Allocation of Excitation Signals for Generic Identifiability of Dynamic Networks

Xiaodong Cheng Shengling Shi Paul M.J. Van den Hof

Control Systems Group, Department of Electrical Engineering, TU Eindhoven

IEEE Conference on Decision and Control
Nice, France December 13, 2019

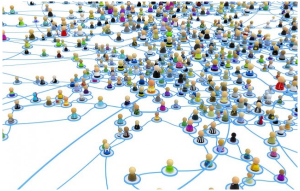
Outline

Introduction – dynamic networks

- Appear in a wide range of applications

Manufacturing

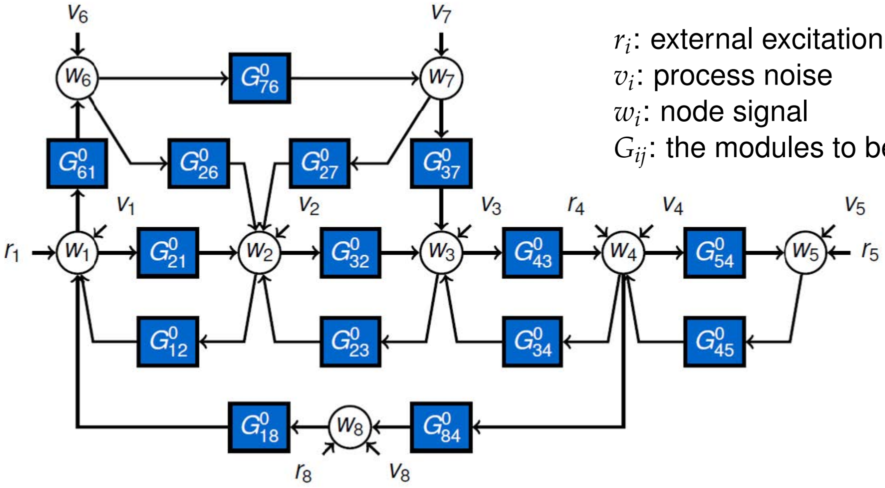
Power Grid



Social Network

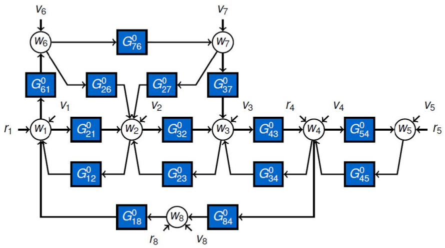
- How to address data-driven modeling problems in a network setting

Dynamic network setup



r_i: external excitation signals
v_i: process noise
w_i: node signal
G_{ij}: the modules to be identified

Dynamic network setup



Assume:

- G_{ij} are proper transfer functions
- the network is well-posed and stable

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 & G_{12} & \cdots & G_{1L} \\ G_{21} & 0 & \ddots & G_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ G_{L1} & G_{L2} & \cdots & G_{LL} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} + R \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_K \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_L \end{bmatrix}$$

Network identifiability

$$w = Gw + Rr + v, \text{ with } v = He,$$

- Transfer function: $T = (I - G)^{-1}[R \;\; H]$ typically can be identified from data

[1] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers, "Identifiability of linear dynamic networks," *Automatica* 2018.

Network identifiability

$$w = Gw + Rr + v, \text{ with } v = He,$$

- Transfer function: $T = (I - G)^{-1}[R \;\; H]$ typically can be identified from data
- A network model is denoted by $\sigma = (G, R, H)$;
- A *network model set* is defined by $\Sigma = \{\sigma(\theta) = (G(\theta), R(\theta), H(\theta)), \theta \in \Theta\}$

[1] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers, "Identifiability of linear dynamic networks," *Automatica* 2018.

Network identifiability

$$w = Gw + Rr + v, \text{ with } v = He,$$

- Transfer function: $T = (I - G)^{-1}[R \;\; H]$ typically can be identified from data
- A network model is denoted by $\sigma = (G, R, H)$;
- A *network model set* is defined by $\Sigma = \{\sigma(\theta) = (G(\theta), R(\theta), H(\theta)), \theta \in \Theta\}$

Definition (Network identifiability[1])

Σ *is identifiable if* $T(\theta_1) = T(\theta_0) \Rightarrow \sigma(\theta_1) = \sigma(\theta_0)$ *holds for all* $\theta_0, \theta_1 \in \Theta$*.*

Generic identifiability holds if this is true for almost all models in Σ (except models in the measure zero set).

[1] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers, "Identifiability of linear dynamic networks," *Automatica* 2018.

Problem

Consider the network $\mathcal{G}$, and all the node signals w_j are measured.

Problem (Generic identifiability condition)

Under what conditions(i.e., the number and locations of external excitation signals), Σ is generically identifiable?

Problem (Allocation of excitation signals)

Where to put external excitation signals in $\mathcal{G}$ to achieve generic identifiability of Σ.

*For simplicity, we will not consider noises, and all the modules are to be identified.

Generic identifiability - revisit

Lemma (Path-based condition[2,3])

Let $\mathcal{R} \subseteq \mathcal{V}$ be the set of vertices that are excited. Σ is generically identifiable if and only if the maximum number of vertex disjoint paths from $\mathcal{R}$ to $\mathcal{N}_i^-$ is equal to $|\mathcal{N}_i^-|$ for all $i \in \mathcal{V}$.

Relevant notations:

- $\mathcal{N}_i^-$: the set of the in-neighbors of vertex i;
- $|\mathcal{N}_i^-|$ the cardinality of the set $\mathcal{N}_i^-$.

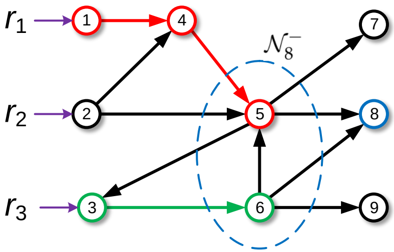
[2] J. M. Hendrickx, M. Gevers, and A. S. Bazanella, "Identifiability of dynamical networks with partial node measurements," *IEEE TAC*, 2018.

[3] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers, "Single module identifiability in linear dynamic networks," *CDC2018*.

Generic identifiability - revisit

Lemma (Path-based condition)

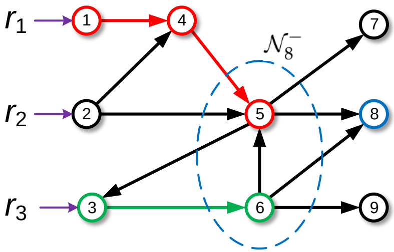
Let $\mathcal{R} \subseteq \mathcal{V}$ be the set of vertices that are excited. Σ is generically identifiable if and only if the maximum number of vertex disjoint paths from $\mathcal{R}$ to $\mathcal{N}_i^-$ is equal to $|\mathcal{N}_i^-|$ for all $i \in \mathcal{V}$.



Generic identifiability - revisit

Lemma (Path-based condition)

Let $\mathcal{R} \subseteq \mathcal{V}$ be the set of vertices that are excited. Σ is generically identifiable if and only if the maximum number of vertex disjoint paths from $\mathcal{R}$ to $\mathcal{N}_i^-$ is equal to $|\mathcal{N}_i^-|$ for all $i \in \mathcal{V}$.

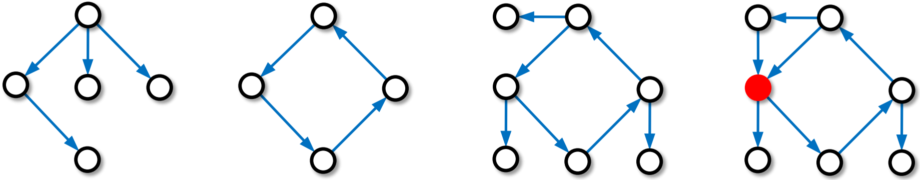


Result provides an analysis tool, but is less suited for the signal allocation problem

Pseudo-trees

Definition (Directed pseudo-trees)

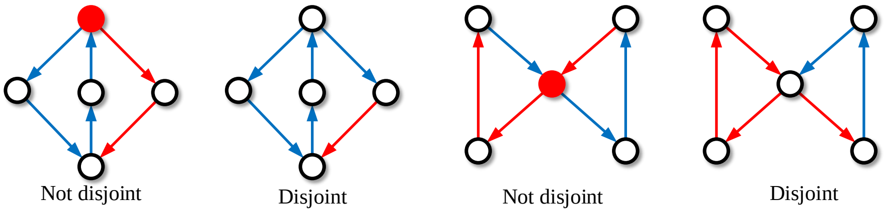
A connected directed graph $\mathcal{T}$ is called a (directed) pseudo-tree if $|\mathcal{N}_i^-| \leq 1$, for all $i \in V(\mathcal{T})$.



Disjoint pseudo-trees

Two pseudo-trees are (edge)-disjoint if

- Any two pseudo-trees should not share common edges
- Each node and its out-going edges are in the same pseudo-tree



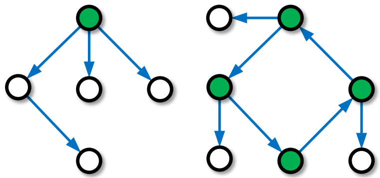
Any directed network can be decomposed into a set of disjoint pseudo-trees

Generic identifiability - new condition

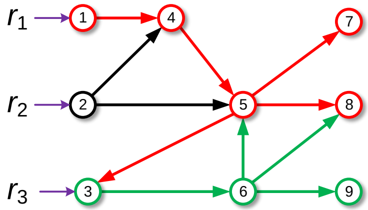
Theorem (Pseudo-tree covering condition[4])

A network is generically identifiable if

- *it can be decomposed into K disjoint pseduotrees,*
- *and there are K independent excitation signals allocated at a root of each pseudotree.*



Left: Tree with root in green
Right: Cycle with outgoing trees; Any node in cycle is root



[4] X. Cheng, S. Shi, and P. M. J. Van den Hof, "Allocation of excitation signals for generic identifiability of dynamic networks," *CDC2019*

Excitation allocation

Excitation allocation problem

Given a directed network, what is the minimal number of excitation signals for the network to be generic identifiable?

Excitation allocation

Excitation allocation problem

Given a directed network, what is the minimal number of excitation signals for the network to be generic identifiable?

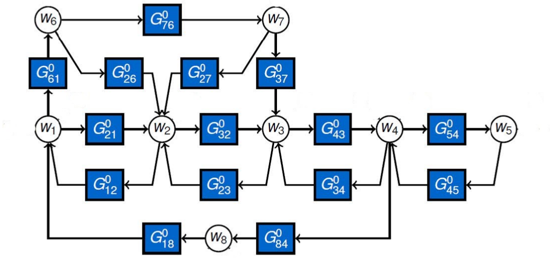
Now, we solve

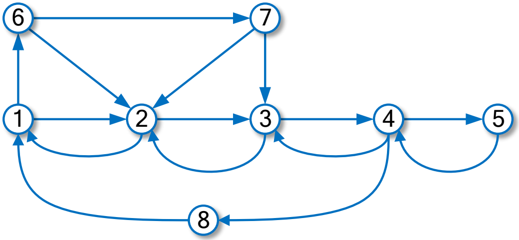
Graph covering problem

Given a directed network, find the minimal number of disjoint pseudo-trees such that all the edges are covered.

Excitation allocation

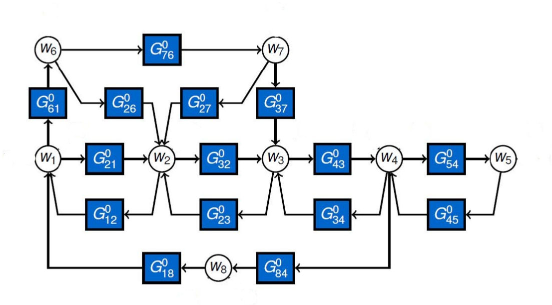
For simple network, it is easy:

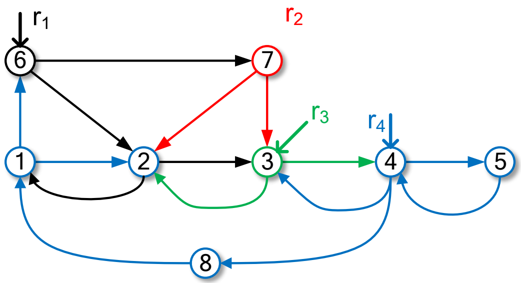




Excitation allocation

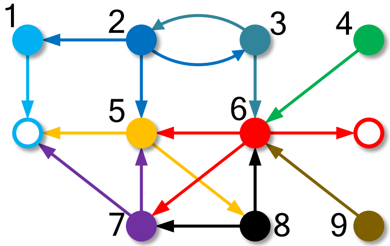
For simple network, it is easy:





Pseudo-tree merging

- A *minimal* pseudo-tree is a graph which only contains one root and all the out-neighbors of this root.
- Any directed network can be covered by a set of disjoint minimal pseudo-trees.



We start with a covering of $\mathcal{G}$ with minimal pseudo-trees.

Mergeability

Definition (Mergeability)

Consider two disjoint pseudo-trees $\mathcal{T}_1$ and $\mathcal{T}_2$. We say $\mathcal{T}_1$ is mergeable to $\mathcal{T}_2$, if

1. *the union of $\mathcal{T}_1$ and $\mathcal{T}_2$ is also a pseudo-tree;*
2. *there is a directed path from the roots of $\mathcal{T}_2$ to each node in $\mathcal{T}_1$.*

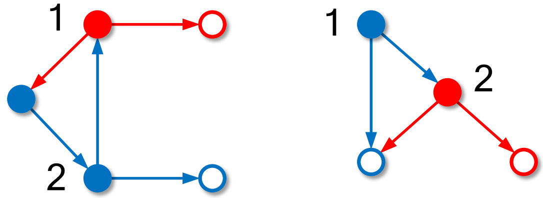


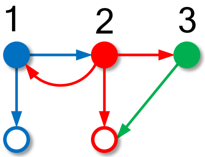
Figure: Left: $\mathcal{T}_1$ is mergeable to $\mathcal{T}_2$, and $\mathcal{T}_2$ is mergeable to $\mathcal{T}_1$
Right: $\mathcal{T}_1$ is NOT mergeable to $\mathcal{T}_2$, and $\mathcal{T}_2$ is mergeable to $\mathcal{T}_1$

Mergeability

Definition (Mergeability matrix)

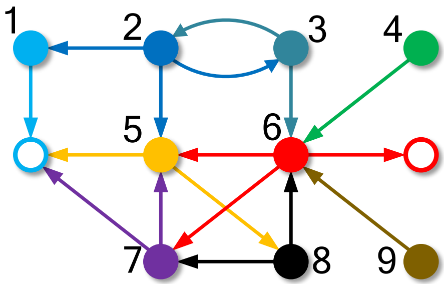
Denote $\mathbb{M} := \{1, \varnothing, 0\}$. *The mergeability matrix* $\mathcal{M}$ *is defined with the* (i,j)*-th entry as*

$$\mathcal{M}(i,j) = \begin{cases} 1 & \text{if } \mathcal{T}_i \text{ is mergeable to } \mathcal{T}_j; \\ \varnothing & \text{if } V(\mathcal{T}_i) \cap V(\mathcal{T}_j) = \emptyset; \\ 0 & \text{otherwise.} \end{cases}$$



$$\begin{array}{cc} & \begin{array}{ccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 \end{array} \\ \mathcal{M} = & \begin{pmatrix} 0 & 1 & \varnothing \\ 1 & 0 & 0 \\ \varnothing & 0 & 0 \end{pmatrix} \begin{array}{c} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \end{array} \end{array}$$

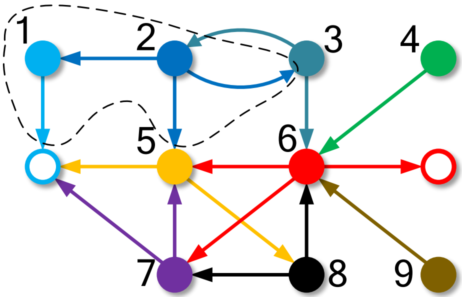
Pseudo-tree merging - step 1



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

- Find a row of $\mathcal{M}$ with only a single “1” entry, while the others are either “0” or “$\varnothing$”.
- If there exist multiple rows containing a single “1” entry, then we choose the one with more “$\varnothing$” entries.

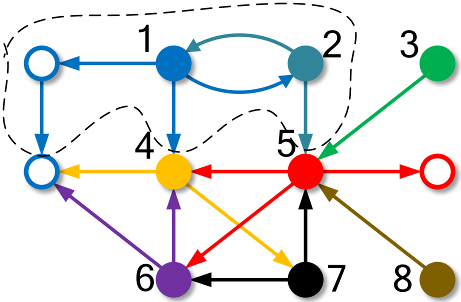
Pseudo-tree merging - step 1



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

- Find a row of $\mathcal{M}$ with only a single "1" entry, while the others are either "0" or "$\varnothing$".
- If there exist multiple rows containing a single "1" entry, then we choose the one with more "$\varnothing$" entries.

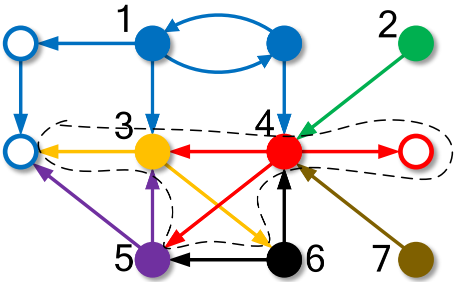
Pseudo-tree merging - step 1



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

- Find a row of $\mathcal{M}$ with only a single “1” entry, while the others are either “0” or “$\varnothing$”.
- If there exist multiple rows containing a single “1” entry, then we choose the one with more “$\varnothing$” entries.

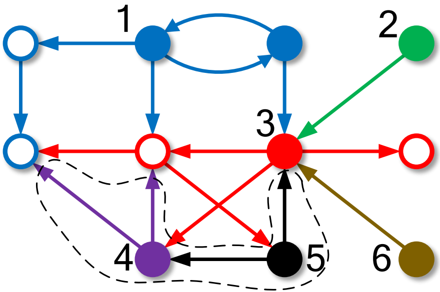
Pseudo-tree merging - step 1



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

- Find a row of $\mathcal{M}$ with only a single "1" entry, while the others are either "0" or "$\varnothing$".
- If there exist multiple rows containing a single "1" entry, then we choose the one with more "$\varnothing$" entries.

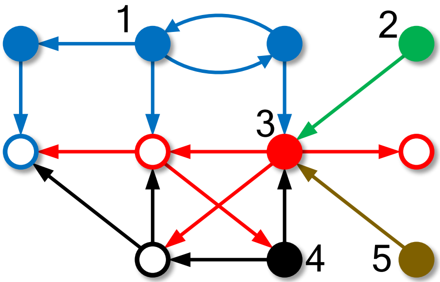
Pseudo-tree merging - step 1



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

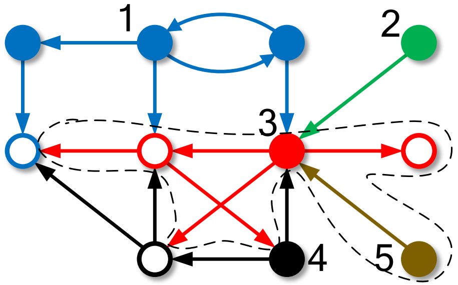
- Find a row of $\mathcal{M}$ with only a single “1” entry, while the others are either “0” or “$\varnothing$”.
- If there exist multiple rows containing a single “1” entry, then we choose the one with more “$\varnothing$” entries.

Pseudo-tree merging - step 1



$$
\begin{array}{cc}
\begin{array}{ccccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \mathcal{T}_5 \end{array} & \\
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix} & \begin{array}{c} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \end{array}
\end{array}
$$

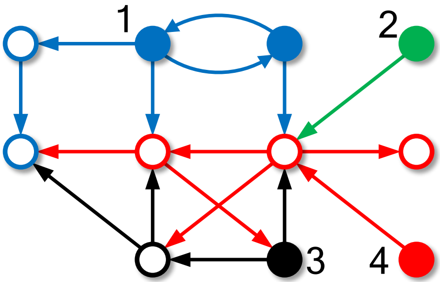
Pseudo-tree merging - step 2



$$\begin{array}{cc} \begin{array}{ccccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \mathcal{T}_5 \end{array} & \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \boxed{1} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \\ \mathcal{T}_5 \end{array} \end{array}$$

- Find a row with at least one “1” entry, and then merge the two pseudo-trees corresponding to an arbitrary “1” entry of this row.
- If there exists multiple rows containing “1” entries, select a row with more “$\varnothing$” entries

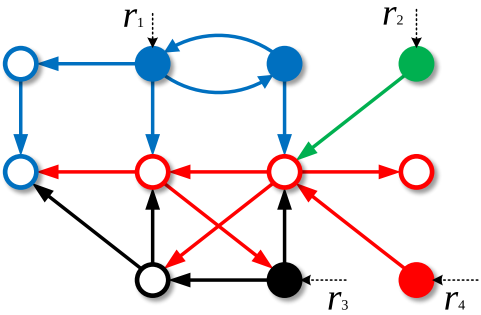
Pseudo-tree merging - step 2



$$\begin{array}{ccccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 & \\ \multicolumn{4}{c}{\multirow{4}{*}{$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$}} & \mathcal{T}_1 \\ &&&& \mathcal{T}_2 \\ &&&& \mathcal{T}_3 \\ &&&& \mathcal{T}_4 \end{array}$$

- There are no mergeable pairs of pseudo trees $\Rightarrow$ end of the algorithm
- Allocate the external excitation signals to the roots

Pseudo-tree merging - step 2



$$\begin{array}{cc} \begin{array}{cccc} \mathcal{T}_1 & \mathcal{T}_2 & \mathcal{T}_3 & \mathcal{T}_4 \end{array} & \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} \mathcal{T}_1 \\ \mathcal{T}_2 \\ \mathcal{T}_3 \\ \mathcal{T}_4 \end{array} \end{array}$$

- There are no mergeable pairs of pseudo trees $\Rightarrow$ end of the algorithm
- Allocate the external excitation signals to the roots

Conclusions

- A graphical condition to characterize the generic identifiability of dynamic network using disjoint pseudo-trees
- An greedy pseudo-tree merging algorithm to solve excitation allocation problem

Extensions

- correlated noises, non-parameterized modules [5]
- better algorithm for the decomposition of a graph into pseudo-trees
- a graph-based condition for the partial measurement case

Thank you for your attention!
Questions?

[5] X. Cheng, S. Shi, and P. M. J. Van den Hof, "Allocation of excitation signals for generic identifiability of linear dynamic networks," *Submitted*