

Research article

## Local controllability of complex networks

Chang Luo\*

College of Mathematics and Statistics, Huanggang Normal University, Hubei, 438000, China

\* **Correspondence:** Email: luochang@hgnu.edu.cn; Tel: +86 13886145010.

**Abstract:** The control of complex networks has been studied extensively in the last decade, with different control models been introduced. In this paper, we propose a new network control framework, called local controllability. Local controllability extends the entire network control onto a local scale, and it concerns about the minimum number of inputs required to control a subset of nodes in a directed network. We develop a mathematical formulation for local controllability as an optimization problem and show that it can be solved by a cubic-time algorithm. Moreover, applications to both real networks and model networks are presented and results of these numerical studies are then discussed.

**Keywords:** complex networks; network control; local controllability; model networks

### 1. Introduction

Complex networks [1, 2], which are usually studied in network theory, have been commonly used to model diverse natural and artificial systems. These systems range from biological systems such as DNA transcriptional regulation in the cell nucleus and neural interaction in the brain, to social, economic systems, and to technological systems such as the World Wide Web and the Internet. Complex networks arising in these systems display substantial nontrivial topological features, with patterns of connection between their elements neither purely regular nor purely random.

With the ever-increasing amount of data arising from complex systems in various fields, complex networks have been shown to be an effective modelling technique for exploiting this complexity and studying the large-scale properties of the complex systems. As a result of multidisciplinary efforts, a large amount of literature that has studied various aspects of complex networks is now available. Among these studies, there are methods that were proposed to characterize the roles of individual nodes or to uncover meaningful small subnetworks within a network [3–5]; development of network models to explain the existing structural patterns in real networks [6, 7];

both theoretical and numerical studies on different network properties considering the dynamics that take place on networks, including diffusion [8], evolution [9], percolation [10], synchronizability [11].

One particularly distinctive modelling aspect of complex networks (also dependent on network dynamics) is controllability. Controllability is a concept that is commonly used in the study of dynamical systems [12], and recently it found important applications in the study of complex networks (network control). Two factors have complicated the understanding of controllability of complex systems: (1) the complex network architecture encapsulating the interactions between the system's components, (2) the dynamical rule that governs the time-dependent (temporal) behavior of each component. As a result, understanding how network structure affects our ability to control complex networks becomes increasingly important for designing optimal control schemes to tame the network dynamics.

A pioneering work on the controllability of complex networks was accomplished by Liu et al. [13], in which a general framework for control of complex networks was proposed, called *network controllability*. Liu et al. showed that the minimum number of driver nodes required to fully control a directed network is determined by a maximum

matching of the digraph, and further revealed that driver nodes tend to avoid ‘hubs’ in both real and model networks [13]. Many subsequent works regarding the control of complex networks have appeared in literature ever since.

### 1.1. Literature review

Here, we first review some of these controllability-related works and then describe how our paper is related to the existing literature in the next subsection 1.2. Nepusz and Vicsek [14] studied the controllability of complex networks in which the dynamics take place on the edges of the network (switchboard dynamics) and showed that edge dynamics lead to different controllability properties from the nodal dynamics. They proved that the minimum number of driver nodes can be determined by the divergent nodes in the network and provided an efficient algorithm with time complexity  $O(m+n)$ . Yuan et al. [15] provided a framework for controlling any network (undirected and weighted) called *exact controllability* based on the PBH rank condition. They proved that the minimum number of driver nodes equals the maximum geometric multiplicity of any eigenvalue of the network adjacency matrix. Note that the framework of exact controllability can reproduce structural controllability when the network is structured and thus it seems to have wider application. When a controlled system is not fully controllable, one can consider the maximal controllable subsystem by determining the generic rank [16]. Liu et al. exploited this result to define the notion of *control centrality*, which captures the generic dimension of the maximal controllable subsystem that any single node is able to induce when we control this node only [17].

The study that is most related or similar to our paper was done by Gao et al., in which they proposed the scheme of *target control* [18]. The focus of their study and ours are both how to control a pre-selected subset of nodes within a directed complex network. Gao et al. used the rank condition for output controllability to derive a ‘k-walk’ theory in case that one driver node is only needed, and further provided a greedy algorithm to approximate the minimum number of inputs sufficient for target control [18]. Furthermore, there exists a comprehensive survey paper about control of complex systems recently crafted by Liu and Barabási [19].

### 1.2. Our contribution

Despite Liu’s pioneering work on network controllability which focuses on graph-theoretic algorithms to identify minimum inputs needed to fully control a directed complex network, it is yet unclear whether there is a polynomial-time algorithm such that only a subset of nodes are controlled within a directed network. Considering that in many real control scenarios only a small fraction of nodes need to be controlled, we try to answer the following research questions:

- (1) Is there a rigorous mathematical formulation for ‘only a fraction of nodes being controlled in a directed network’?
- (2) Is there a polynomial-time algorithm to identify a minimum number of inputs such that the given subset of nodes are controlled?
- (3) What are the possible network properties that affect the ability to control a subset of nodes (local controllability)?

For such research purposes, we introduce the framework of *local controllability* in this paper. Local controllability is defined as the minimum number of inputs required to control a subset of nodes in a directed network (please refer to subsection 2.3 for a detailed problem formulation). Using graph-theoretic tools, we show that the local controllability can be solved by identifying minimum-weight cycle cover or minimum-weight perfect matching in a weighted directed network extended from the original given network. Furthermore, based on minimum-weight perfect matching, we develop a cubic-time algorithm to compute the local controllability and also identify the driver nodes.

With the framework of local controllability, it is convenient (in theory) to study the control of a meaningful subset of nodes in a real network. Simulation studies show that in both ER and scale-free networks, nodes with higher degree/betweenness are easier to control and the hub attack can cause more damage to local controllability than random removal of out-neighbors (in-neighbors) of the pre-selected subset of nodes.

Here, we explain a little bit more on how our work is related to previous studies. The ‘core idea’ of our study is to look for a minimum (local) graph structure that is able to control the pre-selected subset of nodes by injecting a minimum number of inputs to the given directed network.

Thus, our work is established upon the work of Liu et al. on network controllability [13], and can be seen as an extension of network controllability onto a local scale.

In existing literature, the study that is most similar to our paper was conducted by Gao et al., who proposed the scheme called target control [18]. However, there are certain differences in terms of problem formulation and methods. Our framework of local controllability is based on the graph-theoretic characterization of a controllable subsystem that contains the pre-selected subset of nodes required for control and our algorithm can produce the minimum set of driver nodes with cubic time complexity. We argue that local controllability differs from target control by the mathematical formulation and allows for realistic local control by identifying a local ‘cacti’ structure.

The rest of the paper is organized as follows. In section 2, we present the model setup by first reviewing some preliminary results followed by a detailed problem formulation on local controllability. The theoretical results on local controllability are given in section 3. In section 4, numerical results are presented, including an application of local controllability to the mouse brain networks as well as model networks. In section 5, some concluding remarks are given and possible future work is discussed. We leave the formal proof for the main theorems in the Appendix section.

## 2. Model setup

In this section, we first present some preliminary results that are necessary for establishing our model of local controllability and then provide a detailed description of our problem formulation.

### 2.1. Preliminaries on controllability and structural controllability

Controllability is an important concept derived from studies of dynamical systems [12]. The controllability of complex networks is related to the controllability of a linear time-invariant system [13].

A linear time-invariant system is usually represented by the differential equation below:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (2.1)$$

where

$$t \in \mathbb{R}, x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m,$$

and

$$A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}.$$

In the linear time-invariant system (2.1),  $x(t)$  is called the *state vector*,  $u(t)$  is called the *input vector*, whereas  $A, B$  are called the *state matrix*, and the *input matrix*, respectively. A linear time-invariant system is generally denoted by the pair  $(A, B)$ .

For a general dynamical system, controllability is defined as the ability to steer the system to an arbitrary state from any given initial state within finite time. Thus, for a linear time-invariant system  $(A, B)$ , the definition is stated as below.

**Definition 2.1.** *The pair  $(A, B)$  is controllable on  $[t_0, t_1]$  if and only if for any  $x_0, x_1 \in \mathbb{R}^n$ , there exists  $u \in L^2([t_0, t_1], \mathbb{R}^m)$  that steers the system from  $(x_0, t_0)$  to  $(x_1, t_1)$ , that is,*

$$x_1 = e^{t_1 - t_0} x_0 + \int_{t_0}^{t_1} e^{A(t-\tau)} Bu(\tau) d\tau.$$

The best well-known test for controllability of a linear time-invariant system  $(A, B)$  is based on the *controllability matrix*:  $P = [B \ AB \ \dots \ A^{n-1}B]$ . It is known as Kalman’s rank condition [20]: the pair  $(A, B)$  is controllable if and only if  $\text{Rank}(P) = n$ .

The study of network controllability and our proposed local controllability relies on the definition of ‘structural controllability’, which was first introduced by Lin [21]. In order to define structural controllability, we make the assumption that the exact values of the entries in  $A$  and  $B$  are unknown, i.e., the matrices  $A$  and  $B$  are structured matrices.

**Definition 2.2.**  *$M$  is called a structured matrix, if its entries are either fixed zeros or independent free parameters.  $\tilde{M}$  is called an admissible matrix of  $M$  if it can be obtained by fixing the free parameters of  $M$  at some particular real values.*

**Definition 2.3.** *Let  $A, B$  be structured matrices. Then the pair  $(A, B)$  is called structurally controllable if there exists an admissible pair  $(\tilde{A}, \tilde{B})$  that is controllable.*

The power of structural controllability comes from the fact that if a system is structurally controllable then it is

controllable (in the usual sense) for almost all possible parameter realizations [22, 23]. In other words, structural controllability is a generic property [24]. Thus, it is possible to decide the controllability of a networked system, if we have an accurate map of the the system's wiring diagram (network structure) without knowing the precise weight of each link. This is where graph theory tools come into play.

Lin's structural controllability theorem provides a graph-theoretic characterization for structural controllability. Interested readers may refer to [21] and [25] for the formal proof.

**Proposition 2.1. (Lin's structural controllability theorem)** *For a linear structured system  $(A, B)$ , the following statements are equivalent:*

1. *The pair  $(A, B)$  is structurally controllable.*
2. *The graph of  $(A, B)$  contains no non-accessible nodes and no dilation.*
3. *The graph of  $(A, B)$  is spanned by a cacti.*

## 2.2. Preliminaries on network controllability

Network controllability, is a framework of network control first proposed by Liu et al. in his seminal paper in 2012 [13]. It is based on Lin's structural controllability theory [21], by relaxing the requirement on the specific weight of each link in a directed complex network. Thus, given a directed network (only the topology is known), the problem is to identify a set of driver nodes that, if driven by different input signals, can offer full control over the network [13]. In other words, we are interested in making a directed networked system controllable with minimum inputs. With the minimum number of inputs, the way of connecting the inputs to nodes in the network is called a *control configuration*. Generally, there are different control configurations that are able to deliver full control.

Liu et al. showed that the minimum input problem can be mapped to a purely graph-theoretic problem called maximum matching [13]. This important result is known as the minimum input theorem, as given in proposition 2.2.

**Proposition 2.2. (Minimum input theorem)** *The minimum number of inputs,  $N_D$ , is one if there is a perfect matching in a directed network  $G$ . Otherwise, it equals the number of unmatched nodes with respect to any maximum matching*

*$M^*$ , and to fully control  $G$ , each unmatched node needs to be directly connected to a different input node. Therefore,*

$$N_D = \max \{1, n - |M^*|\}. \quad (2.2)$$

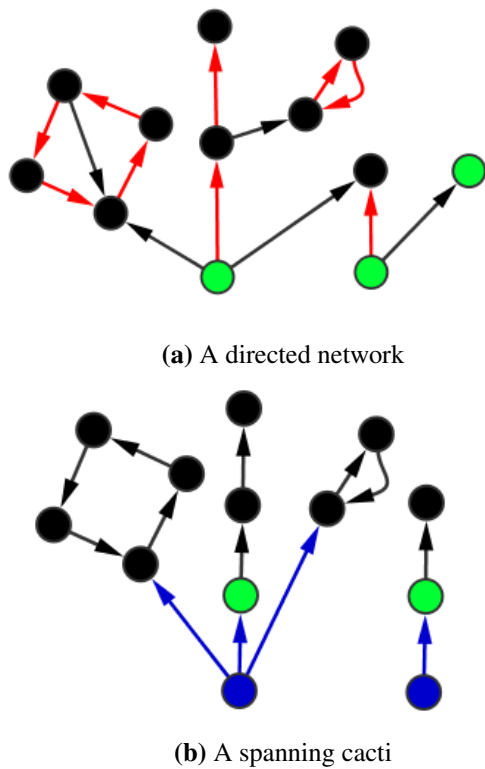
Minimum input theorem provides a way to find a control configuration with the minimum number of inputs to fully control a directed network. Indeed, there exists some algorithm to construct such a control configuration based on minimum spanning cacti by identifying a set of matching paths and matching cycles [26]. In Figure 1, we give an illustration of a maximum matching in a directed network and a spanning cacti that can drive full control over the network, based on minimum input theorem.

To find a maximum matching in a directed network  $G$ , one can find a maximum matching in a (undirected) bipartite representation of  $G$ . Furthermore, a maximum matching in a bipartite graph can be efficiently identified by using the well-known Hopcroft-Karp algorithm [27].

## 2.3. Problem formulation for local controllability

The motivation for local controllability is as follows: (1) it is neither feasible nor necessary to control the full network, due to the large size of the complex network; (2) sometimes it is realistic or useful to control just a subset of nodes for some desirable task. It is important to study local controllability, not only due to its theoretic interest, but also because of its potential applications in the control of real networks considering that there are certain situations when one prefers to control just a subset of nodes in a network. For example, turning on a specific subset of transcription factors can convert one cell type to another and it is useful to control a subset of target molecules (biomarkers) in metabolic networks for the purpose of disease cure [28]. For another example, it might be interesting to control a specific brain area (with specific functions) in the brain network.

We are particularly interested in how to control a subset of nodes in a directed network with a minimum number of (external) inputs. Note that the framework of local controllability is an extension of network controllability onto a local scale, and hence, is based on the theory of structural controllability. The basic modelling assumption for local controllability is that we are given a directed network with only its topology known, i.e., link weight



**Figure 1.** An example of constructing a spanning cacti based on a maximum matching in a directed network. In (a), the set of arcs in red form a maximum matching. In (b), the nodes in blue are the input nodes of the cacti. The nodes in green are the unmatched nodes with respect to the maximum matching.

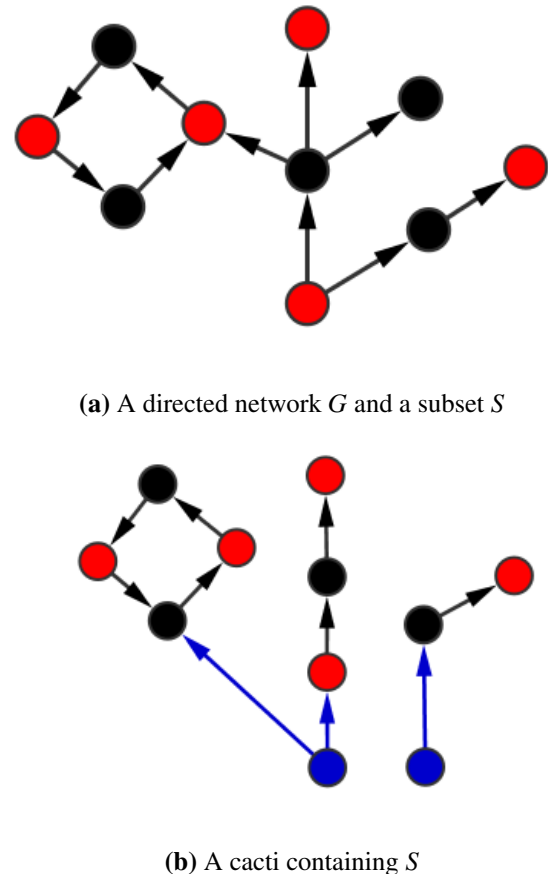
information is not necessary. Then, the core problem we study is formally stated as follows:

*Given any nonempty subset  $S$  of nodes in any directed network  $G = (V(G), A(G))$ , find a minimum number of different inputs  $\{u_1, \dots, u_l\}$  such that when the inputs are connected to the nodes in  $G$  appropriately, the subset  $S$  can be controlled (regardless of whether nodes in  $V(G) \setminus S$  being controlled or not).*

Here, the statement that a subset  $S$  is controlled in a directed network  $G$ , is equivalent to that there is a subsystem which contains  $S$  and is controllable after introducing inputs to the network  $G$ , also equivalent to that there is a cacti containing  $S$  after introducing inputs to the network  $G$ .

The problem above is well-defined, since we know

that the whole networked system becomes controllable by introducing  $N_D$  external inputs. In Figure 2, we give an illustration of controlling a subset of nodes in a directed network.



**Figure 2.** An example of controlling a subset  $S$  of nodes in a directed network  $G$ . In (b), the nodes in blue are the input nodes of the cacti. The subset  $S$  is the set of nodes in red.

Next, we give a rigorous mathematical formulation for the problem stated above, which can be easily translated into an (combinatorial) optimization problem.

**Mathematical formulation:** Suppose  $V(G) = \{v_1, v_2, \dots, v_n\}$  and  $S = \{v_{i_1}, v_{i_2}, \dots, v_{i_r}\}$  with  $1 \leq i_1 < i_2 < \dots < i_r \leq n$ . Let  $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$  be a column vector of  $n$  state functions. Let  $u(t) = (u_1(t), u_2(t), \dots, u_n(t))^T$  be a column vector of  $n$  input functions. Let  $E = (e_{ij}) \in \mathbb{R}^{n \times n}$  be a binary diagonal matrix, that is,  $e_{ii} \in \{0, 1\}, i = 1, 2, \dots, n$ . Let

$F = (f_{ij}) \in \mathbb{R}^{n \times n}$  be a binary diagonal matrix such that  $f_{i_k i_k} = 1, k = 1, 2, \dots, r$ .

Let  $C = (c_{ij}) \in \mathbb{R}^{n \times n}$  be a structured matrix in which each entry is a free parameter. Let  $D = (d_{ij}) \in \mathbb{R}^{n \times n}$  be a structured matrix defined as

$$d_{ij} = \begin{cases} \text{a free paramter,} & \text{if } (v_j \rightarrow v_i) \in A(G), \\ \text{zero,} & \text{if } (v_j \rightarrow v_i) \notin A(G). \end{cases}$$

Define the structured matrix  $A = FDF \in \mathbb{R}^{n \times n}$ , and define the structured matrix  $B = FCE \in \mathbb{R}^{n \times n}$ . The pair  $(A, B)$  represents a linear structured system of the form (2.1).

Then, the local controllability of  $S$  in  $G$  is defined as

$$lc(G, S) = \min_{E, F} \left\{ \sum_{i=1}^n e_{ii} : (A, B) \text{ is structurally controllable} \right\}. \quad (2.3)$$

The local controllability of  $S$  in  $G$ , denoted by  $lc(G, S)$ , is equal to the minimum number of inputs required to control  $S$  in a directed network  $G$ , i.e., the minimum number of inputs required to construct a cacti structure (minimum controllable graph structure) containing  $S$ . As the definition suggests,  $lc(G, S)$  depends on both  $G$  and  $S$ .

### 3. Theoretical results of local controllability

In this section, we present the main theoretical results about local controllability based on the problem formulation described above. We show that the local controllability  $lc(G, S)$  can be found by identifying a minimum-weight cycle cover or a minimum-weight perfect matching in a weighted directed network, which is extended from  $G$ . Before we proceed to our main theorems, we should incorporate some graph-theoretic notations leading to the the definition of minimum-weight cycle cover.

#### 3.1. Minimum-weight cycle cover

Let  $G = (V(G), A(G))$  be a directed network with  $|V(G)| = n$ , and  $S$  be a subset of nodes in  $G$ , i.e.,  $S \subseteq V(G)$ . Now, extend  $G$  to the complete directed network  $G^*$ :  $G^* = (V(G^*), A(G^*))$  with  $V(G^*) = V(G)$  and  $A(G^*) = V(G) \times V(G)$ , where  $V(G) \times V(G)$  denotes the Cartesian product of the set  $V(G)$  with itself, that is, for any node  $u \in V(G^*)$  and any node  $v \in V(G^*)$ , there is a directed edge  $(u, v) \in A(G^*)$ .

Next, we assign a weight function  $w$  onto the edges in  $G^*$  as follows. Let  $V(G) = \{v_1, v_2, \dots, v_n\}$ . Suppose  $S = \{v_{i_1}, v_{i_2}, \dots, v_{i_r}\}$  with an indexing set  $I_S = \{i_1, i_2, \dots, i_r\}$  and  $V(G) \setminus S = \{v_{j_1}, v_{j_2}, \dots, v_{j_{n-r}}\}$ . Denote the set of self-loops for each node in  $V(G) \setminus S$  by  $\Gamma$ , that is,

$$\Gamma = \{(v_{j_1}, v_{j_1}), (v_{j_2}, v_{j_2}), \dots, (v_{j_{n-r}}, v_{j_{n-r}})\}.$$

Then, the weight function  $w$  is a mapping,  $w : A(G^*) \rightarrow \mathbb{R}$ , defined as

$$w((u, v)) = \begin{cases} 0, & (u, v) \in \Gamma, \\ 1, & (u, v) \in A(G) \setminus \Gamma, \\ n, & (u, v) \in A(G^*) \setminus (A(G) \cup \Gamma). \end{cases} \quad (3.1)$$

**Definition 3.1.** Let  $G$  be any directed network and  $S$  be a subset of nodes in  $G$ . A cycle cover of  $S$  in  $G$  is a set of disjoint (independent) cycles in  $G$  that contain all the nodes in  $S$ , that is,  $C = C_1 \cup \dots \cup C_k$  is a cycle cover of  $S$  in  $G$  if all the cycles  $C_1, \dots, C_k$  are disjoint and each node in  $S$  is contained within some cycle  $C_j$ .

**Definition 3.2.** For any directed network  $G = (V(G), A(G))$  and a subset  $S$  of nodes in  $G$ , let  $C = C_1 \cup \dots \cup C_k$  be any cycle cover of  $S$  in  $G$  and let  $w : A(G) \rightarrow \mathbb{R}$  be a weight function. The weight of the cycle cover  $C$ ,  $w(C)$ , is then defined as

$$w(C) = \sum_{i=1}^k w(C_i),$$

where the weight of a cycle,  $w(C_i)$ , is given by

$$w(C_i) = \sum_{(u,v) \in C_i} w((u, v)).$$

Moreover, if  $\bar{C}$  is a cycle cover such that

$$w(\bar{C}) = \min \{w(C) : C \text{ is a cycle cover of } S \text{ in } G\},$$

then  $\bar{C}$  is called a minimum-weight cycle cover of  $S$  in  $G$ .

Since  $G^*$  is a complete directed network, a cycle cover of  $S$  in  $G^*$  always exists. Our first result is that the local controllability  $lc(G, S)$  can be calculated by finding a minimum-weight cycle cover of  $S$  in  $G^*$ , as shown in Theorem 3.1.

#### 3.2. Main theorems

**Theorem 3.1.** Let  $G = (V(G), A(G))$  be a directed network with  $|V(G)| = n$ , and  $S$  be a subset of nodes in  $G$ . Let the weight function  $w : A(G^*) \rightarrow \mathbb{R}$  be defined as in (3.1). Let  $\bar{C}$  be a minimum-weight cycle cover of  $S$  in  $G^*$ , then we have

$$lc(G, S) = \max \left\{ 1, \left\lfloor \frac{w(\bar{C})}{n} \right\rfloor \right\},$$

where  $\lfloor \cdot \rfloor$  is the floor function.

We leave the proof for this theorem in the appendix (note that, the proof relies on Minimum input theorem 2.2). For the problem of minimum-weight cycle cover, it is readily stated as an integer linear program (ILP), which can be relaxed to a linear program (LP) due to total unimodularity. Thus, the local controllability can be calculated by solving a linear program for which efficient algorithms exist.

Alternatively, we present another theorem for computing local controllability in terms of minimum-weight perfect matching. Furthermore, the minimum-weight perfect matching, also known as assignment problem, can be solved by a cubic-time algorithm.

**Theorem 3.2.** Let  $G = (V(G), A(G))$  be a directed network with  $|V(G)| = n$ , and  $S$  be a subset of nodes in  $G$ . Let the weight function  $w : A(G^*) \rightarrow \mathbb{R}$  be defined as in (3.1). Let  $\bar{M}$  be a minimum-weight perfect matching in  $G^*$ , then we have

$$lc(G, S) = \max \left\{ 1, \left\lfloor \frac{w(\bar{M})}{n} \right\rfloor \right\}.$$

*Proof.* To show this theorem, we note that each cycle cover of  $S$  in  $G^*$  corresponds to a perfect matching in  $G^*$  as follows. Suppose  $C_{G^*}(S) = C_1 \cup \dots \cup C_k$  is a cycle cover of  $S$  in  $G^*$ , denote the set of nodes contained in  $C_{G^*}(S)$  by  $V(C_{G^*}(S))$ , then  $S \subseteq V(C_{G^*}(S))$ . Let  $V(G^*) \setminus V(C_{G^*}(S)) = \{w_1, w_2, \dots, w_p\}$ , then by definition,  $\{w_1, w_2, \dots, w_p\}$  is a subset of  $V(G) \setminus S = \{v_{j_1}, v_{j_2}, \dots, v_{j_{n-r}}\}$ . Form a self-loop for each node  $w_i$  and denote it by  $L_i$ , that is,  $L_i = (w_i, w_i)$  for  $i = 1, 2, \dots, p$ . Then  $C_{G^*}(S)$  corresponds to a perfect matching  $M_{G^*}$  in  $G^*$  defined as

$$M_{G^*} = C_{G^*}(S) \cup L_1 \cup L_2 \cup \dots \cup L_p.$$

It is easy to see that  $M_{G^*}$  is a cycle decomposition in  $G^*$ , thus a perfect matching in  $G^*$ . Furthermore, since  $w(L_i) = 0$  for  $i = 1, 2, \dots, p$ , we have  $w(M_{G^*}) = w(C_{G^*}(S))$ .

Conversely, a perfect matching in  $G^*$  easily corresponds to a cycle cover of  $S$  in  $G^*$  as follows. Given a perfect matching  $M_{G^*}$ , since a perfect matching is a cycle decomposition, we can denote  $M_{G^*} = \tilde{C}_1 \cup \tilde{C}_2 \cup \dots \cup \tilde{C}_p$  for some positive integer  $p$ , where each  $\tilde{C}_i$  is a cycle (including self-loop) in  $G^*$ . Identify all the cycles  $\tilde{C}_i$  which contains at least one node in  $S$ , and denote the set of all the identified

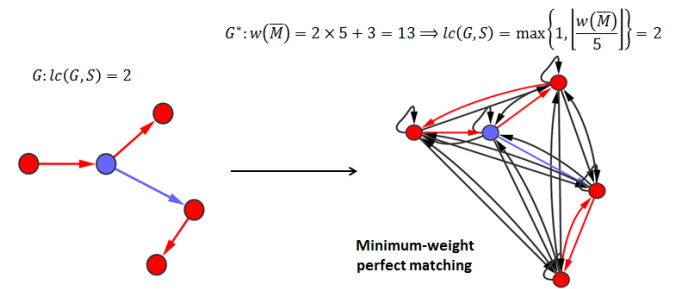
cycles by  $\{\tilde{C}_{j_1}, \tilde{C}_{j_2}, \dots, \tilde{C}_{j_s}\}$ . Then,  $M_{G^*}$  corresponds to a cycle cover  $C_{G^*}(S)$  of  $S$  in  $G^*$  defined as

$$C_{G^*}(S) = \tilde{C}_{j_1} \cup \tilde{C}_{j_2} \cup \dots \cup \tilde{C}_{j_s}.$$

Moreover,  $w(C_{G^*}(S)) \leq w(M_{G^*})$ .

Therefore, a minimum-weight perfect matching in  $G^*$  corresponds to a minimum-weight cycle cover of  $S$  in  $G^*$  and they have equal weight. Thus, this theorem follows from Theorem 3.1.  $\square$

Here, we illustrate Theorem 3.2 (minimum-weight perfect matching) by an example, as shown in Figure 3. In Figure 3, we have drawn the initial network  $G$  (in the left) and the complete directed network  $G^*$  (in the right).



**Figure 3.** An example to show that  $lc(G, S)$  can be calculated by finding a minimum-weight perfect matching in  $G^*$  (marked by the red arcs). The subset  $S$  is the set of all the nodes in red. Note that the set of red arcs also form a minimum-weight cycle cover of  $S$  in  $G^*$ .

Generally, to find a minimum-weight perfect matching in  $G^*$ , we used the Hungarian method for assignment problem [29]. The steps of Hungarian method based on cost matrix are described in the appendix, and our algorithm for local controllability uses the cost matrix-based Hungarian method as its core ingredient. Furthermore, our algorithm (with time complexity  $O(n^3)$ ) can not only calculate  $lc(G, S)$ , but also identify a minimum set of driver nodes and the corresponding control configuration. Note that, our algorithm is based on Theorem 3.2, and it is presented in subsection 3.3.

**Table 1.** The  $p$ -value for the alternative hypothesis that the local controllability of a subdivision is smaller than that of random subsets with the same size. The significant  $p$ -values ( $p_1 < 0.05$ ) are marked in bold.

The $p$ -value $p_1$										
	ipsilateral network					contralateral network				
P-value cutoff	0.05	0.02	0.01	0.005	0.001	0.05	0.02	0.01	0.005	0.001
Isocortex (38)	0.3254	0.1534	0.123	0.0686	<b>0.0188</b>	<b>0.0002</b>	<b>0</b>	<b>0.0004</b>	<b>0.0002</b>	<b>0</b>
Olfactory Areas (11)	0.8756	0.7816	0.94	0.9082	0.9532	0.9744	0.974	0.965	0.9658	0.9966
Hippocampus (11)	0.8756	0.7816	0.742	0.6776	0.5382	0.2194	0.1896	0.4202	0.4198	0.2938
Cortical Subplate (7)	0.948	0.8956	0.8754	0.8414	0.7722	0.9922	0.9894	0.9874	0.983	0.996
Striatum (12)	0.9974	0.9926	0.988	0.9744	0.9866	1	1	1	1	1
Pallidum (8)	0.9354	0.8716	0.8542	0.8072	0.6966	0.391	0.3744	0.3248	0.3202	0.8028
Thalamus (35)	0.97	0.9896	0.982	1	0.9998	1	1	1	1	1
Hypothalamus (20)	0.684	0.5046	0.4572	0.3538	0.2124	0.2594	0.2218	0.1826	0.1796	0.0836
Midbrain (21)	0.6588	0.4792	0.4224	0.332	0.1874	0.0868	0.068	<b>0.0478</b>	0.0522	0.0862
Pons (13)	0.9974	0.9874	0.9974	0.9936	0.9824	0.3672	0.3354	0.2966	0.2868	0.4012
Medulla (25)	0.8366	0.6838	0.6134	0.512	0.3058	<b>0.0062</b>	<b>0.0064</b>	<b>0.0036</b>	<b>0.0028</b>	<b>0.0234</b>
Cerebellum (12)	0.8514	0.758	0.7196	0.6374	0.493	0.6878	0.67	0.6162	0.612	0.4732

### 3.3. Algorithm for local controllability

#### The algorithm for local controllability:

Input: a directed network  $G = (V(G), A(G))$  with  $|V(G)| = n$  and a subset  $S$  of nodes in  $G$ .

Output: the minimum number of inputs required to control  $S$  in  $G$ , i.e.,  $lc(G, S)$  and a minimum set of driver nodes.

**Step 1:** Denote  $V(G) = \{v_1, v_2, \dots, v_n\}$  and assume  $S = \{v_{i_1}, v_{i_2}, \dots, v_{i_r}\}$  with an indexing set  $I_S = \{i_1, i_2, \dots, i_r\}$ . Denote the complement of  $S$  in  $V(G)$  by  $\bar{S}$ , i.e.,  $\bar{S} = V(G) \setminus S$ , and assume that there is an indexing set  $I_{\bar{S}} = \{j_1, j_2, \dots, j_{n-r}\}$  such that  $\bar{S} = \{v_{j_1}, v_{j_2}, \dots, v_{j_{n-r}}\}$ .

**Step 2:** Extend  $G$  to the complete directed network  $G^*$ , which is defined as

$$G^* = (V(G^*) = V(G), A(G^*) = V(G) \times V(G)).$$

Assign a weight function  $w$  onto the directed edges in  $G^*$ , i.e.,  $w : A(G^*) \rightarrow \mathbb{R}$ , defined as in (3.1).

**Step 3:** Construct a  $n \times n$  cost matrix  $C$  to be the weighted adjacency matrix of  $G^*$ . The cost matrix  $C = (c_{ij})$  is therefore defined as

$$c_{ij} = w((v_i, v_j)).$$

**Step 4:** Apply the cost matrix-based Hungarian method to  $C$  to find an optimal assignment  $\sigma$ . Output the local controllability to be

$$lc(G, S) = \max \left\{ 1, \left\lfloor \frac{\sum_{i=1}^n c_{i\sigma(i)}}{n} \right\rfloor \right\}.$$

**Step 5:** Given the optimal assignment  $\sigma$ , identify the set  $D = \{v_{\sigma(i)} : c_{i\sigma(i)} = n\}$ , output  $D$  to be the set of driver nodes.

## 4. Application of local controllability

In this section, we provide some numerical results derived from application of local controllability to both real networks and model networks.

### 4.1. Real networks

It is reasonable to locally control a subdivision (a subset of nodes) in the brain network as a specific subdivision might perform unique functional properties. For instance, the hypothalamus is concerned with the autonomic control of cardiovascular activity, respiratory and alimentary functions; the striatum is a massive nucleus in the basal forebrain that plays a pivotal role in modulating motor activity and higher cognitive function. Here, we applied the framework of local controllability onto mouse brain networks constructed from a mesoscale connectome of the mouse brain [30].

We constructed 10 networks, each consisting of 213 nodes (brain regions) and 12 subdivisions, by using 5 different  $p$ -value cutoffs from the original mouse connectome dataset [30]. For each  $p$ -value cutoff chosen, both ipsilateral and contralateral networks are obtained. By calculating the local controllability of each subdivision in each network, we find that Isocortex is the easiest to control while Striatum and Thalamus are difficult to control.

We then compared with the local controllability of random subsets with equal size (see Tables 1 and 2). The statistical comparisons show that Isocortex and Medulla are significantly easier (all  $p$ -values  $< 0.03$ ) to control than random subsets in contralateral networks, while Striatum



**Table 2.** The  $p$ -value for the alternative hypothesis that the local controllability of a subdivision is larger than that of random subsets with the same size.. The significant  $p$ -values ( $p_2 < 0.05$ ) are marked in bold.

The $p$ -value $p_2$										
P-value cutoff	ipsilateral network					contralateral network				
	0.05	0.02	0.01	0.005	0.001	0.05	0.02	0.01	0.005	0.001
Isocortex (38)	1	1	1	1	1	1	1	1	1	1
Olfactory Areas (11)	1	1	0.258	0.3224	0.175	0.0936	0.1032	0.124	0.1192	<b>0.0178</b>
Hippocampus (11)	1	1	1	1	1	1	1	0.8262	0.8224	0.9014
Cortical Subplate (7)	1	1	1	1	1	<b>0.049</b>	0.061	0.075	0.0806	<b>0.0248</b>
Striatum (12)	<b>0.0236</b>	0.061	0.076	0.1144	0.0624	<b>0</b>	<b>0</b>	<b>0.0004</b>	<b>0</b>	<b>0.0002</b>
Pallidum (8)	1	1	1	1	1	1	1	1	1	0.4638
Thalamus (35)	0.1176	<b>0.0394</b>	0.0576	<b>0.0016</b>	<b>0.002</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Hypothalamus (20)	1	1	1	1	1	0.8976	0.9176	0.9296	0.9346	0.9746
Midbrain (21)	1	1	1	1	1	0.9776	0.9844	0.9916	0.9888	0.9752
Pons (13)	<b>0.0242</b>	0.0724	<b>0.0172</b>	<b>0.0346</b>	0.0786	0.8572	0.8788	0.9012	0.9026	0.812
Medulla (25)	0.4362	0.6114	0.689	0.7688	0.8896	1	1	1	1	0.9948
Cerebellum (12)	1	1	1	1	1	0.5848	0.6046	0.644	0.6552	0.7676

and Thalamus are significantly harder (all  $p$ -values  $< 0.01$ ) to control. These differences might be related to the structural basis of the subdivisions: Isocortex, Medulla are homogeneous [31] and Striatum, Thalamus are heterogeneous in structure [32].

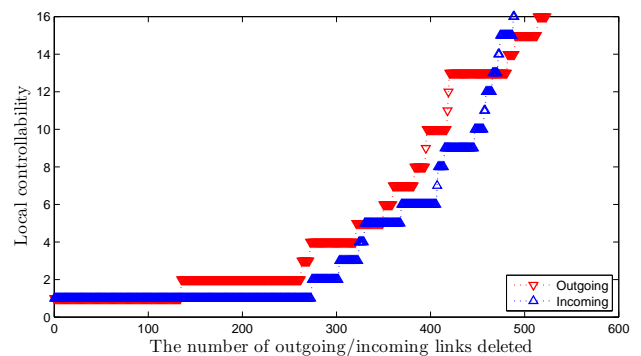
#### 4.2. Model networks

We consider two types of model networks: ER random network and scale-free network. The ER random networks are generated by ER model [33], and the scale-free networks are generated by the directed static model [34]. We have constructed 1000 realizations of both types of networks with 1000 nodes and each varying parameter.

On the one hand, we try to identify how  $lc(G, S)$  relies on the choice of  $S$  by dividing all the 1000 nodes into 10 equal-size subsets based on the degree/in-degree/out-degree of each node. The results show that in ER random networks, nodes with higher degree tend to be easier to control, but there is no definite (monotone) relationship between local controllability and in-degree or out-degree. This is also true for scale-free networks. Furthermore, we show that nodes with higher betweenness (measure) are easier to control in both types of model networks.

On the other hand, we try to explore how  $lc(G, S)$  depends on  $G$  with  $S$  fixed by investigating the robustness of local controllability against link and node removal in model networks. In the process of link removal, we fixed the subset  $S$  to be the top 100 hubs (nodes with highest degree), and randomly deleted an outgoing (incoming) link of  $S$  in each step until all outgoing (incoming) links of  $S$  have been

removed. We observe that this process will increase the local controllability of  $S$  and make it converge to the upper bound  $N_D(G_0[S])$ , which is the network controllability of the subnetwork  $G_0[S]$  of  $G_0$  induced by  $S$ .

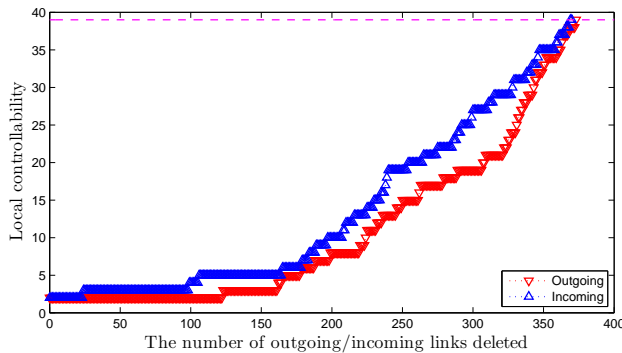


**Figure 4.** The plot of local controllability versus the number of outgoing/incoming links deleted for a scale-free network  $G_0$  with  $n = 1000$ ,  $\langle k \rangle = 4$  and  $\gamma = 2.5$ .

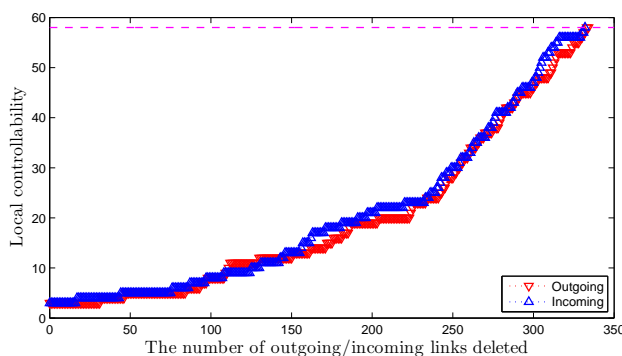
For the scale-free network with small degree exponent  $\gamma = 2.5$ , the hubs are very densely interconnected, and we found that the local controllability of  $S$  (the subset of top 100 hubs) is robust to the removal of both outgoing links and incoming links: only after a large number of outgoing/incoming links have been deleted can the local controllability of  $S$  start to increase (see Figure 4).

For the scale-free network with large degree exponent  $\gamma = 4$  and the ER random network, the local controllability  $lc(G_0, S)$  varies greatly with the network controllability  $N_D(G_0[S])$ , and the removal of outgoing links or incoming

links can increase the local controllability of  $S$  steadily, especially in the later steps (see Figures 5 and 6). Moreover, in these networks, we can see that local controllability is less robust to incoming link removal: the removal of a few incoming links seems to have a greater impact on the local controllability of  $S$  than the removal of the same number of outgoing links.



**Figure 5.** The plot of local controllability versus the number of outgoing/incoming links deleted for a scale-free network  $G_0$  with  $n = 1000$ ,  $\langle k \rangle = 4$  and  $\gamma = 4$ .



**Figure 6.** The plot of local controllability versus the number of outgoing/incoming links deleted for an ER random network  $G_0$  with  $n = 1000$  and  $\langle k \rangle = 4$ .

Similar procedures are performed on both types of model networks under two schemes: random removal and hub attack of in-neighbors (out-neighbors) of  $S$ . We find that in both ER random networks and scale-free networks, hub attack can increase (damage) the local controllability

more effectively than random removal. Furthermore, local controllability is less robust to in-neighbor hub attack than out-neighbor hub attack, though the difference is not that obvious.

## 5. Concluding remarks

In this paper, we briefly review the mathematical foundation for network controllability and then give a formal definition of local controllability and present the theoretical results as well as numerical results. We prove that the local controllability is an mathematical optimization problem which can be solved by minimum-weight cycle cover or minimum-weight perfect matching. Based on minimum-weight perfect matching, we develop an cubic-time algorithm to calculate  $lc(G, S)$  and identify the minimum driver node set for controlling a subset of nodes  $S$  in a directed network  $G$ .

The framework of local controllability can be used to study local control of any meaningful subset of nodes in real networks. We apply it to mouse brain networks, and show that Isocortex and Medulla are significantly easier to control, while Striatum and Thalamus are significantly harder to control. We suggest that these differences might be related to the (anatomical) structural basis of the brain subdivision. In order to explore how  $lc(G, S)$  depends on choices of the subset  $S$  and the underlying network  $G$ , we performed numerical study on two types of model networks: ER random network and scale-free network. We observe that ER random networks and scale-free networks show both similar and different behaviors in terms of local controllability. First, in both ER and SF networks, nodes with higher degree/betweenness are easier to control and the hub attack can cause more damage to local controllability than random removal of out-neighbors (in-neighbors) of  $S$ . Second, it is easier (harder) to control nodes with the highest (lowest) degrees in SF networks than in ER networks and local controllability is more robust to the removal of outgoing and incoming links of  $S$  in scale-free networks than in ER random networks. These differences could be explained by the different structures of ER random networks and scale-free networks: ER random networks are homogeneous, while scale-free networks are

heterogeneous. As the degree exponent  $\gamma$  becomes larger, scale-free networks become less heterogeneous, and their behaviors become more similar to those of ER random networks.

There are still limitations for our model that could provide avenues for future study. First, local controllability is defined on directed networks, and how to locally control a subset of nodes in undirected (and weighted) networks is still unknown. Second, whether the framework of local controllability can provide efficient and realistic local control in real networks requires examination by experiment, and it may incorporate more control characteristic into the study: control energy, control trajectory, the non-linear dynamics and so on.

### Conflict of interest

The author declares there is no conflicts of interest in this paper.

### References

1. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D. Hwang, Complex networks: structure and dynamics, *Physics Reports*, **424** (2006), 175–308.
2. M. Newman, The structure and function of complex networks, *SIAM Rev.*, **45** (2003), 167–256.
3. A. Clauset, M. Newman, C. Moore, Finding community structure in very large networks, *Phys. Rev. E*, **70** (2004), 066111.
4. L. Freeman, Centrality in social networks conceptual clarification, *Social Networks*, **1** (1979), 215–239.
5. R. Milo, S. Shen-Orr, Kashtan, D. Chklovskii, U. Alon, Network motifs: simple building blocks of complex networks, *Science*, **298** (2002), 824–827.
6. A. L. Barabási, R. Albert, Emergence of scaling in random networks, *Science*, **286** (1999), 509–512.
7. D. Watts, S. Strogatz, Collective dynamics of ‘small-world’ networks, *Nature*, **393** (1998), 440–442.
8. R. Pastor-Satorras, A. Vespignani, Epidemic spreading in scale-free networks, *Phys. Rev. Lett.*, **86** (2001), 3200–3203.
9. P. Oikonomou, P. Cluzel, Effects of topology on network evolution, *Nat. Phys.*, **2** (2006), 532–536.
10. D. Callaway, M. Newman, S. Strogatz, D. Watts, Network robustness and fragility: percolation on random graphs, *Phys. Rev. Lett.*, **85** (2000), 5468–5471.
11. F. Sorrentino, M. Bernardo, F. Garofalo, G. Chen, Controllability of complex networks via pinning, *Phys. Rev. E*, **75** (2007), 046103.
12. D. Luenberger, *Introduction to Dynamical Systems: Theory, Models and Applications*, 1979.
13. Y. Liu, J. Slotine, A. Barabási, Controllability of complex networks, *Nature*, **473** (2011), 167–173.
14. T. Nepusz, T. Vicsek, Controlling edge dynamics in complex networks, *Nat. Phys.*, **8** (2012), 568–573.
15. Z. Yuan, C. Zhao, Z. Di, W. Wang, Y. Lai, Exact controllability of complex networks, *Nat. Commun.*, **4** (2013), 1–9.
16. S. Hosoe, Determination of generic dimensions of controllable subspaces and its application, *IEEE T. Automat. Contr.*, **25** (1980), 1192–1196.
17. Y. Liu, J. Slotine, A. Barabási, Control centrality and hierarchical structure in complex networks, *PLoS One*, **7** (2012), e44459.
18. J. Gao, Y. Liu, R. D’Souza, A. Barabási, Target control of complex networks, *Nat. Commun.*, **5** (2014), 1–8.
19. Y. Liu, A. Barabási, Control principles of complex systems, *Rev. Mod. Phys.*, **88** (2016), 035006.
20. R. Kalman, Mathematical description of linear dynamical systems, *J. Soc. Indus. Appl. Math. Ser. A*, **1** (1963), 152–192.
21. C. Lin, Structural controllability, *IEEE T. Automat. Contr.*, **19** (1974), 201–208.
22. K. Glover, L. Silverman, Characterization of structural controllability, *IEEE T. Automat. Contr.*, **21** (1976), 534–537.
23. R. Shields, J. Pearson, Structural controllability of multi-input linear systems, *IEEE T. Automat. Contr.*, **21** (1976), 203–212.

24. J. Dion, C. Commault, J. van der Woude, Generic properties and control of linear structured systems: a survey, *Automatica*, **39** (2003), 1125–1144.
25. H. Mayeda, On structural controllability theorem, *IEEE T. Automat. Contr.*, **26** (1981), 795–798.
26. N. Ahn, *Spanning Cacti for Structurally Controllable Networks*, master thesis, National University of Singapore, 2012.
27. J. Hopcroft, R. Karp, An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.*, **2** (1973), 225–231.
28. P. Csermely, T. Korcsmaros, H. Kiss, G. London, R. Nussinov, Structure and dynamics of molecular networks: a novel paradigm of drug discovery, a comprehensive review, *Pharmacology and Therapeutics*, **138** (2013), 333–408.
29. H. Kuhn, The Hungarian method for the assignment problem, *Naval Research Logistics Quarterly*, **2** (1955), 83–97.
30. S. Oh, J. Harris, et al, A mesoscale connectome of the mouse brain, *Nature*, **508** (2014), 207–214.
31. M. Kirkcaldie, *The Mouse Nervous System*, (2012), 52–111.
32. L. Puelles, M. Martinez-De-La-Torre, J. Ferran, C. Watson, *The Mouse Nervous System*, (2012), 313–336.
33. P. Erdős, A. Rényi, On the evolution of random graphs, *Publications of the Mathematical Institute of the Hungarian Academy of Science*, **5** (1960), 17–61.
34. K. Goh, B. Kahng, D. Kim, Universal behavior of load distribution in scale-free networks, *Phys. Rev. Lett.*, **87**, (2001), 278701.

## Appendix

In the appendix, we show the proof for Theorem 3.1 and the main steps of cost matrix-based Hungarian method.

### Proof for Theorem 3.1:

The proof is due to the correspondence between the set of path-cycle covers of  $S$  in  $G$  and the set of cycle covers of  $S$  in  $G^*$ .

First, define a *path-cycle cover* of  $S$  in  $G$  to be a set of disjoint paths and cycles in  $G$  that contain  $S$ :  $C_G(S) = P_1 \cup \dots \cup P_l \cup C_1 \cup \dots \cup C_k$  is a path-cycle cover of  $S$  in  $G$ , if  $P_1, \dots, P_l$  are directed paths in  $G$ ,  $C_1, \dots, C_k$  are directed cycles in  $G$  (all the paths and cycles are disjoint), and each node in  $S$  is contained in  $C_G(S)$ .

For each path-cycle cover of  $S$  in  $G$ , denoted by  $C_G(S) = P_1 \cup \dots \cup P_l \cup C_1 \cup \dots \cup C_k$ , it corresponds to a cycle cover of  $S$  in  $G^*$ , denoted by  $C_{G^*}(S)$ , by adding a directed edge from the last node to the first node in each directed path  $P_j$ ,  $j = 1, \dots, l$ . After the edge addition, each path thus becomes a (closed) cycle. Note that, in the resulting cycle cover  $C_{G^*}(S)$ , each cycle contains at most one directed edge that is not in  $G$ .

Conversely, for each cycle cover of  $S$  in  $G^*$ , denoted by  $C_{G^*}(S)$ , it corresponds to a path-cycle cover of  $S$  in  $G$ , denoted by  $C_G(S)$ , by removing from  $C_{G^*}(S)$  all the directed edges that are not contained in  $G$ . After the edge removal, some cycles are broken into paths.

We know that  $lc(G, S)$  (the minimum number of inputs required to control  $S$  in  $G$ ) is given by the minimum number of paths in a path-cycle cover of  $S$  in  $G$ , since each independent path requires a different input to control and each independent cycle requires no additional input, according to the minimum input theorem (Theorem 2.2). Note that, in the case that there is actually a cycle cover of  $S$  in  $G$  (a cycle cover is also a path-cycle cover, without any path), then one input is sufficient for control, and  $lc(G, S) = 1$ .

Consider a path-cycle cover of  $S$  in  $G$ ,  $C_G(S) = P_1 \cup \dots \cup P_l \cup C_1 \cup \dots \cup C_k$ , where  $l \neq 0$ . Furthermore, assume that  $C_G(S)$  has the minimum number of paths among all path-cycle covers of  $S$  in  $G$ . Then, the corresponding cycle cover of  $S$  in  $G^*$ ,  $C_{G^*}(S)$ , has weight  $w(C_{G^*}(S)) = l \cdot n + q$ , where  $q$  is the number of edges in  $C_G(S)$  and  $q < n$ . Thus,  $lc(G, S) = l = \lfloor \frac{w(C_{G^*}(S))}{n} \rfloor$ .

Therefore, if  $\bar{C} = \bar{C}_{G^*}(S)$  is a minimum-weight cycle cover of  $S$  in  $G^*$ , it corresponds to a path-cycle cover of  $S$  in  $G$ ,  $\bar{C}_G(S)$ , which should contain exactly  $l$  paths. Otherwise,  $\bar{C}_G(S)$  would contain more than  $l$  paths, and consequently,  $\bar{C}_{G^*}(S)$  would have weight at least  $(l + 1) \cdot n$ . This is a contradiction with the assumption that  $\bar{C} = \bar{C}_{G^*}(S)$  is a minimum-weight cycle cover of  $S$  in  $G^*$ .

These arguments show that if  $l \neq 0$ , then the local controllability  $lc(G, S)$  is given by

$$lc(G, S) = l = \left\lfloor \frac{w(\bar{C})}{n} \right\rfloor.$$

Lastly, note that  $lc(G, S)$  is at least one, thus, the proof is complete.

#### The Hungarian method based on cost matrix:

**Step 1:** For each row of the cost matrix, find the smallest element in this row, and subtract it from every element in this row.

**Step 2:** Find a zero element (denoted by  $Z$ ) in the resulting matrix. If there is no starred zero in its row or column, star  $Z$  (by starring  $Z$ , we mean labelling  $Z$  by a star \*, and  $Z$  is called ‘starred’). Repeat for each zero element in the matrix.

**Step 3:** Cover each column that contains a starred zero (by covering a column, we mean covering the elements in the column by a straight line, and the column together with the elements in the column are called ‘covered’, a column that is not covered is called ‘uncovered’). If  $n$  columns are covered, the starred zeros describe an optimal assignment, stop and output the assignment indicated by a permutation  $\sigma$  defined as follows: denote the set of  $n$  starred zero entries by  $\{c_{1k_1}, \dots, c_{nk_n}\}$ , then  $\sigma(i) = k_i$ , for  $i = 1, \dots, n$ . If less than  $n$  columns are covered, go to Step 4.

**Step 4:** Find an uncovered zero (a zero entry that is not covered by a straight line) and underline it (label it by an underline, and it is called ‘underlined’), If there is no starred zero in the row containing this underlined zero, go to Step 5. Otherwise, cover this row and uncover the column containing the starred zero which is in the row of this underlined zero (by uncovering the column, we mean removing the straight line that covers the column). Continue in this manner until there are no uncovered zeros left. Save the smallest uncovered value and go to Step 6.

**Step 5:** Construct a series of alternating underlined and starred zeros as follows. Let  $Z_0$  denote the uncovered underlined zero found in Step 4. Let  $Z_1$  denote the starred zero in the column of  $Z_0$  (if any). Let  $Z_2$  denote the underlined zero in the row of  $Z_1$  (there will always be one). Continue until the series terminates at an underlined zero that has no starred zero in its column. Unstar each starred zero of the series (by unstarring a starred zero, we

mean removing the star \* that labels this starred zero), star each underlined zero of the series, erase all underlines and uncover every line (a line is a row or a column) in the matrix. Return to Step 3.

**Step 6:** Add the value found in Step 4 to every element of each covered row, and subtract it from every element of each uncovered column. Return to Step 4 without altering any stars, underlines, or covered lines.



AIMS Press

© 2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)