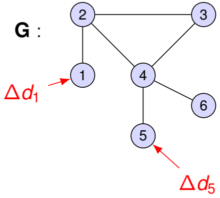
ETH zürich

Control of networked systems by clustering: the degree of freedom concept

Andrea Martinelli • John Lygeros
Automatic Control Laboratory • ETH Zurich
21st IFAC World Congress • July 11-17, 2020

Disturbance suppression in graphs

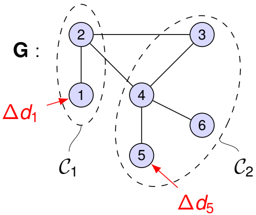
Uncontrolled propagation of local disturbances may lead to catastrophic effects on the dynamics of the whole network[1,2]



[1] Sandell et al. (1978) *IEEE TAC* [2] Hespanha et al. (2007) *Proc. of the IEEE*

Disturbance suppression in graphs

Uncontrolled propagation of local disturbances may lead to catastrophic effects on the dynamics of the whole network[1,2]

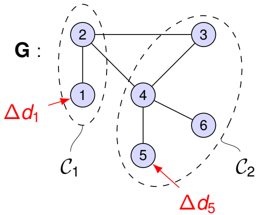


Idea: define a partition $\mathcal{P} = \{\mathcal{C}_1, \dots, \mathcal{C}_m\}$ on **G** such that each cluster can locally adsorb disturbances.

[1] Sandell et al. (1978) *IEEE TAC*   [2] Hespanha et al. (2007) *Proc. of the IEEE*

Disturbance suppression in graphs

Uncontrolled propagation of local disturbances may lead to catastrophic effects on the dynamics of the whole network[1,2]

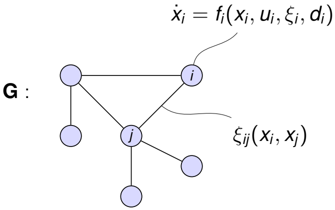


Idea: define a partition $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ on **G** such that each cluster can locally adsorb disturbances.

- What does it mean for a cluster to be structurally inclined to suppress disturbances?
- What is the best way to partition a graph according to our needs?

[1] Sandell et al. (1978) *IEEE TAC* [2] Hespanha et al. (2007) *Proc. of the IEEE*

Networked control system



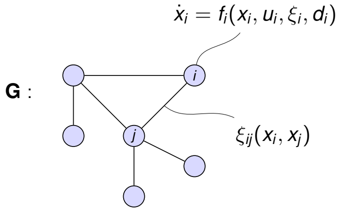
Each node in the graph **G** hosts a dynamical system

$$\dot{x}_i = f_i\left(x_i, u_i, \xi_i, d_i\right),$$

where $x_i \in \mathbb{R}^{N_i}$, $u_i, \xi_i, d_i \in \mathbb{R}$, and

$$\xi_i = \sum_{j \in \mathcal{N}_i} \xi_{ij}(x_i, x_j).$$

Networked control system



Each node in the graph **G** hosts a dynamical system

$$\dot{x}_i = f_i\left(x_i, u_i, \xi_i, d_i\right),$$

where $x_i \in \mathbb{R}^{N_i}$, $u_i, \xi_i, d_i \in \mathbb{R}$, and

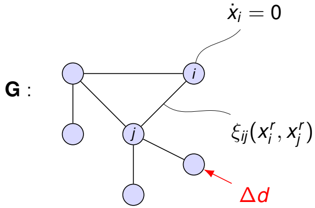
$$\xi_i = \sum_{j \in \mathcal{N}_i} \xi_{ij}(x_i, x_j).$$

Objective: design decentralized maps $u_i = g_i(x_i, x_{j \in \mathcal{N}_i})$ such that

- d_i are locally compensated, and
- $x_i = x_i^r$ in spite of the couplings

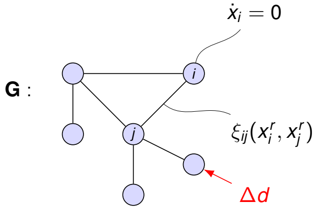
Local flux redistribution

- Saturations, control unbalance,... call for a higher-layer architecture.



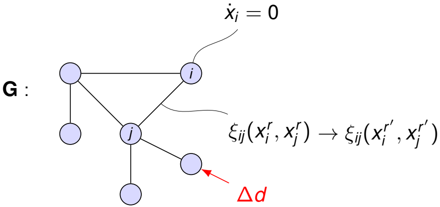
Local flux redistribution

- Saturations, control unbalance,... call for a higher-layer architecture.
- In this context, disturbances are suppressed by a proper choice of x^r.



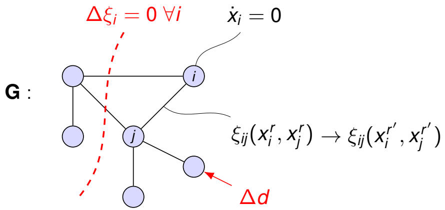
Local flux redistribution

- Saturations, control unbalance,... call for a higher-layer architecture.
- In this context, disturbances are suppressed by a proper choice of x^r.
- The effect of a reference update $x^r \to x^{r'}$ is a flux redistribution on the archs.



Local flux redistribution

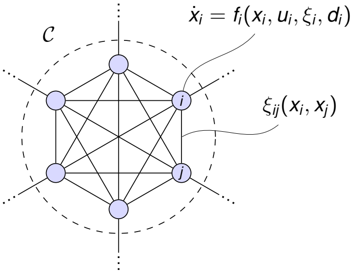
- Saturations, control unbalance,... call for a higher-layer architecture.
- In this context, disturbances are suppressed by a proper choice of x^r.
- The effect of a reference update $x^r \to x^{r'}$ is a flux redistribution on the archs.



Objective: impose a **local** reference update to a subset of nodes, without affecting any external node $\to$ zero inter-cluster flux variation.

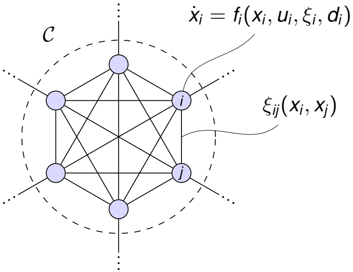
Graph clustering methods

- Define a measure $m : \mathcal{C} \to \mathbb{R}$ that captures your needs, and
- Find the partition $\mathcal{P}^*$ that maximises the aggregated utility



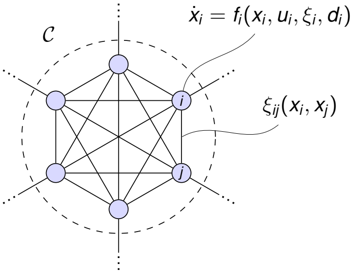
Graph clustering methods

- Define a measure $m : \mathcal{C} \to \mathbb{R}$ that captures your needs, and
- Find the partition $\mathcal{P}^*$ that maximises the aggregated utility
- Our clusters need to have dense internal circulation with zero external Δflux.



Graph clustering methods

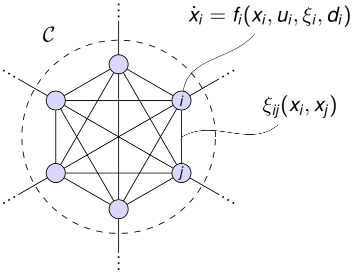
- Define a measure $m : \mathcal{C} \to \mathbb{R}$ that captures your needs, and
- Find the partition $\mathcal{P}^*$ that maximises the aggregated utility
- Our clusters need to have dense internal circulation with zero external Δflux.



Traditional clustering methods: *modularity, persistence probability, spectral analysis,...*

Graph clustering methods

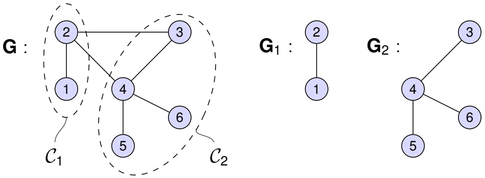
- Define a measure $m : \mathcal{C} \to \mathbb{R}$ that captures your needs, and
- Find the partition $\mathcal{P}^*$ that maximises the aggregated utility
- Our clusters need to have dense internal circulation with zero external Δflux.



Traditional clustering methods: *modularity, persistence probability, spectral analysis,...* do not capture the structural condition we require.

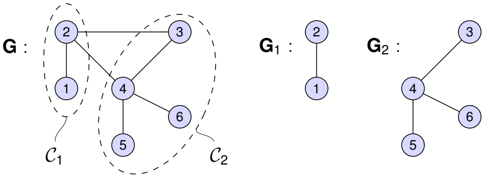
$\rightarrow$ Need for a **new clustering measure**!

Partitioned graph Laplacian (1/2)



$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

Partitioned graph Laplacian (1/2)

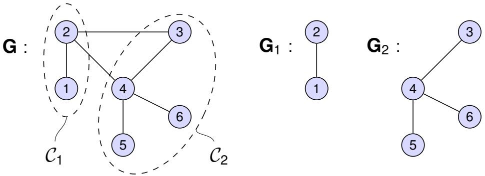


$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

Assumption 1. The graph **G** is connected

Assumption 2. The induced subgraphs identified by the clusters in the partition $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m\}$ are connected. No trivial clusters.

Partitioned graph Laplacian (1/2)



$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

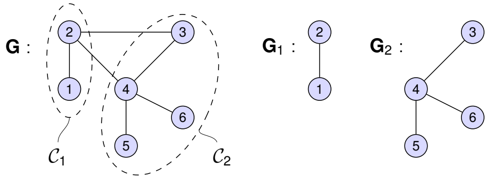
Assumption 1. The graph **G** is connected

Assumption 2. The induced subgraphs identified by the clusters in the partition $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$ are connected. No trivial clusters.

Theorem 1

If Assumptions 1 and 2 are satisfied, then any diagonal block of the Laplacian associated with a cluster in $\mathcal{P}$ is nonsingular and has all eigenvalues with positive real part.

Rank difference function

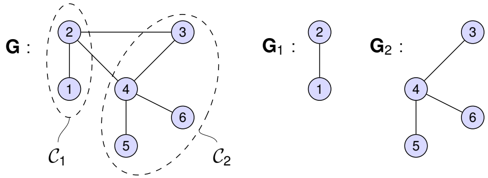


$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

- Let's introduce a rank difference function $\delta : \mathbb{R}^{a \times b} \times \mathbb{R}^{c \times d} \rightarrow \mathbb{N}$ and apply it to a cluster $\mathcal{C}_i$

$$\delta(C_i, Y_i) = \text{rank}(C_i) - \text{rank}(Y_i) = d_i$$

Rank difference function



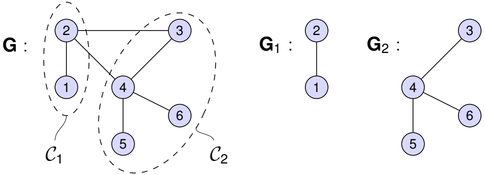
$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

- Let's introduce a rank difference function $\delta : \mathbb{R}^{a \times b} \times \mathbb{R}^{c \times d} \to \mathbb{N}$ and apply it to a cluster $\mathcal{C}_i$

$$\delta(C_i, Y_i) = \text{rank}(C_i) - \text{rank}(Y_i) = d_i \in \big[\, 0, |C_i| \,\big)$$

In fact, C_i is full-rank (Theorem 1) and C_i and Y_i share the same number of rows.

Rank difference function



$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

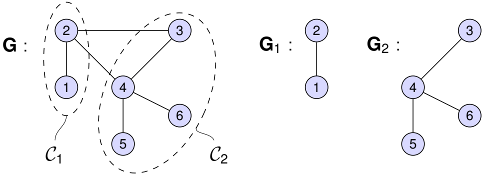
- Let's introduce a rank difference function $\delta : \mathbb{R}^{a \times b} \times \mathbb{R}^{c \times d} \to \mathbb{N}$ and apply it to a cluster $\mathcal{C}_i$

$$\delta(C_i, Y_i) = \text{rank}(C_i) - \text{rank}(Y_i) = d_i \in \big[\, 0, |C_i| \,\big)$$

  In fact, C_i is full-rank (Theorem 1) and C_i and Y_i share the same number of rows.

- We say that a cluster $\mathcal{C}_i$ has d_i **degrees of freedom** if $\delta(C_i, Y_i) = d_i$

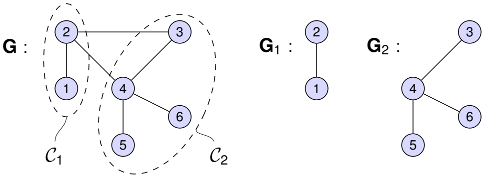
Degrees of freedom for a cluster



$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

- **What does d_i mean?** Fraction of nodes that can modify their state without incurring in a net Δflux with the rest of the network. Example: $d_1 = 1$ and $d_2 = 3$.

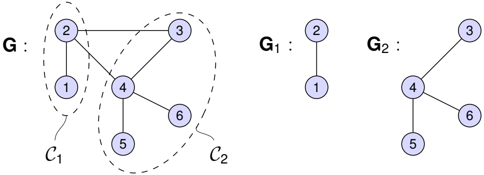
Degrees of freedom for a cluster



$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

- **What does d_i mean?** Fraction of nodes that can modify their state without incurring in a net Δflux with the rest of the network. Example: $d_1 = 1$ and $d_2 = 3$.
- We can use δ as a clustering measure to generate optimal partitions

Degrees of freedom for a cluster



$$L = \begin{pmatrix} C_1 & Y_{12} & \cdots & Y_{1m} \\ Y_{12}^\top & C_2 & & \\ \vdots & & \ddots & \\ Y_{1m}^\top & & & C_m \end{pmatrix}$$

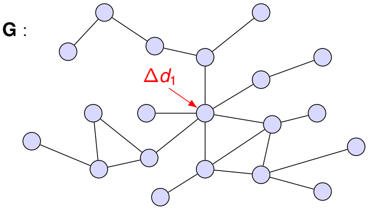
- **What does d_i mean?** Fraction of nodes that can modify their state without incurring in a net Δflux with the rest of the network. Example: $d_1 = 1$ and $d_2 = 3$.
- We can use δ as a clustering measure to generate optimal partitions
- The possible partitions of a set scale according to the Bell numbers (e.g. $B_{20} \approx 10^{12}$) $\rightarrow$ heuristic algorithms are needed.

Example: DC microgrids voltage control

The typical converter-based model of a DGU is a linear affine system

$$\dot{x}_i = f_i\left(x_i, u_i, \xi_i, d_i\right)$$

$$= \begin{bmatrix} 0 & \frac{1}{C_i} \\ -\frac{1}{L_i} & -\frac{R_i}{L_i} \end{bmatrix} x_i + \begin{bmatrix} 0 \\ V_{in,i} \end{bmatrix} u_i + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \xi_i - \begin{bmatrix} 1 \\ 0 \end{bmatrix} d_i$$



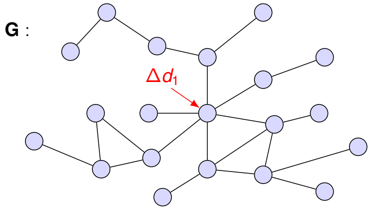
[1] Martinelli et al. (2019), *ECC*

Example: DC microgrids voltage control

The typical converter-based model of a DGU is a linear affine system

$$\dot{x}_i = f_i\left(x_i, u_i, \xi_i, d_i\right)$$

$$= \begin{bmatrix} 0 & \frac{1}{C_i} \\ -\frac{1}{L_i} & -\frac{R_i}{L_i} \end{bmatrix} x_i + \begin{bmatrix} 0 \\ V_{in,i} \end{bmatrix} u_i + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \xi_i - \begin{bmatrix} 1 \\ 0 \end{bmatrix} d_i$$



Passivity-based primary controllers[1]

$$u_i = K_i x_i, \quad K_i \in \mathcal{K}(\epsilon_i),$$

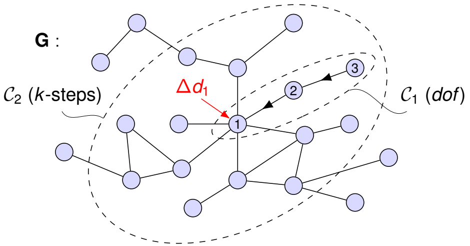
guarantee local reference tracking $x_i = x_i^r$. What happens when a disturbance cannot be internally compensated?

[1] Martinelli et al. (2019), *ECC*

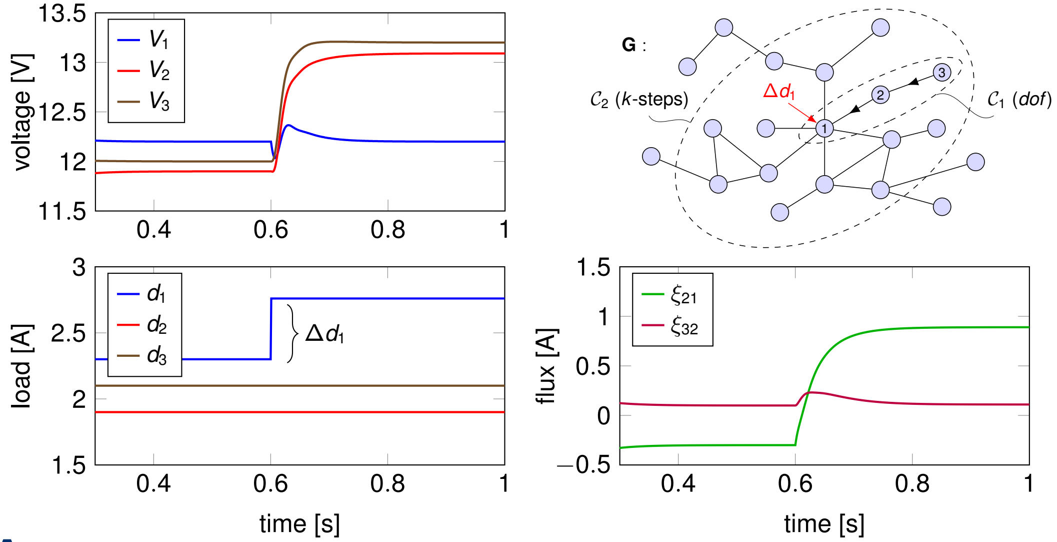
Heuristic clustering strategy

Algorithm 1 Greedy exploration

given overloading node i with set of neighbors $\mathcal{N}_i$
initialize $\mathcal{C} = \{i\}, \mathcal{N}_\mathcal{C} = \mathcal{N}_i, \delta_\mathcal{C} = 0$
repeat
  $\mathcal{H} = \{j \in \mathcal{N}_\mathcal{C} \;:\; \delta_{\mathcal{C},j} > \delta_\mathcal{C}\}$
  if $\mathcal{H} \neq \emptyset$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \left\{j = \arg\max_{j\in\mathcal{H}}(\Psi_j)\right\}$
  else
    $\mathcal{C} \leftarrow \mathcal{C} \cup \left\{j = \arg\max_{j\in\mathcal{N}_\mathcal{C}}(\deg(j))\right\}$
  end if
  $\mathcal{N}_\mathcal{C} \leftarrow (\mathcal{N}_\mathcal{C} \cup \mathcal{N}_j) \setminus (j \cup (\mathcal{C} \cap \mathcal{N}_j))$
  solve optimization problem for nodes in $\mathcal{C}$
until a feasible solution is found
return $x^r_\mathcal{C}$



Voltage tracking in a microgrid



Thank you for the attention!

Contact: andremar@control.ee.ethz.ch