2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

A Randomized Algorithm for Minimum Cost Constrained Input Selection for State Space Structural Controllability

Shana Moothedath, Prasanna Chaporkar and Madhu N. Belur

Abstract*—This paper addresses structural controllability problem in linear time invariant dynamical structured system: the minimum cost constrained input selection problem (minCCIS). Constraints are imposed on the input structure and the aim is to obtain an input selection that minimizes the cost while satisfying the pre-specified constraints on the input matrix. This problem is NP-hard. As a result, our focus is to devise a computationally efficient framework to obtain a solution arbitrarily close to an optimum solution. We propose a randomized algorithm based on Markov Chain Monte Carlo (MCMC) technique. Firstly, we give a polynomial time reduction of the problem to an optimization problem on a system specific bipartite graph constructed, where the objective is to maximize a utility function associated with the matchings in the bipartite graph. Subsequently, we prove that an optimum solution of this bipartite matching problem when translated back gives an optimum solution to the minCCIS problem. Using a randomized algorithm based on the MCMC technique for solving the utility based matching problem, we show that an optimum solution to minCCIS can be found in time polynomial in the number of state variables with high probability. Simulation results that demonstrates the performance of the proposed scheme is also presented.

1. Introduction

Complex networks are an indispensable constituent in most of the networks emerging from science and environment, including social, biological, transportation, distribution and technological networks. This paper deals with controllability of linear time invariant (LTI) complex dynamical systems. Typically, networks are represented by the state matrix $A \in \mathbb{R}^{n\times n}$ and the input matrix $B \in \mathbb{R}^{n\times m}$, where n is the number of states and m is the number of inputs. Here $\mathbb{R}$ denotes the set of real numbers. In complex networks all the system parameters are not precisely known. Further, in some cases even though the graph of the network remains unaltered, the dynamics varies with time, for instance traffic in a transportation network is time varying. Thus, checking controllability of complex networks using conventional control tools is not always feasible. Many papers perform graph theoretic analysis, referred as *structural analysis*, using the sparsity pattern of the system matrices to address various system theoretic problems [1].

In this paper, we analyze controllability of large-scale systems using structural analysis. Due to the large system dimension, devising efficient algorithms to solve various optimization problems associated with these networks is important and at the same time challenging. Controllability of systems with a specified graph pattern is referred as *structural controllability*. In the last few decades, structural controllability and various associated optimization problems were studied and graph theoretic formulations of many of these were done [1], [2], [3], [4], [5] and [6]. Our focus is on optimal selection of inputs for achieving structural controllability of a structured system.

This paper addresses the minimum cost constrained input selection problem (minCCIS). The costs associated with inputs are motivated from the installation, maintenance and monitoring costs associated with it or even the selection preferences. Selection preferences are important in many applications, like leader selection where certain agents are preferred over others for performing some specific tasks [7] and this can be captured as cost associated with the input. Given a structured system, minCCIS aims at finding a minimum cost input set for making the system structurally controllable. If costs are non-zero and uniform, then minCCIS solves minimum cardinality input selection, which is known to be NP-hard [8]. Thus minCCIS is also NP-hard. Under the assumption that the structured system is irreducible[1], a polynomial time solution for a subclass of this problem is given in [6]. These problems are addressed in their generality in [9], where the authors proposed a deterministic polynomial time approximation algorithm and proved its approximation ratio. In this work, we address both these problems in full generality using a randomized scheme. We propose an exact randomized algorithm based on Markov Chain Monte Carlo (MCMC) technique for solving both these problems and show that an optimal solution can be obtained in polynomial time with high probability.

Key contributions of this paper are summarized below.

- We reduce the minimum cost constrained input selection (minCCIS) problem, to a bipartite matching problem with utility maximization (BMUM) in polynomial time (Algorithm 4.1).
- We prove that an optimum solution to the BMUM problem obtained after reduction gives an optimum solution to the minCCIS problem (Theorem 4.1).
- We propose a randomized algorithm based on MCMC to solve the BMUM formulation of the minCCIS problem (Algorithm 5.1).
- We prove that the proposed randomized algorithm gives an optimal solution to the minCCIS problem in $O(n^{2.5}) + Poly(n, \log n)$, where n is the number of states and $Poly(n, \log n)$ is a polynomial in n and $\log(n)$ (Theorem 5.4).

The organization of this paper is as follows: Section 2 is devoted for problem formulation. Graphical representation of structured systems and few existing results is detailed in Section 3. Section 4 explains the bipartite matching problem with utility maximization and the reduction of the constrained minimum input problems to a matching problem with utility maximization. Section 5 presents a randomized algorithm based on MCMC for solving the minCCIS problem. Section 6 discusses simulation results conducted for checking the convergence of the proposed method. Finally, Section 7 gives the concluding remarks.

2. Problem Formulation

Consider the state space representation of an LTI system, $\dot{x}(t) = Ax(t) + Bu(t)$, where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ denote the state matrix and the input matrix respectively. The structured matrices, $\bar{A} \in \{0, \star\}^{n\times n}$ and $\bar{B} \in \{0, \star\}^{n\times m}$, corresponding to the system matrices A and B are defined as

$$\begin{aligned} A_{ij} &= 0 \text{ whenever } \bar{A}_{ij} = 0, \text{ and} \\ B_{ij} &= 0 \text{ whenever } \bar{B}_{ij} = 0. \end{aligned} \tag{1}$$

Here, 0's in $\bar{A}$ and $\bar{B}$ fixes the corresponding entries in A and B matrices as zeros and $\star$ denotes a free independent parameter entry in those locations. In short, the sparsity pattern of (A, B) and $(\bar{A}, \bar{B})$ pairs are the same and hence $(\bar{A}, \bar{B})$ represents an equivalence class of systems. A pair $(\bar{A}, \bar{B})$ that satisfies equation (1) is referred as

The authors are in the Department of Electrical Engineering, Indian Institute of Technology Bombay, India. Email: {shana, chaporkar, belur}@ee.iitb.ac.in.

[1]A digraph is said to be irreducible if there exists a directed path between any two arbitrary nodes in it.

the *structured system* representation of the system with *numerical realization* (A,B). The strength of structural controllability is that it is a generic property. Thus, structural controllability implies controllability of "almost all" pairs of (A,B) which has the same structure as $(\bar{A},\bar{B})$ [10]. Before formulating the problem formally, we define structural controllability.

Definition 2.1. *[10] The pair $(\bar{A},\bar{B})$ is said to be structurally controllable if there exists at least one controllable numerical realization (A,B) with the structure specified by $\bar{A}$ and $\bar{B}$.*

Consider $\mathcal{W} \subseteq \{1,\ldots,m\}$ and let $\bar{B}_{\mathcal{W}}$ be the restriction of $\bar{B}$ to columns only in $\mathcal{W}$. Furthermore, let $\mathcal{K} = \{\mathcal{W} : (\bar{A},\bar{B}_{\mathcal{W}})$ is structurally controllable$\}$. Thus $\mathcal{K}$ is the set of all feasible solutions to the optimization problem addressed in this paper. Note that the set $\mathcal{K}$ is non-empty. This is because if $\mathcal{W} = \{1,\ldots,m\}$, $(\bar{A},\bar{B}_{\mathcal{W}}) = (\bar{A},\bar{B})$ is structurally controllable. The *minimum cost constrained input selection* problem (minCCIS) is posed as: given structurally controllable $(\bar{A}$, $\bar{B})$ and c_u, a vector where every entry $c_u(j)$, $j = 1,2,\ldots,m$, indicates the cost of actuating input u_j, the minCCIS problem consists of finding a minimum cost input selection in $\bar{B}$ such that the system is structurally controllable.

Problem 2.2. *Given structurally controllable $(\bar{A},\bar{B})$ and $c_u(j), j = 1,2,\ldots,m$, find $\mathcal{I}^\star \in \arg\min_{\mathcal{I}\in\mathcal{K}} \sum_{j\in\mathcal{I}} c_u(j)$.*

If the costs are non-zero and uniform, then minCCIS solves minimum cardinality input selection from a constrained input set for structural controllability, referred as the minCIS problem [8]. We analyze and solve only Problem 2.2 here. The proposed analysis and results directly apply to minCIS also.

Remark 2.3. *Note that our results can be generalized to any cost function $c_u : \mathcal{K} \to \mathbb{R}$ that can be computed in $O(Poly(n))$.*

The minCIS problem is shown to be NP-hard [8] and hence minCCIS is also NP-hard. Our approach here is to provide an efficient technique based on randomized algorithms for solving miCCIS problem. To this end, we formulate Problem 2.2 as a bipartite matching[2] problem on a system specific bipartite graph, where the objective is to maximize a utility function associated with the matchings in it. The construction of this bipartite graph is elaborated in Section 4. Before discussing bipartite matching formulations of these problems, we give few graph theoretic concepts and some constructions used in the sequel in the section below.

3. Graphical Representation of Structured Systems

In this section, we define few graph theoretic terminologies associated with the structured system $(\bar{A},\bar{B})$. Given a structured state matrix $\bar{A}$, the digraph representation $\mathcal{D}(\bar{A})$ is a triple consisting of a *vertex set* V_X, an *edge set* E_X, and a function assigning each edge an ordered pair of vertices. The set V_X called as the state nodes is defined as $V_X := \{x_1,x_2,\ldots,x_n\}$ and the edge set is defined as $(x_j,x_i) \in E_X$ if $\bar{A}_{ij} = \star$. Similarly, the system digraph representation of $(\bar{A},\bar{B})$ is denoted as $\mathcal{D}(\bar{A},\bar{B}) := (V_X \cup V_U, E_X \cup E_U)$, where the vertex set $V_U := \{u_1,\ldots,u_m\}$ are the input nodes. The edge set of $\mathcal{D}(\bar{A},\bar{B})$ is the union of edge sets E_X and E_U, where $(u_j,x_i) \in E_U$ if $\bar{B}_{ij} = \star$. Further, the input u_j is said to be assigned to a state variable x_i if $\bar{B}_{ij} = \star$. Thus the digraphs $\mathcal{D}(\bar{A})$ and $\mathcal{D}(\bar{A},\bar{B})$ captures the influences of states and inputs on the dynamics of each state. Using the digraphs constructed here a graph theoretic condition for checking structural controllability of structured systems exists which requires understanding of two concepts, i.e., accessibility and dilation, elaborated below.

In a digraph, a vertex v_i is said to be reachable from another vertex v_j if there exists a directed path from v_j to v_i. A state node x_i, for $i = 1,2,\ldots,n$, is said to be *inaccessible* if node x_i is not reachable from any input vertex. More precisely, an inaccessible node is one which cannot be influenced by an input. On the other hand, a digraph $\mathcal{D}(\bar{A},\bar{B})$ is said to have a *dilation*, if given a set of nodes $S \subset V_X$, the neighbourhood node set of S, $T(S)$ has fewer nodes than S. Here $x_i \in T(S)$, if there exist a directed edge from x_i to a node in S and $u_i \in T(S)$, if there exist a directed edge from u_i to a node in S. Notice that $S \subseteq V_X$ and $T(S) \subseteq V_X \cup V_U$. Lin proved a necessary and sufficient condition for structural controllability using the concepts of accessibility and dilation.

Proposition 3.1. *[11] The system (A,B) is structurally controllable, if and only if the digraph associated with it, i.e., $\mathcal{D}(\bar{A},\bar{B})$ has no inaccessible nodes and no dilations.*

Accessibility crucially depends on the connectivity of the graph. A digraph $\mathcal{D}$ is said to be strongly connected if for each pair of ordered vertices v_i,v_j, there exists an elementary path from v_i to v_j. If a digraph is strongly connected, then it is said to be an irreducible digraph. A subgraph of a graph $G = (V_G,E_G)$ is a graph $H = (V_H,E_H)$ such that $V_H \subseteq V_G$ and $E_H \subseteq E_G$ and the assignment of the endpoints to edges in H is the same as in G. A maximal strongly connected subgraph is a subgraph that is strongly connected and is not properly contained in any other subgraph that is strongly connected. Using these the following definition holds.

Definition 3.2. *[12] A strongly connected component (SCC) of a digraph $\mathcal{D}$ is a maximal strongly connected subgraph of $\mathcal{D}$.*

By condensing each SCC in $\mathcal{D}(\bar{A})$ as a supernode, we generate a directed acyclic graph (DAG) in which each super node corresponds to a single SCC and a directed edge exists between two SCCs *if and only if* there exists a directed edge connecting vertices in the respective SCCs in the original digraph. A characterization of the SCCs in the digraph $\mathcal{D}(\bar{A})$ is given in Definition 3.3.

Definition 3.3. *An SCC is said to be non-top linked if it has no incoming edges to its vertices from the vertices of another SCC.*

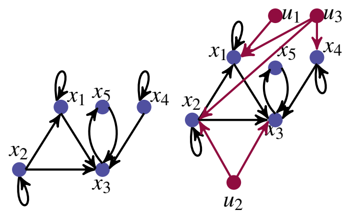
By composing the input nodes on the DAG formed by condensing the SCCs of $\mathcal{D}(\bar{A})$, accessibility of $\mathcal{D}(\bar{A},\bar{B})$ can be verified. All state nodes in $\mathcal{D}(\bar{A},\bar{B})$ are said to be accessible if and only if all non-top linked SCCs in $\mathcal{D}(\bar{A})$ are input accessible. Given a digraph $\mathcal{D} = (V_D,E_D)$, the SCCs and the DAG associated with it can be obtained in $O(|V_D|+|E_D|)$ computations [12]. Thus accessibility of a system can be verified in polynomial number of computations. In order to analyze the dilation condition, there exists an equivalent bipartite graph matching condition. For this the state and the system bipartite graphs $\mathcal{B}(\bar{A})$ and $\mathcal{B}(\bar{A},\bar{B})$ are constructed as follows: the bipartite graph $\mathcal{B}(\bar{A})$ is constructed with vertex set $(V_{X'} \cup V_X)$ and edge set $\mathcal{E}_X$, where $V_X = \{x_1,x_2,\ldots,x_n\}$, $V_{X'} = \{x'_1,x'_2,\ldots,x'_n\}$ and $(x'_j,x_i) \in \mathcal{E}_X \Leftrightarrow (x_i,x_j) \in E_X$. Similarly, the system bipartite graph $\mathcal{B}(\bar{A},\bar{B})$ is constructed with vertex set $(V_{X'} \cup (V_X \cup V_U))$ and edge set $\mathcal{E}_X \cup \mathcal{E}_U$. Here, $V_U = \{u_1,u_2,\ldots,u_m\}$ and $(x'_j,u_i) \in \mathcal{E}_U \Leftrightarrow (u_i,x_j) \in E_U$. Using $\mathcal{B}(\bar{A},\bar{B})$ the following result holds.

Proposition 3.4 (Theorem 2, [4])**.** *A digraph $\mathcal{D}(\bar{A},\bar{B})$ has no dilations if and only if the bipartite graph $\mathcal{B}(\bar{A},\bar{B})$ has a perfect matching.*

Taken together, the structured system $(\bar{A},\bar{B})$ is said to be structurally controllable if and only if all non-top linked SCCs are

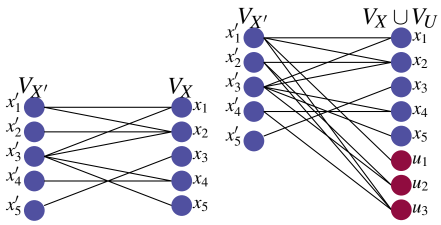
[2] A bipartite graph $G_B = ((V_1 \cup V_2),E_B)$ is a graph satisfying $V_1 \cap V_2 = \emptyset$ and $E_B \subseteq V_1 \times V_2$. A matching is a set of edges such that no two edges share the same end point. For a bipartite graph $G_B = ((V_1 \cup V_2),E_B)$, a perfect matching is a matching whose cardinality is equal to $\min(|V_1|,|V_2|)$.

$$\bar{A}=\begin{bmatrix}\star&\star&0&0&0\\0&\star&0&0&0\\\star&\star&0&\star&\star\\0&0&0&\star&0\\0&0&\star&0&0\end{bmatrix},\bar{B}=\begin{bmatrix}\star&0&\star\\0&\star&\star\\0&\star&0\\0&0&\star\\0&0&0\end{bmatrix}$$



(a) System $(\bar{A},\bar{B})$ (b) $\mathcal{D}(\bar{A})$ (c) $\mathcal{D}(\bar{A},\bar{B})$

Figure 1: The state digraph and system digraph representation of the structured system $(\bar{A},\bar{B})$ given in Figure 1a are shown in Figure 1b and Figure 1c respectively.



(a) $\mathcal{B}(\bar{A})$ (b) $\mathcal{B}(\bar{A},\bar{B})$

Figure 2: The bipartite graphs $\mathcal{B}(\bar{A})$ and $\mathcal{B}(\bar{A},\bar{B})$ of the structured system $(\bar{A},\bar{B})$ given in Figure 1a.

input accessible and there exists a perfect matching in $\mathcal{B}(\bar{A},\bar{B})$. An illustrative example demonstrating the construction of the digraphs associated with a structured system is given in Figure 1. In the example given in Figure 1, there are four SCCs listed as: $\mathscr{C}_1=\{x_1\},\mathscr{C}_2=\{x_2\},\mathscr{C}_3=\{x_3,x_5\}$ and $\mathscr{C}_4=\{x_4\}$. The non-top linked SCCs are $\mathscr{N}_1=\{x_2\}$ and $\mathscr{N}_2=\{x_4\}$. Figure 2 illustrates the bipartite graph construction for this structured system. Next, we reduce the minCCIS problem to a bipartite matching problem with utility maximization.

4. Bipartite matching representation of minCCIS

In this section, we first brief about bipartite matching problem with utility maximization (BMUM). Then, we detail a polynomial time reduction of the minCCIS problem to an instance of the BMUM problem. On obtaining a solution to the matching problem we show how to translate it back to the corresponding solution of the minCCIS problem. We also prove that an optimum solution of the matching problem when translated back corresponds to an optimum solution of the minCCIS problem. Thus, any algorithm that solves the BMUM problem also solves the minCCIS problem. The BMUM problem is described below.

A. Bipartite Matching with Utility Maximization

Consider a *complete*[3] *bipartite graph* $G=((V_1\cup V_2),L)$, where $|V_1|=n_1$, $|V_2|=n_2$ and $(i,j)\in L$ for every $i\in V_1$ and $j\in V_2$. Without loss of generality, let $n_1\leqslant n_2$. A *matching* is a set of edges, $M\subseteq L$ such that no two edges share a common node, i.e., degree of each node in M is one. A *perfect matching* M in the bipartite graph $G=((V_1\cup V_2),L)$ is a matching such that $|M|=\min(|V_1|,|V_2|)$. For a complete graph G, a perfect matching M exists and $|M|=n_1$, since $n_1\leqslant n_2$. Let Ω denote the set of all perfect matchings in G and let $\mathcal{U}:\Omega\to\mathbb{R}$ be a real valued function. Note that $|\Omega|=O(n_2^{n_1})$. The function $\mathcal{U}$ can be thought as assigning utility to each perfect matching and the aim is to find a perfect matching $M^\star$, where $M^\star\in\arg\max_{M\in\Omega}\ \mathcal{U}(M)$.

Even though there is no specific structure imposed on $\mathcal{U}(\cdot)$, we assume that given any perfect matching M, $\mathcal{U}(M)$ can be computed

[3]A bipartite graph $G=((V_1\cup V_2),L)$ is said to be complete if $L=V_1\times V_2$, i.e., all left side nodes have an edge to all the right side nodes in the graph.

Algorithm 4.1 Pseudo-code for reducing minCCIS to bipartite matching with utility maximization

Input: Structured matrices $\bar{A}\in\{0,\star\}^{n\times n}$, $\bar{B}\in\{0,\star\}^{n\times m}$ and cost vector $c_u\in\mathbb{R}^m$
Output: Bipartite graph $G_B=((V,\widetilde{V}),E)$ and utility function $\mathcal{U}(\cdot)$
1: Initialize set $\mathscr{B}_i=\phi$, for $i=1,\dots,m$
2: Find the non-top linked SCCs in $\mathcal{D}(\bar{A})$, $\mathscr{N}=\{\mathscr{N}_i\}_{i=1}^q$
3: $\mathscr{B}_j\leftarrow\{\mathscr{N}_i:\bar{B}_{r,j}=\star$, and $x_r\in\mathscr{N}_i\}$, for $j=1,\dots,m$
4: Construct bipartite graph $G_B=((V,\widetilde{V}),E)$, where $V=\{\mathcal{S}_1,\mathcal{S}_2,\dots,\mathcal{S}_m\}$ and $\widetilde{V}=\{z_1,z_2,\dots,z_{2m}\}$ and $E=V\times\widetilde{V}$
5: Find a perfect matching in G_B, say M
6: Inputs selected under M, $\mathcal{I}(M)\leftarrow\{j:(\mathcal{S}_j,z_i)\in M,i>m\}$
7: Define $\widetilde{\mathcal{S}}(M)\leftarrow\bigcup_{j\in\mathcal{I}(M)}\mathscr{B}_j$
8: **if** $\mathscr{N}\subseteq\widetilde{\mathcal{S}}(M)$ and $\mathcal{B}(\bar{A},\bar{B}_{\mathcal{I}(M)})$ has a perfect matching **then**
9: $\quad\mathcal{U}(M)=\sum_{j\in\{1,2,\dots,m\}}c_u(j)-\sum_{i\in\mathcal{I}(M)}c_u(i)$
10: **else**
11: $\quad\mathcal{U}(M)=-\kappa$
12: **end if**

in time polynomial in the input size. However, since cardinality of the set of perfect matchings (Ω) is huge, finding one with maximum utility using some exhaustive search based technique is not computationally efficient. Further, $\mathcal{U}(\cdot)$ is a global function (since it depends on the matching and not on individual edges) and hence it cannot be captured as edge weights, which excludes the option of using minimum weight matching matching algorithm to solve this. In this paper, we use a randomized algorithm. In the next subsection, we reduce the minCCIS problem to an instance of the BMUM problem.

B. Polynomial Time Reduction of the minCCIS Problem to the BMUM Problem

This section describes a reduction of the minCCIS problem to an instance of the BMUM problem in polynomial time. Further, we also show that an optimal solution to the BMUM problem obtained after the reduction procedure, when translated back gives an optimal solution to the minCCIS problem. In the BMUM problem constructed, every perfect matching corresponds to an input selection. Here, the utility must be defined in such a way that an optimum matching is one that gives a minimum cost input selection over all possible input selections that make $\bar{A}$ controllable with the selected inputs. The pseudo-code for reducing the minCCIS problem to an instance of the BMUM problem is given in Algorithm 4.1.

Given an instance of the minCCIS problem, we reduce it to a BMUM problem using Algorithm 4.1. The steps in Algorithm 4.1 are elaborated here. Consider a structured system $(\bar{A},\bar{B})$ and a cost vector c_u. Let $\mathscr{N}=\{\mathscr{N}_1,\dots,\mathscr{N}_q\}$ denote the set of the non-top linked SCCs in $\mathcal{D}(\bar{A})$. Thus for achieving accessibility of $\mathcal{D}(\bar{A},\bar{B})$, all these SCCs must be input accessible. Now we define a set consisting of elements $\{\mathscr{B}_j\}_{j=1}^m$. Element $\mathscr{B}_j$ corresponds to the j^{th} input and $\mathscr{B}_j$ consists of all those non-top linked SCCs in $\mathcal{D}(\bar{A})$ that have a directed path from input u_j. More precisely, those non-top linked SCCs which have at least one node in it reachable from u_j belong to the set $\mathscr{B}_j$ (Step 3). Next we construct the bipartite graph G_B. The vertex set of G_B is $(V\cup\widetilde{V})$, where $V=\{\mathcal{S}_1,\dots,\mathcal{S}_m\}$ and $\widetilde{V}=\{z_1,\dots,z_{2m}\}$. Here vertex set V is a representative of inputs in $\bar{B}$ and vertex set $\widetilde{V}$ has no additional connection to the structured system except that its cardinality is twice the number of inputs in the system, i.e., $|\widetilde{V}|=2m$. Further, G_B is a complete bipartite graph and hence all vertices in V are connected to all vertices in $\widetilde{V}$.

Let M denote a perfect matching in G_B. Each perfect matching in G_B is an input selection and we describe here how the selection is made. Since $|V| = m$ and $|\widetilde{V}| = 2m$, $|M| = m$. Thus only m nodes in the set $\widetilde{V}$ are matched in M. Associated with M, the input set selected is defined as $\mathcal{I}(M)$ as shown in Step 6. Indices of all $\mathcal{S}_i$'s that are matched in M to nodes in the set $\{z_{m+1},\ldots,z_{2m}\}$ are included in $\mathcal{I}(M)$. In other words, matched nodes in $\widetilde{V}$ whose indices are greater that m results in the selection of some inputs. A set $\widetilde{\mathcal{S}}(M)$ is defined as shown in Step 7. Here $\widetilde{\mathcal{S}}(M)$ consists of all non-top linked SCCs that are input accessible in the input selection $\mathcal{I}(M)$. The utility of the matching M is defined as follows: if the input selection $\mathcal{I}(M)$ results in a controllable input selection, then a utility is assigned to matching M taking into consideration the costs of the inputs in $\mathcal{I}(M)$. However, if the input selection $\mathcal{I}(M)$ results in an uncontrollable input set, then a large negative value is assigned as the utility of the matching M. To assign utility, we need to check the two conditions for structural controllability corresponding to the input selection obtained. Accessibility is guaranteed if $\mathscr{N} \subseteq \widetilde{\mathcal{S}}(M)$ and no-dilation is guaranteed if $\mathcal{B}(\bar{A},\bar{B}_{\mathcal{I}(M)})$ has a perfect matching. If both these conditions are satisfied, then the utility of M, $\mathcal{U}(M) = \sum_{j\in\{1,2,\ldots,m\}} c_u(j) - \sum_{i\in\mathcal{I}(M)} c_u(i)$ (Step 9). The utility is maximum if a minimum cost input set satisfies both the accessibility and the no-dilation conditions. However, if the system is not controllable for the input selection $\mathcal{I}(M)$, then $\mathcal{U}(M) = -\kappa$, where κ is a large positive value (Step 11). The output of Algorithm 4.1 is the bipartite graph G_B and the utility function $\mathcal{U}(\cdot)$.

The theorem below proves that an optimal solution to the BMUM problem constructed for a structured system gives an optimal solution to the minCCIS problem.

Theorem 4.1. *Consider a structured system $(\bar{A},\bar{B})$ and input cost vector c_u. An optimal solution to the bipartite matching problem with global utility maximization on the bipartite graph G_B and utility function $\mathcal{U}(\cdot)$ constructed using Algorithm 4.1 for $(\bar{A},\bar{B})$ gives an optimal solution to the minCCIS problem.*

Proof. Let $M^\star$ be an optimal solution to the bipartite matching problem with utility maximization (BMUM). Then, $\mathcal{U}(M^\star) \geqslant 0$. This is because there exists a perfect matching in G_B whose utility is not $-\kappa$. For example $M = \{(\mathcal{S}_i,z_j) : i = 1,\ldots,m$ and $j = i+m\}$ satisfies $\mathcal{U}(M) = 0$. Note that, $M \in \Omega$, since G_B is a complete graph. Thus $\mathcal{U}(M^\star) \geqslant 0$. Hence from Step 8 of Algorithm 4.1, $\mathcal{I}(M^\star)$ is an input selection such that $(\bar{A},\bar{B}_{\mathcal{I}(M^\star)})$ is structurally controllable. Thus $\mathcal{I}(M^\star) \in \mathcal{K}$. Now we need to show that $\mathcal{I}(M^\star)$ is an optimal solution to the minCCIS problem. We prove this using a contradiction argument. Suppose that $M^\star$ is an optimal matching of the BMUM problem, but $\mathcal{I}(M^\star)$ is not an optimal solution to the minCCIS problem. Then there exists $\mathcal{I}' \subseteq \{1,\ldots,m\}$ such that $(\bar{A},\bar{B}_{\mathcal{I}'})$ is structurally controllable and $\sum_{i\in\mathcal{I}'} c_u(i) < \sum_{i\in\mathcal{I}(M^\star)} c_u(i)$. Thus

$$
\begin{aligned}
-\sum_{i\in\mathcal{I}'} c_u(i) &> -\sum_{i\in\mathcal{I}(M^\star)} c_u(i),\\
\sum_{j\in\{1,2,\ldots,m\}} c_u(j) - \sum_{i\in\mathcal{I}'} c_u(i) &> \sum_{j\in\{1,2,\ldots,m\}} c_u(j) - \sum_{i\in\mathcal{I}(M^\star)} c_u(i),\\
\sum_{j\in\{1,2,\ldots,m\}} c_u(j) - \sum_{i\in\mathcal{I}'} c_u(i) &> \mathcal{U}(M^\star). \qquad (2)
\end{aligned}
$$

Since G_B is a complete graph, there exists a matching $M' = \{(\mathcal{S}_i,z_j) : j \in \mathcal{I}'\}$. Let us define $\mathcal{I}(M') := \mathcal{I}'$. We know $\mathcal{U}(M^\star) \geqslant 0$. Then from equation (2), $\sum_{j\in\{1,2,\ldots,m\}} c_u(j) - \sum_{i\in\mathcal{I}(M')} c_u(i) = \mathcal{U}(M') > 0$. Hence, M' is a perfect matching in G_B and $\mathcal{U}(M') > \mathcal{U}(M^\star)$. This contradicts the assumption that $M^\star$ is an optimum matching in G_B with respect to the utility function $\mathcal{U}(\cdot)$. This completes the proof. □

Thus using an optimal solution to the BMUM problem one can obtain an optimal solution to the minCCIS problem. The complexity of Algorithm 4.1 is given below.

Theorem 4.2. *Consider a structured system $(\bar{A},\bar{B})$ and an input cost vector c_u. Then, Algorithm 4.1 reduces the minCCIS problem to an instance of the BMUM problem in complexity $O(n^2)$, where n denotes the number of states in the system. Also, finding utility for a matching in this bipartite graph involves $O(n^{2.5})$ computations. Further, an optimal solution to the minCCIS problem can be obtained from an optimal solution to the BMUM problem in $O(n)$ computations.*

Proof. Given $\bar{A}$, finding the SCCs of the digraph $\mathcal{D}(\bar{A})$ has complexity $O(|V_X|+|E_X|)$, where $|V_X|$ and $|E_X|$ denote the number of nodes and edges in $\mathcal{D}(\bar{A})$ respectively. Here, $|V_X| = n$, $|E_X| = O(n^2)$ and $m \leqslant n$. Given the set of SCCs in $\mathcal{D}(\bar{A})$, the set of non-top linked SCCs, $\mathscr{N}$, can be found in $O(n)$ computations. Given the set of non-top linked SCCs, the computation required for finding each of the $\mathscr{B}_j$'s in Step 3 is $O(n)$. Thus, constructing the bipartite graph $G_B = ((V,\widetilde{V}),E)$ and the set $\{\mathscr{B}_i\}_{i=1}^m$ has $O(n^2)$ computations. Now, we calculate the complexity involved in calculating utility. Finding a perfect matching involves $O(n^{2.5})$ operations. Given a matching M, finding sets $\mathcal{I}(M)$ and $\widetilde{\mathcal{S}}(M)$ has complexity $O(m)$. For a feasible M, $\mathcal{U}(M)$ can be calculated in $O(m)$ operations. Thus, utility calculation has complexity $O(n^{2.5})$.

Let $M^\star = \{(\mathcal{S}_i,z_j) : i \in \{1,\ldots,m\}$ and $j \in \{1,\ldots 2m\}\}$ be an optimal solution to the BMUM problem. Then complexity of finding an optimal solution, $\mathcal{I}(M^\star) = \{i : (\mathcal{S}_i,z_j) \in M^\star, j > m\}$, to the minCCIS problem is $O(m)$. Thus an optimal solution to the minCCIS problem can be obtained in $O(n)$ computations from an optimal solution of the BMUM problem. □

From the above results, given an efficient scheme to solve the BMUM problem, one can solve the minCCIS problem efficiently. However, the BMUM problem is an NP-hard problem [13, Theorem 2.1]. In the section below, we propose a randomized algorithm based on MCMC technique to solve the minCCIS problem using its BMUM formulation.

5. Randomized Algorithm for solving the minCCIS Problem

This section discusses a randomized algorithm based on MCMC for solving the minCCIS problem using the BMUM problem formulation of the bipartite graph G_B constructed using Algorithm 4.1. Given a problem instance of minCCIS, the BMUM formulation is obtained first using Algorithm 4.1. The input to the randomized algorithm is the bipartite graph G_B and the utility function $\mathcal{U}$ obtained as output of Algorithm 4.1.

The pseudo-code for the randomized algorithm is presented in Algorithm 5.1. Here Ω is the set of all perfect matchings in G_B. We start with a random perfect matching from Ω, say M_0. We run the algorithm for T steps. In step t, we randomly choose a perfect matching $M_t \in \Omega$ where the choice distribution depends on M_{t-1}. Select $\mathcal{S}_i \in V$ and $z_j \in \widetilde{V}$ uniformly at random. Note that since $m < 2m$, in any perfect matching node $\mathcal{S}_i$ is always matched to some node in $\widetilde{V}$. As for z_j, there are three possibilities: (a) z_j is matched to $\mathcal{S}_i$, i.e. the edge $(\mathcal{S}_i,z_j)$ is already a part of the matching M_{t-1}. (b) z_j is not matched, and $(\mathcal{S}_i,z_k)$ is a part of the matching M_{t-1} for some $z_k \in \widetilde{V}$. (c) z_j is matched to $\mathcal{S}_r$, i.e. $(\mathcal{S}_i,z_k)$ and $(\mathcal{S}_r,z_j)$ both belong to M_{t-1}. In each of the possible cases, we do the following: In case (a), we do nothing and retain the same matching as before. Thus, $M_t = M_{t-1}$ in this case. In case (b), we construct a perfect matching $\tilde{M}_t$ by removing edge $(\mathcal{S}_i,z_k)$ and then adding the edge $(\mathcal{S}_i,z_j)$ to M_{t-1}. Similarly in case (c), we construct $\tilde{M}_t$ by removing

Algorithm 5.1 Pseudo-code for the proposed algorithm

Input: Bipartite graph $G_B = ((V \cup \widetilde{V}, E)$, where $|V| = m, |\widetilde{V}| = 2m$
Output: Matching M_T
1: Initialize $V = \{\mathcal{S}_1, \cdots, \mathcal{S}_m\}$ and $\widetilde{V} = \{z_1, \cdots, z_{2m}\}$, step $t = 0$
2: Start with a random complete matching, say M_0
3: **while** $t \leqslant T$ **do**
4: Select nodes $\mathcal{S}_i \in V$ and $z_j \in \widetilde{V}$ uniformly at random
5: **if** $(\mathcal{S}_i, z_j) \in M_t$ **then**
6: $\tilde{M}_{t+1} \leftarrow M_t$
7: **else if** $(\mathcal{S}_i, z_k) \in M_t$ and z_j is unmatched **then**
8: $\tilde{M}_{t+1} \leftarrow (M_t \cup \{(\mathcal{S}_i, z_j)\}) \setminus \{(\mathcal{S}_i, z_k)\}$
9: **else if** $(\mathcal{S}_i, z_k) \in M_t$ and $(\mathcal{S}_r, z_j) \in M_t$ **then**
10: $\tilde{M}_{t+1} \leftarrow (M_t \cup \{(\mathcal{S}_i, z_j), (\mathcal{S}_r, z_k)\}) \setminus \{(\mathcal{S}_i, z_k), (\mathcal{S}_r, z_j)\}$
11: **end if**
12: Calculate $\mathcal{U}(\tilde{M}_{t+1})$
13: $prob = \min\{1, e^{\beta(\mathcal{U}(\tilde{M}_{t+1}) - \mathcal{U}(M_t))}\}$
14: $M_{t+1} \leftarrow \tilde{M}_{t+1}$ w.p. $prob$ and $M_{t+1} \leftarrow M_t$ otherwise
15: $t \leftarrow t+1$
16: **end while**

edges $(\mathcal{S}_i, z_k)$ and $(\mathcal{S}_r, z_j)$, and adding edges $(\mathcal{S}_i, z_j)$ and $(\mathcal{S}_r, z_k)$ to M_{t-1}. Note that these operations are possible since G_B is a complete graph. Once the matching $\tilde{M}_t$ is constructed, we compute $\mathcal{U}(\tilde{M}_t)$. If $\mathcal{U}(\tilde{M}_t) \geqslant \mathcal{U}(M_{t-1})$, then we let $M_t = \tilde{M}_t$; otherwise we only accept $\tilde{M}_t$ as a new choice with probability $\exp\{\beta(\mathcal{U}(\tilde{M}_t) - \mathcal{U}(M_{t-1}))\}$, where β is a constant. Note that in each iteration of the proposed algorithm, we randomly pick a matching from the neighborhood of the current one, and propose to use it. If the chosen matching has equal or more utility than that of the current one, then we accept it as a new one, else we accept it only probabilistically with probability depending on the utility of the proposed and the current matching. Specifically, closer the utility of the proposed matching to that of the existing one, higher is the probability of accepting the proposed matching.

Fix parameter $\beta < \infty$. Let Z_t^{β} be a random variable that denotes the matching used by the proposed algorithm in t^{th} iteration for the given β. Consider the discrete time stochastic process $\mathcal{M}(\beta) = \{Z_t^{\beta}\}_{t \geqslant 0}$. For the Markov chain $\mathcal{M}(\beta)$, with state space Ω, constructed above the following result holds.

Lemma 5.1. *Consider a structured system $(\bar{A}, \bar{B})$ and input cost vector c_u. Let $\mathcal{M}(\beta)$ denote the Markov chain corresponding to the bipartite graph G_B constructed using Algorithm 4.1. Then, for $\beta < \infty$, $\mathcal{M}(\beta)$ is a discrete time Markov chain (DTMC). Further, $\mathcal{M}(\beta)$ is aperiodic and irreducible.*

Proof. Consider the discrete time stochastic process $\{Z_t^{\beta}\}_{t \geqslant 0}$, where Z_t^{β} is a random variable that denotes the matching used by Algorithm 5.1 in t^{th} iteration. The sequence $Z_0^{\beta}, Z_1^{\beta}, \cdots$ is a random sequence as the chain $\mathcal{M}(\beta)$ progress through the state space Ω. From the transition definition given in Algorithm 5.1 (Steps 4-11) it is clear that the state at the $(t+1)^{\text{th}}$ instant, Z_{t+1}^{β}, depends only on the previous state Z_t^{β}. That is, $P\ (Z_{t+1}^{\beta} | Z_0^{\beta}, Z_1^{\beta}, \cdots, Z_t^{\beta}) = P\ (Z_{t+1}^{\beta} | Z_t^{\beta})$. Thus $\mathcal{M}(\beta)$ is a Discrete Time Markov Chain (DTMC) on Ω.

In order to prove $\mathcal{M}(\beta)$ is irreducible, we need to show that any arbitrary state F in the chain can be reached from any arbitrary state I in the chain. The chain is constructed in such a way that between any two states in it there exists a positive probability for state transition (Steps 13-14 of Algorithm 5.1). Thus it is irreducible. The aperiodic proof is also straight forward. Since all states in the chain have a positive probability of remaining in the same state in the next step, the chain is aperiodic. Thus the chain $\mathcal{M}(\beta)$ is a DTMC which is irreducible and aperiodic. □

For a given β, let $P(\beta) = [P_{M'M}(\beta)]_{M, M' \in \Omega}$ denote the transition probability matrix on the DTMC $\mathcal{M}(\beta)$. Since the state space of the MCMC is finite, i.e., $|\Omega|$ is finite, Lemma 5.1 concludes that $\mathcal{M}(\beta)$ is a positive recurrent chain and hence it admits the steady state distribution, say $\boldsymbol{\pi}(\beta) = [\pi_M(\beta)]_{M \in \Omega}$. The DTMC $\mathcal{M}(\beta)$ is said to be reversible if for any two adjacent states M_1 and M_2 in Ω, $\pi_{M_1}(\beta) P_{M_1 M_2}(\beta) = \pi_{M_2}(\beta) P_{M_2 M_1}(\beta)$. Now, we characterize the steady state distribution.

Lemma 5.2. *Fix any $\beta < \infty$. The DTMC $\mathcal{M}(\beta)$ is time reversible and for every $M \in \Omega$, $\pi_M(\beta) = \frac{\exp\{\beta \mathcal{U}(M)\}}{\sum_{M' \in \Omega} \exp\{\beta \mathcal{U}(M')\}}$.*

Proof. Every entry $P_{ij}(\beta)$ of the probability transition matrix $P(\beta)$ is given by $P_{ij}(\beta) = \frac{1}{m \times 2m} a_{ij}(\beta)$, where m and $2m$ are the number of nodes in the respective sets of the bipartite graph and $a_{ij}(\beta)$ is the acceptance probability from state i to state j respectively.
Using $a_{ij}(\beta) = \min\ \{1, \exp\{\beta(\mathcal{U}(j) - \mathcal{U}(i))\}\}$, we get

$$a_{M_1 M_2}(\beta) = \begin{cases} 1 & \text{if } \mathcal{U}(M_2) \geqslant \mathcal{U}(M_1), \\ \exp\{\beta(\mathcal{U}(M_2) - \mathcal{U}(M_1))\} & \text{if } \mathcal{U}(M_2) < \mathcal{U}(M_1). \end{cases}$$

Now consider two cases: (a) $\mathcal{U}(M_2) \geqslant \mathcal{U}(M_1)$ and (b) $\mathcal{U}(M_2) < \mathcal{U}(M_1)$. When $\mathcal{U}(M_2) \geqslant \mathcal{U}(M_1)$,

$$\begin{aligned} \pi_{M_1}(\beta) P_{M_1 M_2}(\beta) &= \frac{\exp\{\beta \mathcal{U}(M_1)\}}{\sum_{M' \in \Omega} \exp\{\beta \mathcal{U}(M')\}} \times \frac{1}{2m^2} \times 1, \\ \pi_{M_2}(\beta) P_{M_2 M_1}(\beta) &= \frac{\exp\{\beta \mathcal{U}(M_2)\}}{\sum_{M' \in \Omega} \exp\{\beta \mathcal{U}(M')\}} \times \frac{1}{2m^2} \times \\ & \quad \frac{\exp\{\beta \mathcal{U}(M_1)\}}{\exp\{\beta \mathcal{U}(M_2)\}}, \\ &= \pi_{M_1}(\beta) P_{M_1 M_2}(\beta). \end{aligned}$$

When $\mathcal{U}(M_2) < \mathcal{U}(M_1)$,

$$\begin{aligned} \pi_{M_1}(\beta) P_{M_1 M_2}(\beta) &= \frac{\exp\{\beta \mathcal{U}(M_1)\}}{\sum_{M' \in \Omega} \exp\{\beta \mathcal{U}(M')\}} \times \frac{1}{2m^2} \times \\ & \quad \frac{\exp\{\beta \mathcal{U}(M_2)\}}{\exp\{\beta \mathcal{U}(M_1)\}}, \\ \pi_{M_2}(\beta) P_{M_2 M_1}(\beta) &= \frac{\exp\{\beta \mathcal{U}(M_2)\}}{\sum_{M' \in \Omega} \exp\{\beta U(M')\}} \times \frac{1}{2m^2} \times 1 \\ &= \pi_{M_1}(\beta) P_{M_1 M_2}(\beta). \end{aligned}$$

Thus DTMC $\mathcal{M}(\beta)$ is time reversible and this completes the proof of Lemma 5.2. □

Lemmas 5.1 and 5.2 thus concludes that the Markov chain $\mathcal{M}(\beta)$ has a unique stationary distribution $\boldsymbol{\pi}(\beta) = [\pi_M(\beta)]_{M \in \Omega}$, where $\pi_M(\beta) = \frac{\exp\{\beta \mathcal{U}(M)\}}{\sum_{M' \in \Omega} \exp\{\beta \mathcal{U}(M')\}}$. Thus the steady state distribution is the desired one as it concentrates on points $M \in \Omega$ such that $\mathcal{U}(M) = \mathcal{U}(M^{\star})$ as $\beta \to \infty$. Further, since $P_{M'M}^{(t)}(\beta) \to \pi_M(\beta)$ as $t \to \infty$ for every M' and $M \in \Omega$, it should be enough to run the algorithm for some "large enough" steps T to be able to "closely" sample from the distribution $\boldsymbol{\pi}(\beta)$. However, determining T for the required sampling accuracy is challenging. To quantify this time, we use the following result from [13].

Proposition 5.3. *[13, Theorem 5.2] The mixing time of the Markov chain $\mathcal{M}$ that has state space Ω as all possible perfect matchings of a complete bipartite graph $G = ((V_1 \cup V_2), L)$ with $|V_1| = n_1, |V_2| = n_2$ and $n_1 \leqslant n_2$ is bounded by $\tau_{\varepsilon} \leqslant 32 {n_1}^2 {n_2}^2 \alpha^6 (-c + n_1 \log(n_2) + \log(\varepsilon^{-1}))$, where $\alpha = \exp\{\beta(\mathcal{U}_{\max} - \mathcal{U}_{\min})$ and $c = \log \alpha$, where $\mathcal{U}_{\max}$ and $\mathcal{U}_{\min}$ are the maximum and the minimum values of utility.*

Using Proposition 5.3 one can obtain an optimum matching in time polynomial in the input size with high probability. In other words, an optimal solution can be obtained efficiently with very high probability. Thus the following result holds.

Theorem 5.4. *Consider a structured system* $(\bar{A}, \bar{B})$ *and an input cost vector* c_u*. Then, using the BMUM formulation of the minCCIS problem, an optimum solution to the minCCIS problem can be obtained with high probability in* $O(n^{2.5}) + Poly(n, \log\ n)$*, where* $Poly(n, \log\ n)$ *is a polynomial in n, log n.*

Proof. Given a structured system there exists a polynomial time reduction of complexity $O(n^{2.5})$ that reduces the minCCIS problem to an instance of the BMUM problem (Theorem 4.2). Let G_B denotes the bipartite graph constructed and let $\mathcal{U}$ be the utility function obtained using Algorithm 4.1. Let $M^\star$ be an optimum matching of the BMUM problem on G_B and $\mathcal{U}$. By Theorem 4.1 an optimum solution to the BMUM problem gives an optimum solution to the minCCIS problem. In addition, by Proposition 5.3, the Markov chain $\mathcal{M}(\beta)$ constructed for finding an optimum matching in the bipartite graph G_B is rapid mixing. Thus the complexity involved in solving the BMUM problem is $Poly(n, \log\ n)$, where $Poly(n, \log\ n)$ is a polynomial in n, log n. Thus, an optimum solution to the minCCIS problem can be obtained with high probability in time $O(n^{2.5}) + Poly(n, \log\ n)$. □

Remark 5.5. *By duality between controllability and observability in LTI systems, all the results in this paper are directly applicable to the minimum cost constrained output selection problem, which consists of selecting a minimum cost output set from a given structured output matrix* $\bar{C}$ *and a output cost vector* c_y *that ensures structural observability of a given* $\bar{A}$*.*

6. Simulation Results

We conducted simulations for finding a minimum cost set of inputs required to make a structured system controllable using the proposed MCMC based method. The input to the experiment is a structured system $(\bar{A}, \bar{B})$, where $\bar{A} \in \{0, \star\}^{500\times 500}$ and $\bar{B} \in \{0, \star\}^{500\times 50}$. The cost vector considered has uniform and non-zero, i.e., $c_u = [1, \ldots, 1]^T$. Thus G_B is a complete bipartite graph with 50 nodes on one side and 100 nodes on the other side. The MCMC is run from 100 different initial conditions and the utility is averaged over 100.

The utility of a perfect matching M is $\mathcal{U}(M) = \sum_{j\in\{1,2,\ldots,m\}} c_u(j) - \sum_{i\in\mathcal{I}(M)} c_u(i)$, where m is the number of inputs (which is 50 here) and $\mathcal{I}(M) = \{j : (\mathcal{S}_j, y_i) \in M, i > m\}$ is the set of selected inputs under matching M. If the matching is not feasible we give utility as -100. The minimum and maximum utilities associated with a feasible matching are 0 and 49 respectively. $\mathcal{U}(M) = 0$, corresponds to a matching where all inputs are selected and $\mathcal{U}(M) = 49$ corresponds to the matching M where exactly one input is selected to make the system controllable. $\mathcal{U}(M) = 50$, is not possible, since atleast one input should be selected. Figure 3 shows the convergence of the algorithm to the optimum utility with respect to the number of iterations. This plot is obtained by running MCMC from 100 different initial conditions and averaging it.

The plot shows that the proposed scheme converges to optimum solution it a very less time and hence is a promising scheme for solving the minCCIS problem in large-scale systems with large system dimensions.

7. Conclusion

This paper deals minimum cost constrained input selection (minCCIS) for structural controllability of large complex systems. Constraints are imposed on the input structure and each input is associated with a cost. The objective is to find a minimum cost input selection

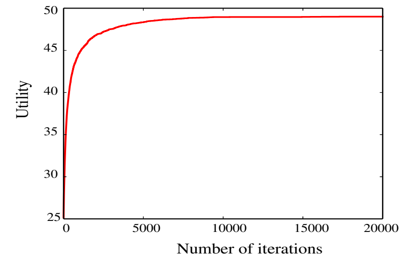


Figure 3: Plot showing convergence of MCMC for minCIS.

that achieves structural controllability. Since the problem is NP-hard, we reduced it to a bipartite graph matching problem with a global utility maximization (BMUM) (Algorithm 4.1) in polynomial time complexity. We also proved that an optimum solution of the BMUM problem gives an optimum solution to the minCCIS problem (Theorem 4.1). Since the BMUM problem is NP-hard, we proposed a randomized algorithm based on MCMC technique to solve the BMUM formulation of the minCCIS problem. The constructed discrete time Markov chain is shown to be irreducible, aperiodic and accommodates a unique stationary distribution that maximizes the utility function. It is shown in [13] that the chain constructed here is rapid mixing. Hence an optimal solution to the minCCIS problem can be obtained in polynomial time with high probability. In other words, using the randomized MCMC based scheme proposed here a solution to the minCIS and the minCCIS problems can be obtained in time polynomial in n, where n is the number of states in the structured system.

References

[1] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Reviews of Modern Physics*, vol. 88, no. 3, pp. 035 006:1–58, 2016.

[2] C. Commault and J.-M. Dion, "Input addition and leader selection for the controllability of graph-based systems," *Automatica*, vol. 49, no. 11, pp. 3322–3328, 2013.

[3] A. Chapman and M. Mesbahi, "On strong structural controllability of networked systems: A constrained matching approach," in *Proceedings of IEEE American Control Conference (ACC)*, Washington, DC, USA, 2013, pp. 6126–6131.

[4] A. Olshevsky, "Minimum input selection for structural controllability," in *Proceedings of IEEE American Control Conference (ACC)*, Chicago, IL, USA, 2015, pp. 2218–2223.

[5] C. Commault and J.-M. Dion, "The single-input minimal controllability problem for structured systems," *Systems & Control Letters*, vol. 80, pp. 50–55, 2015.

[6] S. Pequito, S. Kar, and G. J. Pappas, "Minimum cost constrained input-output and control configuration co-design problem: A structural systems approach," in *Proceedings of IEEE American Control Conference (ACC), 2015*, Chicago, IL, USA, 2015, pp. 4099–4105.

[7] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press: New Jersey, 2010.

[8] S. Pequito, S. Kar, and A. P. Aguiar, "On the complexity of the constrained input selection problem for structural linear systems," *Automatica*, vol. 62, pp. 193–199, 2015.

[9] S. Moothedath, P. Chaporkar, and M. N. Belur, "A Flow-Network-Based Polynomial-Time Approximation Algorithm for the Minimum Constrained Input Structural Controllability Problem," *IEEE Transactions on Automatic Control*, no. 99, 2018.

[10] K. Murota, *Systems Analysis by Graphs and Matroids*. Springer-Verlag Berlin Heidelberg: New York, 1987, vol. 3.

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

[12] R. Diestel, *Graph Theory*. Springer: New York, 2000.

[13] S. Moothedath, P. Chaporkar, and M. N. Belur, "Rapidly Mixing Markov Chain Monte Carlo Technique for Matching Problems with Global Utility Function," *ArXiv e-prints*, Oct. 2017. [Online]. Available: http://adsabs.harvard.edu/abs/2017arXiv171010037M