Automatica 107 (2019) 68–76

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Block-based minimum input design for the structural controllability of complex networks ☆

Ting Bai, Shaoyuan Li *, Yuanyuan Zou, Xiang Yin

Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 30 November 2018
Received in revised form 4 December 2018
Accepted 18 April 2019
Available online xxxx

Keywords:
Complex networks
Minimum input design
Structural controllability
Block-based algorithm

ABSTRACT

Traditional control system design to complex networks is generally implemented by integrated structural analysis aiming at a global network. However, such a global method may be inefficient, in particular, when a massive network with a huge number of nodes and associations is considered. In this paper, motivated by the idea of "dividing and dealing", we propose a block-based approach to the issue of minimum input design for structural controllability (MIDSC) of complex networks that potentially incurs in higher efficiency. Specifically, we consider a large-scale networked system that consists of several local blocks. The main challenge for control configuration design of this class of systems is how to find the minimum inputs of global network according to the local block information while maintaining system's structural controllability. To this end, two block-based graphical algorithms are developed to meet the conditions required for achieving structural controllability, and meanwhile determine an optimal solution for addressing the MIDSC problem. The complexity of the proposed method is analyzed, which is also compared with existing algorithms designed mainly based on monolithic model. In particular, we show that, under some mild conditions on blocking structure, the complexity of the proposed algorithm is strictly lower than that of existing algorithms to the MIDSC problem.

1. Introduction

Complex networks arise in a great deal of natural and man-made systems. The instances in nature involve metabolic network, biological immune network, and brain's neural network, while electrical power grids, multi-agent robotic system and mass transportation network are examples of modern science and technology, to name only a few. Owing to a wide applying foreground, a tremendous amount of studies on complex networks have been prompted. In the application of complexity networks (Kivelä et al., 2014; Maza, Simon, & Boukhobza, 2012; Pan & Li, 2014; Wu, Li, Wang, & Wu, 2016), one of the key questions is to have full control over the network, i.e., the capability of steering the network behavior to an expected state in finite time. Such a capability is referred to as *network controllability* in system theory, and it is of fundamental importance to ensure dependable and effective network functionalities. As a primary property, indeed, network controllability can be analyzed by standard algebraic methods based on the well-known Kalman's rank criterion (Chui & Chen, 2012). Notwithstanding, the precise numerical parameters of network models are sometimes difficult to be obtained. Instead, it is more practical to know whether there exists a link between different states. In this situation, a natural direction for network analysis is to turn to structural system theory and pursue *structural controllability*. As first posed by Lin in the seminal work (Lin, 1974), a complex network is said to be structurally controllable if an array of numerical values can be found for those unknown parameters such that the resulting system is controllable in classic sense (Dion, Commault, & Van Der Woude, 2003). Considering the economic restrictions, i.e., since more actuation incurs in higher control cost, the issue pertaining to minimum input design for structural controllability (MIDSC) of complex networks has drawn extensive interest, which is formally stated as: Identify the smallest subset of actuated states (so-called driver nodes) to manipulate each with an external input so that the resulting network is structurally controllable. It is worth mentioning that, there exist two cases in the MIDSC problem. In one

☆ This paper was supported by the National Natural Science Foundation of China under Grants (61833012, 61590924, and 61773162). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Delin Chu under the direction of Editor Ian R. Petersen..

* Corresponding author.
E-mail addresses: number127578@sjtu.edu.cn (T. Bai), syli@sjtu.edu.cn (S. Li), yuanyzou@sjtu.edu.cn (Y. Zou), yinxiang@sjtu.edu.cn (X. Yin).

https://doi.org/10.1016/j.automatica.2019.05.006

case, each input is connected to multiple system states; while in another case, each control input drives at most a single system state. In this paper, we concentrate on the minimum input design problem for the second case, which is already known to be in polynomial-time (Assadi, Khanna, Li, & Preciado, 2015).

Although great advances have been obtained for solving the MIDSC problem (see references in Liu, Slotine, & Barabási, 2011, Yuan, Zhao, Di, Wang, & Lai, 2013, Pequito, Kar, & Aguiar, 2016, Yin & Zhang, 2016, Carvalho, Pequito, Aguiar, Kar, & Johansson, 2017), including algebraic and graphical techniques, a high-efficiency way that adapts to massive complex networks is still lacking, in particular, for a class of large-scale systems with several local blocks. Lately, it has shown growing interest in this direction and a flurry of papers have concentrated on various ways to improve the algorithm efficiency in addressing the MIDSC problem. In Pequito, Kar, and Aguiar (2013), Pequito et al. first identified the smallest subset of driver nodes for structural controllability in $O(\boldsymbol{m}\boldsymbol{n}^{1.5})$ operations, where $\boldsymbol{n}$ denotes the number of network states and $\boldsymbol{m}$ is the number of nonzero entries of the system state matrix. Shorter after, Olshevsky presented a faster algorithm to the MIDSC problem in Olshevsky (2015) with the total running time $O(\boldsymbol{m}\sqrt{\boldsymbol{n}})$, which is obtained by solving a maximum bipartite matching (MBM) problem in the first place, and then, transforming the found matching into a solution of the MIDSC in the second augmentation stage. Each of the two phases requires $O(\boldsymbol{m}\sqrt{\boldsymbol{n}})$ operations. Furthermore, this result was promoted by Assadi et al. (2015), where it has been proved that the MIDSC issue is computationally equivalent to addressing a MBM problem. In this sense, the efficiency improvement for addressing the MIDSC problem can benefit from all the breakthroughs in MBM computation, such as the well-known Hopcroft–Karp algorithm (Hopcroft & Karp, 1973), Mucha–Sankowski algorithm and Madry's algorithm (Madry, 2013). Therefore, an alternative approach was developed with the time complexity $\min\{O(\boldsymbol{m}\sqrt{\boldsymbol{n}}), \tilde{O}(\boldsymbol{n}^{2.37}), \tilde{O}(\boldsymbol{m}^{10/7})\}$.

Noticeably, as most existing algorithms depend on a monolithic system model, it will inevitably put a high computational burden on massive networks with a huge number of nodes and associations. Motivated by the idea of "dividing and dealing", in this paper, we intend to address the MIDSC problem under a block-based framework for improving the computation efficiency of traditional methods. Specifically, we consider a large-scale networked system consisting of several local blocks. The modular structure nature allows for dimension reduction and parallel processing in the analysis of this class of systems. In such a blocking setting, the main challenge for control configuration design is how to find the minimum input deployment of global network by effectively synthesizing the block information for achieving system's structural controllability.

Using typical graphical tools (Aguilar & Gharesifard, 2015; Liu & Barabási, 2016), we map the MIDSC problem of networked systems with blocking structures into local independent analysis and global integrative optimization. On this basis, a block-based approach to address the MIDSC issue is proposed for the first time. The major contributions of this paper are threefold: (i) two block-based graphical algorithms are proposed to determine a minimum input deployment of global network while maintaining system's structural controllability; (ii) the time complexity of the developed method is analyzed, which is also compared with the fastest (up-to-date) algorithm that designed based on a monolithic system model to demonstrate the higher efficiency of the proposed algorithm; (iii) the validity of the presented method is substantiated by the minimum input design problem of an example with blocking system structure. Our main results show that, under some mild conditions on blocking structure, the complexity of the proposed method is strictly lower than that of existing algorithms.

2. Problem statement

Notation: Let $\mathbb{R}$ denote the real numbers, $\mathbb{R}^n$ the vector space of real n-vectors and $\mathbb{R}^{n\times m}$ the set of $n\times m$ real matrices. $\mathbb{N}^+$ denotes the set of positive integers and $[1:\boldsymbol{n}]$ represents a set of ordered integers $\{1,2,\ldots,\boldsymbol{n}\}$ with $\boldsymbol{n}\in\mathbb{N}^+$. The cardinality of a set $\mathcal{C}$ is denoted by $|\mathcal{C}|$. Given a set $\mathscr{D}\subseteq[1:\boldsymbol{n}]$ and a $\boldsymbol{n}\times\boldsymbol{n}$ identity matrix I, $I_{\mathscr{D}}$ indicates a matrix constructed by the columns of I indexed by $\mathscr{D}$. $\mathbf{0}$ is the zero matrix with appropriate dimension. The zero (quasi) norm is denoted by $\|R\|_0$, which equals to the number of nonzero entries of matrix R.

Throughout this context, we concentrate on a large-scale discrete time, linear and time-invariant directed complex network with autonomous dynamics

$$\Sigma : \boldsymbol{x}(k+1) = A\boldsymbol{x}(k), \tag{1}$$

where $\boldsymbol{x}\in\mathbb{R}^{\boldsymbol{n}}$ denotes global network state. $A\in\mathbb{R}^{\boldsymbol{n}\times\boldsymbol{n}}$ is network state matrix and it is assumed to be a sparsity pattern with $\boldsymbol{m}$ nonzero entries. As a rule, we make the standard assumption $\boldsymbol{m}\geq\boldsymbol{n}$. Now consider a network consisting of r blocks and the time evolution of every local block follows the form of

$$\Sigma_i : \boldsymbol{x}_i(k+1) = \sum_{j=1}^{r} A_{ij}\boldsymbol{x}_j(k), \tag{2}$$

where $\boldsymbol{x}_i\in\mathbb{R}^{\boldsymbol{n}_i}$ is the state of block Σ_i. $A_{ii}\in\mathbb{R}^{\boldsymbol{n}_i\times\boldsymbol{n}_i}$ represents the block state matrix while $\{A_{ij}\}_{i\neq j}$ are the interconnected matrices between different blocks. Constructed by the blocking structure, $\boldsymbol{x}=[\boldsymbol{x}_1^T,\ldots,\boldsymbol{x}_r^T]^T$ with $\boldsymbol{n}=\sum_{i=1}^r\boldsymbol{n}_i$ and $\boldsymbol{m}=\sum_{i=1}^r\sum_{j=1}^r\|A_{ij}\|_0$. Meanwhile, the global network matrix A can be characterized as $[A]_{ij}=A_{ij}$, where $[A]_{ij}$ denotes the matrix block located in the row i column j of matrix A. In structural representation, we use a binary matrix $\bar{A}_{ij}\in\{0,1\}^{\boldsymbol{n}_i\times\boldsymbol{n}_j}$ to encode the structural pattern of A_{ij} by assigning 0 to each zero entry of A_{ij} and 1 otherwise. To well depict associated relationships, the neighbor of every block is brought in.

Definition 1 (*Incoming/Outgoing Neighbor*)**.** The block Σ_j satisfying $\bar{A}_{ij}\neq\mathbf{0}$, $i,j\in[1:r]$ and $j\neq i$ is said to be an incoming neighbor of Σ_i. Accordingly, those Σ_j achieve $\bar{A}_{ji}\neq\mathbf{0}$, $i,j\in[1:r]$ and $j\neq i$ are known as outgoing neighbors of block Σ_i. The set of incoming and outgoing neighbors are denoted by $\mathcal{I}_i$ and $\mathcal{O}_i$, respectively.

In this context, we propose to address the MIDSC problem under a block-based structure setting, which is formally stated as: Given the structural block matrix $\bar{A}_{ii}\in\{0,1\}^{\boldsymbol{n}_i\times\boldsymbol{n}_i}$, the associated structural matrices $\{\bar{A}_{ij}\}_{j\neq i}\in\{0,1\}^{\boldsymbol{n}_i\times\boldsymbol{n}_j}$ and $\{\bar{A}_{ji}\}_{j\neq i}\in\{0,1\}^{\boldsymbol{n}_j\times\boldsymbol{n}_i}$ with $i,j\in[1:r]$, determine the set $\mathscr{D}_i\subseteq[1:\boldsymbol{n}_i]$ of the following optimization problem:

$$\min_{\mathscr{D}_i\subseteq[1:\boldsymbol{n}_i]} \sum_{i=1}^{r}|\mathscr{D}_i| \tag{3}$$
$$s.t.\ (\bar{A}, \mathrm{diag}(I_{\mathscr{D}_1},\ldots,I_{\mathscr{D}_r}))\ \text{is structurally controllable,}$$

where $\mathscr{D}_i$ denotes the index subset of states in block Σ_i. More precisely, for any $l\in\mathscr{D}_i$, it indicates that the state x_i^l of $\boldsymbol{x}_i$ is selected to be driven directly by an dedicated input (i.e., each input is connected to at most a single state), which corresponds to the lth column of the $\boldsymbol{n}_i\times\boldsymbol{n}_i$ matrix I. Meanwhile, $\mathrm{diag}(I_{\mathscr{D}_1},\ldots,I_{\mathscr{D}_r})\in\mathbb{R}^{\boldsymbol{n}\times\boldsymbol{p}}$ and $\boldsymbol{p}=\sum_{i=1}^r|\mathscr{D}_i|$. Indexed by $\mathscr{D}_i$, $\bar{B}_i=I_{\mathscr{D}_i}$, and the optimization (3) is equivalent to minimizing $\sum_{i=1}^r\|\bar{B}_i\|_0$. This is noted, the optimal $\{\mathscr{D}_i\}_{\forall i\in[1:r]}$ exactly constitutes the desired sparsest input matrix $\bar{B}$ of the control network

$$\Sigma_\Lambda : \boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k), \tag{4}$$

where $\boldsymbol{u} \in \mathbb{R}^{\boldsymbol{p}}$ is the input of global network. By definition, the network (4) represented by the pair (A, B) is said to be structurally controllable if there exists at least one controllable network (A, B) that has the same structured pattern with $(\bar{A}, \bar{B})$. Generally speaking, a network is controllable for almost all possible parameter realizations if it is structurally controllable (Dion et al., 2003).

3. Preliminaries and terminology

This section recalls some standard graph theoretic notions and key results used in the analysis of structural networks. The references can be found, for instance, in Liu and Morse (2017), Pequito et al. (2016) and Sundaram and Hadjicostis (2013).

It is customary to associate the network (4) with a *digraph* $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ for structural analysis, where $\mathcal{V}$ denotes the vertex-set and $\mathcal{E}$ is the edge-set so that an edge (v_i, v_j) is directed from vertex v_i to v_j. To this end, denote by $\mathcal{X} = \{x_1, \ldots, x_{\boldsymbol{n}}\}$ and $\mathcal{U} = \{u_1, \ldots, u_{\boldsymbol{p}}\}$ the set of state and input vertices, respectively. Meanwhile, denote by $\mathcal{E}_{\mathcal{X},\mathcal{X}} = \{(x_i, x_j) : \bar{a}_{ji} \neq 0\}$ and $\mathcal{E}_{\mathcal{U},\mathcal{X}} = \{(u_j, x_i) : \bar{b}_{ij} \neq 0\}$ the set of edges, where $\bar{a}_{ji}$ is the element of row j column i of $\bar{A}$ and $\bar{b}_{ij}$ is the element of row i column j of $\bar{B}$. Then we may introduce the *state digraph* $\mathcal{D}(\bar{A}) = (\mathcal{X}, \mathcal{E}_{\mathcal{X},\mathcal{X}})$ and *network digraph* $\mathcal{D}(\bar{A}, \bar{B}) = (\mathcal{X}\cup\mathcal{U}, \mathcal{E}_{\mathcal{X},\mathcal{X}} \cup \mathcal{E}_{\mathcal{U},\mathcal{X}})$. A *directed path* is defined by a sequence of edges $\{(v_1, v_2), (v_2, v_3), \ldots, (v_{k-1}, v_k)\}$. If all the vertices involved in a directed path are distinct, it is also known as an *elementary path*. In $\mathcal{D}(\bar{A}, \bar{B})$, a state vertex is said to be *input-reachable* if there exists a directed path from an input vertex to it. An *outgoing edge* from a vertex v_i is an edge starts in v_i while an *incoming edge* to v_i is an edge ends on v_i. A vertex with an edge to itself (i.e., a self-loop), or an elementary path from v_1 to v_k, together with an edge (v_k, v_1) is denoted by a *cycle*.

A *subgraph* of $\mathcal{D}$ is a digraph $\mathcal{D}_s = (\mathcal{V}_s, \mathcal{E}_s)$ with $\mathcal{V}_s \subset \mathcal{V}$ and $\mathcal{E}_s \subset \mathcal{E}$. $\mathcal{D}$ is said to be *strongly connected* if there exists a directed path between any two vertices of $\mathcal{V}$. A *strongly connected component* (SCC) is a maximal subgraph $\mathcal{D}_s = (\mathcal{V}_s, \mathcal{E}_s)$ of $\mathcal{D}$ such that for any two vertices $v_i, v_j \in \mathcal{V}_s$, there exist paths from v_i to v_j and also from v_j to v_i. An SCC that has no incoming edge to any of its vertices from another SCC is referred to as a *non-top linked SCC* (NT-SCC). Given any $\mathcal{D}$ and vertex-sets $\mathcal{V}_1, \mathcal{V}_2 \subset \mathcal{V}$, the *bipartite graph* is defined by $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, whose vertex-set is $\mathcal{V}_1 \cup \mathcal{V}_2$ and edge-set is $\mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2} \subset \{(v_1, v_2) : v_1 \in \mathcal{V}_1, v_2 \in \mathcal{V}_2\}$. $\mathcal{B}(\mathcal{V}, \mathcal{V}, \mathcal{E})$ is known as the bipartite graph associated with $\mathcal{D}(\mathcal{V}, \mathcal{E})$. In the sequel, we make use of *state bipartite graph* $\mathcal{B}(\bar{A}) = (\mathcal{X}, \mathcal{X}, \mathcal{E}_{\mathcal{X},\mathcal{X}})$ and *network bipartite graph* $\mathcal{B}(\bar{A}, \bar{B}) = \mathcal{B}(\mathcal{U}\cup\mathcal{X}, \mathcal{X}, \mathcal{E}_{\mathcal{X},\mathcal{X}} \cup \mathcal{E}_{\mathcal{U},\mathcal{X}})$.

In $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, a *matching* M is a subset of edges in $\mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2}$ so that no two edges share a common vertex, i.e., given edges $e = (v_1, v_2)$, $e' = (v_1', v_2')$ with $v_1, v_1' \in \mathcal{V}_1$ and $v_2, v_2' \in \mathcal{V}_2$, then $e, e' \in M$ only if $v_1 \neq v_1'$ and $v_2 \neq v_2'$. A *maximum matching* M^* is a matching M including the maximum number of edges among all possible matchings, which may not be unique. Accordingly, the *right (or left)-unmatched vertex*, with respect to a bipartite graph $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$ and a matching M, not necessarily maximum, will refer to the vertex in $\mathcal{V}_2$ (or $\mathcal{V}_1$) that does not belong to a matching edge in M. Otherwise, it is known as a *right (or left)-matched vertex.*

Lemma 1 (*Lin, 1974; Pequito et al., 2016*). *For linear systems described by (4), the following statements are equivalent:*

(1) *the pair $(\bar{A}, \bar{B})$ is structurally controllable;*
(2*a*) *every state vertex of the system is input-reachable,*
(2*b*) *and the generic rank of $[\bar{A}, \bar{B}] = \boldsymbol{n}$;*
(3*a*) *the NT-SCCs of $\mathcal{D}(\bar{A}, \bar{B})$ are comprised of inputs,*
(3*b*) *and there exists a matching of $\mathcal{B}(\bar{A}, \bar{B})$ without right-unmatched vertex.*

Lemma 1 reveals following two things: (i) to achieve the input-reachability condition, one needs to seek all the NT-SCCs of $\mathcal{D}(\bar{A})$ and allocate at least one input vertex to each NT-SCC; (ii) the generic rank condition for structural controllability is equivalent to finding a maximum matching of $\mathcal{B}(\bar{A}, \bar{B})$ with size $\boldsymbol{n}$. A minimum input design meeting both the conditions will render a structurally controllable network. Taking these results as a basis, hereafter, we propose to address the MIDSC problem in a blocking framework for higher algorithm efficiency.

4. Main results

This section introduces the two block-based algorithms for addressing the MIDSC problem. First, a graphical description to local blocks is provided for clear algorithm representation. Next, to meet the input-reachability and generic rank condition of structural controllability, two block-based graphical algorithms are presented, respectively, to find the NT-SCCs and maximum matching of global network, which are further employed to determine a minimum input deployment of the MIDSC issue.

4.1. Graph-based description of local blocks

In view of the inner structure of every block Σ_i, we use $\mathcal{D}(\bar{A}_{ii}) = (\mathcal{X}_i, \mathcal{E}_{\mathcal{X}_i,\mathcal{X}_i})$ to denote the *block state digraph*, where $\mathcal{X}_i = \{x_i^1, \ldots, x_i^{\boldsymbol{n}_i}\}$ is the local state-set and $\mathcal{E}_{\mathcal{X}_i,\mathcal{X}_i} = \{(x_i^l, x_i^m) : \bar{a}^i_{ml} \neq 0,\ \forall l, m \in [1 : \boldsymbol{n}_i]\}$ is the edge-set. $\bar{a}^i_{ml}$ represents the element in $\bar{A}_{ii}$ of row m column l. Associated with $\mathcal{D}(\bar{A}_{ii})$, the *block state bipartite graph* is denoted by $\mathcal{B}(\bar{A}_{ii}) = \mathcal{B}(\mathcal{X}_i, \mathcal{X}_i, \mathcal{E}_{\mathcal{X}_i,\mathcal{X}_i})$. From a local point of view, the states in $\mathcal{X}_i$ are distinguished by various associations between different blocks, which fall into the following three categories: incoming local state, outgoing local state and absolute local state. Specifically, we refer to the state of $\mathcal{X}_i$ that has at least one incoming edge from any state of $\{\mathcal{X}_j\}_{j\in\mathcal{I}_i}$ as an *incoming local state* of block Σ_i. Correspondingly, that of $\mathcal{X}_i$ with at least one outgoing edge to $\{\mathcal{X}_j\}_{j\in\mathcal{O}_i}$ is known as an *outgoing local state*. The rest states of $\mathcal{X}_i$ except for incoming and outgoing ones are called *absolute local state*. The three kinds of local state-sets are described by

$$\mathcal{X}^{in}_{\mathcal{L}_i} = \{x_i^m : \bar{\boldsymbol{a}}^{ij}_{m:} \neq \boldsymbol{0}, \forall m \in [1 : \boldsymbol{n}_i]\} \tag{5a}$$

$$\mathcal{X}^{out}_{\mathcal{L}_i} = \{x_i^l : \bar{\boldsymbol{a}}^{ji}_{:l} \neq \boldsymbol{0}, \forall l \in [1 : \boldsymbol{n}_i]\} \tag{5b}$$

$$\mathcal{X}^{abs}_{\mathcal{L}_i} = \mathcal{X}_i \setminus \{\mathcal{X}^{in}_{\mathcal{L}_i} \cup \mathcal{X}^{out}_{\mathcal{L}_i}\}, \tag{5c}$$

where $\bar{\boldsymbol{a}}^{ij}_{m:}$ denotes the row m of $\bar{A}_{ij}$ while $\bar{\boldsymbol{a}}^{ji}_{:l}$ is the column l of $\bar{A}_{ji}$. Meanwhile, to depict the associated relationships, the *incoming neighborhood state* of block Σ_i is used to denote the states of block $\{\Sigma_j\}_{j\neq i}$ that connects to at least one state of $\mathcal{X}_i$ through incoming edges of Σ_i. Accordingly, that of $\{\Sigma_j\}_{j\neq i}$ connecting by a state of $\mathcal{X}_i$ through outgoing edges of block Σ_i is known as the *outgoing neighborhood state* of block Σ_i. The incoming and outgoing neighborhood state-sets are denoted by

$$\mathcal{X}^{in}_{\mathcal{N}_i} = \{x_j^l : \bar{\boldsymbol{a}}^{ij}_{:l} \neq \boldsymbol{0}, \forall l \in [1 : \boldsymbol{n}_j]\} \tag{6a}$$

$$\mathcal{X}^{out}_{\mathcal{N}_i} = \{x_j^m : \bar{\boldsymbol{a}}^{ji}_{m:} \neq \boldsymbol{0}, \forall m \in [1 : \boldsymbol{n}_j]\}. \tag{6b}$$

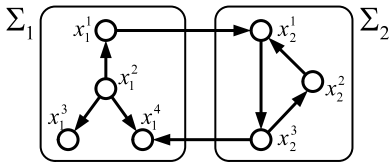


Fig. 1. A network digraph consisting of two local blocks, where the associations are denoted by directed edges.

To make things more concrete, an example is given to illustrate the above state-sets. As shown in Fig. 1, there is $\mathcal{X}_{\mathcal{L}_1}^{in} = \{x_1^4\}$, $\mathcal{X}_{\mathcal{L}_1}^{out} = \{x_1^1\}$, $\mathcal{X}_{\mathcal{L}_1}^{abs} = \{x_1^2, x_1^3\}$, $\mathcal{X}_{\mathcal{N}_1}^{in} = \{x_2^3\}$, $\mathcal{X}_{\mathcal{N}_1}^{out} = \{x_2^1\}$.

On this basis, the *incoming associated digraph* $\mathcal{D}(\bar{A}_{ii}, \bar{A}_{ij})$ and *outgoing associated digraph* $\mathcal{D}(\bar{A}_{ii}, \bar{A}_{ji})$ of block Σ_i can be, respectively, defined by

$$\mathcal{D}(\bar{A}_{ii}, \bar{A}_{ij}) = (\mathcal{X}_i \cup \mathcal{X}_{\mathcal{N}_i}^{in}, \mathcal{E}_{\mathcal{X}_{\mathcal{N}_i}^{in}, \mathcal{X}_i} \cup \mathcal{E}_{\mathcal{X}_i, \mathcal{X}_i}) \tag{7}$$

$$\mathcal{D}(\bar{A}_{ii}, \bar{A}_{ji}) = (\mathcal{X}_i \cup \mathcal{X}_{\mathcal{N}_i}^{out}, \mathcal{E}_{\mathcal{X}_i, \mathcal{X}_i} \cup \mathcal{E}_{\mathcal{X}_i, \mathcal{X}_{\mathcal{N}_i}^{out}}). \tag{8}$$

Definition 2 (*Absolute/Relative Correlation Degree*)**.** The cardinality of $\mathcal{X}_{\mathcal{L}_i}^{in}$ and $\mathcal{X}_{\mathcal{L}_i}^{out}$ are, respectively, referred to as the incoming and outgoing absolute correlation degree of block Σ_i, i.e., $\boldsymbol{d}_i^{in} = |\mathcal{X}_{\mathcal{L}_i}^{in}|$ and $\boldsymbol{d}_i^{out} = |\mathcal{X}_{\mathcal{L}_i}^{out}|$. Meanwhile, $|\mathcal{X}_{\mathcal{N}_i}^{in}|$ and $|\mathcal{X}_{\mathcal{N}_i}^{out}|$ are known as the incoming/outgoing relative correlation degree and denoted by $\boldsymbol{d}_{ri}^{in}$ and $\boldsymbol{d}_{ri}^{out}$. We denote by $\boldsymbol{d}_i = \boldsymbol{d}_i^{in} + \boldsymbol{d}_i^{out}$ and $\boldsymbol{d}_{ri} = \boldsymbol{d}_{ri}^{in} + \boldsymbol{d}_{ri}^{out}$ the absolute/relative correlation degree of Σ_i.

Remark 1. Note that the absolute correlation degree $\boldsymbol{d}_i$ reflects the association strength of block Σ_i to other interconnected blocks, whereas the relative correlation degree $\boldsymbol{d}_{ri}$ denotes the relevance of other associated blocks to Σ_i itself. The two correlation degrees provide an essential indicator in evaluating the blocking structure and play an important role in the following algorithm design for addressing the MIDSC problem.

4.2. Block-based algorithm to find NT-SCCs

As mentioned before, to meet the input reachability condition for structural controllability, the first important thing is to find all the NT-SCCs of global network. To this end, one of the difficulties for algorithm design in block-based framework is how to explore the SCCs separated by different local blocks, and another challenge is how to provide a good judgment on whether the identified SCC is a NT-SCC of global system. The following will detail how we overcome these hurdles.

Now consider that if there exist some states of block Σ_i that are conducive to forming a combined (i.e., a bigger) SCC together with the states of other blocks $\{\Sigma_j\}_{j\neq i}$, the involved local states must contain two kinds of nodes as follows: the *incoming local state* and *outgoing local state*, through which can Σ_i be connected with other blocks. In other words, a pair of nodes consisting of an incoming local state x_i^l and an outgoing local state x_i^m, together with a directed path from x_i^l to x_i^m, represents a way of connecting block Σ_i with other blocks. Formally, we denote by $(x_i^l, x_i^m)^{in}$ the *inward connected pair* of block Σ_i, where $x_i^l \in \mathcal{X}_{\mathcal{L}_i}^{in}$ and $x_i^m \in \mathcal{X}_{\mathcal{L}_i}^{out}$. It can be readily seen that, if an inward connected pair belongs to a combined SCC, all the local states on the directed paths from x_i^l to x_i^m should be involved in the same combined SCC. Taking into account the interconnections between different blocks, the *outward connected pair* of block Σ_i is denoted by $(x_i^m, x_j^l)^{out}$, where $x_i^m \in \mathcal{X}_{\mathcal{L}_i}^{out}$ and $x_j^l \in \mathcal{X}_{\mathcal{N}_i}^{out}$.

Algorithm 1. Block-based algorithm to find the NT-SCCs
Step1.1: Initialization
- find the local SCCs of $\mathcal{D}(\bar{A}_{ii})$, whose number is N_i;
- obtain the vertex-set $\mathcal{C}_{i,k_l}$ of local SCCs, $k_l \in [1 : N_i]$;
- determine the local state vertex-sets $\mathcal{X}_{\mathcal{L}_i}^{in}$ and $\mathcal{X}_{\mathcal{L}_i}^{out}$;
- explore all the inward and outward connected pairs $(x_i^l, x_i^m)^{in}$ and $(x_i^m, x_j^l)^{out}$ in $\mathcal{D}(\bar{A}_{ii}, \bar{A}_{ij})$ and $\mathcal{D}(\bar{A}_{ii}, \bar{A}_{ji})$ while maintaining x_i^l and x_i^m belong to different $\mathcal{C}_{i,k_l}$;

Step1.2: Identify the combined SCCs
- transfer $\{\mathcal{C}_{i,k_l}, (x_i^l, x_i^m)^{in}, (x_i^m, x_j^l)^{out}\}$ to coordination layer;
- obtain every combined SCC and its vertex-set $\mathcal{C}_{\mathcal{N}_{k_c}, k_c}$, $\mathcal{N}_{k_c}$ is the set of involved blocks Σ_j, i.e., $j \in \mathcal{N}_{k_c}$;
- transfer $\{\mathcal{C}_{\mathcal{N}_{k_c}, k_c}\}$ to every involved block Σ_j with $j \in \mathcal{N}_{k_c}$;

Step1.3: Mark up for every SCC
- select $(x_i^l, x_i^m)_{k_c}^{in}$ from $(x_i^l, x_i^m)^{in}$ with $x_i^l, x_i^m \in \mathcal{C}_{\mathcal{N}_{k_c}, k_c}$;
- find the biggest set $\mathcal{C}_{add, k_c}$ of states located on the directed paths from x_i^l to x_i^m so that x_i^l, x_i^m constitute $(x_i^l, x_i^m)_{k_c}^{in}$;
- obtain the marker of local SCCs, where the marker $\mathcal{M}[\mathcal{C}_{i,k_l}] = 0$ if there is no incoming edge to any state of $\mathcal{C}_{i,k_l}$, otherwise, $\mathcal{M}[\mathcal{C}_{i,k_l}] = 1$;
- obtain the marker of state x_i^k that involved in a combined SCC, where $x_i^k \in (\mathcal{C}_{\mathcal{N}_{k_c}, k_c} \cup \mathcal{C}_{add, k_c}) \cap \mathcal{X}_i$ and $\mathcal{M}[x_i^k] = 0$ if there is no incoming edge from a state out of $\mathcal{C}_{\mathcal{N}_{k_c}, k_c} \cup \mathcal{C}_{add, k_c}$ to x_i^k, otherwise, $\mathcal{M}[x_i^k] = 1$;

Step1.4: Identify the NT-SCCs of global network
- transfer $\{\mathcal{M}[\mathcal{C}_{i,k_l}], \mathcal{M}[x_i^k]\}$ to coordination layer;
- identify the NT-SCCs from local SCCs, i.e., if $\mathcal{M}[\mathcal{C}_{i,k_l}] = 0$, then use $\mathcal{C}_{i,k_l}^n$ to represent $\mathcal{C}_{i,k_l}$;
- identify the NT-SCCs from the combined SCCs, i.e., $\forall x_i^k \in \mathcal{C}_{\mathcal{N}_{k_c}, k_c} \cup \mathcal{C}_{add, k_c}$ and $\forall i \in [1 : r]$, if $\mathcal{M}[x_i^k] = 0$, then denote by $\mathcal{C}_{\mathcal{N}_{k_c}, k_c}^n = \mathcal{C}_{\mathcal{N}_{k_c}, k_c} \cup \mathcal{C}_{add, k_c}$;
- obtain the NT-SCCs of global network $\mathcal{C}_k^n$, which is composed by $\mathcal{C}_{i,k_l}^n$ and $\mathcal{C}_{\mathcal{N}_{k_c}, k_c}^n$ with $k \in [1 : N^n]$.

Then, to find the combined SCCs and identify the NT-SCCs of global network, our main idea is: first, seek the SCCs, inward and outward connected pairs of every local block in parallel; second, with the aid of communication, synthesize local information to determine the SCCs of global network, including the local SCCs and the combined ones; third, mark up for every local SCC and the local states involved in combined SCCs; finally, identify the NT-SCCs of global network according to the obtained markers. The complete block-based algorithm to find the NT-SCCs of global system is provided in Algorithm 1.

Remark 2. In the determination of the inward connected pairs, the states of k_l-th local SCC of block Σ_i are denoted by $\mathcal{C}_{i,k_l}$ for brevity. In particular, to include all the states on the directed paths from an incoming local state x_i^l to an outgoing local state x_i^m, it is necessary to put the condition setting that x_i^l and x_i^m should belong to different $\mathcal{C}_{i,k_l}$.

4.3. Block-based algorithm to solve the MIDSC

By Lemma 1, another condition to be considered for addressing the MIDSC problem is $[\bar{A}, \bar{B}] = \boldsymbol{n}$. In structural graph-based theory, it is equivalent to find a maximum matching of $\mathcal{B}(\bar{A}, \bar{B})$

with size $\boldsymbol{n}$. To solve the MBM problem in blocking framework, a prominent issue is that: the total size of maximum matchings of every block may be smaller than the size of the maximum matching of global network. To tackle this challenge, our main idea and method are presented as follows.

Since solving the MBM problem equals to find a smallest right-unmatched vertex-set of the same bipartite graph, if we can find a matching that minimizes the sum of right-unmatched vertices of every local block, then the smallest right-unmatched vertex-set of global network can be obtained. To do this, we analyze the bipartite graph $\mathcal{B}(\bar{A}_{ii})$ of every block and have the following findings: (1) in $\mathcal{B}(\bar{A}_{ii})$, the left vertices of $\mathcal{X}_{\mathcal{L}_i}^{abs}$ can be used to match with the right vertices of block Σ_i only; (2) the left vertices of $\mathcal{X}_{\mathcal{L}_i}^{out}$ can be employed to match with not only the right vertices of $\mathcal{B}(\bar{A}_{ii})$ but also the right vertices of $\mathcal{B}(\bar{A}_{jj})$ with $j \in \mathcal{O}_i$. Therefore, we refer to the states of $\mathcal{X}_{\mathcal{L}_i}^{out}$ and $\mathcal{X}_{\mathcal{N}_i}^{in}$ as the *associated states* of block Σ_i, which may contribute to reducing the right-unmatched vertices of $\mathcal{B}(\bar{A}_{ii})$. The set of associated states of block Σ_i is denoted by $\mathcal{X}_i^{ass} = \mathcal{X}_{\mathcal{L}_i}^{out} \cup \mathcal{X}_{\mathcal{N}_i}^{in}$. Then to address the MBM problem in a block-based manner, the key is how to optimize the allocation of every associated state into the most suitable block so that the total size of the maximum matchings of every block is maximized. To this end, some concepts and theoretical results are presented.

Definition 3 (*Extended Bipartite Graph*)**.** Given a bipartite graph $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, if one adds a vertex-set $\bar{\mathcal{V}}_1$ to $\mathcal{V}_1$ and an edge-set $\mathcal{E}_{\bar{\mathcal{V}}_1,\mathcal{V}_2}$ to $\mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2}$, where $\bar{\mathcal{V}}_1 \cap \mathcal{V}_1 = \emptyset$, then the obtained $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$ is referred to as an extended bipartite graph of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, where $\mathcal{V}_1^+ = \bar{\mathcal{V}}_1 \cup \mathcal{V}_1$ and $\mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2} = \mathcal{E}_{\bar{\mathcal{V}}_1,\mathcal{V}_2} \cup \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2}$.

Lemma 2. *Let s be the size of the right-unmatched vertex-set with respect to a maximum matching M^* of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, and s^+ be the cardinality of the right-unmatched vertex-set with respect to a maximum matching M^{+*} of $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$. If $\bar{\mathcal{V}}_1$ contains only one vertex $\bar{v}$, i.e., $\mathcal{V}_1^+ = \mathcal{V}_1 \cup \bar{v}$, then there is $s^+ = s$ or $s^+ = (s-1)$.*

In Lemma 2, it can be readily seen that if any right-unmatched vertex with respect to M^* of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$ can be matched by $\mathcal{E}_{\bar{v},\mathcal{V}_2}$, then there is $s^+ = (s-1)$. Otherwise, $s^+ = s$ holds true. On this basis, the extensible vertex of a bipartite graph can be defined below.

Definition 4 (*Extensible Vertex*)**.** Let $\bar{\mathcal{V}}_1 = \bar{v}$, if there is a maximum matching M^{+*} of $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$ so that $s^+ = (s-1)$, then $\bar{v}$ is known as an extensible vertex of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$.

Definition 5 (*Valid Matching*)**.** Given an extensible vertex $\bar{v}$ of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$ and a maximum matching M^{+*} of $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$, if $(\bar{v}, v_2) \in M^{+*}$, $v_2 \in \mathcal{V}_2$, then $(\bar{v}, v_2)$ is said to be a valid matching of M^{+*} and denoted by $(\bar{v}, v_2)_v$.

Definition 6 (*Eliminable Matching*)**.** Given an extensible vertex $\bar{v}$ and a maximum matching M^* of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, if there exists an edge $(\bar{v}, v_2^{um})$ while satisfying $v_2^{um} \in \mathcal{V}_2^{um}(M^*)$, where $\mathcal{V}_2^{um}(M^*)$ is the right-unmatched vertex-set of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$ with respect to M^*, then denoted by $(\bar{v}, v_2^{um})_e$ an eliminable matching of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$.

Note that the valid matching depends on M^{+*} while the eliminable matching relies on M^*. To better illustrate the difference of the two matchings, an example is provided in Fig. 2.

Theorem 1. *If $\bar{v}$ is an extensible vertex of the bipartite graph $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, then there must exist an eliminable right vertex v_2^{um} such that $v_2^{um} \in \mathcal{V}_2^{um}(M_g^*)$, where M_g^* is any given maximum matching of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$.*

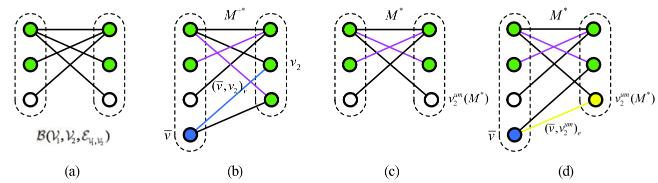


Fig. 2. Given a bipartite graph shown in (a), its extended bipartite graph is shown in (b). The maximum matching M^{+*} is denoted by the pink and blue edges while the valid matching $(\bar{v}, v_2)_v$ is shown by the blue edge in (b). Given a maximum matching M^* of (a), as shown by the pink edges in (c), the eliminable matching $(\bar{v}, v_2^{um})_e$ is denoted by the yellow edge in (d) . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

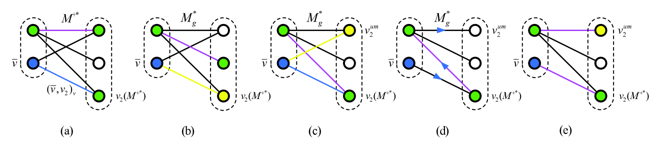


Fig. 3. Given a $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$ shown in (a), where the M^{+*} is denoted by the pink and blue edges of (a). (b) shows the situation (1) of Proof 1, where M_g^* is denoted by the pink edge and $v_2^{um} = v_2(M^{+*})$ is shown by the yellow node. The first case of situation (2) is shown in (c), where M_g^* is shown by the pink edge while $(\bar{v}, v_2^{um})_e$ is denoted by the yellow edge. The second case of situation (2) is provided in (d), where the augmenting path is shown by the edges with blue arrows and $v_2^{um} \in \mathcal{V}_2^{um}(M_g^*)$ is shown by the yellow node of (e) . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Proof 1. By Definition 4, if $\bar{v}$ is an extensible vertex of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, there must exist a maximum matching M^{+*} of $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$ so that $s^+ = (s-1)$ holds true. To proceed, let us denote by $v_2(M^{+*})$ the right vertex of $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$ matched with $\bar{v}$ through $(\bar{v}, v_2(M^{+*}))_v$. Meanwhile, we denote by $\mathcal{V}_2^{um}(M_g^*)$ and $\mathcal{V}_2^{m}(M_g^*)$ the right-unmatched and matched vertex-set with respect to M_g^* of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$. Then review the vertex $v_2(M^{+*})$ in $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$, there exist two situations, i.e., $v_2(M^{+*}) \in \mathcal{V}_2^{um}(M_g^*)$ or $v_2(M^{+*}) \in \mathcal{V}_2^{m}(M_g^*)$, which are discussed, respectively, as follows:
(1) if $v_2(M^{+*}) \in \mathcal{V}_2^{um}(M_g^*)$, then $(\bar{v}, v_2(M^{+*}))_e$ is an eliminable matching of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$ (see Fig. 3(b)). In this situation, another maximum matching $\hat{M}^{+*}$ of $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$ can be obtained by defining $\hat{M}^{+*} = M_g^* \cup (\bar{v}, v_2(M^{+*}))_e$. Then $v_2^{um} = v_2(M^{+*})$. By Definition 6, $v_2^{um} \in \mathcal{V}_2^{um}(M_g^*)$ is proved;
(2) else if $v_2(M^{+*}) \in \mathcal{V}_2^{m}(M_g^*)$, since $s^+ = (s-1)$, there must exist a right-unmatched vertex of some maximum matching $\tilde{M}^*$ of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$ can be matched by a vertex of $\mathcal{V}_1^+$. Specifically, in one case, if $(\bar{v}, v_2^{um})_e$ can be found (see Fig. 3(c)), then $v_2^{um} \in \mathcal{V}_2^{um}(M_g^*)$ definitely holds by Definition 6. In another case, if $(\bar{v}, v_2^{um})_e$ is non-existent (see Fig. 3(d)), one can find an augmenting path (i.e., a path connecting a left-unmatched vertex and a right-unmatched vertex with the matching edge and un-matching edge appear on this path alternatively) from $\bar{v}$ to a vertex of $\mathcal{V}_2^{um}(M_g^*)$. By performing the augmenting path inversion (i.e., changing the un-matching edge of the augmenting path into a matching edge while changing the matching edges into un-matching ones), v_2^{um} matched by a vertex of $\mathcal{V}_1^+$ with satisfying $v_2^{um} \in \mathcal{V}_2^{um}(M_g^*)$ can be finally found (see Fig. 3(e)). Because $\bar{v}$ is an extensible vertex of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, the augmenting path is always existed in the second case. Consequently, $v_2^{um} \in \mathcal{V}_2^{um}(M_g^*)$ is tenable with respect to any given M_g^*, which completes the proof of Theorem 1. □

Remark 3. Theorem 1 reveals that any given maximum matching M_g^* (or equivalently, any right-unmatched vertex-set $\mathcal{V}_2^{um}(M_g^*)$) of a bipartite graph will not effect the judgment on the existence of an eliminable right vertex v_2^{um} corresponding to a given extensible vertex $\bar{v}$. This result indicates that, a random initial right-unmatched vertex-set can be adopted to find the eliminable right vertices of every local block.

Theorem 2. *Let $\bar{v}^1, \ldots, \bar{v}^k$ be k different extensible vertices of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$, whose a group of eliminable matchings with respect to a certain maximum matching M^* of $\mathcal{B}(\mathcal{V}_1, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1,\mathcal{V}_2})$ are $(\bar{v}^1, v_{2,1}^{um})_e, \ldots, (\bar{v}^k, v_{2,k}^{um})_e$. Denote by M^{+*} a maximum matching of $\mathcal{B}(\mathcal{V}_1^+, \mathcal{V}_2, \mathcal{E}_{\mathcal{V}_1^+,\mathcal{V}_2})$, where $\mathcal{V}_1^+ = \bar{\mathcal{V}}_1 \cup \mathcal{V}_1$ and $\bar{\mathcal{V}}_1 = \{\bar{v}^1, \ldots, \bar{v}^k\}$. Let s ($s \geq k$) be the size of $\mathcal{V}_2^{um}(M^*)$ and s^+ be the size of $\mathcal{V}_2^{um}(M^{+*})$. If $v_{2,1}^{um}, \ldots, v_{2,k}^{um} \in \mathcal{V}_2^{um}(M^*)$ and $v_{2,1}^{um}\cap, \ldots, \cap v_{2,k}^{um} = \emptyset$, then there is $s^+ = (s - k)$ holds true.*

Proof 2. Because the k extensible vertices correspond to k different eliminable right vertices with respect to the same M^*, the k eliminable right vertices of $\mathcal{V}_2^{um}(M^*)$ can be matched by the k different eliminable matchings $(\bar{v}^1, v_{2,1}^{um})_e, \ldots, (\bar{v}^k, v_{2,k}^{um})_e$. In consequence, $s^+ = (s - k)$ is true. □

Theorem 2 implies that the effect of each extensible vertex in eliminating the right-unmatched vertex is independent if they can match with a different eliminable right vertex with respect to a certain M^*. Therefore, we can use the following method to solve the MBM problem in a block-based framework: (1) define the *bipartite graph of rest states* $\mathcal{B}(\bar{A}_{ii}^{rest}) = \mathcal{B}(\mathcal{X}_i^{rest}, \mathcal{X}_i, \mathcal{E}_{\mathcal{X}_i^{rest},\mathcal{X}_i})$ for every local block, where $\mathcal{X}_i^{rest} = \mathcal{X}_i \setminus \mathcal{X}_{\mathcal{L}_i}^{out}$ is the rest local state-set; (2) determine a random initial right-unmatched vertex-set $\mathcal{X}_i^{um}(M_i^{rest})$ with respect to any maximum matching M_i^{rest} of $\mathcal{B}(\bar{A}_{ii}^{rest})$ (revealed by Theorem 1); (3) select the extensible vertices from $\mathcal{X}_i^{ass}$ and find the eliminable matchings of every block; (4) determine the optimal allocation of every associated state by computing a maximum matching of the *association bipartite graph* (revealed by Theorem 2) $\mathcal{B}(\bar{E}) = \mathcal{B}(\mathcal{X}^{ass}, \mathcal{X}^{um}, \mathcal{E}_{\mathcal{X}^{ass},\mathcal{X}^{um}})$, where $\mathcal{X}^{ass} = \bigcup_{i=1}^r \mathcal{X}_i^{ass}$, $\mathcal{X}^{um} = \bigcup_{i=1}^r \mathcal{X}_i^{um}(M_i^{rest})$ and $\mathcal{E}_{\mathcal{X}^{ass},\mathcal{X}^{um}} = \{(x_i^{ass}, x_i^{um}) : x_i^{ass} \in \mathcal{X}_i^{ass}, x_i^{um} \in \mathcal{X}_i^{um}(M_i^{rest})$, and $(x_i^{ass}, x_i^{um})_e$ is existed$\}$. The block-based graphical approach to resolve the MIDSC problem is presented in Algorithm 2. Note that, based on the result of Algorithm 1, the solution of Algorithm 2 always satisfies the two conditions required for guaranteeing the structural controllability of global system with using a minimum number of actuated states. Therefore, the resulting solution is globally optimal.

Remark 4. It is noteworthy that in *Setp2.4* of Algorithm 2, if u_k is not matched in M^{+*}, at the same time, $\mathcal{C}_k^n \cap \mathcal{X}^{um} = \emptyset$. Then select any state x_i^l of $\mathcal{C}_k^n$ (which must be matched in M_i^{rest}, $\forall i \in [1:r]$) and match x_i^l to u_k, which brings no change in M^{+*}.

Algorithm 2. Block-based algorithm to solve MIDSC

Step2.1: Initialization

- determine the associated state-set $\mathcal{X}_i^{ass}$;
- obtain $\mathcal{X}_i^{rest}$ and $\mathcal{X}_i^{rest+} = \mathcal{X}_i^{rest} \cup x_i^{ass}, \forall x_i^{ass} \in \mathcal{X}_i^{ass}$;
- construct the bipartite graph $\mathcal{B}(\bar{A}_{ii}^{rest})$;
- find any maximum matching M_i^{rest} of $\mathcal{B}(\bar{A}_{ii}^{rest})$, and obtain the right-unmatched vertex-set $\mathcal{X}_i^{um}(M_i^{rest})$;

Step2.2: Find the eliminable matchings

- construct $\mathcal{B}(\bar{A}_{ii}^{rest+}) = \mathcal{B}(\mathcal{X}_i^{rest+}, \mathcal{X}_i, \mathcal{E}_{\mathcal{X}_i^{rest+},\mathcal{X}_i})$;
- select the extensible vertices of $\mathcal{B}(\bar{A}_{ii}^{rest})$ from $\mathcal{X}_i^{ass}$;
- find all the eliminable matchings $(x_i^{ass}, x_i^{um})_e$ in $\mathcal{B}(\bar{A}_{ii}^{rest+})$ with $x_i^{um} \in \mathcal{X}_i^{um}(M_i^{rest})$;

Step2.3: Construct the extended association bipartite graph

- transfer $\{\mathcal{X}_i^{um}(M_i^{rest}), (x_i^{ass}, x_i^{um})_e\}$ to coordination layer;
- construct the association bipartite graph $\mathcal{B}(\bar{E})$;
- add the set $\mathcal{U}^n = \{u_1, \ldots, u_{N^n}\}$ to the left vertex-set $\mathcal{X}^{ass}$ of $\mathcal{B}(\bar{E})$, and add new edges from every u_k to every right vertex of $\mathcal{B}(\bar{E})$ that is involved in $\mathcal{C}_k^n$. Obtain the extended association bipartite graph $\mathcal{B}(\bar{E}^+)$;

Step2.4: Determine the minimum input design

- compute a maximum matching M^{+*} of $\mathcal{B}(\bar{E}^+)$;
- for any $k \in [1:N^n]$, if u_k is not matched in M^{+*}, select any state x_i^l of $\mathcal{C}_k^n \cap \mathcal{X}^{um}$ (which must be matched in M^{+*}), remove the matching edge of x_i^l and match x_i^l with u_k. Denote the new maximum matching by $\tilde{M}^{+*}$;
- determine the minimum input design $\mathscr{D}_i$ of every block, which is the index-set of states in $\{\mathcal{X}^{um}(\tilde{M}^{+*}) \cup \mathcal{X}^m(\mathcal{U}^n)\} \cap \mathcal{X}_i$, where $\mathcal{X}^{um}(\tilde{M}^{+*})$ is the set of right vertices of $\mathcal{B}(\bar{E}^+)$ that are not matched by $\tilde{M}^{+*}$, and $\mathcal{X}^m(\mathcal{U}^n)$ is the set of states matched by any u_k.

5. Complexity analysis and comparison

In this section, we analyze the complexity of the proposed approach and demonstrate its advantage by comparing it with the fastest (up-to-date) algorithm (Olshevsky, 2015) to the MIDSC problem that is designed based on a monolithic system model. The analytical results show that, the algorithm complexity of the block-based method is strictly lower than that of the fastest algorithm (Olshevsky, 2015) under some mild conditions on the blocking network structure, i.e., if each well partitioned block meets the following criteria

$$(\boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in}) \leq \sqrt{\boldsymbol{n}}, \tag{9}$$

where the correlation degrees $\boldsymbol{d}_i^{out}$ and $\boldsymbol{d}_{ri}^{in}$ are mainly determined by a specific blocking topology.

Hereafter, the complexity of Algorithms 1 and 2 is analyzed, respectively, to obtain the complete complexity of the block-based approach.

(i) Let us denote by $\boldsymbol{n}_i = |\mathcal{X}_i|$, $\boldsymbol{m}_i = |\mathcal{E}_{\mathcal{X}_i,\mathcal{X}_i} \cup \mathcal{E}_{\mathcal{X}_i,\mathcal{X}_{\mathcal{N}_i}^{out}}|$ and $\boldsymbol{d} = \sum_{i=1}^r \boldsymbol{d}_i$. In the worst case, one needs to traverse all the vertices and edges in local computation, and check every associated vertex for coordination. Thus, Algorithm 1 can be certainly performed in the operations of

$$T_1 = \sum_{i=1}^r O(\boldsymbol{m}_i + \boldsymbol{n}_i) + O(\boldsymbol{d}). \tag{10}$$

(ii) Then let $\boldsymbol{m}_{i,1} = |\mathcal{E}_{\mathcal{X}_i^{rest},\mathcal{X}_i}|$ be the number of edges in $\mathcal{B}(\bar{A}_{ii}^{rest})$. Since the randomized Mucha–Sankowski and Madry's algorithms give the correct answer with a certain probability (at least $1 - 1/\boldsymbol{n}$), the determinate Hopcroft–Karp algorithm is employed in finding the maximum matching. That is, the time complexity for determining the M_i^{rest} is denoted as

$$T_{2,1} = \sum_{i=1}^r O(\boldsymbol{m}_{i,1}\sqrt{\boldsymbol{n}_i}).$$

To proceed, let $\boldsymbol{m}_{i,2}(x_i^{ass}) = |\mathcal{E}_{\mathcal{X}_i^{rest},\mathcal{X}_i} \cup \mathcal{E}_{x_i^{ass},\mathcal{X}_i}|$ be the number of edges in $\mathcal{B}(\bar{A}_{ii}^{rest+})$. Since $\boldsymbol{m}_{i,2}(x_i^{ass})$ depends on different x_i^{ass}, we use $\boldsymbol{m}_{i,2}^{max}$ to denote the biggest $\boldsymbol{m}_{i,2}(x_i^{ass})$. Additionally, due to the number of associated states is $|\mathcal{X}_i^{ass}| = \boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in}$, the time consumed to explore the eliminable matchings is derived by

$$T_{2,2} = \sum_{i=1}^r O(\boldsymbol{m}_{i,2}^{max}(\boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in})).$$

Next, denote by $\mathcal{D}(\bar{E}) = (\mathcal{X}^{ass} \cup \mathcal{X}^{um}, \mathcal{E}_{\mathcal{X}^{ass},\mathcal{X}^{um}})$, whose number of vertices and edges are $\boldsymbol{n}_e = |\mathcal{X}^{ass} \cup \mathcal{X}^{um}|$ and $\boldsymbol{m}_e = |\mathcal{E}_{\mathcal{X}^{ass},\mathcal{X}^{um}}|$.

Then in determining the optimal allocation of the associated states, the time complexity is

$$T_{2,3} = O(\boldsymbol{m}_e\sqrt{\boldsymbol{n}_e}).$$

Hence, the complexity of Algorithm 2 is obtained by

$$\begin{aligned} T_2 =& T_{2,1} + T_{2,2} + T_{2,3} \\ =& O(\max_{i\in[1:r]}\{\boldsymbol{m}_{i,1}\sqrt{\boldsymbol{n}_i}, \boldsymbol{m}_{i,2}^{max}(\boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in})\}) + O(\boldsymbol{m}_e\sqrt{\boldsymbol{n}_e}). \end{aligned} \quad (11)$$

Given the above, the complete time complexity of the block-based approach has the form of

$$\begin{aligned} T =& T_1 + T_2 \\ =& O(\max_{i\in[1:r]}\{\boldsymbol{m}_i + \boldsymbol{n}_i, \boldsymbol{d}, \boldsymbol{m}_{i,1}\sqrt{\boldsymbol{n}_i}, \boldsymbol{m}_{i,2}^{max}(\boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in}), \boldsymbol{m}_e\sqrt{\boldsymbol{n}_e}\}). \end{aligned} \quad (12)$$

Eq. (12) reveals that the complexity of the block-based algorithm is closely related to the size and complication (i.e., the number of nodes and edges involved in a block) of the biggest block, the maximal correlation degree and the number of associated edges. It indicates that a blocking structure with fewer local nodes, less local edges and lower correlation degree is more favorable to enhance the algorithm efficiency.

In what follows, we further expound the advantage of the proposed approach by comparing it with the fastest algorithm (Olshevsky, 2015) to the MIDSC problem designed based on a monolithic system model, whose complexity is $O(\boldsymbol{m}\sqrt{\boldsymbol{n}})$. Now let us denote by $\boldsymbol{n} = \sum_{i=1}^{r} \boldsymbol{n}_i$, $\boldsymbol{m} = \sum_{i=1}^{r} \boldsymbol{m}_i$, then there is

$$\max_{i\in[1:r]}\{\boldsymbol{m}_i + \boldsymbol{n}_i\} < \boldsymbol{m} + \boldsymbol{n} < \boldsymbol{m}\sqrt{\boldsymbol{n}}, \quad \boldsymbol{d} \le \boldsymbol{n} < \boldsymbol{m}\sqrt{\boldsymbol{n}}.$$

Since $\mathcal{D}(\bar{A}_{ii}^{rest})$ and $\mathcal{D}(\bar{E})$ are sub-digraphs of $\mathcal{D}(\bar{A})$, we have $\max_{i\in[1:r]}\{\boldsymbol{m}_{i,1}\sqrt{\boldsymbol{n}_i}\} < \boldsymbol{m}\sqrt{\boldsymbol{n}}$, $\boldsymbol{m}_e\sqrt{\boldsymbol{n}_e} < \boldsymbol{m}\sqrt{\boldsymbol{n}}$. Moreover, because the matching edges of $\mathcal{B}(\bar{A}_{ii}^{rest+})$ constitute a subset of matching edges of $\mathcal{B}(\bar{A})$, $\boldsymbol{m}_{i,2}^{max} < \boldsymbol{m}$ holds true. Thus, when a proper blocking structure is attainable with $(\boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in}) \le \sqrt{\boldsymbol{n}}$ is satisfied for every local block, it derives that

$$\max_{i\in[1:r]}\{\boldsymbol{m}_{i,2}^{max}(\boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in})\} < \boldsymbol{m}\sqrt{\boldsymbol{n}}.$$

Taken together, we prove that

$$T < O(\boldsymbol{m}\sqrt{\boldsymbol{n}}), \quad (13)$$

which demonstrates a higher efficiency of the proposed method.

Remark 5. Criteria (9) is a mild condition on the blocking network structure since it can be certainly met for the worst case with $r = 1$, i.e., there is only one block and $\boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in} = 0$. In this case, $(\boldsymbol{d}_i^{out} + \boldsymbol{d}_{ri}^{in}) \le \sqrt{\boldsymbol{n}}$ definitely holds and the algorithm complexity gets back to $O(\boldsymbol{m}\sqrt{\boldsymbol{n}})$ without improvement. However, in some other cases with $r > 1$, we can eventually obtain a favorable blocking structure that satisfies (9) by means of properly merging the blocks that do not meet the criteria into a larger block. In these cases, the algorithm complexity can be certainly reduced by employing the proposed approach.

6. Illustrative example

Next, we validate the proposed approach through an illustrative example from Carvalho et al. (2017) and Pequito et al. (2016). Here we consider a network that has been well partitioned into six blocks. For every block, the criterion (9) is satisfied. Without loss of generality, the blocking structure with an SCC is split by block Σ_1 and Σ_2 is taken into account. The global network digraph $\mathcal{D}(\bar{A})$ with a blocking structure is shown in the following Fig. 4.

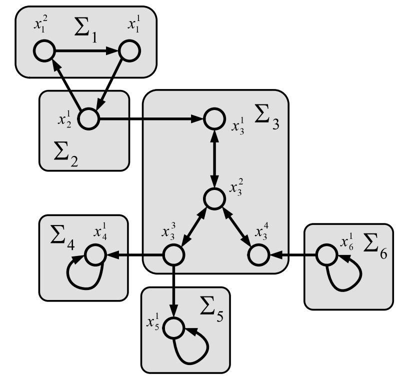


Fig. 4. The global network digraph $\mathcal{D}(\bar{A})$ with a blocking structure.

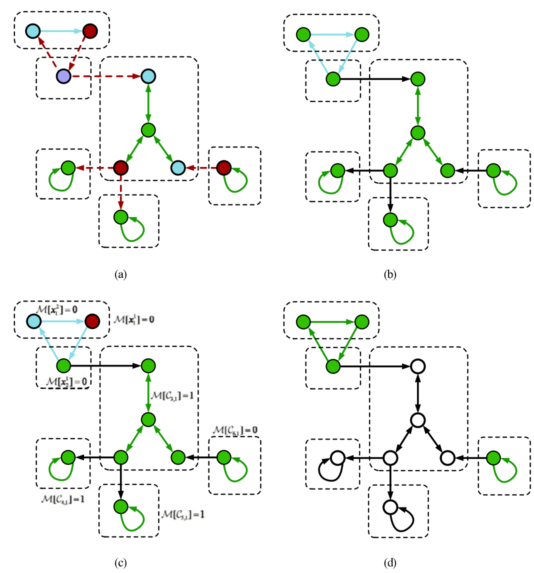


Fig. 5. The implementation of Algorithm 1. The nodes of $\mathcal{X}_{\mathcal{L}_i}^{in}$, $\mathcal{X}_{\mathcal{L}_i}^{out}$, and the intersection of the two sets are, respectively, shown by the blue, red and purple ones in (a). The outward connected pair is connected by a red dotted edge while the inward connected pair is connected by a blue solid edge in (a). The local SCC is denoted by the green nodes connected with green edges in (a). In (b), the obtained combined SCC is shown by green nodes connected with blue edges. (c) shows the markers while (d) shows the NT-SCCs of global network . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The implementation to *Step1.1–Step1.4* of Algorithm 1 is shown in (a)–(d) of Fig. 5 while the inward and outward connected pairs of every block are given in Table 1. In local SCCs, $\mathcal{C}_{3,1} = \{x_3^1, x_3^2, x_3^3, x_3^4\}$. In this example, one combined SCC is found with $\mathcal{C}_{\mathcal{N}_1,1} = \{x_1^1, x_1^2, x_2^1\}$ and $\mathcal{N}_1 = \{1, 2\}$. The obtained NT-SCCs are $\{x_1^1, x_1^2, x_2^1\}$ and $\{x_6^1\}$, which are shown by the green nodes connected with green edges in Fig. 5(d).

Based on the above result, Algorithm 2 is implemented to solve the MIDSC problem. $\mathcal{X}_i^{ass}$ of every block is obtained as

$$\begin{aligned} &\mathcal{X}_1^{ass} = \{x_1^1, x_2^1\}, \ \mathcal{X}_2^{ass} = \{x_1^1, x_2^1\}, \ \mathcal{X}_3^{ass} = \{x_2^1, x_3^3, x_6^1\}, \\ &\mathcal{X}_4^{ass} = \{x_3^3\}, \quad \mathcal{X}_5^{ass} = \{x_3^3\}, \quad \mathcal{X}_6^{ass} = \{x_6^1\}. \end{aligned}$$

In Fig. 6, the constructed $\mathcal{B}(\bar{A})$, $\mathcal{B}(\bar{A}_{ii}^{rest})$ and $\mathcal{B}(\bar{E}^{+})$ are shown by the subgraph (a), (b) and (c), respectively. Meanwhile, the eliminable matchings of every block are provided in Table 1. According to the optimal allocation result shown in Fig. 6(c), we have $\mathcal{X}^{um}(M^{+*}) = \emptyset$, $\mathcal{X}^{m}(\mathcal{U}^{nt}) = \{x_1^2, x_6^1\}$. Consequently, the optimal solution of the MIDSC problem in this example is $\mathscr{D}_1 = \{2\}$ and $\mathscr{D}_6 =$

Table 1
The connected pairs and eliminable matchings of local blocks.

Block	Inward connected pair	Outward connected pair	Eliminable matching
Σ_1	$(x_1^2, x_1^1)^{in}$	$(x_1^1, x_2^1)^{out}$	$(x_2^1, x_1^2)_e$
Σ_2	–	$(x_2^1, x_1^2)^{out}$	$(x_1^1, x_2^1)_e$
Σ_3	–	$(\mathcal{C}_{3,1}, x_4^1)^{out}$	$(x_2^1, x_3^1)_e$
		$(\mathcal{C}_{3,1}, x_5^1)^{out}$	$(x_6^1, x_3^4)_e$
Σ_4	–	–	$(x_6^1, x_6^1)_e$
Σ_5	–	–	–
Σ_6	–	$(\mathcal{C}_{6,1}, x_3^4)^{out}$	–

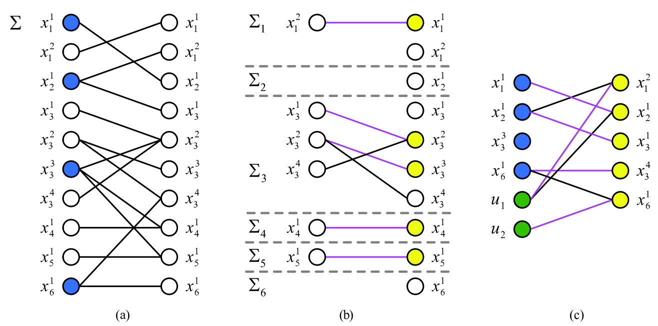


Fig. 6. The construction of $\mathcal{B}(\bar{A})$, $\mathcal{B}(\bar{A}_{ii}^{rest})$ and $\mathcal{B}(\bar{E}^{+})$. (a) shows $\mathcal{B}(\bar{A})$, where the associated states are shown by the blue nodes. (b) shows $\mathcal{B}(\bar{A}_{ii}^{rest})$ of every local block, where M_i^{rest} are shown by the pink edges. The right matched vertices are shown by the yellow nodes while the unmatched vertices are shown by white ones in (b). (c) shows $\mathcal{B}(\bar{E}^{+})$, where $\tilde{M}^{+*}$ is denoted by the pink edges . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

{1}, i.e., the minimum driver nodes for structural controllability are $\{x_1^2, x_6^1\}$. This result is exactly consistent with the solution obtained in Pequito et al. (2016), which verifies the effectiveness of our approach. On the other hand, the algorithm complexity here is reduced by considering the blocking structure.

7. Conclusions and further research

In this work, we have developed a graphical approach to address the minimum input design problem for structural controllability under a block-based structure setting to accommodate the control system design of massive networks. First, we provided tools to identify the smallest subset of driver nodes by two proposed algorithms, which were used to find the NT-SCCs and the maximum matching of global network in a blocking framework, respectively. Then the computational complexity of the proposed method was analyzed. Finally, we shown the advantage of our algorithm by comparing it with existing algorithms. The main results indicate that the complexity of the block-based algorithm is guaranteed to be lower than existing algorithms under a mild condition on the blocking structure. Additionally, the effectiveness of the algorithm was demonstrated by an example.

The blocking framework put forward here raises many open questions, such as an investigation on the efficient graph partitioning strategy for achieving higher algorithm efficiency, which is more challenging but will shed light on a deep understanding of complex network control.

References

Aguilar, C. O., & Gharesifard, B. (2015). Graph controllability classes for the laplacian leader-follower dynamics. *IEEE Transactions on Automatic Control*, *60*(6), 1611–1623.

Assadi, S., Khanna, S., Li, Y., & Preciado, V. M. (2015). Complexity of the minimum input selection problem for structural controllability. *IFAC-PapersOnLine*, *48*(22), 70–75.

Carvalho, J. F., Pequito, S., Aguiar, A. P., Kar, S., & Johansson, K. H. (2017). Composability and controllability of structural linear time-invariant systems: Distributed verification. *Automatica*, *78*, 123–134.

Chui, C. K., & Chen, G. (2012). *Linear systems and optimal control, Vol. 18*. Springer Science & Business Media.

Dion, J. M., Commault, C., & Van Der Woude, J. (2003). Generic properties and control of linear structured systems: a survey. *Automatica*, *39*(7), 1125–1144.

Hopcroft, J. E., & Karp, R. M. (1973). An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, *2*(4), 225–231.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., & Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, *2*(3), 203–271.

Lin, C. T. (1974). Structural controllability. *IEEE Transactions on Automatic Control*, *19*(3), 201–208.

Liu, Y. Y., & Barabási, A. L. (2016). Control principles of complex systems. *Reviews of Modern Physics*, *88*(3), 035006.

Liu, F., & Morse, A. S. (2017). Structural controllability of linear time-invariant systems. arXiv preprint arXiv:1707.08243.

Liu, Y. Y., Slotine, J. J., & Barabási, A. L. (2011). Controllability of complex networks. *Nature*, *473*(7346), 167–173.

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

Maza, S., Simon, C., & Boukhobza, T. (2012). Impact of the actuator failures on the structural controllability of linear systems: a graph theoretical approach. *IET Control Theory & Applications*, *6*(3), 412–419.

Olshevsky, A. (2015). Minimum input selection for structural controllability. In *American control conference* (pp. 2218–2223).

Pan, Y., & Li, X. (2014). Structural controllability and controlling centrality of temporal networks. *PLoS One*, *9*(4), e94998.

Pequito, S., Kar, S., & Aguiar, A. P. (2013). A structured systems approach for optimal actuator–sensor placement in linear time-invariant systems. In *American control conference* (pp. 6108–6113).

Pequito, S., Kar, S., & Aguiar, A. P. (2016). A framework for structural input/output and control configuration selection in large-scale systems. *IEEE Transactions on Automatic Control*, *61*(2), 303–318.

Sundaram, S., & Hadjicostis, C. N. (2013). Structural controllability and observability of linear systems over finite fields with applications to multi-agent systems. *IEEE Transactions on Automatic Control*, *58*(1), 60–73.

Wu, L., Li, M., Wang, J., & Wu, F. X. (2016). Minimum steering node set of complex networks and its applications to biomolecular networks. *IET Systems Biology*, *10*(3), 116–123.

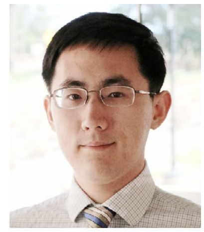
Yin, H., & Zhang, S. (2016). Minimum structural controllability problems of complex networks. *Physica A. Statistical Mechanics and its Applications*, *443*, 467–476.

Yuan, Z., Zhao, C., Di, Z., Wang, W. X., & Lai, Y. C. (2013). Exact controllability of complex networks. *Nature Communications*, *4*.

Ting Bai was born in Shaanxi, China, in 1990. She received the B.S. degree in Automation from Northwestern Polytechnical University in 2013. Since 2013, she has been with the Department of Automation, Shanghai Jiao Tong University, where she is working toward the Ph.D. degree in Control Science and Engineering. Her research interests include distributed model predictive control, complex networked systems, with a recent focus on the minimum control configuration design and distributed optimal control of reconfigurable networked systems.

Shaoyuan Li (SM'05) received the B.S. and M.S. degrees in Automation from Hebei University of Technology, Tianjin, China, in 1987 and 1992, respectively, and the Ph.D. degree from the Department of Computer and System Science, Nankai University, Tianjin, in 1997. Since July 1997, he has been with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, where he is currently a Chair Professor. His research interests include model predictive control, dynamic system optimization for large-scale networked systems. He is a vice president of Asian Control Association (ACA) and a senior member of IEEE.

Yuanyuan Zou is a Professor with the Department of Automation, Shanghai Jiao Tong University. She received her B.S. degree and M.S. degree from Ludong University in 2002 and 2005, respectively, and her Ph.D. degree from Shanghai Jiao Tong University in 2009. Her research interests include predictive control, network-based control systems and distributed control systems.



Xiang Yin was born in Anhui, China, in 1991.

He received the B.Eng. degree from Zhejiang University in 2012, the M.S. degree from the University of Michigan, Ann Arbor, in 2013, and the Ph.D. degree from the University of Michigan, Ann Arbor, in 2017, all in electrical engineering.

Since 2017, he has been with the Department of Automation, Shanghai Jiao Tong University, where he is an Associate Professor.

His research interests include formal methods, control of discrete-event systems, model-based fault diagnosis, security and their applications to cyber and cyber–physical systems.

He received the Outstanding Reviewer Awards from Automatica, the IEEE Transactions on Automatic Control and the Journal of Discrete Event Dynamic Systems.

He also received the IEEE Conference on Decision and Control (CDC) Best Student Paper Award Finalist in 2016.

He is the co-chair of the IEEE CSS Technical Committee on Discrete Event Systems.