Multirobot Systems

The consensus problem and applications
Framework

Master Program in Robotics, Graphics and Computer Vision
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

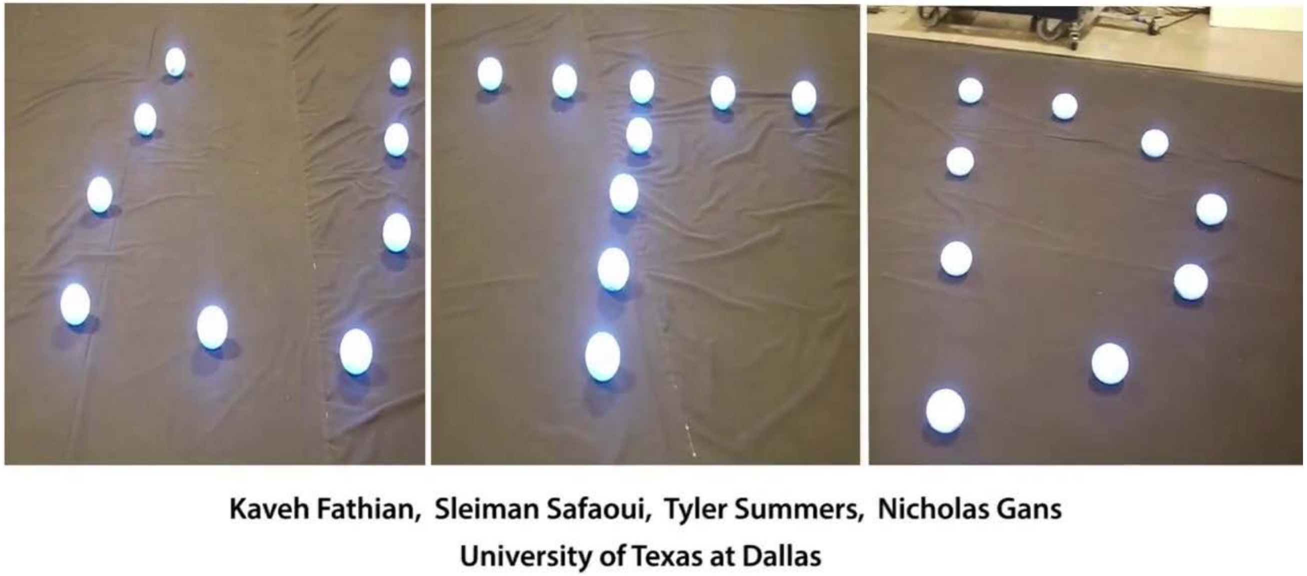
Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

In this lecture

- Collective behaviors
- Taxonomy
- Graphs
- Centralized and distributed architectures
- Algebraic graph theory
- The Laplacian matrix

An example

- With 10 robots, with 100 robots, with 1,000 robots?



https://youtu.be/AxT-fFcGQoA

https://sites.google.com/view/kavehfathian
https://github.com/TSummersLab/SMART_matlab

K. Fathian, S. Safaoui, T. H. Summers and N. R. Gans, "Robust Distributed Planar Formation Control for Higher Order Holonomic and Nonholonomic Agents," in IEEE Transactions on Robotics, doi: 10.1109/TRO.2020.3014022.

Examples of collective motions nature?

Flock of Starlings (National Geographic)

https://www.youtube.com/watch?v=V4f_1_r80RY&t=10s

School of fish

https://youtu.be/15B8qN9dre4?t=48

Herd of sheep

https://www.youtube.com/watch?v=tDQw21ntR64

Image: Diego Delso, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons.
https://commons.wikimedia.org/wiki/File:Banco_de_peces_trompeta_(Macroramphosus_scolopax),_islas_Azores,_Portugal,_2020-07-27,_DD_40.jpg

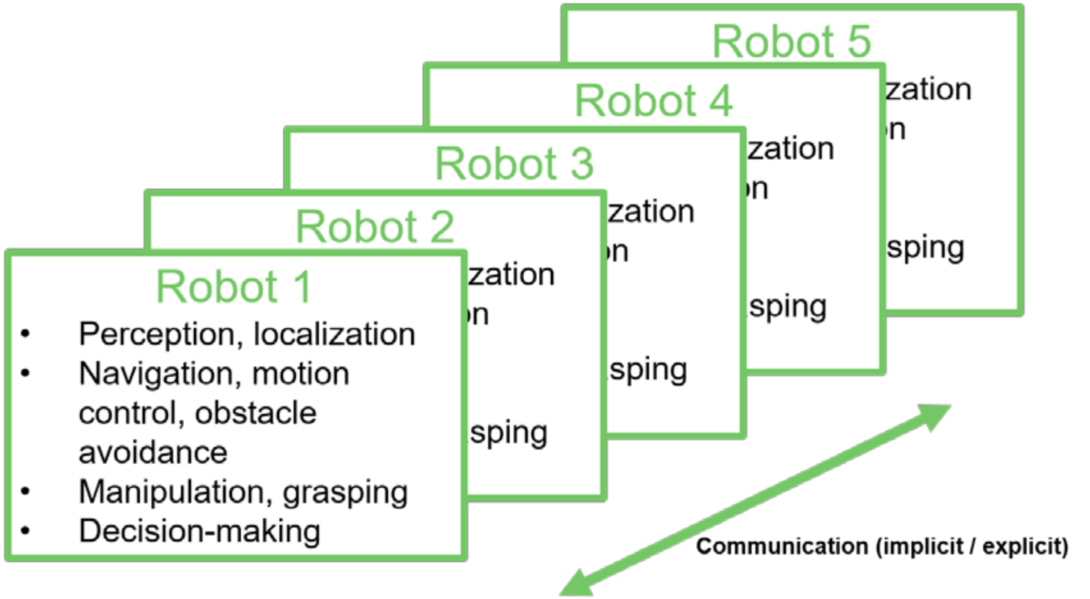
Properties of collective motions

- Collective motion in nature:
  - Properties: no collisions; often no apparent leader; tolerance of loss or gain of group member; merging and splitting; reactivity to obstacles; different species have different flocking characteristics
  - Benefits: energy saving (e.g., geese extend flight range by 70%); signs of better navigation accuracy
- Engineered flocking - decentralized:
  - Reynolds' virtual agents (Boids)
  - Graph-based distributed control for spatial consensus
- Engineered flocking – centralized:
  - E.g.: Controls for each robot computed off-board, in the cloud

Framework multi-robot systems

- Control, perception, decision making, navigation, coordination

- Terms used: robot swarms / robot teams / robot networks
- Distributed nature of many problems and applications
- Increased overall performance: extends the capabilities of what can be done with a single robot
- Redundancy and increased robustness



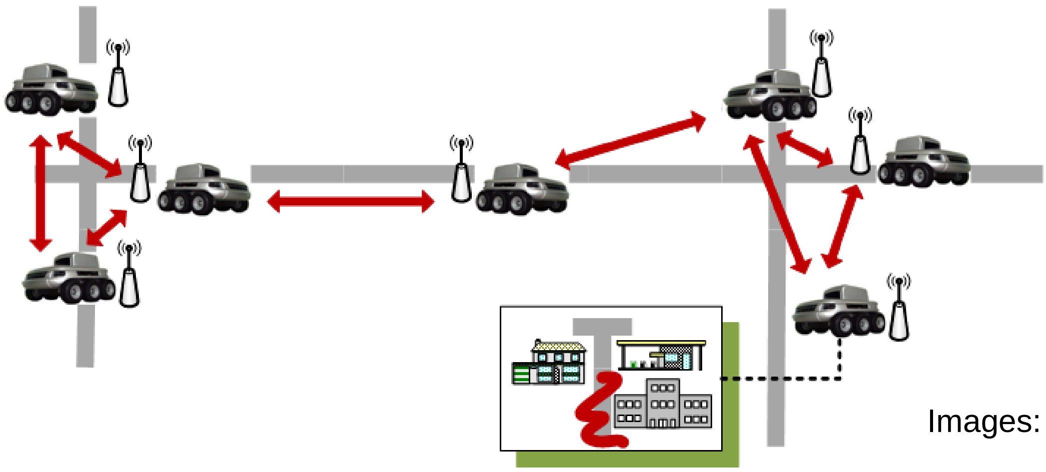
- Challenges: coordinating the team, make decisions on partial and different data, communication..

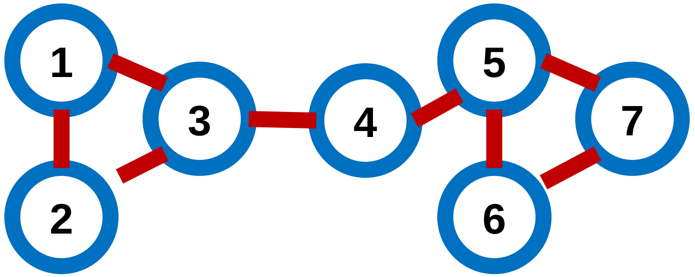
Taxonomy

- Architecture (centralized / decentralized):
  - Centralized: one control/estimation unit communicates with all robots to issue commands
    - Requires synchronized, reliable communication channels
    - Single-point failures
  - Decentralized: distributed between the robots
    - Scalable, robust to failure; often asynchronous
    - Challenges to ensure performance (w.r.t centralized), to properly synchronize / coordinate
- Communication (explicit vs. implicit)
  - Implicit: observable states (e.g., in the environment); information exchanged through common observations
  - Explicit: unobservable states; need to be communicated explicitly
- Heterogeneity (homogeneous vs. heterogeneous)
  - Robot teams can leverage inter-robot complementarities
  - Different robots with different capabilities and roles

Why do we need graphs?

- Graphs are extremely powerful tools for encoding the **information/action flow** among the robots
- We (sometimes implicitly) assume that every robot has a limited ability to:
  - **perceive** the environment with onboard sensors (e.g., other robots)
  - **communicate** information to other robots (via a communication medium)
  - **elaborate** information (gathered from onboard sensors or comm. medium)
  - in general, **plan, act, and influence** the environment (e.g., other robots)



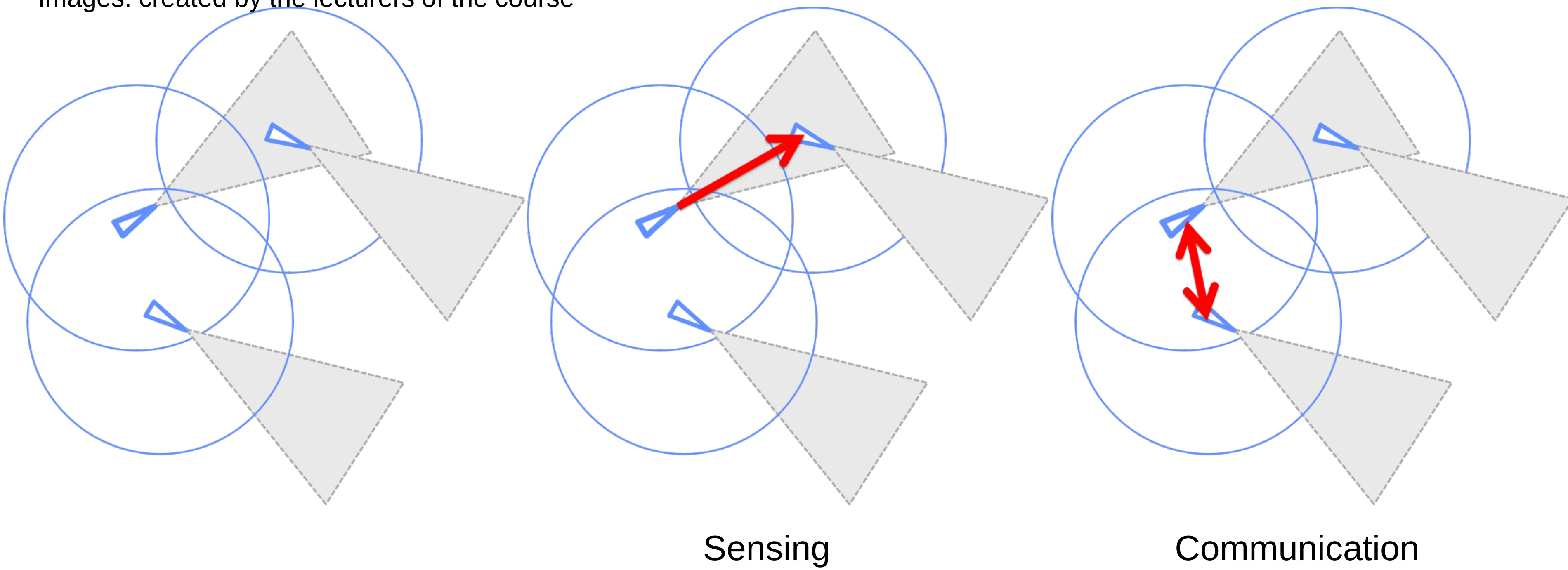


Images: created by the lecturers of the course

Explicit / implicit communication (Sensing vs. Sending data)

Examples of sensing (limited field of view, gray areas) and comm. (blue circular regions)

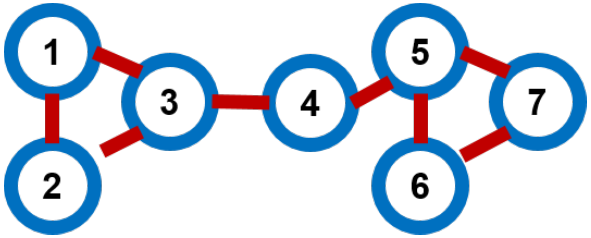
Images: created by the lecturers of the course



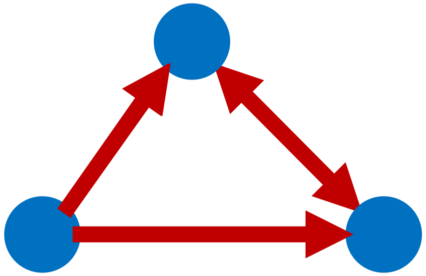
- **Sensing graphs**: for each sensor, encode which robots can be locally sensed
- **Communication graphs**: for each communication medium, encode with which robots a comm. link can be established (uni- or bi-directional)
- **Action graphs**: for each control action, encode what robots will be (locally) affected

Graphs

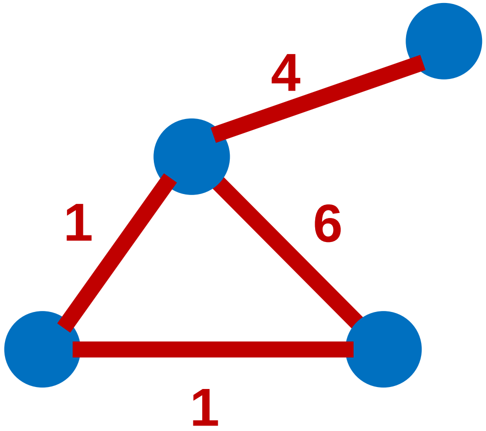
Images: created by the lecturers of the course



Undirected graph



Directed graph



Weighted graph

- Fixed vs. time varying
- Synchronous, asynchronous, event-triggered, gossip (randomized)

- Mesbahi, Mehran, and Magnus Egerstedt. **Graph Theoretic Methods in Multiagent Networks**. PRINCETON; OXFORD: Princeton University Press, 2010. www.jstor.org/stable/j.ctt1287k9b Accessed July 10, 2020. doi:10.2307/j.ctt1287k9b.

Undirected Graphs

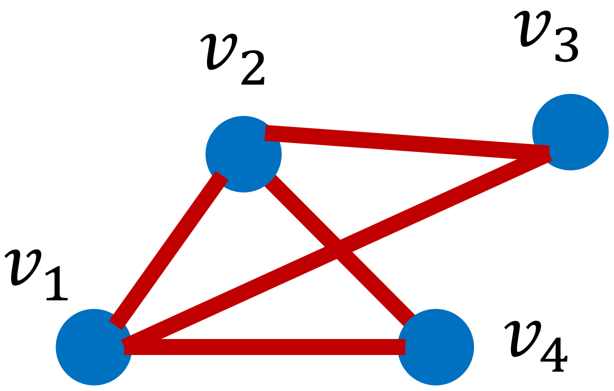
Images: created by the lecturers of the course

- Graph $G = (V, E)$
- Nodes, vertex set (e.g, robots) $V = \{v_1, v_2, \cdots, v_N\}$
- Edges (e.g. comm. / sensing between robots)

$$E \subseteq \{(v_i, v_j)\}, i = 1 \cdots N, j = 1 \cdots N, i \neq j$$

- **Undirected**:

$$(v_i, v_j) \in E \Rightarrow (v_j, v_i) \in E$$



$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4)\}$$

Directed Graphs

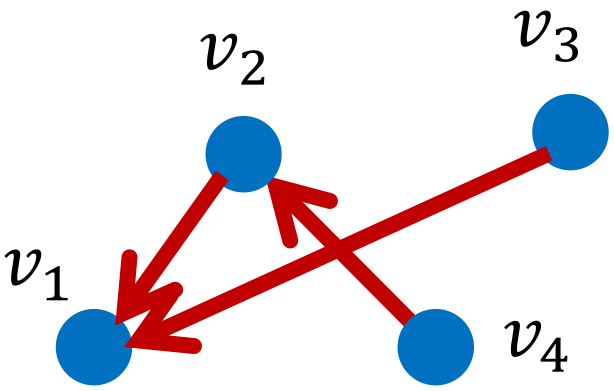
Images: created by the lecturers of the course

- Graph $G = (V, E)$
- Nodes, vertex set (e.g, robots) $V = \{v_1, v_2, \cdots, v_N\}$
- Edges (e.g. comm. / sensing between robots)

$$E \subseteq \{(v_i, v_j)\}, i = 1 \cdots N, j = 1 \cdots N, i \neq j$$

- **Directed**:

$$(v_i, v_j) \in E \nRightarrow (v_j, v_i) \in E$$



$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{(v_2, v_1), (v_3, v_1), (v_4, v_2)\}$$

Definitions

- **Neighbors** (set of neighbors) $N_i = \{v_j \in V \mid (v_j, v_i), \in E\}$
- **Degree** of a node (undirected graphs) $d_i = |N_i|$
- **In-degree** of a node (directed graphs) $d_i^{in} = |N_i|$

  recall that $(v_i, v_j) \in E \nRightarrow (v_j, v_i) \in E$

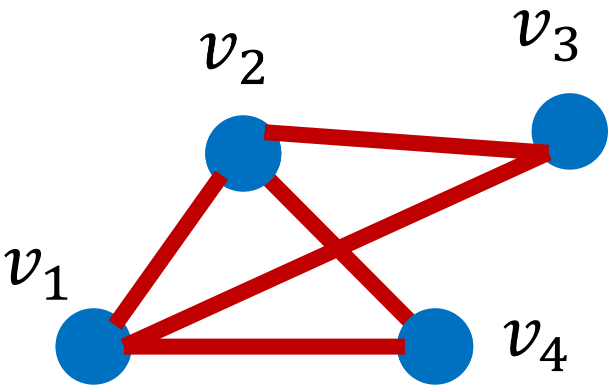
- **Path**: sequence of distinct vertexes such that the vertexes and are adjacent (neighbors)

$$v_{i_0}, v_{i_1}, \ldots, v_{i_m}, s.t.\, v_{i_k}\ and\ v_{i_{k+1}} are\ neighbors$$

- **Cycle** (special case):

$$v_{i_0} = v_{i_m}$$

Definitions



$$V = \{v_1, v_2, v_3, v_4\}$$

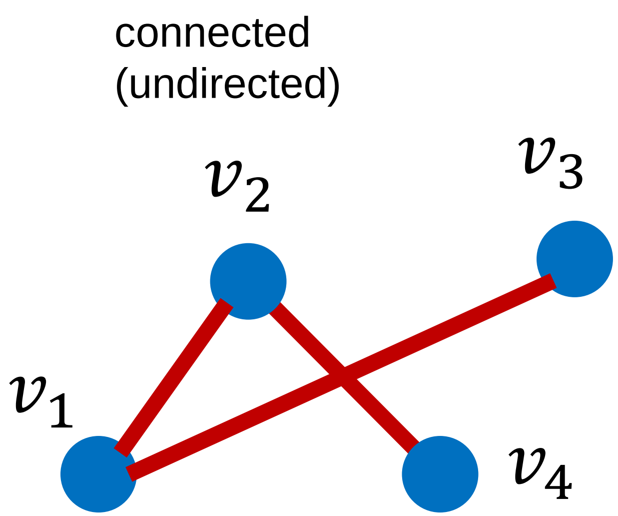
$$E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4)\}$$

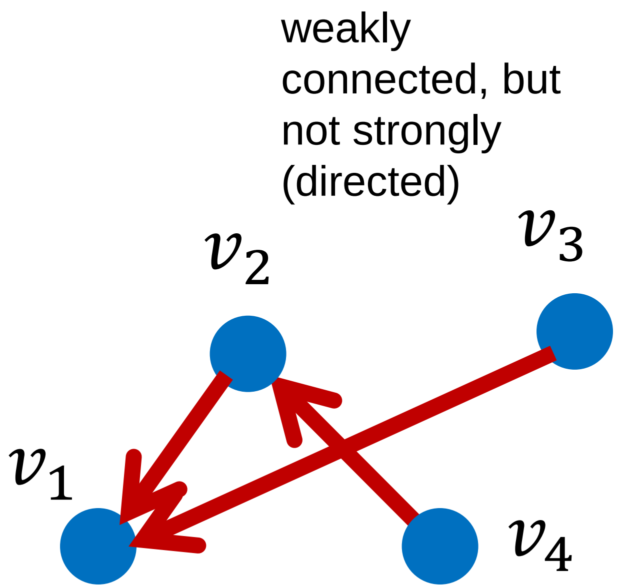
- The neighbors of $v_3: N_3 = \{v_1, v_2\}$
- The degree of v_3 $is\ d_3 = 2$
- There is (at least) a path between v_1 and v_3. E.g.: v_1, v_2, v_3.
- There is (at least) a cycle involving v_4, e.g., v_4, v_2, v_1.

Images: created by the lecturers of the course

Definitions

- An **undirected** graph is said **connected** if there exists a path joining any two nodes
- A **directed** graph is said **strongly connected** if there exists a (directed) path joining any two nodes
- **A directed** graph is said **weakly connected** if there exists an undirected path joining any two nodes



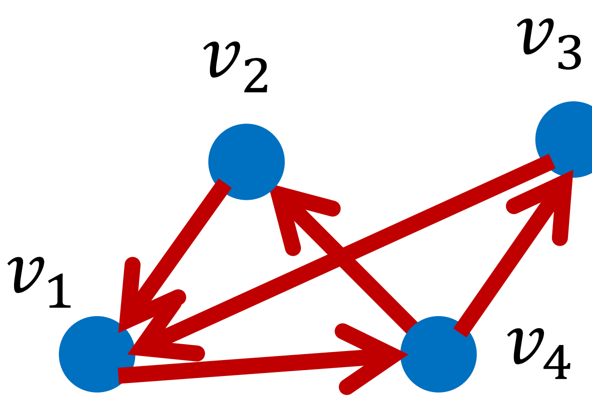


Images: created by the lecturers of the course

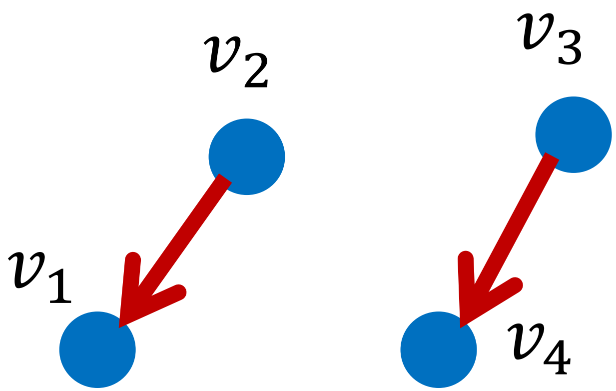
Definitions

- An **undirected** graph is said **connected** if there exists a path joining any two nodes
- A **directed** graph is said **strongly connected** if there exists a (directed) path joining any two nodes
- **A directed** graph is said **weakly connected** if there exists an undirected path joining any two nodes

Strongly connected (directed)



Disconnected (directed)

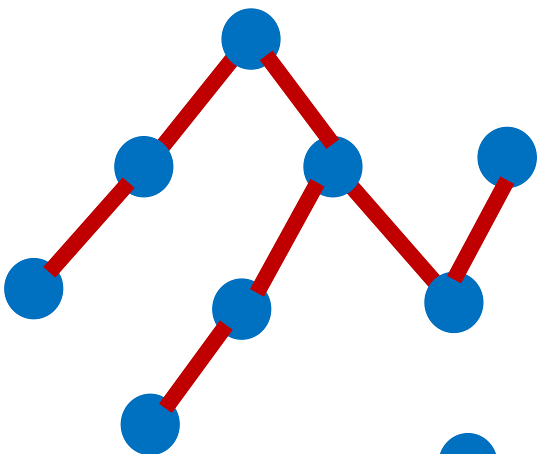


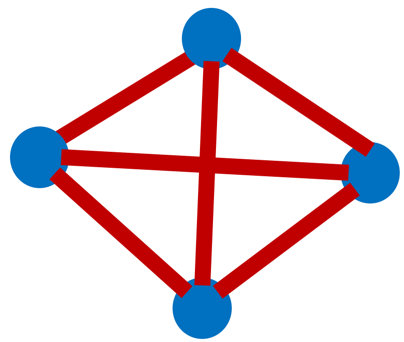
- Why are connectivity notions important? No data exchange / only in a subset == achievable behaviors

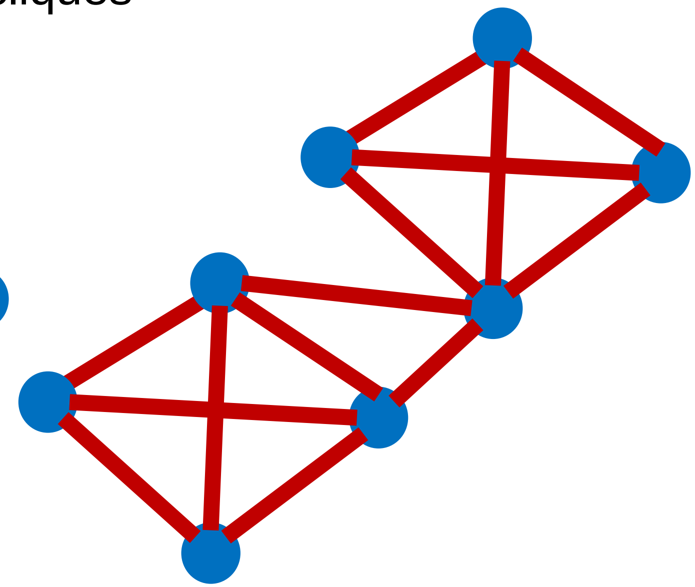
Images: created by the lecturers of the course

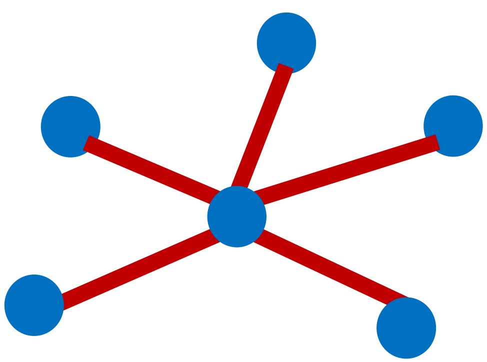
Special graphs / subgraphs

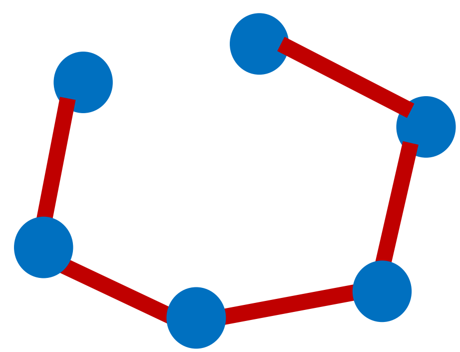
- Trees (and spanning Trees): N nodes, N-1 edges, connected
- Complete graphs (all–to–all, fully connected), cliques
- Star topology
- Line graph (path)











More on graphs, connectivity…

- Switching (time-varying)
- Random
- Jointly connected
- Interval of joint connectivity
- Synchronous / asynchronous, gossip
- Structure:
  - Globally reachable node
  - Rooted spanning trees
  - Regular graphs (all nodes with the same number of neighbors)
  - Lattice graph (mesh graph, or grid graph): regular tiling
- Weighted graphs
- Minimum-distance spanning trees (MST)

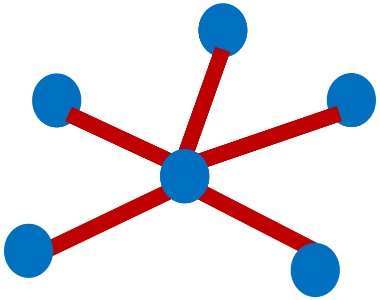
  (…)

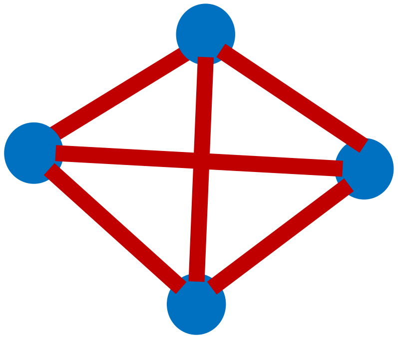
- In this course: definitions will be given when needed for a particular application!

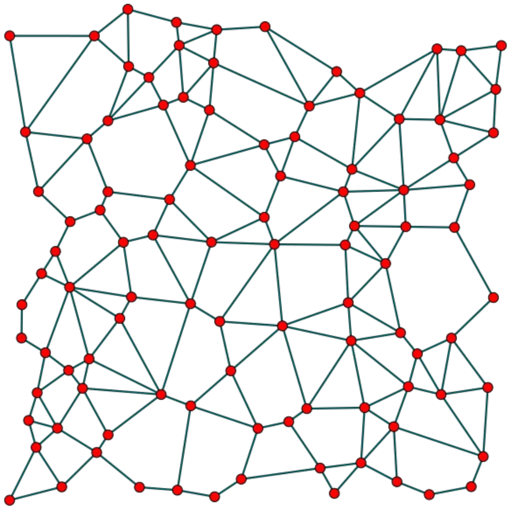
Centralized vs Distributed

- Multi-robot systems: every unit (robot) has:
  - limited sensing/communication (information **gathering**)
  - limited computing power (information **processing**)
  - limited available memory (information **storage**)
- **Centralized**: one unit communicates with all robots to issue commands
  - Single-point failure
  - Robots usually need the gathered information to run its local controller.
  - If the whole state of all the robots is needed: increases with the number of robots
  - It may become unfeasible!

What is more appropriate here?





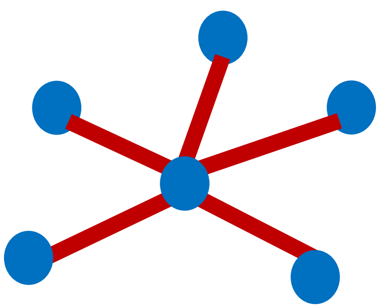


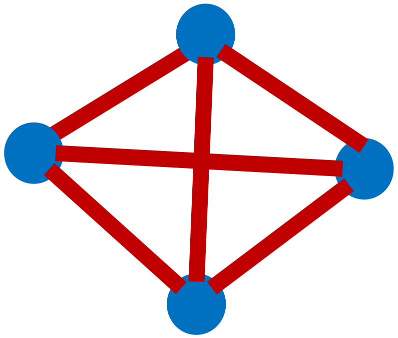
Images left and center: built by the lectures of the course. Image right: David Eppstein, Public domain, via Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Gabriel_graph.svg

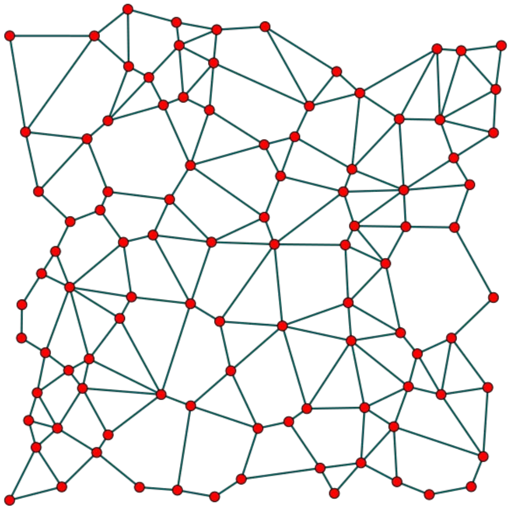
Centralized vs Distributed

- **Decentralized**: distributed between the robots
  - A robot's control action (or estimate) is based on interaction with (or relative observations of) neighbors
  - Adding a robot does not increase the amount of data exchanged / stored / processed (it may increase the convergence time)
  - Scalable, robust to failure; often asynchronous
  - Challenges to ensure *optimal / sub-optimal* performance (w.r.t centralized), to properly synchronize / coordinate

What is more appropriate here?







Images top, left and center: built by the lectures of the course. Image right: David Eppstein, Public domain, via Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Gabriel_graph.svg

Thus.. centralized or distributed?

- It depends on the application! Usually **flexible**! Hybrid architectures more or less degree of components / elements **centralized** / **decentralized**.

- Automated warehouses (Kiva Amazon). Goal: to minimize the time between customer request and product is delivered.
- Automated transport of people in a city. Goal: to minimize the time between a passenger requests a vehicle until it is picked up.

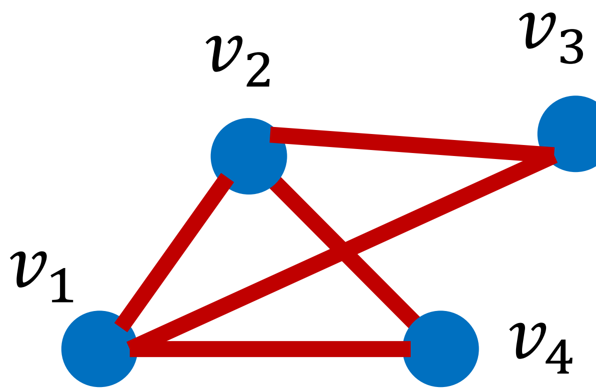
- Surveillance of an area with a swarm of drones. Goal: to minimize the time to find a target / maximize the covered area.
- Platooning of vehicles (autonomous / semi-autonomous vehicles). Goal: to minimize the probability of collision.
- Applications inherently distributed in time and/or in space?

Algebraic graph theory (Graphs & Matrices)

- Several matrixes can be associated to graphs and….
- ….several graph properties deduced from the associated matrices

- Algebraic tools **fundamental** for linking Graph Theory to the study of multi-robot systems

  - ❑ **Adjacency** Matrix
  - ❑ **Degree** Matrix
  - ❑ **Incidence** Matrix
  - ❑ **Laplacian** Matrix



$$G = (V, E) \qquad V = \{v_1, v_2, v_3, v_4\}$$

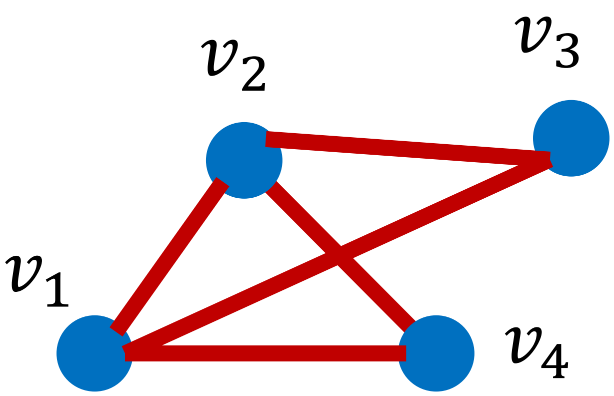
$$E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4)\}$$

Adjacency matrix

- Adjacency matrix $A \in \mathbb{R}^{NxN}$
- Underlying graph structure

$$A_{ii} = 0, A_{ij} = 1 \; if \left(v_j, v_i\right) \in E, A_{ij} = 0 \; if \left(v_j, v_i\right) \notin E$$

- Undirected graphs: square and symmetric
- Directed graphs: square
- It can be generalized to a versions with **weights**



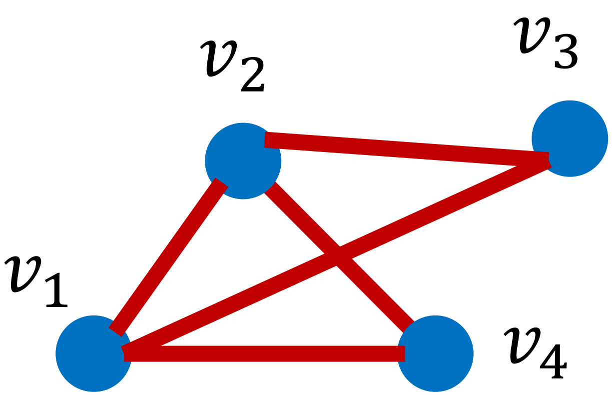
Graph

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Degree matrix

- Degree matrix $\Delta \in \mathbb{R}^{NxN}$
- Degree (number of neighbors) of every node (robot):

$$\Delta = diag(d_i) = diag\left(\sum_{j=1}^{N} A_{ij}\right)$$



$$\Delta = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

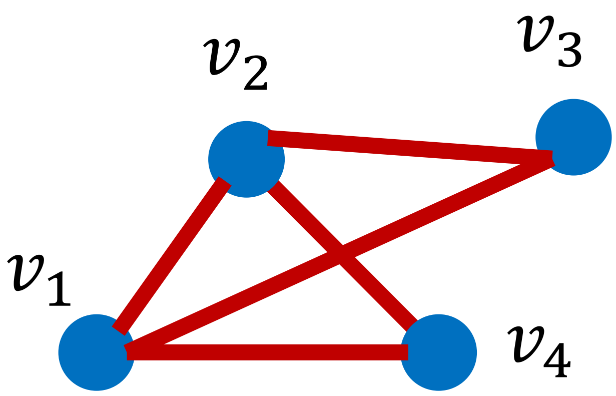
Laplacian matrix

- **Laplacian** matrix $L \in \mathbb{R}^{NxN}$:

$$L = \Delta - A$$

Degree and Adjacency matrices

- Important properties! Convergence, conv. speed MRS algorithms



$$L = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{bmatrix}$$

Laplacian matrix. Properties

- **Undirected graphs: Symmetric and positive semidefinite**
- $L\mathbf{1} = \mathbf{0}$

  (i.e., eigenvector with all entries equal to one, eigenvalue 0)
- All eigenvalues real and $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$
- Properties:

  G **connected** if and only if $0 < \lambda_2$

  λ_2 Algebraic connectivity, Fiedler eigenvalue: more / less connected + convergence speed

  Disconnected graphs: connected subgraphs <-> num. eigvals = 0

Many other properties in the literature! In this course: they will be introduced when needed for a particular application.

Main ideas in this lecture ?

- Multi-robot systems, inher. robust
- Cooperation: collective behaviors
- Architectures (distributed/centralized)
- Sensing / communication interactions
- Graphs
- Matrices: Adjacency, Degree, **Laplacian**

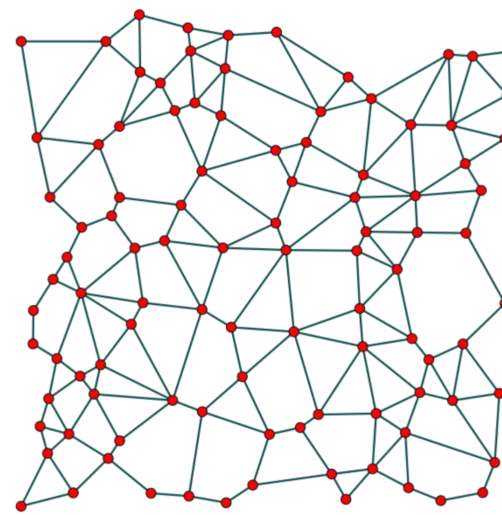


Image left: built by the lectures of the course. Image top: : Diego Delso, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, via Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Banco_de_peces_trompeta_(Macroramphosus_scolopax),_islas_Azores,_Portugal,_2020-07-27,_DD_40.jpg Image right: David Eppstein, Public domain, via Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Gabriel_graph.svg

Next lectures…

- Ok, but how can we program a robot team to make them achieve collective behaviors such as?
  https://youtu.be/AxT-fFcGQoA

- … step by step …
- In the next lecture: **The consensus problem**
- The origin of several distributed MRS algorithms to achieve collective behaviors

Bibliography

- Mesbahi, Mehran, and Magnus Egerstedt. Graph Theoretic Methods in Multiagent Networks. PRINCETON; OXFORD: Princeton University Press, 2010. www.jstor.org/stable/j.ctt1287k9b Accessed July 10, 2020. doi:10.2307/j.ctt1287k9b.
- Bullo, Francesco, Jorge Cortes, and Sonia Martinez. Distributed control of robotic networks: a mathematical approach to motion coordination algorithms. Vol. 27. Princeton University Press, 2009. Freely available: http://coordinationbook.info/index.html
- Related talks / videos:

Amanda Prorok (University of Cambridge). Summer School on Multi-Robot Systems 2019. Talk ”Control and coordination”
Slides: http://mrs.felk.cvut.cz/summer-school-2019/assets/pdf/amanda.pdf
Videos: https://www.youtube.com/watch?v=FeoN-lmdve8&list=PLPjuFI-_2rxxCr3AD7HBcFbCG6nvL56Rg&index=1

To get more knowledge on robot control (from the automatic control point of view)…
Magnus Egerstedt (Georgia Institute of Technology)
Control of Mobile Robots (MOOC)
Videos: https://www.youtube.com/watch?v=aSwCMK96NOw&t=3s (lesson 1.1) and additional videos at youtube up to 5.7.