2015 IEEE 54th Annual Conference on Decision and Control (CDC)
December 15-18, 2015. Osaka, Japan

Minimal Reachability Problems

V. Tzoumas, A. Jadbabaie, G. J. Pappas*

***Abstract*— In this paper, we address a collection of state space reachability problems, for linear time-invariant systems, using a minimal number of actuators. In particular, we design a zero-one diagonal input matrix B, with a minimal number of non-zero entries, so that a specified state vector is reachable from a given initial state. Moreover, we design a B so that a system can be steered either into a given subset, or sufficiently close to a desired state. This work extends the results of [1] and [2], where a zero-one diagonal or column matrix B is constructed so that the involved system is controllable. Specifically, we prove that the first two of our aforementioned problems are NP-hard; these results hold for a zero-one column matrix B as well. Then, we provide efficient algorithms for their general solution, along with their worst case approximation guarantees. Finally, we illustrate their performance over large random networks.**

I. Introduction

Power grids, transportation systems, brain neural circuits and social networks are just a few of the complex dynamical systems that have drawn the attention of control scientists, [3], [4], [5], [6], since their vast size, and interconnectivity, necessitate novel control techniques with regard to:

i. tasks that are collective [7], e.g., reaching consensus in a system of autonomous interacting vehicles [8];
ii. new cost constraints, e.g., with respect to the number of used actuators and the level of the input and communication power [9].

In this paper, we consider a set of minimal state reachability problems, for linear time-invariant systems, where the term 'minimal' captures our objective to use the least number of actuators towards the involved control tasks. Specifically, we design a zero-one diagonal input matrix B, with a minimal number of non-zero entries, so that one of the following (collective) tasks are met: i) the resultant system can be steered into a subset, or ii) to a state, or iii) sufficiently close to a state. Therefore, our work relaxes the objective of [1] and [2], where a zero-one diagonal or column matrix B is constructed, with a minimal number of non-zero entries, so that the designed system is controllable.

This is an important distinction whenever we are interested only in the feasibility of a state transfer, as in power grids [3]; transportation systems [4]; complex neural circuits [5]; infection processes over large-scale social networks [10] (e.g., from the infectious state to the state where all the network nodes are healthy): Consider for example the system

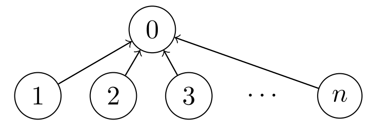

Fig. 1. A $n+1$-node star network: each node represents a state of a linear time-invariant system of the form $dx/dt = Ax + Bu$ (where $x = (x_0, x_1, \dots, x_n)$ is the state vector; A is the system's matrix; B is the input matrix; and u is the input vector). The state of node '0' depends on the states of all the nodes in the network.

in Fig. 1 and assume the transfer from the initial state zero to $(1, 0, 0, \dots, 0)$, where the first entry corresponds to the final state of node '0', the second to that of '1', and so forth; if we impose controllability in the design of B, we get a B with n non-zero elements: $B = \text{diag}(0, 1, 1, \dots, 1)$; that is, states x_1 through x_n must be actuated so that this system is controllable. On the other hand, if we impose only state reachability, we get a B with only one non-zero element, independently of n; e.g., a solution is $B = \text{diag}(1, 0, 0, \dots, 0)$, where only state x_0 is actuated. Thereby, whenever we are interested in the feasibility of a state transfer and in a B with a small number of non-zero elements, the objective of state reachability should not be substituted with that of controllability: under controllability the number of used actuators could grow linearly with n, while under state reachability it could be one for all n. Similar comments carry through with respect to the rest of our objectives.

At the same time, the task to design a sparsest zero-one diagonal matrix B is combinatorial, and, as a result, it may be computationally hard in the worst case. Indeed, we prove that the first two of our aforementioned problems are NP-hard — our proofs hold for a zero-one column matrix B as well. Therefore, we then provide efficient algorithms for their general solution, along with their worst case approximation guarantees; to this end, we use an approximation algorithm that we provide for our third problem, where a sparse zero-one diagonal matrix B is designed so that a system can be steered ϵ-close to a desired state.

These hardness results proceed by reduction to the minimum hitting set problem (MHS), which is NP-hard [11]. In particular, we prove that the problem of state reachability, using a minimal number of actuators, is NP-hard, by reducing it to the controllability problem introduced in [1], which is at least as hard as the MHS. Moreover, we prove that the problem of steering a system into a subset is NP-hard by directly reducing it to the MHS.

Then, we first provide an efficient approximation algorithm so that a system can be steered ϵ-close to a desired state. This algorithm returns a B with a number of non-zero

*All authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104-6228 USA (email: {vtzoumas, pappasg, jadbabai}@seas.upenn.edu).

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, in part by AFOSR Complex Networks Program and in part by ARO MURI W911NF-12-1-0509.

elements up to a multiplicative factor of $O(\ln(\epsilon^{-1}))$ from any optimal solution. Therefore, it allows the designer to select the level of approximation ϵ, with respect to the trade-off between the reachability error ϵ and the number of used actuators (recall that the number of non-zero elements of B coincides with the number of used actuators). Afterwards, we use this algorithm to provide efficient approximation algorithms for the rest of our reachability problems as well.

In addition to [1] and [2], other relevant studies to this paper are [12], [13], [14] and [15], where their authors consider the design of a sparse input matrix B so that an input energy objective is minimized. Moreover, [16] and [17] address the sparse design of the closed loop linear system, with respect to its feedback gain, as well as, a set of sensor placement problems. Other recent works that study sensor placement problems are the [18] and [19].

Furthermore, [20] considers the decidability of a set of problems related to ours; for example, it asks whether the problem of deciding if there exists a control that can drive a given system from an initial state to a desired one is decidable or not. The main difference between this set of problems and ours is that they consider the feasibility of state transfer given a fixed system, whereas we design a system so that the feasibility of a state transfer is guaranteed.

The remainder of this paper is organized as follows. The formulation and model for our reachability problems are set forth in Section II, where the corresponding integer optimization programs are stated. In Section III-A, we prove the intractability of these problems and, then, in Section III-B, we provide efficient algorithms for their general solution, along with their worst case approximation guarantees. Finally, in Section IV, we illustrate our analytical findings, using an instance of the network in Fig. 1, and afterwards, we test the efficiency of the proposed algorithms over large random networks that are commonly used to model real-world networked systems. Section V concludes the paper.

Due to space limitations, all proofs are omitted; they can be found in the full version of this paper, located at the authors websites.

II. PROBLEM FORMULATION

Notation: We denote the set of natural numbers $\{1, 2, \ldots\}$ as $\mathbb{N}$, the set of real numbers as $\mathbb{R}$, and we let $[n] \equiv \{1, 2, \ldots, n\}$ for all $n \in \mathbb{N}$. Also, given a set $\mathcal{X}$, we denote as $|\mathcal{X}|$ its cardinality. Matrices are represented by capital letters and vectors by lower-case letters. For a matrix A, A^T is its transpose and A_{ij} is its element located at the $i-$th row and $j-$th column. Moreover, we denote as I the identity matrix; its dimension is inferred from the context. Additionally, for $\delta \in \mathbb{R}^n$, we let $\text{diag}(\delta)$ denote an $n \times n$ diagonal matrix such that $\text{diag}(\delta)_{ii} = \delta_i$ for all $i \in [n]$. The rest of our notation is introduced when needed.

A. Model

Consider a linear system of n states, $x_1, x_2, \ldots, x_n$, whose evolution is described by

$$\dot{x}(t) = Ax(t) + Bu(t), t > t_0, \tag{1}$$

where $t_0 \in \mathbb{R}$ is fixed, $x \equiv \{x_1, x_2, \ldots, x_n\}$, $\dot{x}(t) \equiv dx/dt$, and $u \in \mathbb{R}^n$ is the input vector. The matrices A and B are of appropriate dimension. Without loss of generality, $u \in \mathbb{R}^n$; in general, whenever the i-th column of B is zero, u_i is ignored. Moreover, we denote (1) as the duple (A, B) and refer to the states $x_1, x_2, \ldots, x_n$ as nodes $1, 2, \ldots, n$, respectively; finally, we denote their collection as $\mathcal{V} \equiv [n]$.

In what follows, A is fixed and the following structure is assumed on B:

Assumption 1: B is a diagonal zero-one matrix: $B = diag(\delta)$, where $\delta \in \{0, 1\}^n$.

Therefore, if $\delta_i = 1$, state x_i is actuated, and if $\delta_i = 0$, is not and u_i is ignored. That is, the number of non-zero elements of B coincides with the number of actuators (inputs) that are implemented for the control of system (1).

In this paper, we design B so that (A, B) satisfies a control objective among the following presented in the next section.

B. Minimal Reachability Problems

We introduce two control objectives, the *state* and *subset reachability*, which we use to define the design problems of this paper. In particular, consider t_0, $t_1 \geq t_0$, and $x(t_0)$ fixed:

Objective 1 (State Reachability): The state $\chi \in \mathbb{R}^n$ is reachable by (A, B) at time $t = t_1$ if and only if there exists input defined over (t_0, t_1) such that $x(t_1) = \chi$.

A parallel notion to the state reachability is the *state feasibility*:

Definition 1 (State Feasibility): The transfer from $x(t_0)$ to $x(t_1) = \chi \in \mathbb{R}^n$ by (A, B), denoted as $x(t_0) \rightarrow x(t_1) = \chi$, is feasible if and only if χ is reachable by (A, B) at time $t = t_1$.

We now present our second objective:

Objective 2 (Subset Reachability): The subset $\mathcal{N} \subseteq \mathbb{R}^n$ is reachable by (A, B) at time $t = t_1$ if and only if there exist $\chi \in \mathcal{N}$ and input defined over (t_0, t_1) such that $x(t_1) = \chi$ is reachable.

The corresponding definition of *subset feasibility* parallels that of state feasibility and it is omitted.

Evidently, Objective 2 generalizes Objective 1: According to it, (A, B) targets from $x(t_0)$ a subset, instead of a single state. Nevertheless, subset reachability of $\mathcal{N}$ does not imply that all states $\chi \in \mathcal{N}$ are reachable. Similarly, although $\chi \in \mathcal{N}$ may not be reachable by (A, B), $\mathcal{N}$ can be; thus, Objective 1 is not a special case of Objective 2. Overall, Objectives 1 and 2 define the two separate design problems that follow.

Problem 1 (Minimal State Reachability): Given $x(t_0)$ and $x(t_1)$, design a B with the smallest number of non-zero elements so that the state transfer $x(t_0) \rightarrow x(t_1)$ is feasible. Note that Problem 1 is always feasible, since for any A, (A, I) is controllable.

Therefore, the objective of Problem 1 relaxes that of [1], [2] where B is designed with the smallest number of non-zero elements so that the resultant (A, B) is controllable.

Problem 2 (Minimal Subset Reachability): Given $x(t_0)$, $\mathcal{N}$ and t_1, design a B with the smallest number of non-zero elements so that the subset $\mathcal{N}$ is reachable from $x(t_0)$ at time t_1.

We refer to Problem 2 as *minimal subset reachability* as well. As with Problem 1, Problem 2 is always feasible, since for any A, (A, I) is controllable.

Evidently, the 'minimal' term in the definition of Problems 1 and 2 captures our objective to design a sparsest[1] B.

Finally, all of our results carry through if we consider the output $y(t) = Wx(t)$ of (1), where W is fixed and of appropriate dimension, instead of $x(t)$. In particular, denote as $\mathcal{R}(W)$ the column space of W and consider the following objectives:

Objective 3 (Output Reachability): The output state $y \in \mathcal{R}(W)$ is reachable by (A, B) at time $t = t_1$ if and only if there exists input defined over (t_0, t_1) such that $y(t_1) = y$.

Naturally, Objectives 1 and 3 coincide for $W = I$. Thereby, a generalized version of Problem 1, where a sparsest B is designed so that an output transfer is feasible, is due. Similar comments apply with respect to the objective below.

Objective 4 (Output Subset Reachability): The $\mathcal{N} \subseteq \mathcal{R}(W)$ is reachable by (A, B) at time $t = t_1$ if and only if there exist $y \in \mathcal{N}$ and input defined over (t_0, t_1) such that $y(t_1) = y$ is reachable.

In what follows, we continue with the original Problems 1 and 2.

III. Main Results

In the first part of this section, III-A, we prove that Problems 1 and 2 are NP-hard. The proofs proceed by reduction to the minimum hitting set problem (MHS), which is NP-hard [11], and is defined as follows:

Definition 2 (Minimum Hitting Set Problem): Given a finite set $\mathcal{M}$ and a collection $\mathcal{L}$ of non-empty subsets of $\mathcal{M}$, find a smallest cardinality $\mathcal{M}' \subseteq \mathcal{M}$ that has a non-empty intersection with each set in $\mathcal{L}$.

In particular, we prove that Problem 1 is NP-hard providing an instance that reduces to the controllability problem introduced in [1], which is at least as hard as the MHS; as a result, we conclude that Problem 1 is as well. Moreover, we prove that Problem 2 is NP-hard by directly reducing it to the MHS.

In the second part of this section, III-B, since Problems 1 and 2 are NP-hard, we provide efficient approximation algorithms for their general solution. Towards this direction, we first generalize Definition 1 as follows:

Definition 3 (ϵ-close feasibility): The transfer $x(t_0) \to x(t_1) = \chi \in \mathbb{R}^n$ by (A, B) is ϵ-feasible if and only if there exists $\chi' \in \mathbb{R}^n$ reachable by (A, B) at time $t = t_1$ such that $\|\chi - \chi'\|^2 \leq \epsilon$, where $\|\cdot\|$ denotes the euclidean norm.

For $\epsilon = 0$, Definitions 1 and 3 coincide.

We use Definition 3 to relax the objective Problem 1, by replacing the feasibility of $x(t_0) \to x(t_1)$ with that of ϵ-close feasibility — from a real-world application perspective, and for small ϵ, this is a weak modification: the convergence of a system exactly to a desired $x(t_1)$ is usually infeasible, e.g., due to external disturbances. We then provide for this problem a polynomial time approximation algorithm, Algorithm 1, that returns a B with sparsity[2] up to a multiplicative factor of $O(\ln(\epsilon^{-1}))$ from any optimal solution of the original Problem 1.

Next, to address Problem 1 with respect to Objective 1, we prove that for all $\epsilon \leq \epsilon(A)$, where $\epsilon(A)$ is positive and sufficiently small, Definitions 1 and 3 still coincide; hence, we implement a bisection-type execution of Algorithm 1, Algorithm 2, that quickly converges to an $\epsilon \leq \epsilon(A)$ and, as a result, returns a B that makes the exact transfer $x(t_0) \to x(t_1)$ feasible.

Finally, we provide an approximation algorithm for Problem 2 when $\mathcal{N} \subseteq \mathbb{R}^n$ is finite, by observing that in this case $\mathcal{N}$ can be approximated as a finite union of euclidean balls in $\mathbb{R}^n$. Specifically, let $\chi_1, \chi_2, \ldots, \chi_{k(\mathcal{N})}$ be their centres and $\epsilon_1, \epsilon_2, \ldots, \epsilon_{k(\mathcal{N})}$ their corresponding radii. Moreover, without loss of generality, assume $x(t_0) = 0$. Then, by executing Algorithm 1 for $(x(t_1) = \chi_i, \epsilon = \epsilon_i)_{i \in [k(\mathcal{N})]}$ and selecting the sparsest solution B among all $i \in [k(\mathcal{N})]$, we return an approximate solution to Problem 2 with Algorithm's 1 worst case guarantees.

A. Intractability of the Minimal Reachability Problems

We prove that Problems 1 and 2 are NP-hard. The proofs proceed with respect to the decision version of Problems 1 and 2 and that of MHS. The latter is defined as follows:

Definition 4 (k-hitting set): Given a finite set $\mathcal{M}$ and a collection $\mathcal{L}$ of non-empty subsets of $\mathcal{M}$, find an $\mathcal{M}' \subseteq \mathcal{M}$ of cardinality at most k that has a non-empty intersection with each set in $\mathcal{L}$.

Without loss of generality, we assume that every element of $\mathcal{M}$ appears in at least one set in $\mathcal{L}$ and all set in $\mathcal{L}$ are non-empty.

The decision versions of Problems 1 and 2 are defined in Sections III-A.1 and III-A.2, where we present their NP-hardness, respectively.

1) Intractability of Problem 1: We prove that the decision version of Problem 1 reduces to the k-hitting set and, as a result, that Problem 1 is NP-hard.

This version of Problem 1 is defined by replacing the feasibility objective with that of k-feasibility:

Definition 5 (k-feasibility): The transfer $x(t_0) \to x(t_1)$ is k-feasible if and only if there exists k-sparse[3] B such that $x(t_0) \to x(t_1)$ is feasible by (A, B).

To present our instance of the decision Problem 1 that reduces to the k-hitting set problem, let $|\mathcal{L}| = p$ and $\mathcal{M} = \{1, 2, \ldots, m\}$, with respect to Definition 4, and define $\Phi \in \mathbb{R}^{p \times m}$ such that $\Phi_{ij} = 1$ if the i-th set contains the element j and zero otherwise.

Lemma 1: For $i \in \mathbb{N}$, denote as $e_{i \times l}$ the $i \times l$ matrix of all-ones and set $n = m + p + 1$, $A = V_1^{-1}\text{diag}(1, 2, \ldots,$

[1]A matrix is *sparse* if it has a small number of non-zero elements compared to each dimension.

[2]The *sparsity* of a matrix is the number of its non-zero elements.

[3]A matrix is *k-sparse* if it has k non-zero elements.

$m+p+1)V_1$, where[4]

$$V_1 = \left[\begin{array}{ccc} 2I_{m\times m} & 0_{m\times p} & e_{m\times 1} \\ \Phi & (m+1)I_{p\times p} & 0_{p\times 1} \\ 0_{1\times m} & 0_{1\times p} & 1 \end{array} \right],$$

and $x(t_0) = 0$, as well as, $\chi = V_1^{-1} e_{n\times 1}$. For any $t_1 > t_0$, $0 \to x(t_1) = \chi$ is $k+1$-feasible if and only if $\mathcal{L}$ has a k-hitting set.

Therefore, with Lemma 1 we provide an instance of Problem 1 that is $k+1$-feasible if and only if any instance of $\mathcal{L}$, (that is, also the hardest ones with respect to the hitting set problem), has a k-hitting set. Hence (cf. [11]):

Theorem 1: Problem 1 is NP-hard.

Thereby, the generalized version of Problem 1, with respect to Objective 3, is NP-hard as well (for the above instance where we additionally set $W = I$).

We illustrate the proof Lemma 1: The instance of A and the initial and final condition are constructed so that the $0 \to \chi$ is $k+1$-feasible if and only if there exists $k+1$-sparse B such that (A, B) is controllable; on the other hand, the latter holds if and only if $\mathcal{L}$ has a k-hitting set [1]. Thereby, the theorem follows. Additionally, due to the controllability properties of linear time-invariant systems [22], it holds for any $t_1 > t_0$.

Lemma 1 extends to the case where B is a column zero-one vector as well. Furthermore, in Theorem 1 the assumption $x(t_0) = 0$ is without loss of generality, since we consider the linear dynamics (1) [22].

In the following paragraphs, we prove the NP-hardness of Problem 2.

2) Intractability of Problem 2: We prove that the decision version of Problem 2 reduces to the k-hitting set and, as a result, that Problem 2 is NP-hard.

This version of Problem 2 is defined by replacing the reachability objective with that of k-reachability:

Definition 6 (k-reachability): The subset $\mathcal{N} \subseteq \mathbb{R}^n$ is k-reachable if and only if there exists k-sparse B such that $\mathcal{N}$ is reachable by (A, B).

To present our instance of the decision Problem 2 that reduces to the k-hitting set problem, let $|\mathcal{L}| = p$ and $\mathcal{M} = \{1, 2, \ldots, m\}$, with respect to Definition 4, and define $\Phi \in \mathbb{R}^{p\times m}$ such that $\Phi_{ij} = 1$ if the i-th set contains the element j and zero otherwise.

Lemma 2: Set $\mathcal{N} = \{(x_1, x_2, \ldots, x_n) : x_1 = x_2 = \ldots = x_m = 0, x_{m+1}, x_{m+2}, \ldots, x_{m+p} > 0\}$ and

$$A = \left[\begin{array}{cc} 0_{m\times m} & 0_{m\times p} \\ \Phi & 0_{p\times p} \end{array} \right].$$

$\mathcal{N}$ is k-reachable if and only if $\mathcal{L}$ has a k-hitting set.

Therefore, with Lemma 2 we provide an instance of Problem 2 that is k-feasible if and only if any instance of $\mathcal{L}$, (that is, also the hardest ones with respect to the hitting set problem), has a k-hitting set. Hence (cf. [11]):

Theorem 2: Problem 2 is NP-hard.

Thereby, the generalized version of Problem 2, with respect to Objective 4, is NP-hard as well (for the above instance where we additionally set $W = I$).

Since Problems 1 and 2 are NP-hard, we need in the worst case to provide approximate algorithms for their solution; this is the subject of the next section.

B. Approximation Algorithms for the Minimal Reachability Problems

We provide efficient approximation algorithms for the general solution of Problems 1 and 2.

To implement an approximation algorithm for Problem 1, we use Definition 3 to relax Objective 1, by replacing the feasibility of $x(t_0) \to x(t_1)$ with that of ϵ-close feasibility. We then provide Algorithm 1, that returns a B with sparsity up to a multiplicative factor of $O(\ln(\epsilon^{-1}))$ from any optimal solution of the original Problem 1.

Next, to address Problem 1 with respect to Objective 1, we prove that for all $\epsilon \leq \epsilon(A)$, where $\epsilon(A)$ is positive and sufficiently small, Definitions 1 and 3 still coincide; hence, we implement a bisection-type execution of Algorithm 1, Algorithm 2, that quickly converges to an $\epsilon \leq \epsilon(A)$ and, as a result, returns a B that makes the exact transfer $x(t_0) \to x(t_1)$ feasible.

Finally, using Algorithm 1, we provide an approximation algorithm for Problem 2 as well.

1) Approximation Algorithm for Problem 1: We develop the notation and tools that lead to an efficient approximation algorithm for Problem 1.

For $\mathcal{N} \subseteq \mathbb{R}^n$ and $v \in \mathbb{R}^{n\times 1}$, we denote as $v[\mathcal{N}]$ the projection of v onto $\mathcal{N}$ and as $\|v\|$ its euclidean norm. Moreover, we denote as $\mathcal{C}(A)$ the set of columns of $\left[I|A|\ldots|A^{n-1}\right]$, as e_i the i-th unit vector and as C_i the set of columns $\{e_i, Ae_i, \ldots, A^{n-1}e_i\}$. For B per Assumption 1, we set

$$\mathcal{S}(B) \equiv \operatorname{span}\left[B|AB|\ldots|A^{n-1}B\right].$$

Since the dynamics (1) are linear, $x(t_0) \to x(t_1)$ is feasible if and only if $0 \to x(t_1) - \exp[A(t_1 - t_0)]x(t_0) \equiv v(t_1)$ is. Moreover, since these dynamics are also continuous and time-invariant, whenever $0 \to v(t_1)$ is feasible for some $t_1 > t_0$, it is also for any $t_1' > t_0$ [22]. Hence, we study directly $0 \to v$, suppressing t_1.

In particular, $0 \to v$ is feasible if and only if $v \in \mathcal{S}(B)$ [22]. Therefore, $0 \to v$ is feasible if and only if $v = v[\mathcal{S}(B)]$: if $v = v[\mathcal{S}(B)]$, $v \in \mathcal{S}(B)$, while, if $v \neq v[\mathcal{S}(B)]$, $v - v[\mathcal{S}(B)] \in \mathcal{S}(B)^{\perp}$, that is, $v \notin \mathcal{S}(B)$[5]. Similarly, $0 \to v$ is feasible if and only if $\|v\| = \|v[\mathcal{S}(B)]\|$: if $v = v[\mathcal{S}(B)]$, $\|v\| = \|v[\mathcal{S}(B)]\|$, while, if $v \neq v[\mathcal{S}(B)]$, $\|v[\mathcal{S}(B)]\| < \|v\|$.

Definition 3 is restated as follows:

Definition 7 (ϵ-close feasibility): The $0 \to v$ is *ϵ-close feasible* by (A, B) if and only if $\|v\|^2 - \|v[\mathcal{S}(B)]\|^2 \leq \epsilon$.

Remark 1: Since $v - v[\mathcal{S}(B)]$ is orthogonal to $v[\mathcal{S}(B)]$, $\|v[\mathcal{S}(B)]\|^2 + \|v - v[\mathcal{S}(B)]\|^2 = \|v\|^2$ and, as a result, ϵ-close feasibility implies $\|v - v[\mathcal{S}(B)]\|^2 \leq \epsilon$.

[4] V_1 is invertible since it strictly diagonally dominant.

[5] $\mathcal{S}(B)^{\perp}$ is the orthogonal complement of $\mathcal{S}(B)$.

We provide the following greedy approximation algorithm for Problem 1 with respect to the relaxed feasibility objective of Definition 7. Its quality of approximation is quantified in Theorem 3.

Algorithm 1 Approximation Algorithm for the relaxed Problem 1 with respect to Definition 7.

Input: Matrix $\mathcal{C}(A)$, vector $v \equiv x(t_1) - \exp[A(t_1 - t_0)]x(t_0)$, approximation level ϵ.
Output: B such that $x(t_0) \to x(t_1)$ is ϵ-close feasible.

```
B = 0_{n×n}.
while ‖v‖² − ‖v[S(B)]‖² > ε do
    Find an i ∈ [n] such that: i) B_ii = 0 and ii) i is a
  maximizer for ‖v[S(B) + span{C_i}]‖² − ‖v[S(B)]‖².
    Set B_ii = 1.
end while
```

Theorem 3: Given the transfer $x(t_0) \to x(t_1)$, denote as $B^\star$ an optimal solution to Problem 1 and as B the corresponding output of Algorithm 1. Then, $x(t_0) \to x(t_1)$ is ϵ-close feasible by (A, B) and

$$\sum_{i=1}^{n} B_{ii} \leq \lceil \ln(\|x(t_1) - \exp[A(t_1 - t_0)]x(t_0)\|^2/\epsilon) \rceil \sum_{i=1}^{n} B_{ii}^\star.$$

That is, the polynomial time approximation Algorithm 1 returns a B with sparsity up to a multiplicative factor of $O(\ln(\epsilon^{-1}))$ from any optimal solution of the original Problem 1, and makes the $x(t_0) \to x(t_1)$, or $0 \to v$, ϵ-close feasible.

Next, to address Problem 1 with respect to Objective 1, we show that there exists $\epsilon(A)$, positive, such that for any $\epsilon \leq \epsilon(A)$, Definitions 1 and 3 coincide. Thereby, running Algorithm 1 with $\epsilon \leq \epsilon(A)$, results to a B that makes the exact transfer $x(t_0) \to x(t_1)$ feasible.

In particular, for $i \in [n]$, let $C_i \equiv \{e_i, Ae_i, \ldots, A^{n-1}e_i\}$; that is, C_i is the sub-matrix of $\mathcal{C}(A)$ that is also present in $\left[B|AB|\ldots|A^{n-1}B\right]$ if and only if $B_{ii} = 1$. Moreover, for $S \subseteq [n]$, consider $B_{ii} = 1$ if and only if $i \in S$. Moreover, assume that $0 \to v$ is infeasible by B, i.e., $v[\text{span}\{\bigcup_{j\in S} C_j\}] \neq v$. Then, denote as $\Xi(S)$ the event where $0 \to v$ can become feasible by making one more element of B one, that is, $\Xi(S) \equiv \{v[\text{span}\{\bigcup_{j\in S} C_j\}] \neq v \text{ and } \exists i \in [n] \setminus S, v[\text{span}\{\left(\bigcup_{j\in S} C_j\right) \cup C_i\}] = v\}$. It is,

$$\epsilon(A) = \min_{S\subseteq[n]:\Xi(S) \text{ is true.}} \left(\|v\|^2 - \|v[S]\|^2\right).$$

Therefore, $\epsilon(A)$ is positive.

In general, $\epsilon(A)$ is unknown in advance. Hence, we need to search for a sufficiently small value of ϵ so that $\epsilon \leq \epsilon(A)$. Since ϵ is lower and upper bounded by 0 and $\|v\|^2$, respectively, we achieve this by performing a binary search. In particular, we implement Algorithm 2, where we denote as $[\text{Algorithm1}](\mathcal{C}(A), 0 \to v, \epsilon)$ the matrix that Algorithm 1 returns for given A, v and ϵ.

In the worst case, when we first enter the `while` loop, the `if` condition is not satisfied and, as a result, ϵ is set to a lower value. This process continues until the `if` condition is satisfied for the first time, from which point and on, the algorithm converges, up to the accuracy level a, to $\epsilon(A)$; specifically, $|\epsilon - \epsilon(A)| \leq a/2$, due to the mechanics of the bisection. Then, Algorithm 2 exits the `while` loop and the last `if` statement ensures that ϵ is set below $\epsilon(A)$ so that $0 \to v$ is feasible.

Algorithm 2 Approximation Algorithm for Problem 1.

Input: Matrix $\mathcal{C}(A)$, vector $v \equiv x(t_1) - \exp[A(t_1 - t_0)]x(t_0)$, bisection's accuracy level a.
Output: B such that $x(t_0) \to x(t_1)$ is feasible.

```
B = 0_{n×n}, l ← 0, u ← ‖v‖², ε ← (l + u)/2
while u − l > a do
B ← [Algorithm1](C(A), 0 → v, ε)
   if ‖v‖² − ‖v[S(B)]‖² > ε then
      u ← ε
   else
      l ← ε
   end if
ε ← (l + u)/2
end while
if ‖v‖² − ‖v[S(B)]‖² > ε then
   u ← ε, ε ← (l + u)/2
end if
B ← [Algorithm1](C(A), 0 → v, ε)
```

The efficiency of Algorithm 2 for Problem 1 is summarized below.

Corollary 1: Given the transfer $x(t_0) \to x(t_1)$, denote as $B^\star$ an optimal solution to Problem 1 and as B the corresponding output of Algorithm 2. Then, $x(t_0) \to x(t_1)$ is feasible by (A, B) and

$$\sum_{i=1}^{n} B_{ii} \leq \lceil \ln(\|x(t_1) - \exp[A(t_1 - t_0)]x(t_0)\|^2/\epsilon) \rceil \sum_{i=1}^{n} B_{ii}^\star.$$

where ϵ is the approximation level where Algorithm 2 had converged when terminated.

The results of this section apply to the generalized version of Problem 1 with respect to Objective 3 by replacing $\mathcal{C}(A)$, C_i and $\mathcal{S}(B)$ with $W\mathcal{C}(A)$, WC_i and span $\left[WB|WAB|\ldots|WA^{n-1}B\right]$, respectively (where W is the output matrix of (1)). Similarly with regard to the approximation algorithm described below.

2) Approximation Algorithm for Problem 2: Due to space limitations, the approximation algorithm for Problem 2 is omitted; it can be found in the full version of this paper, located at the authors websites.

We illustrate our analytical findings, and test their performance, in the next section.

IV. Examples and Discussions

A. Star Network

We illustrate the mechanics and efficiency of Algorithm 2 using the star network of Fig. 1, where $n = 4$ and

$$A = \begin{bmatrix} -1 & 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

In particular, we run Algorithm 2 for the $\tau_1 \equiv 0 \to (1,0,0,0,0)$, $\tau_2 \equiv 0 \to (0,1,1,0,0)$ and $\tau_3 \equiv 0 \to (1,1,1,0,0)$ and for $a = .001$. The algorithm returned a B equal to $\text{diag}(1,0,0,0,0)$, $\text{diag}(0,1,1,0,0)$ and $\text{diag}(0,1,1,0,0)$, respectively; indeed, τ_1 is feasible by the minimum number of actuators if and only if either $x_0(t)$ is actuated or one among $x_1(t), x_2(t), x_3(t), x_4(t)$ is; τ_2 is feasible by the minimum number of actuators if and only if $x_1(t)$ and $x_2(t)$ are actuated and, finally, τ_3 is feasible by the minimum number of actuators if and only if $x_1(t)$ and $x_2(t)$ are actuated. Overall, Algorithm 2 operated optimally.

Evidently, this star network is controllable by the minimum number of actuators if and only if all $x_1(t), x_2(t), x_3(t), x_4(t)$ are actuated. Therefore, whenever we are interested merely in the feasibility of a state transfer, it is cost-effective, with respect to the number of actuators that should be implemented, to design a B that does not result to a controllable system as well.

B. Erdős-Rényi Random Networks

Erdős-Rényi random graphs are commonly used to model real-world networked systems [23]. According to this model, each edge is included in the generated graph with some probability p independently of every other edge. We implemented this model for varying network sizes n where the directed edge probabilities were set to $p = 2\log(n)/n$. In particular, we first generated the binary adjacencies matrices for each network size so that each edge is present with probability p and then we replaced every non-zero entry with an independent standard normal variable to generate a randomly weighted graph. The network size varied from 1 to 100, with step 1.

For each network size, we run Algorithm 2 for a $0 \to \chi$, where χ was randomly generated using MATLAB's "randn" command; for all cases, the algorithm returned a 1-sparse B. This is in accordance with the simulation results of [1], where similarly randomly generated networks were made controllable by actuating one or two states.

Extending the simulations of this section to the algorithm for Problem 2 is straightforward and, as a result, due to space limitations we omit this discussion.

V. Concluding Remarks

We addressed a collection of state (and output) space reachability problems for a linear system, under the additional objective of sparse control, i.e., the control using a minimal number of actuators. In particular, we proved that these problems are NP-hard and provided efficient approximation algorithms for their general solution, along with worst case approximation guarantees. Finally, we illustrated the efficiency of these algorithms with a set of simulations. Optimal behaviour was observed.

References

[1] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, Sept 2014.

[2] S. Pequito, G. Ramos, S. Kar, A. P. Aguiar, and J. Ramos, "On the Exact Solution of the Minimal Controllability Problem," *ArXiv e-prints*, Jan. 2014.

[3] M. Amin and J. Stringer, "The electric power grid: Today and tomorrow," *MRS bulletin*, vol. 33, no. 04, pp. 399–407, 2008.

[4] *California Partners for Advanced Transit and Highways*, 2006. [Online]. Available: http://www.path.berkeley.edu/

[5] S. Gu, F. Pasqualetti, M. Cieslak, S. T. Grafton, and D. S. Bassett, "Controllability of Brain Networks," *ArXiv e-prints*, Jun. 2014.

[6] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, ser. Princeton Series in Applied Mathematics. Princeton University Press, 2010.

[7] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[8] J. Cortes, S. Martinez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, Aug 2006.

[9] E. A. Lee, et al., "The swarm at the edge of the cloud," *IEEE Design Test*, vol. 31, no. 3, pp. 8–20, June 2014.

[10] V. M. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. Pappas, "Optimal Resource Allocation for Network Protection Against Spreading Processes," *ArXiv e-prints*, Sep. 2013.

[11] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[12] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *ArXiv e-prints*, Apr. 2014.

[13] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, March 2014.

[14] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with optimal control constraints," in *Proceedings of the American Control Conference (ACC)*, 2015.

[15] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Transactions on Control of Network Systems*, 2015, in press.

[16] 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 (CDC)*, 2014.

[17] U. Munz, M. Pfister, and P. Wolfrum, "Sensor and actuator placement for linear systems based on h_2 and h_∞ optimization," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2984–2989, Nov 2014.

[18] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[19] H. Jamali-Rad, A. Simonetto, and G. Leus, "Sparsity-aware sensor selection: Centralized and distributed algorithms," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 217–220, Feb 2014.

[20] V. D. Blondel and J. N. Tsitsiklis, "A survey of computational complexity results in systems and control," *Automatica*, vol. 36, no. 9, pp. 1249–1274, 2000.

[21] M. Egerstedt and V. Blondel, "How hard is it to control switched systems?" in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3, 2002, pp. 1869–1873 vol.3.

[22] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. New York, NY, USA: Oxford University Press, Inc., 1998.

[23] M. Newman, A.-L. Barabási, and D. Watts, *The structure and dynamics of networks*. Princeton University Press, 2006.

[24] D. Moshkovitz, "The projection games conjecture and the np-hardness of ln n-approximating set-cover," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2012, pp. 276–287.