Convex relaxations and Gramian rank constraints for sensor and actuator selection in networks

Tyler Summers and Iman Shames

Abstract— **We propose a convex relaxation heuristic for sensor and actuator selection problems in dynamical networks using Gramian metrics. We also propose heuristic algorithms to enforce a rank constraint on the Gramian that can be used in conjunction with combinatorial greedy algorithms and the convex relaxation. This allows selection of sensor or actuator sets that optimize an objective function while preserving a certain amount of observability or controllability throughout the state space, combining previous methods that focus exclusively on either rank or Gramian metrics. We illustrate and compare the greedy and convex relaxation heuristics in several numerical examples involving random and regular networks.**

I. Introduction

Recent advances in sensing technologies are revealing unprecedented data streams about diverse physical, technological, and social network dynamics. Further advances in computation, communication, and actuation technologies are increasing our ability to use this data to understand and manipulate network dynamics using feedback control techniques. This includes many important applications such as efficient, reliable, and secure delivery of energy in power grids; throughput maximization and congestion control in the Internet and in transportation networks; unraveling natural designs and ultimately synthesizing new functions in biological networks; and understanding and controlling contagion processes in social networks like financial defaults, disease, and unhealthy behaviors.

A basic problem in the context of large dynamical networks is the selection or placement of sensors and actuators in the network to provide desirable observability and controllability properties. The notions of controllability and observability have been recognized for decades as fundamental properties of dynamic systems, but there has been significant renewed interest in quantifying these notions in large networks. The recent literature can be broadly categorized by the metrics used to quantify controllability or observability and the methods used for selecting sensors and actuators.

Liu et al. and much follow up work [14], [21], [8], [16], [30], [17], [22], [19] focus on a binary quantification using the classical Kalman rank condition and the notion of structural controllability that reduces the problem to a purely combinatorial one. A minimal set of actuators that render a network structurally controllable can be obtained using a standard maximum matching algorithm for graphs. However, the focus on Kalman rank and structural controllability excludes other important considerations; there are many alternative controllability metrics that provide a much richer quantification and can be optimized by appropriate actuator selections.

Other recent work has considered problems of placing actuators to maximize these alternative controllability metrics. One line the work uses Gramian-based metrics that quantify controllability in terms of input energy required for state transfer [25], [26], [18], [7], [28], [6]. Combinatorial greedy algorithms can be used to obtain effective actuator selections, and submodularity properties that provide theoretical performance guarantees have been discovered in many cases. Other work uses metrics based on associated optimal feedback control problems with greedy algorithms [24] and optimization and convex relaxation techniques [13], [4], [20], [15], [9]. Although heuristic selections for these metrics in general do not come with theoretical performance guarantees, good actuator selections can be obtained. Further work on related network topology design problems involving leader selection, coherence, rigidity, and user-interfaces can be found in [3], [5], [27], [23], [29].

The main contributions of the present paper are threefold. First, we present a convex relaxation for Gramian-based metrics that provides a performance bound on the original selection problem and gives several heuristic methods for providing a good approximate solution. The relaxation results in a semidefinite program for several interesting metrics of the Gramian. This method complements previous results using the greedy algorithm and submodularity properties, in which a worst case approximation bound is obtained, by obtaining a best case performance bound. Second, we propose several modified greedy algorithms that allow a rank constraint on the Gramian to be enforced in various ways. This effectively brings together previous methods that focus exclusively on either rank or Gramian metrics. Third, we illustrate and compare the greedy and convex relaxation heuristics in several numerical examples involving random and regular networks. We observe that neither method produces superior actuator selections in all instances and that the greedy algorithm scales to larger network sizes. We focus here on controllability and actuator selection, but all results have analogous counterparts and interpretations for observability and sensor selection.

The rest of the paper is structured as follows. Section II presents set function and mixed integer optimization formulations of the actuator placement problem. Section III presents the convex relaxation and greedy heuristics that

T. Summers is with the Department of Mechanical Engineering at the University of Texas at Dallas, email: tyler.summers@utdallas.edu. I. Shames is with the Department of Electrical Engineering at the University of Melbourne, email: iman.shames@unimelb.edu.au.

can be used to obtain approximate solutions. Section IV presents numerical examples that illustrate and compare the greedy and convex relaxation heuristics in random and regular network. We give concluding remarks in Section V.

II. Problem Statement and Convex Relaxation

A. Network model and problem statement

Consider the linear time-invariant dynamical system modeling the network dynamics

$$\dot{x}(t) = Ax(t) + B_0 u(t), \tag{1}$$

where $x(t) \in \mathbf{R}^n$ is the network state and $u(t) \in \mathbf{R}^m$ is an external input. The dynamics matrix $A \in \mathbf{R}^{n\times n}$ is assumed to be stable, and the input matrix $B_0 \in \mathbf{R}^{n\times m}$ corresponds to a (possibly empty, i.e., $m = 0$) set of existing actuators. Let $V = \{b_1, ..., b_M\}$ be a finite set of vectors $b_i \in \mathbf{R}^n$ that corresponds to possible placements of additional actuators into the system.

We consider the set function optimization problem

$$\begin{array}{ll} \underset{S\subseteq V}{\text{maximize}} & f(S) \\ \text{subject to} & |S| = k \\ & (A, B_S) \quad \text{controllable} \end{array} \tag{2}$$

where $f : 2^V \to \mathbf{R}$ is a set function that quantifies the controllability of the pair (A, B_S), which we assume can be evaluated efficiently, and $B_S = [B_0 \;\; b_i]$, $b_i \in S$, i.e., we want to choose a k-element subset of V to maximize f subject to the system being controllable.

Following [25], [26], [18], [7], [28], [6], we consider metrics associated with the controllability Gramian. For $S \subseteq V$ we associate the controllability Gramian

$$X_S = \int_0^\infty e^{At} B_S B_S^T e^{A^T t} dt, \tag{3}$$

which is the unique positive semidefinite solution to the Lyapunov equation

$$AX_S + X_S A^T + B_S B_S^T = 0. \tag{4}$$

Various scalarizations of the Gramian, e.g., $\mathbf{tr} X_S$, $\mathbf{tr}[(X_S)^{-1}]$, $\log\det X_S$, or $\lambda_{min}(X_S)$, quantify in various ways the energy required to move the system around in the state space.

The set function optimization problem (2) is a difficult combinatorial optimization problem. For small problems, it can be solved by brute force checking of all subsets. For large problems without the controllability constraint, certain combinatorial structure in f, viz. modularity or submodularity, allows efficient optimization or approximation guarantees using a simple greedy algorithm [25].

The set function optimization problem (2) can be expressed as a mixed-integer optimization problem with data $A \in \mathbf{R}^{n\times n}$, $b_1, ..., b_M$, and k and variables $X = X^T \in \mathbf{R}^{n\times n}$ and $z \in \{0, 1\}^M$

$$\begin{array}{ll} \underset{X,z}{\text{maximize}} & g(X) \\ \text{subject to} & AX + XA^T + \sum_{i=1}^M z_i b_i b_i^T = 0 \\ & X \succeq 0, \quad z_i \in \{0, 1\}, \quad \mathbf{1}^T z = k \\ & \text{rank}(X) = n, \end{array} \tag{5}$$

where the objective function is a spectral measure of the Gramian variable X.

B. Convex relaxation

By removing the rank constraint and replacing the Boolean constraints $z_i \in \{0, 1\}$ with the convex constraints $z_i \in [0, 1]$, we obtain the relaxation

$$\begin{array}{ll} \underset{X,z}{\text{maximize}} & g(X) \\ \text{subject to} & AX + XA^T + \sum_{i=1}^M z_i b_i b_i^T = 0 \\ & X \succeq 0, \quad z_i \in [0, 1], \quad \mathbf{1}^T z = k. \end{array} \tag{6}$$

This turns out to be a convex optimization problem when the objective function f is one of several interesting concave functions the Gramian variable, e.g., $g(X) = \mathbf{tr}(X)$, $g(X) = -\mathbf{tr}(X^{-1})$, $g(X) = \log\det(X)$, or $g(X) = \lambda_{min}(X)$. These can be expressed as semidefinite programs and can be solved efficiently using interior point methods. The solution of the relaxed problem does not immediately give a solution to the original problem, but the optimal objective value of the relaxed problem gives an upper bound on the original problem and can serve as a quality indicator for any heuristic method. In particular, if any heuristic method yields a solution whose objective value is close to the optimal objective value of the relaxed problem, then the solution is close to optimal.

C. Selection heuristics based on the convex relaxation

There are several heuristics for obtaining selections from the convex relaxation.

1) Sparsity-inducing regularization: First, one can introduce a sparsity-inducing regularizer into (7) (and drop the constraint $\mathbf{1}^T z = k$, which is unnecessary here for obtaining a selection) and solve

$$\begin{array}{ll} \underset{X,z}{\text{maximize}} & g(X) + \lambda \|z\|_1 \\ \text{subject to} & AX + XA^T + \sum_{i=1}^M z_i b_i b_i^T = 0 \\ & X \succeq 0, \quad z_i \in [0, 1]. \end{array} \tag{7}$$

The parameter λ is used to trade off sparsity of z with the controllability metric f. As λ is made larger, z will tend to have fewer nonzero elements. If exactly k actuators are to be added, bisection can be performed on λ.

2) Directly from the solution of the relaxation (7)*:* A second simple method to generate a heuristic selection is as follows. Let X^* and z^* denote an optimizer of (7). Let $z^*_{i1}, ..., z^*_{iM}$ denote the elements of z^* sorted in descending order, with ties broken arbitrarily. The heuristic selection is $S_{relax} = \{i_1, ..., i_k\}$, corresponding to the indices of the k largest elements of z^*.

3) Probabilistic: A final heuristic is to view $z/(\mathbf{1}^T z)$ as a probability distribution over the set of possible actuator selections. One can then use this distribution to sample heuristic subsets that may yield good objective values.

The convex relaxation complements recent work using the greedy algorithm and submodularity properties [25], [28], [6]. The greedy methods give a worst case approximation guarantee in certain cases, whereas here we get a best case performance bound. However, as with previous work using the greedy algorithm, the rank constraint is not guaranteed to be satisfied by the heuristic selections based on the convex relaxation. In the next section, we will discuss ways that the rank constraint could be enforced while optimizing a Gramian metric.

D. Dual problems

We now formulate a dual problem of (7) for the case where $g(X) = \log\det X$. The dual problem is

$$\begin{array}{ll} \underset{\Lambda,\psi,\phi}{\text{minimize}} & \log\det(\Lambda A + A^T\Lambda)^{-1} - n + k\phi + \mathbf{1}^T\psi \\ \text{subject to} & \phi + b_i^T\Lambda b_i + \psi_i \geq 0, \quad i = 1, ..., M \\ & \psi \geq 0 \end{array} \tag{8}$$

with variables $\Lambda = \Lambda^T \in \mathbf{R}^{n\times n}$, $\psi \in \mathbf{R}^M$, and $\phi \in \mathbf{R}$. This can be interpreted as finding a quadratic Lyapunov function associated with the system $\dot{z}(t) = A^T z(t)$ that has a dissipation function with minimum volume level sets and contains at least $M - k$ of the points b_i. Dual problems and associated interpretations can also be obtained for other objective functions.

III. Greedy Gramian Metric Optimization Under Rank Constraints

In this section we consider the problem of optimizing a certain Gramian metric subject to a rank constraint. Recall that a linear dynamical system with given dynamics and input matrices is controllable if and only if the associated controllability Gramian X is full rank, i.e., $\text{rank}(X) = n$.

We have the following assumptions.

Assumption 1: There exists a set $\mathcal{M}^\star \subseteq \{1, \ldots, M\}$ such that the solution of $AX + XA^T + \sum_{i\in\mathcal{M}^\star} b_i b_i^T = 0$ is full-rank.

Assumption 2: There is no $i, j = 1, \ldots, n$ such that $\lambda_i(A) \neq -\lambda_j(A)$ where $\lambda_i(A)$ and $\lambda_j(A)$ are eigenvalues of A.

Lemma 1: Consider Assumptions 1 and 2. Let X_i satisfy

$$AX_i + X_iA^T + b_ib_i^T = 0 \tag{9}$$

where b_i is the i-th column of B. Moreover, assume that for some $i, j \in \mathcal{M}$, $\text{rank}(X_i) \geq \text{rank}(X_j)$ and $\text{rank}(X_i + X_j) = \text{rank}(X_i)$. Then $\text{rank}(\sum_{i\in\mathcal{M}\setminus\{j\}} X_i) = n$.

Proof: First, in the light of Assumption 2 note that each X_i, $i \in \mathcal{M}$ is the unique solution to (10) and is positive semidefinite. Additionally because of Assumption 1 $\text{rank}(\sum_{i\in\mathcal{M}} X_i) = n$. Let $\overline{X} \triangleq \sum_{i\in\mathcal{M}\setminus\{i,j\}} X_i$. It is enough to show $u^T(X_i + \overline{X})u > 0$ for all $u \in \mathbb{R}^n$. Assume there is a $v \in \mathbb{R}^n$ such that $v^T(X_i + \overline{X})v = 0$. We know $u^T(X_i + X_j + \overline{X})u > 0$ for all $u \in \mathbb{R}^n$. Thus, $v^T X_j v > 0$ and $v^T X_i v = 0$. On the other hand, let $r_i \triangleq \text{rank}(X_i)$. Moreover, $u^T(X_i + X_j)u > 0$ if $u^T X_i u > 0$ or $u^T X_j u > 0$. We know $u^T X_i u > 0$ for all u in the r_i-dimensional space spanned by r_i eigenvectors of X_i associated with its nonzero eigenvalues. Hence, $u^T(X_i + X_j)u > 0$ for all u in the same space, and since $\text{rank}(X_i + X_j) = r_i$ there is no u such that $u^T X_i u = 0$ and $u^T X_j u > 0$. Thus there is no v such that $v^T X_j v > 0$ and $v^T X_i v = 0$ and it completes the proof. ■

To satisfy the rank constraint for controllability, we would like to find $z \in \{0, 1\}^M$ such that $X^\star$, the solution to

$$AX^\star + X^\star A^T + \sum_{i=1}^{M} z_i b_i b_i^T = 0 \tag{10}$$

with b_i being the i-th column of B, satisfies $\text{rank}(X^\star) = n$. Moreover, it is desired that $\sum_{i=1}^M z_i$ is minimized. This can be posed as the following optimization problem:

$$\begin{array}{ll} \underset{X,z,k}{\text{minimize}} & k \\ \text{subject to} & AX + XA^T + \sum_{i=1}^{M} z_i b_i b_i^T = 0 \\ & \text{rank}(X) = n, \quad z_i \in \{0, 1\}, \quad k = \sum_{i=1}^{M} z_i \end{array} \tag{11}$$

Although this is known to be an extremely difficult problem [17], Algorithm 1 proposes a greedy heuristic.

Algorithm 1 A Greedy Algorithm for (12)

Require: A, B
- $X \leftarrow 0$, $\overline{\mathcal{S}} \leftarrow \{1, \ldots, M\}$, $\mathcal{S} \leftarrow \emptyset$
- X_i satisfies $AX_i + X_iA^T + b_ib_i^T = 0$, $\forall i \in \{1, \ldots, M\}$
- **while** $\text{rank}(X) < n$ **do**
  - $i^\star \in \underset{i\in\overline{\mathcal{S}}}{\text{argmax}} \quad [\text{rank}(X + X_i) - \text{rank}(X)]$
  - $X \leftarrow X + X_{i^\star}$
  - $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^\star\}$
  - $\overline{\mathcal{S}} \leftarrow \overline{\mathcal{S}} \setminus \{i^\star\}$
- **end while**
- $X^\star \leftarrow X$, $\mathcal{S}^\star \leftarrow \mathcal{S}$

Proposition 1: Under Assumptions 1 and 2, Algorithm 1 returns a feasible solution at most after M steps.

Proof: Without loss of generality assume $g(X_1) \leq \cdots \leq g(X_M)$. First, note that the problem is feasible at each iteration l, if $\text{rank}(\sum_{i\in\mathcal{S}} X_i) = n$ at the start of the iteration. Again without loss of generality assume that up to step i, $n > \text{rank}(X_1 + \cdots + X_{i-1}) > \text{rank}(X_1 + \cdots + X_{i-2})$. However, at the i-th step $\text{rank}(X_1+\cdots+X_i) = \text{rank}(X_1+\cdots+X_{i-1})$, that results in the exclusion of $\{i\}$ from $\mathcal{M}$. Nevertheless, due to Lemma 1, $\text{rank}(\sum_{i\in\mathcal{S}=\mathcal{M}\setminus\{i\}} X_i) = n$. Reapplying Lemma 1 at each step l such that $\text{rank}(X_1+\cdots+X_l) = \text{rank}(X_1+\cdots+X_{l-1})$ ensures that at each step there is a set $\mathcal{J} \in \overline{\mathcal{S}}$ such that $\text{rank}(\sum_{i\in\mathcal{S}\bigcup\mathcal{J}} X_i) = n$ and in the worst case scenario, $\mathcal{J} = \{M\}$ which means the rank condition $\text{rank}(X) = n$ is satisfied at the last evaluation and the algorithm terminates at the M-th step. ∎

Remark 1: For the case where $g(X)$ is a modular set function [26], it holds that $g(\sum_{i\in\mathcal{S}^\star} X_i) \leq g(\sum_{i\in\overline{\mathcal{M}}} X_i)$ where $\mathcal{S}^\star$ is obtained from the algorithm and $\overline{\mathcal{M}} \subset \mathcal{M}$ such that $|\overline{\mathcal{M}}| = |\mathcal{S}^\star|$ and $\text{rank}(\sum_{i\in\overline{\mathcal{M}}} X_i) = n$.

Remark 2: Assuming that X_i are given, the worst case computational complexity of Algorithm 1 is M^2n^3.

Remark 3: The computations in each step of Algorithm 1 are amenable to parallel computations. Each of the Lyapunov equations to find X_i can be solved independently of the other ones using parallel algorithms [12]. In turn, the singular value decomposition required to check the rank increase can be done in parallel for each X_i.

In general, to connect back to our original problem statement, we would like to maximize a performance metric of the controllability Gramian while finding a small set of actuators that provides controllability. For the rest of this section we consider the trace of the controllability Gramian as this performance function, though any efficiently computable function can be used. This problem can be formulated as a multi-objective optimization problem of the following form:

$$\begin{aligned}
\underset{X,z,k}{\text{minimize}} \quad & (k,\ -\text{tr}(X)) \\
\text{subject to} \quad & AX + XA^T + \sum_{i=1}^{M} z_i b_i b_i^T = 0 \\
& \text{rank}(X) = n, \quad z_i \in \{0,1\}, \quad k = \sum_{i=1}^{M} z_i.
\end{aligned} \tag{12}$$

Algorithm 1 can be modified to produce alternative heuristics for this problem. Two different greedy algorithms that return a solution to (13) are presented in Algorithms 2 and 3. These algorithms are not guaranteed to return optimal selections, but they can provide small actuator subsets to provide good controllability in terms of the Gramian metric while satisfying the rank constraint.

Similarly, to improve the heuristic outputs of Algorithms 1, 2, or 3, one can employ a stingy algorithm to prune those elements of $\mathcal{S}^\star$ that do not contribute to satisfying the controllability conditions of the system while minimizing the impact of their removal. Algorithm 4 is proposed to achieve such pruning.

Algorithm 2 A Greedy Algorithm for (13).

Require: A, B
- $X \leftarrow 0$, $\overline{\mathcal{S}} \leftarrow \{1,\ldots,M\}$, $\mathcal{S} \leftarrow \emptyset$
- X_i satisfies $AX_i + X_iA^T + b_ib_i^T = 0, \forall i \in \{1,\ldots,M\}$
- **while** $\text{rank}(X) < n$ **do**
  - $\mathcal{I}^\star \leftarrow \left\{ i : i \in \underset{i\in\overline{\mathcal{S}}}{\text{argmax}} \quad [\text{rank}(X+X_i) - \text{rank}(X)] \right\}$
  - $i^\star \in \underset{i\in\mathcal{I}^\star}{\text{argmax}} \quad \text{trace}(X_i)$
  - $X \leftarrow X + X_{i^\star}$
  - $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^\star\}$
  - $\overline{\mathcal{S}} \leftarrow \overline{\mathcal{S}} \setminus \{i^\star\}$
- **end while**
- $X^\star \leftarrow X$, $\mathcal{S}^\star \leftarrow \mathcal{S}$

Algorithm 3 A Greedy Algorithm for (13).

Require: A, B
- $X \leftarrow 0$, $\overline{\mathcal{S}} \leftarrow \{1,\ldots,M\}$, $\mathcal{S} \leftarrow \emptyset$, $z_i \leftarrow 0$, $\forall i \in \{1,\ldots,M\}$
- X_i satisfies $AX_i + X_iA^T + b_ib_i^T = 0, \forall i \in \{1,\ldots,M\}$
- **while** $\text{rank}(X) < n$ **do**
  - $i^\star = \underset{i\in\overline{\mathcal{S}}}{\text{argmax}} \quad [\text{trace}(X+X_i) - \text{trace}(X)]$
  - **if** $\text{rank}(X) < \text{rank}(X + X_{i^\star})$ **then**
    - $z_i \leftarrow 1$
    - $X \leftarrow X + X_{i^\star}$
    - $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^\star\}$
  - **end if**
  - $\overline{\mathcal{S}} \leftarrow \overline{\mathcal{S}} \setminus \{i^\star\}$
- **end while**
- $\mathcal{S}^\star \leftarrow \mathcal{S}$

Algorithm 4 A Stingy Algorithm for Pruning $\mathcal{S}^\star$

Require: A, B,
- # $\widehat{\mathcal{S}}$ and $\widehat{X}$ are the outputs of Algorithms 1, 2, or 3.
- $\mathcal{S} \leftarrow \widehat{\mathcal{S}}$, $X \leftarrow \widehat{X}$
- X_i satisfies $AX_i + X_iA^T + b_ib_i^T = 0, \forall i \in \mathcal{S}^\star$
- $\overline{\mathcal{S}} = \{i \in \mathcal{S} : \text{rank}(X - X_i) = n\}$
- **while** $\overline{\mathcal{S}} \neq \emptyset$ **do**
  - $i^\star \in \underset{i\in\mathcal{I}^\star}{\text{argmin}} \quad \text{trace}(X_i)$
  - $X \leftarrow X - X_{i^\star}$
  - $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i^\star\}$
  - $\overline{\mathcal{S}} = \{i \in \mathcal{S} : \text{rank}(X - X_i) = n\}$
- **end while**
- $X^\star \leftarrow X$, $\mathcal{S}^\star \leftarrow \mathcal{S}$

IV. Empirical comparison of greedy and convex relaxation heuristics

In this section we present several illustrative numerical examples to compare the greedy and convex relaxation heuristics in both random and regular networks. In all examples, we generate a network whose structure defines non-zero entries in the dynamic matrix and then randomly generate weights associated with non-zero entries by drawing independently from a standard normal distribution. We then shift the matrix so that it is stable, with the slowest eigenvalue(s) having real part -0.05. We assume that an input signal could be injected into any state node, i.e., the set V of possible input matrix vectors corresponds to the standard basis for the state space. For the convex relaxation, we use the second heuristic

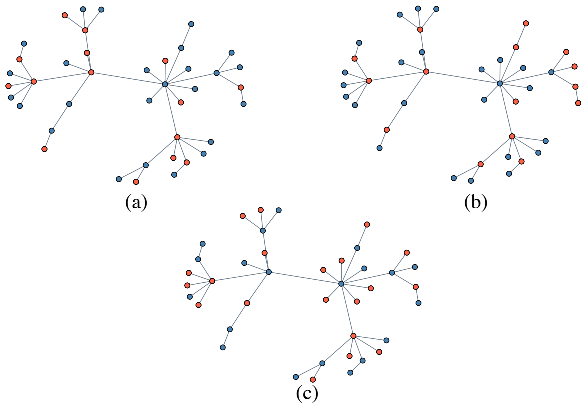

Fig. 1. Selecting actuator nodes in a 40-node Barabási-Albert random network to minimize the trace of the controllability Gramian (pseudo) inverse: (a) convex relxation, (b) greedy algorithm without rank constraints, (c) greedy algorithm with rank constraints. Actuator nodes are highlighted in red. Each algorithm produces a different set of 15 actuators, while the greedy algorithm with rank constraints requires 20 actuators to be added to satisfy the constraint.

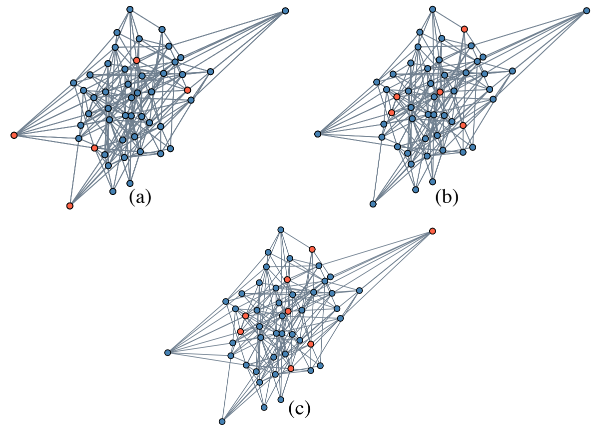

Fig. 2. Selecting actuator nodes in a 50-node Erdős-Rényi random network to minimize the trace of the controllability Gramian (pseudo) inverse: (a) convex relaxation, (b) greedy algorithm without rank constraints, (c) greedy algorithm with rank constraints. Actuator nodes are highlighted in red. Each algorithm produces a different set of 5 actuators, while the greedy algorithm with rank constraints requires 7 actuators to be added to satisfy the constraint.

described in Section II-C to obtain actuator selections.

Barabási-Albert networks: We first consider selecting 15 actuator nodes to minimize the trace of the controllability Gramian (pseudo) inverse in a 40-node Barabási-Albert network, which is generated with a preferential attachment mechanism that produces power law degree distributions [2]. A representative instance of this problem and results for the convex relaxation and greedy algorithms with and without rank constraints is shown in Fig. 1.

Erdős-Rényi networks: We next consider selecting 5 actuator nodes to minimize the trace of the controllability Gramian (pseudo) inverse in a 50-node Erdős-Rényi random networks, with the edge probability chosen to be 0.08, above the critical value of $\ln(50)/50$ to ensure connectivity of the network [10]. A representative instance of this problem and results for the convex relaxation and greedy algorithms with and without rank constraints is shown in Fig. 2.

Regular cycle networks: Finally, we consider selecting 10 actuator nodes to minimize the trace of the controllability Gramian (pseudo) inverse in a 50-node regular cycle network. An instance of this problem is shown in Fig. 3.

Discussion: We observe that each method generally produces a different selection of actuator nodes, though in some cases certain actuator nodes are common to all three selected sets. The greedy algorithm with rank constraints may require additional or even fewer actuators to enforce the constraint but generally produces actuator subsets the provide worse controllability in terms of trace of the inverse Gramian.

We now compare the greedy algorithm without rank constraints and the convex relaxation. Here in each of these instances, the greedy algorithm produced actuator subsets that provide better controllability in terms of trace of the inverse Gramian. However, we generated many instances of the random networks and varied parameters such as network size and number of actuators added and observed that neither

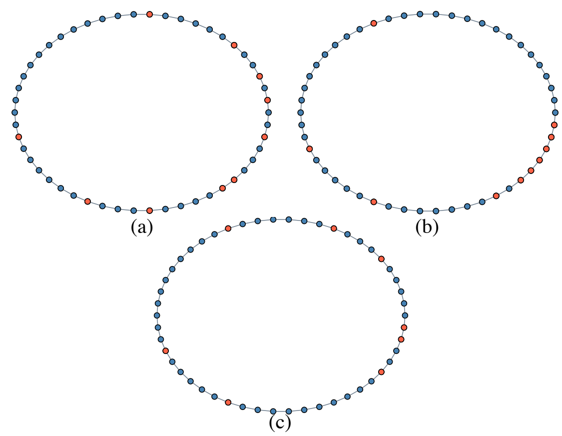

Fig. 3. Selecting actuator nodes in a 50-node regular cycle network to minimize the trace of the controllability Gramian (pseudo) inverse: (a) convex relaxation, (b) greedy algorithm without rank constraints, (c) greedy algorithm with rank constraints. Actuator nodes are highlighted in red. Each algorithm produces a different set of 10 actuators, while the greedy algorithm with rank constraints requires only 8 actuators to be added to satisfy the constraint.

method is clearly superior to the other in these network types. For example, we ran the algorithms on 200 40-node Barabási-Albert networks, and the greedy algorithm produced a better actuator subset on just over half of the instances.

A difficulty in comparing these algorithms is that even for these moderately sized networks, the Gramian often has several very small eigenvalues corresponding to state space directions that require large input energy to achieve state transfer. The trace of the Gramian pseudoinverse and rank computations are then highly sensitive to the threshold defining which eigenvalues are considered numerically zero. The appropriate threshold depends highly on modeling and

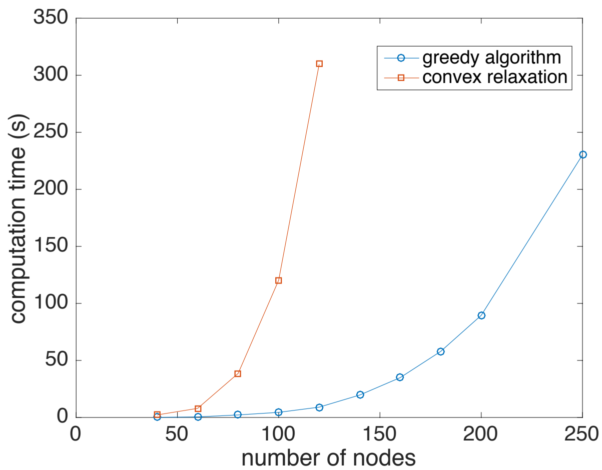

Fig. 4. Computation times for placing $n/10$ actuators in a n-node Barabási-Albert random network to minimize the trace of the controllability Gramian (pseudo) inverse. The greedy algorithm scales to larger networks.

application context to interpret whether the required input energy for state transfer in certain directions is feasible.

Finally, we compared computation times for the convex relaxation and greedy algorithm without rank constraints (rank constraints were not observed to contribute significantly to computation times). We implemented the convex relaxation using the parser CVX [11] with the semidefinite programming solver Mosek [1]. All computations were performed on a 1.7 GHz Intel Core i7 processor. Results are shown in Fig. 4, where it can be seen that the greedy algorithm can scale to larger networks. It is possible to use custom semidefinite programming algorithms [9] and accelerated or parallel greedy algorithms [27] to speed up both methods, so a comprehensive comparison will require further study.

V. Conclusion

We considered an actuator placement problem in a dynamical network using Gramian metrics to quantify controllability. We proposed a convex relaxation that provides lower bounds and can be used in various ways to generate heuristic actuator subsets. We also propose a modified greedy algorithm that allows a rank constraint to be enforced. Finally, we presented numerical experiments that compare the convex relaxation with greedy algorithms.

References

[1] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015.

[2] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] L. Bushnell, A. Clark, and R. Poovendran. A supermodular optimization framework for leader selection under link noise in linear multi-agent systems. *IEEE Transactions on Automatic Control*, 59(2):283–296, 2014.

[4] D.J. Chmielewski, J.-K. Peng, and A.M. Manthanwar. Convex methods in actuator placement. In *American Control Conference*, pages 4309–4314. IEEE, 2002.

[5] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran. Minimizing convergence error in multi-agent systems via leader selection: A supermodular optimization approach. *IEEE Transactions on Automatic Control*, 59(6):1480–1494, 2014.

[6] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran. Joint performance and controllability of networked systems. In *Submodularity in Dynamics and Control of Networked Systems*, pages 175–198. Springer, 2016.

[7] F.L. Cortesi, T.H. Summers, and J. Lygeros. Submodularity of energy related controllability metrics. In *IEEE Conference on Decision and Control, Los Angeles, CA*, pages 2883–2888, 2014.

[8] N.J. Cowan, E.J. Chastain, D.A. Vilhena, J.S. Freudenberg, and C.T. Bergstrom. Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. *PLOS ONE*, 7(6):e38398, 2012.

[9] N.K. Dhingra, M.R. Jovanovic, and Z.-Q. Luo. An ADMM algorithm for optimal sensor and actuator selection. In *IEEE Conference on Decision and Control*, pages 4039–4044. IEEE, 2014.

[10] P Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[11] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, 2014.

[12] A.S. Hodel and K. Poolla. Parallel solution of large lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1189–1203, 1992.

[13] F. Li, M. de Oliveira, and R.E. Skelton. Integrating control design and sensor/actuator selection. In *IEEE Conference on Decision and Control*, pages 6612–6617. IEEE, 2006.

[14] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

[15] U. Munz, M. Pfister, and P. Wolfrum. Sensor and actuator placement for linear systems based on and optimization. *IEEE Transactions on Automatic Control*, 59(11):2984–2989, 2014.

[16] T. Nepusz and T. Vicsek. Controlling edge dynamics in complex networks. *Nature Physics*, 8(7):568–573, 2012.

[17] A. Olshevsky. Minimal controllability problems. *IEEE Transactions on Control of Network Systems*, 1(3):249–258, 2014.

[18] F. Pasqualetti, S. Zampieri, and F. Bullo. Controllability metrics, limitations and algorithms for complex networks. *IEEE Transactions on Control of Network Systems*, 1(1):40–52, 2014.

[19] S. Pequito, S. Kar, and A.P. Aguiar. A framework for structural input/output and control configuration selection in large-scale systems. *IEEE Transactions on Automatic Control*, 61(2):303–318, 2016.

[20] B. Polyak, M. Khlebnikov, and P. Shcherbakov. An lmi approach to structured sparse feedback design in linear control systems. In *European Control Conference*, pages 833–838. IEEE, 2013.

[21] I. Rajapakse, M. Groudine, and M. Mesbahi. Dynamics and control of state-dependent networks for probing genomic organization. *Proceedings of the National Academy of Sciences*, 108(42):17257–17262, 2011.

[22] J. Ruths and D. Ruths. Control profiles of complex networks. *Science*, 343(6177):1373–1376, 2014.

[23] I. Shames and T.H. Summers. Rigid network design via submodular set function optimization. *IEEE Transactions on Network Science and Engineering*, 2(3):84–96, 2015.

[24] T.H. Summers. Actuator placement in networks using optimal control performance metrics. In *submitted to the IEEE Conference on Decision and Control*. IEEE, 2016.

[25] T.H. Summers, F. Cortesi, and J. Lygeros. On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, 3(1):91–101, 2016.

[26] T.H. Summers and J. Lygeros. Optimal sensor and actuator placement in complex dynamical networks. In *IFAC World Congress, Cape Town, South Africa*, pages 3784–3789, 2014.

[27] T.H. Summers, I. Shames, J. Lygeros, and F. Dörfler. Topology design for optimal network coherence. In *European Control Conference*, pages 575–580. IEEE, 2015.

[28] V. Tzoumas, M. A. Rahimian, G. Pappas, and A. Jadbabaie. Minimal actuator placement with bounds on control effort. *IEEE Transactions on Control of Network Systems*, 3(1):67–78, 2016.

[29] A. Vinod, T.H. Summers, and M. Oishi. User-interface design for mimo lti human-automation systems through sensor placement. In *American Control Conference*, pages 5276–5283. IEEE, 2016.

[30] W.-X. Wang, X. Ni, Y.-C. Lai, and C. Grebogi. Optimizing controllability of complex networks by minimum structural perturbations. *Physical Review E*, 85(2):026115, 2012.