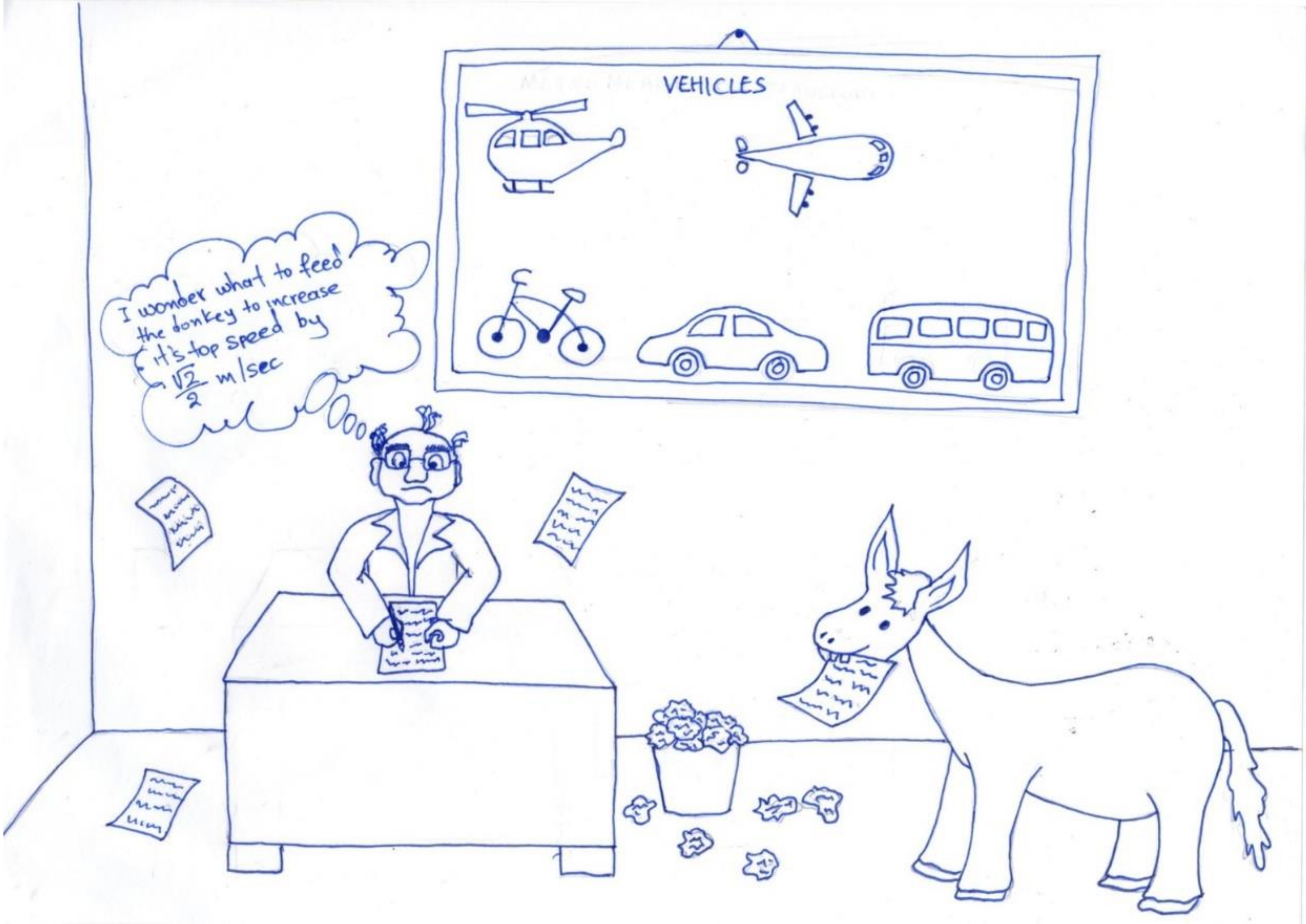
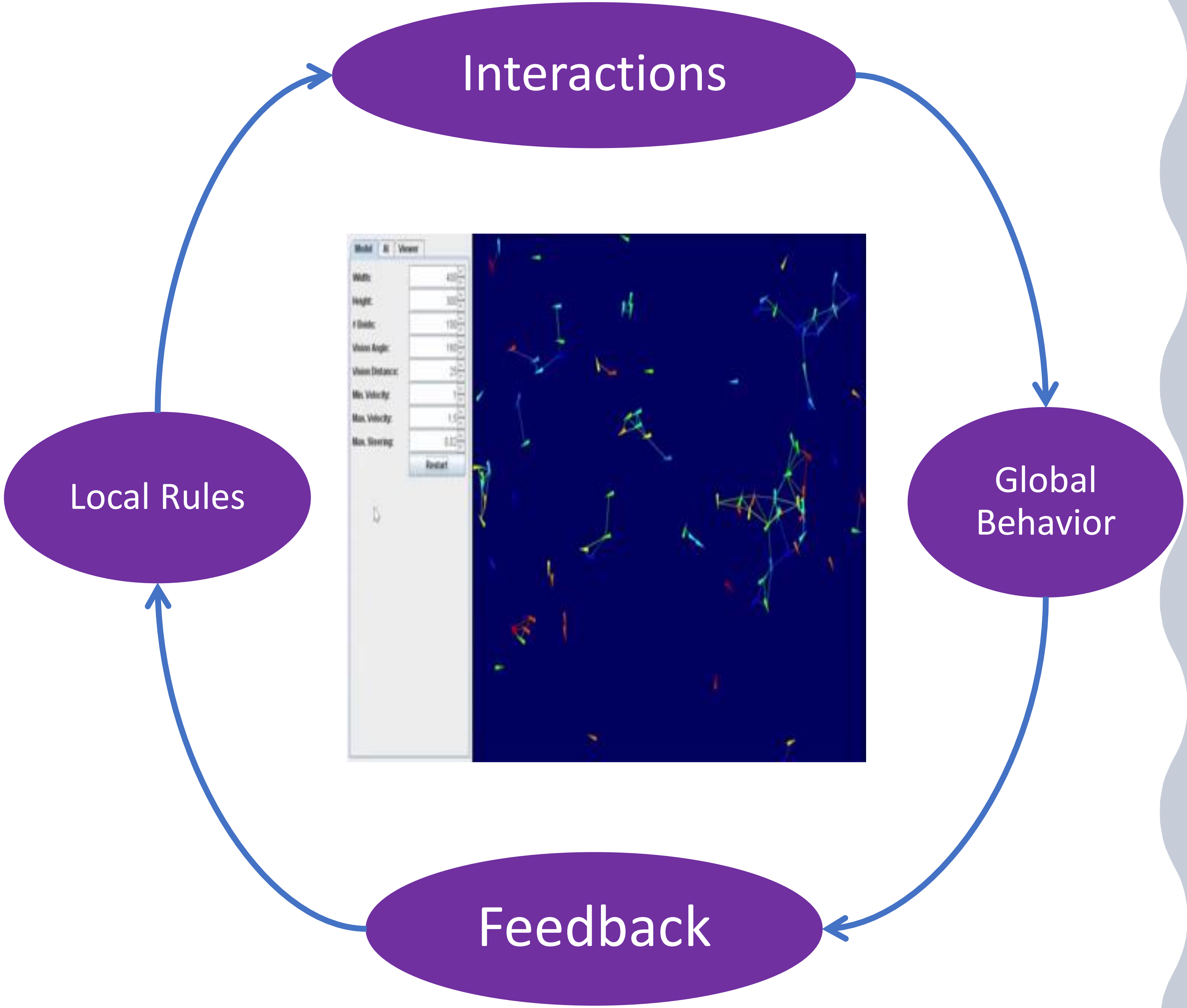


VEHICLES
I wonder what to feed the donkey to increase it's top speed by $\frac{\sqrt{2}}{2}$ m/sec



Αποκεντρωμένος Υπολογισμός & Μοντελοποίηση

Κωνσταντίνος Τσίχλας, Αν. Καθηγητής

Βασίλης Θωμόπουλος, Υπ. Διδάκτορας

Αποκεντρωμένος Υπολογισμός & Μοντελοποίηση

Υπεύθυνος μαθήματος: Τσίχλας Κωνσταντίνος

e-mail: ktsichlas@ceid.upatras.gr

Ιστότοπος: https://eclass.upatras.gr/courses/CEID1220/

Τηλ: 2610-996908

Γραφείο: 1^{ος} όροφος, Γρ. 4

Επικουρικό Έργο: Υπ. Διδάκτορας Θωμόπουλος Βασίλειος

email: thomopul@ceid.upatras.gr

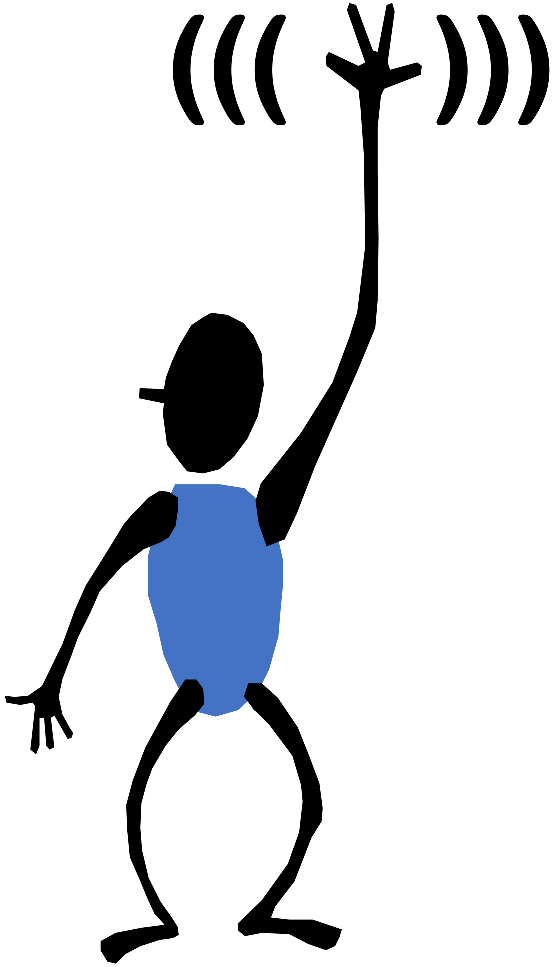
Γραφείο: Εργαστήριο ML@Cloud, 2^{ος} Όροφος

Σχετικά με το μάθημα...

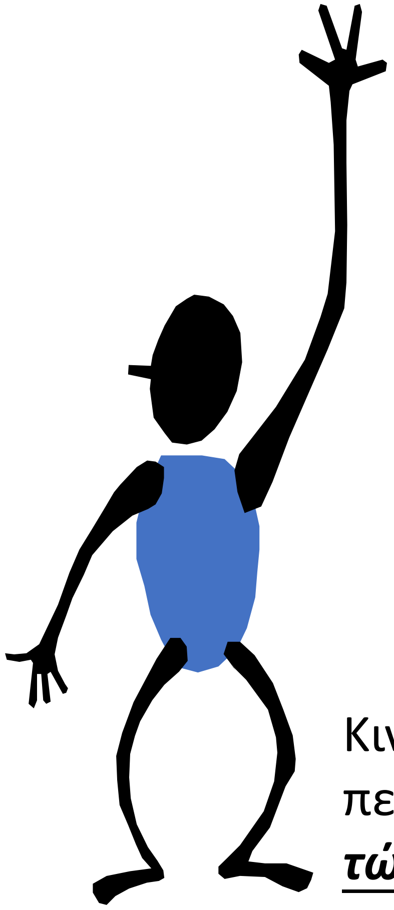
- Ώρες Γραφείου: Οποτεδήποτε (στείλτε email και κανονίζουμε)
- email: ktsichlas@upatras.gr
- Δικτυακός Τόπος Μαθήματος: https://eclass.upatras.gr/courses/CEID1220/
- https://eclass.upatras.gr/courses/CEID1227/ (ΥΔΑ)
- Ώρες Μαθήματος:
  - **Τετάρτη (10:00-12:00 Θ) 12:00-13:00 (Φ) – Αίθουσα Ε2**
  - Εργαστήριο: Θα κανονιστεί – Εξ’ Αποστάσεως (zoom)
- Τρόπος Εξέτασης: Εργασίες

Κάποιες Συμβουλές

- Διακόψτε με ερωτήσεις (εδώ και αν ξεφεύγω!!!)
- Παρακολουθήστε τις διαλέξεις (ελπίζω να έχουν πλάκα – θα χαθείτε διαφορετικά)
- Αν δεν καταλαβαίνετε κάτι ελάτε στο γραφείο μου (ίσως το καταλάβουμε μαζί)
- Δουλέψτε τη NETLOGO



Κινούμενο χέρι: Θέλετε κάτι να πείτε σε σχέση με αυτά που ***<u>λέω τώρα</u>***



Σταθερό χέρι: Ερώτηση ή σχόλιο γενικής φύσης

Κάποιες Συμβουλές (Ίσχυαν επί COVID)

- Διακόψτε με ερωτήσεις (μερικές φορές ξεφεύγω)
- Παρακολουθήστε τις διαλέξεις (ελπίζω να έχουν πλάκα)
- Λύστε ασκήσεις (ας είναι ήδη λυμένες)
- Αν δεν καταλαβαίνετε κάτι στείλτε ένα email και είτε σας το εξηγώ ή κανονίζουμε ένα skype (ίσως το καταλάβουμε μαζί)

Θέλετε κάτι να πείτε σε σχέση με αυτά που ***λέω τώρα***

Ερώτηση ή σχόλιο γενικής φύσης

Η πιο Δύσκολη Ερώτηση: Γιατί να Πάρω αυτό το Μάθημα;

Σύνολο
7,56
8,47
0,75
0,90
10,61
11,03
5,25
9,54
11,35
7,08
8,32
9,67
12,77
12,47
9,54
9,08
2,21
13,16

- Κατανεμημένος (π.χ., clusters) και αποκεντρωμένος (π.χ., blockchains) υπολογισμός παντού.
- Φυσικά αλλά και τεχνητά πολύπλοκα συστήματα (αποικία μυρμηγκιών – ομάδα συνεργαζόμενων ρομπότ). Σχεδίαση – Ανάλυση
- Γιατί έχει πλάκα ☺

❖ Για τον βαθμό:

\+

- Πρόκληση
- Έχει Πλάκα
- Εφαρμογές;;;;

–

- Δύσκολο;
- Δεν υπάρχει βιβλίο
- Είναι η τρίτη χρονιά που γίνεται και θα το αλλάξω αρκετά
- Δεν έχω βρει κάτι αντίστοιχο και στο εξωτερικό…
- Κάτι δεν μου αρέσει σε αυτή τη διαφάνεια…

When I Started…

Construct

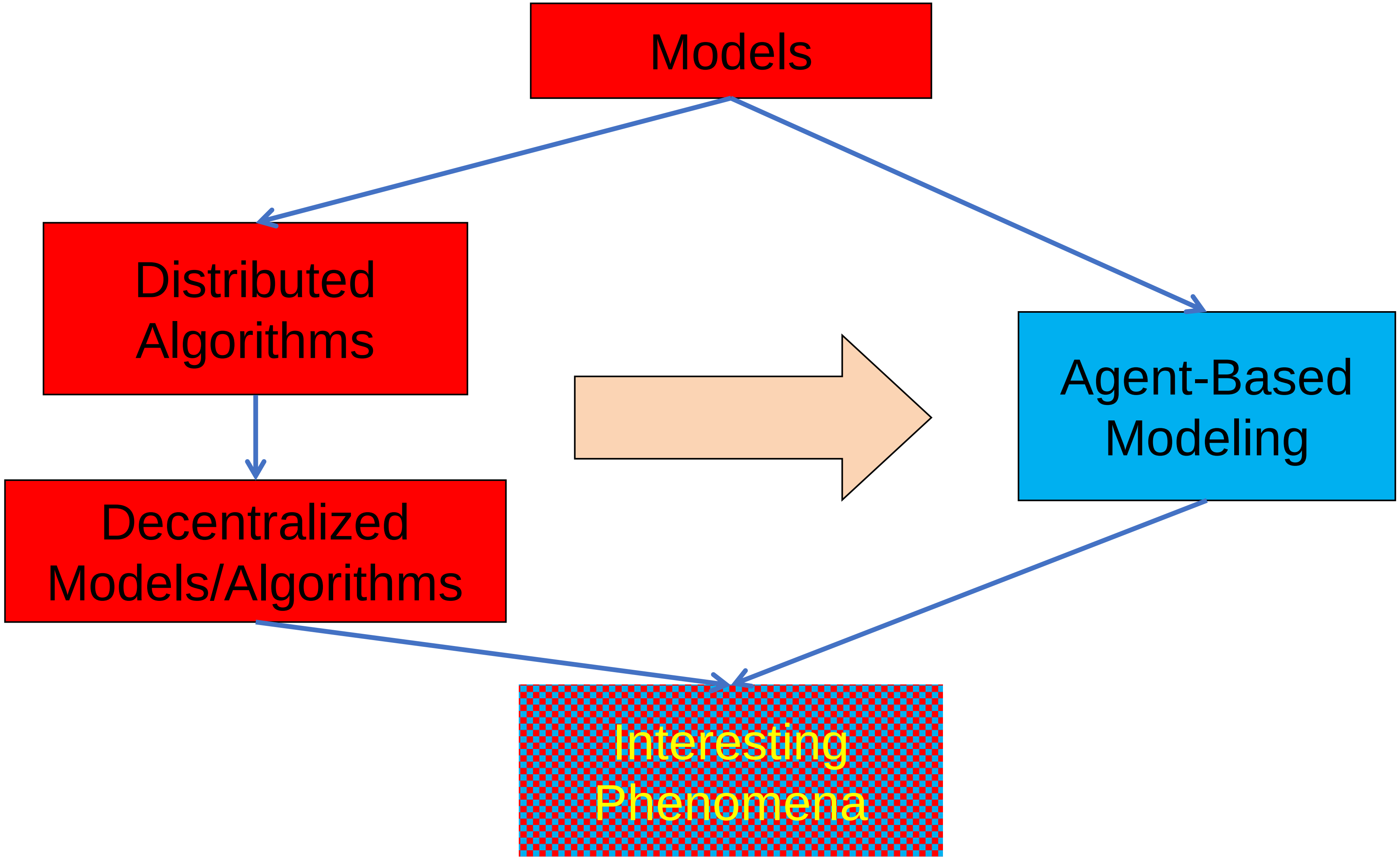
After I Started…

Understand

Way After I Started…

Explain

Content


Models
Distributed Algorithms
Decentralized Models/Algorithms
Agent-Based Modeling
Interesting Phenomena

Θ + Φ – Δεν θα τα καταφέρω όλα…

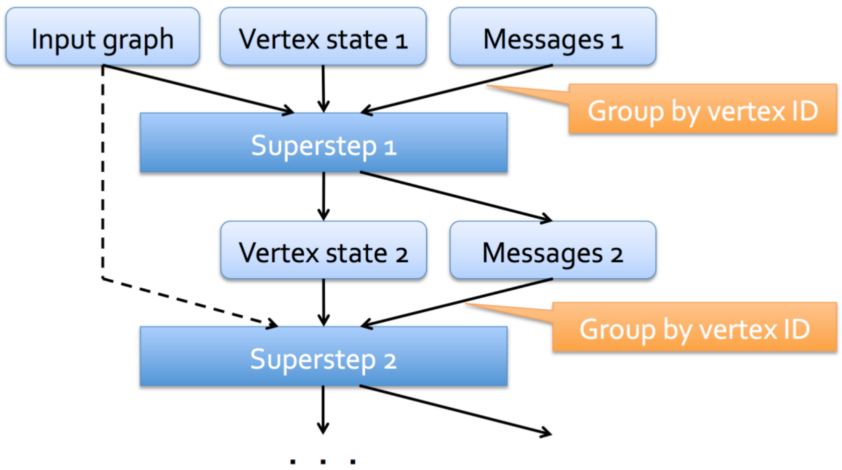
Εβδομάδα	Θεωρία (2 ώρες + 1 ώρα φροντιστήριο όπου χρειάζεται) – Δια ζώσης
1	Introduction
Construct	
2	Distributed Computing: Maximal Independent Set
3	Self-Stabilization
4	Population Protocols
5	Decentralized Computing: Blockchain
6-7	Decentralized Consensus
Understand	
8	Discrete Time Averaging Systems
9	Opinion Dynamics (De Groot – HK models)
10	Lotka-Volterra Population Dynamics
11	Kuramoto Coupled Oscillators
Explain	
12-13	Cellular Automata

Εργαστήριο – NETLOGO

Εβδομάδα	Εργαστήριο – Εξ αποστάσεως – ατομικές/ομαδικές ασκήσεις
1	Intro to NETLOGO (1)
2	Intro to NETLOGO (2)
3	What is Agent based Modeling? A Simple Model
4	Creating Agent-Based Models. Predator-Prey Model in NETLOGO.
5	The components of ABM: Agents, Environments, Interactions, Observers
6	Analysis of ABM. Virus Propagation (network or spatial). Behaviorspace.
7	Verification, Validation and Replication – Voter Models. Flocking Models.

Εργασίες για Φοιτητές	
Theoretical Exercises	Maximal Independent Set (MIS) in NETLOGO
	Self-Stabilization Algorithm in NETLOGO
	Extensions to existing ABM Models in NETLOGO
	Opinion Dynamics in NETLOGO
	Predator-Prey Models in NETLOGO
	Various Crazy Exercises…

Full Project



For this year (and the previous):

Pregel (think-like-a-vertex paradigm): a data flow paradigm and system for large-scale graph processing

Pregel is essentially a message-passing interface constrained to the edges of a graph. The idea is to ”think like a vertex” - algorithms within the Pregel framework are algorithms in which the computation of state for a given node depends only on the states of its neighbors.

1η Χρονιά: Αξιολόγηση από Φοιτητές (21-22)

Σχόλιο 1: Σημειώσεις – δεν είναι καλό που υπάρχει τόσο πολύ διαφορετική ύλη και πάρα πολλές βιβλιογραφικές πηγές. Μάλιστα ένας φοιτητής δήλωσε ότι το μάθημα ήταν περισσότερο συζήτησης και δεν είχε δομή.

Απάντηση: Το μάθημα είναι μεταπτυχιακού επιπέδου. Δεν έγινε σαφές αυτό από την αρχή. Θα γίνει όμως αυτή τη μέρα.

Σχόλιο 2: Οπωσδήποτε NETLOGO αναλυτικά

Απάντηση: Θα γίνει προσπάθεια να αλλάξει η δομή του εργαστηρίου

Σχόλιο 3: Προσοχή στις ασκήσεις. Ήταν πολύ απαιτητικές.

Απάντηση: Θα αλλάξει η δομή των ασκήσεων

Σχόλιο 4: Χρήση άλλων βιβλιοθηκών για προγραμματισμό (π.χ., MESA).

Απάντηση: Χρειάζεται προετοιμασία και χρόνο που θα επενδυθεί σε θέματα ύλης φέτος. Η NETLOGO επίσης προσφέρει ένα φιλικό περιβάλλον για το μάθημα.

2η Χρονιά: Αξιολόγηση από Φοιτητές (22-23)

Σχόλιο 1: Η διεξαγωγή και η καταγραφή των φροντιστηρίων για την Netlogo είναι πολύ καλή και συνέβαλε σημαντικά στην εκμάθησή της

Σχόλιο 2: Δεν υπάρχει ξεκάθαρη ύλη και σύγκραμα το οποίο όμως είναι λογικό, εφόσον το μάθημα αφορά αρκετά νέες τεχνολογίες αλλά θεωρώ πώς υπάρχουν συγκράματα τα οποία θα μπορούσαν να προταθούν. Επίσης, δεδομένου το ότι το μάθημα δεν διδάσκεται αρκετά χρόνια δεν έχει συγκεκριμένη δομή ως προς την ύλη που διδάσκεται. Ωστόσο, γίνεται μεγάλη και ικανοποιητική προσπάθεια από τον διδάσκοντα και αξίζει η προσπάθεια από κάποιον που τον ενδιαφέρει το αντικείμενο.

Απάντηση: Δεν υπάρχουν σχετικά συγγράμματα από τον Έυδοξο. Δίνω αναλυτικότατη ύλη τόσο από ηλεκτρονικά βιβλία όσο και από δημοσιεύσεις. Η δομή του πράγματι μεταβάλλεται συνεχώς και είναι ρευστή λόγω της περιέργειας του διδάσκοντα.

Some General Remarks…

Data vs Models

[The End of Theory: The Data Deluge Makes the Scientific Method Obsolete (2008)]

"…There is now a better way. Petabytes allow us to say: "Correlation is enough." We can stop looking for models. We can analyze the data without hypotheses about what it might show. We can throw the numbers into the biggest computing clusters the world has ever seen and let statistical algorithms find patterns where science cannot…"

Data vs Models

To Build Truly Intelligent Machines, Teach Them Cause and Effect (2018)

"…But as Pearl sees it, the field of AI got mired in probabilistic associations. These days, headlines tout the latest breakthroughs in machine learning and neural networks. We read about computers that can master ancient games and drive cars. Pearl is underwhelmed. As he sees it, the state of the art in artificial intelligence today is merely a souped-up version of what machines could already do a generation ago: find hidden regularities in a large set of data. "All the impressive achievements of deep learning amount to just curve fitting," he said recently…"

Distributed vs Decentralized Simply

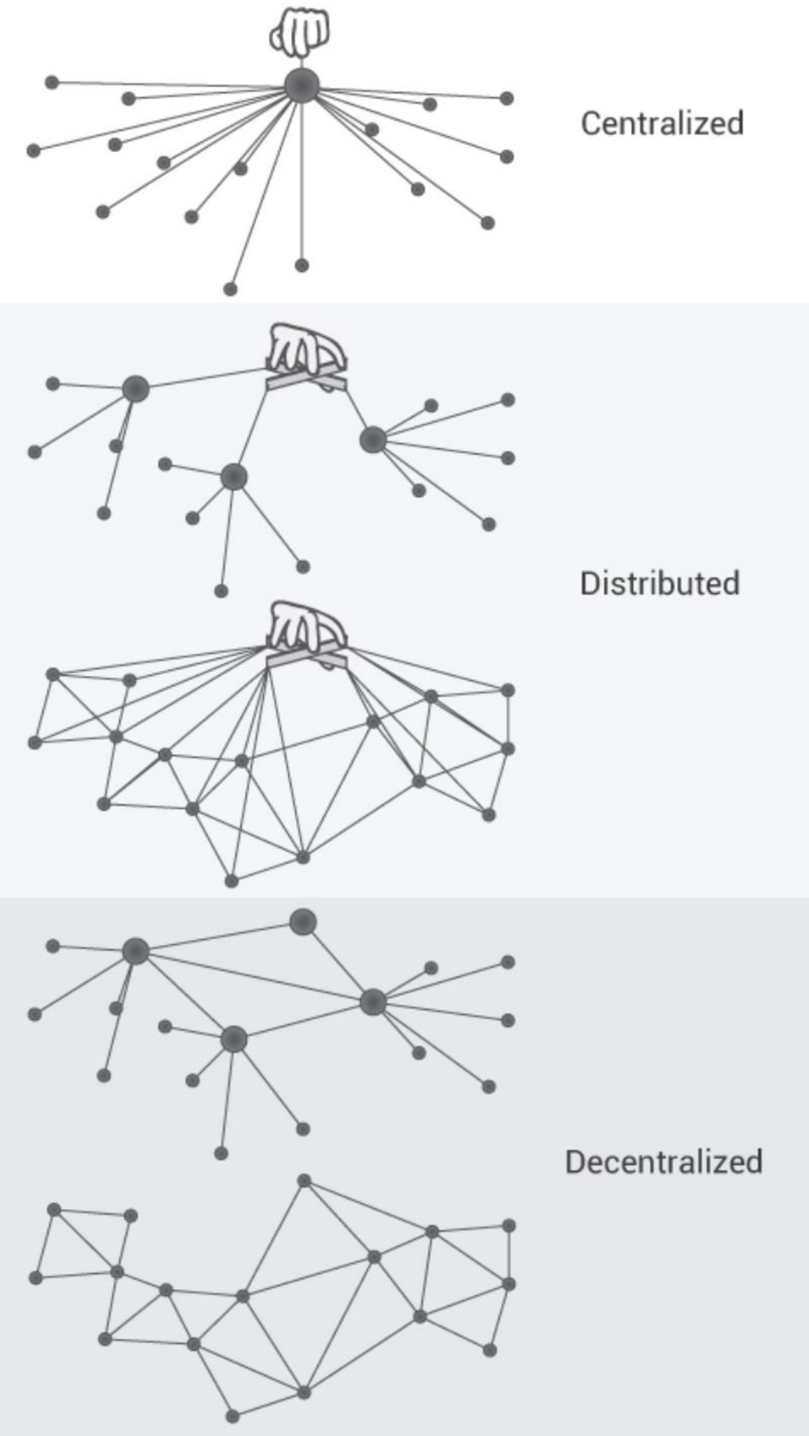
Location vs Control

- Using Word in your PC (centralized: Microsoft) (non-distributed: PC)
- Using Open Office in your PC (decentralized) (non-distributed:PC)
- Cloud service provider for storage (centralized: the provider) (distributed)
- Bitcoin (decentralized: cannot be altered by any entity) (distributed: runs as a P2P system)

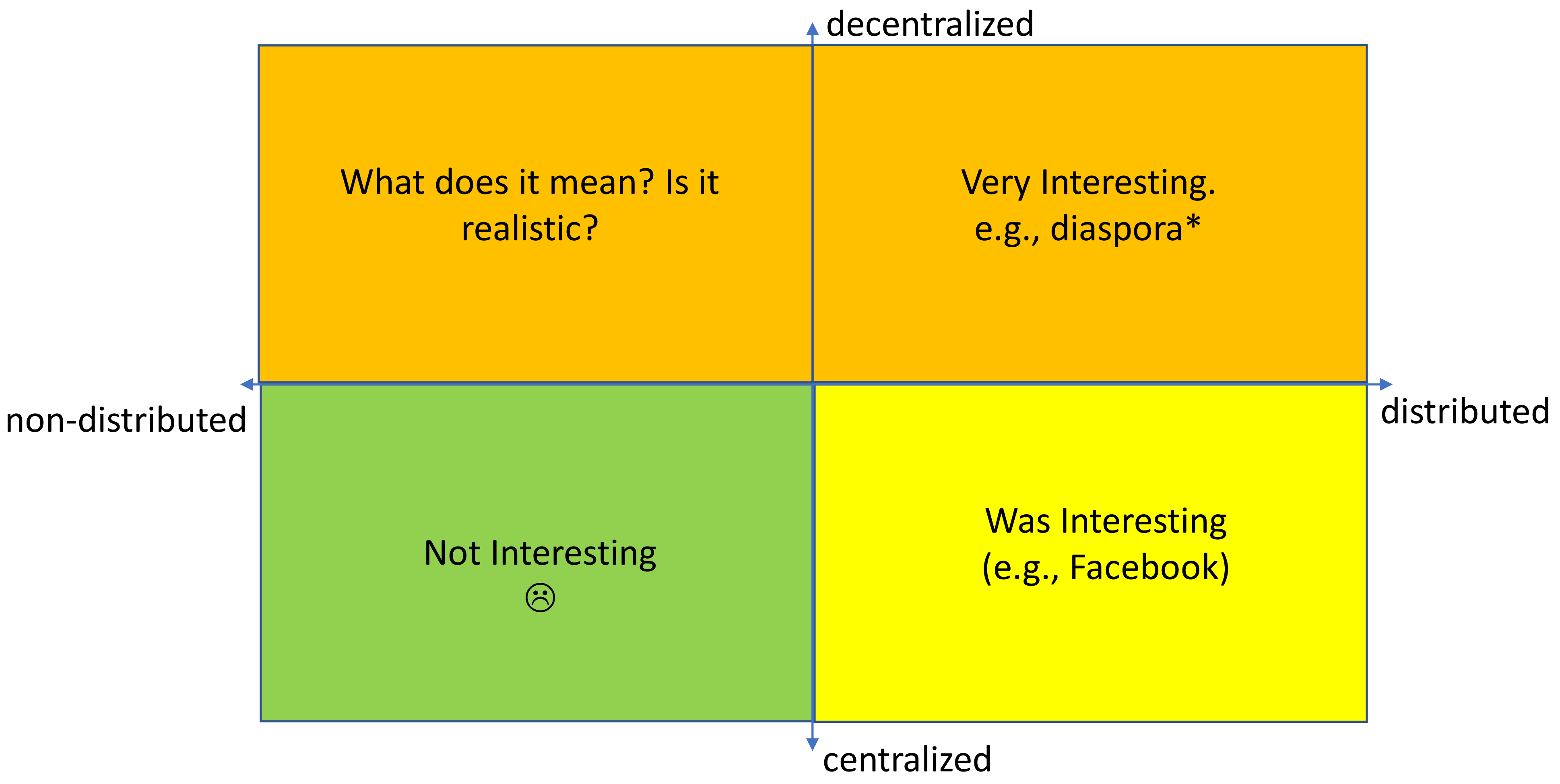
Distributed vs Decentralized

For example, consider the system of transportation of a city and many agents trying to go from some initial location to a destination. Let efficiency in this case mean the average time for an agent to reach the destination. In the "centralized" solution, a central authority can tell each agent which path to take in order to minimize the average travel time. In the "decentralized" version, each agent chooses its own path.

The Price of Anarchy measures the ratio between average travel time in the two cases.
But this is another relative course.



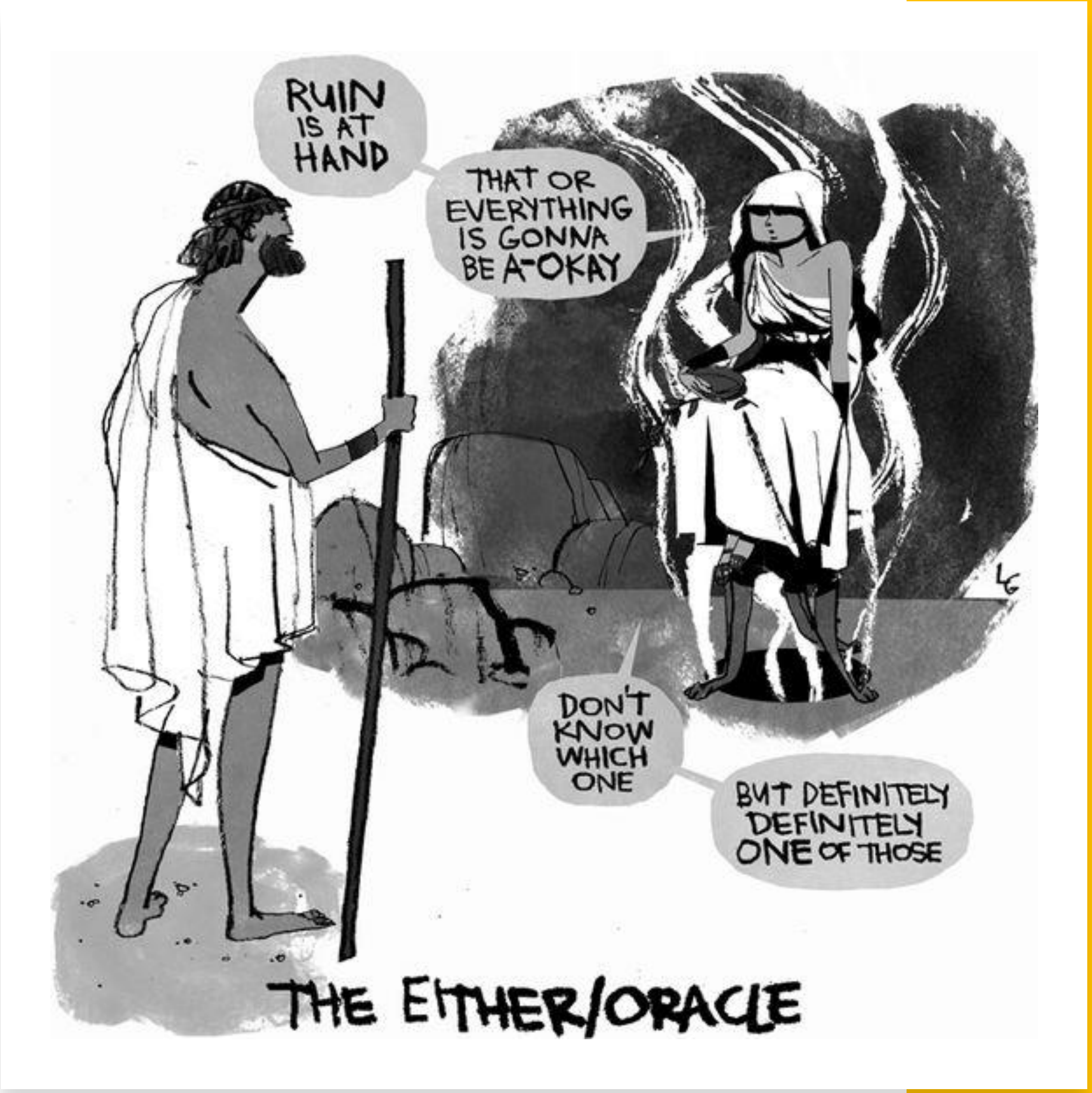
Designing a Decentralized Social Network



Complicated vs Complex Systems

Looking at the Future

I mean the future lectures …

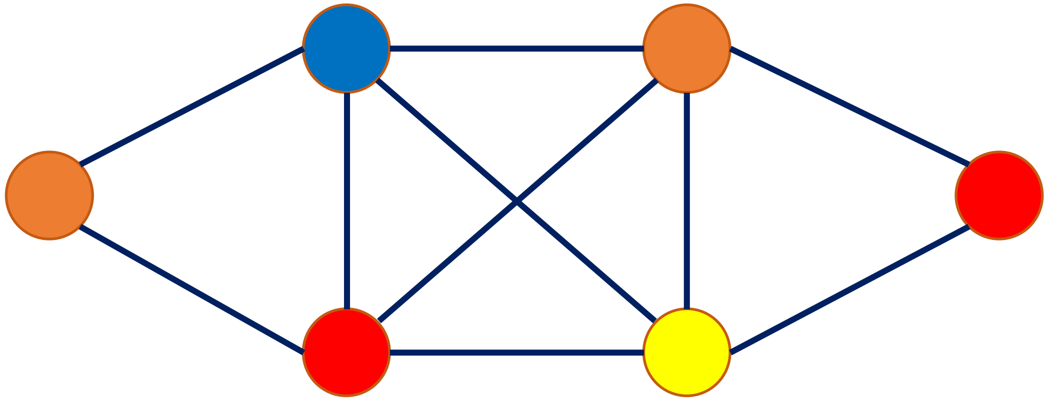


Construct

Graph Coloring…

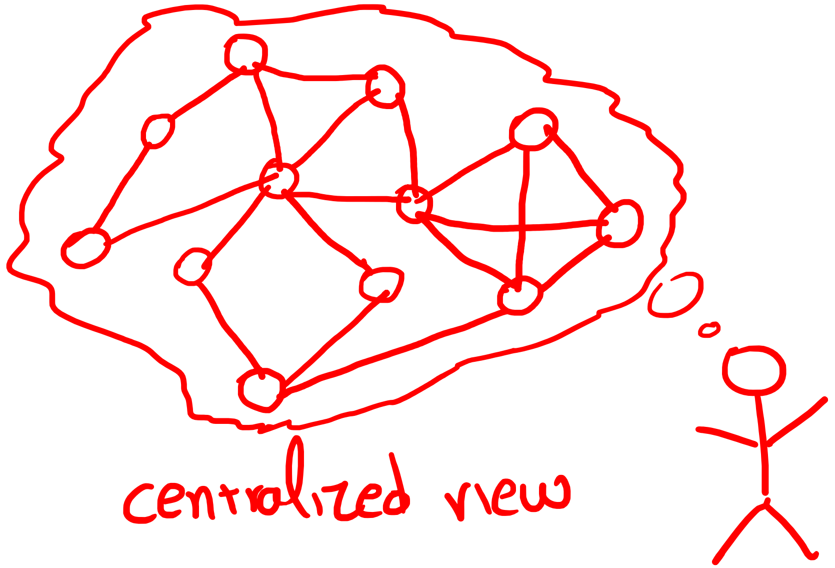
Coloring Graphs

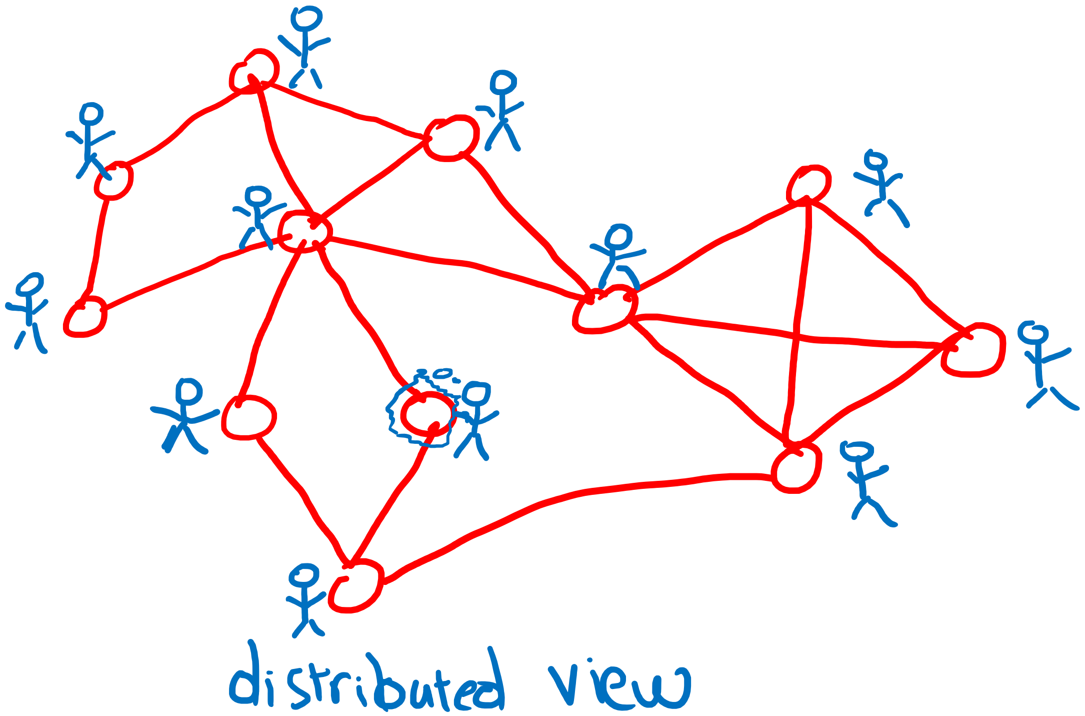
Definition: A graph has been colored if a color has been assigned to each vertex in such a way that adjacent vertices have different colors.



Definition: The chromatic number of a graph is the smallest number of colors with which it can be colored.

In the example above, the chromatic number is 4.





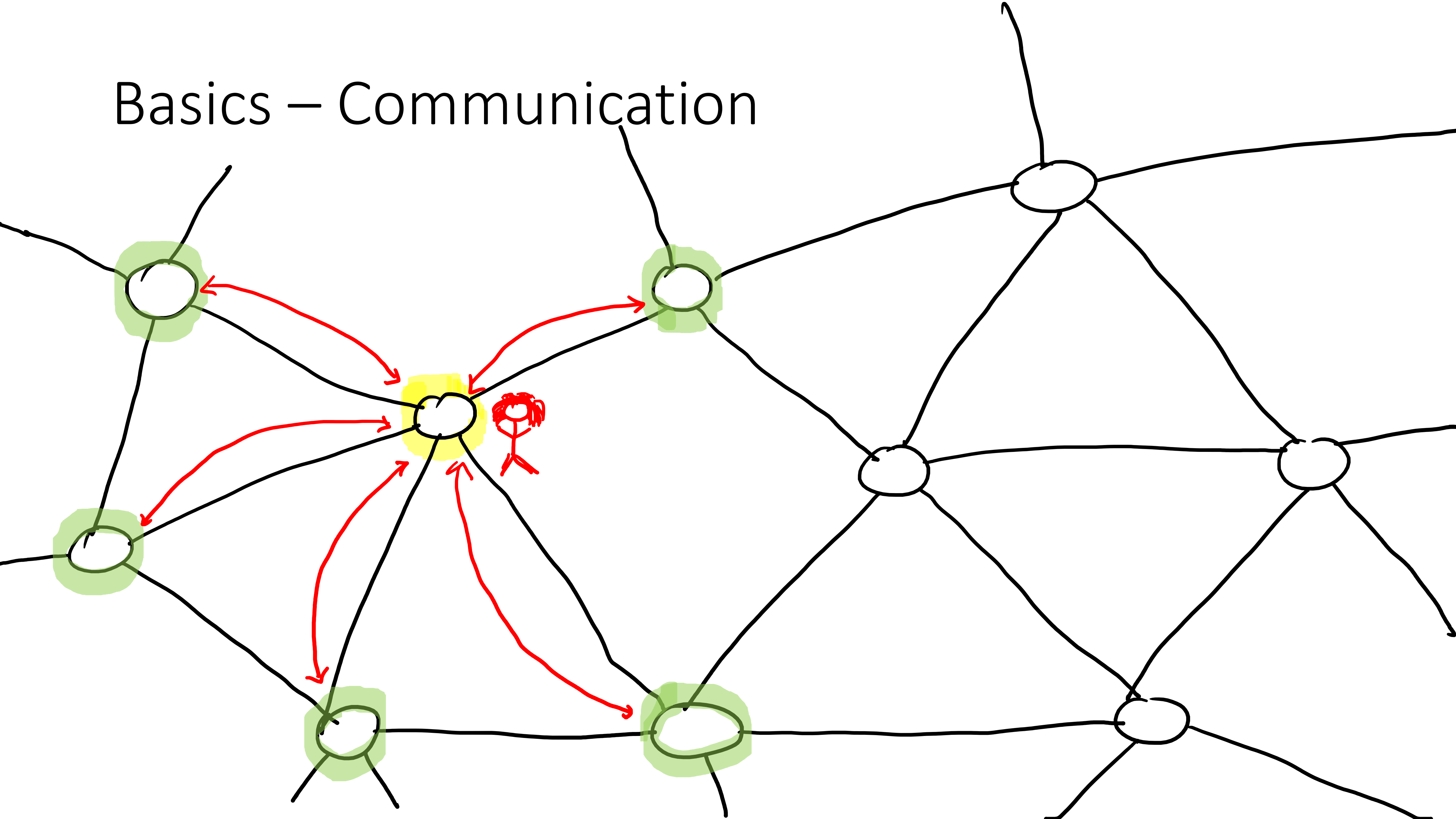
Models

PN model, LOCAL and CONGEST

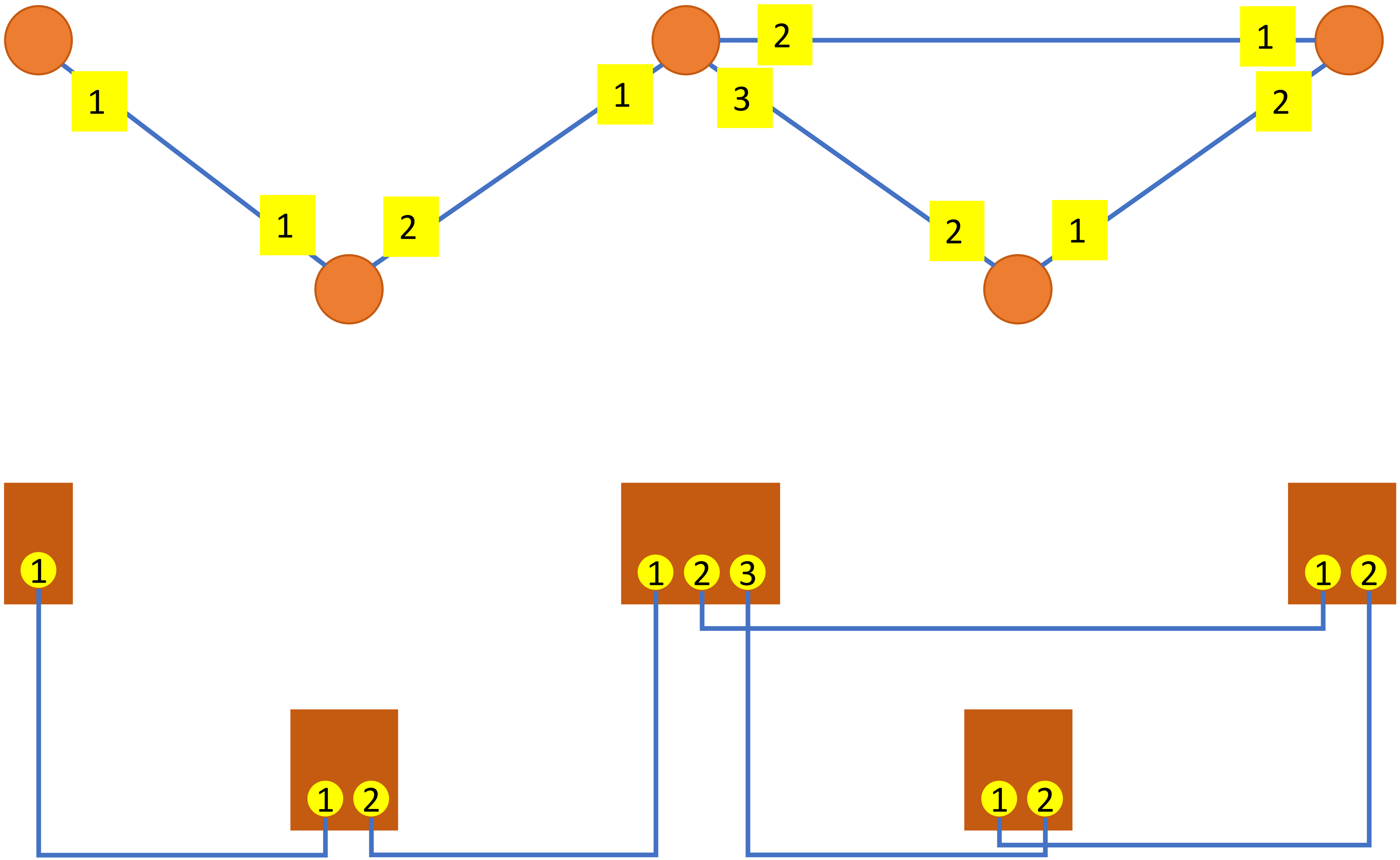
Αν θέλετε πιο τυπικά και σε μεγαλύτερο βάθος θα πρέπει να πάρετε το μάθημα «Κατανεμημένα Συστήματα» – το προτείνω ανεπιφύλακτα…

Εδώ θα μιλήσουμε μόνο στο επίπεδο εκείνο που μας χρειάζεται για να κατανοήσουμε ό,τι ακολουθεί…

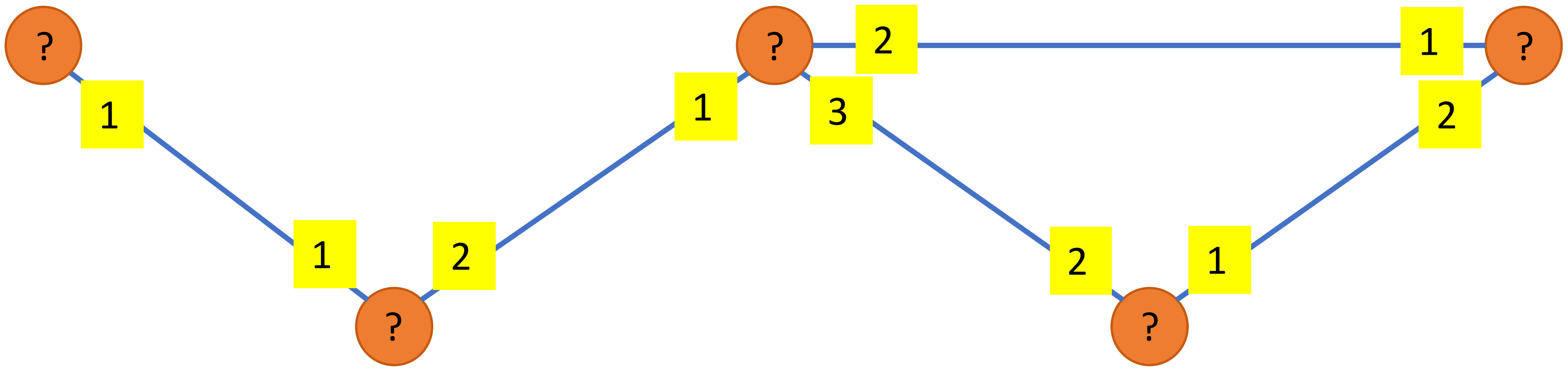
Basics – Communication



Port-Numbering Model (PN Model)



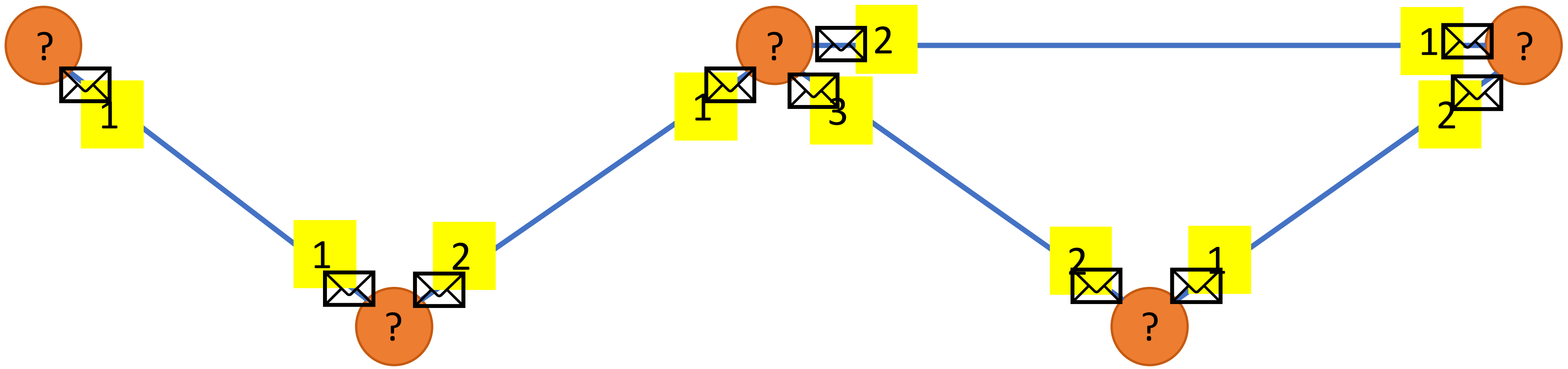
Distributed Algorithms



Assumptions

1. All nodes are identical and run the same algorithm/protocol (heterogenous???)
2. The node knows its own degree (radio networks???)
3. No unique identifiers of nodes (???)
4. Each node has a local state (if not additional local knowledge then initial state based only on degree)

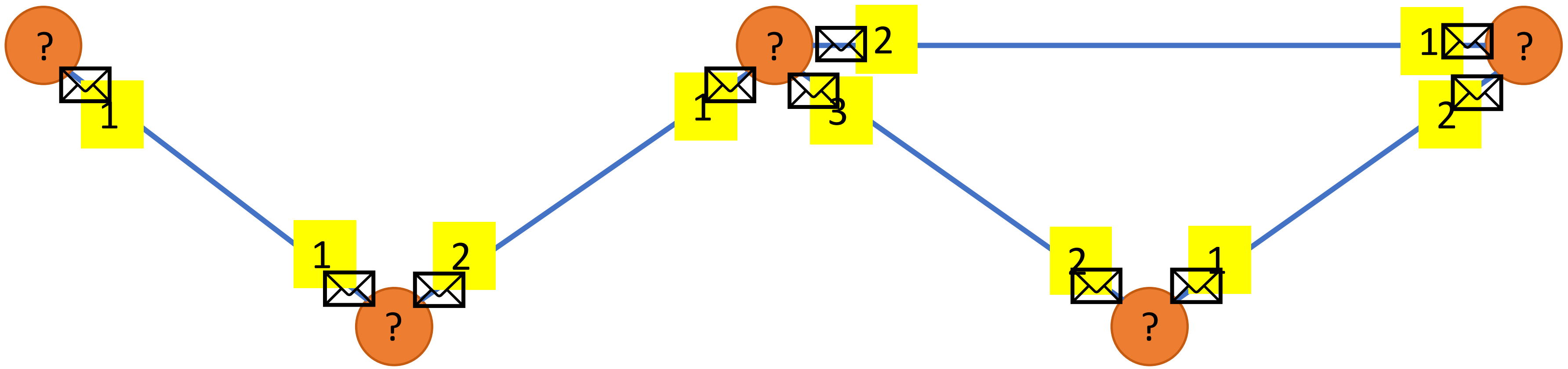
Distributed Computation



Computation in synchronous rounds

1. Construct outgoing messages in parallel

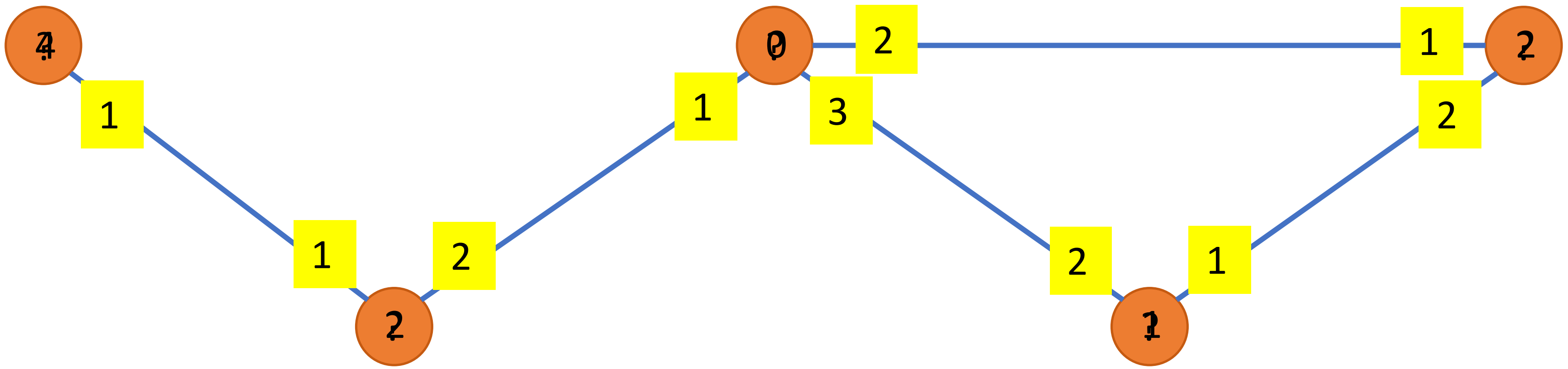
Distributed Computation



Computation in synchronous rounds

1. Construct outgoing messages in parallel
2. Send messages in parallel
3. Update state based on incoming messages and previous state in parallel
4. Repeat forever

Termination of Computation



Some of the states are stopping states

- A node in a stopping state will not change its state again

When all nodes have reached a stopping state:

- The set of all stopping states will be the **output** of the algorithm

Complexity Measures

Time = Number of Rounds

Message Complexity = How Many Messages (number or in bits)?

Internal computation is usually considered to be free since it is faster than communication???

The Distributed Algorithm

Choose initial states of nodes

Construct outgoing messages

Update state based on incoming messages

Init

Send

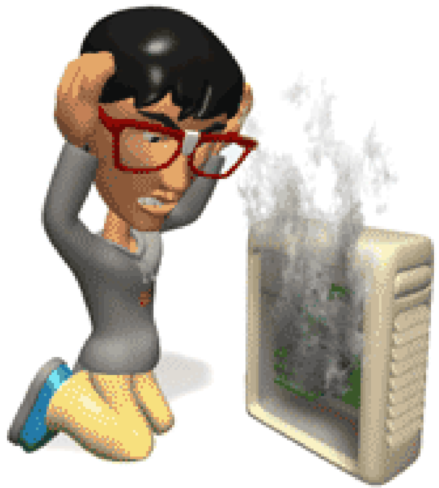
Receive

Distributed Algorithm

Challenges

Locality (symmetry breaking)

1. IDs
2. Randomization



Congestion (bandwidth limitation)

1. Small messages in each round
2. Usually $O(\log n)$ bits

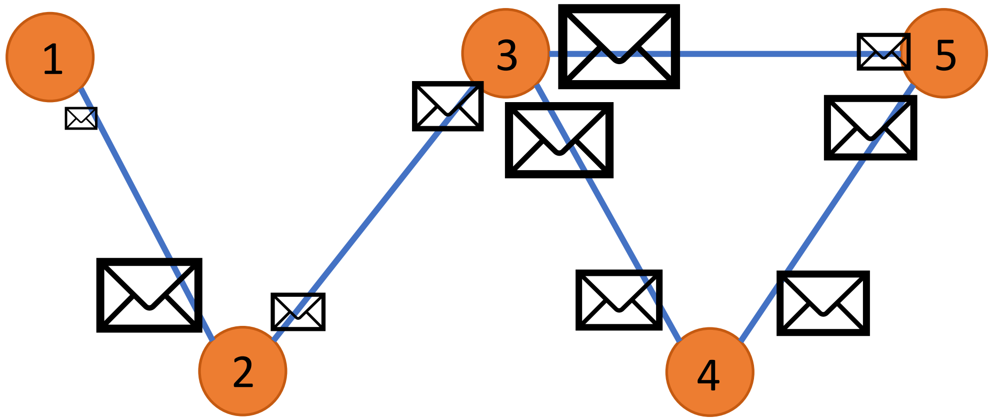
The LOCAL Model (Linial, FOCS '87)

PN model + n nodes with unique identifiers from $\{1,2,\ldots,poly(n)\}$

Initially each node knows:

- Its ID and IDs of its neighbors
- Estimate on global parameters:

e.g., number of nodes, max-degree, etc.

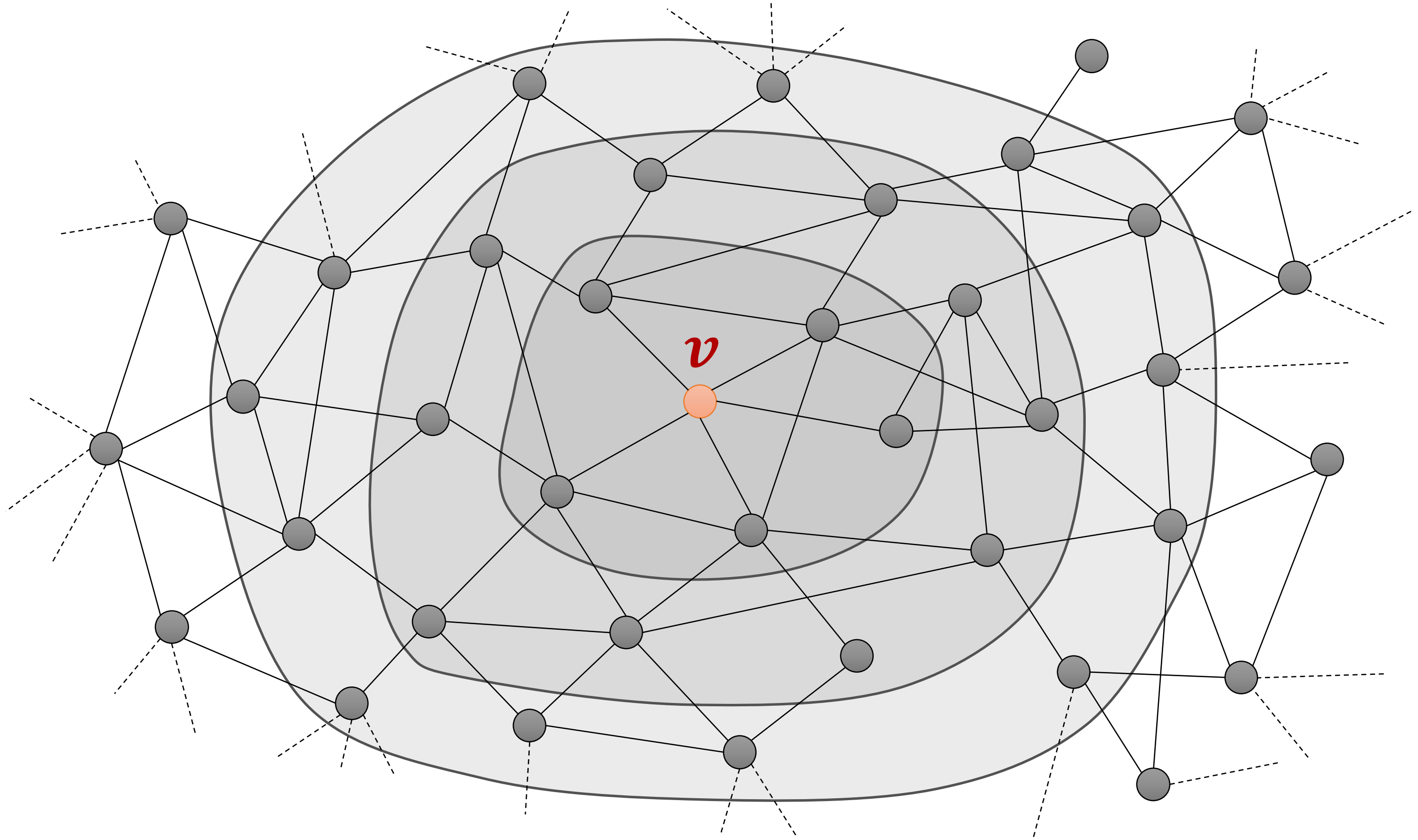


Synchronous rounds:

1. Each node/computer does some internal computation
2. Send a message to each neighbor (possibly unbounded)
3. Receive message from each neighbor

Unbounded internal computation & message size

The LOCAL Model



Trivial upper bound:
$O(Diam(G))$ rounds

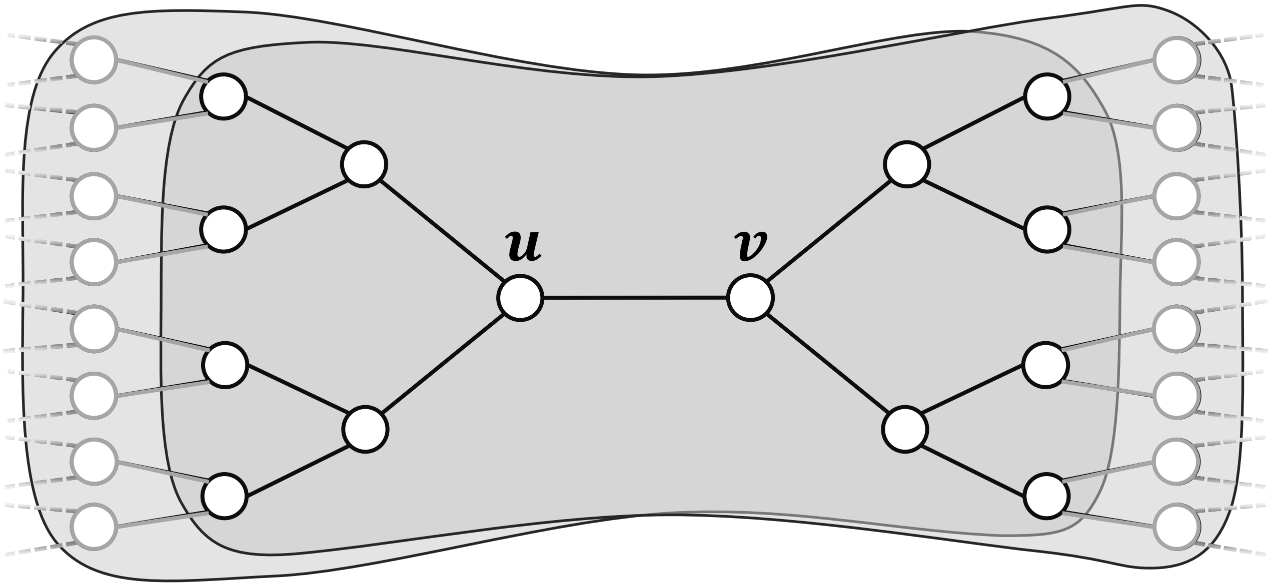
r-Round Algorithm:

- Each node computes its output as a function of the initial state of its r-neighborhood

Challenges in the LOCAL Model

Slide by Fabian Kuhn

Symmetry Breaking / Local Coordination



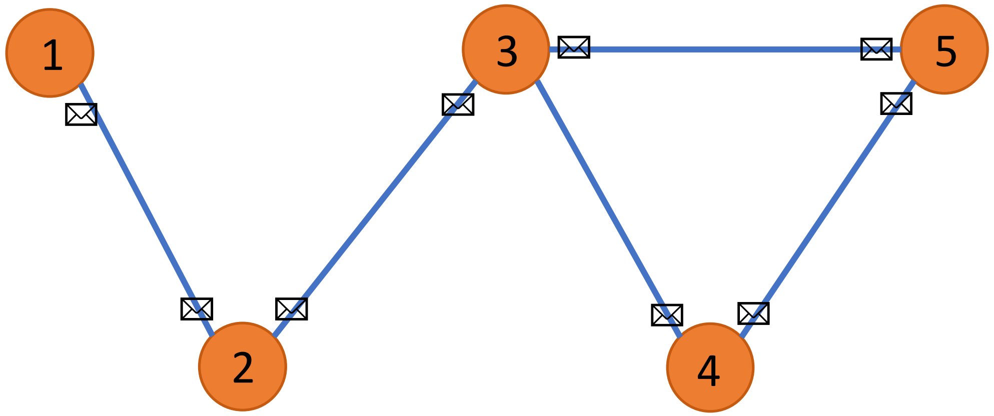
- Neighboring / nearby nodes need to output different values
  - e.g., different colors, at most one can be in an MIS, etc.
- Nodes need to decide in parallel

Key Challenge: locally coordinate among nearby nodes

- randomization naturally helps (e.g., choose color at random)

The CONGEST Model (Peleg '90)

- n nodes with unique identifiers (LOCAL)



Initially each node knows:

- Its ID and IDs of its neighbors
- Estimate on global parameters:

e.g., number of nodes, max-degree, etc.

Synchronous rounds:

1. Each node/computer does some internal computation
2. Send a message to each neighbor
3. Receive message from each neighbor

} $O(\log n)$ bits

Unbounded internal computation & message size $O(\log n)$

Deterministic vs Randomized Model

Init : The initial state is a random variable

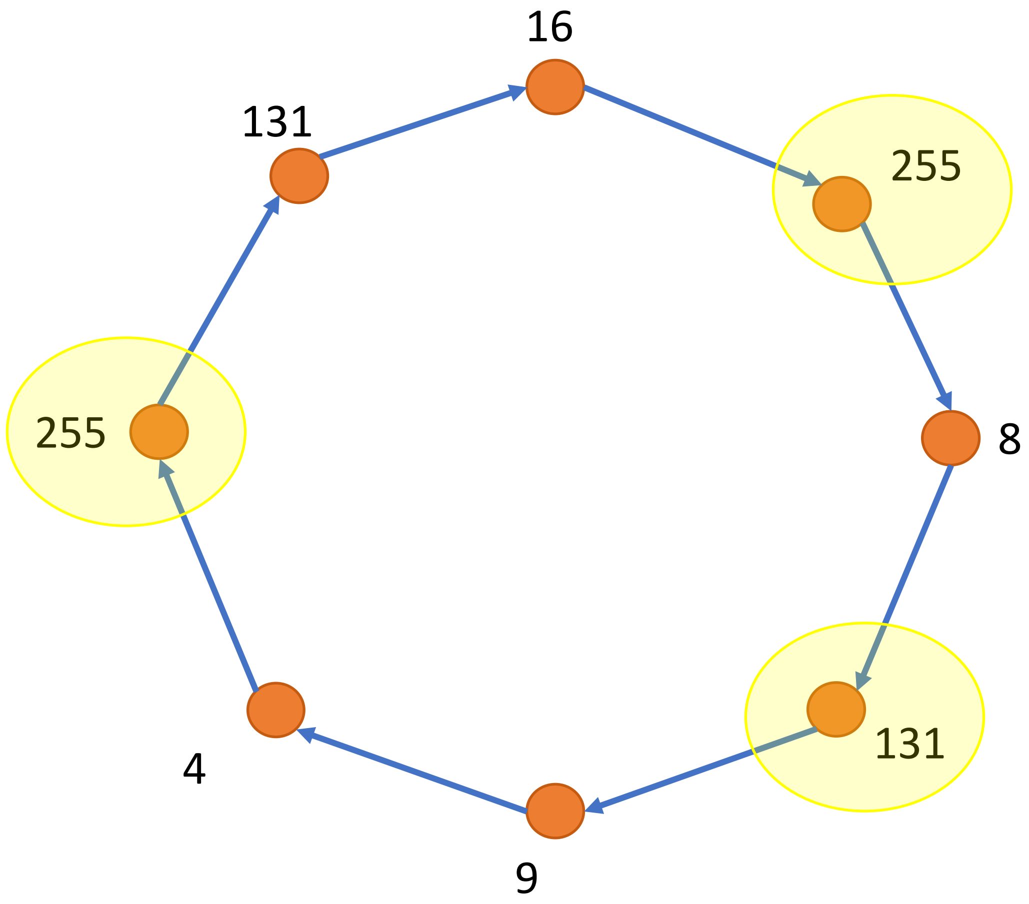
Receive : The state at time t is a random variable

Send : Deterministic

Greedy Color Reduction

LOCAL Model

Let's Look At Oriented Cycles First

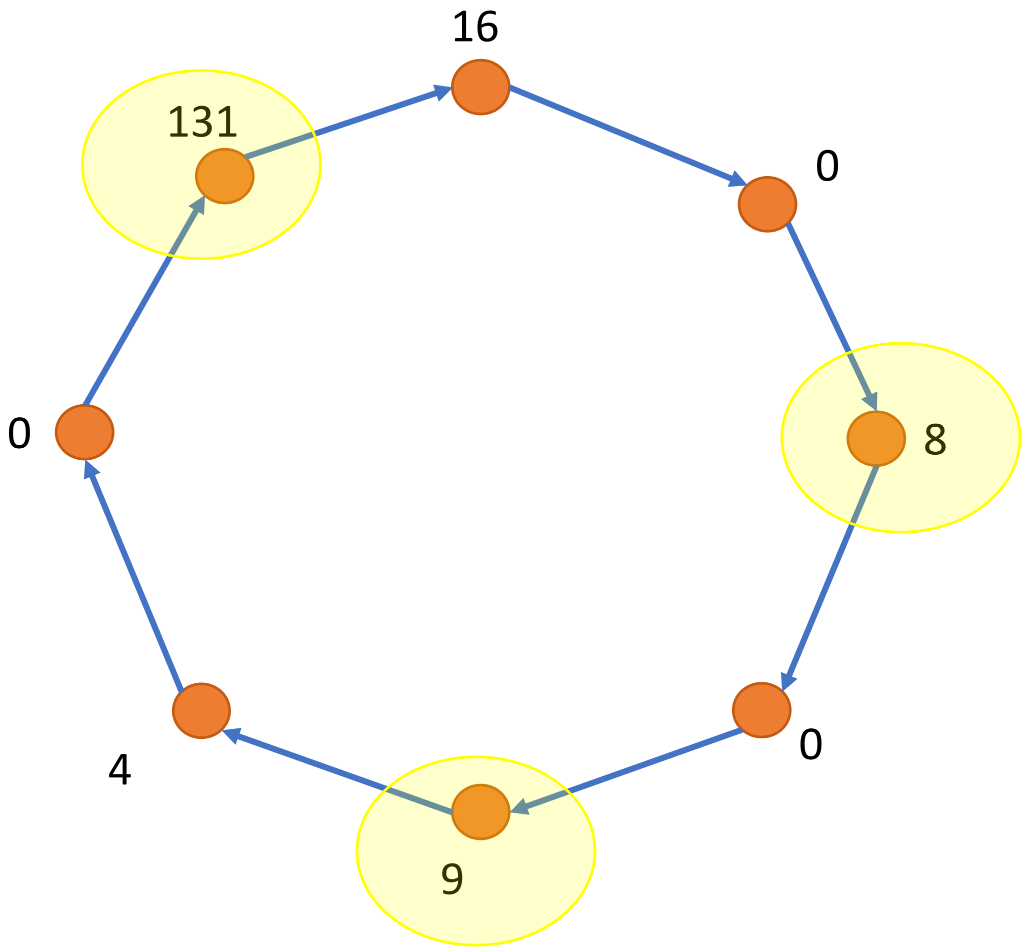


Assume 256 colors (0-255)

Nodes that have the largest color among their neighbors can pick a color from the set $\{0,1,2\}$.

Iterate.

Let’s Look At Oriented Cycles First

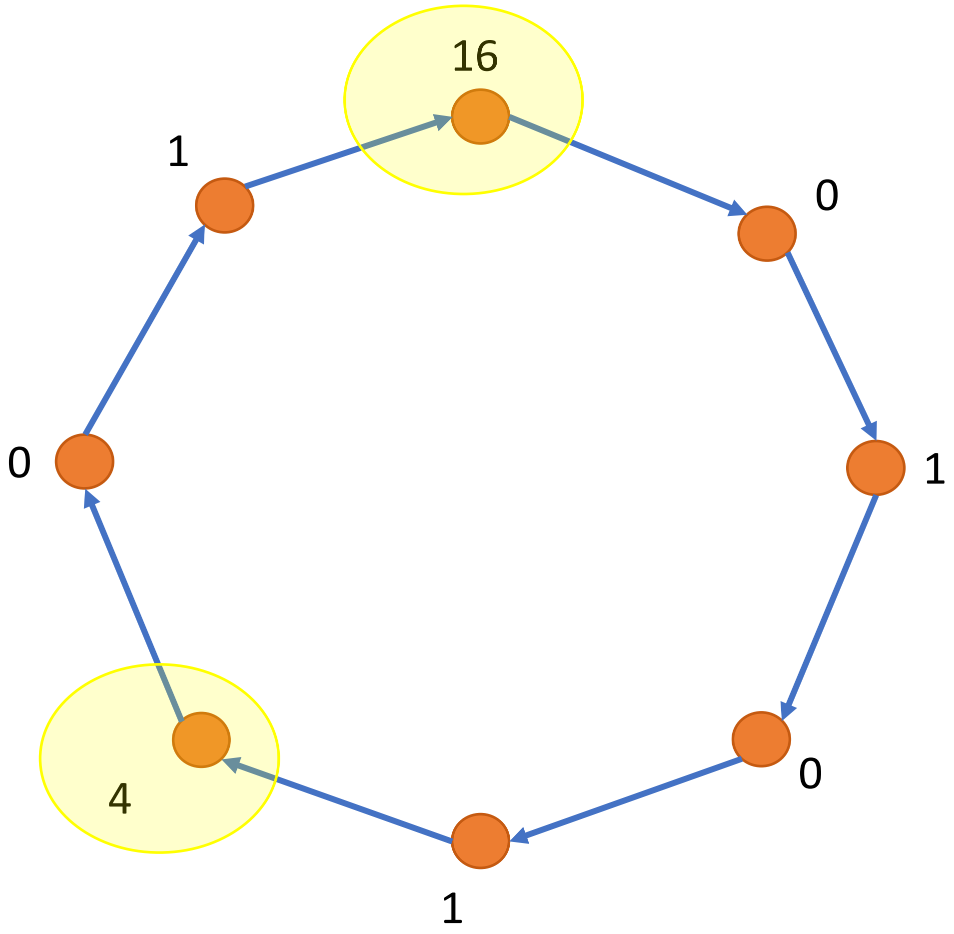


Assume 256 colors (0-255)

Nodes that have the largest color among their neighbors can pick a color from the set $\{0,1,2\}$.

Iterate.

Let’s Look At Oriented Cycles First

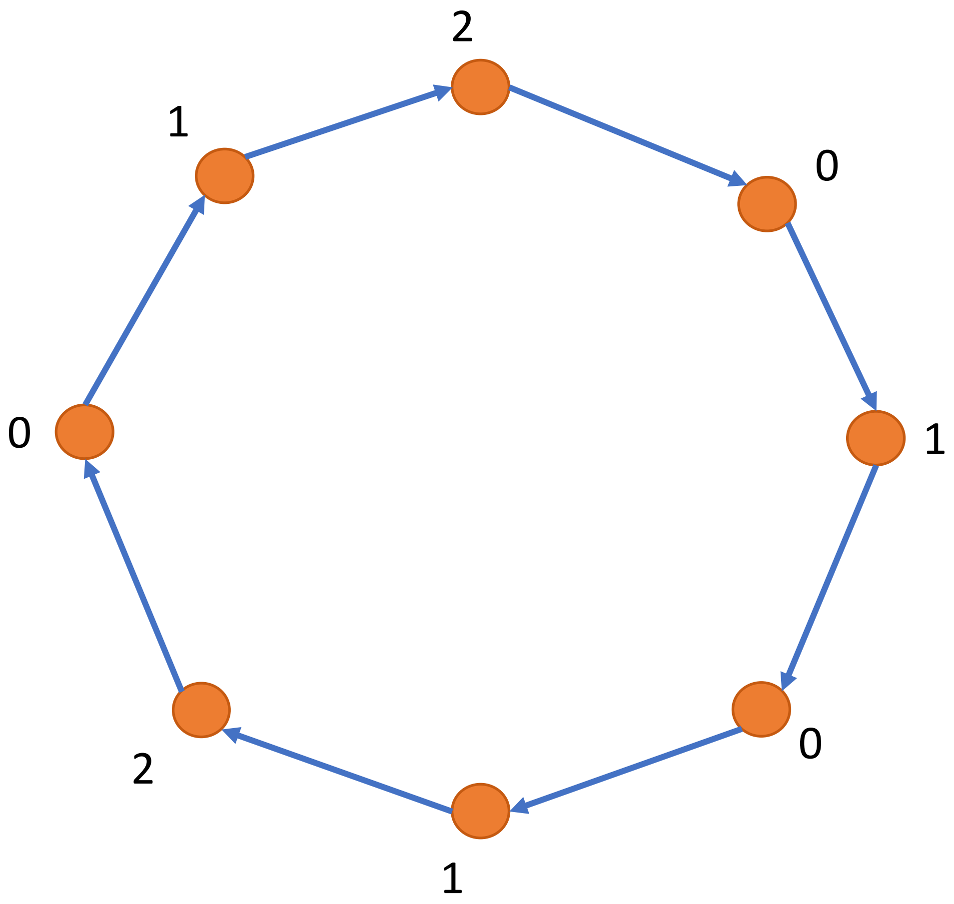


Assume 256 colors (0-255)

Nodes that have the largest color among their neighbors can pick a color from the set $\{0,1,2\}$.

Iterate.

Let’s Look At Oriented Cycles First



Assume 256 colors (0-255)

Nodes that have the largest color among their neighbors can pick a color from the set $\{0,1,2\}$.

Iterate.

Complexity?

Greedy Color Reduction

Assume we start with a graph $G = (V, E)$ with x colors

We will reduce them to

$$y = \max\{x - 1, \Delta + 1\}$$

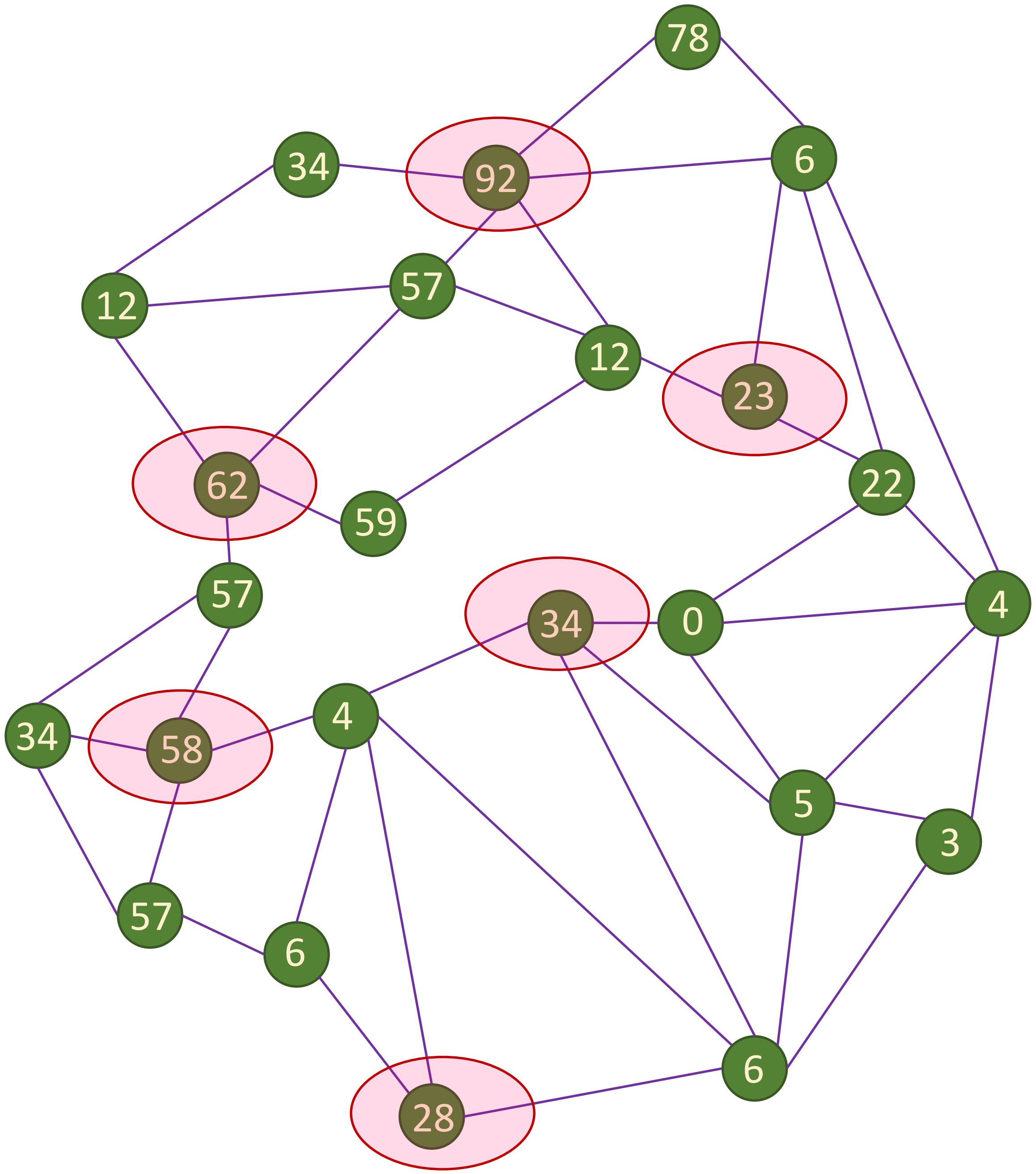
Δ is the maximum degree of the graph

Similar Idea to the Cycle

Repeat forever:

1. Send message c to all neighbors //c is the current color of the node
2. Receive messages from all neighbors. Let M be the set of all messages
3. If $c \notin \{0,1,\dots,\Delta\}$ and $c > \max\{M\}$ then
   $c \leftarrow \min\{\{0,1,\dots,\Delta\} - M\}$

A node that has its color changed is a local maxima w.r.t. color and it is active, otherwise passive.



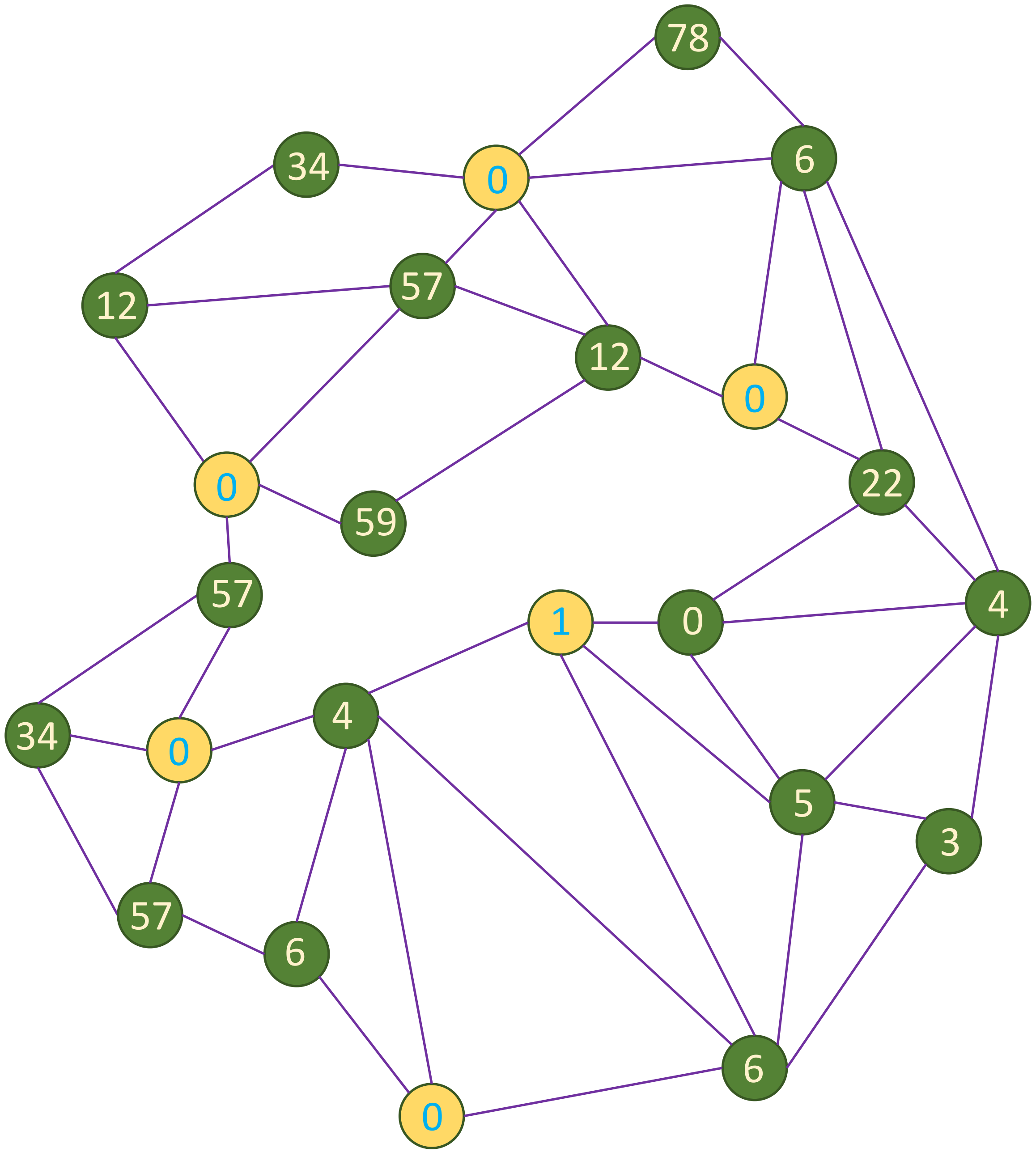
Similar Idea to the Cycle

Repeat forever:

1. Send message c to all neighbors //c is the current color of the node
2. Receive messages from all neighbors. Let M be the set of all messages
3. If $c \notin \{0,1,\ldots,\Delta\}$ and $c > \max\{M\}$ then

$$c \leftarrow \min\{\{0,1,\ldots,\Delta\} - \mathrm{M}\}$$

A node that has its color changed is a local maxima w.r.t. color and it is active, otherwise passive.



Analysis

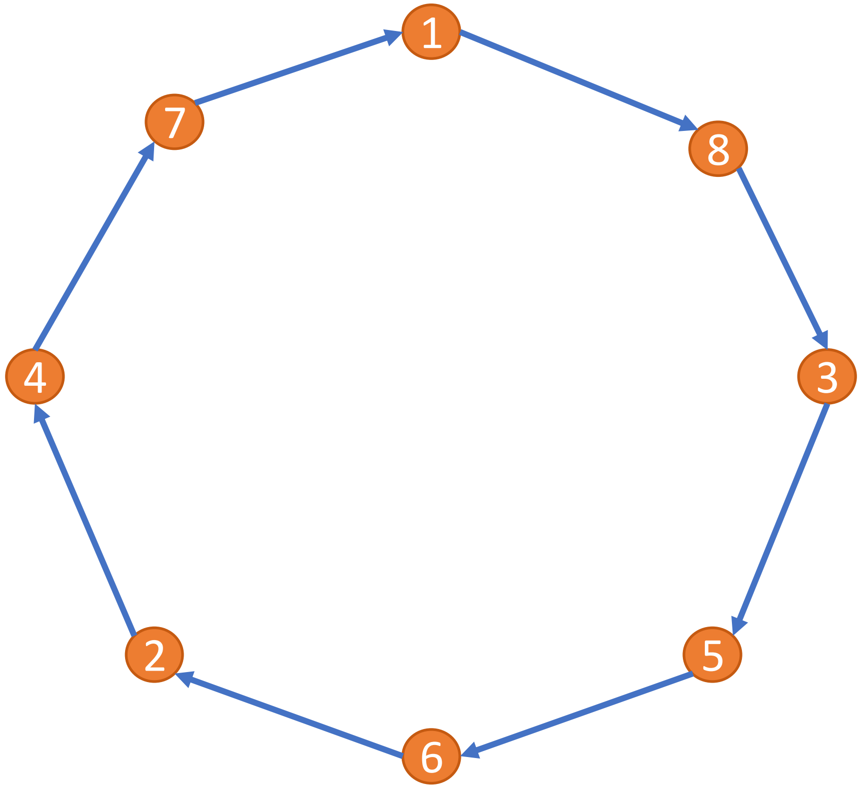
- The active nodes form an independent set
- No need to know the number of colors x or the maximum degree Δ.
- Repetitively apply greedy color reduction. This ends when no active node exists – the algorithm must change accordingly to take into account stopping conditions
- Time complexity: $O(x - \Delta)$

Linial's Lower Bound for Deterministic Coloring

Linial's Lower Bound Made Easy, J. Laurinharju, J. Suomela PODC 2014

Directed n-cycles

- Deterministic algorithms
- LOCAL model
- Nodes with unique IDs in $\{1,2,\ldots,n\}$
- Colors from set $\{1,2,3\}$



- Orientation only for additional help to the algorithm. Messages can be sent both ways.

<u>Linial's Lower Bound:</u> A deterministic distributed coloring algorithms requires at least $\frac{\log^* n}{2} - 1$ communication rounds for a directed n-cycle.

A Randomized Algorithm

LOCAL model

Very Simple and very efficient…

Deterministic $O(\Delta + \log^* n)$ vs Randomized $O(\log n)$

Guarantees

Monte Carlo:

- Guaranteed running time
- Probabilistic output quality

Las Vegas:

- Probabilistic running time
- Guaranteed output quality

Success with high probability (w.h.p.)

Simple Idea

1. Every node tries to pick a random free color

2. Stop if successful

Very simple but the analysis is not so simple…

Still Simple…

1. Nodes are active with probability $\frac{1}{2}$
2. Every active node tries to pick a random free color
3. Stop if successful

Correctness is self-evident.

Why does it stop fast with high probability?

Analysis

Lemma: A node that is still running, will stop in this round with probability ≥ 0.25.

Corollary: The node is still running after T rounds with probability $\leq 0.75^T$.

Corollary: All nodes will stop after $O(\log n)$ rounds w.h.p..

References

1. J. Laurinharju and J. Suomela. Linial's Lower Bound Made Easy. In PODC '14.
2. J. Hirvonen and J. Suomela. Distributed Algorithms 2020.

We have already done it for the randomized algorithm...

Explain

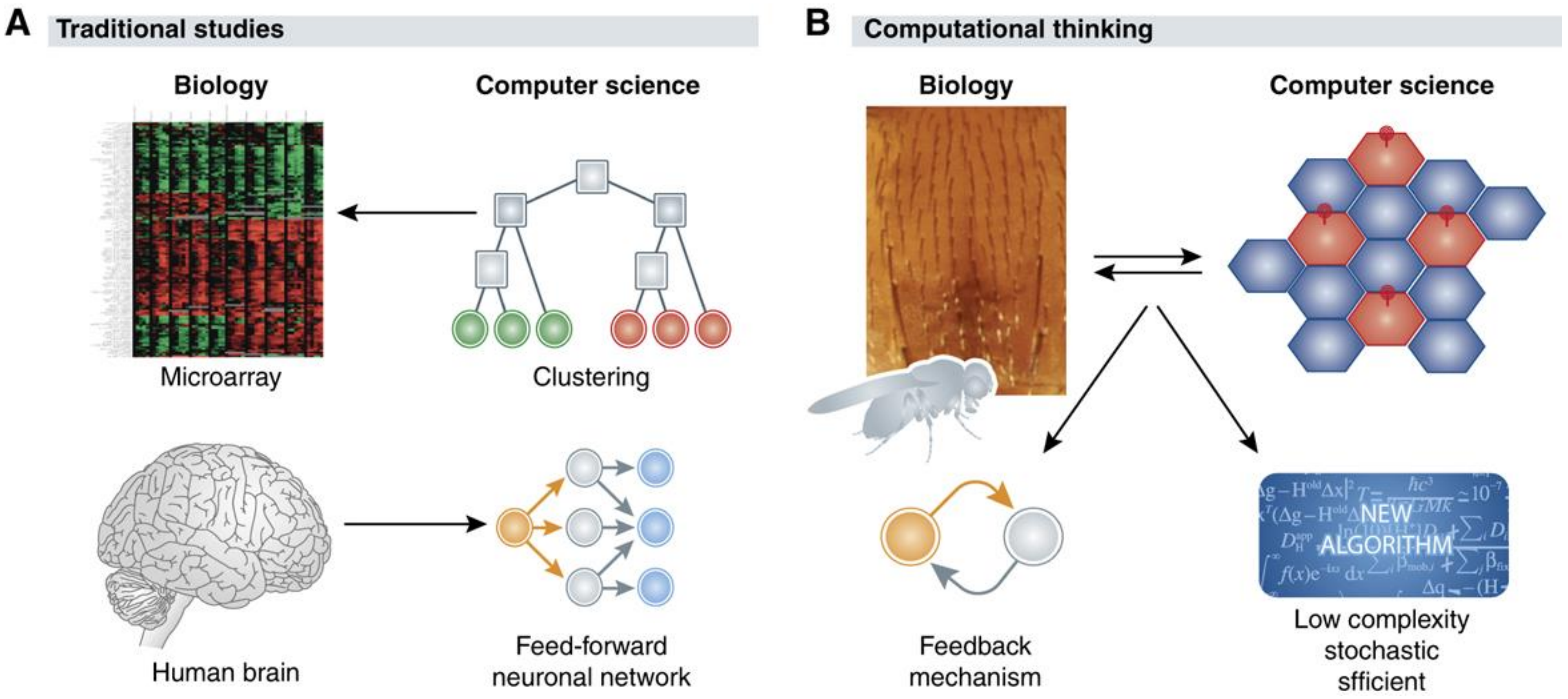


Image taken from http://www.algorithmsinnature.org/

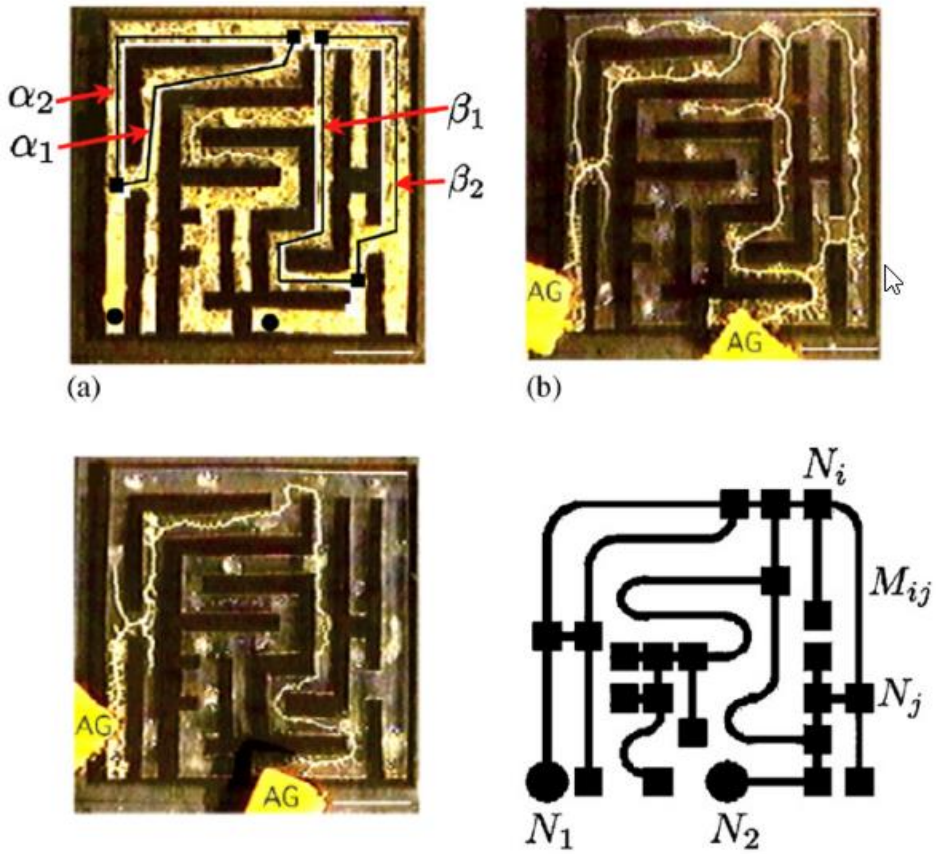
Nature → Algorithms

Nature ← Algorithms

An Example

Finding Shortest Paths (not only!) with Physarum Polycephalum

If a maze is covered with slime mold and there are two food sources on it, then the slime mold retracts to the shortest path between these two locations on the maze.



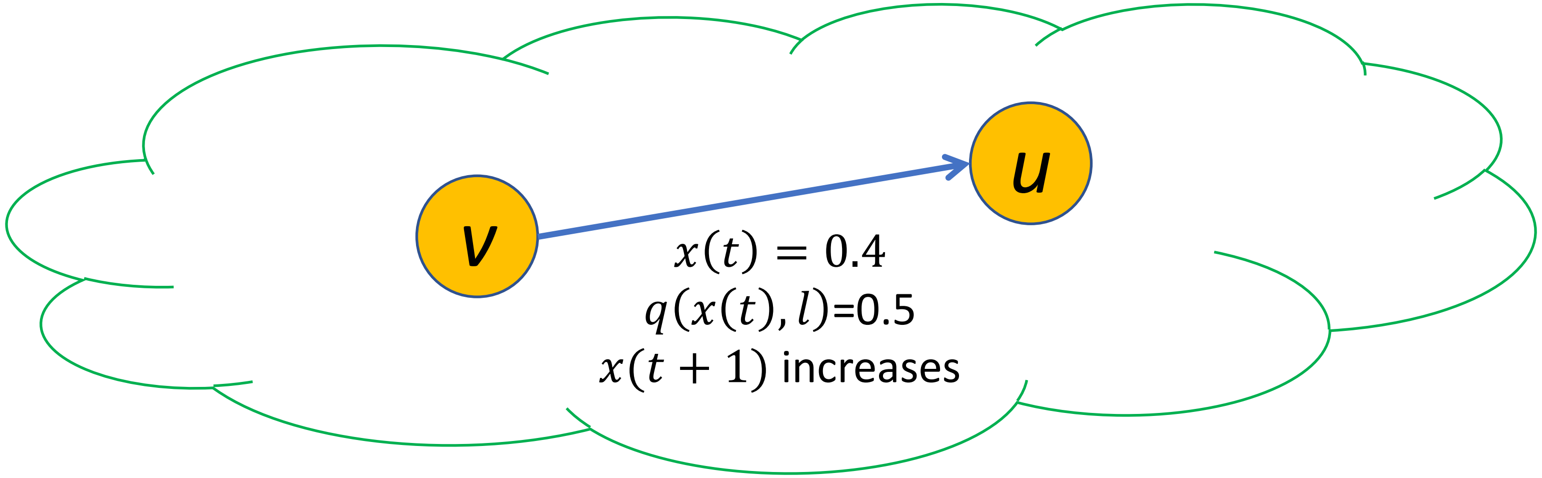
Picture taken from “A Mathematical Model for Adaptive Transport Network in Path Finding by True Slime Mold”.

A Descriptive Mathematical Model

Network of capacities $x_e(t)$ and lengths l_e for an edge e.

The current flow between the two food sources on an edge e is $q_e(x(t), l)$:

$$\frac{dx}{dt} = |q(x, l)| - x$$



How do you Compute $q(x, l)$?

Thomson's Principle (physics):

The unit value flow between two nodes s_0 and s_1 in a network, is distributed on its edges such that the total energy is minimized

$$\min \; q^T R q$$

such that $Aq = b$

The Proof of the Natural Algorithm

1. An equilibrium point of the dynamical system is a path
2. Find an appropriate upper bound *V* on the energy (max flow – min cut theorem)
3. Prove that V is non-negative and decreasing (Lyapunov function) – thus it converges to a path by (1)
4. Prove that it converges to the shortest path

Rule-based Physarum Dynamics?

Either:

- One has to somehow incorporate Thomson’s principle in the local rule
- This means that one has to look at how this principle emerges from local rules… (it has been done to some extent)

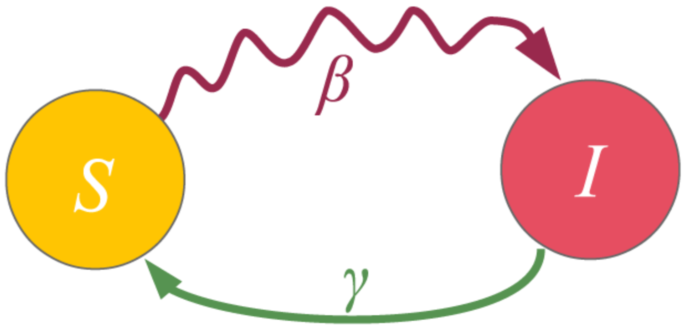
OR

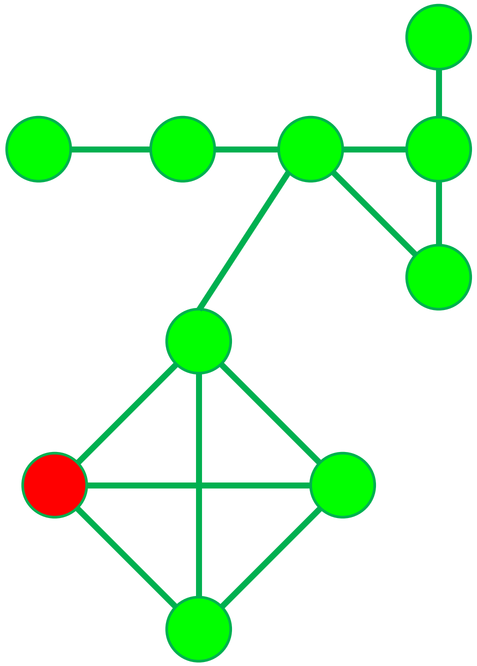
- Completely change the model

Virus Propagation on ***Static*** Networks (Dynamics on the Network)

Epidemiological Models

- Susceptible – Infected – Susceptible (SIS)

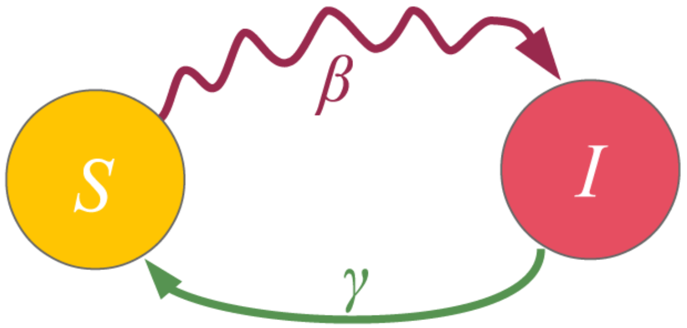




Virus Propagation on *Static* Networks (Dynamics on the Network)

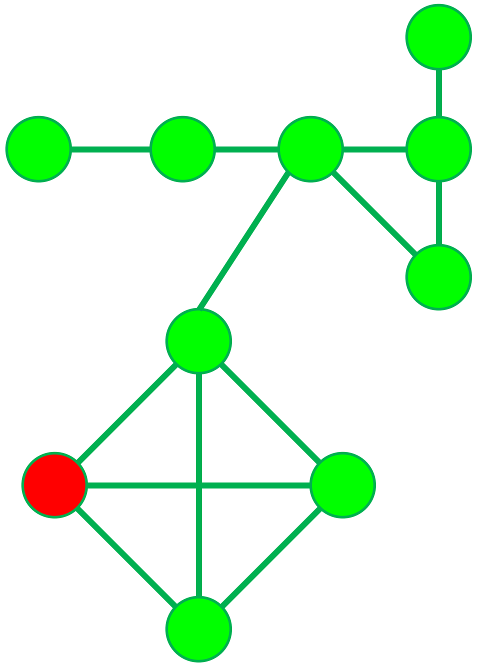
Epidemiological Models

- Susceptible – Infected – Susceptible (SIS)



Some Questions:

- When does the virus flood the network?
- What if you have multiple profiles?
- What if you have many interacting viruses?
- Other epidemic models?
- …



Virus Propagation on Dynamic Networks???

What are the questions to be asked?

- How does the network(s) change?
    - Assumptions?

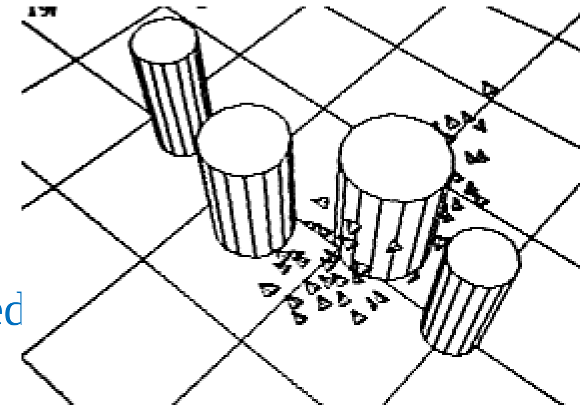
- When does a virus flood the network?
- Interacting Viruses?
- Profiles?
- Other epidemic models?
- …

Unanswered or partially answered

Understand

Craig Reynolds and “Boids”

Craig Reynolds is a computer graphics researcher, who revolutionised animation in games and movies with his classic paper :

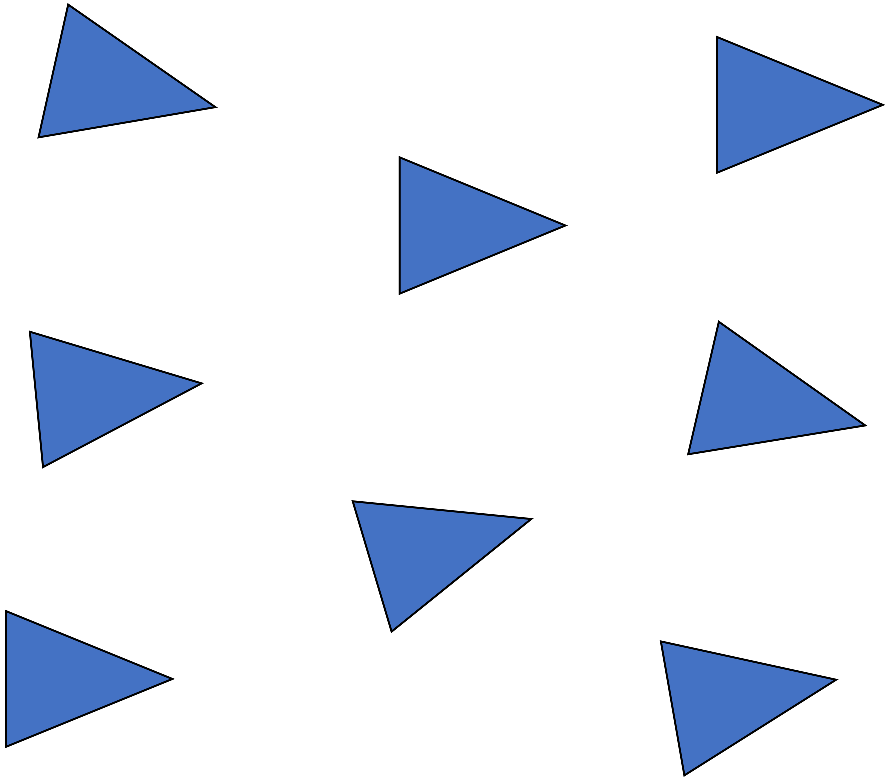


Reynolds, C. W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model, in *Computer Graphics*, **21**(4):25-34.

The story is:

- before this paper, animations of flocks, swarms, groups, and so on, behaved nothing at all like the real thing. Nobody knew how to make it realistic. (we still have that problem (?) with fire, explosions, and realistic human movement, etc. …)
- Reynold’s solved the problem by trying a very simple approach, which was inspired by a sensible view of how animals actually do it.

The Problem

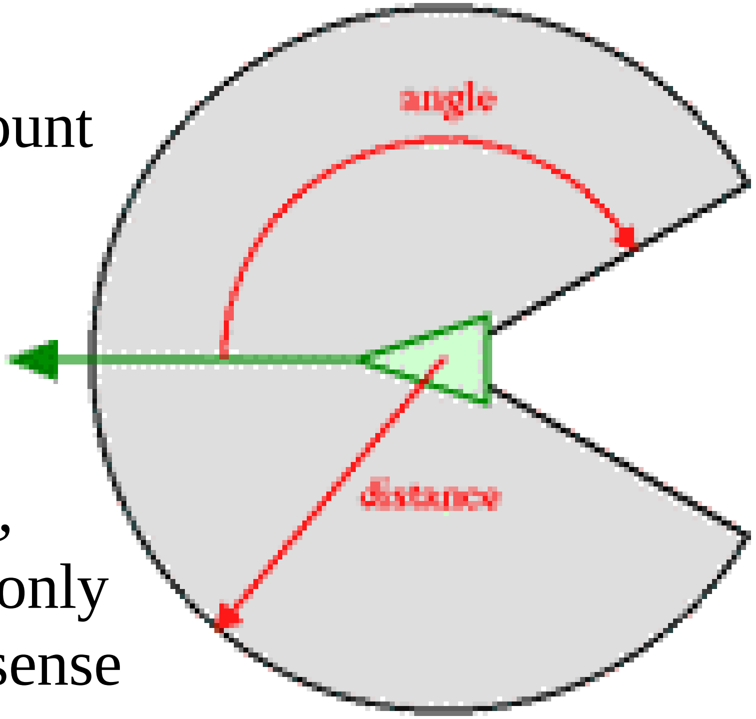


We would like these to move like a realistic flock of birds.
(The heading of each one is suggested by where it's pointing)
But how? Perhaps in the next timestep, they should all move the same small distance? They should all change their velocity in some way?

A Simple Perceptual System

The green boid can see a certain amount ahead, and is also aware of any flockmates within limits on either side.

Two parameters, **angle** and **distance**, define the system. So, this boid will only be influenced by those others it can sense according to these parameters.

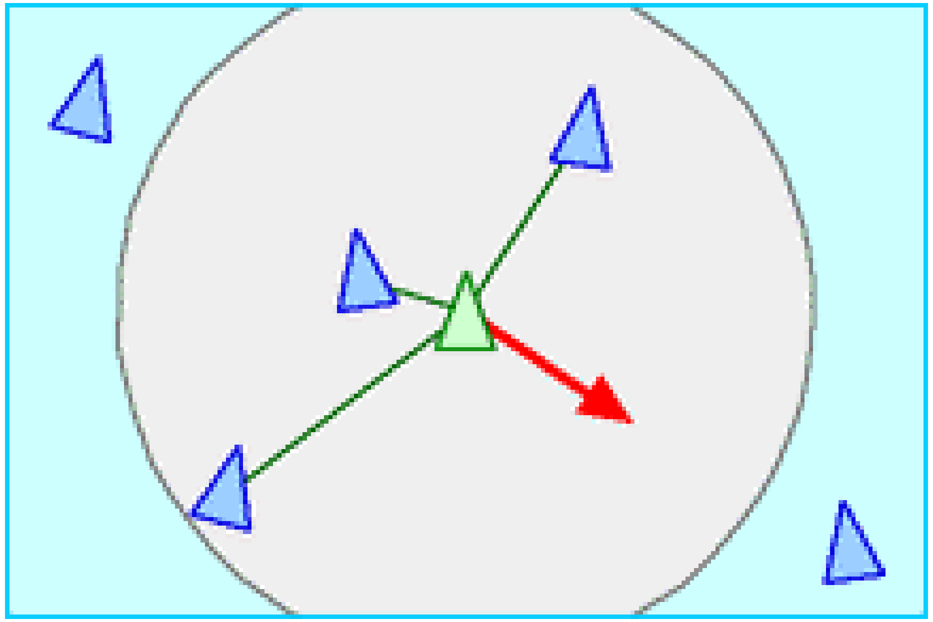


Picture is from Reynold’s boids page.

Rule 1: Separation

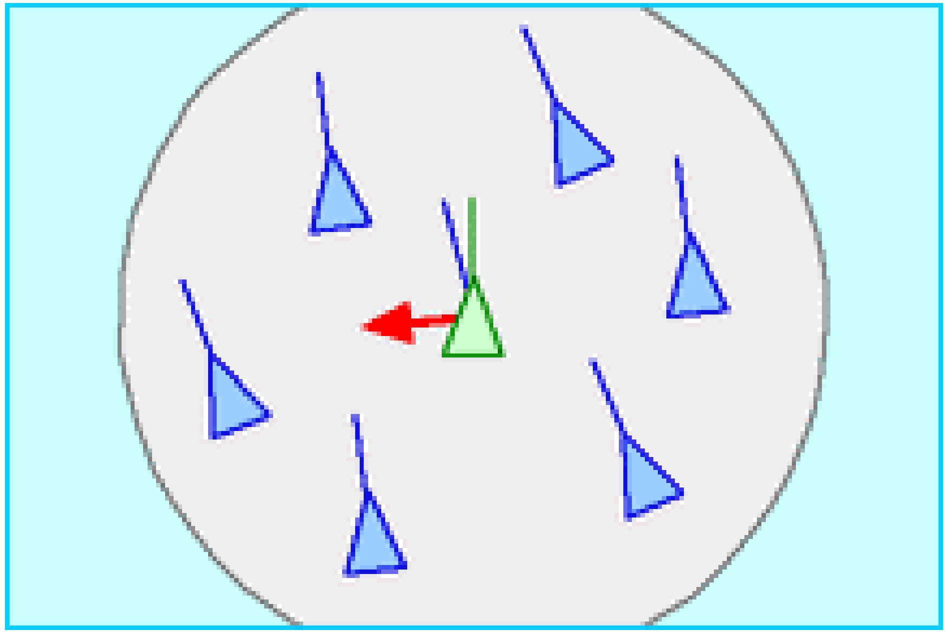
At each iteration, a boid makes an adjustment to its velocity according to the following rule:

Avoid getting too close to local (the ones it is aware of) flockmates.

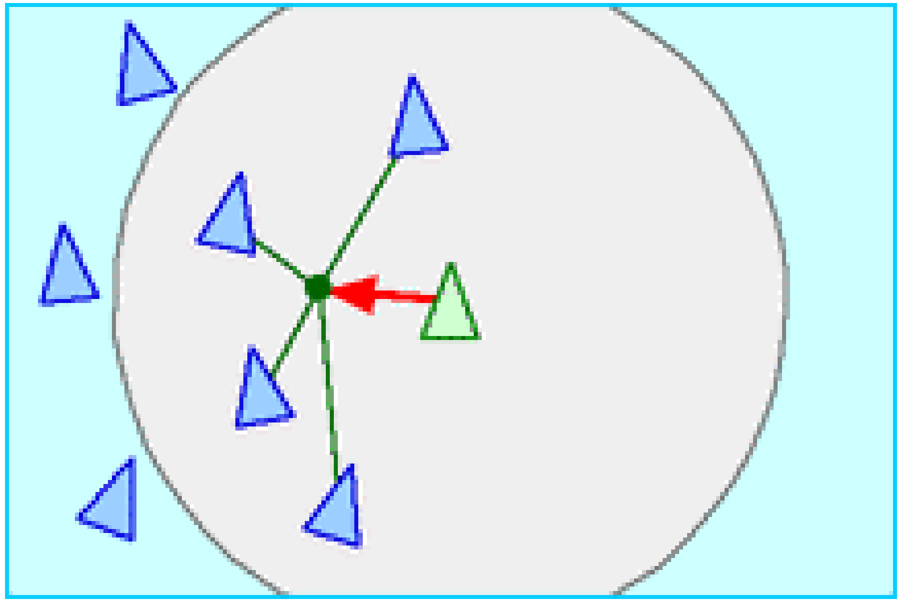


Rule 2: Alignment

At each iteration, a boid makes an adjustment to match its velocity to the average of that of its local flockmates.



Rule 3: Cohesion



At each iteration, a boid makes an adjustment to its velocity towards the centroid of its flockmates.

It’s not *Quite* as Simple as that to get Realistic Behaviour

1. Appropriate definition of the perceptive range. (What happens if it is too high or too low?)

2. Appropriate combination of the three rules. Opposing forces may cancel out.

3. Note that the cohesion rule is interesting – it leads to “bifurcating” around obstacles – a follow-the-leader approach to flocking would not achieve that.

4. The simple rules also realistically lead to “flash expansion” if started too close together.

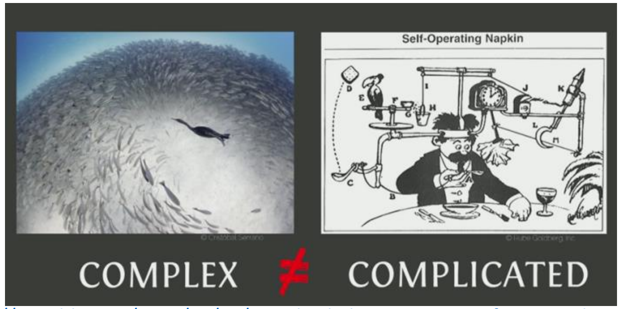
Declarative vs Rule-based Control for Flocking Dynamics

- Reynold's model is a *mechanistic* (operational) model. The desired flocking behavior *emerges* from the rules.

- *A declarative (descriptive) model* attains the desired flocking behavior by explicitly setting it as a goal in a cost function
  - Cost function: cohesion term + separation term
  - Model Predictive Control is used to define the respective controllers

Emergence: *The Whole is Greater than the Sum of its Parts* ????

- Complexity



https://social-biz.org/2014/02/10/complex-behavior-emerges-from-simple-rules/

- Different Time/Space–Scales

https://www.quantamagazine.org/evolution-runs-faster-on-short-timescales-20170314/

- Unexpected Behavior

https://en.wikipedia.org/wiki/Flock_(birds)

Star Trek Beyond

Emergent Algorithm

An emergent algorithm implements a set of simple *building block* behaviors that when combined exhibit more complex (not complicated necessarily) behaviors.

An *emergent algorithm* may have the following characteristics:

1. it achieves predictable global effects
2. it does not require global visibility
3. it does not assume any kind of centralized control
4. it is self-stabilizing

Emergence in CS

1. Distributed algorithms (simple rules like averaging and plurality)

2. Nature-inspired methods (e.g., ant colony systems and genetic algorithms)

3. Artificial neural networks (iterative coarse-graining)

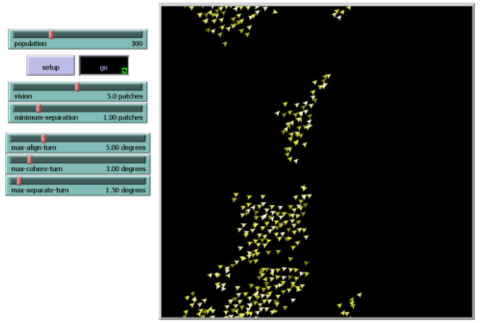
4. Cellular Automata

NETLOGO

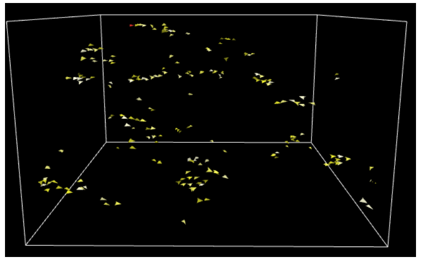
I prefer experiments to analysis…

More with V. Thomopoulos…

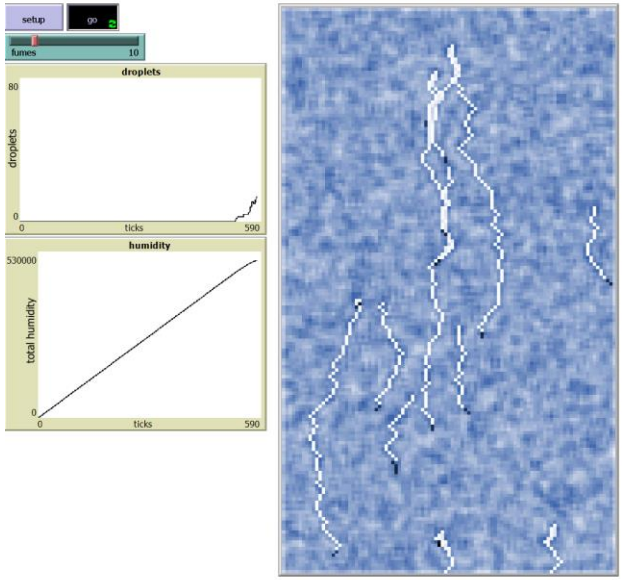
Various Models

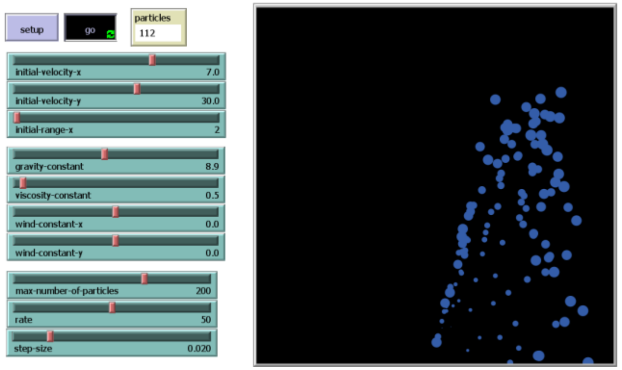


- Flocking



- Flocking in 3D
- Particle Systems (fountain)





- One of your projects