Surveillance Distribuée de Systèmes Multi-agents

UPJV, Département EEA

Master 2 3EA, EC53

UE alternants

Fabio MORBIDI

Laboratoire MIS
Équipe Perception Robotique
fabio.morbidi@u-picardie.fr

Mardi, Mercredi et Jeudi
13h30-16h00 (CM et TD salle CURI 305)
13h30-17h30 (TP salle CURI 305)

AU 2022-2023

UNIVERSITÉ de Picardie Jules Verne

Plan du cours

Chapitre 1: **Modélisation de l’incertitude**

1. Introduction
2. Représentation de l’erreur
3. Incertitude d’un capteur
4. Propagation d’incertitude

Chapitre 2: **Traitement des mesures**

1. Réseaux multi-capteurs
2. Fusion des mesures

Chapitre 3: **Estimation distribuée**

1. Introduction à la théorie des graphes
2. Protocole de consensus
3. Dynamic average consensus estimators
4. Méthode des moindres carrés distribuée
5. Filtre de Kalman distribué

Découverte des laboratoires: Activités de recherche du laboratoire MIS

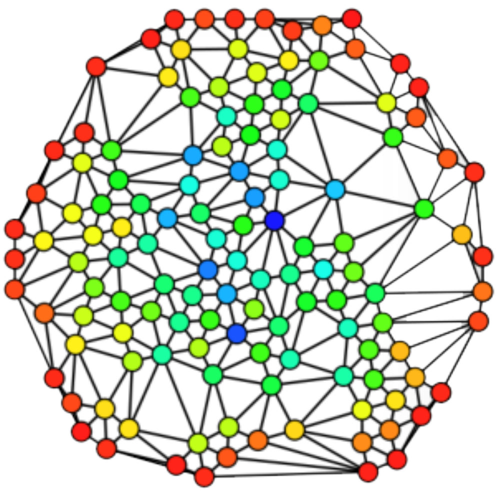
Ch. 3: Estimation distribuée

- Introduction à la théorie des graphes **Partie 1**
- Protocole de consensus **Partie 2**
- Dynamic average consensus estimators **Partie 3**
- Méthode des moindres carrés distribuée **Partie 4**
- Filtre de Kalman distribué **Partie 5**

Introduction à la théorie des graphes

- Les graphes fournissent une **abstraction naturelle** pour l'échange d'informations entre les agents dans un réseau
- L'abstraction basée sur la théorie des graphes fournit une description à **haut niveau** de la topologie du réseau par rapport à des objets appelés *sommets* et *arêtes*

Dans ce qui suit, nous verrons quelques notions de base de:



- **Théorie algébrique des graphes**
  (matrice d'adjacence, matrice des dégres, matrice d'incidence, matrice laplacienne)
- **Théorie spectrale des graphes**
  (spectre de la matrice laplacienne)

Théorie des graphes

- Un graphe *fini*, *non orienté*, *simple*, ou **graphe** en bref, est fondé sur l‘ensemble fini:

  $V = \{v_1, v_2, \ldots, v_n\}$ l’ensemble des **sommets** (ou **nœuds**)

- On définit aussi l‘ensemble des **arêtes** (ou **arcs**):

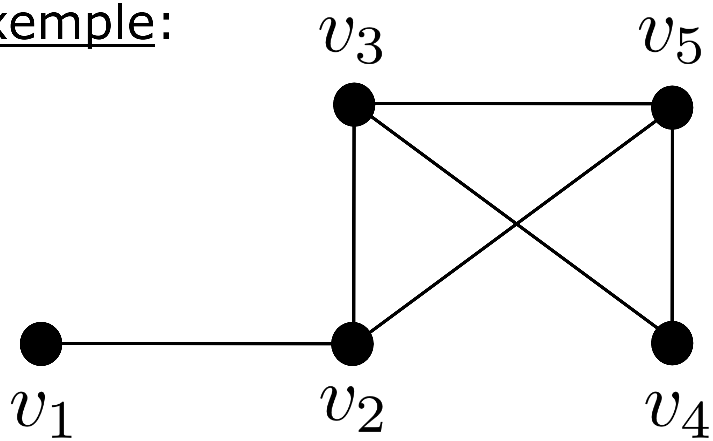
  $E \subseteq V \times V$ Il consiste en éléments de la forme $\{v_i, v_j\}$ ou $v_i v_j$ tels que $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$.

- Le graphe $\mathcal{G}$ est alors défini formellement comme le couple:

$$\mathcal{G} = (V, E)$$

Théorie des graphes

Exemple:



$$\mathcal{G} = (V,\, E)$$

$$V = \{v_1,\, v_2, \ldots,\, v_5\}$$

$$E = \{v_1v_2, v_2v_3, v_3v_4, v_3v_5, v_2v_5, v_4v_5\}$$

- S'il y a une arête entre deux sommets v_i et v_j, ils sont appelés **adjacents** qui s'écrit $v_i \sim v_j$. Dans ce cas, v_iv_j est appelée **incidente** avec sommets v_i, v_j
- Le **voisinage** $\mathcal{N}(i) \subseteq V$ du sommet v_i est l'ensemble:

$$\{v_j \in V \,|\, v_iv_j \in E\}$$

c'est-à-dire, l'ensemble des sommets adjacents à v_i.
Dans l'exemple ci-dessus: $\mathcal{N}(5) = \{v_2,\, v_3,\, v_4\}$

Théorie des graphes

- Un **chemin de longueur** m dans $\mathcal{G}$ est donné par la séquence de sommets distincts:

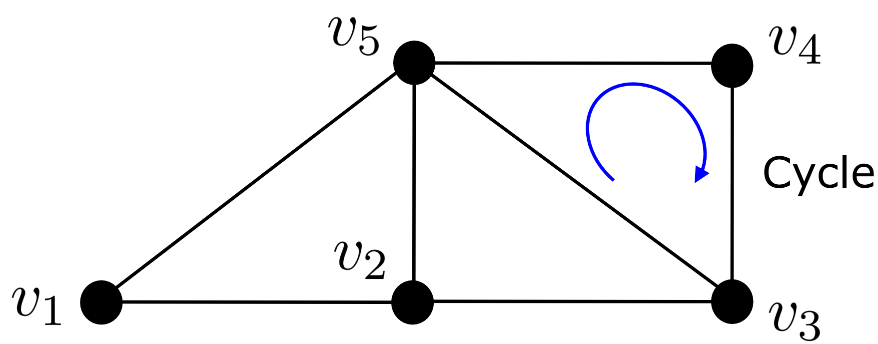
$$v_{i_0}, \, v_{i_1}, \, \ldots, v_{i_m}$$

  tels que pour $k \in \{0, \, 1, \, \ldots, m-1\}$, les sommets v_{i_k} et $v_{i_{k+1}}$ sont adjacents

- Si tous les sommets du chemin sont distincts sauf le sommet de départ et le sommet final, le chemin s'appelle **cycle**
- Un graphe $\mathcal{G}$ est **connexe** s'il existe un chemin entre tout couple de sommets en $V(\mathcal{G})$ (c'est-à-dire, s'il est d'un seul tenant). Sinon on dit que le graphe est **non connexe**

<u>Exemple</u>:

Le graphe à droite est connexe et il contient 6 cycles

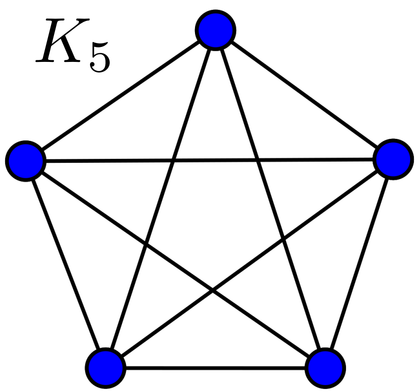


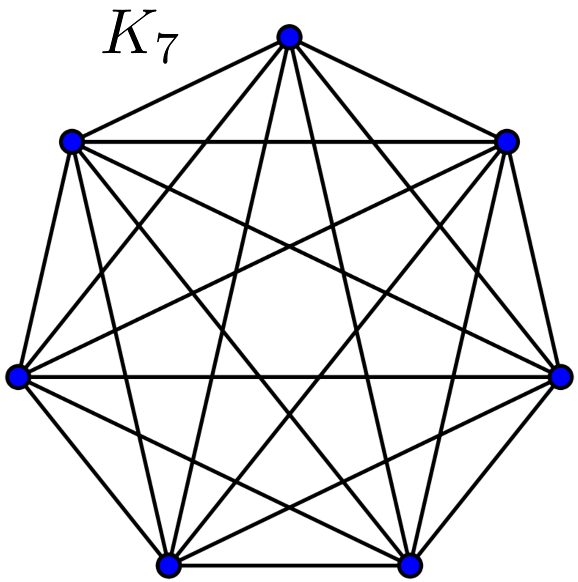
Familles de graphes

- K_n : ***Graphe complet*** (ou ***complètement connexe***)

Dans un graphe complet chaque sommet est relié
à tous les autres

Exemples:



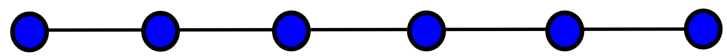


Familles de graphes

- P_n : ***Graphe chaîne***

$P_n = (\{v_1, \ldots, v_n\}, E_P)$, où $v_i v_j \in E_P$ si et seulement si $j = i + 1,\ i \in \{1, \ldots, n-1\}$

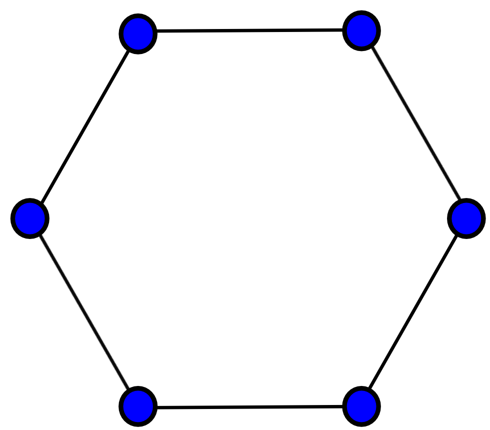
Exemple: P_6



- C_n : ***Graphe cycle***

$C_n = (\{v_1, \ldots, v_n\}, E_C)$, où $v_i v_j \in E_C$ si et seulement si $i - j \equiv \pm 1 \mod n$

Exemple: C_6



Note: $a \equiv b \mod n$ (c'est-à-dire, a et b sont congruents modulo n)

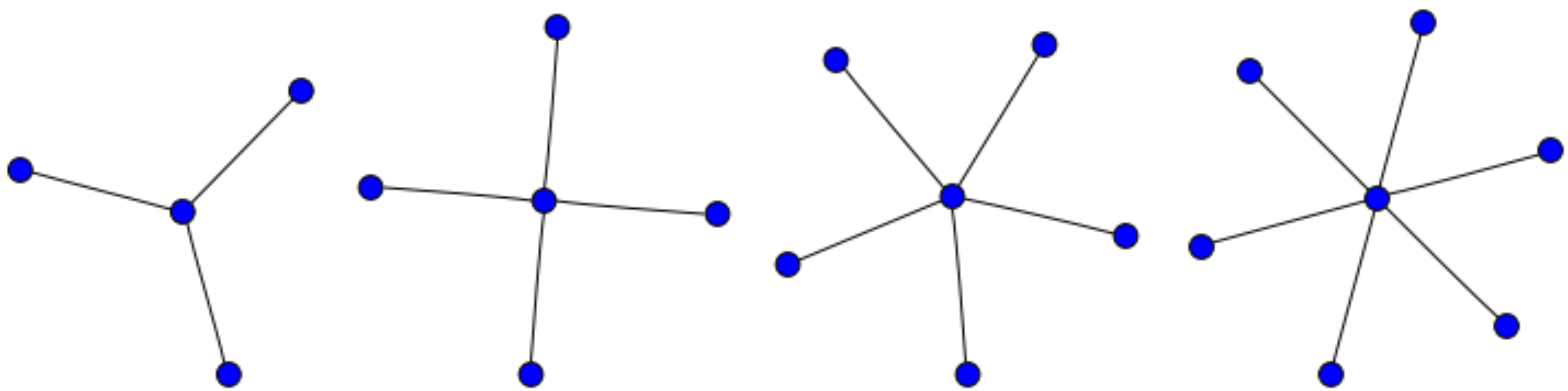
si $a - b$ est un multiple entier de n

Familles de graphes

- S_n : ***Graphe étoile***

$S_n = (\{v_1, \ldots, v_n\}, E_{\mathrm{star}})$, où $v_i v_j \in E_{\mathrm{star}}$ si et seulement si $i = 1$ ou $j = 1$

<u>Exemples</u>:



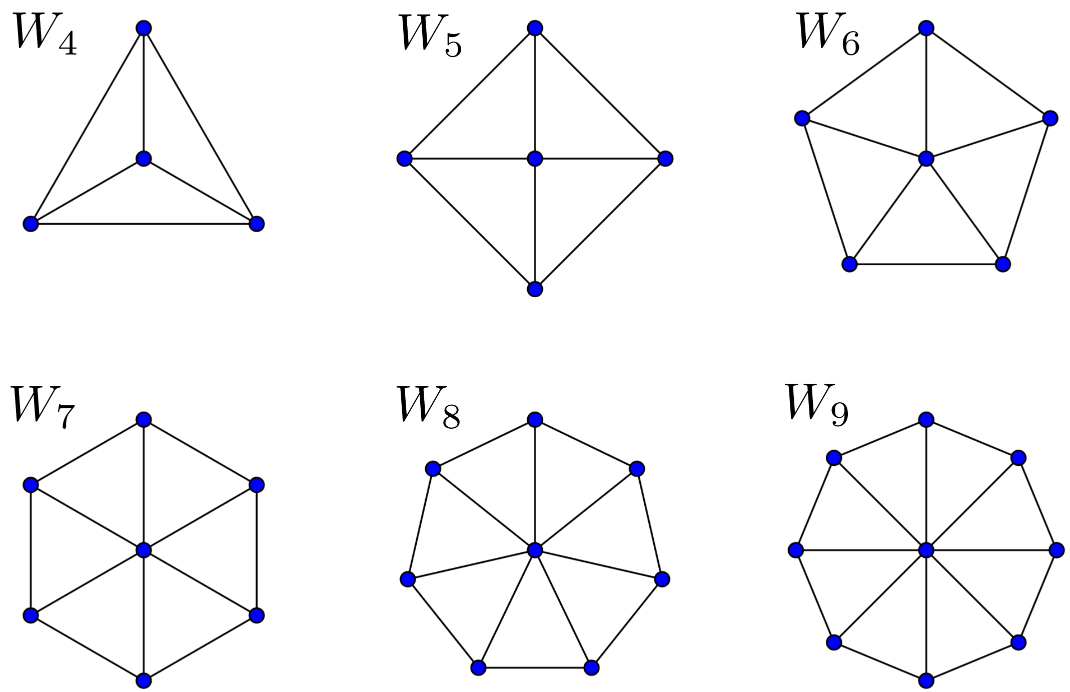
S_4 S_5 S_6 S_7

Familles de graphes

- W_n : ***Graphe roue***

Le graphe roue W_n est un graphe avec $n \geq 4$ sommets formé en ajoutant un sommet « centre » connecté à tous les sommets du graphe cycle C_{n-1}

<u>Exemples</u>:



Familles de graphes

- Un **graphe régulier** est un graphe où tous les sommets ont le *même nombre de voisins*, c'est-à-dire le même degré

- Un graphe régulier dont les sommets sont de degré k est appelé un graphe **k-régulier** (ou graphe régulier de degré k)

<u>Exemples</u>:

C_n : 2-régulier

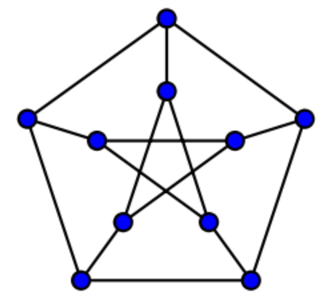
K_n : $(n-1)$-régulier

Graphe de Petersen : 3-régulier

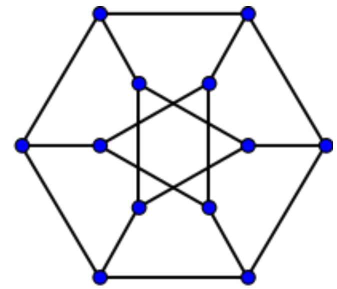
Graphe de Dürer : 3-régulier

Graphe de Clebsch : 5-régulier

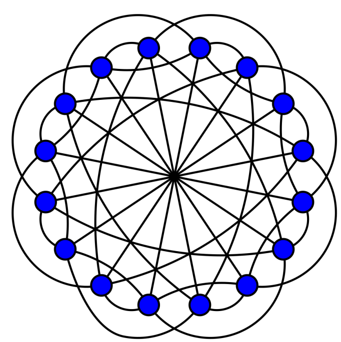
Graphe de Schläfli : 16-régulier



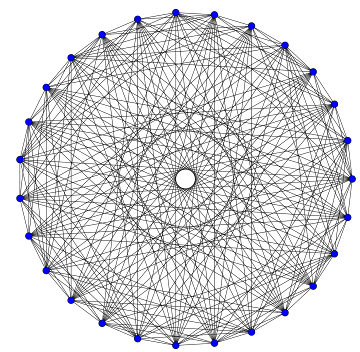
Petersen



Dürer



Clebsch



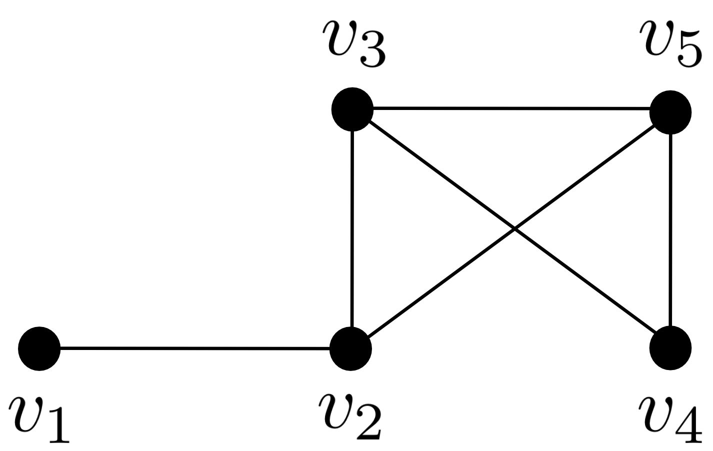
Schläfli

Graphes et matrices

- Outre une représentation graphique en termes de *sommets* et d'*arêtes*, chaque graphe admet aussi une représentation sous forme ***matricielle***

- Pour un graphe non orienté $\mathcal{G}$, le **degré** d'un sommet v_i, noté $d(v_i)$, est la cardinalité du voisinage $\mathcal{N}(i)$, c'est-à-dire le nombre de sommets qui sont adjacents au sommet v_i dans $\mathcal{G}$

<u>Exemple</u>:



$$d(v_1) = 1$$
$$d(v_2) = 3$$
$$d(v_3) = 3$$
$$d(v_4) = 2$$
$$d(v_5) = 3$$

Matrice des degrés et matrice d'adjacence

- La **matrice des degrés** d'un graphe $\mathcal{G}$ est la matrice diagonale $n \times n$ qui contient le degré de chaque sommet de $\mathcal{G}$ sur la diagonale:

$$\mathbf{\Delta}(\mathcal{G}) = \mathrm{diag}(d(v_1),\, d(v_2), \ldots, d(v_n)).$$

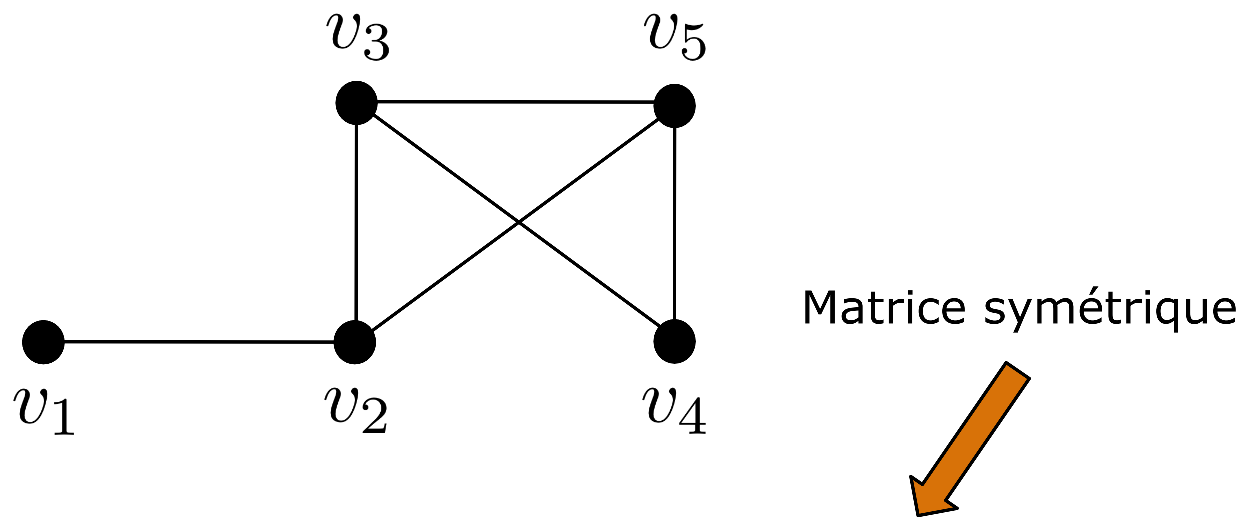
- La **matrice d'adjacence** $\mathbf{A}(\mathcal{G})$ est la matrice symétrique $n \times n$ qui codifie les relations d'adjacence dans le graphe $\mathcal{G}$:

$$[\mathbf{A}(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{si } v_i v_j \in E, \\ 0 & \text{sinon.} \end{cases}$$

Élément $(i,\, j)$ de la matrice $\mathbf{A}(\mathcal{G})$

Matrice des degrés et matrice d'adjacence

Exemple:



$$\mathbf{\Delta}(\mathcal{G}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}, \quad \mathbf{A}(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Matrice d’incidence

- Soit $\mathcal{G}^o$ le digraphe obtenu en associant une orientation arbitraire aux arêtes du graphe $\mathcal{G}$. La **matrice d‘incidence** $\mathbf{D}(\mathcal{G}^o)$ est une matrice $n \times m$ définie comme suit ($m =$ nombre de sommets et $n =$ nombre d‘arêtes):

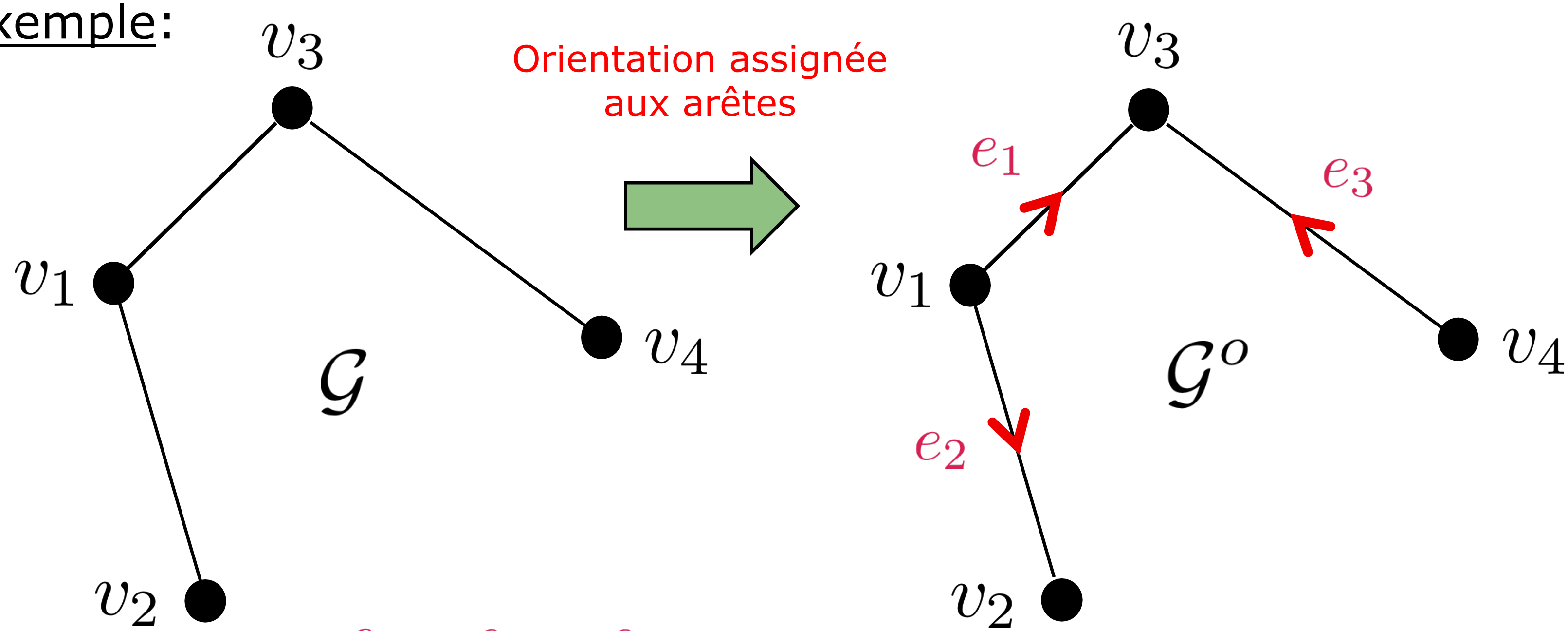
$$[\mathbf{D}(\mathcal{G}^o)]_{ij} = \begin{cases} -1 & \text{si } v_i \text{ est la queue de } e_j, \\ 1 & \text{si } v_i \text{ est la tête de } e_j, \\ 0 & \text{sinon.} \end{cases}$$

Remarque :

$\mathbf{D}(\mathcal{G}^o)$ capture non seulement la relation d’adjacence, mais aussi l’information sur l’*orientation* que le graphe possède désormais

Matrice d’incidence

<u>Exemple</u>:



$$\mathbf{D}(\mathcal{G}^o) = \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \begin{bmatrix} \boxed{\begin{array}{c} -1 \\ 0 \\ 1 \\ 0 \end{array}} & \begin{array}{c} -1 \\ 1 \\ 0 \\ 0 \end{array} & \begin{array}{c} 0 \\ 0 \\ 1 \\ -1 \end{array} \end{bmatrix} \end{array} \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}$$

Remarque :
La somme des éléments de chaque colonne de $\mathbf{D}(\mathcal{G}^o)$ est zéro

Matrice laplacienne

La **matrice laplacienne** associée au graphe non orienté $\mathcal{G}$ est définie comme suit:

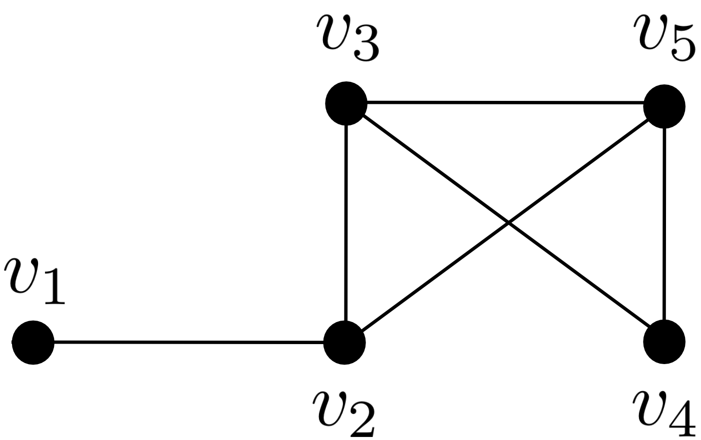
$$\mathbf{L}(\mathcal{G}) = \mathbf{\Delta}(\mathcal{G}) - \mathbf{A}(\mathcal{G}).$$

Pierre-S. Laplace
(1749-1827)

Il découle de cette définition que pour tout graphe, la somme des éléments dans chaque ligne de la matrice laplacienne est **zéro**:

$$\mathbf{L}(\mathcal{G})\mathbf{1} = \mathbf{0} \text{ où } \mathbf{1} = [1,\, 1,\, \ldots,\, 1]^T.$$

<u>Exemple</u>:



$$\mathbf{L}(\mathcal{G}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}$$

Matrice laplacienne : définition alternative

Étant donnée une *orientation arbitraire* de l'ensemble des arêtes $E(\mathcal{G})$, la **matrice laplacienne** du graphe $\mathcal{G}$ peut être définie de façon alternative comme suit:

$$\mathbf{L}(\mathcal{G}) = \mathbf{D}(\mathcal{G}^o)\,\mathbf{D}^T(\mathcal{G}^o)$$

Cette définition révèle que $\mathbf{L}(\mathcal{G})$ est une:

- Matrice symétrique (c'est-à-dire, $\mathbf{L}(\mathcal{G}) = \mathbf{L}^T(\mathcal{G})$)
- Matrice semi-définie positive (c'est-à-dire, $\mathbf{x}^T\,\mathbf{L}(\mathcal{G})\,\mathbf{x} \geq 0,\ \forall\,\mathbf{x} \in \mathbb{R}^n$)

Remarque : Les deux définitions que nous avons donné sont *équivalentes* et puisque la notion d'orientation n'est pas nécessaire dans la première, la matrice laplacienne est indépendante de l'orientation

TD3, ex. I-II

Théorie spectrale des graphes

- La **théorie algébrique des graphes** associe des *objets algébriques* (par ex. la matrice des degrés, d'adjacence ou laplacienne) à un graphe
- La **théorie spectrale des graphes** étudie les *valeurs propres* associées à la matrice d'adjacence et à la matrice laplacienne

---

Petit rappel …

Définition
Un vecteur non nul $\mathbf{u} \in \mathbb{R}^n$ est un *vecteur propre* de la matrice $\mathbf{A} \in \mathbb{R}^{n \times n}$ si et seulement si il existe un scalaire λ tel que:

$$\mathbf{A}\,\mathbf{u} = \lambda\,\mathbf{u}$$

où λ s'appelle *valeur propre* associée à $\mathbf{u}$.

On trouve les n valeurs propres $\{\lambda_1, \ldots, \lambda_n\}$ et vecteur propres $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ de $\mathbf{A}$ en résolvant le système d'équations:

$$(\mathbf{A} - \lambda\,\mathbf{I}_n)\mathbf{u} = \mathbf{0}$$

Spectre de la matrice laplacienne

Soit $\mathbf{L}(\mathcal{G})$ la matrice laplacienne du graphe $\mathcal{G}$. Cette matrice est *symétrique* et *semi-définie positive* : par conséquent ses n *valeurs propres réelles* peuvent être ordonnées de la manière suivante:

$$0 = \lambda_1(\mathbf{L}(\mathcal{G})) \leq \lambda_2(\mathbf{L}(\mathcal{G})) \leq \ldots \leq \lambda_n(\mathbf{L}(\mathcal{G})).$$

Théorème :

Le graphe $\mathcal{G}$ est **connexe** si et seulement si:

$$\lambda_2(\mathbf{L}(\mathcal{G})) > 0$$

Spectre de la matrice laplacienne

- **Il n'est pas aisé** de trouver le spectre $\{\lambda_1(\mathbf{L}),\, \lambda_2(\mathbf{L}), \ldots, \lambda_n(\mathbf{L})\}$ de la matrice laplacienne d'un *graphe quelconque*

- Toutefois, les valeurs propres (et les vecteurs propres) de la matrice laplacienne de certaines familles de graphes admettent une expression de forme fermée

---

<u>Graph complet</u>

Puisque $\mathbf{L}(K_n) = -\mathbf{1}\mathbf{1}^T + n\mathbf{I}_n$, le spectre de $\mathbf{L}(K_n)$ est celui de $-\mathbf{1}\mathbf{1}^T$ décalé de n. Le spectre de la matrice:

$$\mathbf{1}\mathbf{1}^T = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

est $\{0, \ldots, 0, n\}$, donc le spectre de la matrice laplacienne de K_n est:

$$\{0, n, \ldots, n\}$$

Spectre de la matrice laplacienne

Graphe chaîne

Les valeurs propres de la matrice laplacienne de P_n pour $n \geq 3$, sont:

$$2 - 2\cos\left(\frac{\pi k}{n}\right),\ k \in \{0,\ 1, \ldots,\ n-1\}$$

Graphe cycle

Les valeurs propres de la matrice laplacienne de C_n pour $n \geq 3$, sont:

$$2 - 2\cos\left(\frac{2\pi k}{n}\right),\ k \in \{0,\ 1, \ldots,\ n-1\}$$

Graphe étoile

Le spectre de la matrice laplacienne de S_n est:

$$\{0,\ 1,\ \ldots,\ 1,\ n\}$$

où la valeur propre 1 a multiplicité algébrique $n-2$

Spectre de la matrice laplacienne

Graphe roue

Le spectre de la matrice laplacienne du graphe roue W_n, pour $n \geq 4$ est:

$$\left\{3 - 2\cos\left(\frac{2k\pi}{n-1}\right),\ k \in \{1, 2, \ldots, n-2\}\right\} \cup \{0,\, n\}$$

Graphe de Petersen

Le spectre de la matrice laplacienne du graphe de Petersen est:

$$\{0,\ 2,\ 2,\ 2,\ 2,\ 2,\ 5,\ 5,\ 5,\ 5\}$$

Graphe de Clebsch

Le polynôme caractéristique de la matrice d'adjacence du graphe de Clebsch est $\det(\lambda \mathbf{I}_{16} - \mathbf{A}) = (\lambda - 5)(\lambda - 1)^{10}\,(\lambda + 3)^5$ (il est le seul graphe avec ce polynôme caractéristique). Étant un graphe 5-régulier, le spectre de la matrice laplacienne du graphe de Clebsch est:

$$\{0,\ \underbrace{2, \ldots, 2}_{\text{5 fois}},\ \underbrace{4, \ldots, 4}_{\text{10 fois}}\}$$

Spectre de la matrice laplacienne

Miroslav Fiedler
(1926–2015)

$$0 = \lambda_1(\mathbf{L}(\mathcal{G})) \leq \boxed{\lambda_2(\mathbf{L}(\mathcal{G}))} \leq \ldots \leq \lambda_n(\mathbf{L}(\mathcal{G})).$$

$\lambda_2(\mathbf{L}(\mathcal{G}))$ est la deuxième valeur propre la plus faible de la matrice laplacienne

Elle est appelée valeur de Fiedler et le vecteur propre associé $\mathbf{u}_2$, vecteur de Fiedler

Remarque:

La valeur de Fiedler est cruciale non seulement comme **mesure de robustesse** ou **du niveau de connexité** d’un réseau, mais aussi pour les **propriétés de convergence** d’un *algorithme distribué* important:

Protocole de consensus

Toolbox de Matlab pour la théorie des graphes

À partir de Matlab R2015b

Quelques commandes de la toolbox

`graph`

G = graph(A) uses the square symmetric matrix A as an adjacency matrix and constructs a weighted graph with edges corresponding to the nonzero entries of A.

G = graph(S,T) constructs a graph with edges specified by the node pairs (S,T). S and T must have the same number of elements or be scalars.

G = graph(S,T,...,'OmitSelfLoops') does not add self-loops to the graph

graph properties:

Edges - Table containing edge information
Nodes - Table containing node information

Quelques commandes de la toolbox

graph methods:

`numnodes` - Number of nodes in a graph
`numedges` - Number of edges in a graph

`addnode` - Add nodes to a graph
`rmnode` - Remove nodes from a graph
`addedge` - Add edges to a graph
`rmedge` - Remove edges from a graph

`degree` - Degree of nodes in a graph
`neighbors` - Neighbors of a node in a graph
`subgraph` - Extract an induced subgraph

`adjacency` - Adjacency matrix of a graph
`incidence` - Incidence matrix of a graph
`laplacian` - Graph Laplacian

Quelques commandes de la toolbox

graph methods:

bfsearch	- Breadth-first search (BFS)
dfsearch	- Depth-first search (DFS)
shortestpath	- Compute shortest path between two nodes
shortestpathtree	- Compute single source shortest paths
distances	- Compute all pairs distances
nearest	- Compute nearest neighbors (at a distance D) of a node
conncomp	- Compute connected components of a graph
minspantree	- Compute minimum spanning tree of a graph
isisomorphic	- Determine whether two graphs are isomorphic
plot	- Plot an undirected graph