Complexity of the Minimum Input Selection Problem for Structural Controllability ⋆

Sepehr Assadi * **Sanjeev Khanna** * **Yang Li** *
Victor M. Preciado **

* *Department of Computer and Information Science, University of Pennsylvania, PA 19104 USA, (e-mail: {sassadi,sanjeev,yangli2}@cis.upenn.edu).*
** *Department of Electrical and Systems Engineering, University of Pennsylvania, PA 19104 USA, (e-mail: preciado@seas.upenn.edu).*

Abstract: We consider the minimum input selection problem for structural controllability (MISSC), stated as follows: Given a linear system $\dot{x} = Ax$, where A is a $n \times n$ state matrix with m nonzero entries, find the minimum number of states that need to be driven by an external input so that the resulting system is structurally controllable. The fastest algorithm to solve this problem was recently proposed by Olshevsky in (Olshevsky, 2015) and runs in $\Theta\left(m\sqrt{n}\right)$ operations. In this paper, we propose an alternative algorithm to solve MISSC in $\min\{O\left(m\sqrt{n}\right), \tilde{O}\left(n^{2.37}\right), \tilde{O}\left(m^{10/7}\right)\}$ operations. This running time is obtained by (i) proving that MISSC problem is computationally equivalent to the maximum bipartite matching (MBM) problem and (ii) considering the three fastest algorithms currently available to solve MBM, namely, the Hopcraft-Karp algorithm, the Mucha-Sankowski algorithm, and Madry's algorithm. Furthermore, our algorithm can directly benefit from future improvements in MBM computation. Conversely, we also show that any algorithmic improvement on solving MISSC would result in an improvement in MBM computation, which would be of great interest for theoretical computer scientists.

Keywords: Linear systems; Structured systems; Graph theory; Structural controllability; Computational complexity.

1. INTRODUCTION

During the last decade, the study of complex networks has diffused throughout all branches of science and engineering. Most research in this field has been focused on structural questions: How do we characterize the connectivity structure of the Internet, the power grid, or the human brain? In recent years, this question has been extended to encompass dynamic and control issues: How can we steer complex networks of dynamic agents towards a desired state? From an engineering perspective, the ultimate frontier in our understanding of complex networks is reflected in our ability to control them.

Although control theory offers sophisticated mathematical tools to steer and control dynamic systems, a framework adapted to massive networked systems is still lacking. In this direction, a recent collection of papers are concerned with various minimum controllability problems. This recent wave of results was initiated by the work in (Liu, Slotine & Barabási, 2011), where the authors considered the problem of finding the smallest number of independent input signals needed to control a structured linear dynamic systems. Liu et al. proposed an algorithm based on the *maximum bipartite matching* (MBM) computation to solve the problem in $O\left(m\sqrt{n}\right)$ operations, where n is the number of states and m is the number of nonzero entries in the state matrix [1]. Following the work of (Liu, Slotine & Barabási, 2011), Commault and Dion have recently studied alternative formulations of the minimum controllability problem (Commault & Dion, 2013a,b, 2014), such as the problem of controlling the system using a single input signal. Pequito, Kar, and Aguiar have recently proposed an interesting framework to solve the minimum input/output control configuration problem for structured systems (Pequito, Kar & Aguiar, 2013, 2015).

In this paper, we focus our attention on the *minimum input selection problem for structural controllability* (MISSC). This problem can be stated as follows: find the minimum number of states that need to be driven by an external input so that the resulting system is *structurally controllable.* Pequito et al. proposed in (Pequito, Kar & Aguiar, 2013) the first algorithm to solve MISSC in $O\left(mn^{1.5}\right)$operations. An alternative approach with a

⋆ This work was supported by the United States National Science Foundation under grants CNS-1302222, IIS-1447470, and CCF-1116961.

[1] We make the standard assumption that $m \geq n$. In the cases where $m < n$, there might be an extra $O(n)$ additive factor in the running times to account for the time required to read the input.

running time of $O\left(n^3\right)$ was recently proposed in (Pequito, Kar & Aguiar, 2015). Our work is closely related to the work in (Olshevsky, 2015), where Olshevsky developed the fastest algorithm (up-to-date) to solve the MISSC problem. Olshevsky's algorithm solves the MISSC problem in two stages. In the first stage, the algorithm solves a maximum matching problem in a bipartite graph that represents the sparsity pattern of a given linear system. Using the well-known Hopcraft-Karp algorithm (Hopcroft & Karp, 1973), the running time of this stage is $\Theta\left(m\sqrt{n}\right)$. In the second stage of Olshevsky's algorithm, an augmentation process was proposed to turn the matching found in the previous stage into a solution to the MISSC problem. The second stage requires additional $\Theta\left(m\sqrt{n}\right)$ operations.

The following observation is key to motivate our work: if it were somehow possible to solve the maximum bipartite matching problem faster than $\Theta\left(m\sqrt{n}\right)$, Olshevsky's algorithm would still run in $\Theta\left(m\sqrt{n}\right)$ time. This is due to the bottleneck induced by the augmentation process in the second stage. In fact, several breakthroughs in the maximum bipartite matching computation currently allow to solve this problem in $\min\{O\left(m\sqrt{n}\right), \tilde{O}\left(n^{\omega}\right), \tilde{O}\left(m^{10/7}\right)\}$ time, where ω is the matrix multiplication factor [2]. This running time is obtained if one considers the three fastest algorithms currently available to solve the maximum bipartite matching problem, namely, (i) the Hopcraft-Karp algorithm (which runs in $O\left(m\sqrt{n}\right)$), (ii) the Mucha-Sankowski algorithm (Mucha & Sankowski, 2004) (which runs in $\tilde{O}(n^{\omega})$), and (iii) Madry's algorithm, recently proposed in (Madry, 2013) (which runs in $\tilde{O}(m^{10/7})$). It is worth mentioning that while Hopcraft-Karp algorithm is deterministic, Mucha-Sankowski and Madry's algorithms are randomized and outputs the correct answer with high probability, i.e., probability at least $1 - 1/n$. Due to the augmentation process in the second stage of Olshevsky's algorithm, its resulting running time cannot benefit from these (and possible future) breakthroughs in the MBM computation.

In this paper, we propose a new approach for solving MISCC, and show that the asymptotic complexity of solving MISCC problem and MBM problem is the same. In other words, we prove that the running time required to solve MISSC is essentially equal to the running time of the fastest algorithm available to solve MBM (up to constant factors). As a consequence, any improvement in the running time required to solve the MISSC problem would result in a breakthrough for the MBM problem (and vice versa).

The paper is structured as follows. In Section 2, we introduce our notation, present some preliminary results, and rigorously state the MISSC problem. In Section 3, we first introduce a common graph-theoretic reformulation of the MISSC problem and briefly describe Olshevsky's algorithm. We then introduce an alternative formulation that allows us to solve the MISSC problem using the fastest available algorithm to solve the MBM problem. We conclude the paper in Section 4.

[2] The lowest value of this factor is currently $\omega \approx 2.37$ (Williams, 2012; Le Gall, 2014).

2. PRELIMINARIES AND PROBLEM STATEMENT

We first introduce our notation. For any $n \in \mathbb{N}$, we define $[n] = \{1, \ldots, n\}$. Vectors are represented with bold and matrices with capital letters. I denotes the $n \times n$ identity matrix. Given a matrix M, $\|M\|_0$ is the number of nonzero entries of M. Given a matrix B and a set $J \subseteq [n]$, B_J denotes the matrix formed by the columns of B indexed by J. Throughout the paper, we use big-O asymptotic notation to quantify the computational complexity of algorithms (see (Knuth, 1997), Section 1.2.11, for a thorough treatment). In our exposition, we use $\tilde{O}$ notation to suppress polylog factors, meaning that $\tilde{O}(f) = O(f) \cdot \text{polylog}(f)$.

In the following subsections, we briefly introduce the concept of structural controllability and state the problem under consideration.

2.1 Structural Controllability

We study linear time-invariant (LTI) systems of the form:

$$\dot{\mathbf{x}}\left(t\right) = A\mathbf{x}\left(t\right) + B\mathbf{u}\left(t\right), \tag{1}$$

where $\mathbf{x}\left(t\right) \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$ are the vector of states and the state matrix, respectively; similarly, $\mathbf{u}\left(t\right) \in \mathbb{R}^p$ and $B \in \mathbb{R}^{n\times p}$ are the vector of p input signals and the input matrix, respectively. In short, we denote the system in (1) by (A, B). Notice that, the number of nonzero columns of B equals the maximum number of independent inputs signals that can be used to control the system, while the number of nonzero rows of B equals the number of states influenced by these input signals.

The system (A, B) is said to be *controllable* if, given a time duration $T > 0$ and two arbitrary states $\mathbf{x}_0, \mathbf{x}_T \in \mathbb{R}^n$, there exists a piecewise continuous input function $t \mapsto \overline{\mathbf{u}}(t)$ from $[0, T]$ to $\mathbb{R}^p$, such that the integral curve $\overline{\mathbf{x}}(t)$ generated by $\overline{\mathbf{u}}$ with $\overline{\mathbf{x}}(0) = \mathbf{x}_0$, satisfies $\overline{\mathbf{x}}(T) = \mathbf{x}_T$. A classical result by Kalman states that the system (A, B) is controllable if and only if its controllability matrix $\mathcal{C} = \left[B, AB, A^2B, \ldots, A^{n-1}B\right]$ has full row rank. In many practical problems, it is not possible to exactly retrieve the entries of the state and input matrices, A and B, in which case it is not possible to verify Kalman's controllability condition. On the other hand, it is much easier to infer the sparsity pattern of these matrices from the structure of the system. The theory of structured linear systems study control-theoretic properties of linear systems when only the sparsity patterns of the system matrices are known. A specially relevant property is *structural controllability*, first introduced in (Lin, 1974), extended in (Shields & Pearson, 1976), and briefly described below.

Given an $n_1 \times n_2$ matrix M, the *sparsity pattern* of M is defined as the set $\mathcal{S}\left(M\right) = \{(i, j) \in [n_1] \times [n_2] \colon M_{ij} = 0\}$, i.e., the set of matrix entries that are known to be zero. Two matrices M_1 and M_2 of identical dimensions are said to have the same sparsity pattern if $\mathcal{S}\left(M_1\right) = \mathcal{S}\left(M_2\right)$. Given two Boolean matrices $\overline{A} \in \{0, 1\}^{n\times n}$ and $\overline{B} \in \{0, 1\}^{n\times p}$, we define the *structured system* $\langle\overline{A}, \overline{B}\rangle$ as the following set of LTI systems:

$$\langle\overline{A}, \overline{B}\rangle = \big\{(A, B) \colon \mathcal{S}(A) = \mathcal{S}(\overline{A}), \mathcal{S}(B) = \mathcal{S}(\overline{B})\ , \\ \text{for A} \in \mathbb{R}^{n\times n} \text{ and } B \in \mathbb{R}^{n\times p}\big\}\,.$$

The structured system $\langle\overline{A},\overline{B}\rangle$ is said to be *structurally controllable* if there exists a system $(A,B)\in\langle\overline{A},\overline{B}\rangle$ that is controllable. It turns out that if there exists a controllable system in $\langle\overline{A},\overline{B}\rangle$, then almost all systems in $\langle\overline{A},\overline{B}\rangle$ are controllable, except a subset of zero Lebesgue measure (Davison & Wang, 1973).

2.2 Problem Statement

We are concerned with the problem of *minimum input selection for structural controllability* (MISSC), recently studied in (Pequito, Kar & Aguiar, 2013, 2015; Olshevsky, 2015). This problem can be stated as follows:

Problem 1. (MISSC). *Given a structured matrix $\overline{A}\in\{0,1\}^{n\times n}$, find the set $J\subseteq[n]$ that solves*

$$\underset{J\subseteq[n]}{\text{minimize}}\,|J| \tag{2}$$

$$\text{subject to } \langle\overline{A},I_J\rangle \text{ is structurally controllable.} \tag{3}$$

Several remarks are in order. Observe that, for any $j\in J$, the j-th column of I_J indicates that an input signal is actuating on the j-th state variable $x_j(t)$; hence, we call J the set of *directly controlled states*. Also, as argued in (Olshevsky, 2015), MISSC is computationally equivalent to the problem of finding the structured input matrix $\overline{B}$ having the fewest number of rows with a nonzero entry such that $\langle\overline{A},\overline{B}\rangle$ is structurally controllable. Furthermore, MISSC is also computationally equivalent to the problem of finding the structured matrix $\overline{B}$ that minimizes $\|\overline{B}\|_0$ such that $\langle\overline{A},\overline{B}\rangle$ is structurally controllable (Olshevsky, 2015).

In this paper, we propose an algorithm to solve the MISSC problem and show that the running time required to solve it is equal to the running time of the fastest possible algorithm to find a maximum bipartite matching (MBM), which is currently given by $\min\{O(m\sqrt{n}),\tilde{O}(n^{2.37}),\tilde{O}(m^{10/7})\}$ operations where $m=\|\overline{A}\|_0$. This is an improvement on the currently fastest algorithm to solve the MISSC problem, which runs in $O(m\sqrt{n})$ operations[3]. As we mentioned in Section 1, this running time corresponds to an algorithm recently proposed by Olshevsky in (Olshevsky, 2015). Olshevsky's algorithm runs in two stages. In its first stage, the algorithm solves a MBM problem. In its second stage, the algorithm uses an augmentation process to turn the maximum matching found in the previous stage into a solution of the MISSC problem. The second stage runs in $\Theta(m\sqrt{n})$ operations. Although the first stage in this algorithm could benefit from improvements in the MBM problem, the second stage induces a bottleneck that caps the running time to $O(m\sqrt{n})$ operations. In the following section, we propose an algorithm that does not rely on this augmentation process, removing the bottleneck associated to Olshevsky's algorithm. As a consequence, the computational complexity of solving the MISSC problem is equivalent to the one of solving the MBM problem.

[3] Notice that $O(m\sqrt{n}),\tilde{O}(n^{2.37})$, and $\tilde{O}(m^{10/7})$ are not comparable running times. For example, for dense graphs $n^{2.37}=o(m\sqrt{n})$, while for sparse graphs $m\sqrt{n}=o(n^{2.37})$.

3. COMPUTATIONAL COMPLEXITY OF MISSC

We start this section by introducing a set of graph-theoretical conditions for a structured system to be structurally controllable. These conditions are closely related to the problem of finding the maximum matching in a bipartite undirected graph representing the sparsity pattern of the system. We then briefly describe Olshevsky's approach to solve the MISSC problem, which is closely related to the Hopcraft-Karp algorithm for solving the maximum matching problem. We take this work as our foundation and propose an algorithm to solve the MISSC problem as fast as the fastest available algorithm to solve the MBM problem.

3.1 Graph-theoretic Reformulation of MISSC

Given two Boolean matrices $\overline{A}\in\{0,1\}^{n\times n}$ and $\overline{B}\in\{0,1\}^{n\times p}$, a structured system $\langle\overline{A},\overline{B}\rangle$ can be conveniently represented using a directed graph $\mathcal{G}_{\overline{A},\overline{B}}=(\mathcal{V},\mathcal{E})$, called the *system graph*, where $\mathcal{V}$ is a set of $n+p$ nodes and $\mathcal{E}$ is a set of $\|\overline{A}\|_0+\|\overline{B}\|_0$ edges, constructed as follows. The node set $\mathcal{V}$ is the union of two disjoint sets $\mathcal{X}=\{x_1,\ldots,x_n\}$ and $\mathcal{U}=\{u_1,\ldots,u_p\}$, called state-node and input-node sets, respectively. The set $\mathcal{E}$ consists of all the directed edges (x_i,x_j) for which $\overline{A}_{ji}=1$ and edges (u_r,x_t) for which $\overline{B}_{tr}=1$. For a given structured state matrix $\overline{A}$, it is also convenient to define the *state subgraph* $\mathcal{G}_{\overline{A}}=(\mathcal{X},\mathcal{E}_{\mathcal{X}})$ where $\mathcal{E}_{\mathcal{X}}$ is the set of all the edges (x_i,x_j) with $\overline{A}_{ji}=1$. Notice that $\mathcal{G}_{\overline{A}}$ is the subgraph of $\mathcal{G}_{\overline{A},\overline{B}}$ induced by the nodes in $\mathcal{X}$. In what follows, we define several elements needed to state a graph-theoretic condition for structural controllability.

Given a directed graph $\mathcal{G}$, a *directed path* in $\mathcal{G}$ is defined as an ordered sequence of distinct nodes $(v_{i_1},v_{i_2},\ldots,v_{i_{l+1}})$, where $v_{i_k}\in\mathcal{V}$ and $(v_{i_k},v_{i_{k+1}})\in\mathcal{E}$ for $k\in[l]$. The nodes v_{i_1} and $v_{i_{l+1}}$ are called the *root* and the *terminal* nodes of the path, respectively. Consider a system graph $\mathcal{G}_{\overline{A},\overline{B}}$. A directed path in $\mathcal{G}_{\overline{A},\overline{B}}$ is said to be $\mathcal{U}$-rooted if $v_{i_1}\in\mathcal{U}$. We say that a node $x\in\mathcal{X}$ is *$\mathcal{U}$-reachable* if there exists a $\mathcal{U}$-rooted directed path for which x is its terminal node.

Given a directed graph $\mathcal{G}$, a *matching* $\mathcal{M}$ in $\mathcal{G}$ is defined as a subset of edges $\mathcal{M}\subseteq\mathcal{E}$ such that no two edges in $\mathcal{M}$ have a common source or a common destination. We say that a node $v\in\mathcal{V}$ is unmatched with respect to the matching $\mathcal{M}$ if there is no edge in $\mathcal{M}$ which has v as its destination. A matching is said to be *perfect* if no node is unmatched. As pointed out in (Dion, Commault & van der Woude, 2003; Liu, Slotine & Barabási, 2011; Olshevsky, 2015), there is a close relationship between structural controllability of a structured system and matchings in its system graph. In this paper, we will review some recent results on matching algorithms and derive implications on the complexity of the MISSC problem.

We can map the structure of a system graph $\mathcal{G}_{\overline{A},\overline{B}}$ into an undirected bipartite graph $\mathcal{B}_{\overline{A},\overline{B}}$, called the *bipartite system graph*, as described below. The bipartite system graph contains $2n+p$ nodes partitioned into three sets $\mathcal{U}_s=\{u_1^s,\ldots,u_p^s\}$, $\mathcal{X}_s=\{x_1^s,\ldots,x_n^s\}$, and $\mathcal{X}_d=\{x_1^d,\ldots,x_n^d\}$. It also contains $\|\overline{A}\|_0+\|\overline{B}\|_0$ undirected edges consisting of all the edges $\{x_i^s, x_j^d\}$ for which $\overline{A}_{ji} = 1$ and edges $\{u_r^s, x_t^d\}$ for which $\overline{B}_{tr} = 1$. Note that we can define the *bipartite representation* of the state subgraph $\mathcal{G}_{\overline{A}}$, as the bipartite graph $\mathcal{B}_{\overline{A}}$ in the same exact way.

In the seminal papers (Lin, 1974; Shields & Pearson, 1976), we find the following graph-theoretical characterization of structural controllability:

Theorem 1. *The structured system $\langle \overline{A}, \overline{B} \rangle$ is structurally controllable if and only if the associated system graph $\mathcal{G}_{\overline{A},\overline{B}}$ satisfies the following two conditions:*

(1) All state nodes $x \in \mathcal{X}$ are $\mathcal{U}$-reachable in $\mathcal{G}_{\overline{A},\overline{B}}$.
(2) There exist a matching in $\mathcal{G}_{\overline{A},\overline{B}}$ such that no node in $\mathcal{X}$ is unmatched.

Notice that there can be more than one matching in the system graph satisfying Condition (2) in Theorem 1.

From a computational complexity point of view, it is convenient to restate the problem of finding a matching in $\mathcal{G}_{\overline{A},\overline{B}}$ into the equivalent problem of finding a matching in the bipartite graph $\mathcal{B}_{\overline{A},\overline{B}} = (\mathcal{U}_s \cup \mathcal{X}_s \cup \mathcal{X}_d, \mathcal{E}_{\text{bip}})$, where $\mathcal{E}_{\text{bip}} \subseteq (\mathcal{U}_s \cup \mathcal{X}_s) \times \mathcal{X}_d$. A matching in $\mathcal{B}_{\overline{A},\overline{B}}$ is defined as a subset of edges $\mathcal{M}_{\text{bip}} \subseteq \mathcal{E}_{\text{bip}}$ such that no two edges in $\mathcal{M}_{\text{bip}}$ share a common vertex. Notice that there is always a simple bijection between the set of directed matchings in a system graph $\mathcal{G}_{\overline{A},\overline{B}}$ and bipartite matchings in $\mathcal{B}_{\overline{A},\overline{B}}$. Hence, finding a matching in $\mathcal{G}_{\overline{A},\overline{B}}$ is computationally equivalent to finding a matching in its bipartite counterpart $\mathcal{B}_{\overline{A},\overline{B}}$. Furthermore, note that Condition (2) in Theorem 1 is satisfied if there is a matching in the bipartite system graph $\mathcal{B}_{\overline{A},\overline{B}}$ such that all nodes in $\mathcal{X}_d$ are matched.

Finally, we introduce three graph-theoretic definitions needed to describe our algorithm (see (Diester, 2005) for a more thorough explanation). Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a *strongly connected component* (SCC) is defined as the maximal[4] subset of nodes $\mathcal{S} \subseteq \mathcal{V}$ for which there exists a directed path from every node in $\mathcal{S}$ to every other node in $\mathcal{S}$. Let us consider the set of all strongly connected components contained in a directed graph, denoted by $\text{CC} = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$. Notice that the set CC is also a partition of the nodes in $\mathcal{V}$. We can then build the so-called *condensation graph* of $\mathcal{G}$, as follows. The condensation graph contains k nodes, where each node is associated to each one of the SCC's of $\mathcal{G}$. The condensation graph contains a directed edges (c_i, c_j) if and only if there exists a directed edge from a node in $\mathcal{S}_{c_i}$ pointing towards a node in $\mathcal{S}_{c_j}$. The resulting condensation graph is a *directed acyclic graph*[5]. Those strongly connected components of $\mathcal{G}$ that correspond to nodes with no incoming edges in the condensation graph are called *source strongly connected components* (SSCC).

[4] A subset of nodes is maximal with respect to a particular property if (*1*) it satisfies this property, and (*2*) including any additional node in the subset would result in a new subset not satisfying the property.
[5] A directed acyclic graph (DAG) is a directed graph with no closed directed paths, i.e, *cycles*.

3.2 Olshevsky's Algorithm

In (Olshevsky, 2015), Olshevsky proposed the fastest (up-to-date) algorithm to solve the MISSC problem by reducing it to the following combinatorial problem. Let $\mathcal{G}_{\overline{A}} = (\mathcal{X}, \mathcal{E}_{\mathcal{X}})$ be the directed state graph of a structured system with state matrix $\overline{A}$. As proved in (Olshevsky, 2015), the MISSC problem is equivalent to the problem of finding a matching in $\mathcal{G}_{\overline{A}}$ such that the number of unmatched state vertices plus the number of source strongly connected components without an unmatched vertex is minimized. Olshevsky proposed an algorithm to solve this reformulation in two stages. In the first stage, the algorithm finds an initial maximum matching in $\mathcal{G}_{\overline{A}}$ (without any restriction) using the well-known Hopcraft-Karp algorithm (Hopcroft & Karp, 1973) in $\Theta(m\sqrt{n})$ operations. In the second stage, the algorithm uses an *augmentation process* to turn the matching obtained in the first stage into the optimal solution of the aforementioned combinatorial problem in $\Theta(m\sqrt{n})$ operations. This step is based on a careful modification of Hopcraft-Karp algorithm and, hence, depends heavily on this particular choice of algorithm for the MBM problem.

3.3 Alternative Formulation

We propose an alternative combinatorial reformulation of the MISSC problem that enables us to bypass the augmentation process proposed by Olshevsky completely. This reformulation is based on a transformation of the state graph $\mathcal{G}_{\overline{A}}$ into another graph such that the solution to the MISSC problem can be obtained by a direct application of *any* maximum bipartite matching algorithm on this new graph.

In what follows, we first introduce some definitions needed to state our main result.

Definition 3.1. We denote by $\mathsf{mbmt}(m, n)$ *the running time of the fastest possible algorithm for finding a maximum matching in a bipartite graph with n vertices and m edges. Similarly, we denote by* $\mathsf{missct}(m, n)$ *the running time of the fastest algorithm for solving the MISSC problem on an $n \times n$ structural state matrix with m non-zero entries.*

Our main result is stated in the following theorem:

Theorem 2. *Given a structured state matrix $\overline{A} \in \{0,1\}^{n \times n}$ with m non-zero entries, the MISSC problem can be solved in* $\mathsf{mbmt}(\Theta(m), \Theta(n))$ *time.*

To prove Theorem 2, we will make use of the following lemma, which is an immediate consequence of the results in (Pequito, Kar & Aguiar, 2015; Dion & Commault, 2013):

Lemma 1. Given a structured state matrix $\overline{A} \in \{0,1\}^{n \times n}$ and a set of directly controlled states $J \subseteq [n]$, the structured system $\langle \overline{A}, I_J \rangle$ is structurally controllable if and only if the following two conditions are satisfied:

(1) Every SSCC of the state subgraph $\mathcal{G}_{\overline{A}}$ of $\mathcal{G}_{\overline{A},I_J}$ contains at least one directly controlled state node.
(2) There exists a matching $\mathcal{M}$ in the system graph $\mathcal{G}_{\overline{A},I_J}$ in which every state node $x \in \mathcal{X}$ is matched.

We call $\mathcal{M}$ *a* witness matching *of* J.

The result in Theorem 2 is based on the algorithmic procedure described in Algorithm 1. As we will certify below, this algorithm solves the MISSC problem via a direct application of *any* maximum bipartite matching algorithm, without the need of an augmentation process (or any other time consuming post-processing).

Algorithm 1

Input: A structured state matrix $\overline{A} \in \{0,1\}^{n\times n}$.
Output: A minimum-size set J of directly controlled states.

(1) Let $\mathcal{G}_{\overline{A}}$ be the digraph associated with the structured state matrix $\overline{A}$. Let $\mathcal{S}_1, \ldots, \mathcal{S}_k$ be the k SSCC's of $\mathcal{G}_{\overline{A}}$.
(2) Let $\mathcal{B}_{\overline{A}} = (\mathcal{X}_s \cup \mathcal{X}_d, E_{\mathcal{X}})$ be the *bipartite representation* of $\mathcal{G}_{\overline{A}}$, where $E_{\mathcal{X}} \subseteq \mathcal{X}_s \times \mathcal{X}_d$.
(3) Construct an augmented state graph $\widetilde{\mathcal{G}}_{\overline{A}}$ from $\mathcal{B}_{\overline{A}}$ by adding k new vertices $\widetilde{\mathcal{X}}_s = \{\widetilde{x}_1, \ldots, \widetilde{x}_k\}$ to $\mathcal{X}_s$ where each $\widetilde{x}_i$ has an edge to each x_j^d whenever $x_j \in \mathcal{S}_i$.
(4) Compute a maximum matching $\mathcal{M}$ of the graph $\widetilde{\mathcal{G}}_{\overline{A}}$. For any $i \in [k]$, if $\widetilde{x}_i$ is not matched in $\mathcal{M}$, pick any neighbor x_j^d of $\widetilde{x}_i$ (which must be matched in $\mathcal{M}$), remove the matching edge of x_j^d, and match x_j^d with $\widetilde{x}_i$. Let this new matching be $\widetilde{\mathcal{M}}$.
(5) Let J be the indices of the set of vertices in $\mathcal{X}_d$ that are *not* matched by $\widetilde{\mathcal{M}}$, union with all vertices that are matched to some vertex in $\widetilde{\mathcal{X}}_s$; return J.

We now have all the elements to provide a proof of Theorem 2.

Proof. We first show that for the output of the algorithm, i.e., the set J, $\langle \overline{A}, I_J \rangle$ is structurally controllable (feasibility), and then prove $|J|$ is minimum (minimality). To show feasibility, by Lemma 1, we only need to prove the existence of a witness matching. This can be obtained by taking the matching $\widetilde{\mathcal{M}}$ (Step 4 in Algorithm 1), and matching all unmatched node in $\widetilde{\mathcal{M}}$ to an input node in J.

To show the minimality, first note that by the definition of J (Step 5 in Algorithm 1)

$$|J| = n - |\widetilde{\mathcal{M}}| + k \Rightarrow |\widetilde{\mathcal{M}}| = n + k - |J|.$$

Let J^* be any optimal solution to MISSC with input matrix $\widetilde{A}$. We will define a matching $\mathcal{M}_{J^*}$ in $\widetilde{\mathcal{G}}_{\overline{A}}$ (the graph defined in Step 3 of Algorithm 1) with matching size $(n + k - |J^*|)$. Since $\widetilde{\mathcal{M}}$ is a maximum matching in $\widetilde{\mathcal{G}}_{\overline{A}}$, $|\widetilde{\mathcal{M}}| \geq |\mathcal{M}_{J^*}|$, which implies $|J| \leq |J^*|$, implying that J is also an optimal solution. We now show how to construct such a matching $\mathcal{M}_{J^*}$.

By Lemma 1, there exist a witness matching $\mathcal{M}$ of J^*, where every state node is matched and every SSCC contains a node matched by an input node. Remove all edges in $\mathcal{M}$ that are incident on an input node and denote the remaining matching $\mathcal{M}'$. Then

$$|\mathcal{M}'| = |\mathcal{M}| - |J^*| \Rightarrow |J^*| = |\mathcal{M}| - |\mathcal{M}'| = n - |\mathcal{M}'|.$$

Note that $\mathcal{M}'$ is a matching in $\mathcal{G}_{\overline{A}}$, and can be directly transformed into a matching in the bipartite representation $\mathcal{B}_{\overline{A}}$ of $\mathcal{G}_{\overline{A}}$. With a slight abuse of notation, we still denote the resulting matching $\mathcal{M}'$. Extend the graph $\mathcal{G}_{\overline{A}}$ to $\widetilde{\mathcal{G}}_{\overline{A}}$ (as in Step 3 of Algorithm 1), and define the matching $\mathcal{M}_{J^*}$ in $\widetilde{\mathcal{G}}_{\overline{A}}$ by adding the following additional matching edges to $\mathcal{M}'$: match each $\widetilde{x}_i$ to an arbitrary unmatched vertex in S_i (i.e., the i-th SSCC). Since each SSCC contains at least one node unmatched in $\mathcal{M}'$, the size of the matching increases by exactly k, and hence

$$|\mathcal{M}_{J^*}| = k + |\mathcal{M}'| = n + k - |J^*|.$$

proving the minimality of $|J|$.

Finally, note that all steps of Algorithm 1 except for Step 4 can be implemented in $O(n + m)$ time. Moreover, we have that the number of nodes and edges in $\widetilde{\mathcal{G}}_{\overline{A}}$ satisfy $|\mathcal{V}(\widetilde{\mathcal{G}}_{\overline{A}})| = \Theta(n)$ and $|\mathcal{E}(\widetilde{\mathcal{G}}_{\overline{A}})| = \Theta(m)$, and hence Step 4 requires $\mathsf{mbmt}(\Theta(m), \Theta(n))$ time by Definition 3.1 and since $\mathsf{mbmt}(\Theta(m), \Theta(n))$ is always $\Omega(m + n)$ (the input size), the running time requirement of Algorithm 1 is $\mathsf{mbmt}(\Theta(m), \Theta(n))$. ■

As mentioned in Section 1, the fastest available algorithm to find the maximum bipartite matching runs in $\min\{O(m\sqrt{n}), \tilde{O}(n^{2.37}), \tilde{O}(m^{10/7})\}$ operations. Hence, the following corollary is a direct consequence of Theorem 2:

Corollary 1. *Given a structured state matrix* $\overline{A} \in \{0,1\}^{n\times n}$ *with* m *non-zero entries, the MISSC problem can be solved in* $\min\{O(m\sqrt{n}), \tilde{O}(n^{2.37}), \tilde{O}(m^{10/7})\}$ *operations.*

Furthermore, apart from designing an algorithm (Algorithm 1) to solve the MISSC problem as fast as the MBM problem (up to constant factors), the following theorem states that any (possible) algorithm for the MISSC problem has to essentially solve the instances of MBM problem also, implying that use of algorithms for MBM problem as a subroutine in MISSC computation is indeed necessary.

Theorem 3. *Given any undirected bipartite graph* $\mathcal{B}$ *with* n *vertices and* m *edges, the* maximum matching size *in* $\mathcal{B}$ *can be determined in* $\mathsf{missct}(\Theta(m), \Theta(n))$ *time.*

In order to prove Theorem 3, we design Algorithm 2.

Algorithm 2

Input: A bipartite graph $\mathcal{B}(P \cup Q, E)$, with $E \subseteq P \times Q$.
Output: The size s of a maximum matching in $\mathcal{B}$

(1) Transform $\mathcal{B}$ into a digraph by directing each edge from P to Q. Let $\overline{A} \in \{0,1\}^{n\times n}$ be the adjacency matrix of the resulting digraph.
(2) Solve MISSC on the matrix $\overline{A}$ and denote the output by J.
(3) Return $(n - |J|)$ as the size of the maximum matching in $\mathcal{B}$.

Proof. When we direct every edge in $\mathcal{B}(P, Q, E)$ from P to Q, every vertex in P has zero in-degree and hence forms a single-vertex SSCC. By Theorem 1, the solution J must ensure that every state node is reachable from an input node, and hence J must contain an input node for each vertex in P. Again by Theorem 1, there exists a matching in the associate system graph $\mathcal{G}_{\overline{A}, I_J}$ that matches every state node. Since all the state nodes in P must be matched with input nodes, the matching only needs to match the state nodes in Q. If the size of the maximum matching in the original graph $\mathcal{B}$ is s, then at most s vertices in Q can be matched using distinct vertices in P, and the remaining vertices must be matched using input nodes. Therefore, $|J| \geq |P| + (|Q| - s) = n - s$. Moreover, $|J| = n - s$ is achievable by extending a maximum matching in $\mathcal{B}$ (direct the edges) and matching all vertices in P and all unmatched vertices in Q to input nodes. ∎

Note that Theorem 3 essentially states that if one is able to somehow speed up the MISSC problem computation beyond the bounds given by Corollary 1, then one obtains an algorithm which is faster than the current best algorithms for determining the *size* of a maximum matching in a bipartite graph; a result which would be a great breakthrough in the field of theoretical computer science.

4. CONCLUSIONS

We have considered the minimum input selection problem for structural controllability (MISSC). The fastest algorithm (up-to-date) to solve this problem was recently proposed by Olshevsky in (Olshevsky, 2015). Olshevsky's algorithm is based on a careful modification of Hopcraft-Karp algorithm and, hence, cannot run faster than $\Theta\left(m\sqrt{n}\right)$ operations. In this paper, we have proposed an alternative algorithm to solve MISSC based on a reformulation that allows to use *any* algorithm to solve the maximum bipartite matching problem, including Mucha-Sankowski and Madry's algorithms, hence, yielding a faster algorithm for the MISSC problem. Conversely, we have also showed that any algorithmic improvement on solving the MISSC problem would result in an improvement in MBM computation, which would be of great interest for theoretical computer scientists.

ACKNOWLEDGEMENTS

We would like to thank Dr. Sergio Pequito for fruitful discussions on structural controllability problems and pointers to current literature. We are also grateful to the anonymous reviewers for their insightful comments.

REFERENCES

C. Commault and J.-M. Dion. Controllability through input addition for graph-based systems. *Proceedings of the IFAC Symposium on Systems Structure and Control*, 2013.

C. Commault and J.-M. Dion. Input addition and leader selection for the controllability of graph-based systems. *Automatica*, vol. 49, pp. 3322-3328, 2013.

C. Commault and J.-M. Dion. The minimal controllability problem for structured systems. *hal-01064961*, 2014.

E.J. Davison and S.H. Wang. Properties of linear time-invariant multivariable systems subject to arbitrary output and state feedback. *IEEE Transactions on Automatic Control*, vol. 18, pp. 24-32, 1973.

R. Diestel. *Graph theory.* Springer-Heidelberg.

J.-M. Dion, C. Commault, and J. van der Woude. Generic properties and control of linear structured systems: a survey. *Automatica*, vol. 39, no. 7, pp. 1125-1144, 2003.

J.-M. Dion and C. Commault. Input addition and leader selection for the controllability of graph-based systems. *Automatica*, vol. 49, pp. 3322-3328, 2013.

J.E. Hopcroft and R.M. Karp. An nˆ5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

D. Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, Third Edition. Addison-Wesley, 1997.

F. Le Gall. Powers of tensors and fast matrix multiplication. *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, 2014.

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

Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási. Controllability of complex networks *Nature*, vol. 473, pp. 167-173, 2011.

A. Madry. Navigating central path with electrical flows: from flows to matchings, and back. *IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 253–262, 2013.

M. Mucha and P. Sankowski. Maximum matchings via gaussian elimination. *IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 248–255, 2004.

A. Olshevsky. Minimum input selection for structural controllability. *ArXiv:1407.2884v2 [math.OC]*, 2015.

S. Pequito, S. Kar, and A.P. Aguiar. A structured systems approach for optimal actuator-sensor placement in linear time-invariant systems. *Proceedings of the American Control Conference*, 2013.

S. Pequito, S. Kar, and A. P. Aguiar. A framework for structural input/output and control configuration selection in large-scale systems. *IEEE Transactions on Automatic Control*, accepted for publication.

R. Shields and J.B. Pearson. Structural controllability of multiinput linear systems. *IEEE Transactions on Automatic Control*, vol. 21, no. 2, pp. 203-212, 1976.

V.V. Williams. Multiplying matrices faster than Coppersmith-Winograd. *ACM Symposium on Theory of Computing (STOC)*, 2012.