Controllability Robustness under Actuator Failures: Complexities and Algorithms

Yuan Zhang

Abstract

The problem of determining the minimal number of inputs (i.e., actuators) whose removal destroy controllability of a linear time invariant (LTI) system is addressed. This problem is inverse to the recently well-studied minimal controllability problems, and significant in measuring controllability robustness of an LTI system under denial of service attacks on actuators. It is first proven that this problem is generally NP-hard. Then, a pseudo-polynomial time algorithm is suggested for solving this problem on systems with bounded maximum geometric multiplicities. Moreover, this algorithm is extended to the case where each actuator has a non-negative cost to be removed.

Index Terms

Controllability robustness, actuator removals, security analysis, complexity

I. INTRODUCTION

Input/output selections under the objectives to meet/optimize certain system performances have long been active but challenging issues in control community [8]. In a recent paper [5], it is first shown that, given an autonomous system as (1), it is NP-hard to determine the minimal number of state variables that need to be actuated by an input to ensure system controllability,

\[ \dot{x}(t) = Ax(t), \]

where \( x(t) \) is the state vector, and \( A \) is state transition matrix. A more general proposition is that, given a collection of possible input matrix columns, it is NP-hard to choose the minimal number of columns to form an input matrix so that the resulting system can be controllable [5], [7]. In this paper, we consider an inverse problem to that. That is, given a state transition matrix and its associated input matrix, determining the minimal number of inputs whose removal destroy controllability of the resulting system. This problem is well-encountered when one needs to evaluate whether controllability of a system is robust against malicious attacks on its actuators, for example, the denial of service attacks on actuators.

Main Contributions: It is proven that the above problem is generally NP-hard for the first time. Nevertheless, a pseudo-polynomial time algorithm is provided for solving that problem on systems with bounded maximum geometric multiplicities. This algorithm uses traversals over a recursive tree built from the left eigenspaces of the system state transition matrix and the input matrix. Moreover, this algorithm is extended to the case where each actuator has a non-negative cost. In addition, the cardinality-constraint submodular function minimization structure of the involved problems are revealed.

Notations: \( \mathbb{R}, \mathbb{C}, \mathbb{Z} \) and \( \mathbb{N} \) denote the set of real, complex, integral and non-negative integral numbers, respectively. For a matrix \( M \), \( M_{ij} \) denotes its \((i, j)\)-th entry if no confusion is made.

II. PROBLEM FORMULATION

Consider the following linear time invariant (LTI) system

\[ \dot{x}(t) = Ax(t) + Bu(t), \]

where \( x(t) \in \mathbb{R}^n \) is the state vector, \( u(t) \in \mathbb{R}^m \) is the input vector, \( A \in \mathbb{R}^{n \times n} \) and \( B \in \mathbb{R}^{n \times m} \) are respectively the state transition matrix and input matrix. Without loss of generality, assume that every column of \( B \) is a non-zero vector. Moreover, let \( V = \{1, \ldots, m\} \), and \( B_S \) denotes the submatrix of \( B \) consisting of columns of \( B \) indexed by \( S \subseteq V \).

System (2) is said to be controllable, if for any two states \( x_0, x_1 \in \mathbb{R}^n \), there exists an input \( u(t) \) that can drive the system states from \( x_0 \) to \( x_1 \) in finite time. We just simply say \((A, B)\) is controllable if System (2) is controllable. For practical

Yuan Zhang was with Tsinghua University, Beijing, 100084, P. R. China (email: zhangyuan14@mails.tsinghua.edu.cn).
cyber-physical systems, we may want to know whether a given system like (2) can preserve its controllability under denial of service attacks on a cardinality-constrained set of actuators. Notice that the denial of service attack on an actuator means that the attacked actuator cannot inject signals into the state variables, which we just call ‘actuator failure’ or ‘actuator removal’. We say System (2) is p-robust against actuator failures, $p \in \mathbb{N}$, if under the failures of an arbitrary set of $p$ actuators, the resulting system is still controllable. To measure such controllability robustness, the following optimization problem is considered in this paper

**Problem 1:** Given $(A, B)$ in (2)

\[
\begin{align*}
\min_{S \subseteq V} & \quad |S| \\
\text{s.t.} & \quad (A, B_{V \setminus S}) \text{ is uncontrollable}
\end{align*}
\]

Denote the cardinality of the optimal solution to Problem 1 by $p^*$. Then, it is easy to see that, System (2) is $p$-robust against actuator failures for any nonzero integers $p \leq p^* - 1$. Furthermore, a variant of Problem 1 is to determine the minimum dimension of controllable subspaces of a given system under cardinality-constrained actuator failures. This problem asks for the worst damage in controllable subspaces that the failures of a set of actuators with cardinality upper bound can cause. Denote the controllability matrix of a pair $(A, B)$ by $C(A, B)$, i.e., $C(A, B) = [B, AB, \ldots, A^{n-1}B]$. Then, this problem can be formulated as

**Problem 2:** Given $(A, B)$ in (2) and $l \in \mathbb{N}$

\[
\begin{align*}
\min_{S \subseteq V, |S| \leq l} & \quad \text{rank } C(A, B_{V \setminus S}) \\
\text{s.t.} & \quad (A, B_{V \setminus S}) \text{ is uncontrollable}
\end{align*}
\]

The following well-known PBH test gives a necessary and sufficient condition for System (2) to be controllable.

**Lemma 1:** System (2) is controllable, if and only if for each eigenvalue $\lambda$ of $A$, there exists no $x \in \mathbb{C}^n$, $x \neq 0$, such that $x^TA = \lambda x^T$ and $x^TB = 0$.

Moreover, assume that $A$ has $p \leq n$ distinct eigenvalues, and denote the $i$th eigenvalue by $\lambda_i$; for $i = 1, \ldots, p$. Let $k_i$ be the dimension of the left null space of $\lambda_iI - A$; that is, $k_i$ is the geometric multiplicity of $\lambda_i$. In addition, let $X_i = [x_{i1}, \ldots, x_{ik_i}]$ be a left eigenbasis of $A$ associated with $\lambda_i$; that is, $X_i$ is stacked by $k_i$ vectors which are linearly independent spanning the left null space of $\lambda_iI - A$. With these definitions, from Lemma 1 it is easy to obtain the following corollary.

**Corollary 1:** System (2) is controllable, if and only if $X_i^TB$ is of full row rank for $i = 1, \ldots, p$.

### III. Complexity Analysis

A natural question is whether the above two problems are solvable in polynomial time. In this section we give a negative answer to that question.

**Theorem 1:** Problem 1 and Problem 2 are both NP-hard.

**Proof:** If Problem 2 is solvable in polynomial time, let $l$, which is the cardinality upper bound of $S$ in Problem 2, increase from 0 to $n$. It is easy to see that, the first $l$ that makes the optimal value of Problem 2 less than $n$, is exactly the optimal value of Problem 1. Hence, to show the NP-hardness of Problem 2 it suffices to show the NP-hardness of Problem 1. To this end, in the following we give a reduction from the Linear Degeneracy Problem 1.

The linear degeneracy problem is to determine whether a given $d \times p$ rational matrix $F = [f_1, \ldots, f_p]$ contains a degenerate submatrix of order $d$, i.e., $\det[f_{i1}, \ldots, f_{id}] = 0$, for some $i_1, \ldots, i_d \in \{1, \ldots, p\}$. In [1], it is proven that this problem is NP-complete, and there are infinitely many integral matrices associated with which the linear degeneracy problem is NP-complete. Now, let $X$ be an arbitrary $k \times n$ integral matrix ($k < n$) with full row rank, and $\alpha_1 = \max\{|X_{ij}|\}$. Notice that the full row rank constraint does not alter the NP-completeness of the linear degeneracy problem associated $X$. Let $X^\perp$ be the basis matrix of the null space of $X$, i.e., $X^\perp$ is an $n \times (n-k)$ matrix spanning the null space of $X$. $X^\perp$ can be constructed via the Gaussian elimination method in polynomial time $O(n^3)$. Moreover, let $\alpha_2 = \max\{|X^\perp_{ij}|\}$, and $\alpha_{\max} = \max\{\alpha_1, \alpha_2\}$. Notice that the entries of $X^\perp$ are rational, and $X^\perp$ multiplied by any nonzero scalars is still a basis matrix of the null space of $X$. Hence, the encoding length of $\alpha_2$ (i.e., $\log_2 \alpha_2$) can be polynomially bounded by $k$ and $n$; so is $\alpha_{\max}$. Next, define $H(\eta)$ as

$$H(\eta) = \begin{bmatrix} X & (X^\perp)^T \end{bmatrix}^T + \eta I_{(n-k) \times n},$$
where $1_{(n-k)\times n}$ denotes the $(n-k) \times n$ matrix whose entries are all one. Then, clearly $\det H(0) \neq 0$. Select one rational number $\eta^*$ that satisfies $\eta^* > \alpha_2$ and $\det H(\eta^*) \neq 0$. We will show that such $\eta^*$ can be found in polynomial time. First, $\det H(\eta)$ is a polynomial of $\eta$ with degree at most one by noting that the coefficient matrix of $\eta$ in $H(\eta)$ is rank one, and that $\det H(0) \neq 0$. Hence, in arbitrary set consisting of 2 distinct rational numbers which are bigger than $\alpha_2$, there must exist one $\eta^*$, such that $\det H(0)(1-\eta^*) \neq \det H(1)\ast \eta^*$, leading to $\det H(\eta^*) \neq 0$. Second, it holds that $\det H(0) \leq \alpha_{\max} n^n$ by noting that $\det H(0)$ consists of the summations of $n! \leq n^n$ signed products of precisely one entry per row and column of $H(0)$. Similarly, $\det H(1) \leq (\alpha_{\max} + 1)^n n^n$. Hence, both $\det H(0)$ and $\det H(1)$ have encoding lengths polynomially bounded by $n$ (i.e., the encoding lengths are $n \log_2 \alpha_{\max} + n \log_2 n$ and $n \log_2 (\alpha_{\max} + 1) + n \log_2 n$ respectively). After determining an $\eta^*$ satisfying the above requirements, let matrices $P = H(\eta^*)$, $\Gamma = \text{diag}\{I_k, 2, 3, \ldots, n-k+1\}$, construct the system $(A, B)$ as

$$A = P^{-1} \Gamma P, B = I_n.$$  

Since the encoding lengths of entries of $P$ are polynomially bounded by $n$, its inversion $P^{-1} = \text{adj}(P)/\det P$ can be computed in polynomial time and has polynomially bounded encoding lengths too, where $\text{adj}(P)$ denotes the adjugate matrix of $P$. Hence, $A$ can be computed in polynomial time.

We claim that the optimal value of Problem 1 associated with $(A, B)$ is no more than $n-k$, if and only if there exists an $k \times k$ submatrix of $X$ which has zero determinant.

Indeed, from the construction of $A$, the $(k+i)$th row of $P$, denoted by $P_{[k+i]}$, is the left eigenvector of $A$ associated with the eigenvalue $i+1$, $i = 1, \ldots, n-k$. Notice that, all entries of $P_{[k+i]}$ are nonzero. Hence, all $n$ columns of $B$ need to be removed, such that the resulting $P_{[k+i]}B_{V\setminus S}$ fail to be of full row rank, where $V = \{1, \ldots, n\}$. From Corollary 1 the optimal value of Problem 1 associated with $(A, B)$ then equals the minimal number of columns whose removal from $X$ makes the resulting $X$ fail to be of full row rank. If such value is no more than $n-k$, there must exist an $k \times k^+$ submatrix of $X$ which is not of full row rank for some $k^+ \geq k$. Then clearly, the linear degeneracy problem associated with $X$ is yes.

Conversely, suppose that there is an $k \times k$ submatrix of $X$ with zero determinant, and denote it by $X_{\bar{S}}$, $\bar{S} \subseteq V$. Then, clearly one just needs to remove the columns indexed by $V\setminus\bar{S}$ from $X$, such that the resulting $X$ fail to be of full row rank. Hence, the optimal value of Problem 1 is no more than $|V\setminus\bar{S}| = n-k$. Combining the fact that the linear degeneracy problem associated with $X$ is NP-complete, this proves the NP-hardness of Problem 1.

We present some corollaries of Theorem 1.

First, notice that in the proof of Theorem 1 $B = I_n$, which means that each input actuates only one state variable. Hence, removing one input corresponds to that exactly one state variable loses its direct input signals. The following corollary is immediate.

**Corollary 2:** Given a system $(A, B)$, it is NP-hard to determine the minimal number of state variables that need to be blocked from their direct inputs, such that the resulting system becomes uncontrollable.

Next, suppose that System 2 is measured by the following equation:

$$y(t) = Cx(t),$$

where $y(t)$ is the output vector, $C$ is the output matrix. By duality between controllability and observability, it is easy to obtain the following corollary on observability robustness under output failures.

**Corollary 3:** For System 2-3, it is NP-hard to determine the minimal number of outputs whose failures make the resulting system unobservable.

Taking the input and output into consideration together, we have the following conclusion.

**Corollary 4:** For System 2-3, it is NP-hard to determine the minimal total number of inputs and outputs whose failures make the resulting system neither controllable nor observable.

**Proof:** Construct $(A, B)$ as suggested in the proof of Theorem 1 i.e., $A = P^{-1} \Gamma P, B = I_n$. Construct $C$ such that $C \in \mathbb{R}^{k \times n}$ and $(A, C)$ is observable. Since the eigenbases $P$ of $A$ are available, the matrix $C$ can be constructed in polynomial time as suggested in 1. Notice that $k$ is the maximum geometric multiplicity of eigenvalues of $A$. According to 3, the minimal number of outputs that ensure observability of the associated system equals the maximum geometric multiplicity of $A$. Hence, removing any one of these $k$ outputs can make the system unobservable. As a result, the minimal total number of inputs and outputs whose failures cause uncontrollability and unobservability equals $p^* + 1$, where $p^*$ is the minimal number of inputs.
that need to be removed to make the associated system uncontrollable. The latter problem is shown to be NP-hard in Theorem 1. The required result follows.

IV. PSEUDO-POLYNOMIAL TIME ALGORITHMS

In this section, we give a pseudo-polynomial time algorithm for Problem 1 for systems with bounded eigenvalue geometric multiplicities. This algorithm is based on the traversals over a recursive tree, constructed from the eigenspaces of the system state transition matrix and input matrix. Then, we extend this algorithm to the case where each input has a non-negative cost, and the purpose is to find the minimal cost of input set whose removal causes uncontrollability. Finally, we point out that Problem 2 has the structure of cardinality-constrained submodular function minimization.

We shall assume that a collection of left eigenbases of \(A\) are computationally available, and denote them by \(X_i^p\). We will use \(r_{\text{min}}\) to denote the optimal value of Problem 1 associated with \((A, B)\). Recall that \(p\) is the number of distinct eigenvalues of \(A\), and \(k_i\) is the geometric multiplicity of the \(i\)th eigenvalue. Moreover, rewrite the input matrix as \(B = [b_1, \ldots, b_m]\). We first consider a simple case where \(A\) has no repeated eigenvalues (i.e., \(A\) is simple), and then the general case.

A. Simple Dynamics Case

Assume that matrix \(A\) has no repeated eigenvalues. Then, \(\{X_i^p\}_{i=1}^n\) becomes a collection of left eigenvectors of \(A\). For the \(i\)th eigenvalue of \(A\), define \(F_i\) as the collection of columns of \(B\) that are not orthogonal to \(X_i\); that is

\[
F_i = \{ j : X_i^T b_j \neq 0, j \in \{1, \ldots, m\} \}.
\]

Since \(X_i\) is a vector, \(X_i^T b_j\) becomes a scalar. Then, from Corollary 1 it’s obvious that

\[
r_{\text{min}} = \min_{i \in \{1, \ldots, n\}} |F_i|.
\]

Obviously, finding \(r_{\text{min}}\) can be done in polynomial time. Indeed the above analysis is nothing but trivial. Can we extend it to the general case where the geometric multiplicities of eigenvalues of \(A\) can be greater than one?

B. General Case

Now we assume that the geometric multiplicities of eigenvalues of \(A\) are bounded by some constant \(k \in \mathbb{N}\). That is, \(k_i \leq k\), \(i = 1, \ldots, p\), as \(n\) and \(m\) increase. For most practical systems, this assumption is reasonable. Indeed, it is found that random square matrices generically have no repeated eigenvalues [2].

Let us focus on an individual eigenvalue \(\lambda_i\), \(i \in \{1, \ldots, p\}\). Let \(r_i\) be the minimum number of columns whose removals from \(Y^{(i)} = [X_i^T b_1, \ldots, X_i^T b_m]\) make the remaining matrix fail to be full of row rank. To determine \(r_i\), a pure combinatorial method needs to compute the ranks of at most \(\binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{m-k_i}\) to \(O(2^m)\) submatrices, which increases exponentially with \(m\) even when \(k_i\) is bounded. Hence, the direct combinatorial method is not computationally efficient.

In what follows, a pseudo-polynomial time algorithm based on traversals over a recursive tree is presented.

The pseudo code of this algorithm is given in Algorithm 1. The intuition behind Algorithm 1 is that, for the \(\kappa\)th eigenvalue of \(A\), instead of directly determining \(r_\kappa\), we try to determine \(m - r_\kappa\), which is the maximum number of columns of \(Y^{(\kappa)}\) that fail to have full row rank. Algorithm 1 first builds a recursive tree and then searches the maximum return value among the leafs of this recursive tree, i.e., \(T_{\text{max}}^{(\kappa)}\), which equals exactly \(m - r_\kappa\). An illustrative example of this recursive tree is given in Fig. 1. To build the recursive tree, in the \(\tau\)th iteration for the \(\kappa\)th eigenvalue, all the matrices \(Y^{(\kappa)}_{T_{\tau}^{(\kappa)}}\) have rank \(\tau\), \(i = 1, \ldots, c_{\text{end}}^{(\tau-1)}\), where \(c_{\text{end}}^{(\tau-1)}\) denotes the last element of the sequence \(\{c_0^{(\tau-1)}, c_1^{(\tau-1)}, \ldots, c_{\text{end}}^{(\tau-1)}\}\). Then, \(Y^{(\kappa)}_{T_{\tau}^{(\kappa)}}\) is obtained by adding the maximal columns of \(Y^{(\kappa)}\) to \(Y^{(\kappa)}_{T_{\tau}^{(\kappa)}}\) while its rank is preserved. Next, each set \(T_{\tau}^{(\kappa)}\) generates \(|Ω_\tau^{(\kappa)}|\) child nodes in the recursive tree, namely \(\{T_{\tau+1}^{(\kappa)} \mid c_i^{(\tau)} \in \{c_0^{(\tau)}, \ldots, c_{\text{end}}^{(\tau)}\}\}\), such that each matrix \(Y^{(\kappa)}_{T_{\tau+1}^{(\kappa)}}\) has rank \(\tau + 1\), \(i = c_{\text{end}}^{(\tau)} + 1, \ldots, c_i^{(\tau)} + |Ω_\tau^{(\kappa)}|\).

\textbf{Theorem 2:} Given a system \((A, B)\) with geometric multiplicities of eigenvalues of \(A\) bounded by \(\tilde{k}\), \(A \in \mathbb{R}^{n \times n}\), \(B \in \mathbb{R}^{n \times m}\), Algorithm 1 can determine the optimal solution to Problem 1 in time complexity at most \(O(\tilde{k}^2 m^{k+1} n + mn^3)\).

\textbf{Proof:} As argued above, the main procedure of Algorithm 1 is to determine the maximum number of columns of \(Y^{(\kappa)}\) that fail to have full row rank for each eigenvalue \(\lambda_\kappa\), \(\kappa = 1, \ldots, p\). To justify the recursive procedure, observe that for any subset \(S \subseteq \mathcal{V}\),
Algorithm 1: A pseudo-polynomial time algorithm for Problem [I]

Input: System parameters \((A, B)\)

Output: The optimal solution to Problem [I]

1: Calculate the left eigenbases \(\{X_i\}_{i=1}^4\) of \(A\).
2: for \(k = 1\) to \(p\) do
3: Initialize \(\tau = 0, T_1^{(0)} = \phi, c_0^{(-1)} = 1\).
4: while \(\tau < k\) do
5: for \(i = 1\) to \(c_{\text{end}}^{(\tau-1)}\) do
6: \(T_i^{(\tau)} = T_i^{(\tau)} \cup \{j : \text{span}(Y_j^{(\kappa)}) \subseteq \text{span}(Y_{T_i^{(\tau)}}^{(\kappa)}), j \in V \setminus T_i^{(\tau)}\}\).
7: \(\Omega_i^{(\tau)} = \{j : \text{span}(Y_j^{(\kappa)}) \in \text{span}(T_i^{(\tau)}) \cup \{j : \text{span}(Y_j^{(\kappa)}) \in \text{span}(V \setminus T_i^{(\tau)})\}, \text{ Let } c_i^{(\tau)} = i \in |\Omega_i^{(\tau)}|, c_0^{(\tau)} = 0\).
8: Rewrite \(\Omega_i^{(\tau)} = \{\tau_1, ..., \tau_{q(l)}^{(\tau)}\}\) and let \(T_i^{(\tau+1)} = T_i^{(\tau)} \cup \{j_q\}\) for \(q = 1, ..., |\Omega_i^{(\tau)}|\).
9: end for
10: \(\tau + 1 \leftarrow \tau\).
11: end while
12: \(T_{\text{max}}^{(\kappa)} = \arg \max_{T_{\kappa-1}^{(\kappa-1)}} \left\{ |T_1^{(k_{\kappa-1})}|, \cdots, |T_{c_{\text{end}}^{(\kappa-2)}}^{(k_{\kappa-1})}| \right\}\), \(F_\kappa = V \setminus T_{\text{max}}^{(\kappa)}\).
13: end for
14: Return \(F_{\text{min}} = \arg \min \{|F_1|, ..., |F_p|\}\) and \(r_{\text{min}} = |F_{\text{min}}|\).

Fig. 1. An illustrative example of a recursive tree in Algorithm [I]. Note that this recursive tree, depending on the considered system parameters, is not necessarily a binary tree.

if \(S' \subseteq V \setminus S\) and \(\text{span}(Y_S^{(\kappa)}) \subseteq \text{span}(Y_{S'}^{(\kappa)})\), then \(\text{rank}Y_{S'}^{(\kappa)} = \text{rank}Y_S^{(\kappa)}\); if \(\text{span}(Y_j^{(\kappa)}) \notin \text{span}(Y_{S'}^{(\kappa)})\), then \(\text{rank}Y_j^{(\kappa)} = \text{rank}Y_{S'}^{(\kappa)} + 1\). Hence, for the \(\kappa\)th eigenvalue, the recursive tree has depth exactly \(k_\kappa\). Moreover, \(\{T_1^{(k_{\kappa-1})}, ..., T_{c_{\text{end}}^{(k_{\kappa-2})}}^{(k_{\kappa-1})}\}\) contain all the submatrices formed by columns of \(Y^{(\kappa)}\) with the property that: it has rank \(k_\kappa - 1\), and adding any rest columns from \(Y^{(\kappa)}\) can make it have full row rank. The optimality of the solution returned from Algorithm [I] then follows immediately.

In the recursive tree of the \(\kappa\)th eigenvalue, each parent node has at most \(m\) child nodes. Thus, the \(\tau\)th layer has at most \(m^\tau\) nodes, \(\tau = 0, ..., k_\kappa - 1\). Hence, the total nodes in that recursive tree is at most \(\sum_{\tau=0}^{k_\kappa-1} m^\tau \rightarrow m^{k_\kappa}\). For each node, the rank update procedure incurs \(O(k_\kappa^2 m\tau)\) using the singular value decomposition (Line 6 to Line 7 of Algorithm [I]). To obtain \(Y^{(\kappa)}_{p\kappa+1}\), it incurs computational complexity \(O(m^3)\). Note that \(k_\kappa \leq \bar{k}\) for \(k = 1, ..., \bar{p}\), and \(p \leq n\). To sum up, the total computational complexity of Algorithm [I] is at most \(O(\bar{k}^2 m^{\bar{k}+1} n + m^3)\).

Remark 1: In computational complexity theory, an algorithm runs in pseudo-polynomial time if its running time is a polynomial in the numeric value of the input (the largest integer present in the input). A NP-hard problem is said to be weakly NP-hard, if there is a pseudo-polynomial time algorithm for it; otherwise it is strongly NP-hard. The above results indicate that, provided that the left eigenbases of \(A\) are computationally available, Problem [I] is weakly NP-hard in a broader sense.
(with respect to the maximum geometric multiplicities of $A$). In other words, the computational intractability of Problem 1 is essentially caused by the geometric multiplicities of eigenvalues of $A$, rather than the dimensions of states and inputs of the systems. By contrast, from [5], it is known that the minimal controllability problems discussed therein are strongly NP-hard. This distinction is somehow surprising to the authors.

C. Minimal Cost Actuator Failures

In actual systems, different actuators may incur different costs to be removed. Here cost can be used to measure the budget of an actuator to be removed, or the difficulty/fragility of an actuator to be attacked. In such case, an attacker/protector may be more interested in the minimal cost of a set of actuators whose removal causes uncontrollability. Let $c(i) \geq 0$ denote the cost of the $i$th actuator, $i \in V$. Given $S \subseteq V$, let $c(S) = \sum_{i \in S} c(i)$. Then, Problem 1 can be generalized to the following Problem 3

The following theorem reveals that Problem 3 is pseudo-polynomial time solvable with bounded maximum geometric multiplicities of $A$.

**Theorem 3**: Given a system $(A, B)$ in (2) with geometric multiplicities of eigenvalues of $A$ bounded by $\bar{k}$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, Algorithm 2 can determine the optimal solution to Problem 3 with complexity at most $O(\bar{k}^2 m^{k+1} n + mn^3)$.

**Algorithm 2**: A pseudo-polynomial time algorithm for Problem 3

**Input**: System parameters $(A, B)$

**Output**: The optimal solution to Problem 3

1. Calculate the left eigenbases $\{X_i^{\kappa}\}_{\kappa=1}^p$ of $A$.

2. for $\kappa = 1$ to $p$ do

3. Run the same procedure from Line 3 to Line 11 in Algorithm 1

4. $T_{\text{max}}^{[\kappa]} = \arg \max \{T(k-1)_{\text{max}}^{[\kappa-1]} \}$

5. end for

6. Return $F_{\text{min}} = \arg \min \{c(F_1), \ldots, c(F_p)\}$ and $r_{\text{min}} = c(F_{\text{min}})$.

**Proof**: Note that the distinction between Algorithm 2 and Algorithm 1 lies in Line 4 of Algorithm 2. To be specific, in Algorithm 2 the cardinality of a set of actuators is replaced by its total cost. With this observation, the proof of Theorem 3 follows similar arguments to that of Theorem 2. The details are omitted.

D. Structure of Problem 2

Given $(A, B)$ in (2), define a function $f(S) : 2^V \rightarrow \mathbb{N}$ as

$$f(S) = \text{rank}C(A, BS).$$

It is known that $f(S)$ is submodular on $S \subseteq V$. From the property of submodularity, $f(V \setminus S)$ is also submodular on $S \subseteq V$. This means that Problem 2 can be formalized as a cardinality-constrained submodular function minimization problem. Although a submodular function minimization problem is strongly polynomial time solvable [6], any cardinality-constraint on it is NP-hard [4]. Currently, there is no known algorithm that can return a constant factor approximation for a general cardinality-constrained submodular function minimization problem [4]. Nevertheless, exploiting some special inherent structure of Problem 2 and finding efficient approximation algorithms remain our further work.

---

1 It is proven in [5] that the minimal controllability problems considered therein cannot be approximated within a constant multiplicative factor even when the involved state transition matrix has no repeated eigenvalues.
V. SOME DISCUSSIONS

The pseudo-polynomial time Algorithm [1] is based on the eigenspace decomposition of the system state transition matrix $A$. However, even when the exact value of $A$ is known, its (left) eigenbases might cannot be precisely obtained in its numeric value within a polynomial bounded number of operations. A natural question arises that, can we find $r_{\text{min}}$ by simply querying $\text{rank}C(A, B_{V \setminus S})$? In other words, can we solve Problem [1] by regarding the system $(A, B)$ as a block-box system and querying $\text{rank}C(A, B_{V \setminus S})$ within polynomial bounded times for $S \subseteq V$?

The results of [9] show that many random networks can be controllable by only one actuator. A more interesting extension of Problem [2] is to use some frequently-used quantitative controllability metrics [7] instead of the dimension of controllable subspaces, such as trace of the inverse of controllability Gramian, the minimal eigenvalue of controllability Gramian, etc.

VI. CONCLUSIONS

An inverse problem of the minimal controllability problems is addressed in this paper, that is, determining the minimal number of inputs whose removal destroy controllability of an LTI system. It is shown that this problem is weakly NP-hard with bounded maximum geometric multiplicities of the given state transition matrices for the first time. Some variants, as well as some further directions, are also discussed.

Although most techniques/algorithms in this paper are designed for controllability, they may be applied to some other fields, like the minimal cost critical columns in the linear degeneracy problem [1].

REFERENCES

[1] L. Khachiyan, On the complexity of approximating extremal determinants in matrices, Journal of Complexity, vol. 11, no. 1, pp. 138-153, 1995.
[2] T. Tao and V. Vu, Random matrices have simple spectrum, preprint, arXiv:1412.1438, 2014.
[3] T. Zhou, Minimal inputs/outputs for a networked system, IEEE Control Systems Letters, vol. 1, no. 2, pp. 298-303, 2017.
[4] Z. Svitkina and L. Fleischer, Submodular approximation: Sampling-based algorithms and lower bounds, SIAM Journal on Computing, vol. 40, no. 6, pp. 1715-1737, 2011.
[5] A. Olshevsky, Minimal controllability problems, IEEE Transactions on Control of Network Systems, vol. 1, no. 3, pp. 249-258, 2014.
[6] S. Iwata, L. Fleischer, and S. Fujishige, A combinatorial strongly polynomial algorithm for minimizing submodular functions, Journal of the ACM (JACM), vol. 48, no. 4, pp. 761-777, 2001.
[7] T. H. Summers, F. L. Cortesi and J. Lygeros, On submodularity and controllability in complex dynamical networks, IEEE Transactions on Control of Network Systems, vol. 3, no. 1, pp. 91-101, 2016.
[8] V. D. Wal and B. D. Jager, A review of methods for input/output selection, Automatica, vol. 37, pp. 487-410, 2001.
[9] S. O’Rourke and B. Touri, On a conjecture of Godsil concerning controllable random graphs, SIAM Journal on Control and Optimization, vol. 54, no. 6, pp. 3347-3378, 2016.