Graph-Theoretical Methods for Identifiability Analysis and Synthesis in Dynamic Networks

Xiaodong Cheng

Control Group, Department of Engineering, University of Cambridge, UK

The 2021 European Control Conference (Virtual Conference)

Shengling Shi

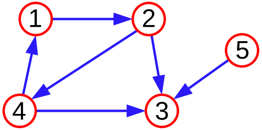
Paul Van den Hof

Outline

1. Problem Setting
2. Vertex-Disjoint Paths
3. Disconnecting Set
   - Generic Identifiability Analysis for Single Module
   - Excitation Allocation for Single Module Identifiability
4. Pseudotree Covering
   - Generic Identifiability Analysis for Full Network
   - Excitation Allocation For Full Network Identifiability
5. Conclusions

Graph Theory

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with node set $\mathcal{V} = \{1, 2, \cdots, L\}$, and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

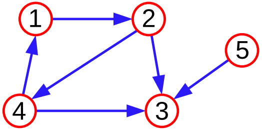


$$\mathcal{V} = \{1, 2, 3, 4, 5\}$$

$$\mathcal{E} = \{(1, 2), (2, 3), (2, 4), (4, 1), (4, 3), (5, 3)\}$$

Graph Theory

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with node set $\mathcal{V} = \{1, 2, \cdots, L\}$, and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$



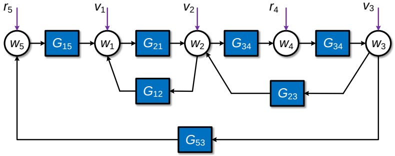
$$\mathcal{V} = \{1, 2, 3, 4, 5\}$$
$$\mathcal{E} = \{(1, 2), (2, 3), (2, 4), (4, 1), (4, 3), (5, 3)\}$$

Terminologies:

- $\mathcal{N}_k^-$: in-neighbors of k, e.g., $\mathcal{N}_3^- = \{2, 4, 5\}$
  $\mathcal{N}_k^+$: out-neighbors of k, e.g., $\mathcal{N}_2^+ = \{3, 4\}$
- source: a node without in-neighbors, e.g., 5;
  sink: a node without out-neighbors, e.g., 3;
- **path**: a sequence of nodes where each node appears at most once
  e.g., $1 \to 2 \to 4 \to 3$ (*a single node can also be viewed as a path*)

Graph Representation of Dynamic Networks

Consider a dynamic network $w(t) = G(q,\theta)w(t) + Rr(t) + H(q,\theta)e(t)$[1]



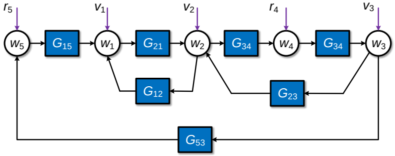
Dynamic network with process noises v_1 and v_2 correlated

$$R = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, H(\theta) = \begin{bmatrix} H_{11}(\theta) & H_{12}(\theta) & 0 \\ H_{21}(\theta) & H_{22}(\theta) & 0 \\ 0 & 0 & H_{33}(\theta) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

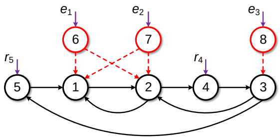
[1] In this talk, all the modules in $G(q)$ are parameterized

Graph Representation of Dynamic Networks

Consider a dynamic network $w(t) = G(q, \theta)w(t) + Rr(t) + H(q, \theta)e(t)$[1]



Dynamic network with process noises v_1 and v_2 correlated



Extended graph $\mathcal{G}$: dashed edges indicate the noise corelation strucutre

$$R = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, H(\theta) = \begin{bmatrix} H_{11}(\theta) & H_{12}(\theta) & 0 \\ H_{21}(\theta) & H_{22}(\theta) & 0 \\ 0 & 0 & H_{33}(\theta) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$G_{ji} \neq 0 \Rightarrow (i, j) \in \mathcal{G}$$

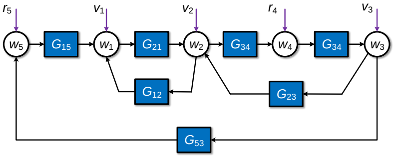
$$H_{i,k} \neq 0 \Rightarrow (i, k + L) \in \mathcal{G}$$

$\mathcal{R} = \{4, 5, 6, 7, 8\}$: excited nodes

[1] In this talk, all the modules in $G(q)$ are parameterized

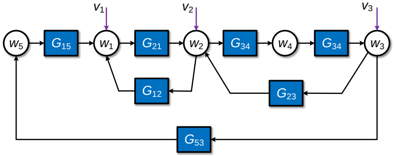
Problems

Analysis problem: generic identifiability condition



Under what conditions (i.e., the number and locations of external excitation signals), $\mathcal{M}$ is generically identifiable?

Synthesis problem: allocation of excitation signals



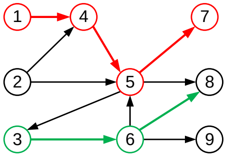
Given a network without external excitation signals, where to allocate excitation signals to achieve generic identifiability?

Vertex-Disjoint Paths

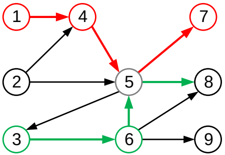
- Two paths in a graph $\mathcal{G}$ are vertex-disjoint if they do not share any common node.

Vertex-Disjoint Paths

- Two paths in a graph $\mathcal{G}$ are vertex-disjoint if they do not share any common node.

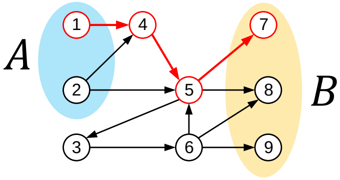


Node-disjoint



Not Node-disjoint

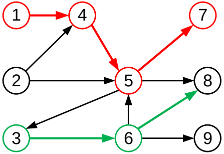
- $b_{\mathcal{A}\to\mathcal{B}}$: the maximum number of vertex-disjoint paths from node set $\mathcal{A}$ to node set $\mathcal{B}$



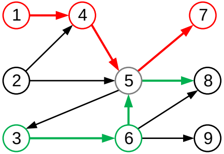
$$\mathcal{A} = \{1, 2\},\ \mathcal{B} = \{7, 8, 9\}$$
$$\Rightarrow b_{\mathcal{A}\to\mathcal{B}} = 1$$

Vertex-Disjoint Paths

- Two paths in a graph $\mathcal{G}$ are vertex-disjoint if they do not share any common node.

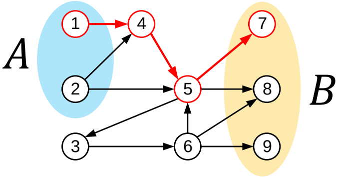


Node-disjoint



Not Node-disjoint

- $b_{\mathcal{A}\to\mathcal{B}}$: the maximum number of vertex-disjoint paths from node set $\mathcal{A}$ to node set $\mathcal{B}$



$$\mathcal{A} = \{1,2\},\ \mathcal{B} = \{7,8,9\}$$
$$\Rightarrow b_{\mathcal{A}\to\mathcal{B}} = 1$$

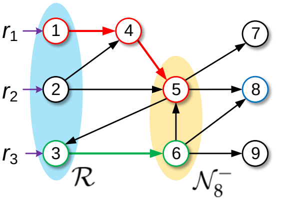
The largest set of vertex-disjoint paths from $\mathcal{A}$ to $\mathcal{B}$ is not unique!

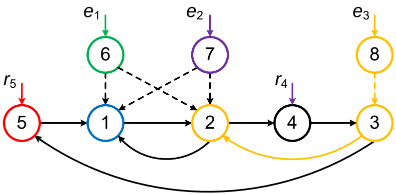
Path-Based Condition

Let $\mathcal{R} \subseteq \mathcal{V}$ be the set of excited nodes, and all the nodes in $\mathcal{V}$ are measured.

Theorem (Path-based condition)

The transfer functions from $\mathcal{N}_j^-$ to j, i.e. $G_{j,\mathcal{N}_j^-}$ are generically identifiable if and only if there are $|\mathcal{N}_j^-|$ node disjoint paths from $\mathcal{R}$ to $\mathcal{N}_j^-$.
The model set $\mathcal{M}$ is generically identifiable, if the above condition holds for all $i \in \mathcal{V}$.



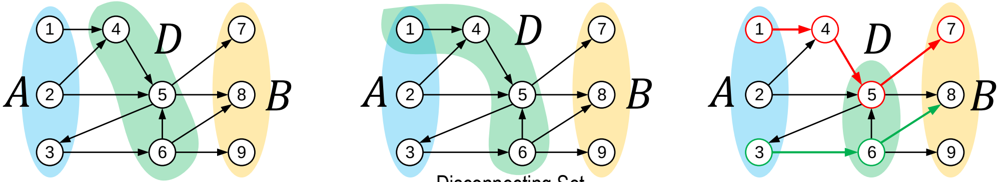


$\mathcal{R} = \{4, 5, 6, 7, 8\}$,
$\mathcal{N}_1^- = \{2, , 5, 67\}$,
$\Rightarrow b_{\mathcal{R}\to\mathcal{N}_1^-} = 4 = |\mathcal{N}_1^-|$

[2][1] Van der Woude, 1991; [2] Hendrickx, Gevers & Bazanella, CDC 2017; [3] Weerts et al., CDC 2018

Disconnecting Set

- In $\mathcal{G}$, a set of nodes $\mathcal{D}$ is a **disconnecting set** from $\mathcal{A}$ to $\mathcal{B}$, if removing $\mathcal{D}$ leads to no paths from $\mathcal{A}$ to $\mathcal{B}$. ($\mathcal{D}$ may share common vertices in the sets $\mathcal{A}$ and $\mathcal{B}$.)
- $\mathcal{D}$ is a **minimum disconnecting set** from $\mathcal{A}$ to $\mathcal{B}$ if $|\mathcal{D}|$ is minimum



Disconnecting Set

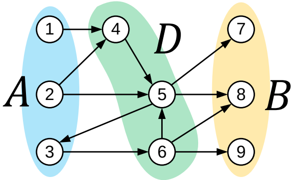
Disconnecting Set
(A and D have overlap)

Minimum Disconnecting Set

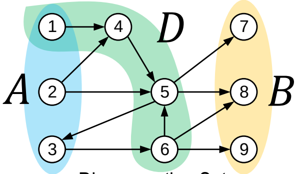
[3] Schrijver 2003

Disconnecting Set

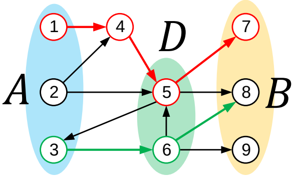
- In $\mathcal{G}$, a set of nodes $\mathcal{D}$ is a **disconnecting set** from $\mathcal{A}$ to $\mathcal{B}$, if removing $\mathcal{D}$ leads to no paths from $\mathcal{A}$ to $\mathcal{B}$. ($\mathcal{D}$ may share common vertices in the sets $\mathcal{A}$ and $\mathcal{B}$.)
- $\mathcal{D}$ is a **minimum disconnecting set** from $\mathcal{A}$ to $\mathcal{B}$ if $|\mathcal{D}|$ is minimum



Disconnecting Set



Disconnecting Set
(A and D have overlap)



Minimum Disconnecting Set

- Menger's theorem: $|\mathcal{D}| = b_{\mathcal{X}\to\mathcal{Y}}$
  - $\mathcal{D}$: a minimum disconnecting set from $\mathcal{A}$ to $\mathcal{B}$
  - $b_{\mathcal{X}\to\mathcal{Y}}$: the maximum number of vertex-disjoint paths from $\mathcal{A}$ to $\mathcal{B}$.

[3] Schrijver 2003

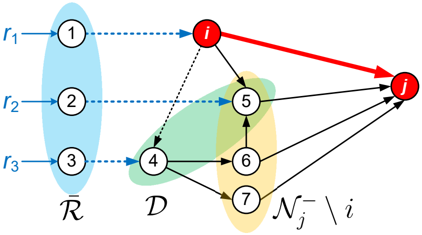
A Disconnecting Set Condition (Single Module)

Let $\mathcal{R} \subseteq \mathcal{V}$ be the set of excited nodes, and all the nodes in $\mathcal{V}$ are measured.

Theorem (Disconnecting set condition)

A single module G_{ji} is generically identifiable if and only if for some $\bar{\mathcal{R}} \subseteq \mathcal{R}$, there is a disconnecting set $\mathcal{D}$ from $i \cup \bar{\mathcal{R}}$ to the other in-neighbors of j (i.e. $\mathcal{N}_j^- \setminus i$) such that

$$b_{\bar{\mathcal{R}} \to \mathcal{D} \cup i} = |D| + 1$$



$\bar{\mathcal{R}} = \{1, 2, 3\}$, $\mathcal{N}_j^- \setminus i = \{5, 6, 7\}$,

$\mathcal{D} = \{4, 5\}$

$\Rightarrow b_{\bar{\mathcal{R}} \to \mathcal{D} \cup i} = b_{\{1,2,3\} \to \{4,5,i\}} = |D| + 1 = 3$

G_{ji} is generic identifiable!

[4] [1] Shi et al., IFAC-PapersOnLine 2020; [2] Shi et al., arXiv:2008.01495

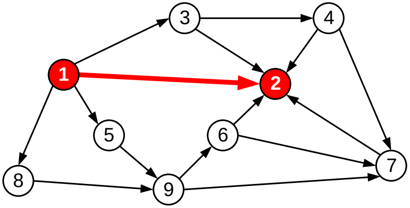
Allocation of Excitation Signals

Problem

How to allocate a minimum number of excitation signals for G_{ji} to be generic identifiable?

The disconnecting set result leads to a simple synthesis procedure:

- **Step 1**: Find a minimum disconnecting set $\mathcal{D}$ from $\mathcal{N}_i^+$ to $\mathcal{N}_j^- \setminus i$.
- **Step 2**: Assign independent external signals at $\mathcal{D}$ and i.



Target module: G_{21}

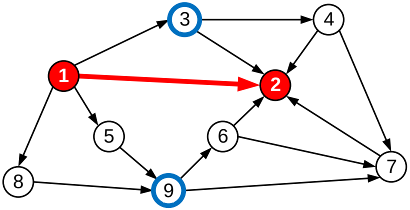
Allocation of Excitation Signals

Problem

How to allocate a minimum number of excitation signals for G_{ji} to be generic identifiable?

The disconnecting set result leads to a simple synthesis procedure:

- **Step 1**: Find a minimum disconnecting set $\mathcal{D}$ from $\mathcal{N}_i^+$ to $\mathcal{N}_j^- \setminus i$.
- **Step 2**: Assign independent external signals at $\mathcal{D}$ and i.



Target module: G_{21}

- Observe that $\mathcal{N}_1^+ = \{2, 3, 5, 8\}$, $\mathcal{N}_2^- \setminus \{1\} = \{3, 4, 6, 7\}$
- Find a minimum disconnecting set from $\mathcal{N}_1^+$ to $\mathcal{N}_2^- \setminus \{1\}$: $\mathcal{D} = \{3, 9\}$

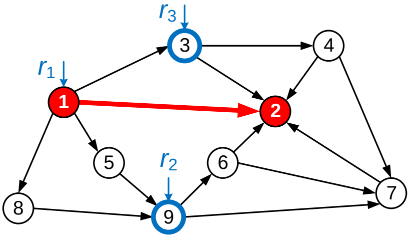
Allocation of Excitation Signals

Problem

How to allocate a minimum number of excitation signals for G_{ji} to be generic identifiable?

The disconnecting set result leads to a simple synthesis procedure:

- **Step 1**: Find a minimum disconnecting set $\mathcal{D}$ from $\mathcal{N}_i^+$ to $\mathcal{N}_j^- \setminus i$.
- **Step 2**: Assign independent external signals at $\mathcal{D}$ and i.

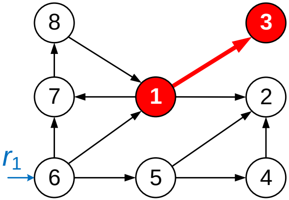


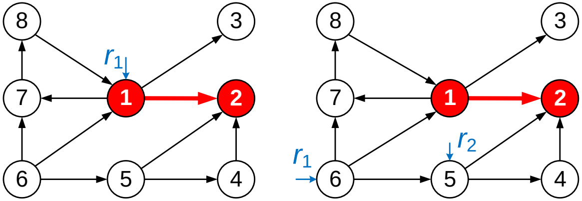
Target module: G_{21}

- Observe that $\mathcal{N}_1^+ = \{2, 3, 5, 8\}$, $\mathcal{N}_2^- \setminus \{1\} = \{3, 4, 6, 7\}$
- Find a minimum disconnecting set from $\mathcal{N}_1^+$ to $\mathcal{N}_2^- \setminus \{1\}$: $\mathcal{D} = \{3, 9\}$
- Excite $\mathcal{D}$ and node 1

Allocation of Excitation Signals - Special Cases

Find a disconnecting set $\mathcal{D}$ from $i \cup \bar{\mathcal{R}}$ to $\mathcal{N}_j^- \setminus i$ such that $b_{\bar{\mathcal{R}} \to \mathcal{D} \cup i} = |D| + 1$ holds for some excited nodes $\bar{\mathcal{R}}$.



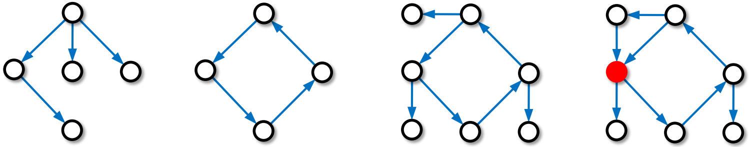


Case 1: $\mathcal{N}_j^- \setminus i = \emptyset$, i.e. i is the only in-neighbor of j
$\mathcal{D} = \emptyset$, then $b_{\bar{\mathcal{R}} \to i} = 1$.
Excite any node that has a path to i

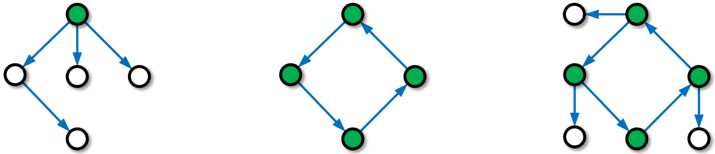
Case 2: there is no path from i to $\mathcal{N}_j^- \setminus i$
(1) Excite node i, $\mathcal{D} = \emptyset$, and $b_{\bar{\mathcal{R}} \to i} = 1$.
(2) Excite any node k having a path to i and
If k has a path to $\mathcal{N}_j^- \setminus i$, then excite a disconnecting set $\mathcal{D}$ from $\mathcal{N}_k^+$ to $\mathcal{N}_j^- \setminus i$

Pseudotrees

- A connected directed graph $\mathcal{T}$ is called a (directed) **pseudotree** if $|\mathcal{N}_i^-| \leq 1$, for every node i in $\mathcal{T}$.
  (including directed tree, directed cycle, or directed cycle with out-going branches)



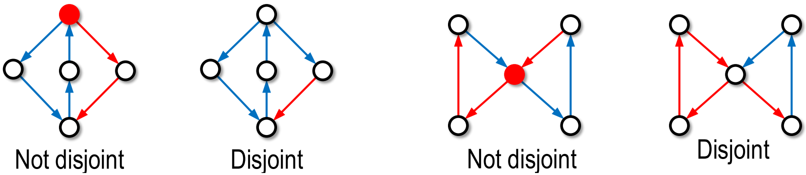
- **Roots**: nodes having directed paths to all nodes in the pseudotree



[5] [1]Cheng et al., CDC2019, [2] Cheng et al., TAC2022, to appear

Disjoint Pseudotrees

- A set of pseudotrees are called **disjoint** if
  1. The pseudotrees *do not share common edges*
  2. All the edges leaving from the same node are in the same pseudotree



- Any directed network can be decomposed into a set of disjoint pseudotrees

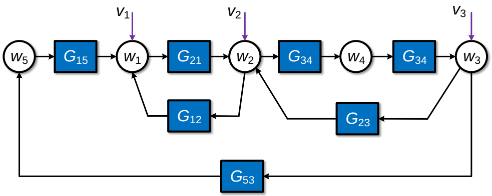
[5] [1]Cheng et al., CDC2019, [2] Cheng et al., TAC2022, to appear

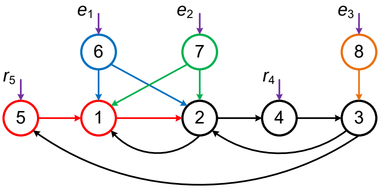
A Pseudotree Covering Condition (Full Network)

Theorem (Pseudo-tree covering condition)

A network model set $\mathcal{M}$ is generically identifiable if

- *there are a set of disjoint pseudotrees covering all the edges of $\mathcal{G}$, and*
- *and at least one of the roots in each pseudotree is an excited node.*





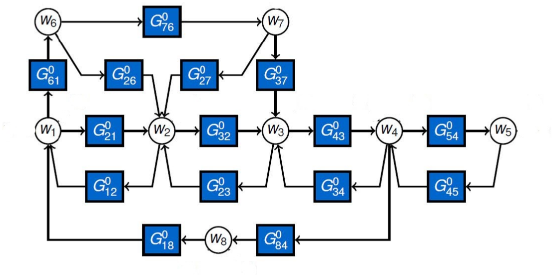
Generically identifiable!

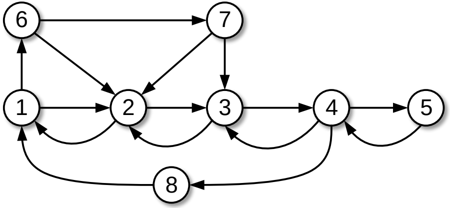
[5] [1]Cheng et al., CDC2019, [2] Cheng et al., TAC2022, to appear

Excitation Allocation in a Full Network

Procedure to allocate a minimum number of excitation signals:

- **Step 1**: Find a minimum number of disjoint pseudotrees that cover all the edges.
- **Step 2**: Assign an independent excitation signal at a root of each pseudotree.
- **Step 3**: Remove redundant excitation signals if generic identifiability remains.

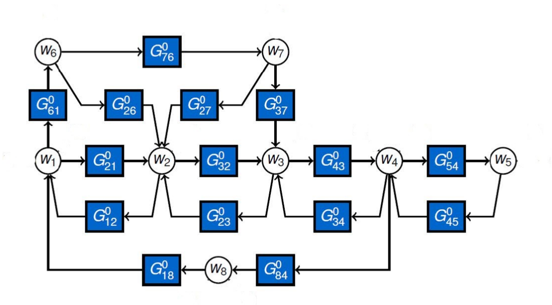


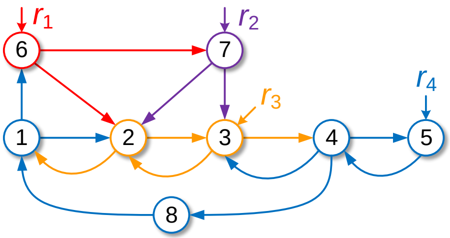


Excitation Allocation in a Full Network

Procedure to allocate a minimum number of excitation signals:

- **Step 1**: Find a minimum number of disjoint pseudotrees that cover all the edges.
- **Step 2**: Assign an independent excitation signal at a root of each pseudotree.
- **Step 3**: Remove redundant excitation signals if generic identifiability remains.

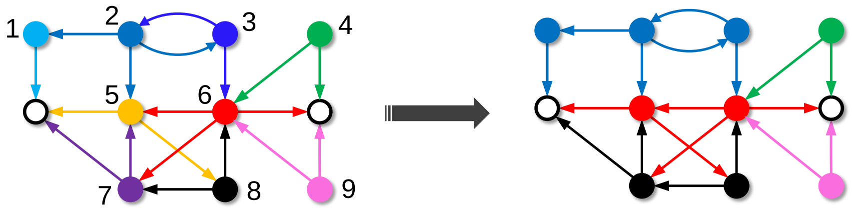




Pseudotree Merging Algorithm

Algorithm Steps:

1. Partition $\mathcal{G}$ into a set of minimal pseudotree.
   A minimal pseudotree contains one root and all the edges from it to its out-neighbors.
2. Iteratively merge two pseudotrees if they are mergeable.
   $\mathcal{T}_i$ is mergeable to $\mathcal{T}_j$ if their union remains a pseudotree, and the roots of $\mathcal{T}_j$ have paths to every node in $\mathcal{T}_i$. (e.g., $\mathcal{T}_1$ is mergeable to $\mathcal{T}_2$, but not vise versa)



[6] Any directed network can be covered by a set of disjoint minimal pseudotrees.

Mergeability

- Define the set $\mathbb{M} := \{1, \varnothing, 0\}$ with **operation rule** on $\mathbb{M}$:

$$1 \odot 1 = 1,\ 1 \odot 0 = 0,\ 1 \odot \varnothing = 1,\ 0 \odot 0 = 0,\ \varnothing \odot 0 = 0,\ \varnothing \odot \varnothing = \varnothing.$$

- **Mergeability matrix** $\mathcal{M}$:

$$\mathcal{M}_{ij} = \begin{cases} 1 & \text{if } \mathcal{T}_i \text{ is mergeable to } \mathcal{T}_j; \\ \varnothing & \text{if } \mathcal{T}_i \text{ and } \mathcal{T}_j \text{ do not share nodes;} \\ 0 & \text{otherwise.} \end{cases}$$

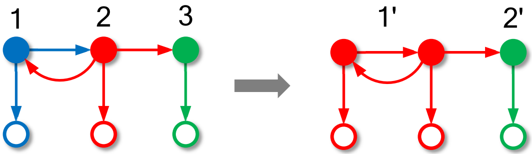
Mergeability

- Define the set $\mathbb{M} := \{1, \varnothing, 0\}$ with **operation rule** on $\mathbb{M}$:

$$1 \odot 1 = 1,\ 1 \odot 0 = 0,\ 1 \odot \varnothing = 1,\ 0 \odot 0 = 0,\ \varnothing \odot 0 = 0,\ \varnothing \odot \varnothing = \varnothing.$$

- **Mergeability matrix** $\mathcal{M}$:

$$\mathcal{M}_{ij} = \begin{cases} 1 & \text{if } \mathcal{T}_i \text{ is mergeable to } \mathcal{T}_j; \\ \varnothing & \text{if } \mathcal{T}_i \text{ and } \mathcal{T}_j \text{ do not share nodes;} \\ 0 & \text{otherwise.} \end{cases}$$



$$\mathcal{M} = \begin{bmatrix} 0 & 1 & \varnothing \\ 1 & 0 & 0 \\ \varnothing & 1 & 0 \end{bmatrix} \Rightarrow \hat{\mathcal{M}} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$0 \odot 1 \odot 1 \odot 0 = 0$

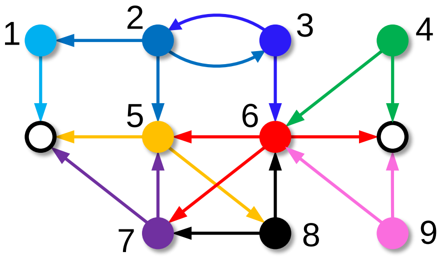
$\varnothing \odot 0 = 0$

$\varnothing \odot 1 = 1$

Algebraic operations on the matrix correspond to the merging in the graph!

Pseudotree Merging Algorithm

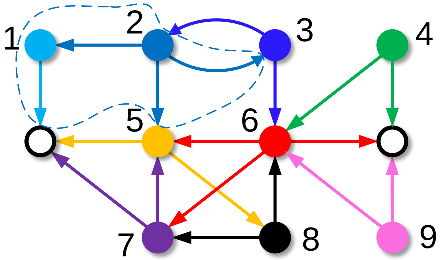
- **Procedure**: find an entry $\mathcal{M}_{ij} = 1$, and merge the i-th and j-th columns and rows
- **Policy**: First select $\mathcal{M}_{ij} = 1$ when row i has less "1" but more "$\varnothing$"
- **Stop**: if there is no "1" entry in $\mathcal{M}$



$$
\begin{array}{c}
\begin{array}{ccccccccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \mathcal{T}_5 & \mathcal{T}_6 & \mathcal{T}_7 & \mathcal{T}_8 & \mathcal{T}_9 \end{array} \\
\begin{pmatrix}
0 & 1 & \varnothing & \varnothing & 0 & \varnothing & 0 & \varnothing & \varnothing \\
0 & 0 & 1 & \varnothing & 0 & 0 & 0 & \varnothing & \varnothing \\
\varnothing & 1 & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
\varnothing & \varnothing & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
0 & 1 & \varnothing & \varnothing & 0 & 1 & 0 & 0 & \varnothing \\
\varnothing & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \varnothing & \varnothing & 0 & 0 & 0 & 1 & \varnothing \\
\varnothing & \varnothing & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\varnothing & \varnothing & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0
\end{pmatrix}
\begin{array}{c} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \\ \mathcal{T}_6 \\ \mathcal{T}_7 \\ \mathcal{T}_8 \\ \mathcal{T}_9 \end{array}
\end{array}
$$

Pseudotree Merging Algorithm

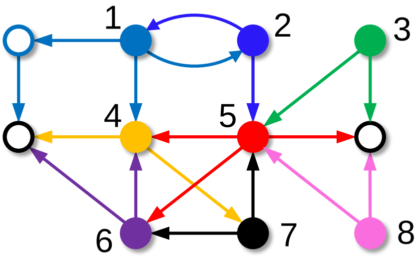
- **Procedure**: find an entry $\mathcal{M}_{ij} = 1$, and merge the i-th and j-th columns and rows
- **Policy**: First select $\mathcal{M}_{ij} = 1$ when row i has less "1" but more "$\varnothing$"
- **Stop**: if there is no "1" entry in $\mathcal{M}$



$$
\begin{array}{c c}
\begin{matrix} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \mathcal{T}_5 & \mathcal{T}_6 & \mathcal{T}_7 & \mathcal{T}_8 & \mathcal{T}_9 \end{matrix} & \\
\begin{pmatrix}
0 & 1 & \varnothing & \varnothing & 0 & \varnothing & 0 & \varnothing & \varnothing \\
0 & 0 & 1 & \varnothing & 0 & 0 & 0 & \varnothing & \varnothing \\
\varnothing & 1 & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
\varnothing & \varnothing & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
0 & 1 & \varnothing & \varnothing & 0 & 1 & 0 & 0 & \varnothing \\
\varnothing & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \varnothing & \varnothing & 0 & 0 & 0 & 1 & \varnothing \\
\varnothing & \varnothing & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\varnothing & \varnothing & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0
\end{pmatrix} &
\begin{matrix} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \\ \mathcal{T}_6 \\ \mathcal{T}_7 \\ \mathcal{T}_8 \\ \mathcal{T}_9 \end{matrix}
\end{array}
$$

Pseudotree Merging Algorithm

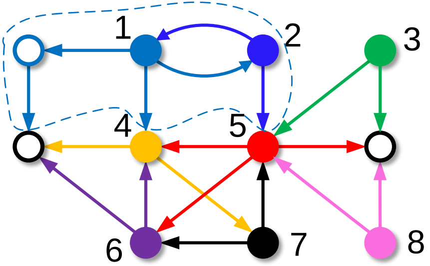
- **Procedure**: find an entry $\mathcal{M}_{ij} = 1$, and merge the i-th and j-th columns and rows
- **Policy**: First select $\mathcal{M}_{ij} = 1$ when row i has less "1" but more "$\varnothing$"
- **Stop**: if there is no "1" entry in $\mathcal{M}$



$$
\begin{array}{c c}
\begin{array}{cccccccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \mathcal{T}_5 & \mathcal{T}_6 & \mathcal{T}_7 & \mathcal{T}_8 \end{array} & \\
\begin{pmatrix}
0 & 1 & \varnothing & 0 & 0 & 0 & \varnothing & \varnothing \\
1 & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
\varnothing & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
0 & \varnothing & \varnothing & 0 & 1 & 0 & 0 & \varnothing \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \varnothing & \varnothing & 0 & 0 & 0 & 1 & \varnothing \\
\varnothing & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\varnothing & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0
\end{pmatrix} &
\begin{array}{c} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \\ \mathcal{T}_6 \\ \mathcal{T}_7 \\ \mathcal{T}_8 \end{array}
\end{array}
$$

Pseudotree Merging Algorithm

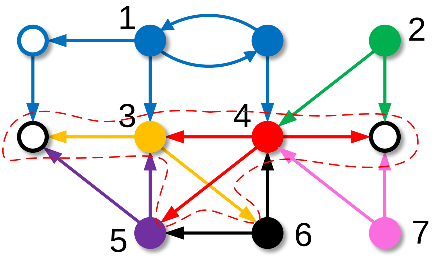
- **Procedure**: find an entry $\mathcal{M}_{ij} = 1$, and merge the i-th and j-th columns and rows
- **Policy**: First select $\mathcal{M}_{ij} = 1$ when row i has less "1" but more "$\varnothing$"
- **Stop**: if there is no "1" entry in $\mathcal{M}$



$$
\begin{array}{c}
\begin{array}{cccccccc}
\mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \mathcal{T}_5 & \mathcal{T}_6 & \mathcal{T}_7 & \mathcal{T}_8
\end{array} \\
\begin{pmatrix}
0 & 1 & \varnothing & 0 & 0 & 0 & \varnothing & \varnothing \\
1 & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
\varnothing & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
0 & \varnothing & \varnothing & 0 & 1 & 0 & 0 & \varnothing \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \varnothing & \varnothing & 0 & 0 & 0 & 1 & \varnothing \\
\varnothing & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\varnothing & 0 & 0 & \varnothing & 0 & \varnothing & 0 & 0
\end{pmatrix}
\end{array}
\begin{array}{c}
\\ \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \\ \mathcal{T}_6 \\ \mathcal{T}_7 \\ \mathcal{T}_8
\end{array}
$$

Pseudotree Merging Algorithm

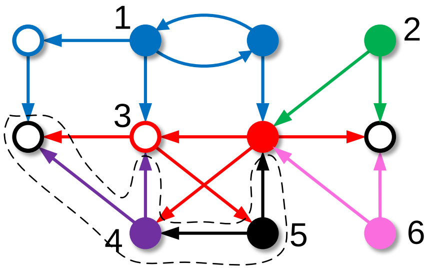
- **Procedure**: find an entry $\mathcal{M}_{ij} = 1$, and merge the i-th and j-th columns and rows
- **Policy**: First select $\mathcal{M}_{ij} = 1$ when row i has less “1” but more “$\varnothing$”
- **Stop**: if there is no “1” entry in $\mathcal{M}$



$$
\begin{array}{c}
\begin{array}{ccccccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \mathcal{T}_5 & \mathcal{T}_6 & \mathcal{T}_7 \end{array} \\
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \varnothing & 0 & \varnothing & 0 & 0 \\
0 & \varnothing & 0 & 1 & 0 & 0 & \varnothing \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \varnothing & 0 & 0 & 0 & 1 & \varnothing \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & \varnothing & 0 & \varnothing & 0 & 0
\end{pmatrix}
\end{array}
\begin{array}{c} \\ \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \\ \mathcal{T}_6 \\ \mathcal{T}_7 \end{array}
$$

Pseudotree Merging Algorithm

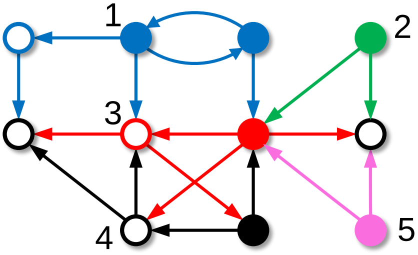
- **Procedure**: find an entry $\mathcal{M}_{ij} = 1$, and merge the i-th and j-th columns and rows
- **Policy**: First select $\mathcal{M}_{ij} = 1$ when row i has less "1" but more "$\varnothing$"
- **Stop**: if there is no "1" entry in $\mathcal{M}$



$$
\begin{array}{c c}
\begin{array}{cccccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \mathcal{T}_5 & \mathcal{T}_6 \end{array} & \\
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \varnothing & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & \varnothing & 0 & 0 & 1 & \varnothing \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \varnothing & 0 & 0
\end{pmatrix} &
\begin{array}{c} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \\ \mathcal{T}_6 \end{array}
\end{array}
$$

Pseudotree Merging Algorithm

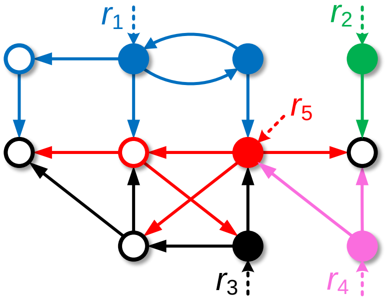
- **Procedure**: find an entry $\mathcal{M}_{ij} = 1$, and merge the i-th and j-th columns and rows
- **Policy**: First select $\mathcal{M}_{ij} = 1$ when row i has less "1" but more "$\varnothing$"
- **Stop**: if there is no "1" entry in $\mathcal{M}$



$$
\begin{array}{c|c}
 & \mathcal{T}_1 \quad \mathcal{T}_2 \quad \mathcal{T}_3 \quad \mathcal{T}_4 \quad \mathcal{T}_5 \\
\begin{matrix} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \end{matrix} &
\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}
\end{array}
$$

Excitation Allocation Based on Pseudotrees Covering

1. Excite a root in every pseudotree
2. Remove redundant excitation signals
   - r_k is redundant, if the network remains generic identifiability after removing r_k.
   - Only check $\mathcal{T}_k$ using the generic identifiability conditions based on vertex-disjoint paths

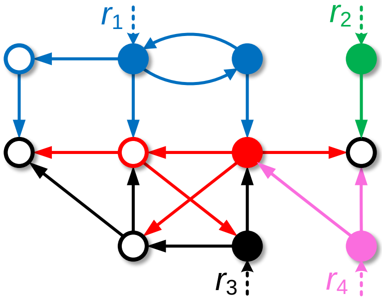


After removing r_5, the nodes in the red pseudotree satisfy the path-based condition

$\Rightarrow$ r_5 is redundant and thus can be removed

Excitation Allocation Based on Pseudotrees Covering

1. Excite a root in every pseudotree
2. Remove redundant excitation signals
   - r_k is redundant, if the network remains generic identifiability after removing r_k.
   - Only check $\mathcal{T}_k$ using the generic identifiability conditions based on vertex-disjoint paths



Generic identifiability with 4 excitations!

1. Problem Setting
2. Vertex-Disjoint Paths
3. Disconnecting Set
   - Generic Identifiability Analysis for Single Module
   - Excitation Allocation for Single Module Identifiability
4. Pseudotree Covering
   - Generic Identifiability Analysis for Full Network
   - Excitation Allocation For Full Network Identifiability
5. Conclusions

Conclusions

- **Graph-theoretical analysis** for identifiability of dynamic network
  (1) *vertex-disjoint paths*; (2) *disconnecting sets*; (3) *disjoint pseudotree covering*
- **Signal allocation** for generic identifiability: *minimum disconnecting set* (single module), *pseudotree merging algorithm* (full network)

Relevant works and generalizations

- Identifiability for more general model sets: correlated noises, non-parameterized (fixed) or switching modules [1], [2], [3]
- Identifiability results for dynamic networks in the setting of partial excitation and partial measurement [2], [4], [5]
- Conditions for global identifiability [6] and local generic identifiability [7]

[1] [1] Cheng et al., TAC2022; [2] Shi et al., arXiv:2008.01495; [3] Dreef et al. SYSID 2021 [4] Bazanella, CDC 2019; [5] Cheng et al., arXiv:2105.03187; [6] van Waarde et al., TAC 2019 [7] Legat et al., CDC2020

Relevant References

- A. S. Bazanella, M. Gevers, J. M. Hendrickx, and A. Parraga, “Identifiability of dynamical networks: which nodes need be measured?” in *Proc. IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 5870–5875.
- X. Cheng, S. Shi, and P. M. J. Van den Hof, “Allocation of excitation signals for generic identifiability of dynamic networks,” in *Proc. 58th IEEE Conf. Decision and Control*, 2019, pp. 5507–5512.
- ——, “Allocation of excitation signals for generic identifiability of linear dynamic networks,” *IEEE Transactions on Automatic Control*, vol. 67, no. 2, 2022.
- J. M. Hendrickx, M. Gevers, and A. S. Bazanella, “Identifiability of dynamical networks with partial node measurements,” *IEEE Transactions on Automatic Control*, vol. 64, no. 6, pp. 2240–2253, June 2019.
- A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*. Springer Science & Business Media, 2003, vol. 24.
- S. Shi, X. Cheng, and P. M. J. Van den Hof, “Excitation allocation for generic identifiability of a single module in dynamic networks: A graphic approach,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 40–45, 2020, Proc. 21th IFAC World Congress.
- ——, “Generic identifiability of subnetworks in a linear dynamic network: the full measurement case,” *arXiv preprint arXiv:2008.01495*, 2020.
- J. W. Van der Woude, “A graph-theoretic characterization for the rank of the transfer matrix of a structured system,” *Mathematics of Control, Signals and Systems*, vol. 4, no. 1, pp. 33–40, 1991.
- H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers, “Identifiability of linear dynamic networks,” *Automatica*, vol. 89, pp. 247–258, 2018.
- ——, “Single module identifiability in linear dynamic networks,” in *Proc. 57th IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4725–4730.

Thanks for your attention!