JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Multiagent Systems

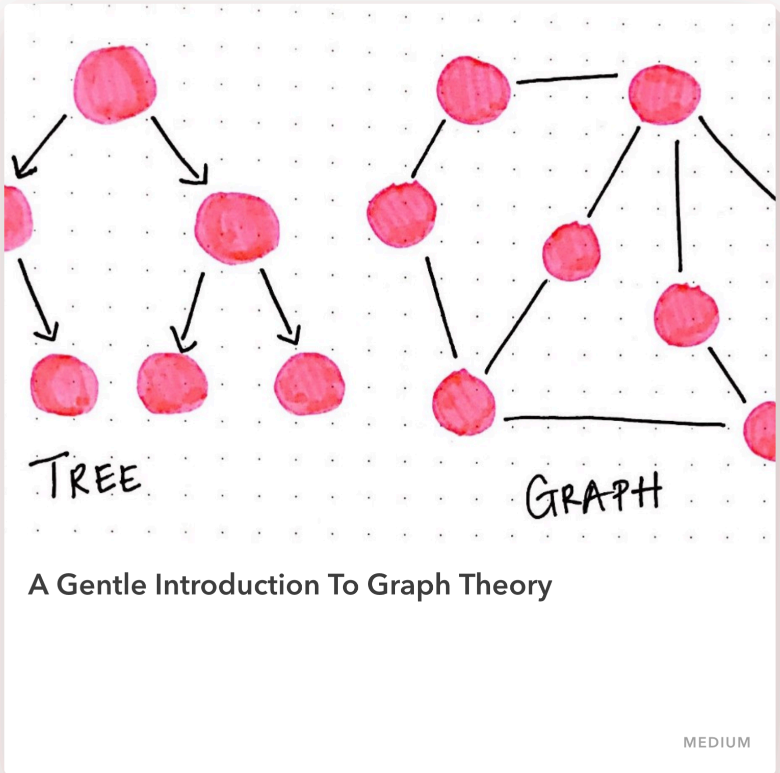
601.464
Artificial Intelligence
TR 10.30AM—11.45AM

Material

- Graphs
  - Vertex set
  - Edge sets
  - Neighborhood
  - Degree
  - Complete graphs (K_1, K_2, K_3, …)
- Network Analysis
  - Connected Components
  - Isolated Components
  - Clustering Coefficient (for each node)
- Boids Rule (one of the assigned papers)
- Consensus Equation/Protocol

Additional Resources

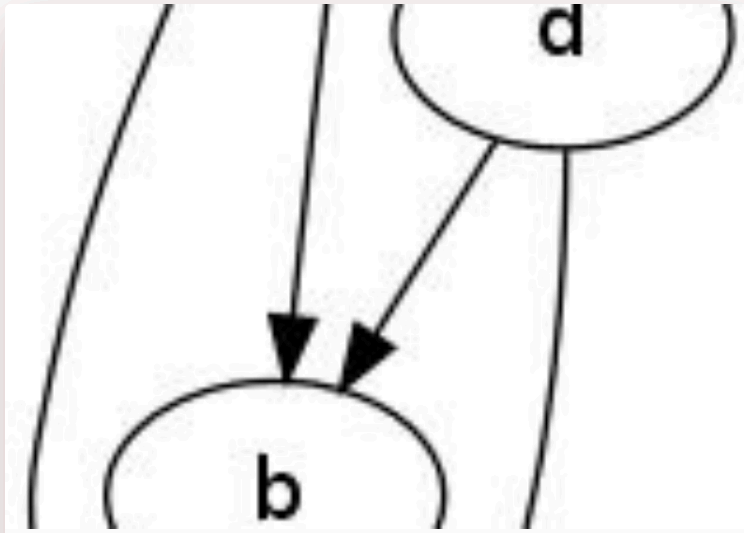
https://medium.com/basecs/a-gentle-introduction-to-graph-theory-77969829ead8



Not required

Additional Resources

https://www.cs.colorado.edu/~srirams/courses/csci2824-spr14/graphs-29.html



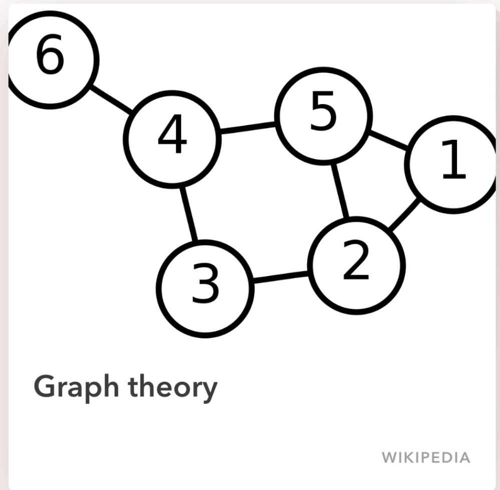
CSCI 2824 Lecture 29: Graph Theory (Basics)

COLORADO

Not required

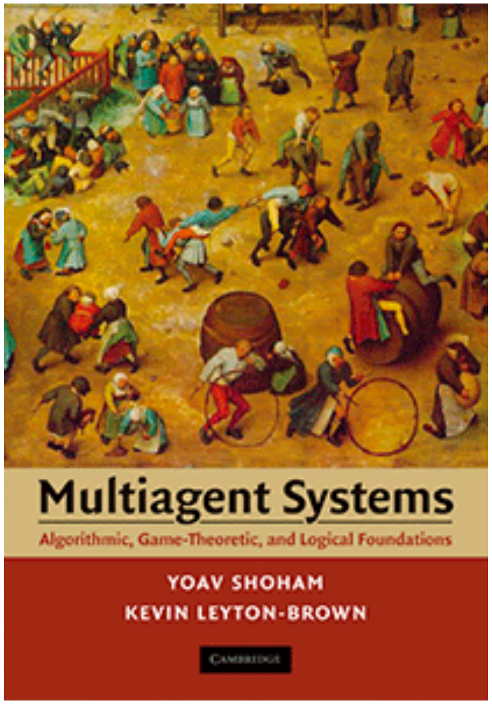
Additional Resources

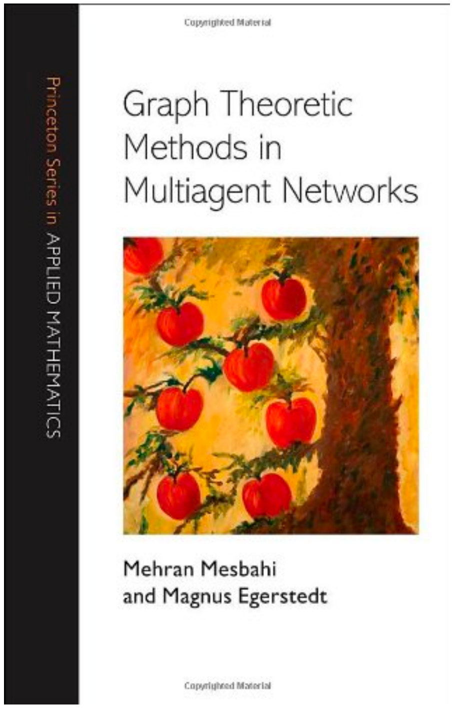
https://en.wikipedia.org/wiki/Graph_theory



if all else fails

Fancier Resources



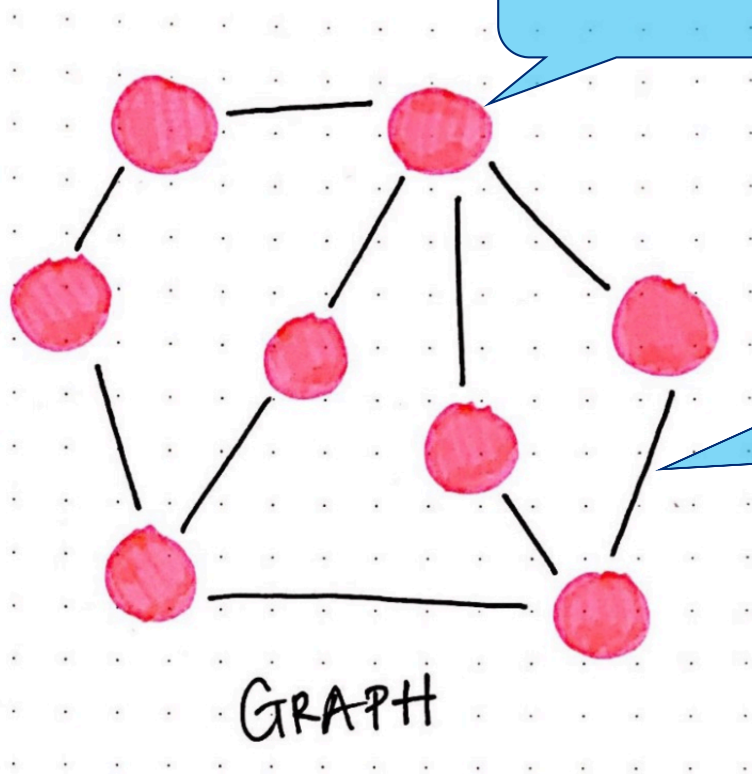


Not required

Graphs

A graph is a tuple that consists of a vertex set and an edge set

$$G = (V, E)$$



A node (or vertex)

An edge (or link)

Graphs

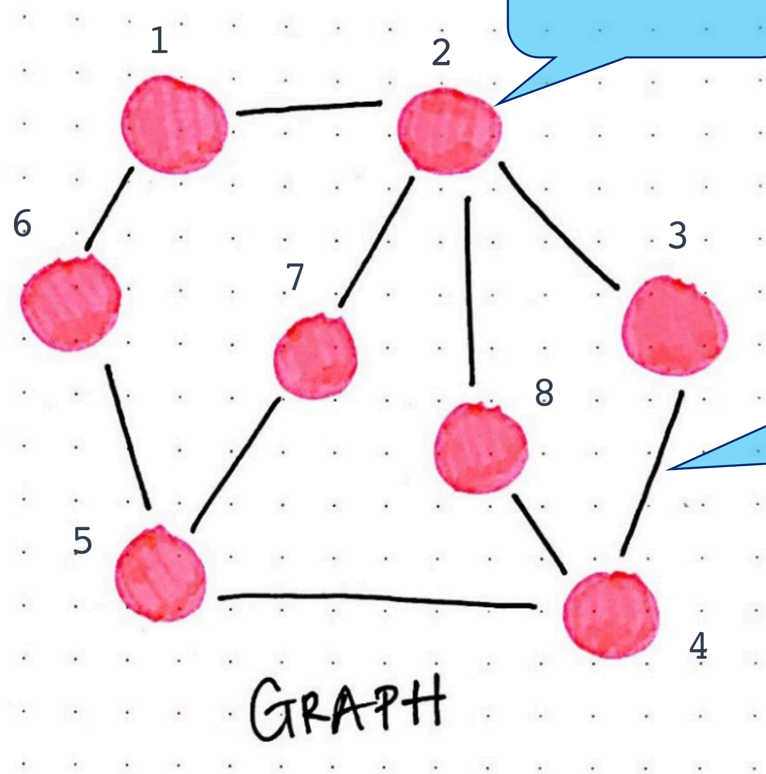
Graph, G

$G = (V, E)$

$V = \{1, 2, 3, 4, 5, 6, 7, 8\}$

A node (or vertex)

Added some labels to the nodes

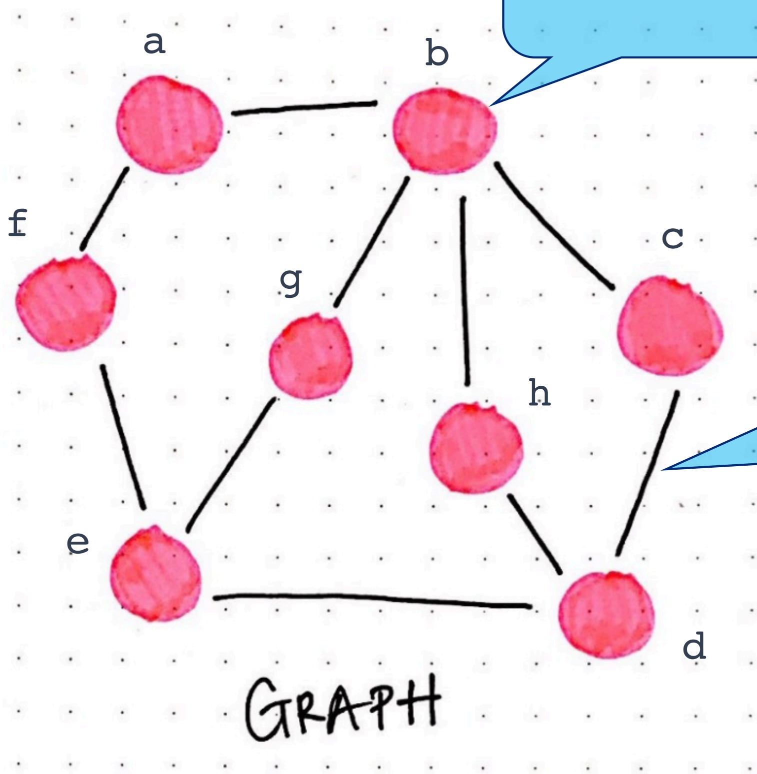


An edge (or link)

Graphs

Graph, G

$$G = (V, E)$$
$$V = \{a, b, c, d, e, f, g, h\}$$



A node (or vertex)

Label don't necessarily have to be boring numbers

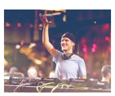
An edge (or link)

Graphs

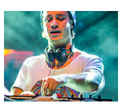
Graph, G

$$G = (V, E)$$

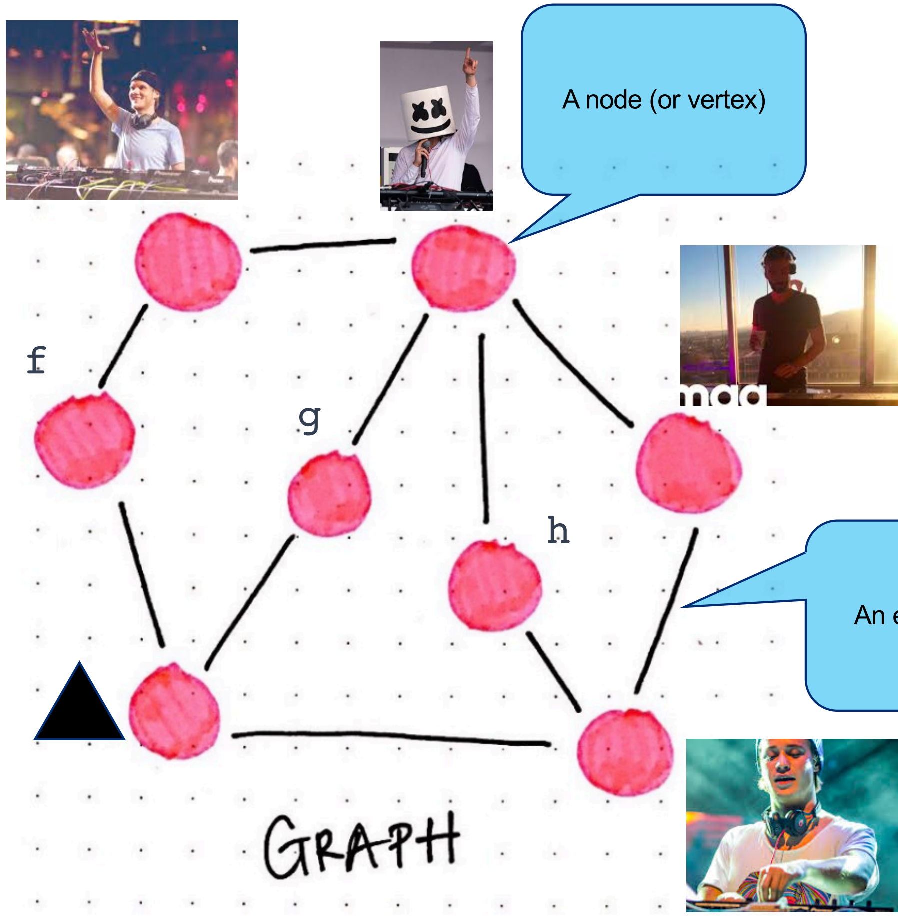
$$V = \{f, g, h, \blacktriangle,$$


,


,


,


$\}$



A node (or vertex)

Another example of the vertex set

An edge (or link)

Graphs

An edge is a tuple of two “connected” nodes

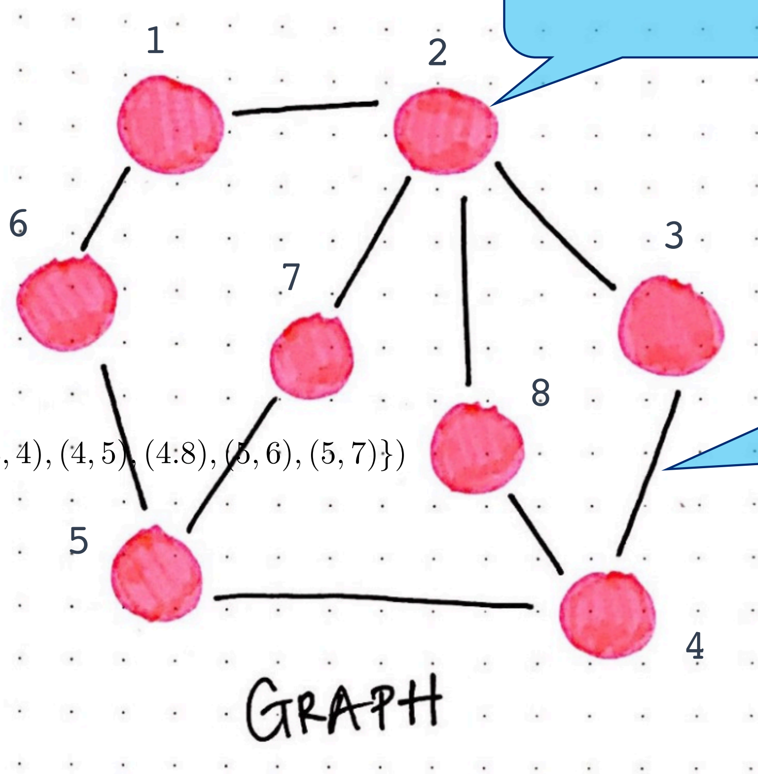
An edge set is the collection of all those tuples in the graph

$G = (V, E)$

$V = \{1, 2, 3, 4, 5, 6, 7, 8\}$

$E = \{(1,2), (1,6), (2,3), (2,7), (2,8), (3,4), (4,5), (4,8), (5,6), (5,7)\})$

In this class, we’ll assume that (1,2) and (2,1) refer to the same edge



A node (or vertex)

V and *E* are sets

An edge (or link)

For instance (1,2)

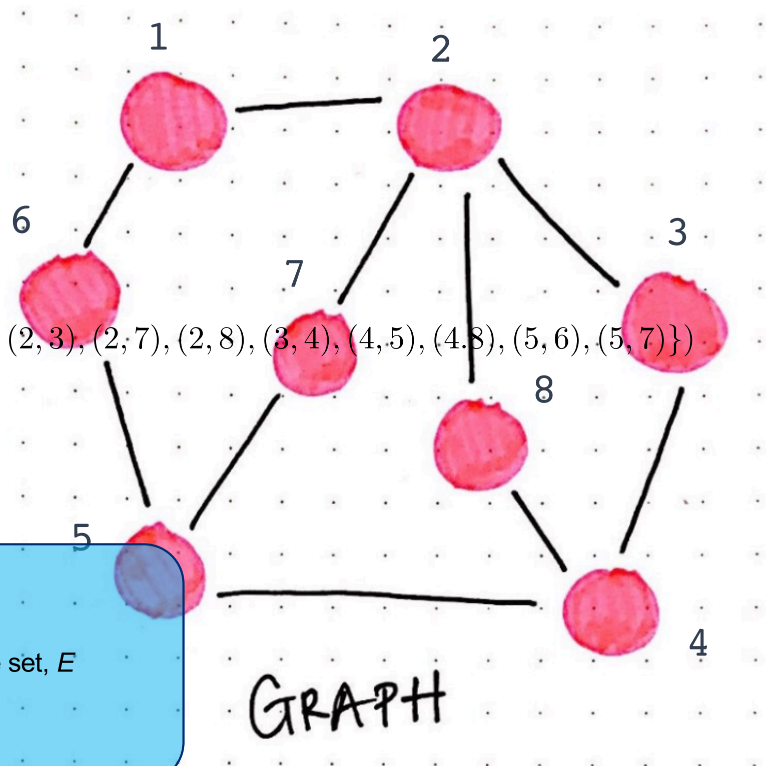
Graphs

G is a tuple of two sets

$$G = (\{1,2,3,4,5,6,7,8\}, \{(1,2),(1,6),(2,3),(2,7),(2,8),(3,4),(4,5),(4,8),(5,6),(5,7)\})$$

vertex set, *V*

Edge set, *E*



Why Bother?

Graph theory gives us the tools needed to model, and hence talk about, multiagent systems

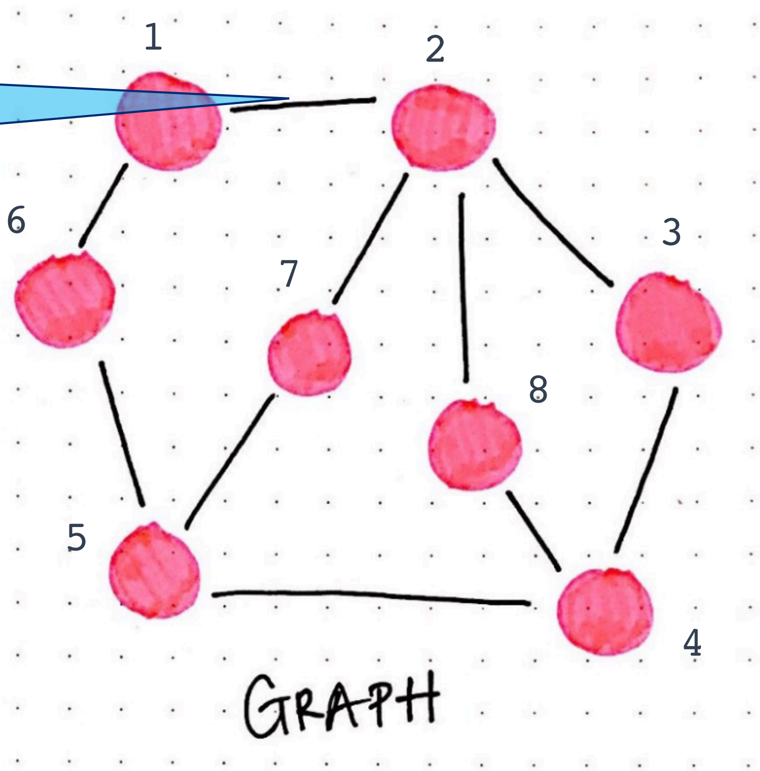
Where do the lines come from?

"Talk" can be actual talking, access to each other's state, a Facebook connection, and so on

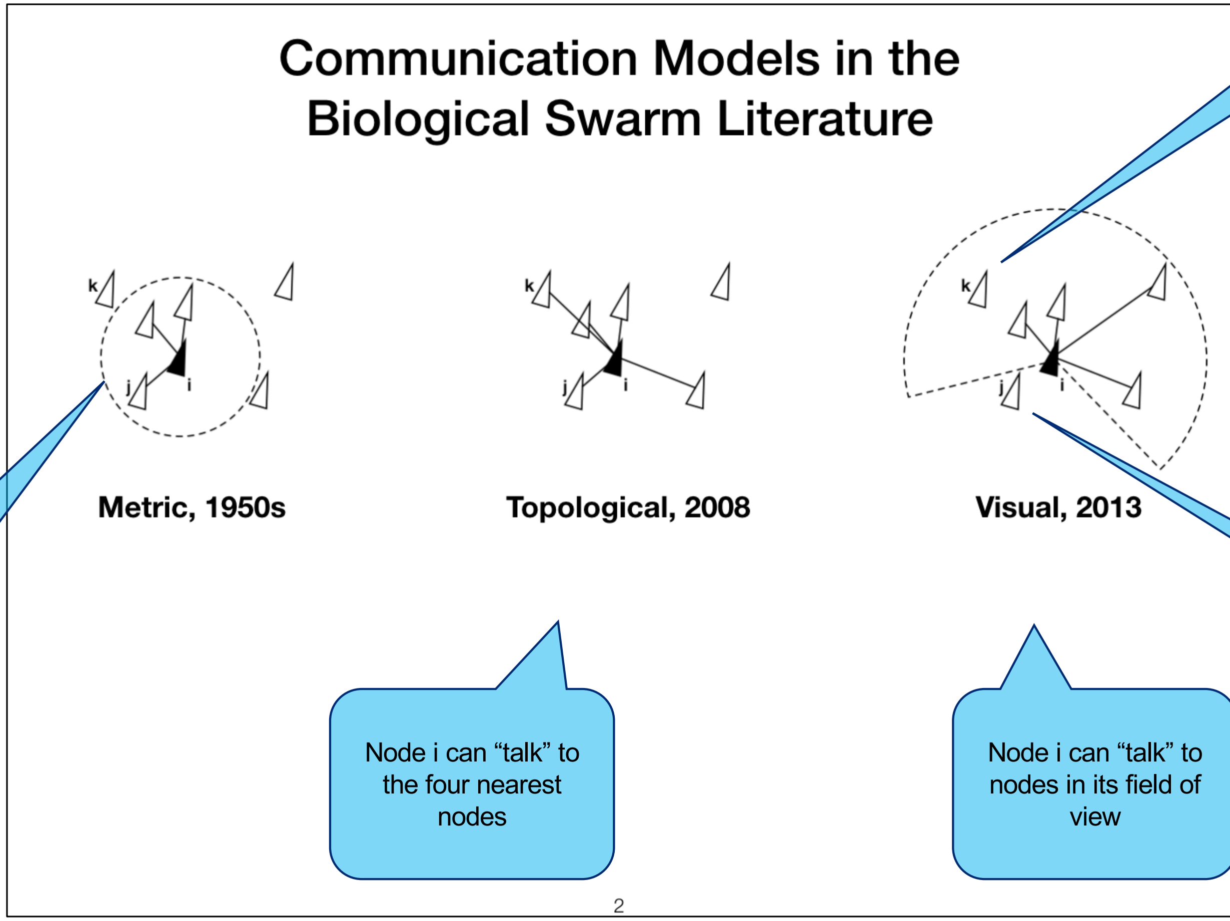
We usually say node 1 and node 2 can "talk" to each other

But why are nodes 1 and 2 connected by an edge?

The choice of a **communication model** governs who talks to whom (who?)



Examples of Communication Models

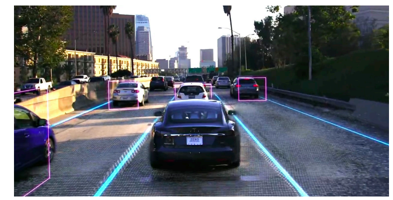


Node i can “talk” to anyone inside the circle around it

Node i can “talk” to the four nearest nodes

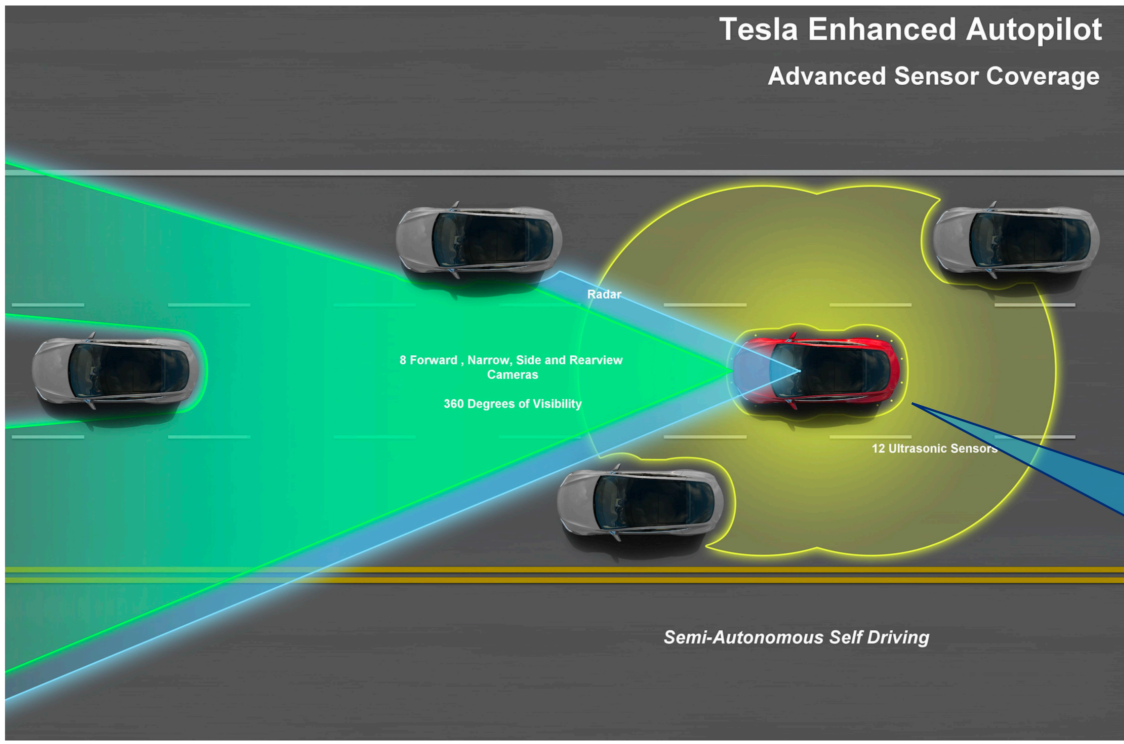
Node i can “talk” to nodes in its field of view

Features line of sight, which is why Node k is not a friend ☹



Features a blind spot and line of sight, that’s why j is not a friend ☹

Another Example of a Communication Model



sense vs talk

Neighborhoods

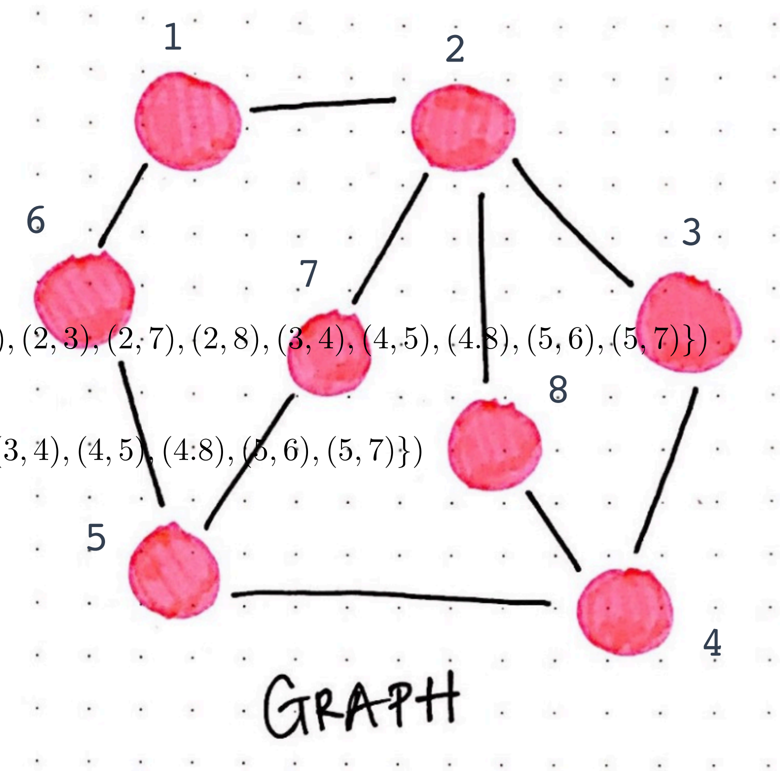
Neighborhood of Node i

$$N(i) = \{j \in V \mid (i,j) \in E\}$$

is the set of all nodes j in the vertex set

such that

the (i,j) tuple is in the edge set



$G = (\{1,2,3,4,5,6,7,8\}, \{(1,2),(1,6),(2,3),(2,7),(2,8),(3,4),(4,5),(4,8),(5,6),(5,7)\})$

$V = \{1,2,3,4,5,6,7,8\}$

$E = \{(1,2),(1,6),(2,3),(2,7),(2,8),(3,4),(4,5),(4{:}8),(5,6),(5,7)\})$

$N(1) = \{2,6\}$

$N(2) = \{1,3,7,8\}$

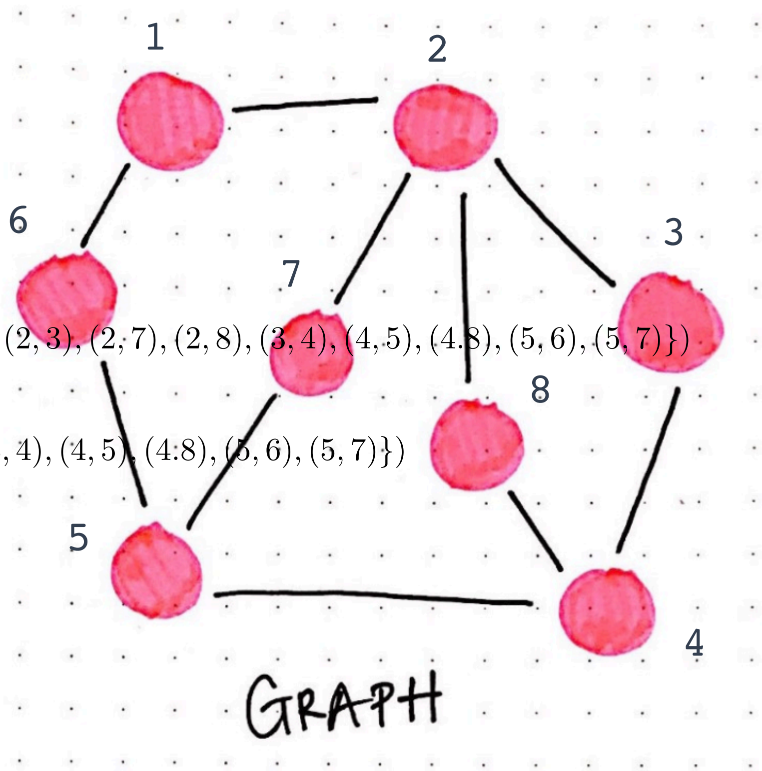
if there's a Node j that forms an edge with i, then j gets added to Node i's **neighborhood** set

Degree

Degree of Node i

$$N(i) = \{j \in V \mid (i,j) \in E\}$$
$$Deg(i) = |N(i)|$$

is the cardinality of Node i's neighborhood set

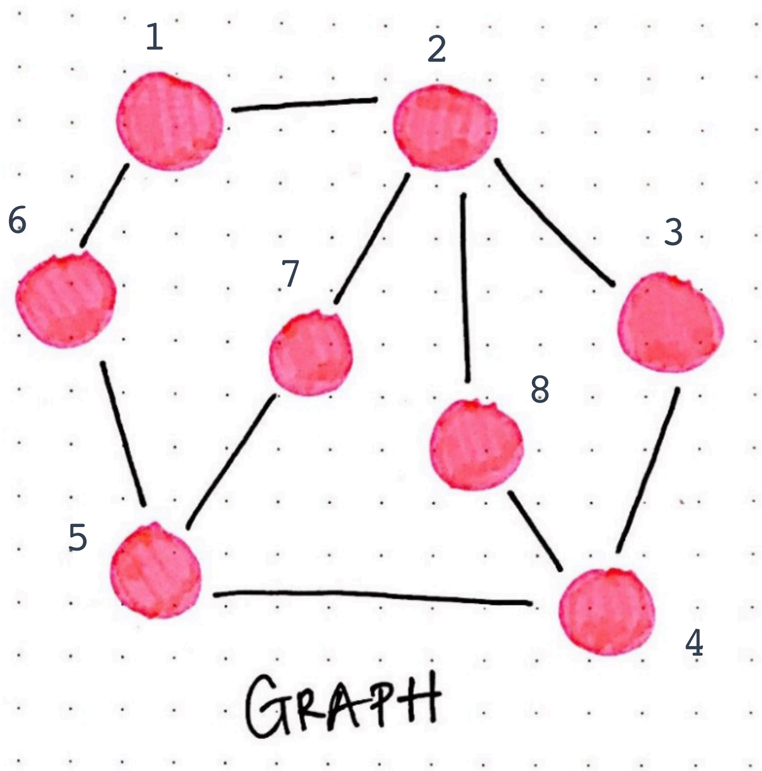


$$G = (\{1,2,3,4,5,6,7,8\}, \{(1,2),(1,6),(2,3),(2,7),(2,8),(3,4),(4,5),(4,8),(5,6),(5,7)\})$$
$$V = \{1,2,3,4,5,6,7,8\}$$
$$E = \{(1,2),(1,6),(2,3),(2,7),(2,8),(3,4),(4,5),(4,8),(5,6),(5,7)\})$$
$$N(1) = \{2,6\} \qquad Deg(1) = 2$$
$$N(2) = \{1,3,7,8\} \qquad Deg(2) = 4$$

Degree of Node i is its number of neighbors

Connected

A graph is **connected** if there's a path along the edges between any two nodes



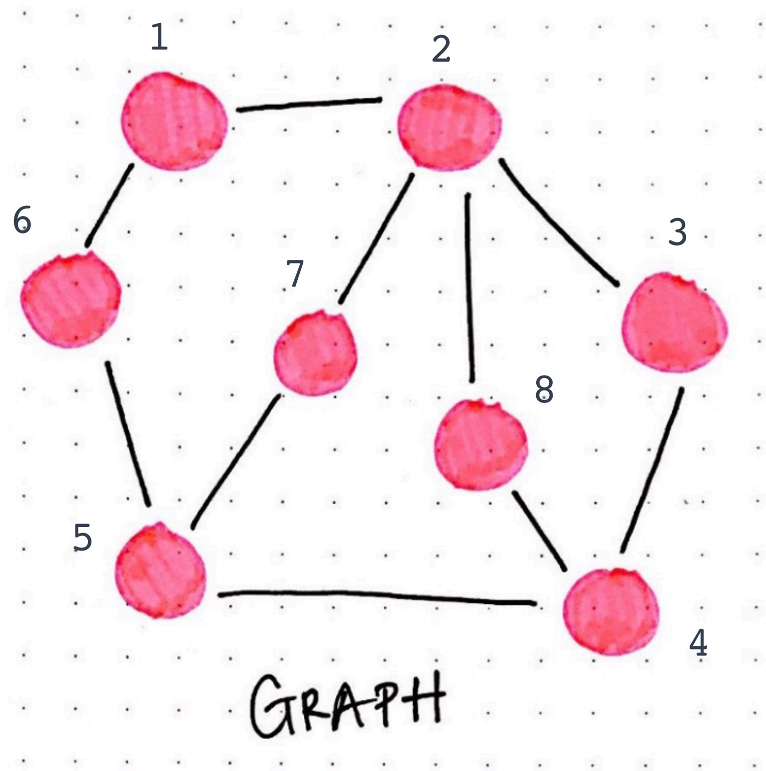
This is a connected graph

Information between Nodes 7 and 8 can flow via Node 2, for instance

Connected Components

A **connected component** is the largest collection of nodes that are connected

$V = \{1, 2, 3, 4, 5, 6, 7, 8\}$

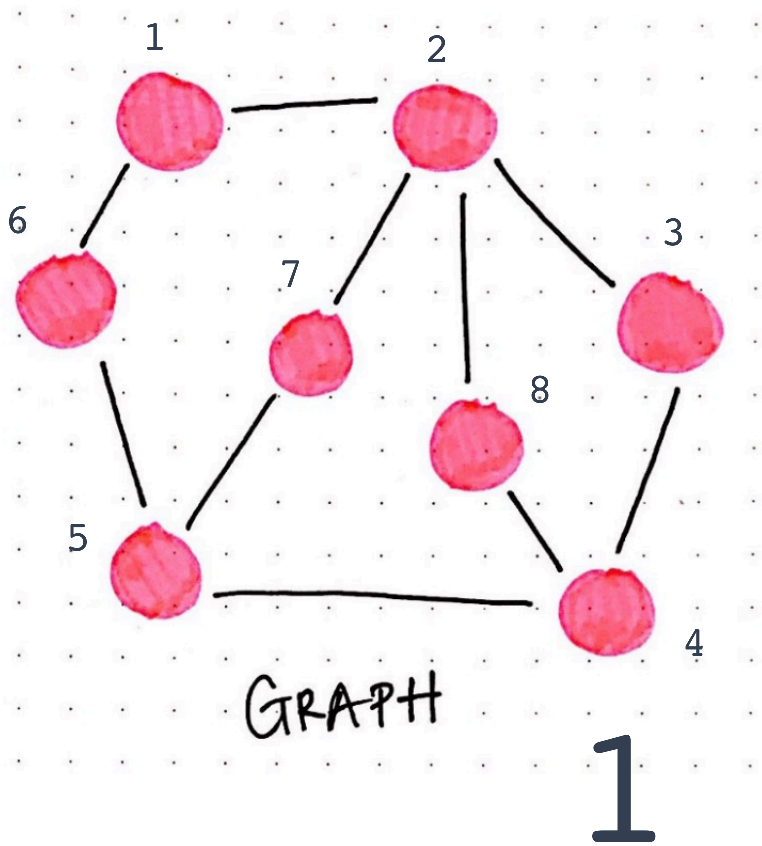


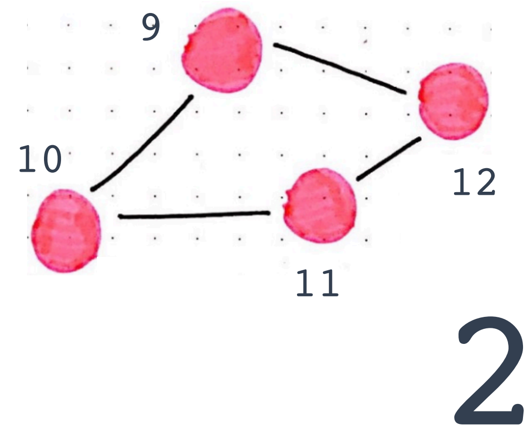
There is only one connected component here, the entire graph itself

Connected Components

A **connected component** is the largest collection of nodes that are connected

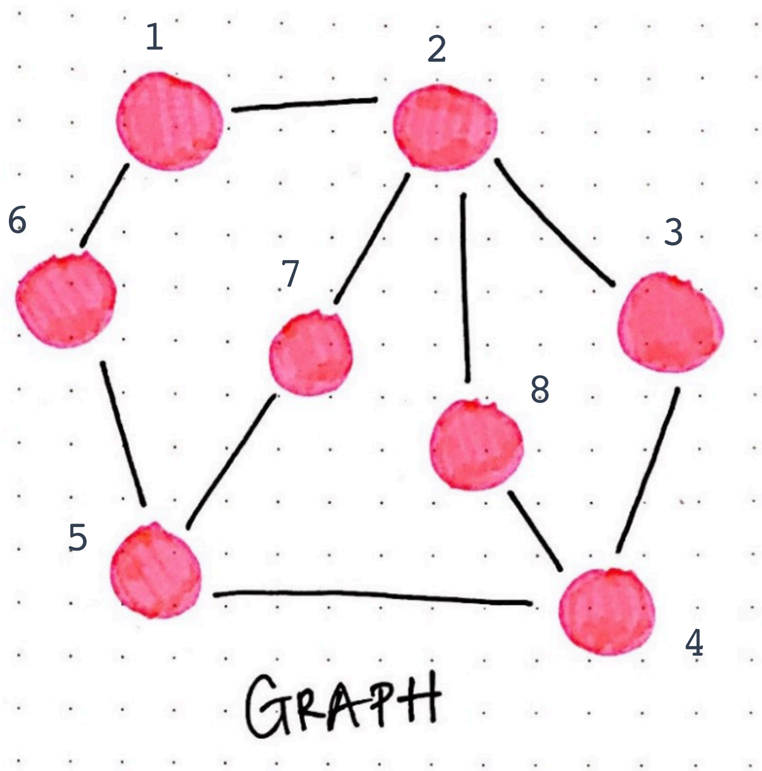
$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$





This new graph, which consists of 12 nodes, has two connected components

Isolated Components

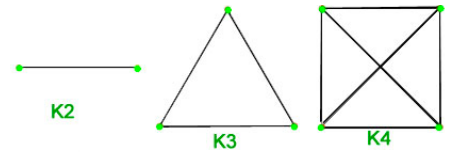


An **isolated component** has no neighbors

$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

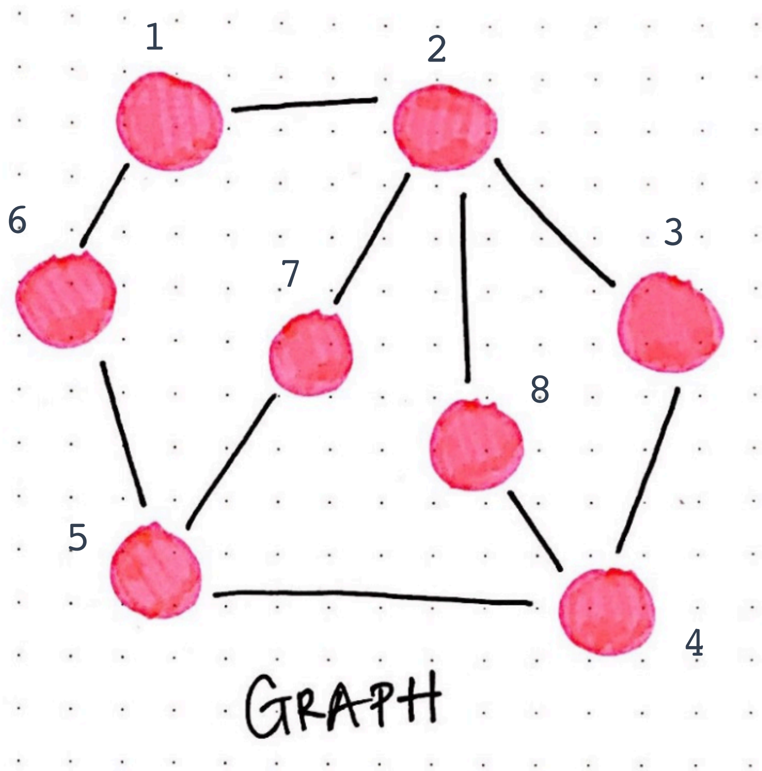
Node 9 is isolated

Completeness



Shorthand notation for complete graphs, K_N, where N is the number of nodes

A graph is **complete** if every node can directly "talk" to all the other nodes (loose definition)

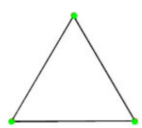
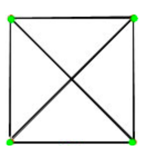


Not a complete graph

Nodes 3 and 5 are not neighbors, for instance.

They are, however, one-hop neighbors (via Node 4)

Links in a Complete Graph

	Shorthand Notation	Number of Nodes in the Graph (N)	Number of Links
	K_1	1	0
	K_2	2	1
	K_3	3	3
	K_4	4	6
	K_N	N	N*(N-1) / 2

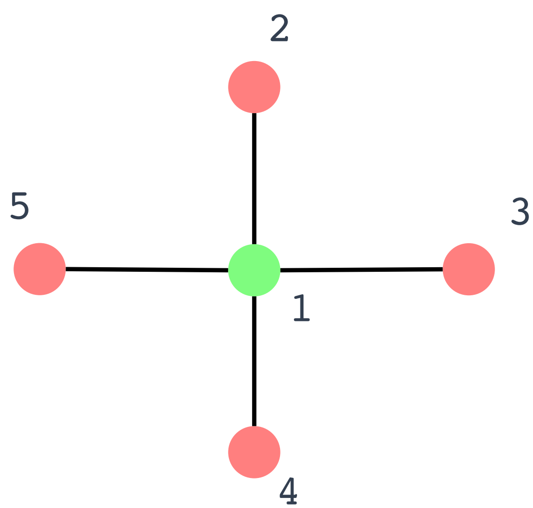
Clustering Coefficient

The **clustering coefficient** measures the cliquishness of a node

Is that node part of a bigger group?

Measure between 0 and 1

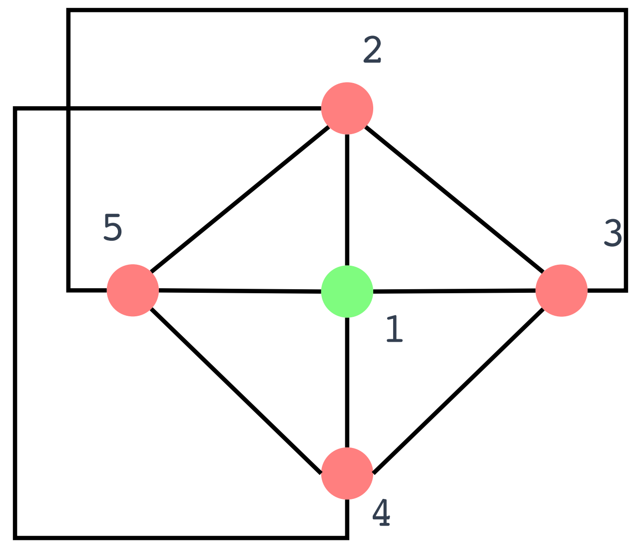
The **swarm clustering coefficient** is the average clustering coefficient of the nodes in the graph



$CC(1) = 0$

Node 1 has 4 friends

None of those 4 friends are friends with each other

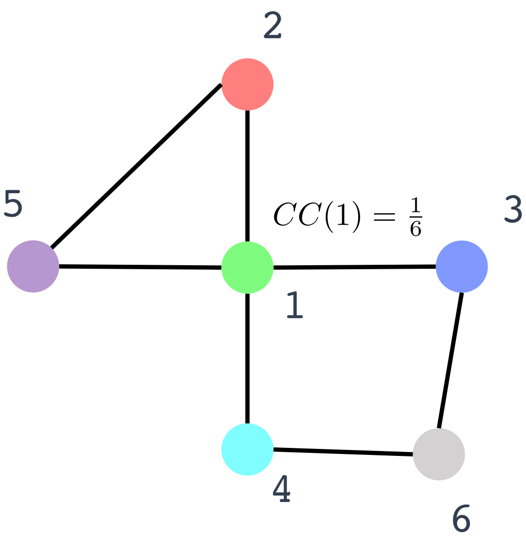


$CC(1) = 1$

Node 1 has 4 friends

Any pair of friends of Node 1 are friends with each other

Calculating the Clustering Coefficient of a Node



Let's focus on Node 1

Step 1. Determine Deg(1) (i.e., the degree of Node 1)?

Step 2. How many links are there in a complete graph with Deg(1) nodes?

In other words, how many links are there in the $K_{Deg(1)}$ graph?
Set this to L

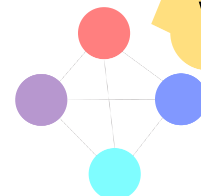
Step 3. How many links *are there* between neighbors of Node 1?

N(1) = { }

Deg(1) = | N(1) | = 4

K_4 has 4*(4-1) / 2 links = 6

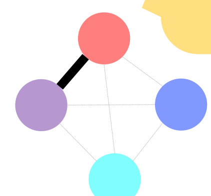
if all of Node 1's friends were friends with each other, their graph without would look like this



A graph of just Node 1's friends

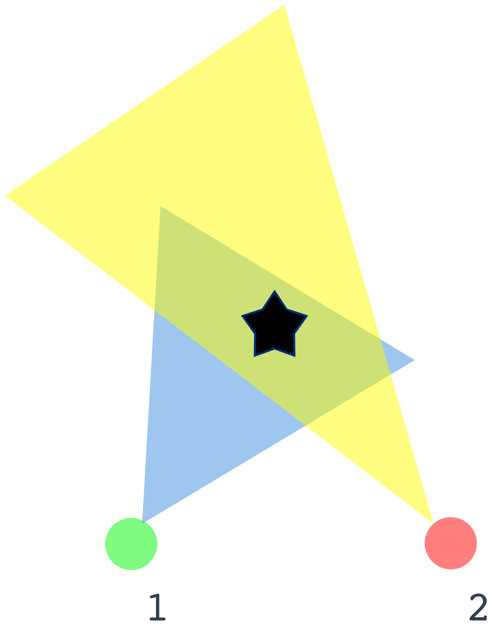
But that's not the cases

There's only one link



1

Agreement Protocol



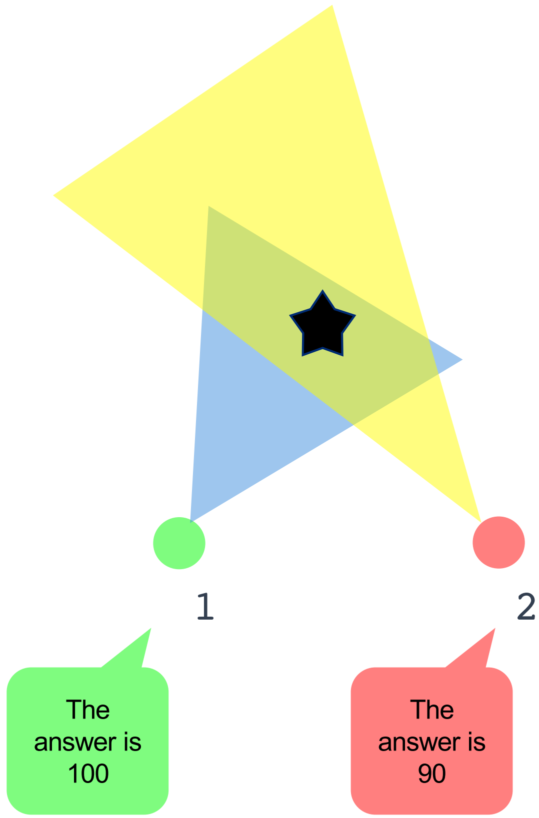
Here's two robots (or agents)

Imagine they are estimating something

Agreement Protocol

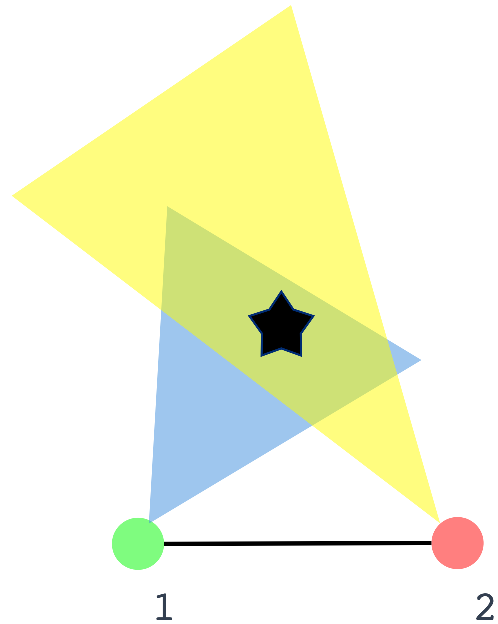
Here's two robots (or agents)

Imagine they are estimating something



Agreement Protocol

Whenever these **cooperative robots** can talk to each other they want to come to an agreement over what they've measured



The answer is 100

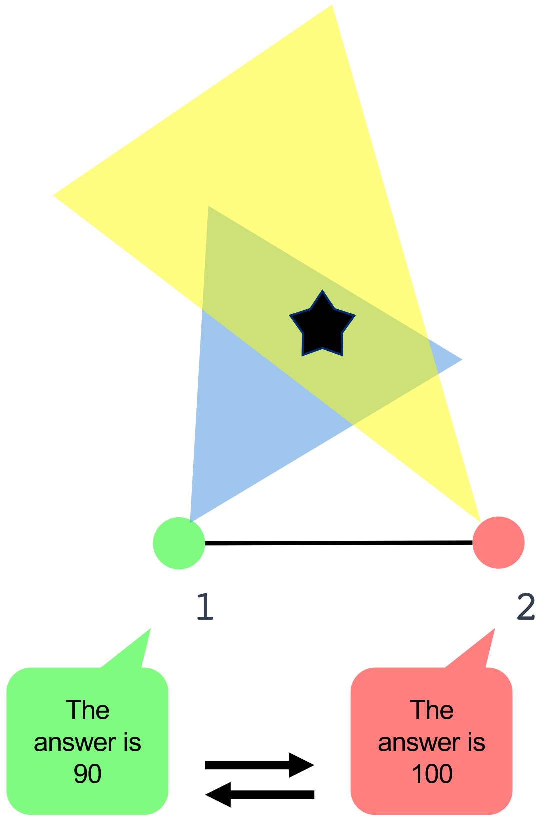
The answer is 90

Old estimate needs to be updated since I can now talk to Robot 2

Old estimate needs to be updated since I can now talk to Robot 1

Agreement Protocol

Bad idea: just set my estimate to my friend's estimate



you've basically swapped values

Consensus Equation

How much should Node i change it's estimate

weight

Node i's estimate

Node j's estimate

$$\dot{x}_i = -\omega \sum_{j \in N(i)} (x_i - x_j)$$

Sum over every j that is in the neighborhood of i

Any type of agreement protocol realizes the answer is somewhere in the middle

The **Consensus Equation** is one such algorithm that these robots can use

Node i's new estimate

$$x_i^{new} = x_i^{old} + (\Delta t * \dot{x}_i)$$

time step

Do this for all Node i's in the vertex set of the graph

$$\forall\ i \in V$$

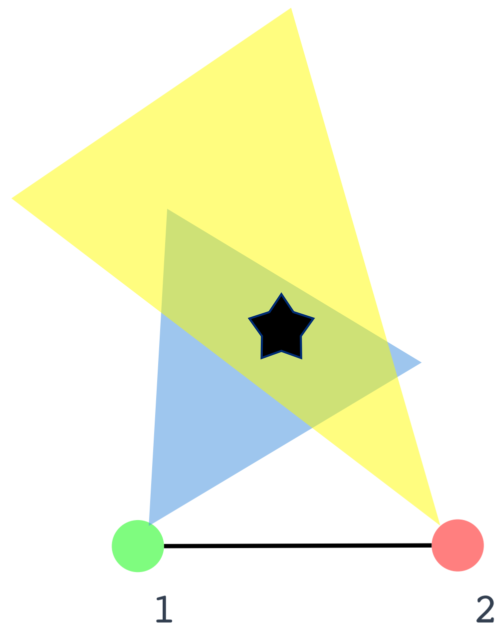
Consensus Equation

$$\Delta t = 0.20$$

You can play with this value

$$N(1) = \{2\}$$

$$N(2) = \{1\}$$



$$x_1 = 100$$

$$x_2 = 90$$

$$\dot{x}_1 = -(100 - 90) = -10$$

$$\dot{x}_2 = -(90 - 100) = +10$$

$$x_1 = 100 + (0.20 * (-10)) = 98$$

$$x_2 = 90 + (0.20 * (+10)) = 92$$

The answer is 100

The answer is 90

The answer is 98

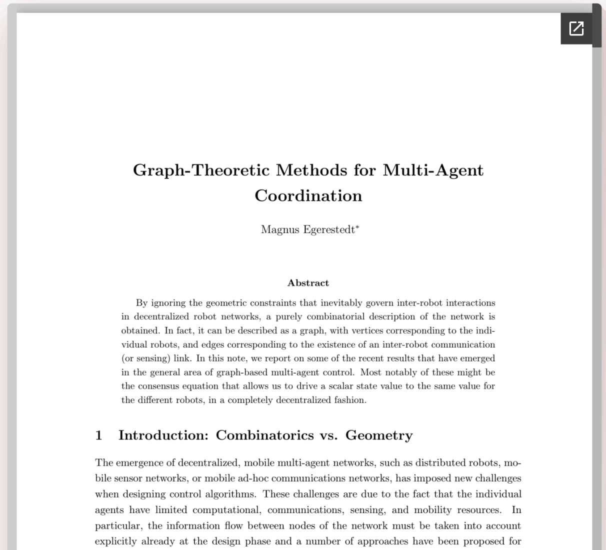
The answer is 92

And so on…

If they can continue to talk, this is how they would come to an agreement

Additional Resources

https://magnus.ece.gatech.edu/Papers/ROBOMAT07.pdf



Graph-Theoretic Methods for Multi-Agent Coordination

Magnus Egerestedt*

Abstract

By ignoring the geometric constraints that inevitably govern inter-robot interactions in decentralized robot networks, a purely combinatorial description of the network is obtained. In fact, it can be described as a graph, with vertices corresponding to the individual robots, and edges corresponding to the existence of an inter-robot communication (or sensing) link. In this note, we report on some of the recent results that have emerged in the general area of graph-based multi-agent control. Most notably of these might be the consensus equation that allows us to drive a scalar state value to the same value for the different robots, in a completely decentralized fashion.

1 Introduction: Combinatorics vs. Geometry

The emergence of decentralized, mobile multi-agent networks, such as distributed robots, mobile sensor networks, or mobile ad-hoc communications networks, has imposed new challenges when designing control algorithms. These challenges are due to the fact that the individual agents have limited computational, communications, sensing, and mobility resources. In particular, the information flow between nodes of the network must be taken into account explicitly already at the design phase and a number of approaches have been proposed for

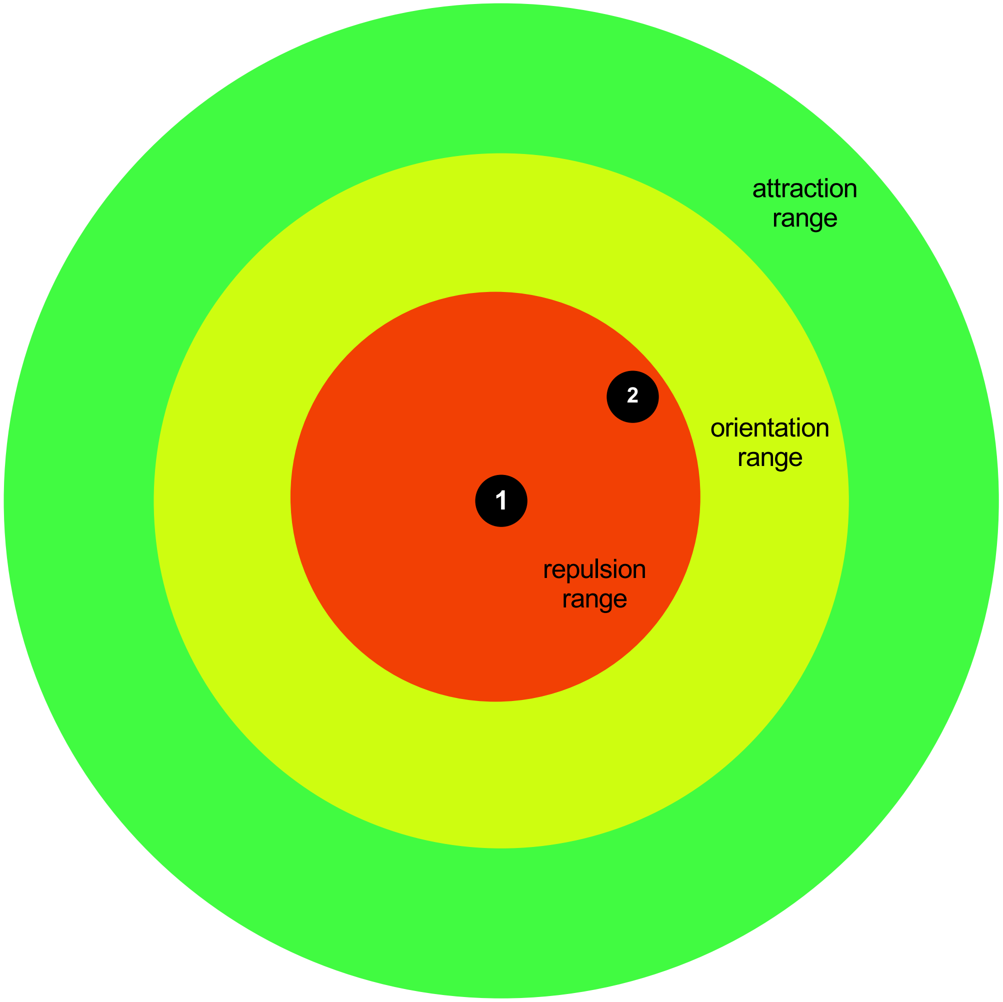
https://www.youtube.com/watch?v=ULKyXnQ9xWA

Not required

Boids Rule

Consensus can be one way agents in the network think

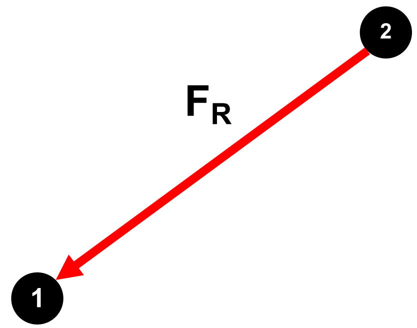
Boids Rule is one way in which the agents can move



Node 2 is in Node 1's repulsion range

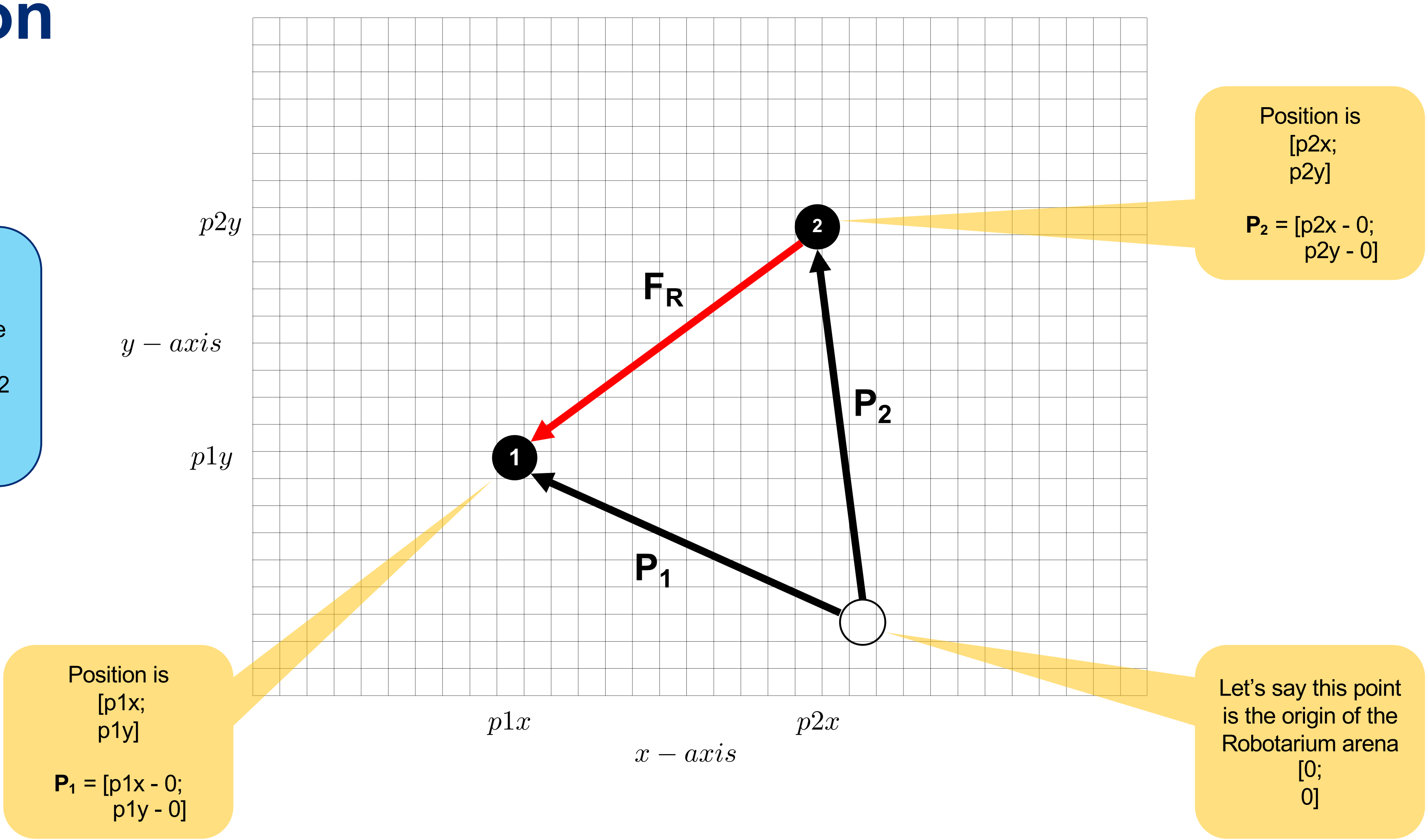
Repulsion

To repulse, need a force to act on Node 1 that points away from Node 2

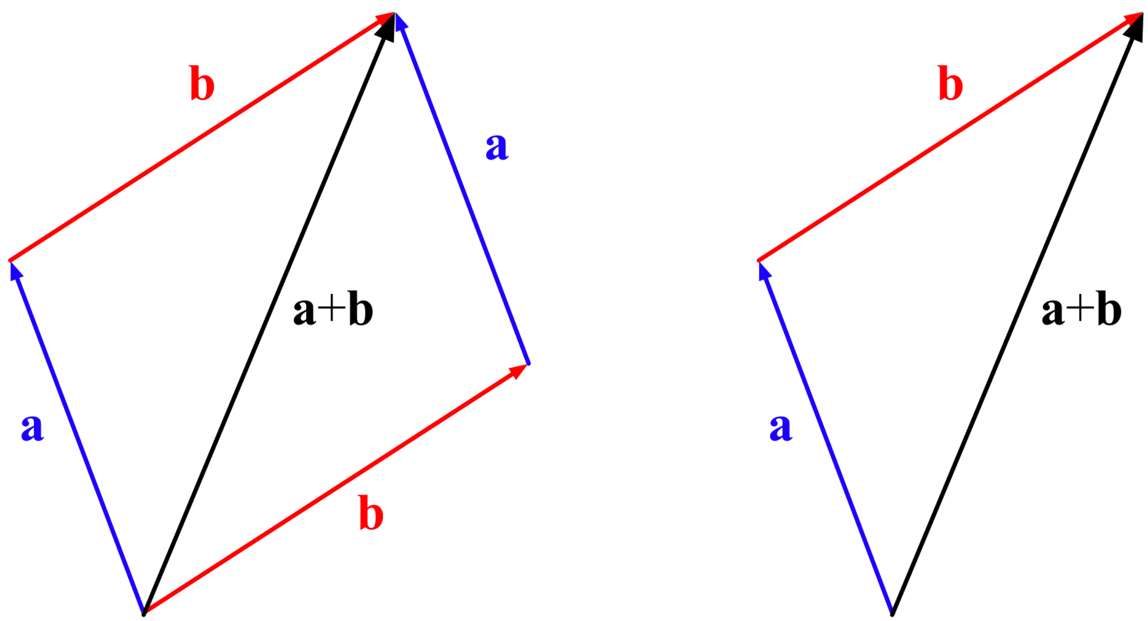


Repulsion

To repulse, need a force to act on Node 1 that points away from Node 2



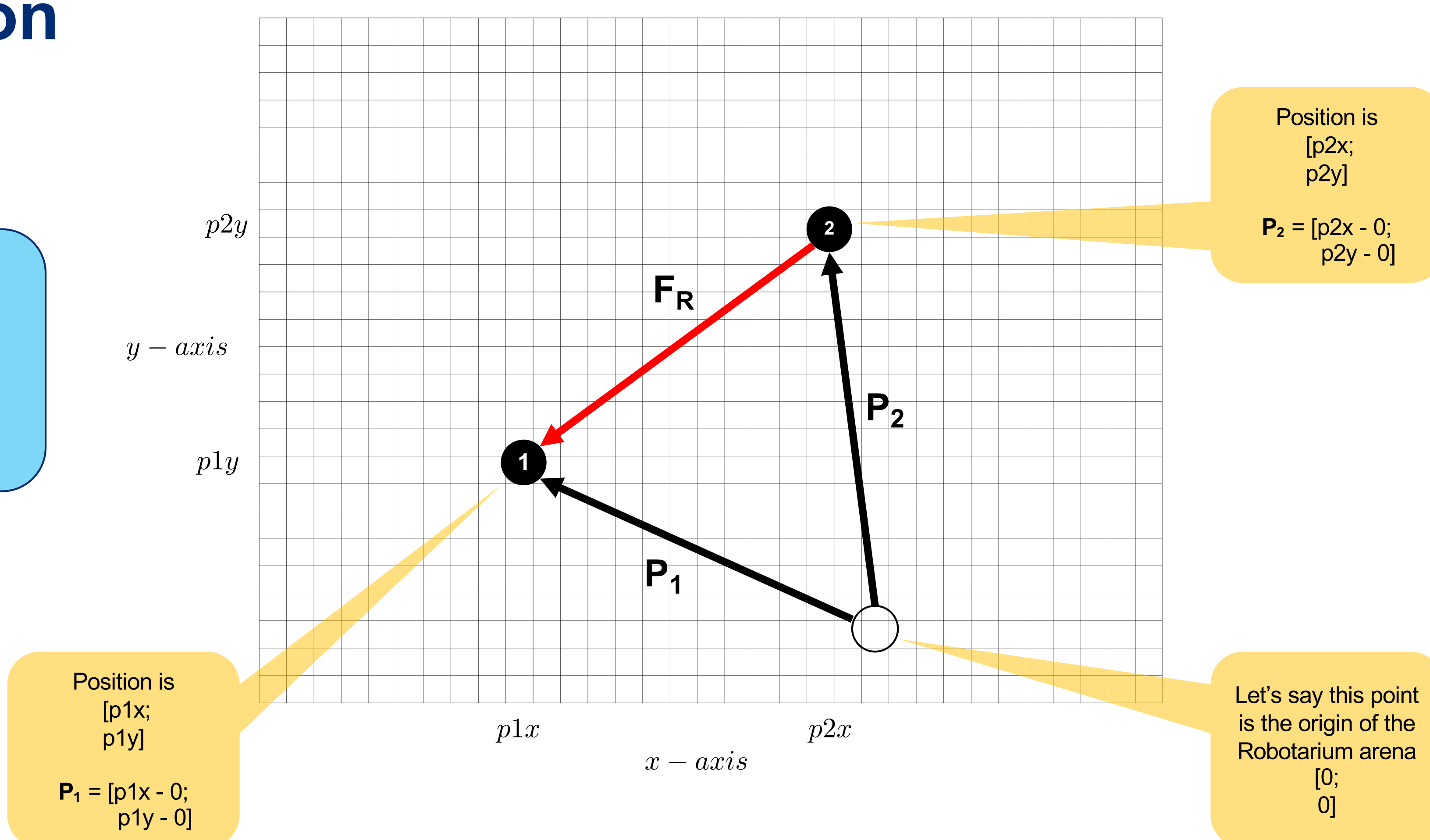
Vector Addition



wikipedia

Repulsion

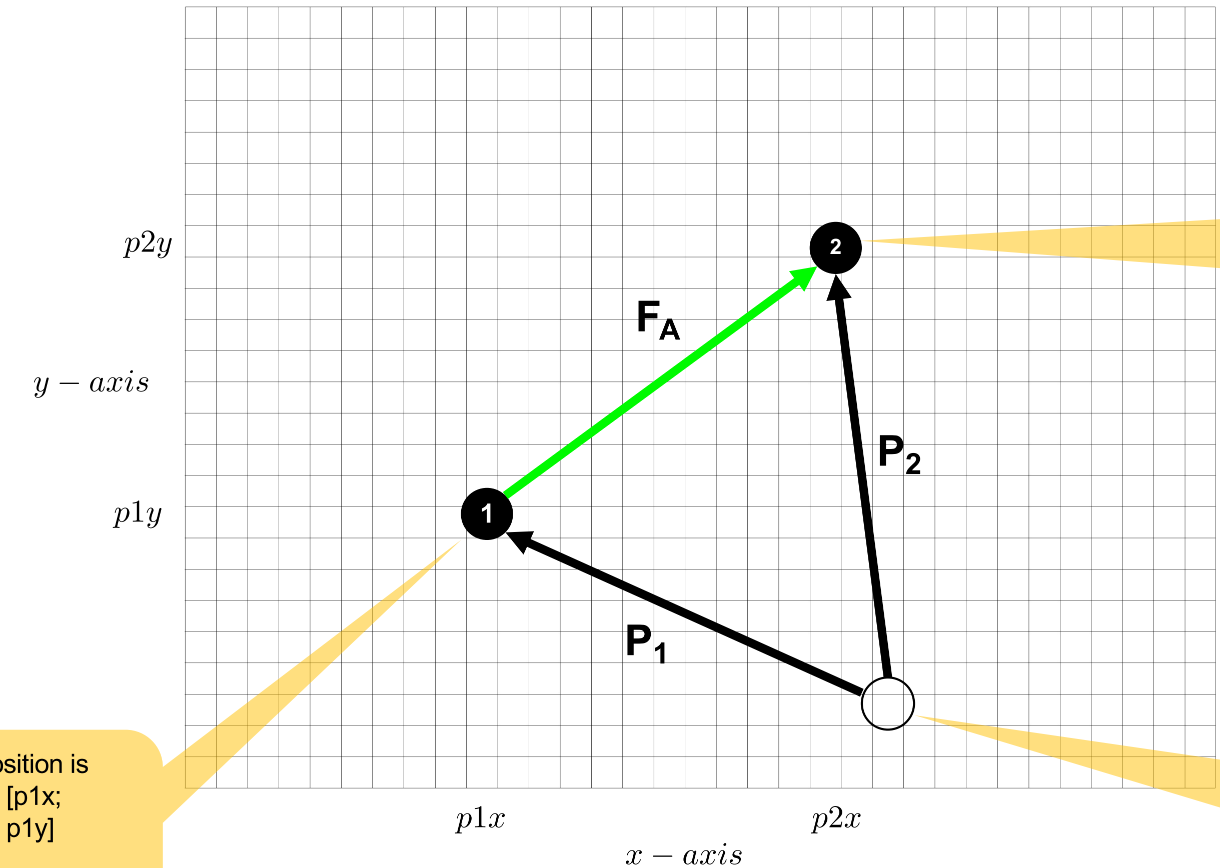
$F_R = P_1 - P_2$



Attraction

Consider the opposite scenario where Node 1 needs to move toward Node 2

$\mathbf{F_A}$ = ?



Position is
[p2x;
p2y]

$\mathbf{P_2}$ = [p2x - 0;
p2y - 0]

Position is
[p1x;
p1y]

$\mathbf{P_1}$ = [p1x - 0;
p1y - 0]

Let’s say this point is the origin of the Robotarium arena
[0;
0]

Additional Resources

http://www.red3d.com/cwr/papers/1987/SIGGRAPH87.pdf

Flocks, Herds, and Schools: A Distributed Behavioral Model [1]

Craig W. Reynolds
Symbolics Graphics Division

[obsolete addresses removed[2]]

Abstract
The aggregate motion of a flock of birds, a herd of land animals, or a school of fish is a beautiful and familiar part of the natural world. But this type of complex motion is rarely seen in computer animation. This paper explores an approach based on simulation as an alternative to scripting the paths of each bird individually. The simulated flock is an elaboration of a particle system, with the simulated birds being the particles. The aggregate motion of the simulated flock is created by a distributed behavioral model much like that at work in a natural flock; the birds choose their own course. Each simulated bird is implemented as an independent actor that navigates according to its local perception of the dynamic environment, the laws of simulated physics that rule its motion, and a set of behaviors programmed into it by the "animator." The aggregate motion of the simulated flock is the result of the dense interaction of the relatively simple behaviors of the individual simulated birds.

Categories and Subject Descriptors: 1.2.10 [Artificial Intelligence]: Vision and Scene Understanding; 1.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; 1.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism-Animation: 1.6.3 [Simulation and Modeling]: Applications.

General Terms: Algorithms, design.

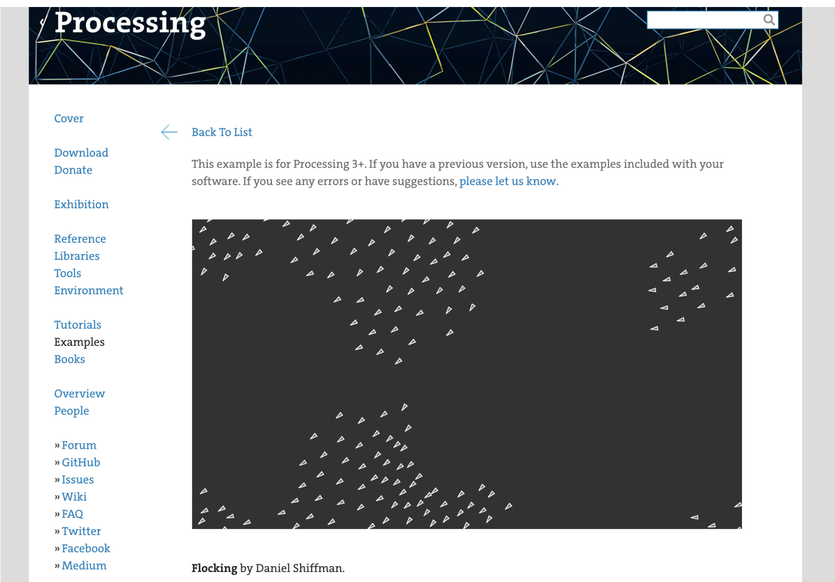
Additional Key Words, and Phrases: flock, herd, school, bird, fish, aggregate motion, particle system, actor, flight, behavioral animation, constraints, path planning.

[1] **Note**: this is a reprint of the original publication in the proceeding of SIGGRAPH '87 (Computer Graphics 21(4), July 1987, edited by Maureen C. Stone, pages 25-34). It was produced by applying optical character recognition software to scanned images of the original hardcopy pages. The author wishes to thank Ken Cushman of SGI who generously donated his time and facilities to perform the OCR work, which allowed this old paper to get back online. This version (5-14-95) contains no illustrations, font styles have been lost and not fully restored. Be forewarned: the OCR process introduces errors into the text. Most of these have been corrected through spell-checking and spotty proof-reading. Some errors may persist.

[2] Author's current address: Silicon Studio, 2011 North Shoreline Boulevard, MS 980, Mountain View, CA 94043, USA --

Not required

Additional Resources



Not required

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY