, where W is a diagonal matrix with the values $R(i, j)^{-1}$ as entries. Then the covariance of the estimation error for node u is given by $(L_f^{-1})_{uu}$.

In this context, we denote the value $(L_f^{-1})_{uu}$ as $R(S, u)$. Let $R(S) \triangleq \sum_{u \in V\setminus S} R(S, u)$. Then $R(S)$ is a measure of the total estimation error within the system, and hence also the system's deviation from its ideal state. Note that $R(S)$ can be computed by finding the trace of L_f^{-1}, which can be performed in polynomial time for a given graph. In what follows, we let $R(S|G)$ denote the value of $R(S)$ for a specific topology G.

[1] This assumption comes from the fact that the noise characteristics of each link are symmetric.

C. Graph effective resistance

In this section, the connection between the error covariance $R(S,u)$ and the effective resistance of a graph is explored. The following lemmas prove that $R(S,u)$ is equivalent to the effective resistance of an equivalent circuit. It is then shown, through a generalization of results in [8] to sets, that the effective resistance is proportional to the commute time of a random walk on a graph.

Definition 1: Consider an electrical network with graph structure $G = (V,E)$, where V is the set of nodes and a resistance $R(i,j)$ exists across each edge $(i,j) \in E$. Let u be a node, and let $S \subseteq V$ be a set of nodes with $u \notin S$. Define J_u to be the current exiting node u when the nodes in S are grounded (i.e. have voltage 0) and node u has voltage 1. Then the effective resistance $\tilde{R}(S,u)$ between u and S is equal to $1/J_u$.

The following lemma describes the connection between the effective resistance of a graph G and the error experienced by the nodes in the multi-agent system.

Lemma 1: The functions $R(S,u)$ and $\tilde{R}(S,u)$ are equivalent.

A proof of Lemma 1 can be found in the appendix. Interpreting R according to Definition 1 allows use of the probabilistic definition of effective resistance, which is based on random walks on graphs. In this definition, the probability of transitioning from node u to node v is equal to

$$P(u,v) = \frac{C(u,v)}{\sum_{t \in N(u)} C(u,t)} \tag{5}$$

where $C(u,v) = R(u,v)^{-1}$ is the conductance of link (u,v). The following lemma establishes the connection between random walks and effective resistance.

Lemma 2: Let $\kappa(S,u)$ be the expected time for a random walk originating at node u to reach a node in S and return to u. Then $R(S,u) = \frac{1}{2}\left(\sum_{(i,j)\in E} C(i,j)\right)^{-1} \kappa(S,u)$.

The proof of Lemma 2 can be found in the appendix.

D. Review of submodular functions

A submodular function is defined as follows.

Definition 2: Let V be a finite set, and let $f : 2^V \rightarrow \mathbf{R}$ be a function that maps subsets of V to real numbers. Then f is *submodular* if, for any sets A and B satisfying $A \subseteq B$ and any element $j \notin B$,

$$f(A \cup \{j\}) - f(A) \geq f(B \cup \{j\}) - f(B) \tag{6}$$

Eq. (6) states that, if A is a subset of B, then adding an element j to A results in a larger increase in f than adding j to B. The following useful fact about submodular functions appears in [9].

Lemma 3: Let $f_1, \ldots, f_r : 2^V \rightarrow \mathbf{R}$ be submodular functions, and let $a_1, \ldots, a_r$ be nonnegative constants. Then $f(S) = \sum_i a_i f_i(S)$ is a submodular function.

A function f is *supermodular* if the function $-f$ is submodular.

III. Problem Formulation – Static Network

In this section, the optimal leader selection problem is formulated for the case where the edge set E is constant in time. As noted in the introduction, there are two possibilities for this problem. In the first, due to resource constraints, only a fixed number of nodes may be selected as leaders. The problem is to choose the leader set in order to minimize the total error. In the second, there is a performance requirement, expressed as a bound on the error, and the problem is to find a set of nodes of minimum size that meets the requirement.

A. Case I – Choosing a fixed number of leader nodes

The problem of choosing a given fixed number k of leader nodes in order to minimize the overall error $R(S)$ is formulated as

$$\begin{array}{ll} \text{minimize} & R(S) = \sum_{u \in V \setminus S} R(S,u) \\ \text{s.t.} & |S| \leq k \end{array} \tag{7}$$

Theorem 2: For a fixed element $u \in V \setminus S$, $R(S,u)$ is a supermodular function of S.

Proof: By Lemma 2, $R(S,u)$ is proportional to $\kappa(S,u)$, which is the expected time for a random walk starting at u to reach any point in S and then return to u. Thus, by Lemma 3, proving that $R(S,u)$ is supermodular is equivalent to proving that $\kappa(S,u)$ is supermodular. By Definition 2, $\kappa(S,u)$ is a supermodular function of S if and only if, for any sets A and B with $A \subseteq B$ and for any $j \notin B$,

$$\kappa(A,u) - \kappa(A \cup \{j\}, u) \geq \kappa(B,u) - \kappa(B \cup \{j\}, u) \tag{8}$$

Consider the quantity $\kappa(A,u) - \kappa(A \cup \{j\}, u)$. Define $A' = A \cup \{j\}$, and define T_{Au} and $T_{A'u}$ to be the (random) times for a random walk to reach A (respectively A') and return to u. Then by definition, $\kappa(A,u) = \mathbf{E}(T_{Au})$ and $\kappa(A',u) = \mathbf{E}(T_{A'u})$. This implies $\kappa(A,u) - \kappa(A',u) = \mathbf{E}(T_{Au} - T_{A'u})$.

Let $\tau_j(A)$ denote the event where the random walk reaches node j before any of the nodes in A. Further, let U_{jAu} be the expected time for the random walk to travel from j to A and then to u, while U_{ju} is the expected time to travel directly from j to u.

$$\begin{aligned} \kappa(A,u) - \kappa(A',u) &= \mathbf{E}(T_{Au} - T_{A'u} | \tau_j(A)) Pr(\tau_j(A)) \quad (9) \\ &\quad + \mathbf{E}(T_{Au} - T_{A'u} | \tau_j(A)^c) Pr(\tau_j(A)^c) \\ &= (U_{jAu} - U_{ju}) Pr(\tau_j(A)) \quad (10) \end{aligned}$$

noting that, if the walk reaches A first, then T_{Au} and $T_{A'u}$ are equal. Hence (8) becomes

$$(U_{jAu} - U_{ju}) Pr(\tau_j(A)) \geq (U_{jBu} - U_{ju}) Pr(\tau_j(B)) \tag{11}$$

Now, note that if $A \subseteq B$, then a random walk that reaches set A automatically reaches set B as well; conversely, if the walk has not reached set B, then it has not reached A either. This implies that $U_{jAu} \geq U_{jBu}$ and $Pr(\tau_j(A)) \geq Pr(\tau_j(B))$. Hence (8) holds, thus proving Theorem 2. ■

$R(S) = \sum_{u \in V \setminus S} R(S,u)$ is therefore a sum of supermodular functions, and so $R(S)$ is supermodular as well by Lemma 3.

B. Optimization algorithm

Since $R(S)$ is supermodular, (7) defines a *submodular maximization problem*. Although the submodular maximization problem is known to be NP-hard [10], there exist polynomial-time algorithms that find a set S^* such that $R(S^*)$ is within a factor of $(1 - 1/e)$ of the optimum value, denoted R^*.

The algorithm for solving (7) is as follows. Let S_i^* denote the set of leader nodes at the i-th iteration of the algorithm. S_0^* is initialized to $\emptyset$. At the i-th iteration of the algorithm, the element $s_i^* \in V$ is found such that $\{R(S_{i-1}^*) - R(S_{i-1}^* \cup \{s_i^*\})\}$ is maximized. S_i^* is then updated to $(S_{i-1}^* \cup \{s_i^*\})$. The algorithm terminates when either $R(S_i^*) = R(S_i^* \cup \{j\})$ for all j, or when $i = k$ (i.e., when the number of leaders is equal to k), whichever is reached first. A pseudocode description of the algorithm is given in the figure below.

```
CHOOSE_$k$_LEADER_NODES
Input: $G = (V, E)$, covariances $R(i, j)$
       Maximum number of leader nodes $k$
Output: Set of leader nodes $S^*$
Initialization: $S^* \leftarrow \emptyset$, $i \leftarrow 0$
while $i \leq k$
    $s_i^* \leftarrow \arg\max_{j \in V \setminus S} \{R(S^*) - R(S^* \cup \{j\})\}$
    if $\{R(S^*) - R(S^* \cup \{s_i^*\})\} \leq 0$
        return $S^*$; exit
    else
        $S^* \leftarrow S^* \cup \{s_i^*\}$
        $i \leftarrow i + 1$
    end
end
return $S^*$; exit
```

Fig. 1. Algorithm for solving (7).

The following theorem analyzes the computation time and performance of this algorithm, making use of the fact that (7) is a submodular maximization problem.

Theorem 3: Define R_{max} to be

$$R_{max} \triangleq \max_i \sum_{u \in V} R(i, u) \tag{12}$$

and let R^* be the optimal value of (7). Then the following are true for the algorithm in Figure 1:

(a) The algorithm terminates in polynomial time in k and n.

(b) S^* satisfies

$$R(S^*) \leq \left(1 - \left(\frac{k-1}{k}\right)^k\right) R^* + \frac{1}{e} R_{max} \tag{13}$$

$$\approx \left(1 - \frac{1}{e}\right) R^* + \frac{1}{e} R_{max} \tag{14}$$

(c) If $|S^*| < k$, then $R(S^*) = R^*$, in other words S^* is the optimal solution to (7).

Proof: The algorithm in Figure 1 requires $O(kn)$ computations of the effective resistance, which, as noted in Section II-B, can be performed in polynomial time. This establishes (a). (b) follows from rearranging terms in Proposition 4.3 of [9], and the fact that $(1 - \frac{1}{k})^k$ converges to $1/e$ as k increases. (c) follows from the fact that, if $R(S_i^*) = R(S_i^* \cup \{j\})$ for all j, then the effective resistance cannot be decreased by adding any element j to S_i^*. Hence S_i^* must be the optimum. ■

C. Case II – Choosing nodes to achieve an error bound

When the system is required to operate below a given error bound α, the problem of choosing a minimal set of leaders that achieves this bound can be stated as

$$\begin{array}{cc} \text{minimize} & |S| \\ \text{s.t.} & \sum_{u \in V \setminus S} R(S, u) = R(S) \leq \alpha \end{array} \tag{15}$$

Note that, for any $\alpha \geq 0$, there exists at least one S meeting the condition $R(S) \leq \alpha$, namely the set $S = V$.

The supermodularity of $R(S)$ enables efficient solution of (15) as well. The algorithm is as follows. The set of leaders is initialized to $S_0^* = \emptyset$. As with the algorithm in Figure 1, the node s_i^* that maximizes $\{R(S_{i-1}^*) - R(S_{i-1}^* \cup \{s_i^*\})\}$ is added at the i-th iteration. The difference is that, instead of terminating after k iterations, the algorithm will not terminate until $R(S^*) \leq \alpha$. A detailed description of the algorithm can be found in Figure 2.

```
CHOOSE_LEADER_NODES_ERROR_LIMIT
Input: $G = (V, E)$, covariances $R(i, j)$
Error at termination $\alpha$
Output: Set of leader nodes $S^*$
Initialization: $S^* \leftarrow \emptyset$, $error \leftarrow \alpha + 1$
while $error > \alpha$
    $s^* \leftarrow \arg\max_{j \in V \setminus S} \{R(S^*) - R(S^* \cup \{s^*\})\}$
    if $(R(S^*) - R(S^* \cup \{s^*\})) \leq 0$
        return $S^*$; exit
    else
        $S^* \leftarrow S^* \cup \{s^*\}$
        $error \leftarrow R(S^*)$
    end
end
return $S^*$; exit
```

Fig. 2. Algorithm for solving (15).

The following theorem gives bounds on the optimality of the set S^* returned by the above algorithm.

Theorem 4: Let k^* be the optimum value of (15). The algorithm described above terminates in polynomial time in n. If the algorithm terminates after step k, so that $|S^*| = k$, then

$$\frac{k}{k^*} \leq 1 + \log\left\{\frac{R_{max}}{R(S_{k-1}^*)}\right\} \tag{16}$$

holds, where R_{max} is as defined in Theorem 3.

Proof: Optimality follows from supermodularity of R and Theorem 1 of [11]. In the worst case, the algorithm will not terminate until $S = V$, i.e. after n iterations. This will require $O(n^2)$ computations of effective resistance, each of which is polynomial in n. ■

IV. Problem Formulation – Time-varying Network

In this section, leader selection in multi-agent systems with time-varying topologies is considered, including networks with random link failures, networks that switch between a set of predefined topologies, and networks where the topology varies arbitrarily in time.

A. Case I – Random link failures

The random link failure model captures the fact that, while the relative positions of nodes may remain the same, the wireless links between them may become unavailable due to noise or interference. The random link failure model assumes *uniform, independent* link failures, so that at a given time each link $(i,j) \in E$ is available with probability p. In this case, the resulting optimization problem is

$$\begin{array}{cc} \text{minimize} & \mathbf{E}(R(S)) \\ \text{s.t.} & |S| \leq k \end{array} \tag{17}$$

This is also a submodular optimization problem, as shown by the following lemma.

Lemma 4: $\mathbf{E}(R(S))$ is a supermodular function of S.

Proof: Let $\mathcal{G}$ denote the set of possible network topologies. Then

$$\mathbf{E}(R(S)) = \sum_{G \in \mathcal{G}} Pr(G) R(S|G) \tag{18}$$

By Theorem 2, $R(S|G)$ is a supermodular function of S. Since $Pr(G) \geq 0$ for all G, $\mathbf{E}(R(S))$ is a nonnegative weighted sum of supermodular functions, and so is supermodular by Lemma 3. ■

As a result, (17) can be solved using a polynomial number of computations of $\mathbf{E}(R(S))$. Computing $\mathbf{E}(R(S))$, however, requires evaluation of the sum in (18), which has a number of terms exponential in the number of links, requiring Monte Carlo approximation in the general case. When the number of link failures is small, so that $p \approx 1$, a gradient-based bound can also be used.

Lemma 5: Recall that $R(S)$ is equal to $Tr(L_f^{-1})$, where L_f is defined in Theorem 1. Let $\tilde{L_f} = L_f - \Delta$ be the corresponding matrix for topology $\tilde{G}$, which is obtained by removing links from G. Note that, since links are bidirectional, Δ is symmetric. Then, when $||\Delta||_2 \leq \delta$,

$$R(S|\tilde{G}) = Tr(\tilde{L}_f^{-1}) \leq Tr(L_f^{-1}) + \frac{\delta}{\lambda_{min}(L_f)^2} \tag{19}$$

where $\lambda_{min}(L_f)$ is the smallest eigenvalue of L_f.

Proof: Letting Δ represent the deviation from L_f caused by link failures, $R(S)$ becomes

$$\begin{aligned} Tr((L_f - \Delta)^{-1}) &\approx Tr(L_f^{-1}) + Tr(L_f^{-1} \Delta L_f^{-1}) \qquad (20) \\ &\leq Tr(L_f^{-1}) + Tr(U\Lambda^{-1}U^T \Delta U \Lambda^{-1} U^T) \\ &\leq Tr(L_f^{-1}) + \sup_{\Delta} Tr(U\Lambda^{-1}U^T \Delta U \Lambda^{-1} U^T) \end{aligned}$$

where $L_f = U\Lambda U^T$ is the eigen-decomposition of L_f. The upper bound occurs when $\Delta = U\Omega U^T$ for some positive semidefinite diagonal matrix Ω. This implies that

$$\begin{aligned} Tr((L_f - \Delta)^{-1}) &\leq Tr(L_f^{-1}) + Tr(\Lambda^{-1}\Omega\Lambda^{-1}) \qquad (21) \\ &\leq Tr(L_f^{-1}) + \frac{\delta}{\lambda_{min}^2} \qquad (22) \end{aligned}$$

as desired. ■

This upper bound can then be used as a worst-case value for $R(S)$ in the presence of link failures. Since $R(S)$ represents the variance of system error, using an upper bound as an objective function leads to a conservative network design. Problem (17) can be solved by modifying the algorithm in Figure 1 so that, at Line 3, $\{\mathbf{E}(R(S)) - \mathbf{E}(R(S \cup \{j\}))\}$ is maximized instead of $\{R(S) - R(S \cup \{j\})\}$.

B. Case II – Switching between predefined topologies

Suppose that the system switches between topologies $G_1, \ldots, G_r$. Two leader selection criteria can then be considered: the average error $\frac{1}{r}\sum_i R(S|G_i)$, and the worst-case error $\max_i R(S|G_i)$. Since the average error is a weighted sum of supermodular functions, it is supermodular by Lemma 3, and so algorithms analogous to those in Figure 1 and 2 can be used.

For the worst-case error, the function $\max_i R(S|G_i)$ is not supermodular[2]. Still, it can be shown that the following problem can be reformulated as a submodular optimization problem.

$$\begin{array}{cc} \text{minimize} & |S| \\ \text{s.t.} & R(S|G_i) \leq \alpha, \quad i = 1, \ldots, r \end{array} \tag{23}$$

Define the function $F_i(S) \triangleq \max\{R(S|G_i), \alpha\}$.

Lemma 6: $F_i(S)$ is supermodular.

The proof can be found in the appendix.

Lemma 7: Let $F(S) = \frac{1}{r}\sum_i F_i(S)$. Then F is a supermodular function of S. Furthermore, $F(S) \leq \alpha$ if and only if $R(S|G_i) \leq \alpha$ for all i.

Proof: The supermodularity of F follows from Lemmas 3 and 6. Since $F_i(S) \geq \alpha$ for all i, $F(S) \leq \alpha$ if and only if $F_i(S) = \alpha$ for all i. This in turn is equivalent to $R(S|G_i) \leq \alpha$ for all i. ■

Problem (23) can then be reformulated as

$$\begin{array}{cc} \text{minimize} & |S| \\ \text{s.t.} & F(S) \leq \alpha \end{array} \tag{24}$$

Figure 2 can be modified to solve this problem by maximizing over $\{F(S) - F(S \cup \{j\})\}$ in Line 3.

C. Case III – Arbitrarily varying topology

In this section, networks that experience random changes in topology over time are considered. In such networks, it may be difficult to choose an appropriate set of leader nodes in the absence of any information about future network topologies. Instead, it is assumed that the set of leader nodes

[2] Algorithms for minimizing such functions have been proposed [12]. Due to space constraints, they are not discussed here.

is periodically computed, leading to an *online submodular optimization problem* [10] of the form

$$\begin{array}{ll} \text{minimize} & \sum_t R(S_t|G_t) \\ \text{s.t.} & |S_t| \leq k, t = 1, \ldots, T \end{array} \tag{25}$$

where G_t is the topology for the t-th time interval.

The method for choosing a set of leader nodes S_t^* for the t-th time interval is as follows. Consider the algorithm of Figure 1. Since G_t is unknown and random, the node j that maximizes $\{R(S_{t,i}^*) - R(S_{t,i}^* \cup \{j\})\}$ at the i-th iteration of the algorithm is also random. Let

$$\pi_{ti}(l) = Pr(\arg\max_j \{R(S_{t,i}^*) - R(S_{t,i}^* \cup \{j\})\} = l) \tag{26}$$

Then for time interval t, instead of selecting $s_{t,i}^*$ deterministically as in Line 3 of Figure 1, $s_{t,i}^*$ is selected probabilistically with distribution π_{ti}.

In general, the exact values of π_{ti} will not be known during leader selection. To address this, an online learning technique is used to estimate π_{ti} based on observations from the previous $t-1$ time intervals. Under this approach, a set of weights $\mathbf{w}_{t1}, \ldots, \mathbf{w}_{tk}$ is maintained, where $\mathbf{w}_{ti}$ is a vector in $\mathbf{R}^n$ with $\mathbf{w}_{ti}(j)$ representing the weight assigned to choosing node j as the i-th leader during interval t. Define

$$s_{t,i}^{opt} \triangleq \arg\max_j \{R(S_{t,i-1}^*|G_t) - R(S_{t,i-1}^* \cup \{j\}|G_t)\} \tag{27}$$

In other words, the best possible choice of $s_{t,i}^*$ for the topology G_t. Then define the *loss* $l_{t,i,j}$ associated with choosing $s_{t,i}^* = j$ to be

$$l_{t,i,j} \triangleq 1 - \frac{R(S_t^*|G_t) - R(S_t^* \cup \{j\}|G_t)}{R(S_t^*|G_t) - R(S_t^* \cup \{s_{t,i}^{opt}|G_t)\}} \tag{28}$$

At the end of the t-th time interval, the value of $\mathbf{w}_{t,i}(j)$ is updated to

$$\mathbf{w}_{t+1,i}(j) = \beta^{l_{t,i,j}} \mathbf{w}_{t,i}(j) \tag{29}$$

where $\beta \in (0,1]$ is a system parameter that can be tuned to adjust the performance of the learning algorithm. Eq. (29) penalizes node j for suboptimal performance during interval t. By increasing the value of β, nodes experiencing higher losses $l_{t,i,j}$ will be much less likely to be selected during the $(t+1)$-th time interval. $\mathbf{w}_{t+1,i}$ is then normalized to obtain the estimated distribution $\pi_{t+1,i}$. This process is described in detail in Figure 3.

In analyzing this approach, the total error $\sum_t R(S_t|G_t)$ can be compared to the error achievable when all T topologies are known in advance. The following theorem gives a bound on the difference between these two.

Theorem 5: Suppose that the algorithm described in Figure 3 is executed for T rounds, and let $G_1, \ldots, G_T$ be the topologies during those intervals. Let R_{max} be defined as in Theorem 3. Define the error K to be

$$K \triangleq (1 - 1/e) \sum_{t=1}^{T} R(S_t|G_t) - \left(\max_{|S|=k} \left\{ \sum_{t=1}^{T} R(S|G_t) \right\} \right) \tag{30}$$

Then $K \leq O(\sqrt{R_{max} kT \log n})$.

```
ADAPTIVE_LEADER_SELECTION
Input: Current weights $\mathbf{w}_{t,1}, \ldots, \mathbf{w}_{t,k}$
$G_t = (V, E_t)$, covariances $R_t(i,j)$
Parameter $\beta \in (0,1]$
Maximum number of leader nodes $k$
Current set of leader nodes $S_t^*$
Output: Updated weights $\mathbf{w}_{t+1,1}, \ldots, \mathbf{w}_{t+1,k}$
        Set of leader nodes $S_{t+1}^*$
Initialization: $S_{t+1}^* \leftarrow \emptyset$
for $i = 1, \ldots, k$
    $s_{t,i}^{opt} \leftarrow \arg\max_j \{R(S_{t,i-1}^*) - R(S_{t,i-1}^* \cup \{j\})\}$
    for $j = 1, \ldots, n$
        $l_{t,i,j} \leftarrow 1 - \frac{R(S_{t,i-1}^*) - R(S_{t,i-1}^* \cup \{j\})}{R(S_{t,i-1}^*) - R(S_{t,i-1}^* \cup \{s_{t,i}^{opt}\})}$
        $\mathbf{w}_{t+1,i}(j) \leftarrow (\mathbf{w}_{t,i}(j))\beta^{l_{t,i,j}}$
    end
    $\pi_{t+1,i} \leftarrow \mathbf{w}_{t+1,i}/\mathbf{1}^T \mathbf{w}_{t+1,i}$
    Choose $s_{t+1,i}^*$ randomly with distribution $\pi_{t+1,i}$
    $S_{t+1}^* \leftarrow S_{t+1}^* \cup \{s_{t+1,i}^*\}$
end
```

Fig. 3. One iteration of leader selection algorithm for arbitrary time-varying topology.

Proof: $R(S|G_1), \ldots, R(S|G_T)$ is a sequence of supermodular functions bounded above by R_{max}. Then, by Lemma 4 of [10],

$$\begin{aligned} (1 - 1/e) \sum_{t=1}^{T} R((S_t^*|G_t) - \left(\max_{|S|=k} \left\{ \sum_{t=1}^{T} R(S|G_t) \right\} \right) \\ \leq O(\sqrt{R_{max} kT \log n}) \end{aligned} \tag{31}$$

as desired.

■

V. Experimental Evaluation

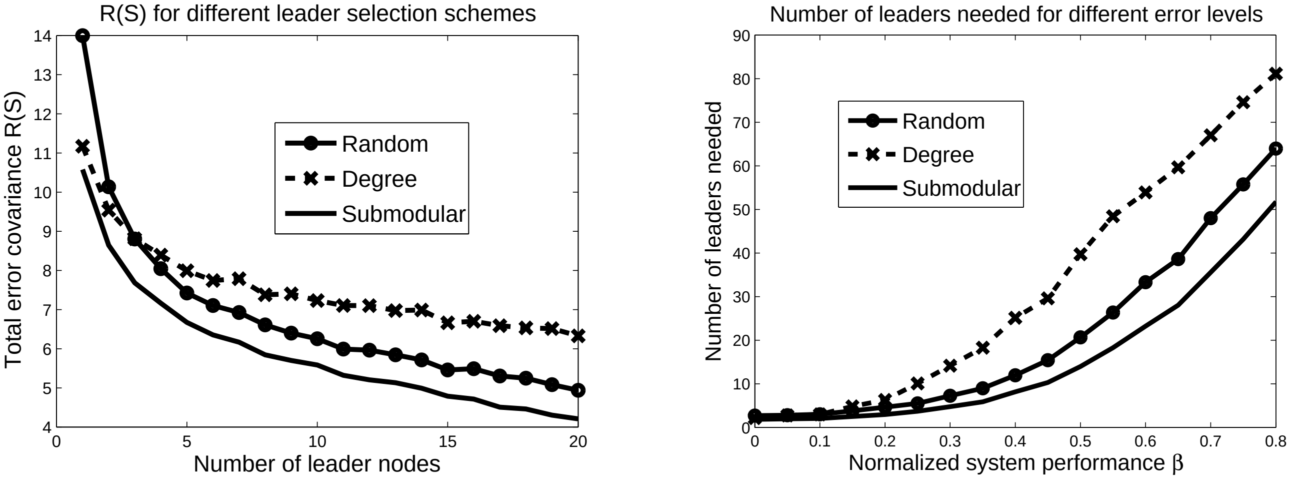
In this section, experimental evaluation of the leader selection procedure is provided.

A. Experimental setup

A network of 100 agents was simulated using Matlab. Agents were assumed to be deployed uniformly over a region of 1000x1000m. Each agent's neighbor set consisted of all agents within 300 meters. It was assumed that link noise had covariance proportional to the distance between the nodes comprising the link. The proposed submodular optimization approach was compared to two other leader selection algorithms. In the first algorithm, leaders were chosen at random from the set of nodes. In the second, the nodes with highest degree were selected as leaders.

B. Simulation Results

Evaluation of minimizing error subject to a constraint on the number of nodes is shown in Figure 4a. The proposed submodular optimization approach outperforms both other heuristics. Furthermore, selecting high-degree nodes as leaders results in higher error then both other schemes. This is consistent with the fact, observed in previous work [6],



(a) (b)

Fig. 4. Comparison of leader selection schemes. (a) The total error covariance $R(S)$ resulting from selection of a fixed number of leader nodes. Submodular optimization represents an improvement over random and degree based heuristics. (b) The number of nodes required to achieve a given network performance under the algorithm in Figure 2. As expected, for networks with high performance requirement β, the submodular approach requires fewer leader nodes.

that multi-agent systems may be uncontrollable when highly-connected nodes are chosen as leaders.

Figure 4b evaluates the algorithm for minimizing the number of nodes required to achieve a given error bound. The values on the horizontal axis are equal to $\beta = (\alpha_{max} - \alpha)/\alpha_{max}$, where α ranges from 2 to $\alpha_{max} = 10$, so that the required error bound decreases as β increases. The figure shows that, for high values of β, corresponding to stringent conditions on tolerated error, the submodular optimization approach requires far fewer leader nodes than both the degree-based and random heuristics. For $\beta = 0.7$, for example, the submodular scheme requires 35 leaders, while the random and degree-based algorithms require 48 and 66 leaders, respectively.

VI. Conclusion

In this paper, the problem of optimal leader selection in linear multi-agent systems with noisy links was mapped to a submodular optimization framework. The applicability of submodular optimization techniques was demonstrated by first observing that the effective resistance of a graph is proportional to the commute time of a random walk on the graph, which we proved to be supermodular. By exploiting the fact that the total error covariance of the system is equal to its effective resistance, it was shown that minimizing the total error covariance is a submodular optimization problem.

Two versions of the optimal leader selection problem were considered. In the first case, a fixed number of leaders are chosen to minimize the overall error, while in the second case a minimum number of leader nodes is chosen to achieve a performance requirement. These problems can be interpreted as dual to one another, and it was shown that both problems can be addressed using our proposed framework.

Moreover, techniques for solving both problems under different system conditions, including static networks, networks with link failures, networks that switch between pre-defined topologies, and networks with arbitrary time-varying topologies, were derived using our framework. For each case, bounds on the performance of the resulting leader selection were given.

The response of the system to link noise is one criterion for selecting leader nodes, but other criteria, such as controllability, can be used. Our future work will develop a framework for choosing leader nodes based on their controllability as well as response to noise.

References

[1] J. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4. IEEE, 2002, pp. 2864–2869.

[2] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1465–1476, 2004.

[3] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, 2006, pp. 168–176.

[4] 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.

[5] T. Borsche and S. Attia, "On leader election in multi-agent control systems," in *Control and Decision Conference (CCDC), 2010 Chinese*. IEEE, 2010, pp. 102–107.

[6] H. Tanner, "On the controllability of nearest neighbor interconnections," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3. IEEE, 2005, pp. 2467–2472.

[7] P. Barooah and J. Hespanha, "Graph effective resistance and distributed control: Spectral properties and applications," in *Decision and Control, 2006 45th IEEE Conference on*. IEEE, 2007, pp. 3479–3485.

[8] L. Lovász, "Random walks on graphs: A survey," *Combinatorics, Paul Erdos is Eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[9] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[10] M. Streeter and D. Golovin, "An online algorithm for maximizing submodular functions," 2007.

[11] L. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[12] A. Krause, H. Mcmahan, C. Guestrin, and A. Gupta, “Selecting observations against adversarial objectives,” in *In NIPS, 2007a*. Citeseer, 2007.

APPENDIX

In this appendix, proofs of Lemmas 1 and 2 from Section II, as well as Lemma 6 from Section IV, are given.

Proof: (Proof of Lemma 1) Let v denote the vector of node voltages, and let J denote the vector where J_i is the net current exiting from node i. Further, let matrix $L = AWA^T$, where A is the incidence matrix of the graph and W is defined as in Theorem 1. Assume, without loss of generality, that the first $n - |S|$ entries of v and J correspond to the followers, while the last $|S|$ entries are the leaders. L can be decomposed into sub-matrices

$$L = \left(\begin{array}{c|c} L_{11} & L_{12} \\ \hline L_{21} & L_{22} \end{array}\right) \tag{32}$$

where L_{11} has dimension $(n - |S|) \times (n - |S|)$. Note that L_{11} is equal to the matrix L_f defined in Theorem 1.

Kirchoff’s Laws, together with Ohm’s Law, imply that $Lv = J$. Let J_f and v_f denote the follower node currents and voltages, respectively. Then the decomposition of L and the fact that v_j is assumed to be zero for all $j \in S$ implies that $J_f = L_f v_f$. This in turn implies that $v_f = L_f^{-1} J_f$. Multiplying both sides of this equation on the left by e_u^T yields $e_u^T v_f = e_u^T L_f^{-1} J_f$. Since $v_u = 1$, this implies that

$$(L_f^{-1})_{u1} J_1 + \cdots + (L_f^{-1})_{u(n-|S|)} J_{n-|S|} = 1 \tag{33}$$

Now, by Kirchoff’s current law, the net current flowing out of each node $i \in V \setminus S$, where $i \neq u$, must be zero. Hence (33) reduces to $(L_f^{-1})_{uu} J_u = 1$, implying that $J_u^{-1} = (L_f^{-1})_{uu}$. Since $\tilde{R}(S,u) = J_u^{-1}$ and $R(S,u) = (L_f^{-1})_{uu}$, this establishes the lemma. ■

In order to prove Lemma 2, the following intermediate result is needed.

Lemma 8: Define $\phi_{Su}(v)$ to be the voltage of node v when a unit voltage is applied to node u and a voltage of 0 is applied to the nodes in S, as above. Let $\tilde{\phi}_{Su}(v)$ be the probability that a random walk originating at v reaches u before reaching any node in the set S. Then $\phi_{Su}(v) = \tilde{\phi}_{Su}(v)$.

Proof: (Proof of Lemma 8) Let $C(u,v)$ be defined as in the problem statement, and let

$$P(u,v) = \frac{C(u,v)}{\sum_{t \in N(u)} C(u,t)} \tag{34}$$

By definition,

$$\tilde{\phi}_{Su}(v) = \sum_{t \in N(v)} P(v,t) \tilde{\phi}_{Su}(t) \tag{35}$$

Eq. (35) implies that $\tilde{\phi}$ is *harmonic* in P. If it can be shown that ϕ is also harmonic in P, then the Uniqueness Principle [8] implies that $\phi = \tilde{\phi}$, since they agree on the boundary points S and u.

By Kirchoff’s Current Law, the net current exiting node v must be zero for any v outside the boundary. By Ohm’s Law, this is equivalent to

$$\sum_{t \in N(v)} C(v,t)(\phi_{Su}(t) - \phi_{Su}(v)) = 0 \tag{36}$$

Rearranging terms yields

$$\phi_{Su}(v) = \sum_{t \in N(v)} P(v,t) \phi_{Su}(t) \tag{37}$$

In other words, ϕ_{Su} is also harmonic in P. Hence the two functions must be equivalent. ■

Proof: (Proof of Lemma 2) The effective resistance is defined to be the inverse of the total current exiting node u. By Kirchoff’s Law, this is equal to

$$R(S,u) = \left(\sum_{v \in N(u)} C(u,v)(1 - \phi_{Su}(v)) \right)^{-1} \tag{38}$$

At the same time, the probability that a random walk starting at u reaches S before returning to u is equal to $1 - \sum_{v \in N(u)} \phi_{Su}(v) P(u,v)$, where $P(u,v)$ is defined in Eq. (34). By Proposition 2.3 of [8],

$$1 - \sum_{v \in N(u)} \phi_{Su}(v) P(u,v) = \frac{2 \sum_{(s,t) \in E} C(s,t)}{\kappa(S,u) \sum_{v \in N(u)} C(u,v)} \tag{39}$$

Rearranging terms yields

$$\kappa(S,u) = \frac{2 \sum_{(s,t) \in E} C(s,t)}{\sum_{v \in N(u)} C(u,v)(1 - \phi_{Su}(v)} \tag{40}$$

$$= 2 \left(\sum_{(s,t) \in E} C(s,t) \right) R(S,u) \tag{41}$$

as desired. ■

Proof: (Proof of Lemma 6) Let f be a supermodular function satisfying $f(A) \geq f(B)$ for any $A \subseteq B$ (we call this the *monotonicity property*). It is enough to show that for any α, $F(S) \triangleq \max\{f(S), \alpha\}$ is supermodular. The proof uses of the fact that a function F is supermodular if and only if ([9])

$$F(A) + F(B) \leq F(A \cap B) + F(A \cup B) \tag{42}$$

as well as the fact that $f(A \cup B) \leq (f(A), f(B)) \leq f(A \cap B)$. There are four cases.

Case 1: $\alpha < f(A \cup B)$: In this case, (42) follows from the supermodularity of f.

Case 2: $f(A) < \alpha$, $f(B) > \alpha$: Under this case, (42) is equivalent to $f(B) \leq f(A \cap B)$, which follows from the monotonicity property. The case where $f(A) > \alpha$ and $f(B) < \alpha$ is similar.

Case 3: $f(A) < \alpha$, $f(B) < \alpha$, $f(A \cap B) > \alpha$: For this case, (42) is equivalent to $f(A \cap B) \geq \alpha$, which is true by assumption.

Case 4: $f(A \cap B) \leq \alpha$: Eq. (42) is trivially satisfied in this case. ■