Computing through Simplicity: **Towards a Theory of Dynamics**

Emanuele Natale*

Assemblée Générale
I3S, Sophia Antipolis
20 Decembre 2018

* At max planck institut informatik until 31 December

My Algorithmic Biography

- 2016 - PhD at Sapienza University, in **Theory of Distributed Computing**

My Algorithmic Biography

- 2016 - PhD at Sapienza University, in **Theory of Distributed Computing**

SAPIENZA
UNIVERSITÀ DI ROMA

- 2016-now - PostDoc at
  Max Planck Institute for Informatics
  D1 - Algorithms & Complexity

max planck institut
informatik

My Algorithmic Biography

- 2016 - PhD at Sapienza University, in **Theory of Distributed Computing**

SAPIENZA
UNIVERSITÀ DI ROMA

- 2016-now - PostDoc at
  Max Planck Institute for Informatics
  D1 - Algorithms & Complexity

max planck institut
informatik

- 2016 & 2018 - **Fellow** of
  Simons Institute for the
  Theory of Computing

SIMONS INSTITUTE
for the Theory of Computing

Berkeley
UNIVERSITY OF CALIFORNIA

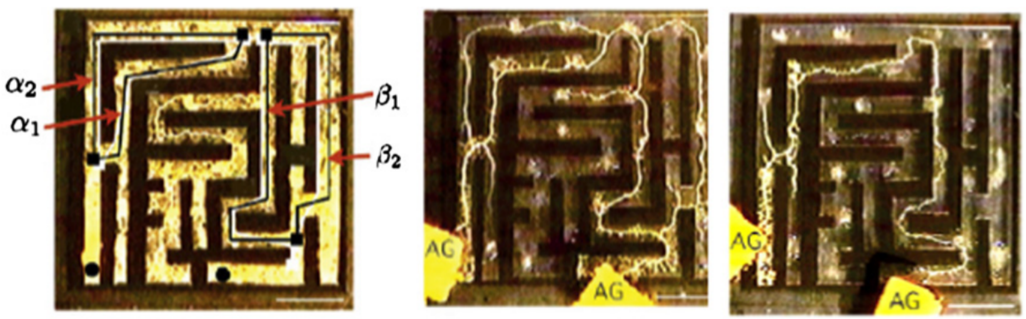
Computational Dynamics

Natural Algorithms

How do flocks of birds synchronize their flight? [Chazelle '09]

Natural Algorithms

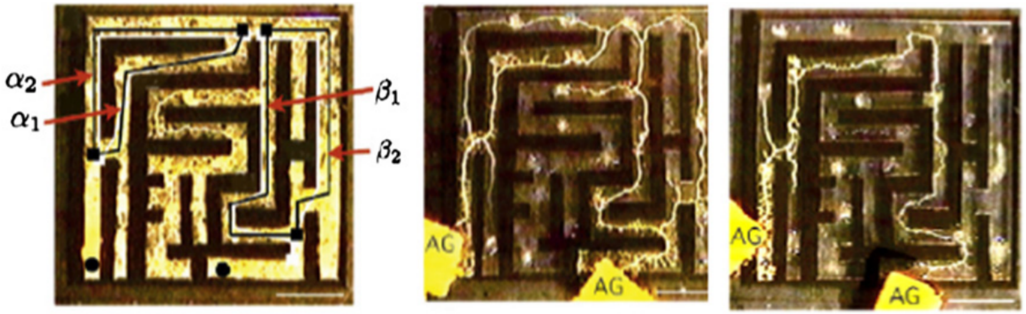
How do flocks of birds synchronize their flight? [Chazelle '09]



How does Physarum polycephalum finds shortest paths? [Mehlhorn et al. 2012-...]

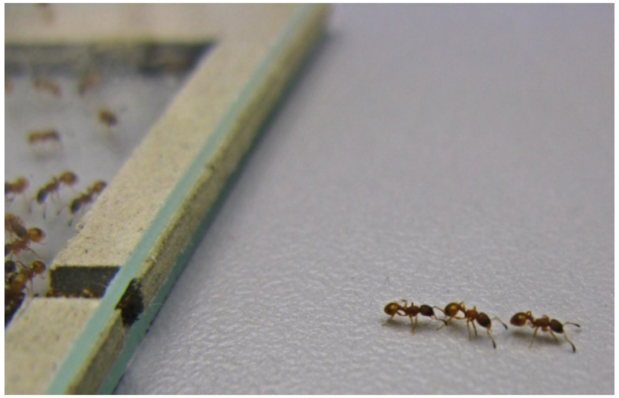
Natural Algorithms

How do flocks of birds synchronize their flight? [Chazelle '09]

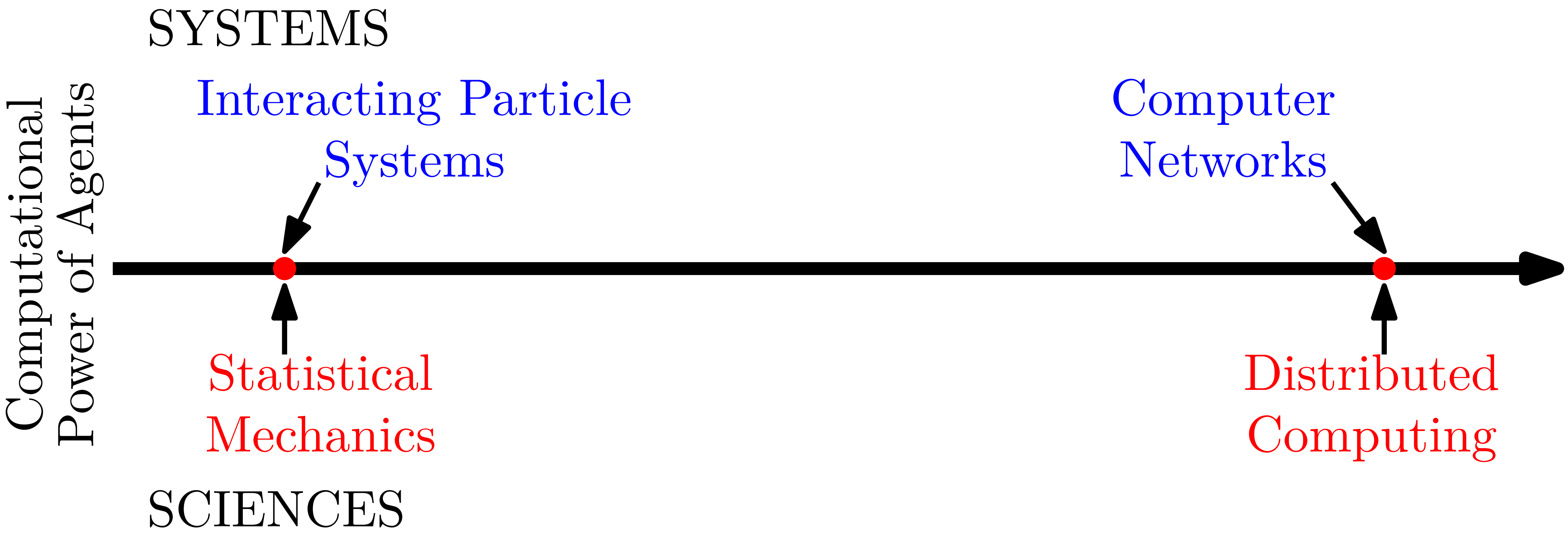


How does Physarum polycephalum finds shortest paths? [Mehlhorn et al. 2012-...]

How ants perform collective navigattion? How do they decide where to relocate their nest?

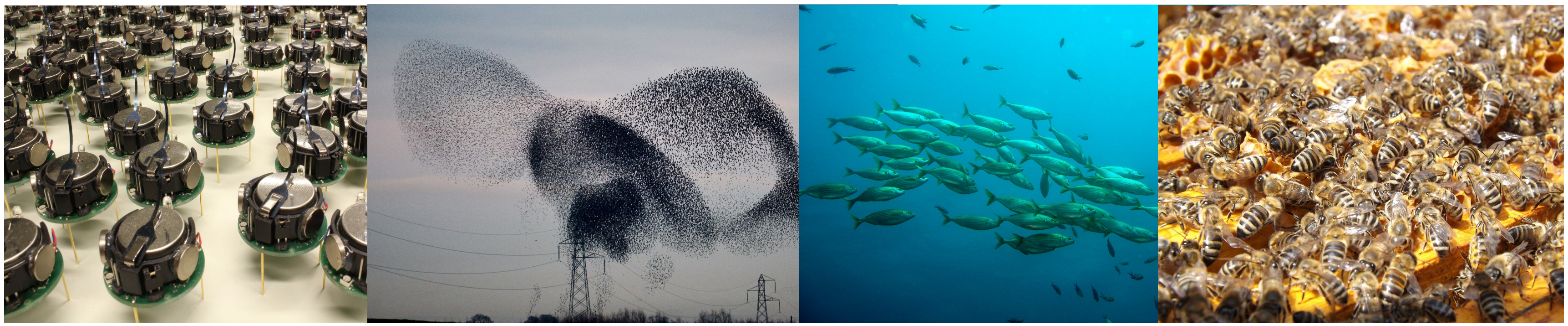


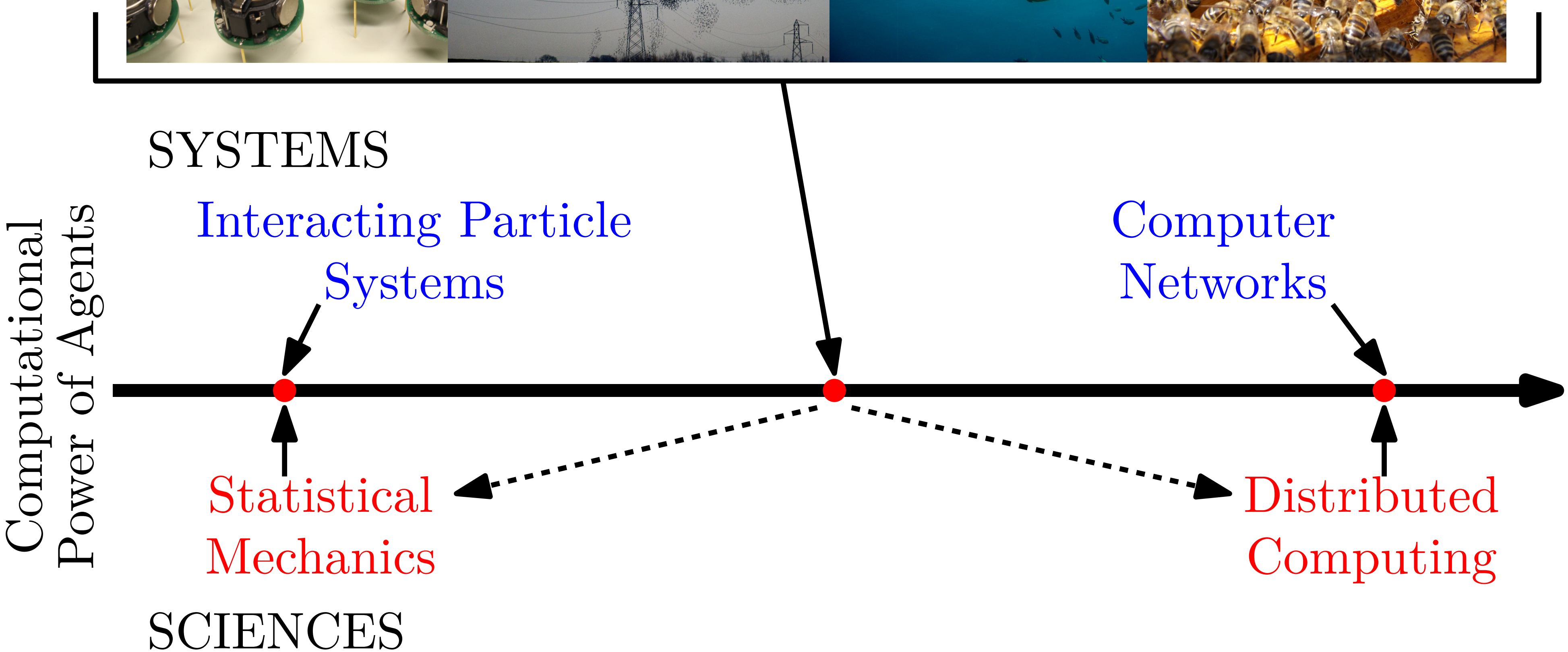
How can *Locally-Simple* Systems *Compute*?



How can *Locally-Simple* Systems *Compute*?

A **computational lens** on how
global behavior emerges from
simple local interactions among individuals

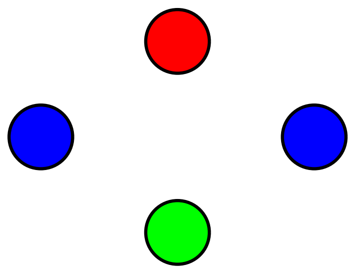




Computational Dynamics

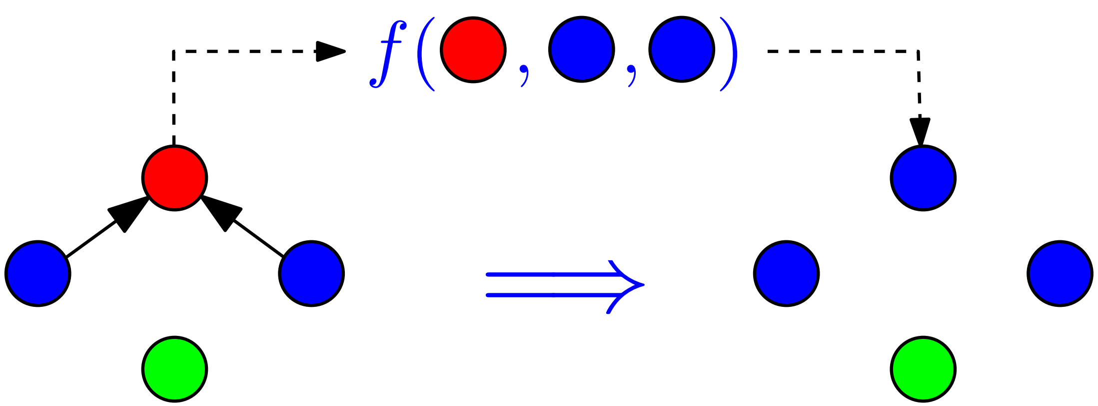
Anonymous agents

- small set of possible states
- *simple* update function f



At each step:

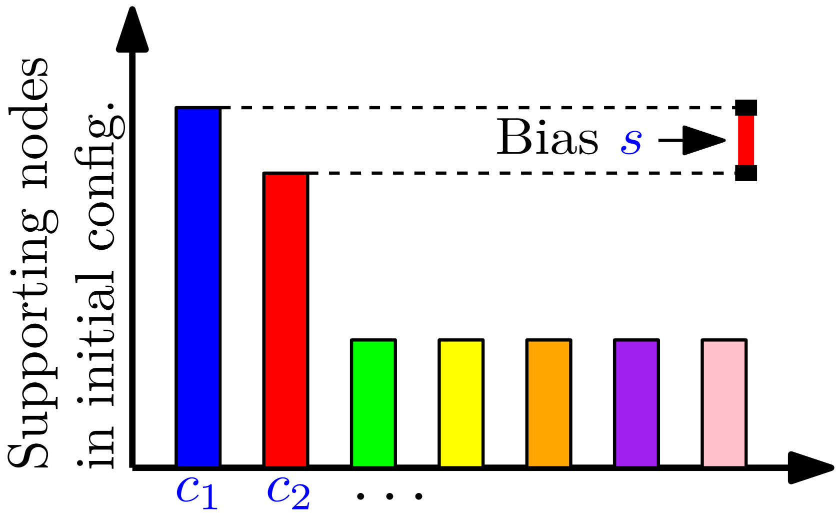
Update depends on states of random subset of agents



Dynamics for Plurality Consensus I

Plurality Consensus.

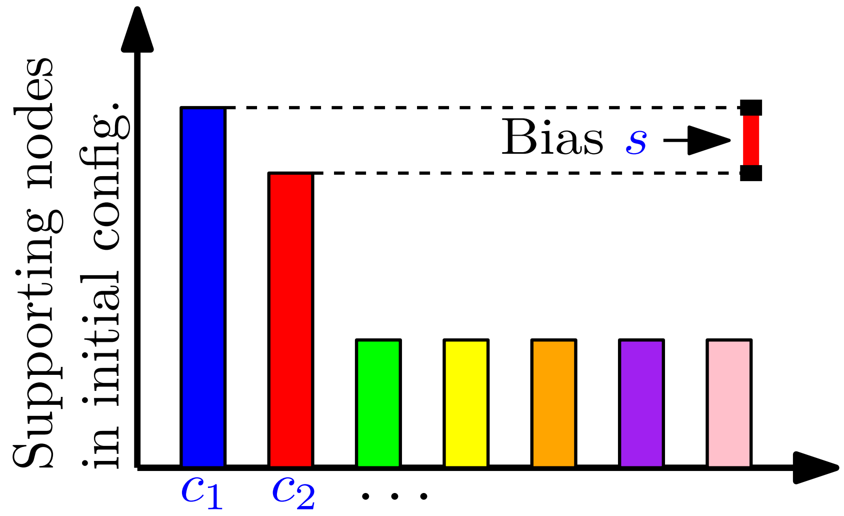
- Each agent initially has a value in $\{1, ..., k\}$.
- $\Omega(\sqrt{kn \log n})$ initial **bias** (majority − 2nd-majority color).
- Each agent eventually has the most frequent initial value.



Dynamics for Plurality Consensus I

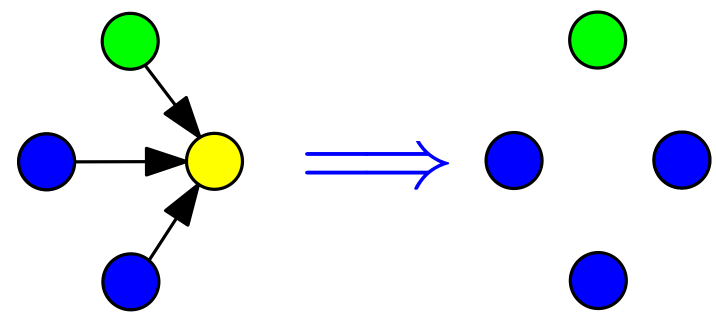
Plurality Consensus.

- Each agent initially has a value in $\{1, ..., k\}$.
- $\Omega(\sqrt{kn \log n})$ initial **bias** (majority − 2nd-majority color).
- Each agent eventually has the most frequent initial value.



3-Majority Dynamics.
At each round, each agent samples 3 agents and adopts the majority color.

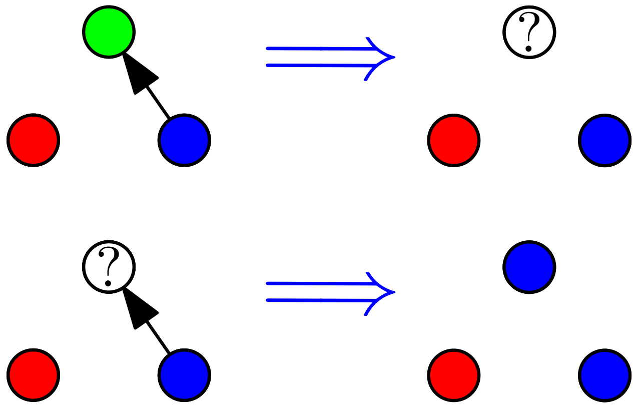
Theorem.
3-Majority Dynamics converges to plurality in $\mathcal{O}(k \log n)$ rounds



Dynamics for Plurality Consensus II

Undecided-State Dynamics.
Each agent u samples an agent v:

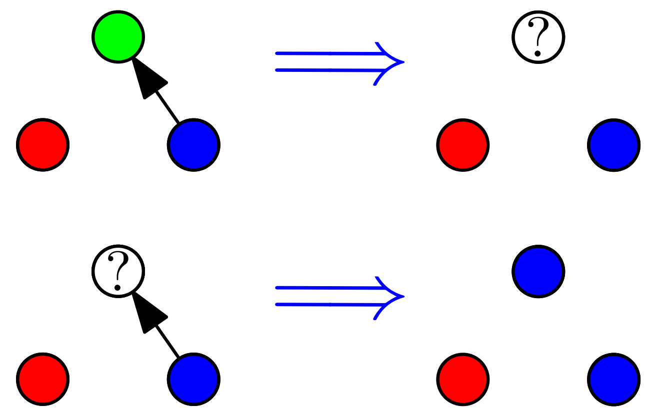
- *If v has a different color, u becomes **undecided**.*
- *If undecided, u copies the color of v.*



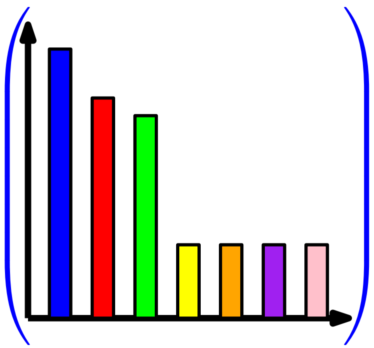
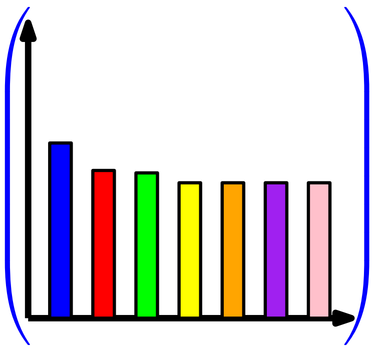
Dynamics for Plurality Consensus II

Undecided-State Dynamics.
Each agent u samples an agent v:

- *If v has a different color, u becomes* ***undecided****.*
- *If undecided, u copies the color of v.*

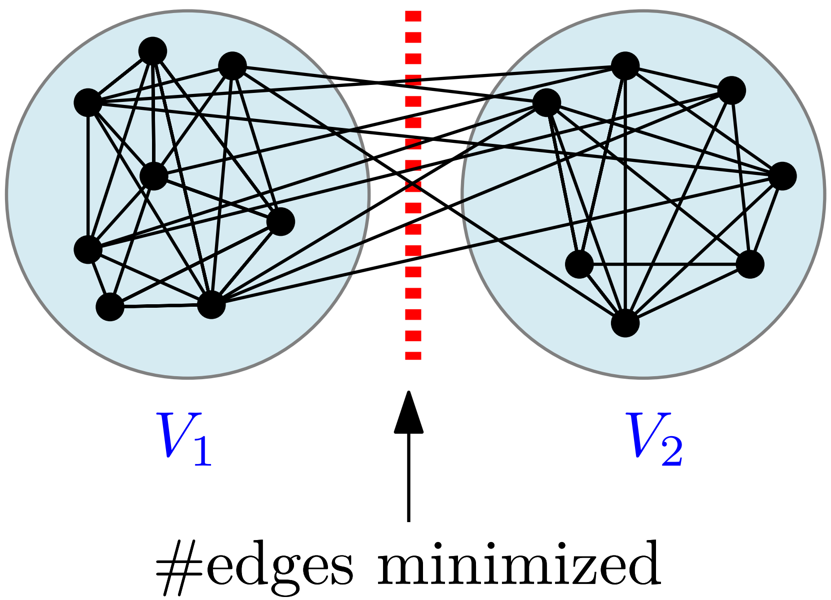


Theorem (Monochromatic Distance).
Undecided-State Dynamics converges to plurality within $\tilde{\Theta}(\mathrm{md}(\text{initial configuration}))$ rounds with high probability.

$1 \leq \mathrm{md}\Big($  $\Big) \ll \mathrm{md}\Big($  $\Big) \leq k$

Clustering

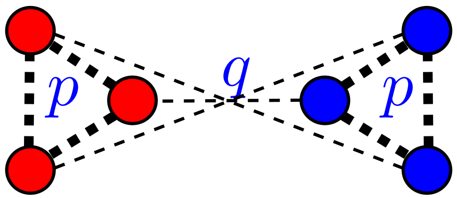
Minimum Bisection Problem.
Find balanced bipartition $|V_1| = |V_2|$ that minimizes cut.



[Garey et al. '76]: Minimum bisection problem is NP-Complete!

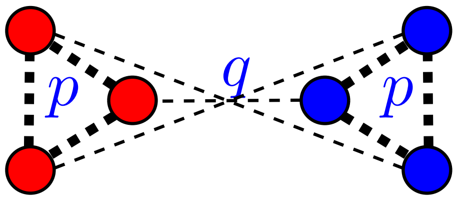
Stochastic Block Model (SBM)

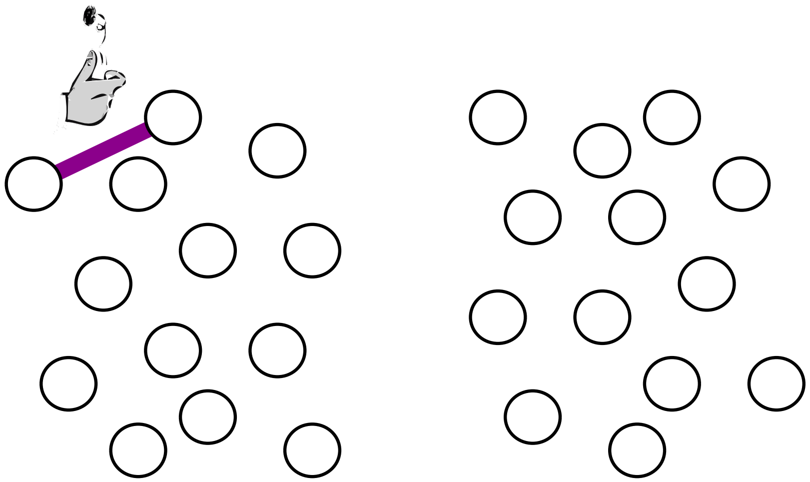
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



Stochastic Block Model (SBM)

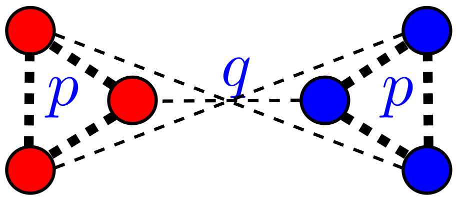
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.

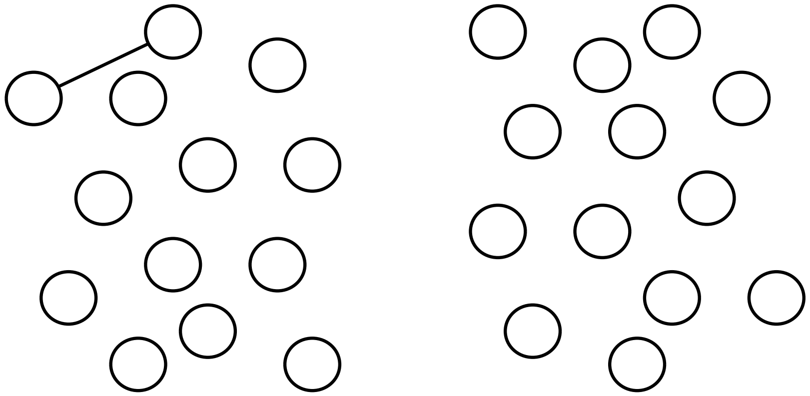




Stochastic Block Model (SBM)

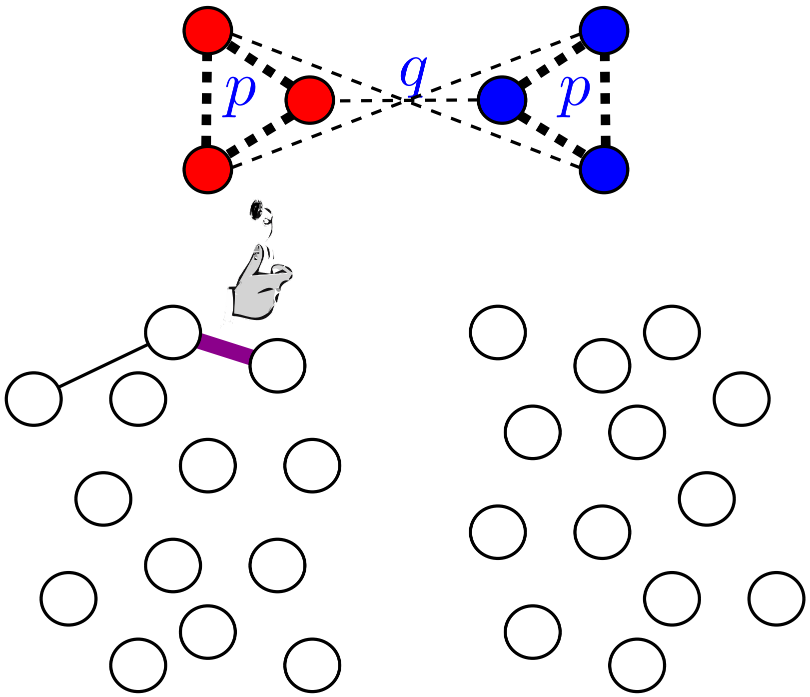
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.





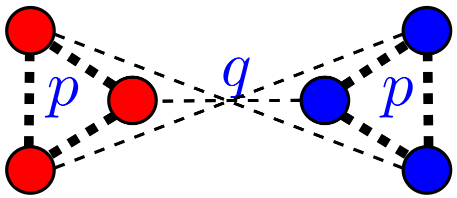
Stochastic Block Model (SBM)

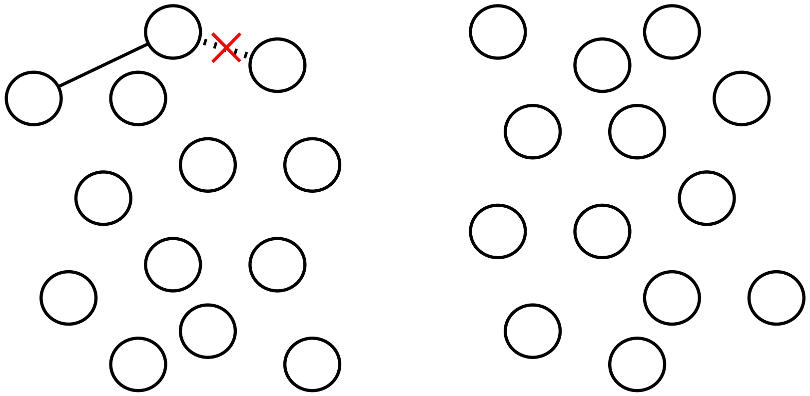
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



Stochastic Block Model (SBM)

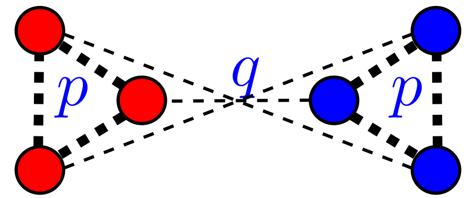
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.

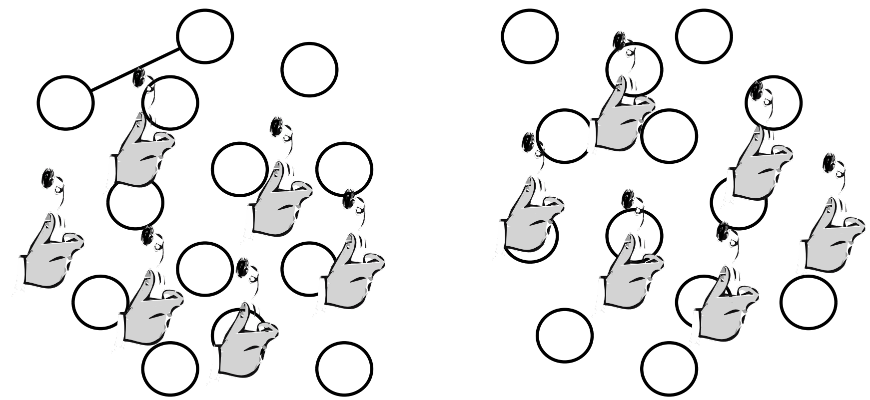




Stochastic Block Model (SBM)

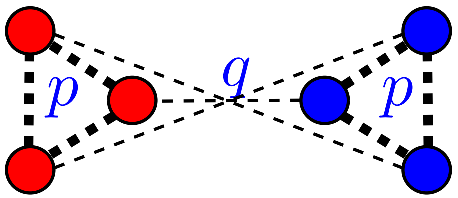
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.

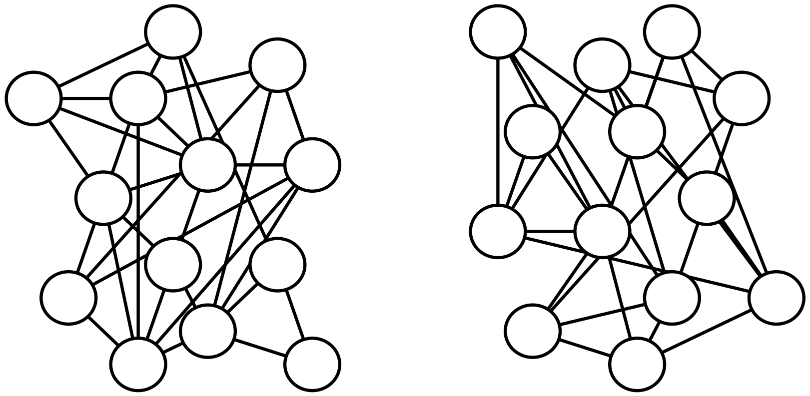




Stochastic Block Model (SBM)

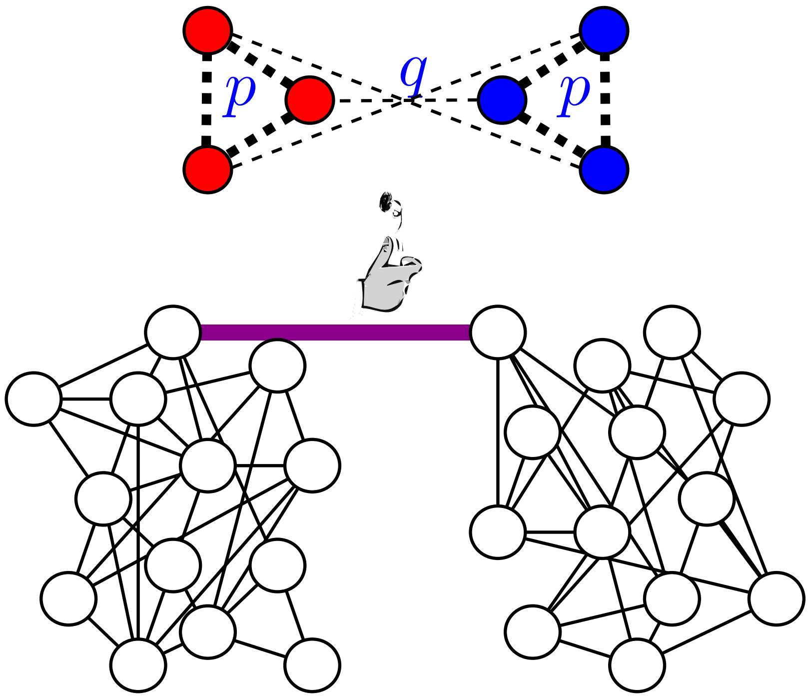
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.





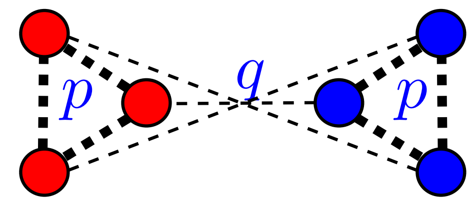
Stochastic Block Model (SBM)

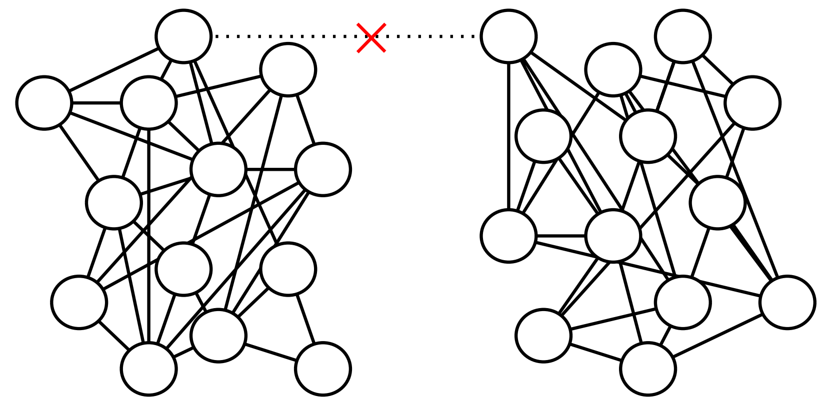
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



Stochastic Block Model (SBM)

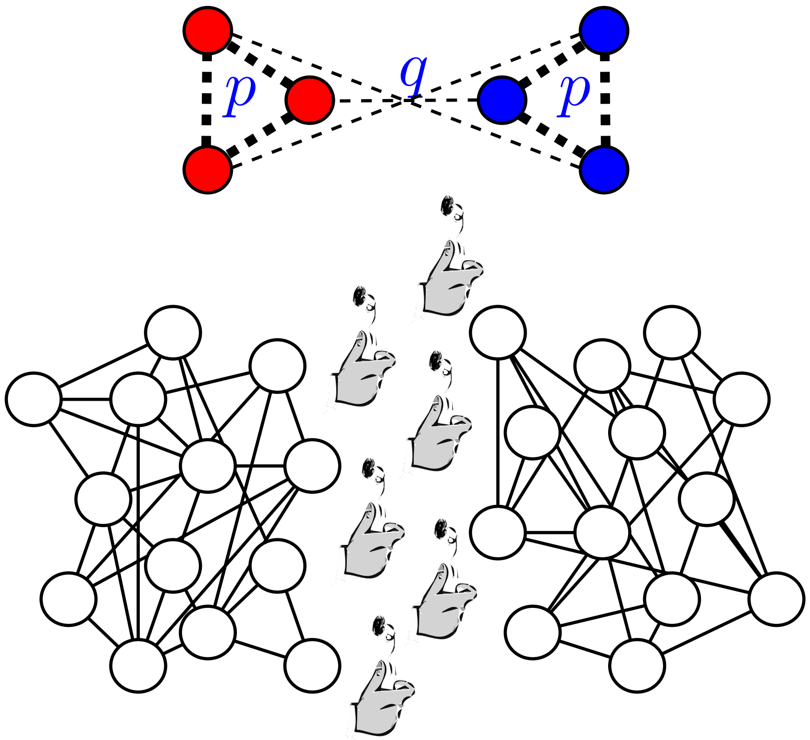
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.





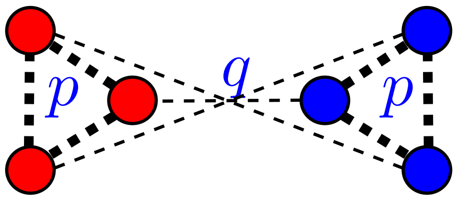
Stochastic Block Model (SBM)

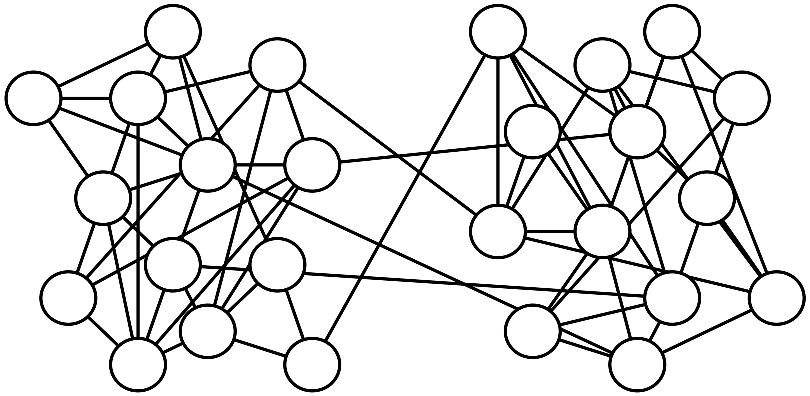
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



Stochastic Block Model (SBM)

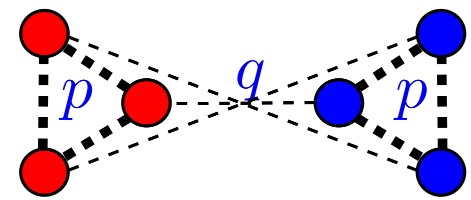
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



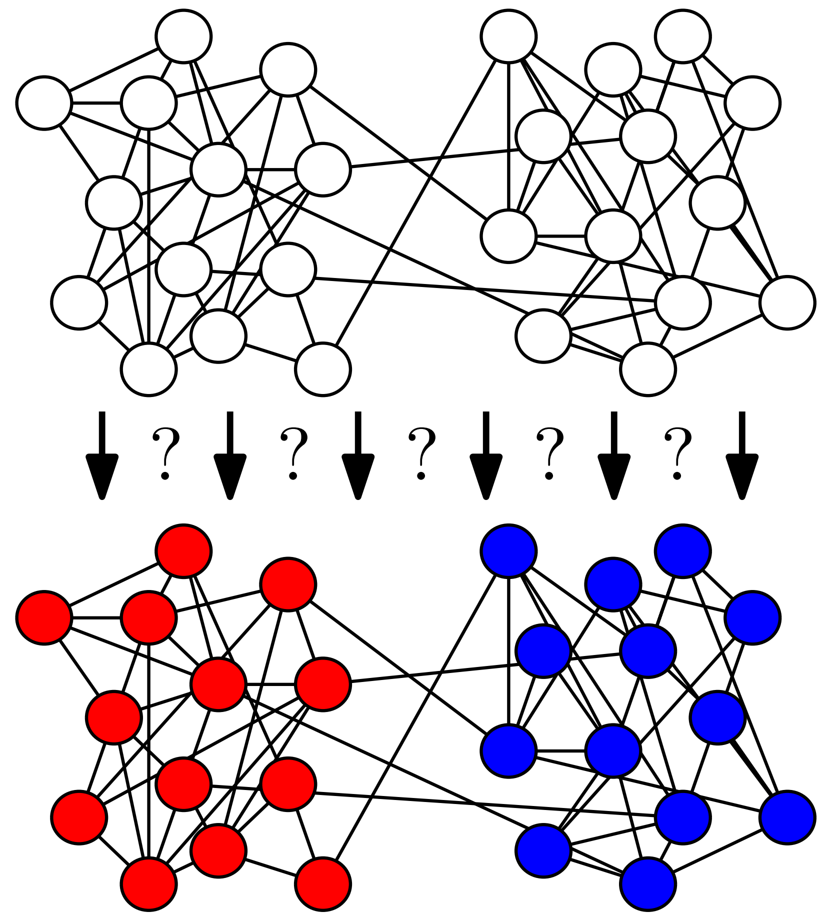


Stochastic Block Model (SBM)

- “Communities” V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.

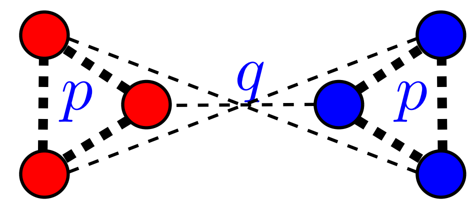


“Reconstruction” problem. Given graph generated by SBM, find original clusters.

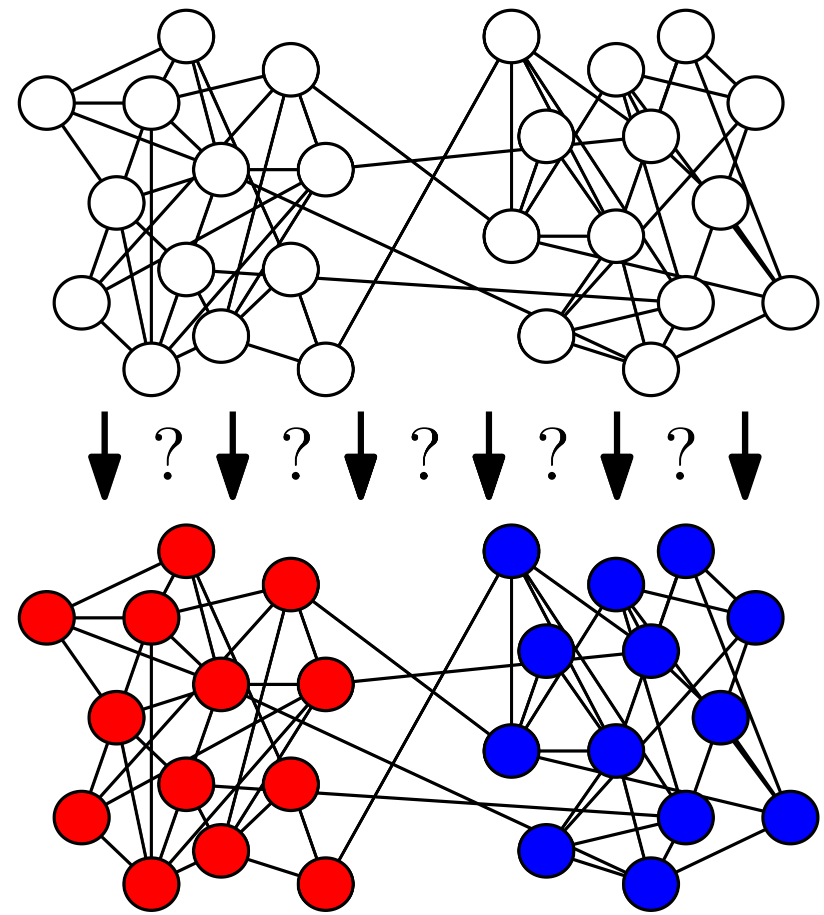


Stochastic Block Model (SBM)

- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



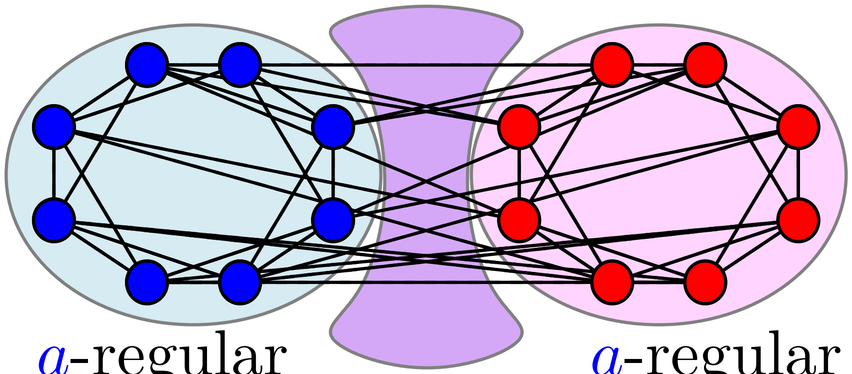
"Reconstruction" problem. Given graph generated by SBM, find original clusters.



Theorem. [Mossel et al. 2012-] Clustering possible **if and only if** p and q in a precise regime.

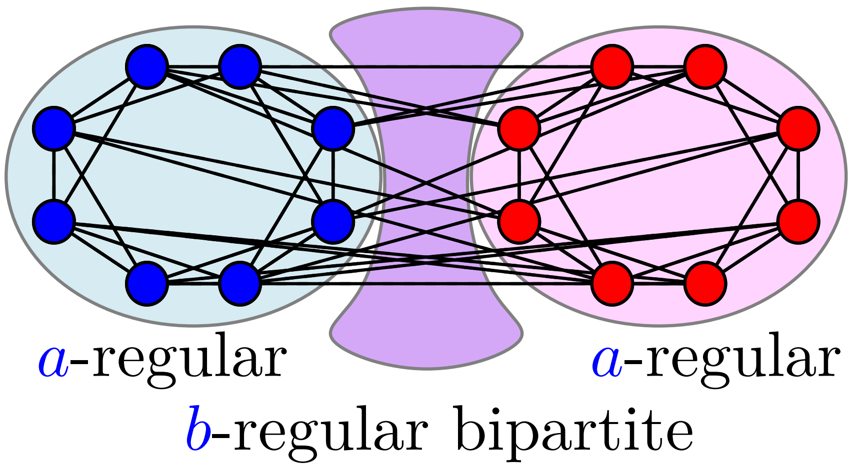
Clustering with **Averaging Dynamics**

Regular Stochastic Block Model:



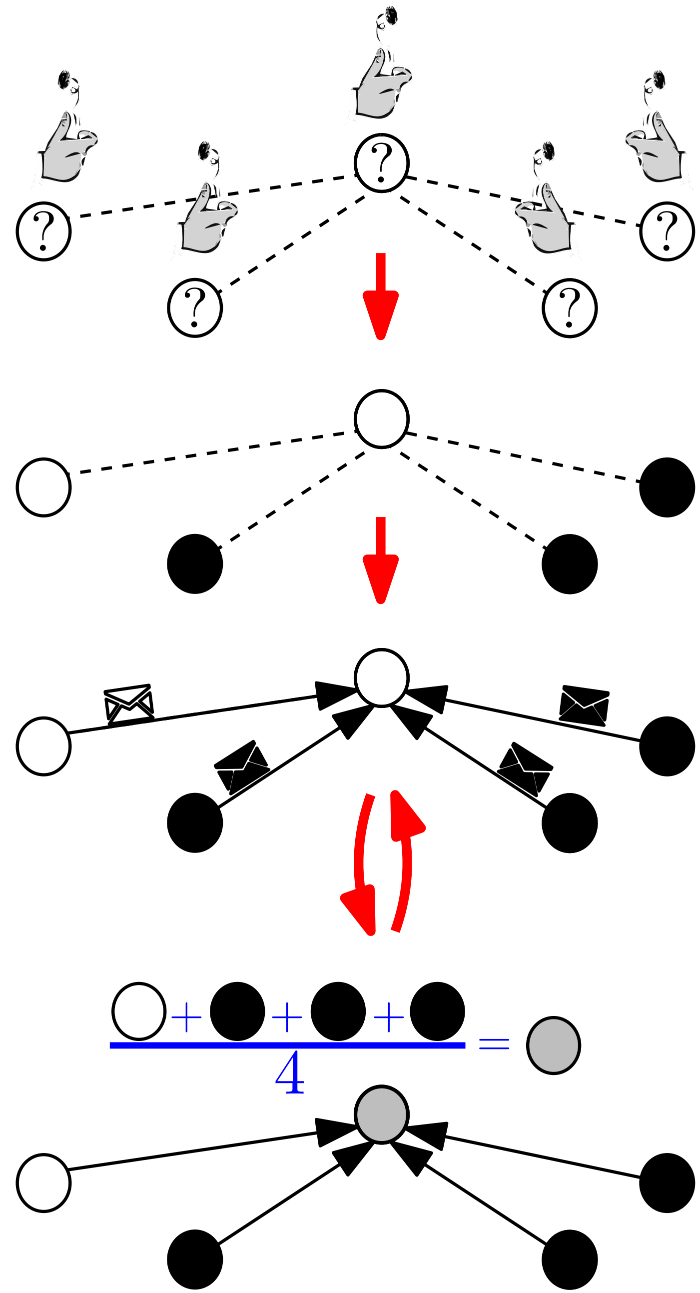
Clustering with **Averaging Dynamics**

Regular Stochastic Block Model:

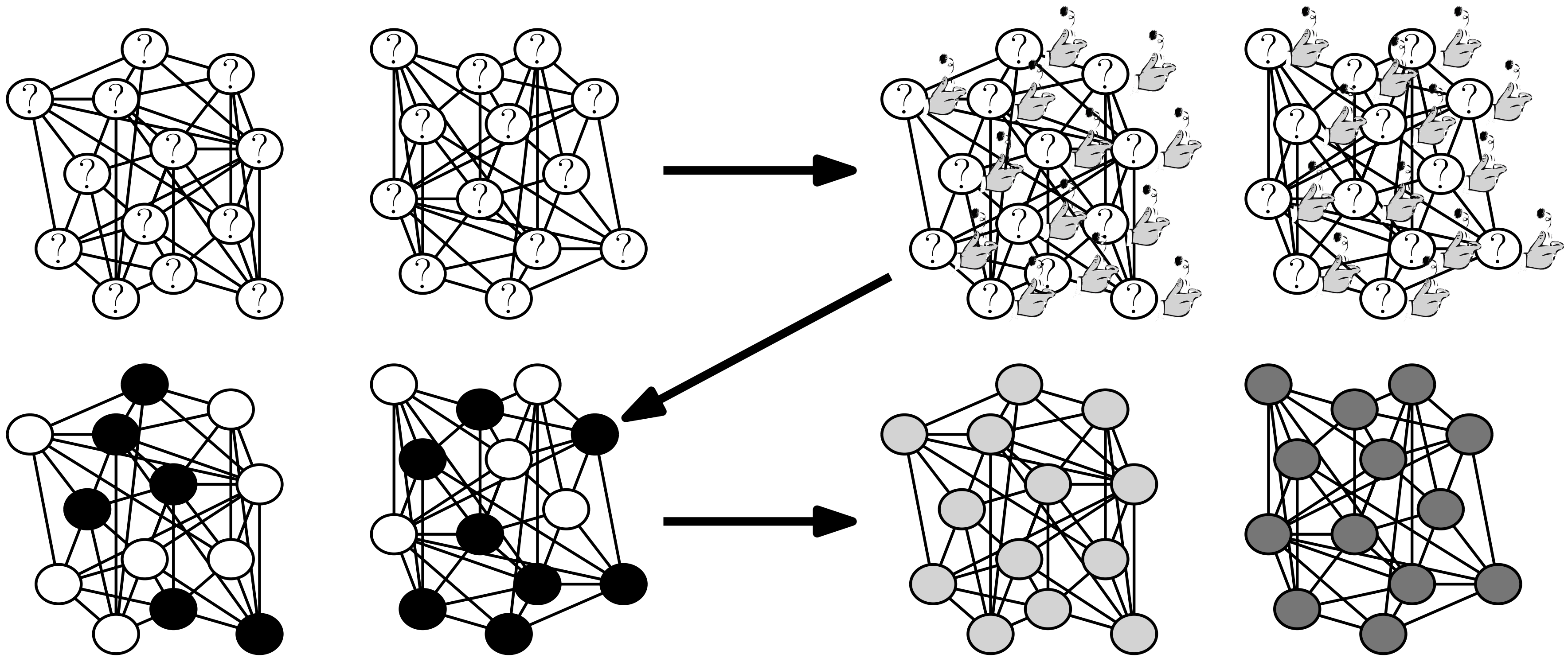


All nodes at the same time:

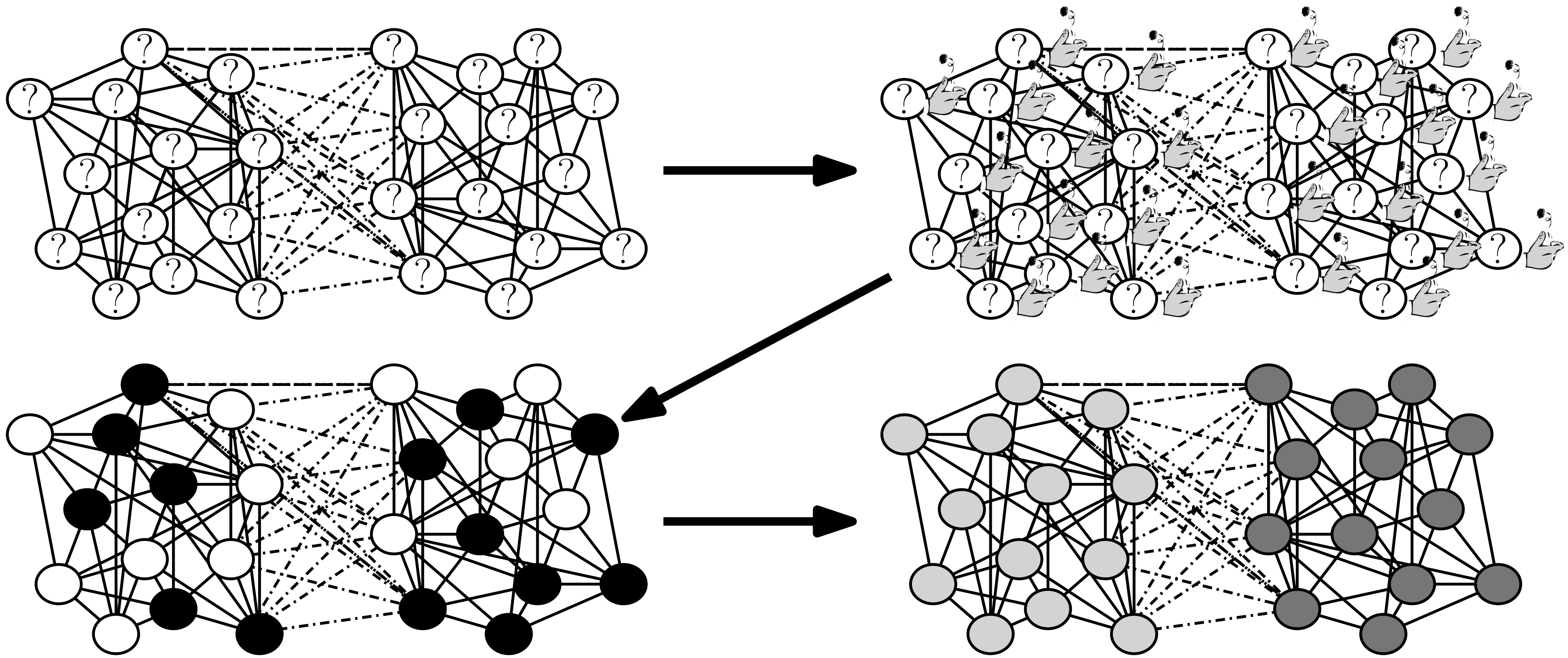
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$
- Then, at each round set value $x^{(t)}$ to average of neighbors



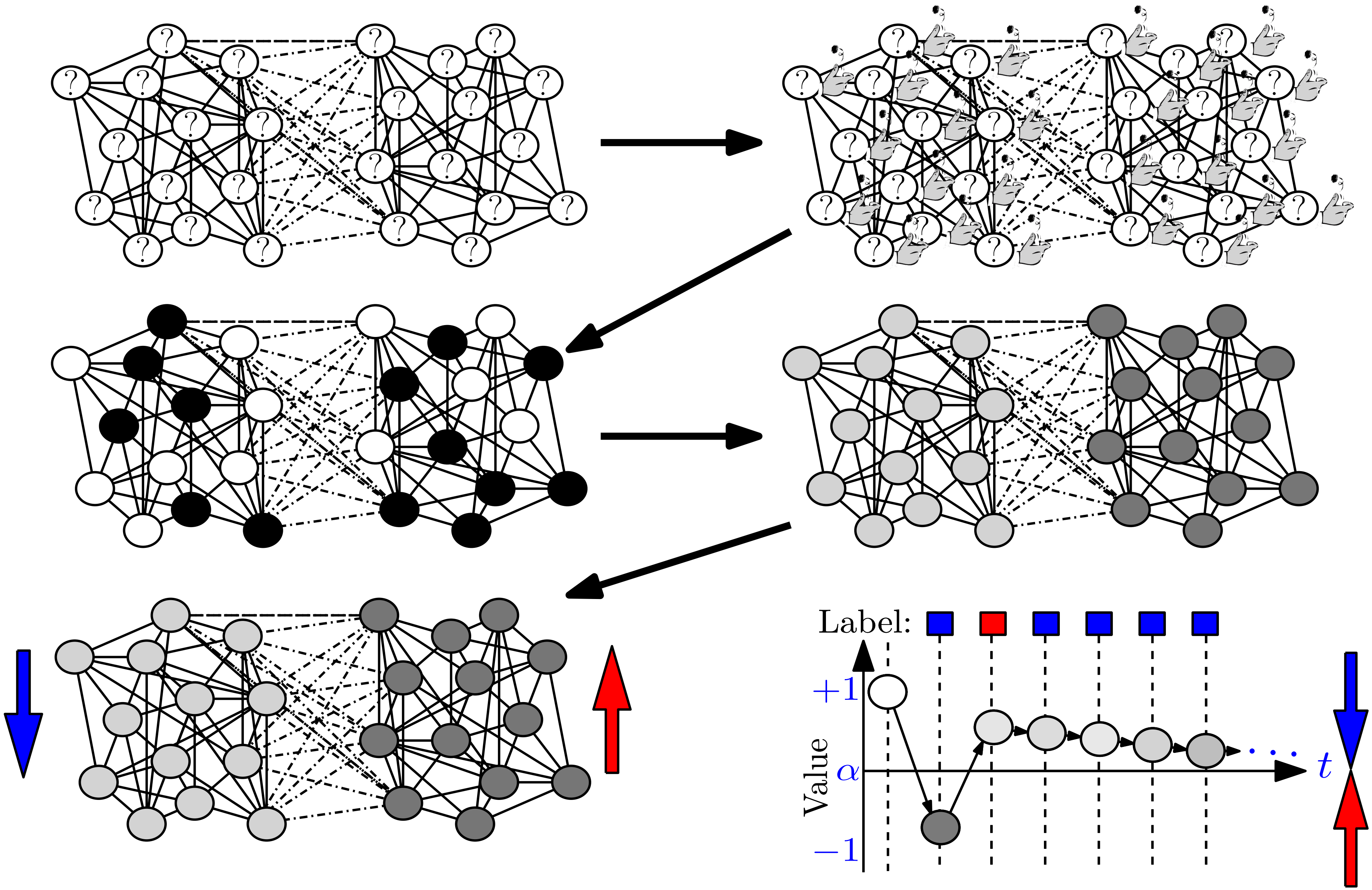
Why it Works: Intuition



Why it Works: Intuition



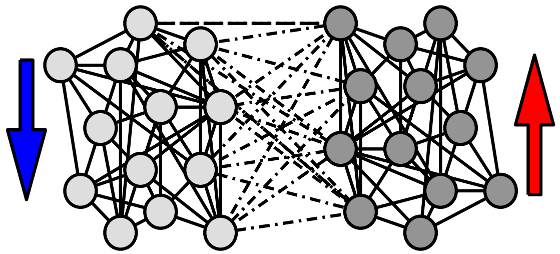
Why it Works: Intuition



- Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise

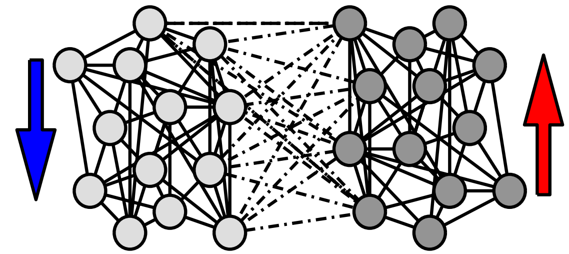
Why It Works: Proof Idea

Theorem. In Regular Stochastic Block Model with $a - b > \sqrt{2(a+b)}$, Averaging Dynamics finds clusters after $\frac{\log n}{\log \lambda_2/\lambda_3}$ steps with high probability.



Why It Works: Proof Idea

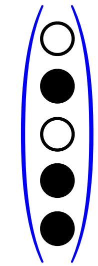
Theorem. In Regular Stochastic Block Model with $a - b > \sqrt{2(a+b)}$, Averaging Dynamics finds clusters after $\frac{\log n}{\log \lambda_2/\lambda_3}$ steps with high probability.



Averaging is a **linear** dynamics:

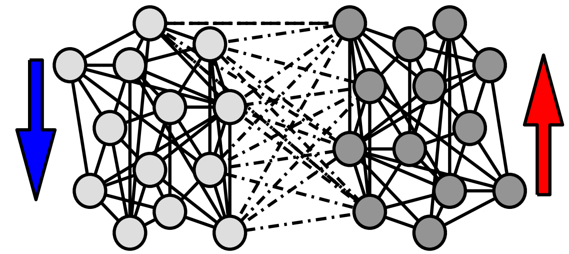
$$\mathbf{x}^{(t)} = P \cdot \mathbf{x}^{(t-1)} = P^t \cdot \mathbf{x}^{(0)}$$

P transition matrix of random walk on G and $\mathbf{x}^{(t)} =$



Why It Works: Proof Idea

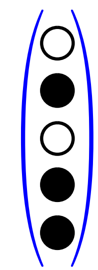
Theorem. In Regular Stochastic Block Model with $a - b > \sqrt{2(a+b)}$, Averaging Dynamics finds clusters after $\frac{\log n}{\log \lambda_2/\lambda_3}$ steps with high probability.



Averaging is a **linear** dynamics:

$$\mathbf{x}^{(t)} = P \cdot \mathbf{x}^{(t-1)} = P^t \cdot \mathbf{x}^{(0)}$$

P transition matrix of random walk on G and $\mathbf{x}^{(t)} =$



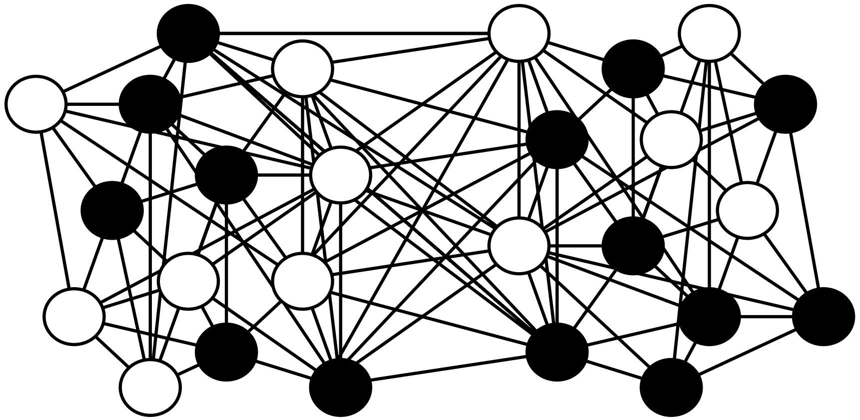
$$\mathbf{x}^{(t)} = \frac{1}{\tilde{\Theta}(\sqrt{n})} \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \left(\frac{a-b}{a+b}\right)^t \frac{1}{\tilde{\Theta}(\sqrt{n})} \begin{pmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{pmatrix} + \mathbf{e}^{(t)}$$

negligible after $t \gg \frac{\log n}{\log \lambda_2/\lambda_3}$

$$\mathrm{sign}(\mathbf{x}^{(t)}(u) - \mathbf{x}^{(t-1)}(u)) = \mathrm{sign}\left(\begin{pmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}\right)$$

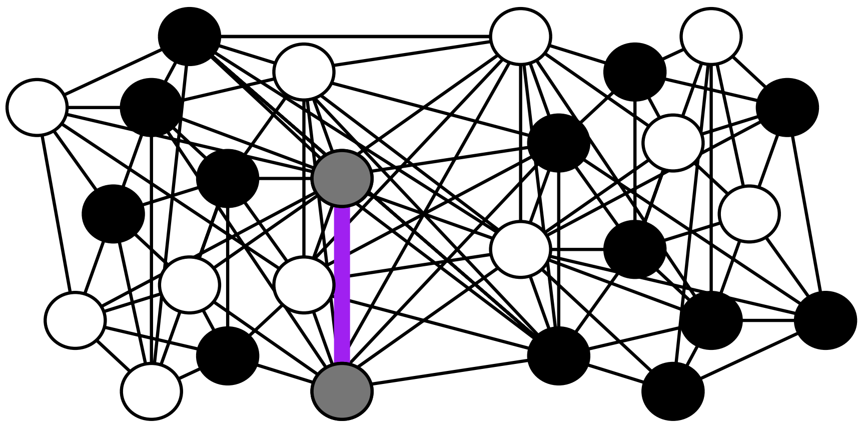
Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u *initially flips a coin and gets value* $+1$ *or* -1*.*
At each step, an edge $\{u, v\}$ *is chosen u.a.r. and* u *and* v *average their values.*



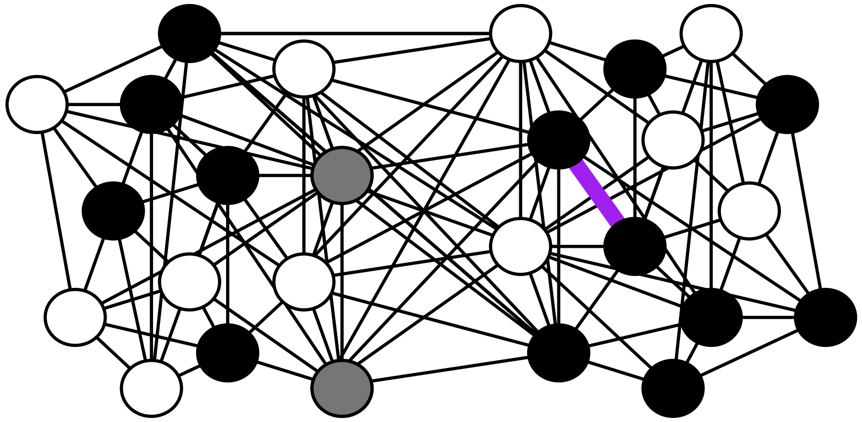
Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u *initially flips a coin and gets value* $+1$ *or* -1*. At each step, an edge* $\{u, v\}$ *is chosen u.a.r. and* u *and* v *average their values.*



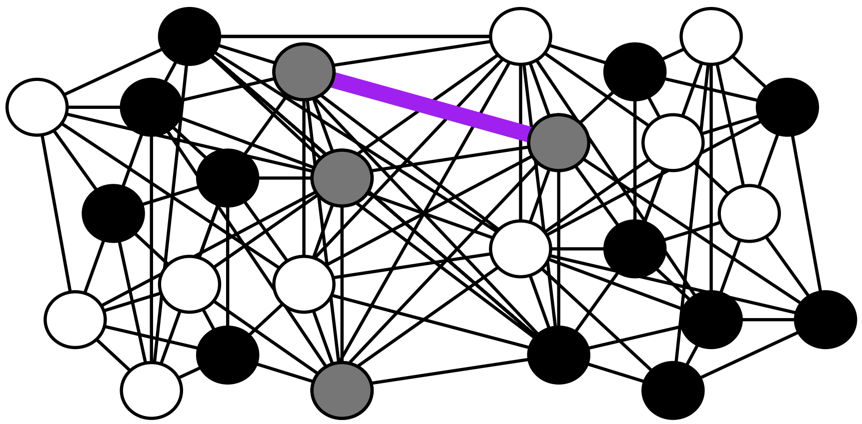
Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u initially flips a coin and gets value $+1$ or -1.
At each step, an edge $\{u, v\}$ is chosen u.a.r. and u and v average their values.



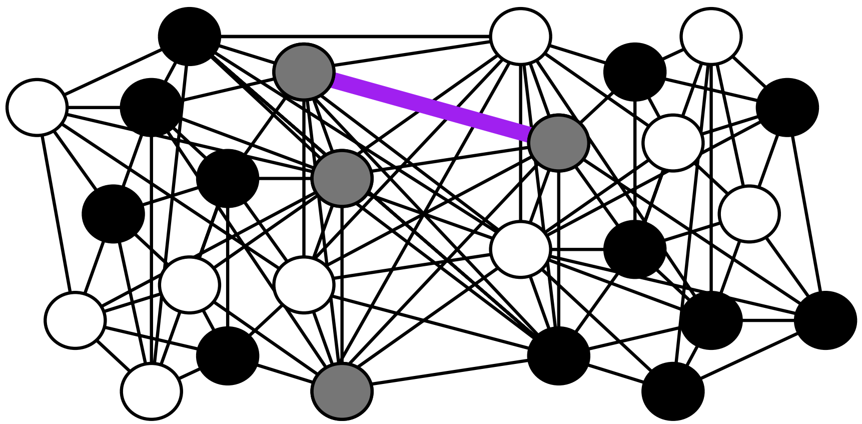
Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u *initially flips a coin and gets value* $+1$ *or* -1*.*
At each step, an edge $\{u, v\}$ *is chosen u.a.r. and* u *and* v *average their values.*



Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u initially flips a coin and gets value $+1$ or -1. At each step, an edge $\{u, v\}$ is chosen u.a.r. and u and v average their values.



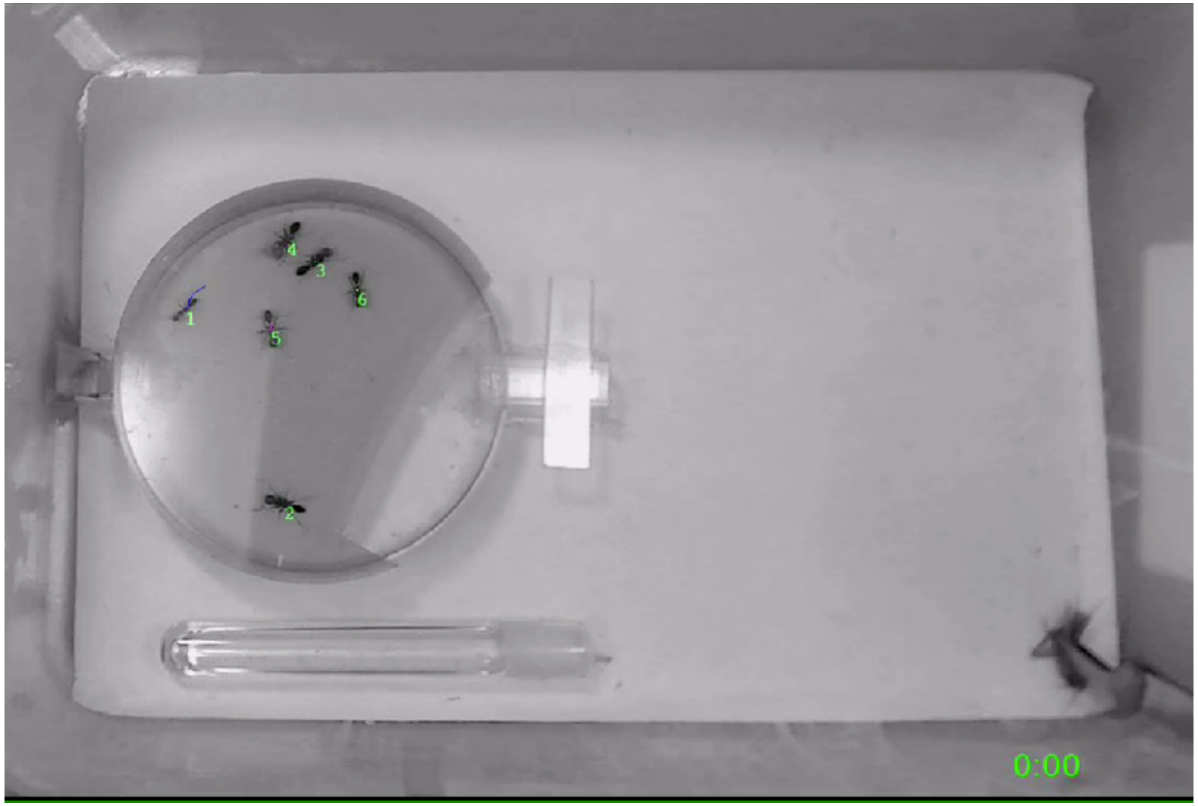
Theorem. In Regular Stochastic Block Model

- An AAD-based protocol finds clusters in $C_{\lambda_2 - \lambda_1} n(\frac{a}{b} + \log n)$ with high probability.
- If $\lambda_2 \ll \frac{\lambda_3^2}{\log^2 n}$, another AAD-based protocol finds clusters after $\mathcal{O}(\frac{n}{\lambda_3} \log^2 n)$ steps with high probability.

Part II

Biological Distributed Algorithms

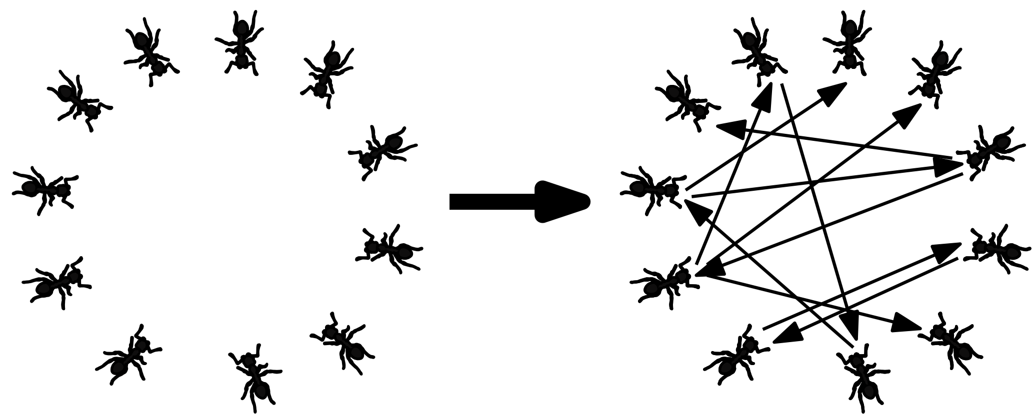
Recruitment in Desert **Ants**



Cataglyphis niger needs to recruit nest mates to carry food.
Data suggest that they communicate by simple, *stochastic noisy interactions*.
We provide **mathematical evidence** on why stochastic noisy interactions imply *small group size*.

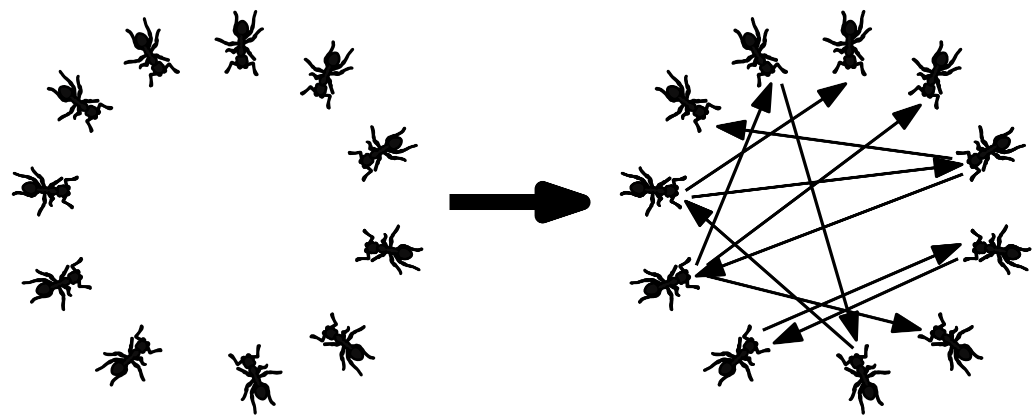
Noisy & Stochastic Interactions

Stochastic Interactions.
At each round, each agent receives a message from another random agent.

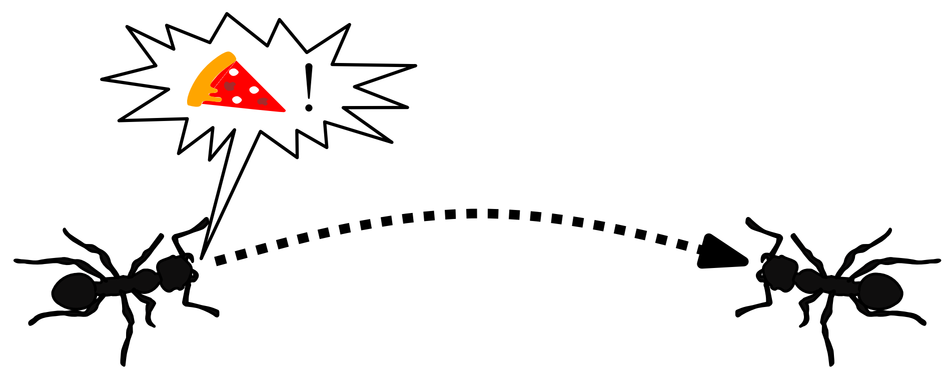


Noisy & Stochastic Interactions

Stochastic Interactions. At each round, each agent receives a message from another random agent.

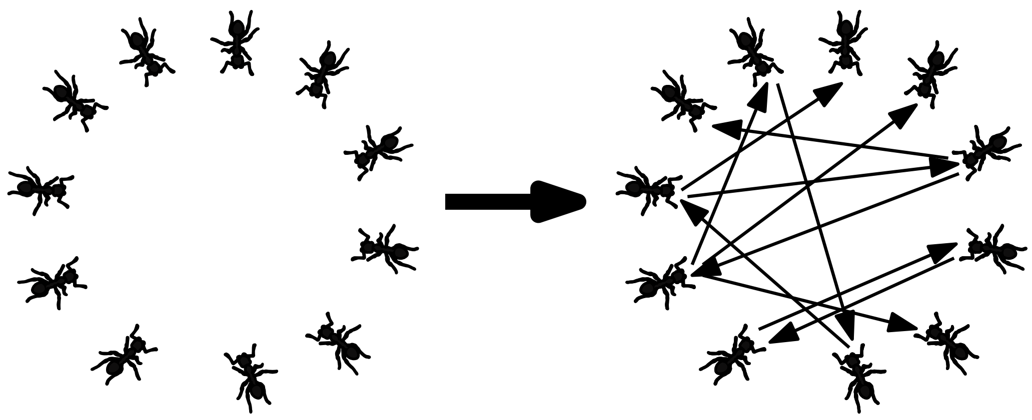


Noisy Communication. Before being received, each bit is flipped with probability $1/2 - \epsilon_n$.

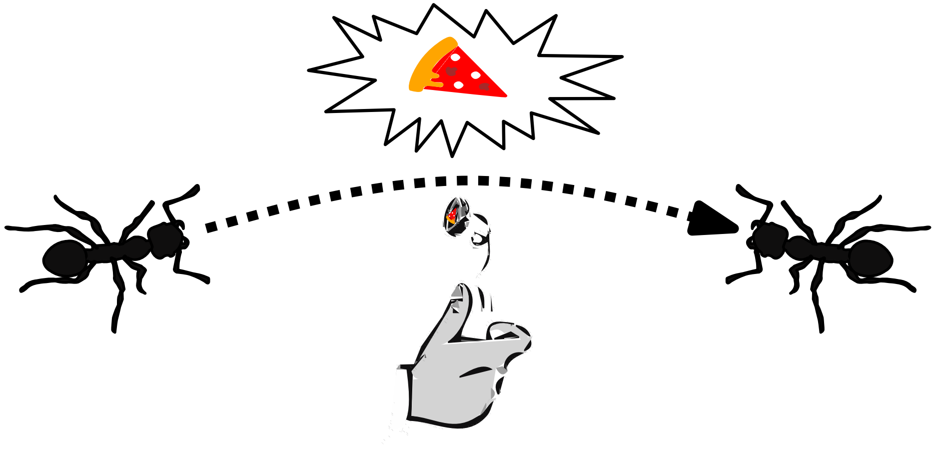


Noisy & Stochastic Interactions

Stochastic Interactions. At each round, each agent receives a message from another random agent.

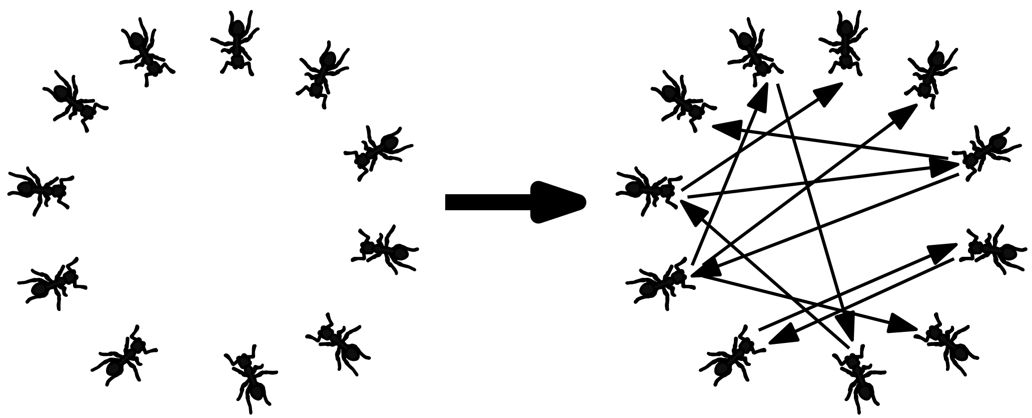


Noisy Communication. Before being received, each bit is flipped with probability $1/2 - \epsilon_n$.

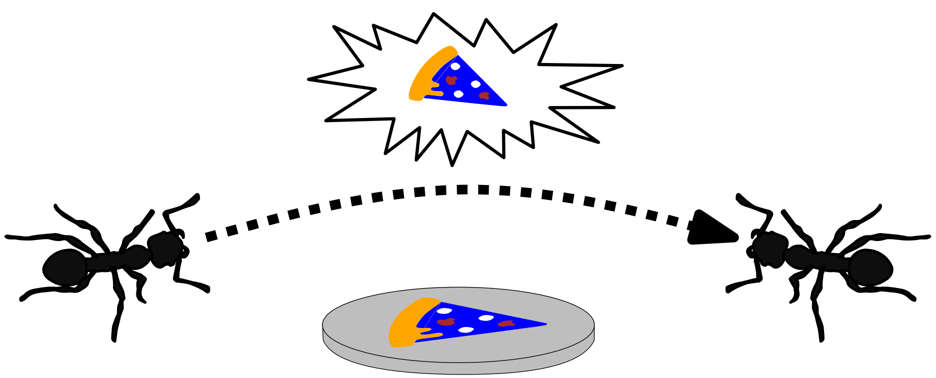


Noisy & Stochastic Interactions

Stochastic Interactions. At each round, each agent receives a message from another random agent.

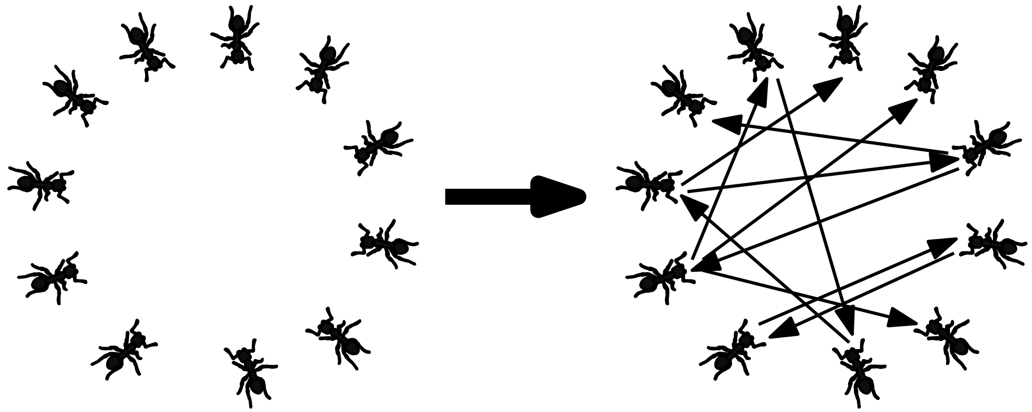


Noisy Communication. Before being received, each bit is flipped with probability $1/2 - \epsilon_n$.

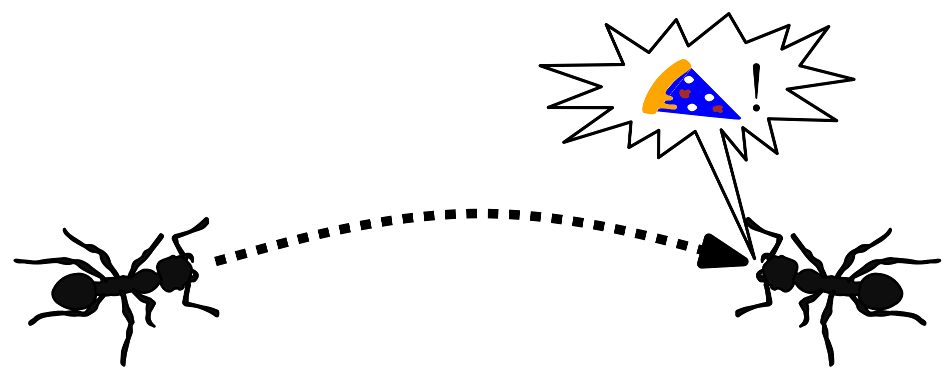


Noisy & Stochastic Interactions

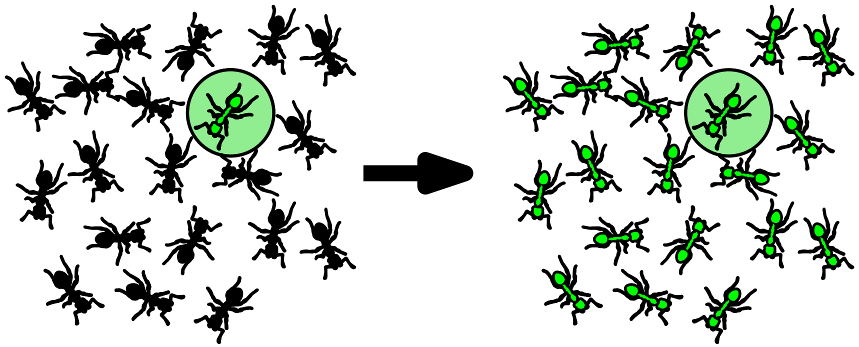
Stochastic Interactions.
At each round, each agent receives a message from another random agent.



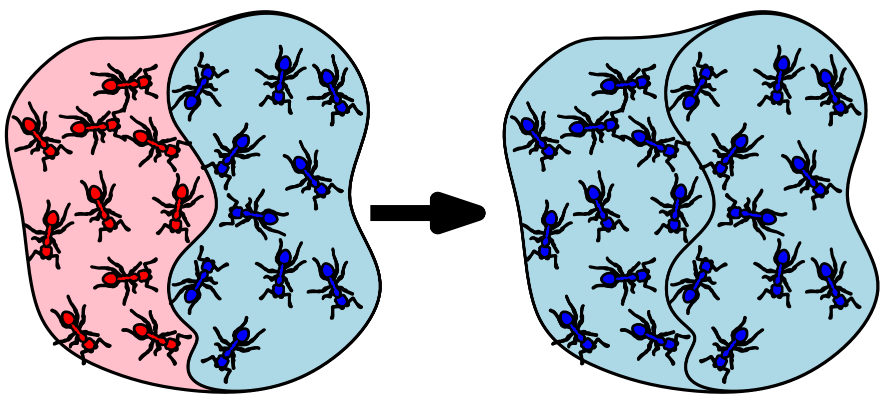
Noisy Communication.
Before being received, each bit is flipped with probability $1/2 - \epsilon_n$.



Noisy vs Noiseless Broadcast and Consensus



Broadcast. All nodes eventually receive the message of the source.

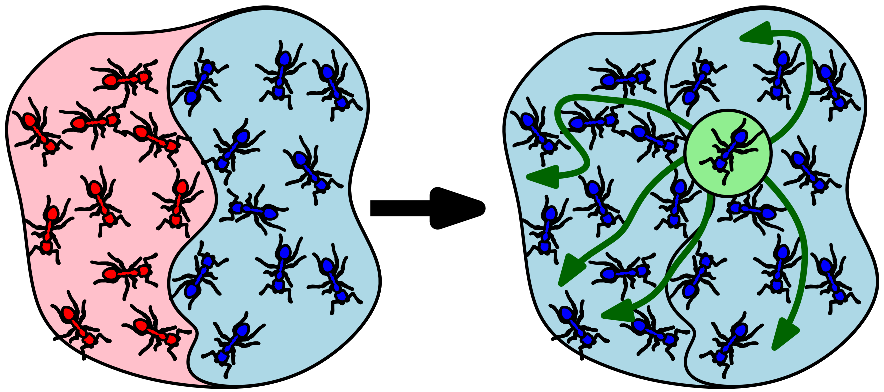


(Valid) Consensus. All nodes eventually support the value initially supported by one of them.

Reductions and Lower Bounds

Broadcast $\Longrightarrow$ Consensus

Noiseless Consensus
$\Longrightarrow$ **Noiseless**
(variant of) Broadcast

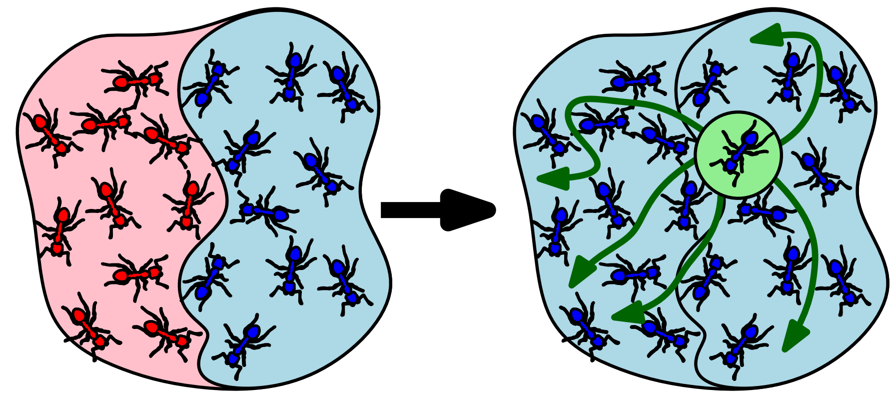


Noiseless Consensus and Broadcast are "*equivalent*"

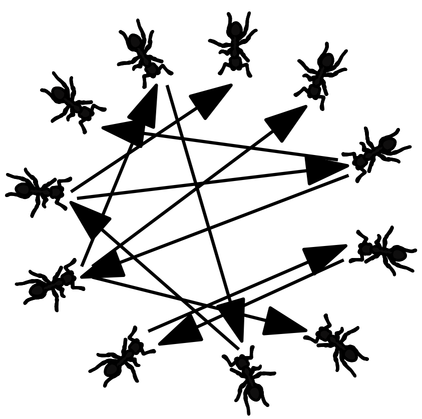
Reductions and Lower Bounds

Broadcast $\Longrightarrow$ Consensus

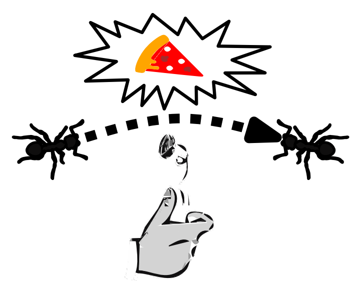
Noiseless Consensus
$\Longrightarrow$ **Noiseless**
(variant of) Broadcast



Noiseless Consensus and Broadcast are "*equivalent*"



$+$



$\Longrightarrow$

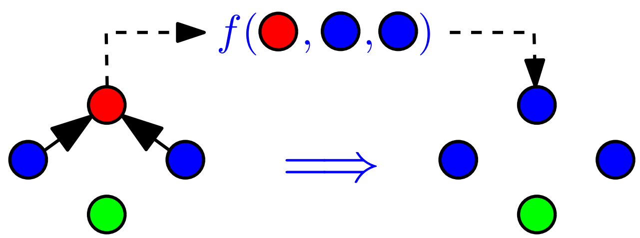
Noisy Consensus:
$\Theta(\frac{\log n}{\epsilon^2})$ rounds

Noisy Broadcast:
$\Theta(n \cdot \frac{\log n}{\epsilon^2})$ rounds

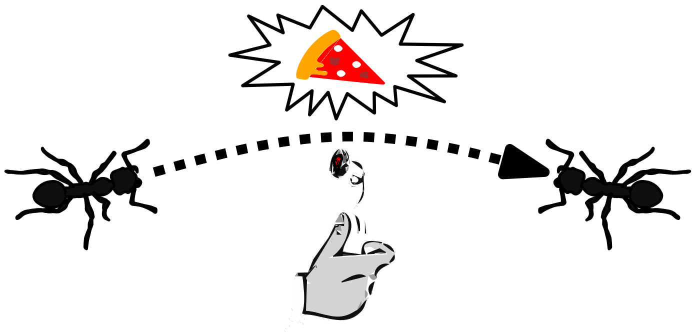
Noisy Broadcast is *exponentially harder*
than **Noisy** Consensus

Directions

- **Computational Dynamics.** Achieving simplicity in randomized distributed algorithms.



- **Biological Distributed Algorithms.** Going into biology and back, through the algorithmic lens (Natural Algorithms).



Thank You!

Come talk to me!