Computing through Dynamics: Principles for Distributed Coordination

Emanuele Natale

max planck institut informatik

INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

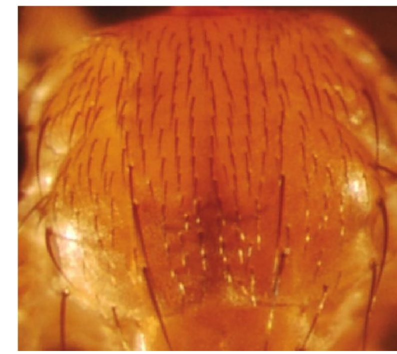
IRIF
Algorithms and Complexity seminar
14 December 2017, Paris

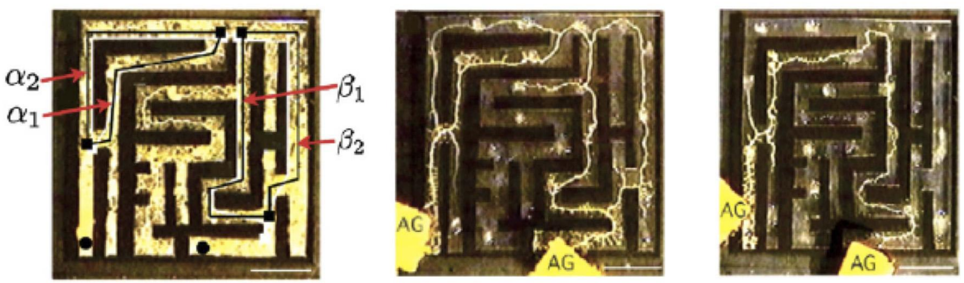
talk.enatale.name

Examples of “Natural” Algorithms

How birds of flocks synchronize their flight [Chazelle ’09]

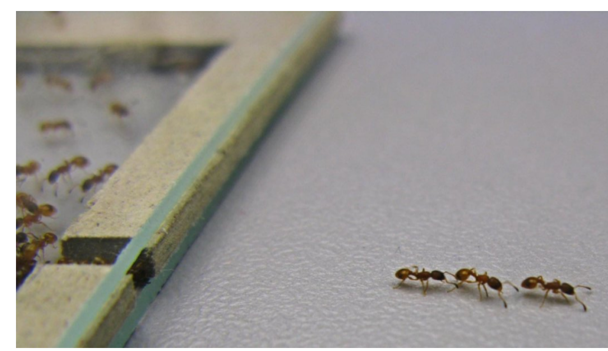
How are sensory organ precursor cells selected in a fly’s nervous system [AABHBB ’11]





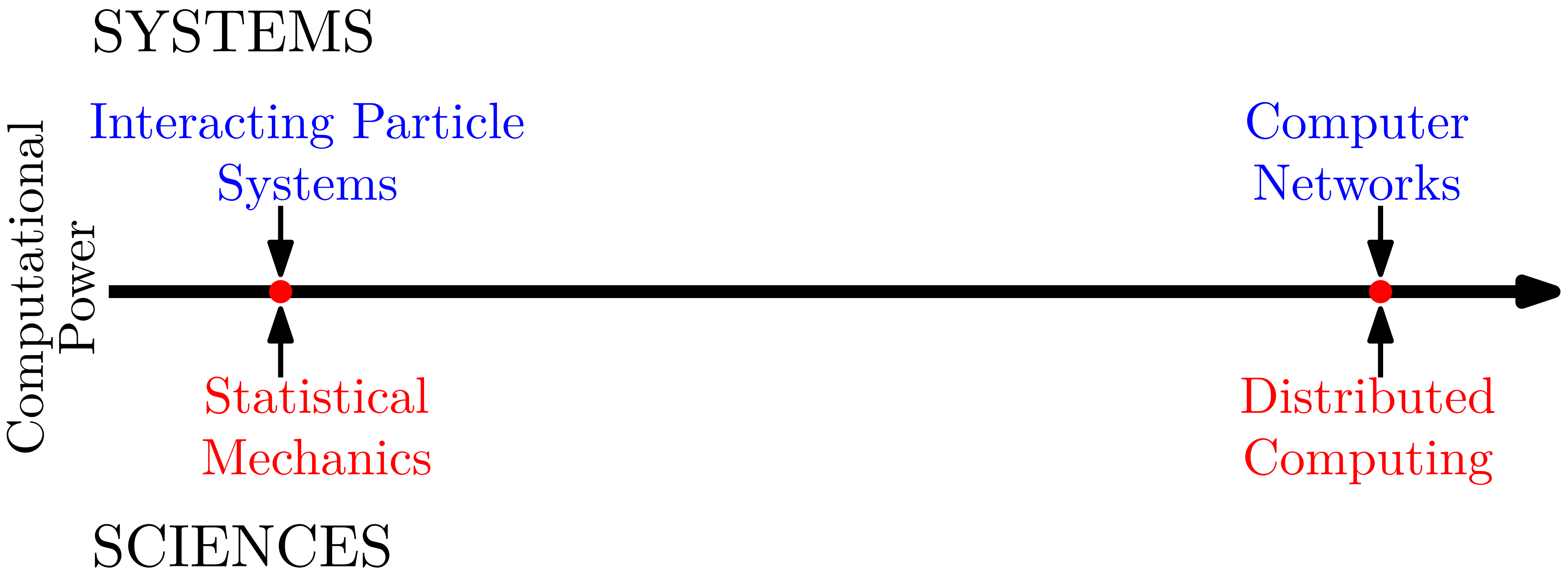
How Physarum polycephalum finds shortest paths [BBDKM ’14]

How do ants decide where to relocate their nest? [GMRL ’15]



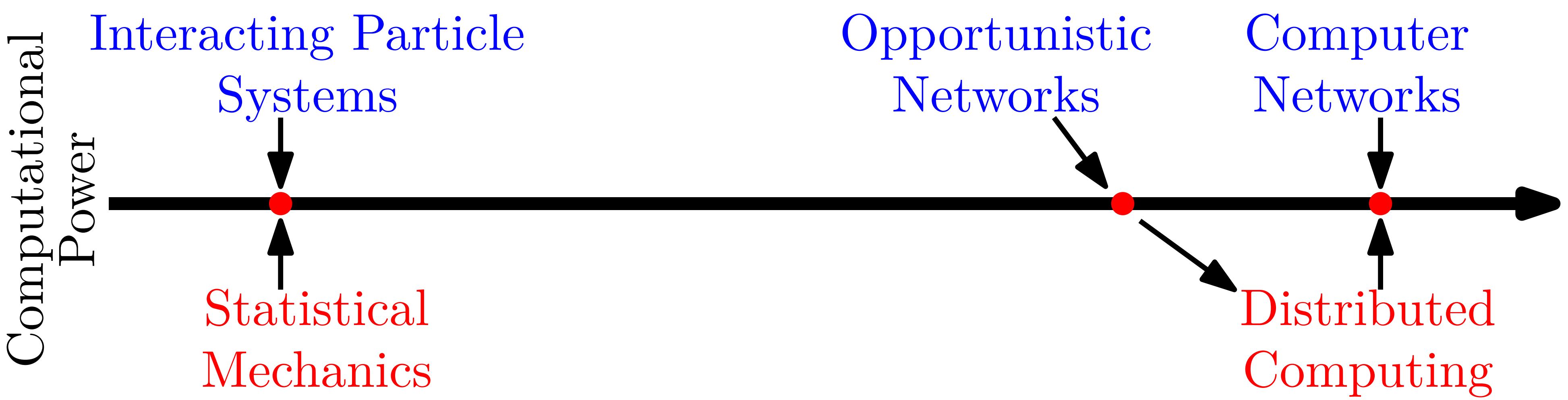
How ants perform collective navigattion [FHBGKKF ’16]

What can *Simple* Systems do?



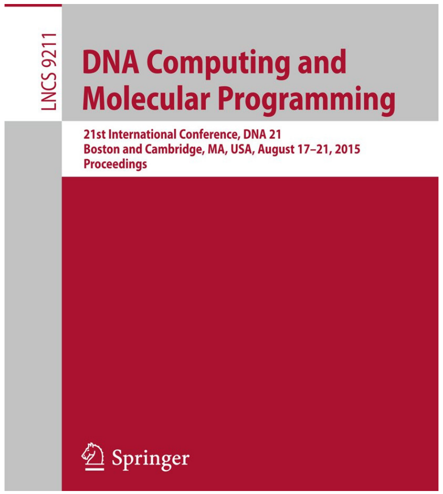
What can *Simple* Systems do?

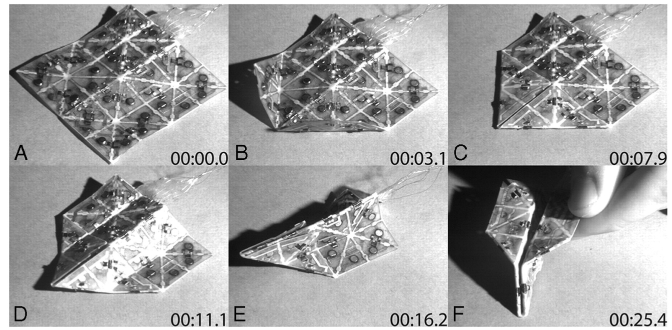
SYSTEMS

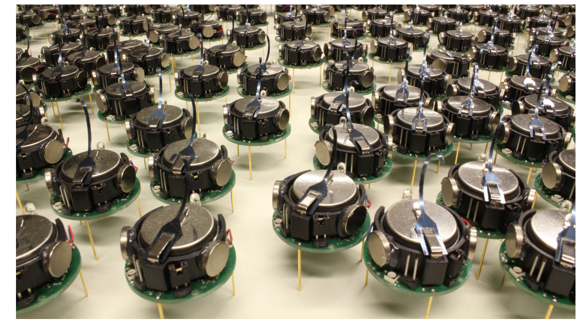


SCIENCES

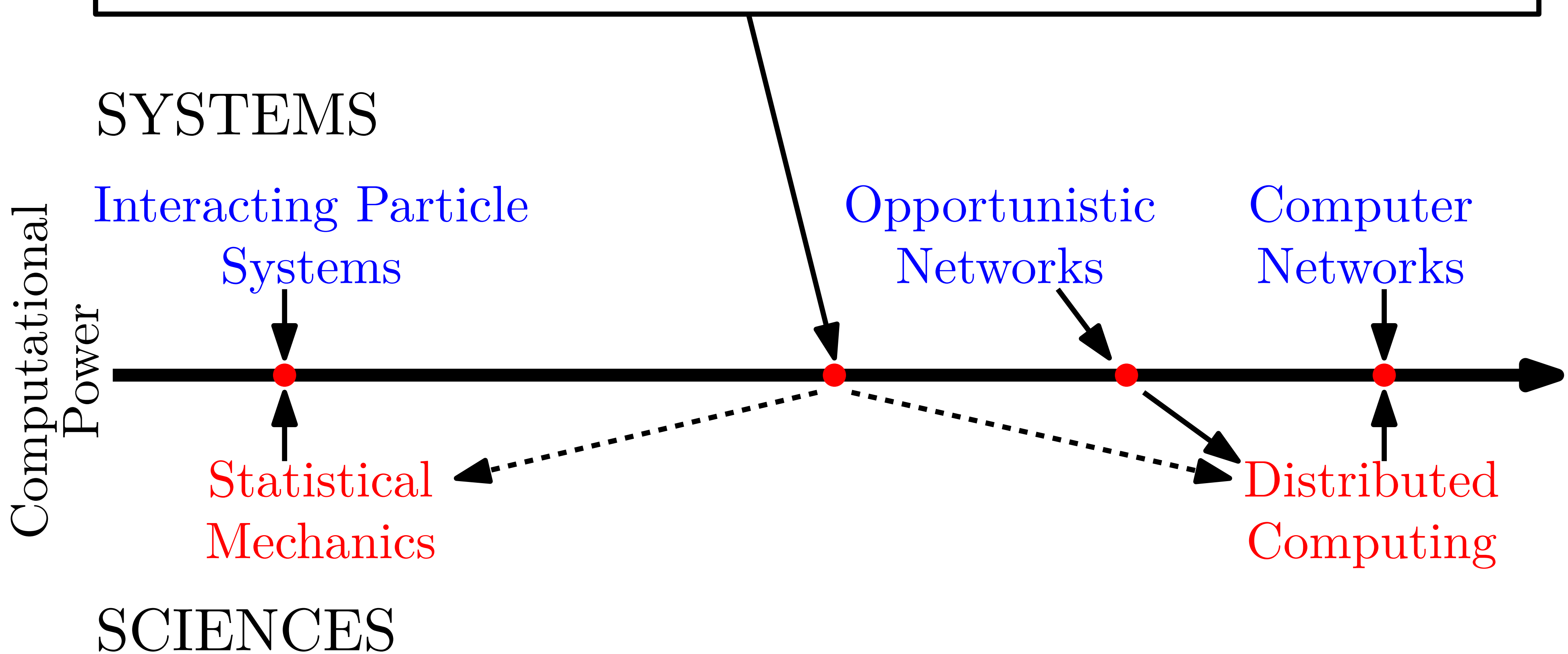
What can *Simple* Systems do?







DNA/Molecular Computing, Programmable Matter, Swarms of Simple Robots



What can *Simple* Systems do?

Schools of fish
[Sumpter et al. '08]

Insects colonies
[Franks et al. '02]

Flocks of birds
[Ben-Shahar et al. '10]

Biological Systems

SYSTEMS

Interacting Particle Systems

Opportunistic Networks

Computer Networks

Computational Power

Statistical Mechanics

Distributed Computing

SCIENCES

Unstructured Communication Models

Requirements:

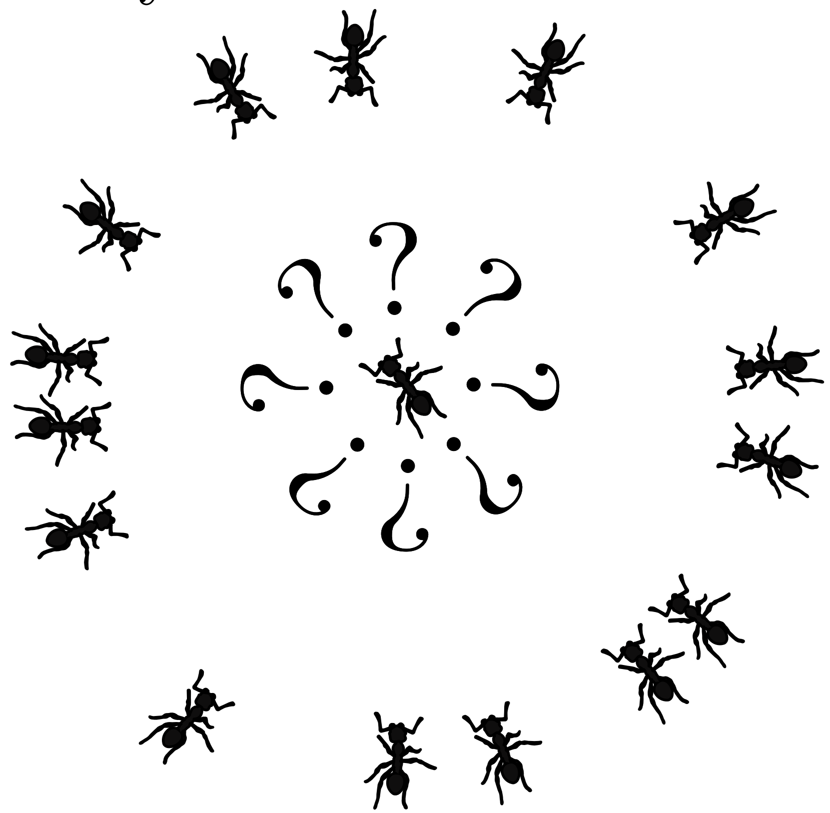
- Chaotic
- Anonymous
- Parsimonious
- Uni-directional (Passive/Active)
- Noisy

Unstructured Communication Models

Requirements:

- Chaotic
- Anonymous
- Parsimonious
- Uni-directional (Passive/Active)
- Noisy

$\mathcal{PULL}(h, \ell)$ **model [1]**: at each round each agent can *observe* h other agents chosen independently and uniformly at random, and *shows* ℓ *bits* to her observers.



[1] A. Demers et al., "Epidemic algorithms for replicated database maintenance," in Proc. of 6th ACM PODC, 1987.

Unstructured Communication Models

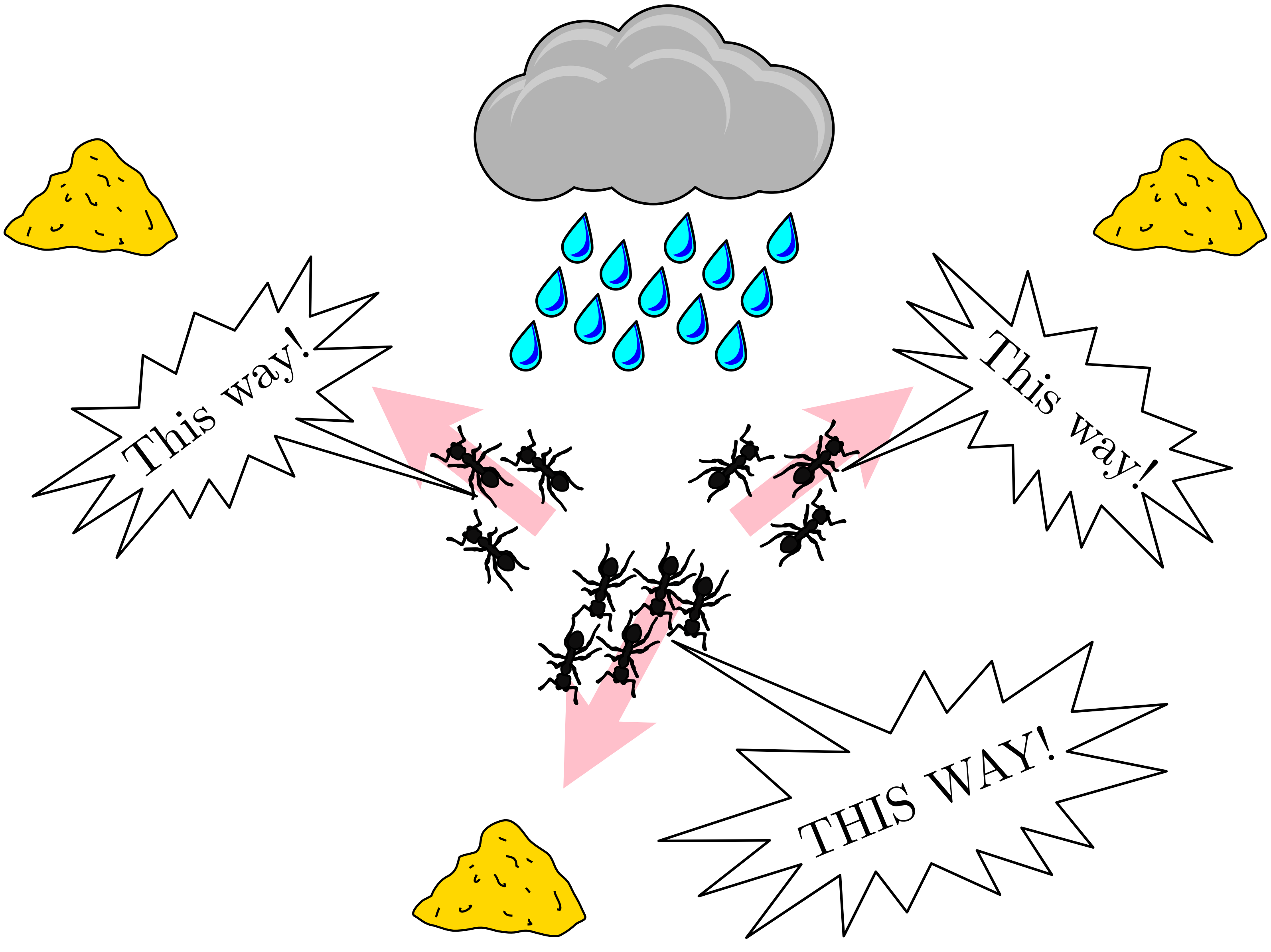
Requirements:

- Chaotic
- Anonymous
- Parsimonious
- Uni-directional (Passive/Active)
- Noisy

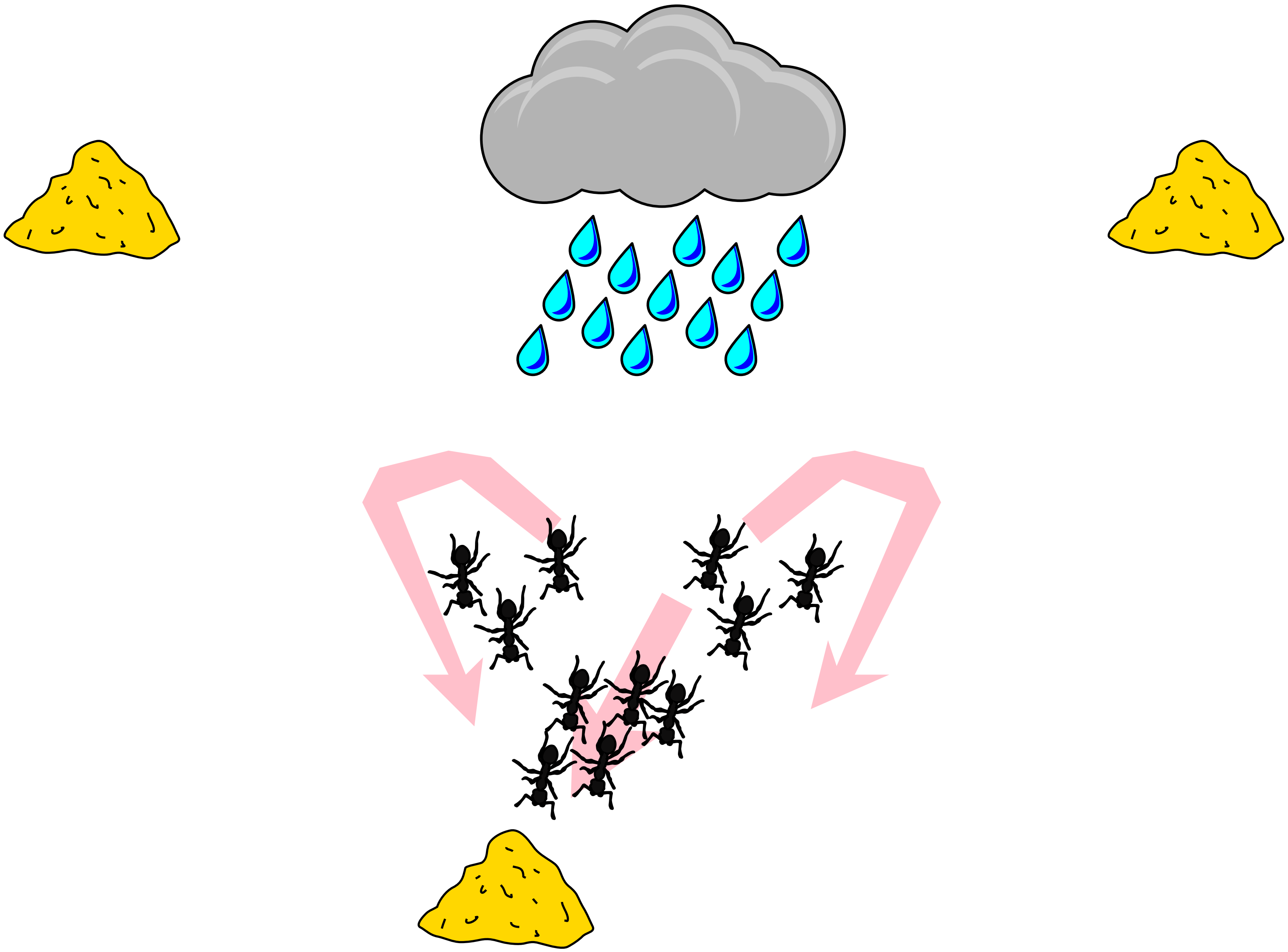
$\mathcal{PULL}(h,\ell)$ **model [1]**: at each round each agent can *observe* h other agents chosen independently and uniformly at random, and *shows* ℓ *bits* to her observers.

[1] A. Demers et al., "Epidemic algorithms for replicated database maintenance," in Proc. of 6th ACM PODC, 1987.

Natural Algorithms for Consensus



Natural Algorithms for Consensus



Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.

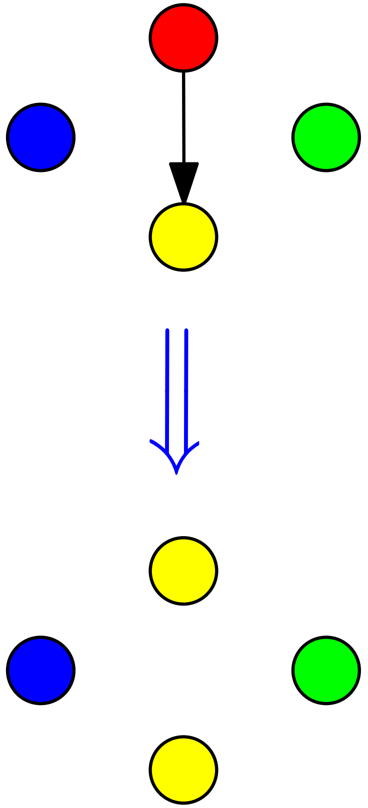
Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.

Examples of Dynamics

- Voter dynamics



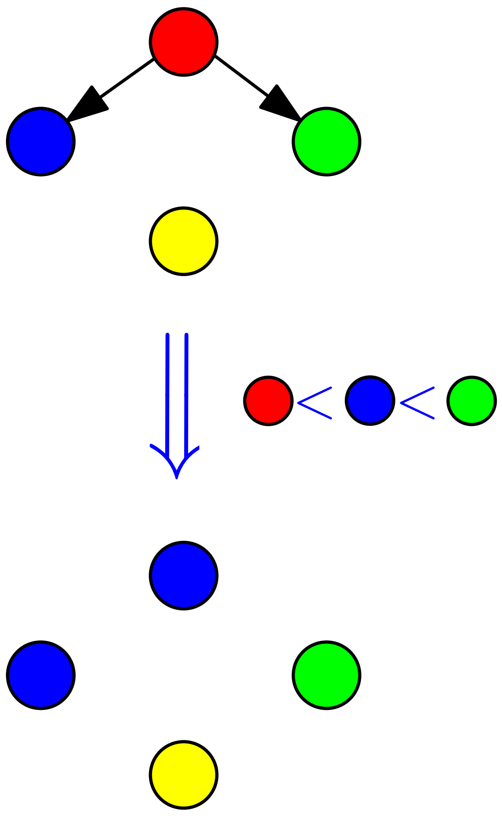
Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.

Examples of Dynamics

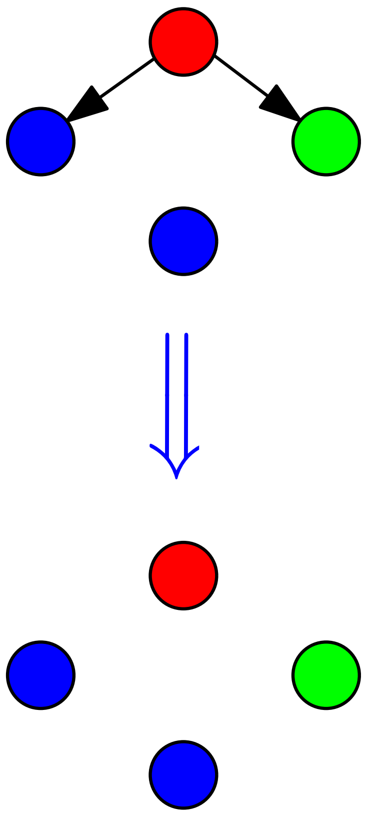
- Voter dynamics
- 2-Median dynamics



Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.



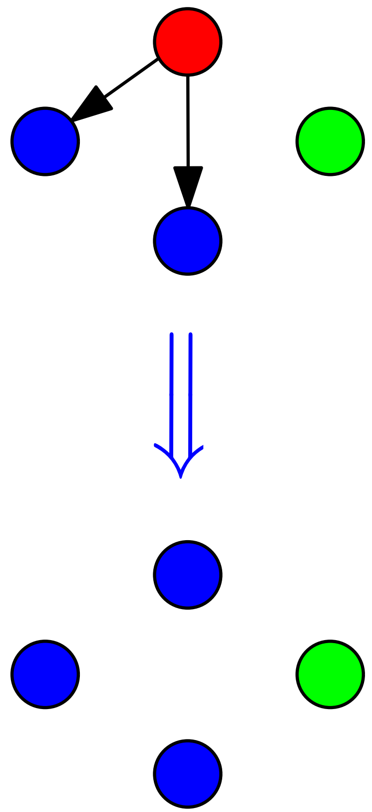
Examples of Dynamics

- Voter dynamics
- 2-Median dynamics
- 2-Choice dynamics

Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.



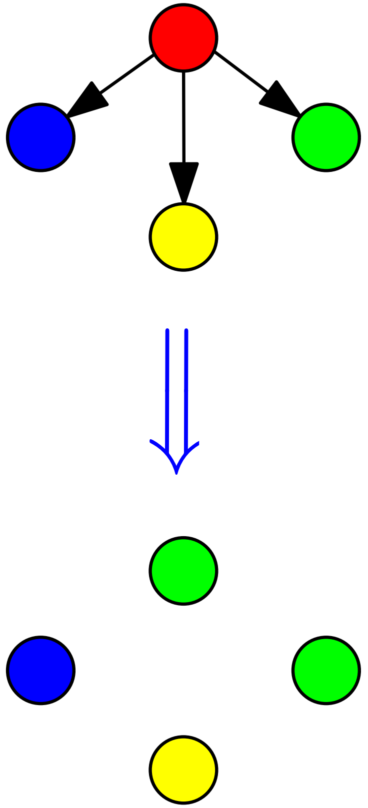
Examples of Dynamics

- Voter dynamics
- 2-Median dynamics
- 2-Choice dynamics

Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.



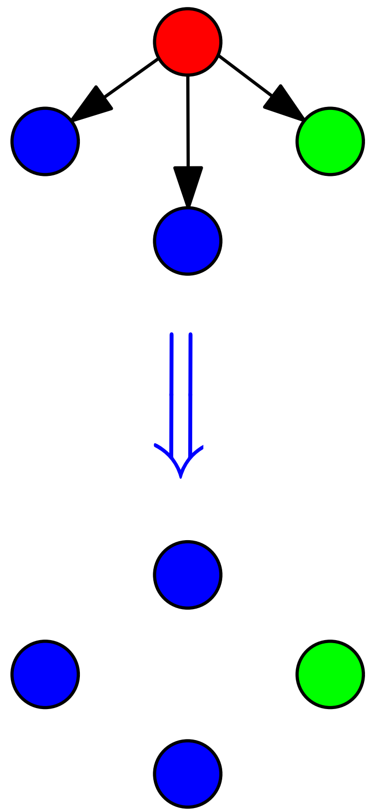
Examples of Dynamics

- Voter dynamics
- 2-Median dynamics
- 2-Choice dynamics
- 3-Majority dynamics

Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.



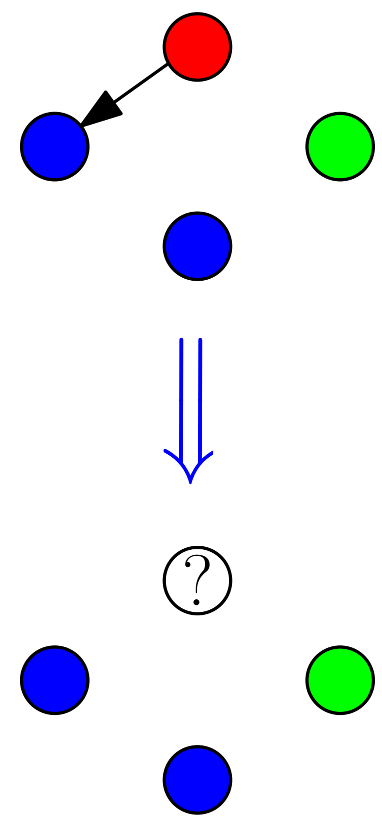
Examples of Dynamics

- Voter dynamics
- 2-Median dynamics
- 2-Choice dynamics
- 3-Majority dynamics

Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.



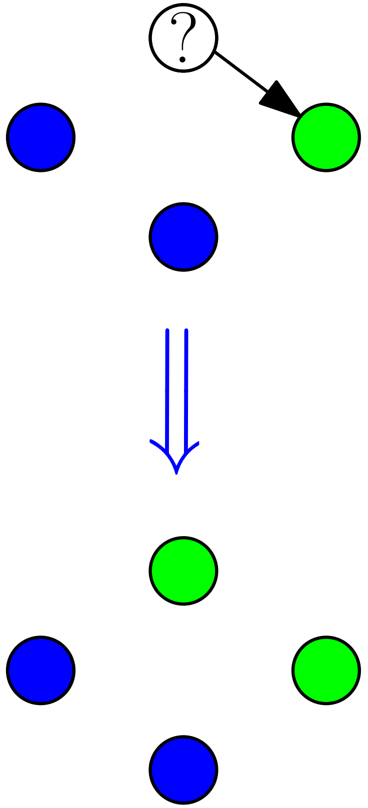
Examples of Dynamics

- Voter dynamics
- 2-Median dynamics
- 2-Choice dynamics
- 3-Majority dynamics
- Undecided-State dynamics

Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.



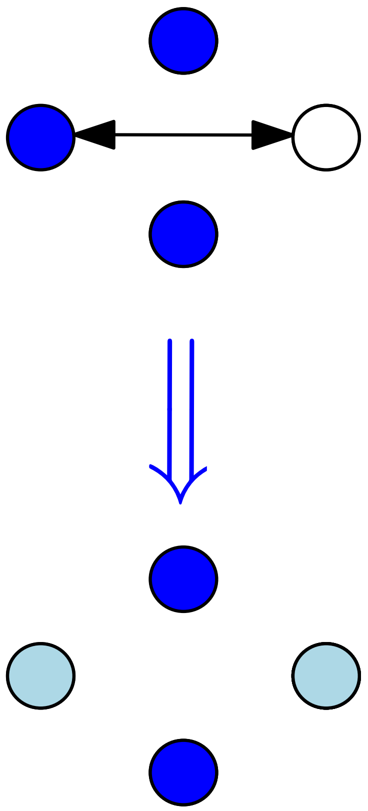
Examples of Dynamics

- Voter dynamics
- 2-Median dynamics
- 2-Choice dynamics
- 3-Majority dynamics
- Undecided-State dynamics

Model: ✓. Problem: ✓. Algorithms: Dynamics

(informal)

Very simple distributed algorithms: For every graph $G = (V, E)$, agent $u \in V$ and round $t \in \mathbb{N}$, states are updated according to fixed rule $f(\sigma(u), \sigma(S))$ of current state $\sigma(u)$ and symmetric function of states $\sigma(S)$ of a random sample S of neighbors.



Examples of Dynamics

- Voter dynamics
- 2-Median dynamics
- 2-Choice dynamics
- 3-Majority dynamics
- Undecided-State dynamics
- Averaging dynamics (asynchronous)

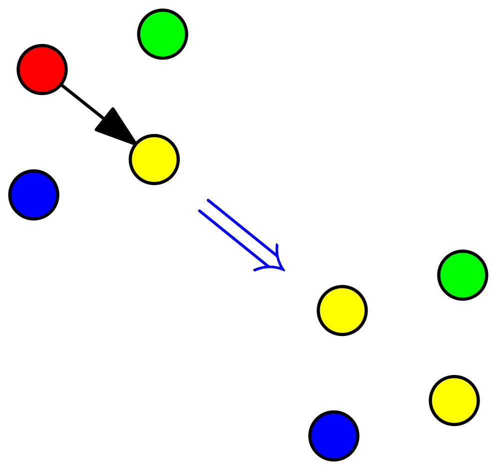
We ask 4 Questions

- Can dynamics be used to perform algorithmically-interesting tasks?
- What are the minimal model requirements which allow effective information spreading?
- Can we develop a *comparative* approach to dynamics?
- Can dynamics solve problems which are *non-trivial* even in centralized setting?

The Simplest One: Voter Dynamics

Widely studied process since ’70s.

Martingale argument shows probability color wins $\propto$ its initial volume.

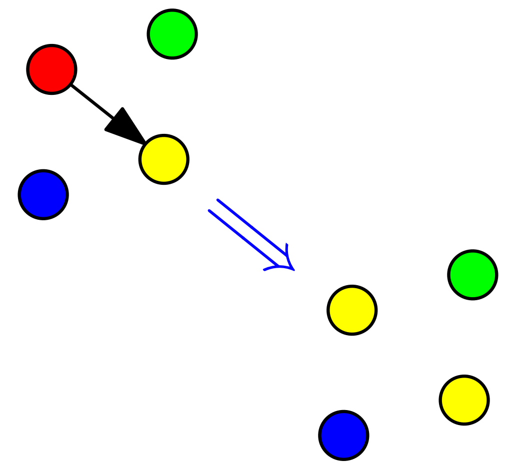


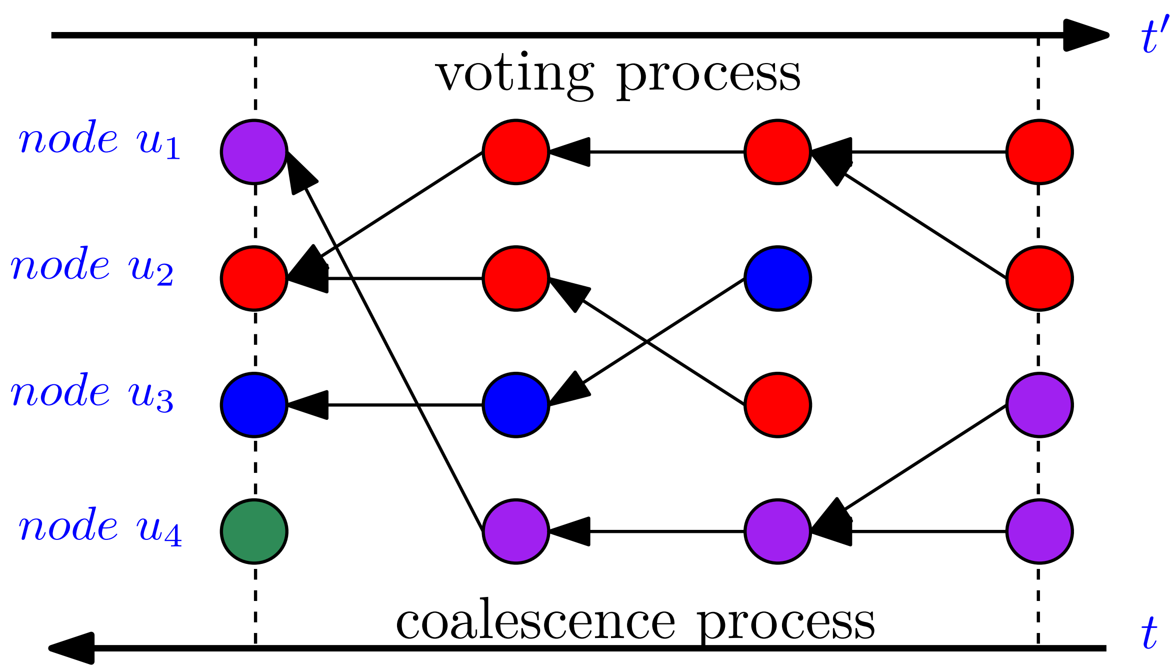
The Simplest One: Voter Dynamics

Widely studied process since '70s.

Martingale argument shows probability color wins $\propto$ its initial volume.

Polynomial convergence time, even on good expanders.





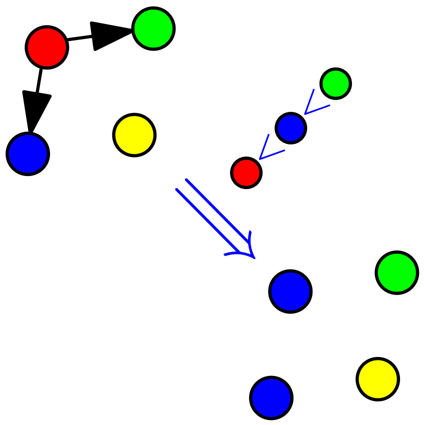
A random walk starts at each node. When two walkers meet, they *coalesce*. This process, observed *backwards*, is distributed like the Voter dynamics.

Question 1/4

Can dynamics, other than the few studied in physics, be rigorously analyzed and used to perform algorithmically-interesting tasks?

The Power of Dynamics: Plurality Consensus

Computing the Median

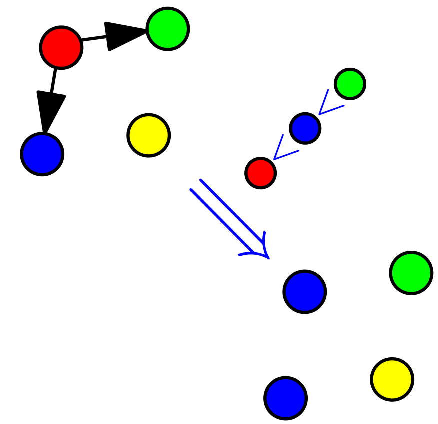


2-Median dynamics [1]. Converge to $\mathcal{O}(\sqrt{n \log n})$ approximation of median of system in $\mathcal{O}(\log n)$ rounds w.h.p., even if $\mathcal{O}(\sqrt{n})$ states are arbitrarily changed at each round ($\mathcal{O}(\sqrt{n})$-bounded adversary).

[1] B. Doerr, Leslie A. Goldberg, L. Minder, T. Sauerwald, and C. Scheideler, "Stabilizing Consensus with the Power of Two Choices," in Proc. of 23rd ACM SPAA, 2011.
[2] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, R. Silvestri, and L. Trevisan, "Simple dynamics for plurality consensus," Distrib. Comput., pp. 1–14, Nov. 2016.
[3] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, "Stabilizing Consensus with Many Opinions," in Proc. of 27th ACM-SIAM SODA, 2016.

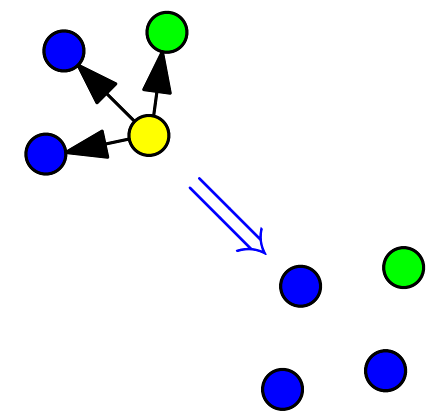
The Power of Dynamics: Plurality Consensus

Computing the Median



2-Median dynamics [1]. Converge to $\mathcal{O}(\sqrt{n \log n})$ approximation of median of system in $\mathcal{O}(\log n)$ rounds w.h.p., even if $\mathcal{O}(\sqrt{n})$ states are arbitrarily changed at each round ($\mathcal{O}(\sqrt{n})$-bounded adversary).

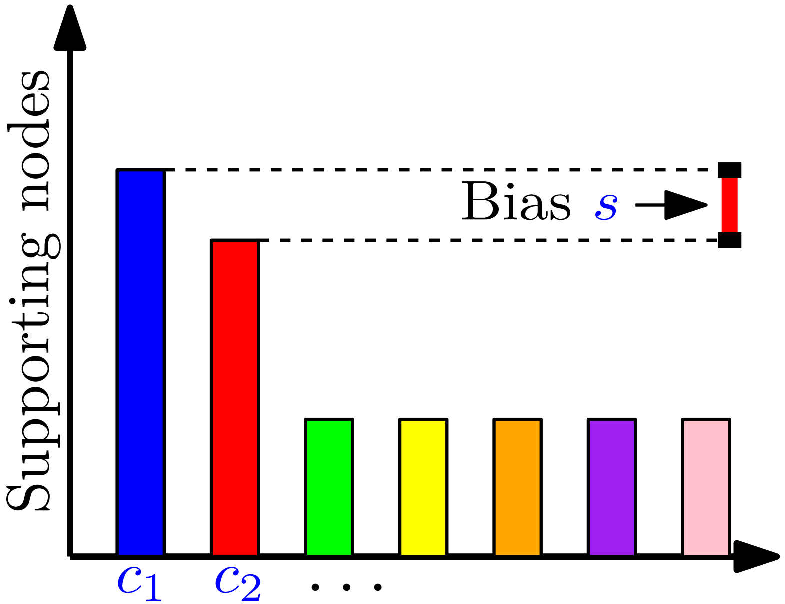
Computing the Majority

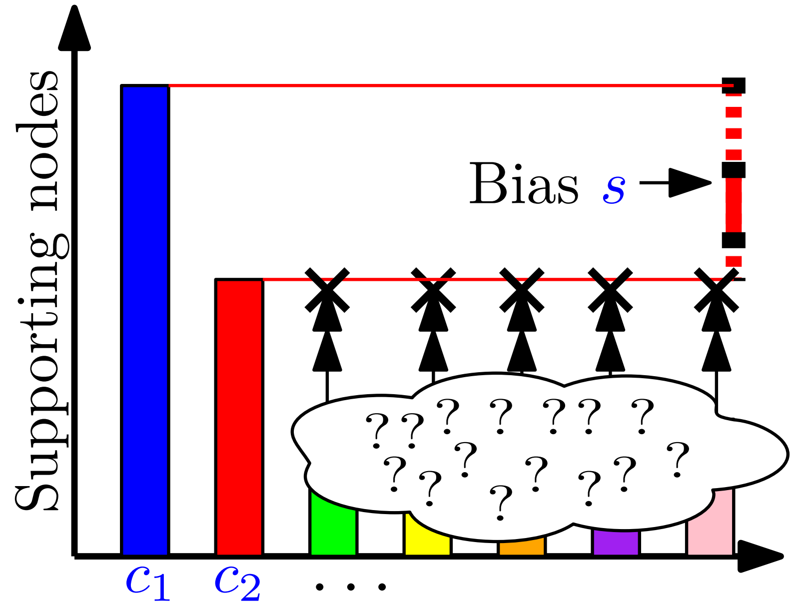


3-Majority dynamics [2,3]. If plurality has **bias** $\mathcal{O}(\sqrt{kn \log n})$, converges to it in $\mathcal{O}(k \log n)$ rounds w.h.p., even against $o(\sqrt{n/k})$-bounded adversary.
Without bias, converges in $\text{poly}(k)$.
h-majority converges in $\Omega(k/h^2)$.

[1] B. Doerr, Leslie A. Goldberg, L. Minder, T. Sauerwald, and C. Scheideler, "Stabilizing Consensus with the Power of Two Choices," in Proc. of 23rd ACM SPAA, 2011.
[2] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, R. Silvestri, and L. Trevisan, "Simple dynamics for plurality consensus," Distrib. Comput., pp. 1–14, Nov. 2016.
[3] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, "Stabilizing Consensus with Many Opinions," in Proc. of 27th ACM-SIAM SODA, 2016.

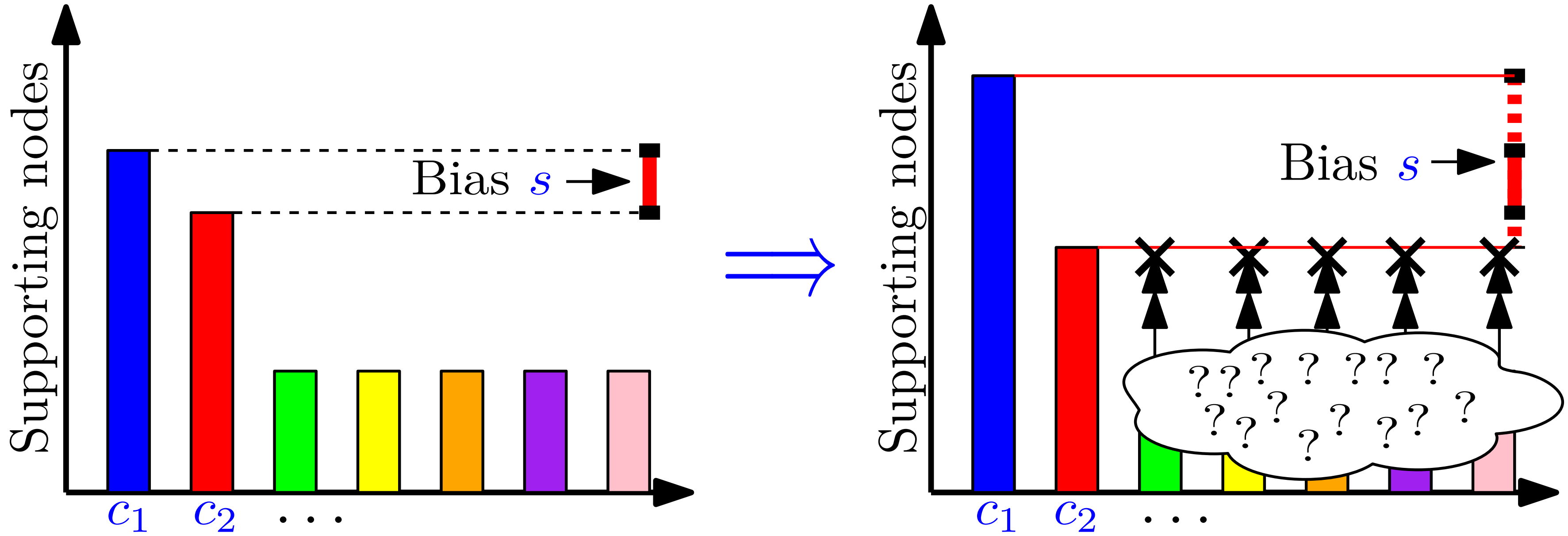
Breaking Symmetry in Dynamics





Previous results crucially rely on amplifying an initial *bias*

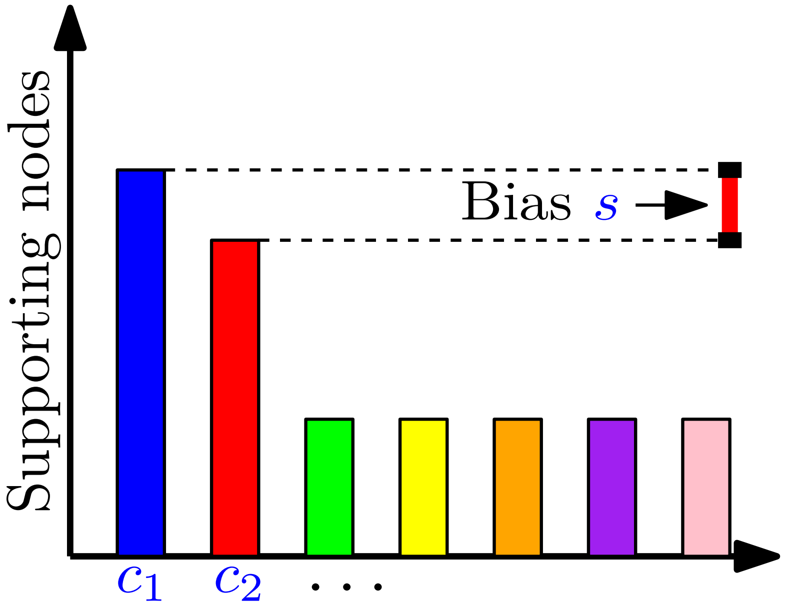
Breaking Symmetry in Dynamics



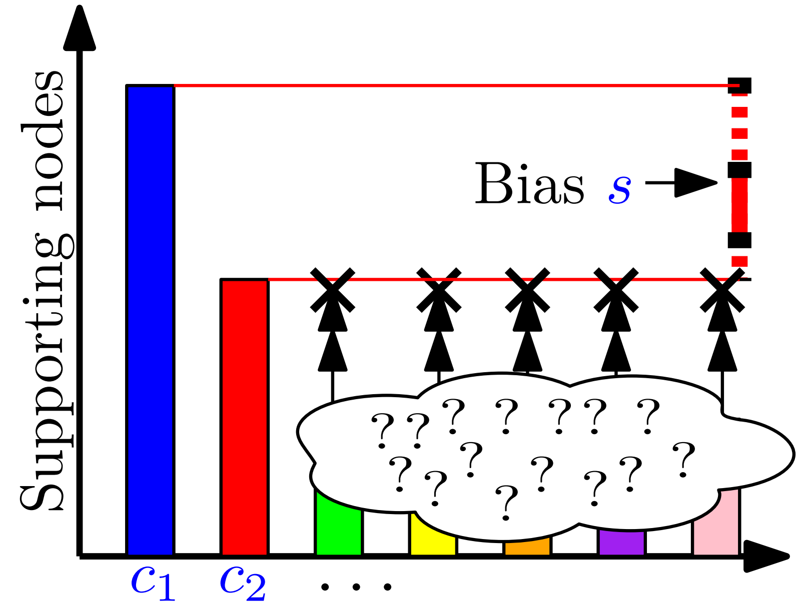
Previous results crucially rely on amplifying an initial *bias*

What if we have no bias?

Breaking Symmetry in Dynamics

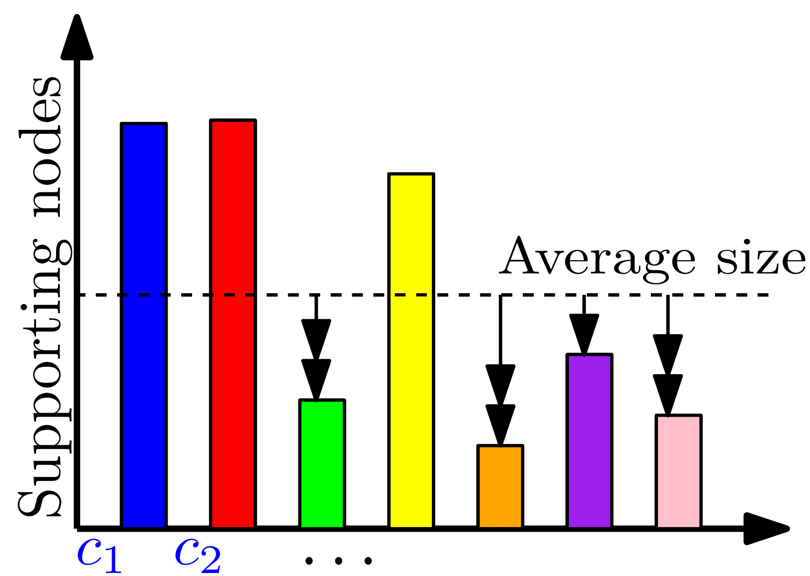


$\Rightarrow$



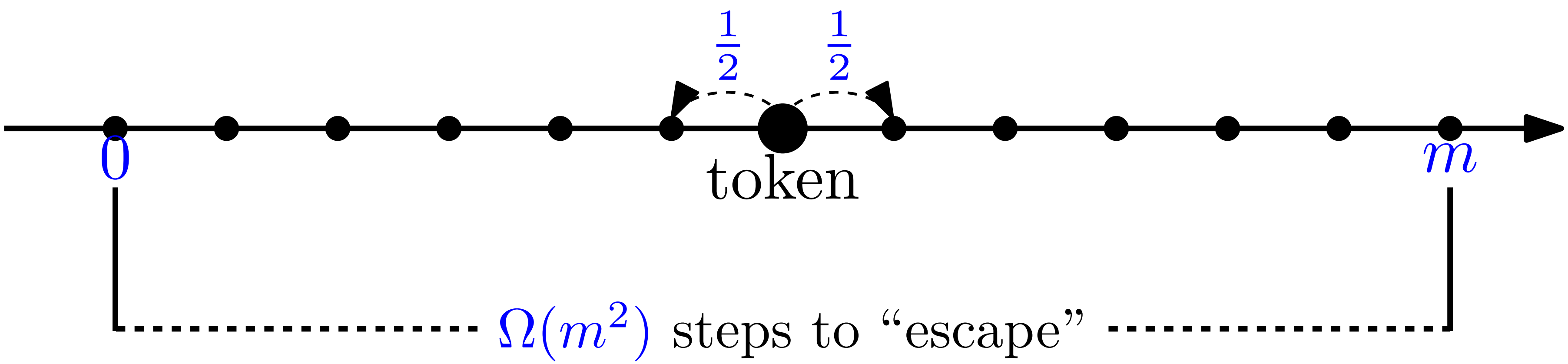
Previous results crucially rely on amplifying an initial *bias*

What if we have no bias?



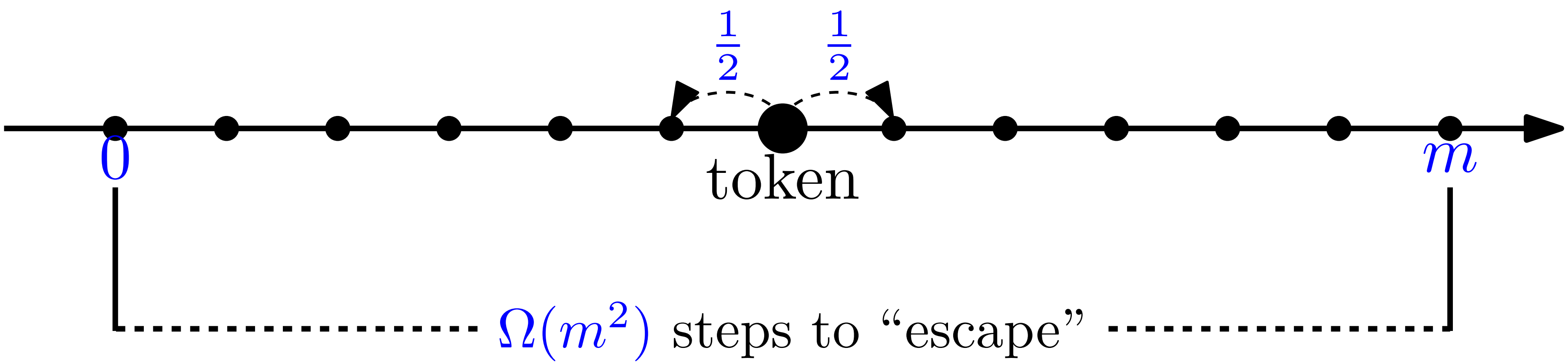
Breaking Symmetry in Dynamics

Simple symmetric random walk:

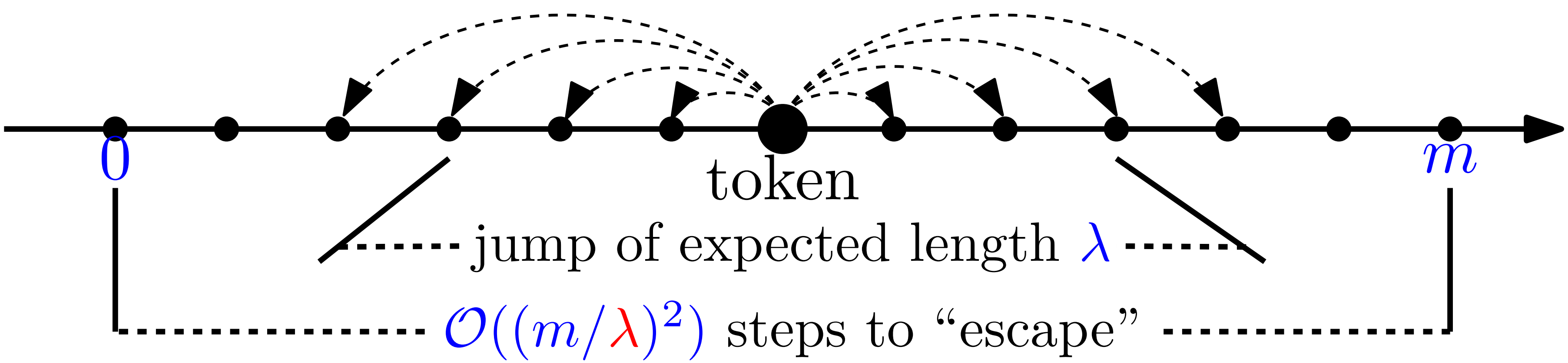


Breaking Symmetry in Dynamics

Simple symmetric random walk:



Stationary-in-expectation random walk:



Breaking Symmetry in Dynamics

Folklore Lemma [1].
$\{X_t\}_t$ a Markov chain with finite state space Ω,
$f : \Omega \to \mathbf{N}$, $Y_t = f(X_t)$,
$m \in [n]$ a "target value" and
$\tau = \inf\{t \in \mathbb{N} \, : \, Y_t \geq m\}$.
If $\forall x \in \Omega$ with $f(x) \leq m - 1$, it holds

1. *Positive drift*: $\mathbf{E}[Y_{t+1} \,|\, X_t = x] \geqslant f(x) + \psi$ $(\psi > 0)$,
2. *Bounded jumps*: $\Pr\{Y_\tau \geq \alpha m\} \leq \alpha m / n \;\; (\alpha > 1)$,

then

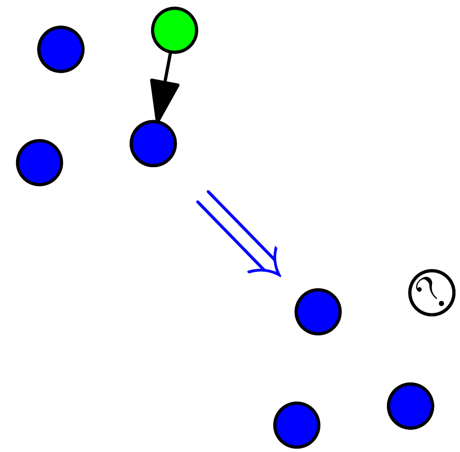
$$\mathbf{E}[\tau] \leq 2\alpha \frac{m}{\psi}.$$

[1] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, "Stabilizing Consensus with Many Opinions," in Proc. of 27th ACM-SIAM SODA, 2016.

A Global Measure of Bias

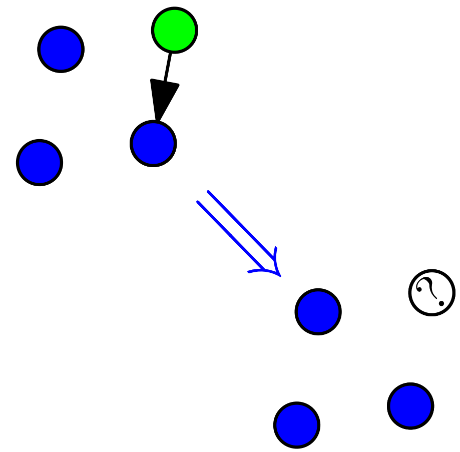
3-Majority converges in $\tilde{\Theta}(k)$ rounds...

Undecided-State dynamics [1]. If majority/second-majority $(c_{maj}/c_{2^{nd}maj})$ is at least $1+\epsilon$, system converges to plurality within $\tilde{\Theta}(\mathrm{md}(\mathbf{c}))$ rounds w.h.p.



[1] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and R. Silvestri, "Plurality Consensus in the Gossip Model," in Proc. of 26th ACM-SIAM SODA, 2015.

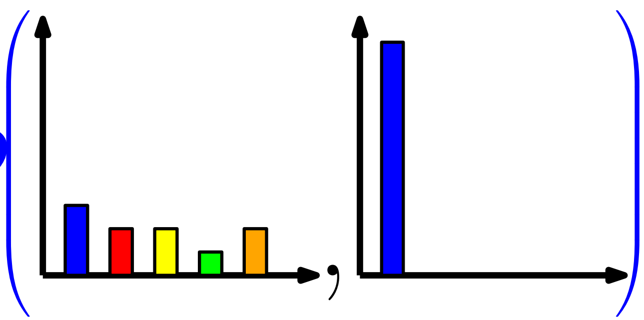
A Global Measure of Bias



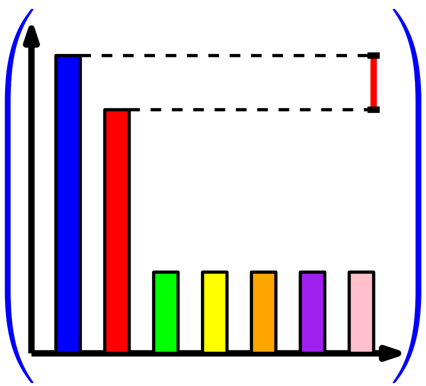
3-Majority converges in $\tilde{\Theta}(k)$ rounds...

Undecided-State dynamics [1]. If majority/second-majority ($c_{maj}/c_{2^{nd}maj}$) is at least $1+\epsilon$, system converges to plurality within $\tilde{\Theta}(\mathrm{md}(\mathbf{c}))$ rounds w.h.p.

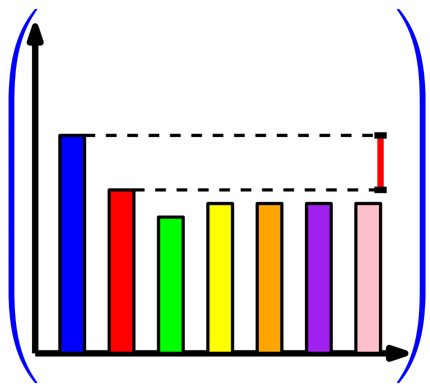
$$\mathrm{md}(\mathbf{c}^{(\mathbf{0})}) := \sum_{i=1}^{k} \left(\frac{c_i^{(0)}}{c_{maj}^{(0)}} \right)^2 = 1 + \mathcal{D}$$



$$1 \leq \mathrm{md}$$



$$\ll \mathrm{md}$$

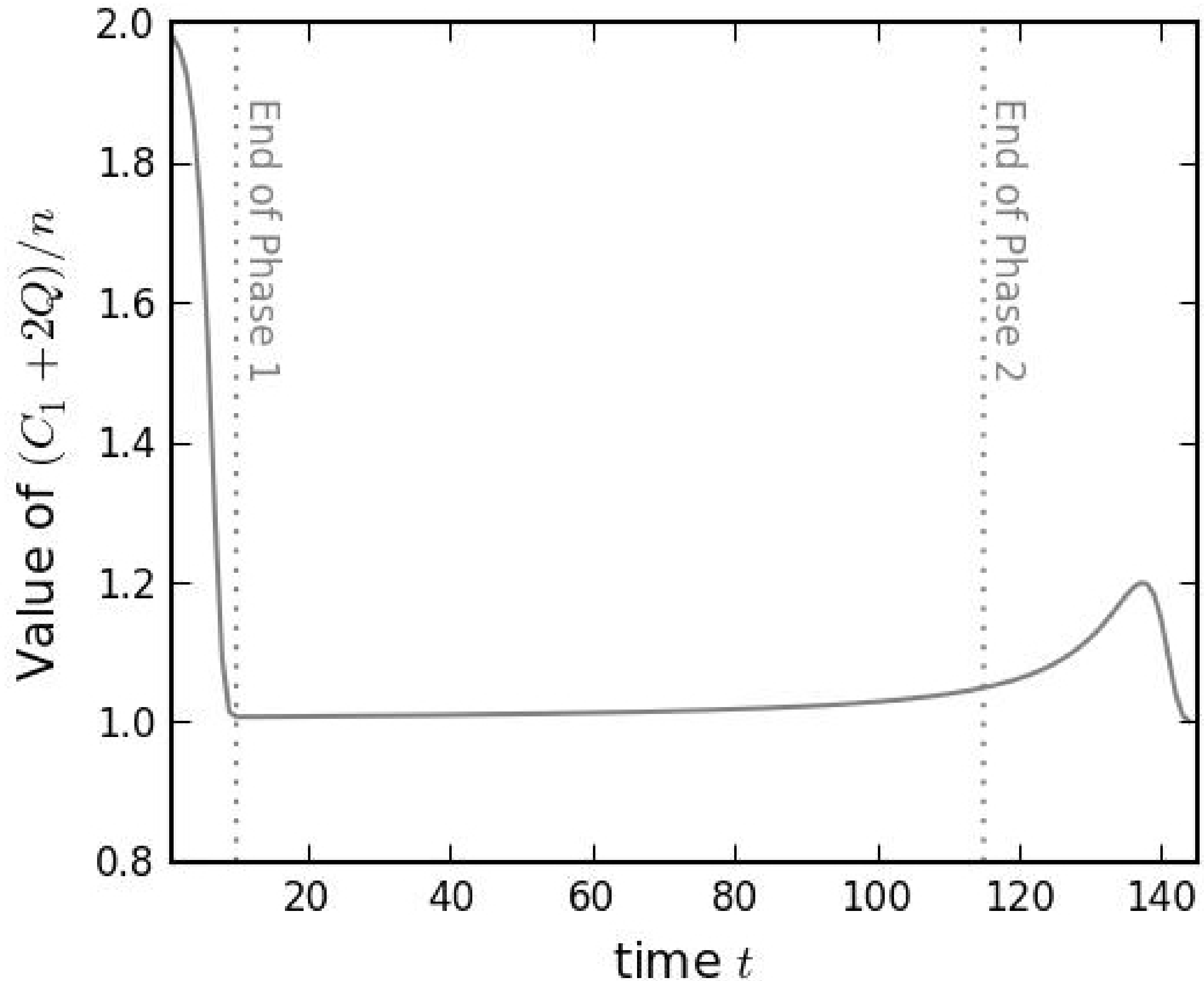


$$\leq k$$

[1] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and R. Silvestri, "Plurality Consensus in the Gossip Model," in Proc. of 26th ACM-SIAM SODA, 2015.

Evolution of Undecided-State Dynamics

Simulation of the growth factor:



$$\mathbf{E}\left[c_i^{(t+1)} \,\middle|\, \mathbf{c}^{(t)}\right]$$

$$= c_i^{(t)} \cdot \underbrace{\frac{c_i^{(t)} + 2q^{(t)}}{n}}_{\text{Growth factor}}$$

$$1 + \frac{\left(n - 2q^{(t)} - c_1^{(t)}\right)^2}{n^2} + \frac{2\left(\sum_{i=1}^{k} \frac{c_i^{(t)}}{c_1^{(t)}} - \mathrm{md}(\mathbf{c}^{(t)})\right) \cdot (c_1)^2}{n^2}$$

From Consensus to Information Spreading

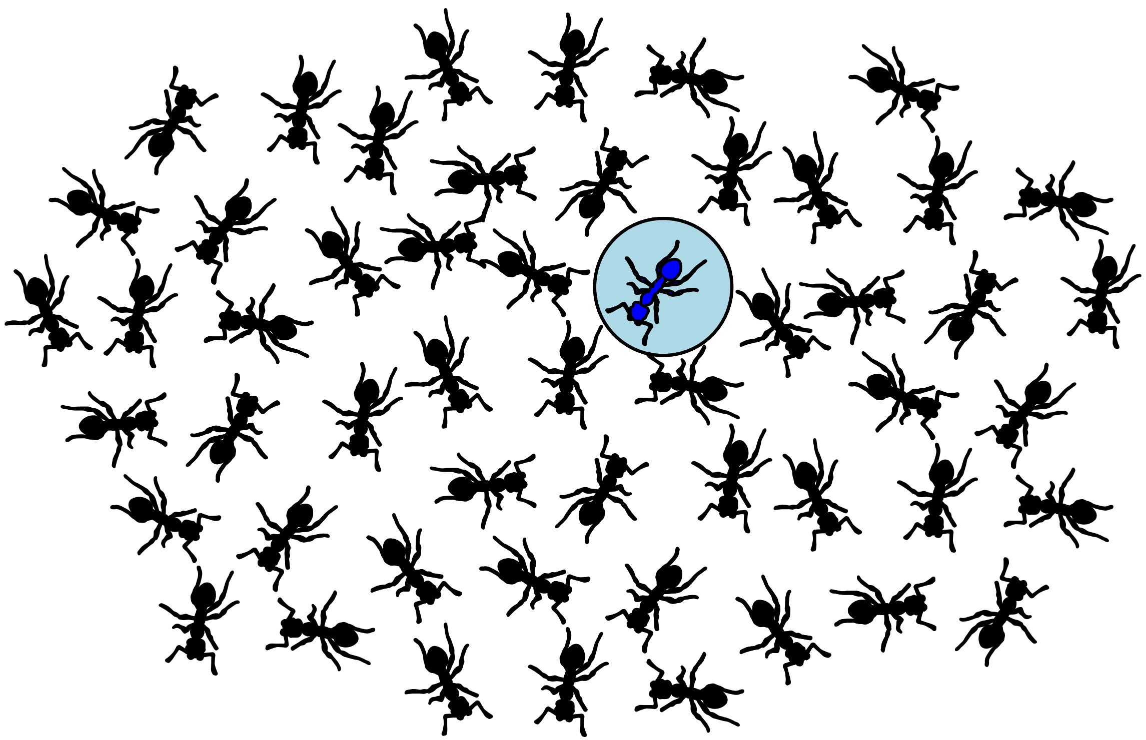
From Consensus to Information Spreading

Question 2/4

What are the minimal model requirements with respect to achieving basic information dissemination tasks under conditions of increased uncertainty?

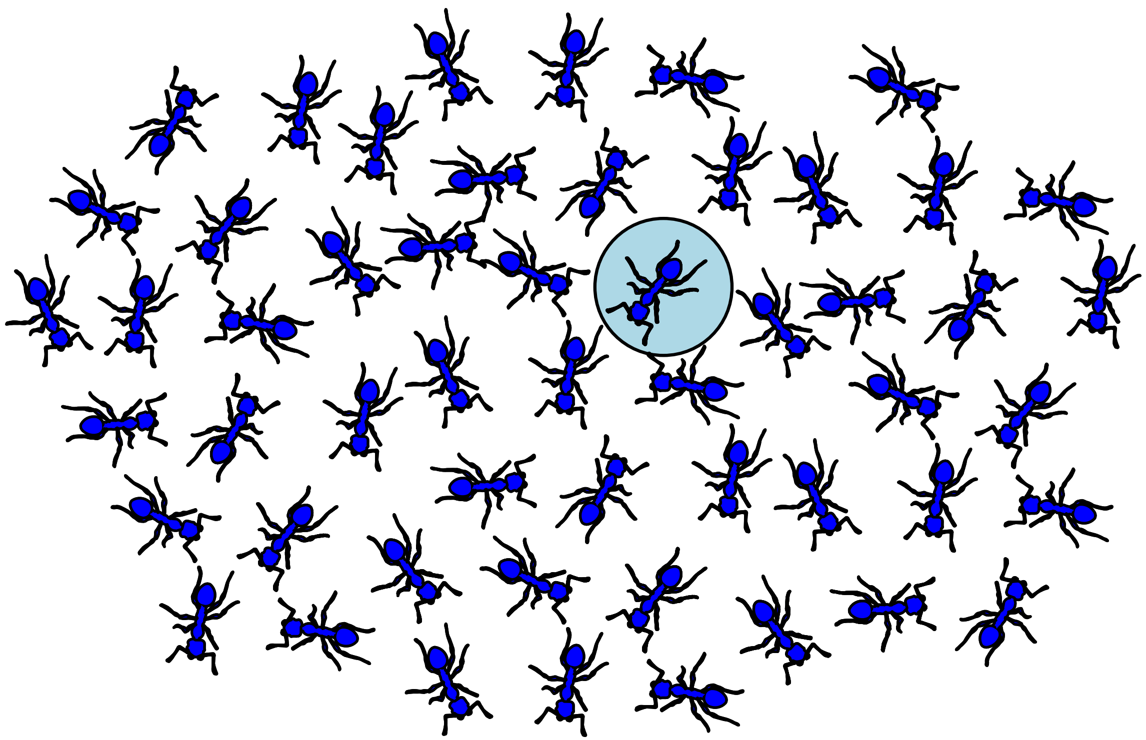
Self-stabilizing Information Spreading

Sources' bits (and other agents' states) may change in response to *external environment*.



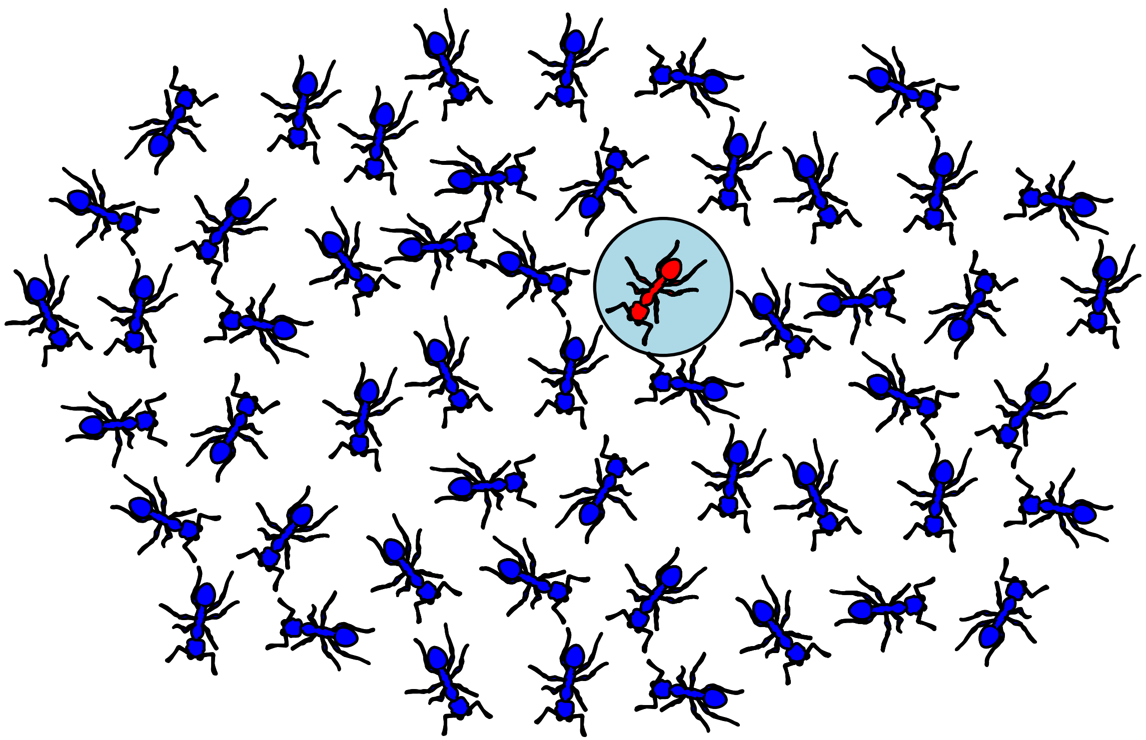
Self-stabilizing Information Spreading

Sources' bits (and other agents' states) may change in response to *external environment*.



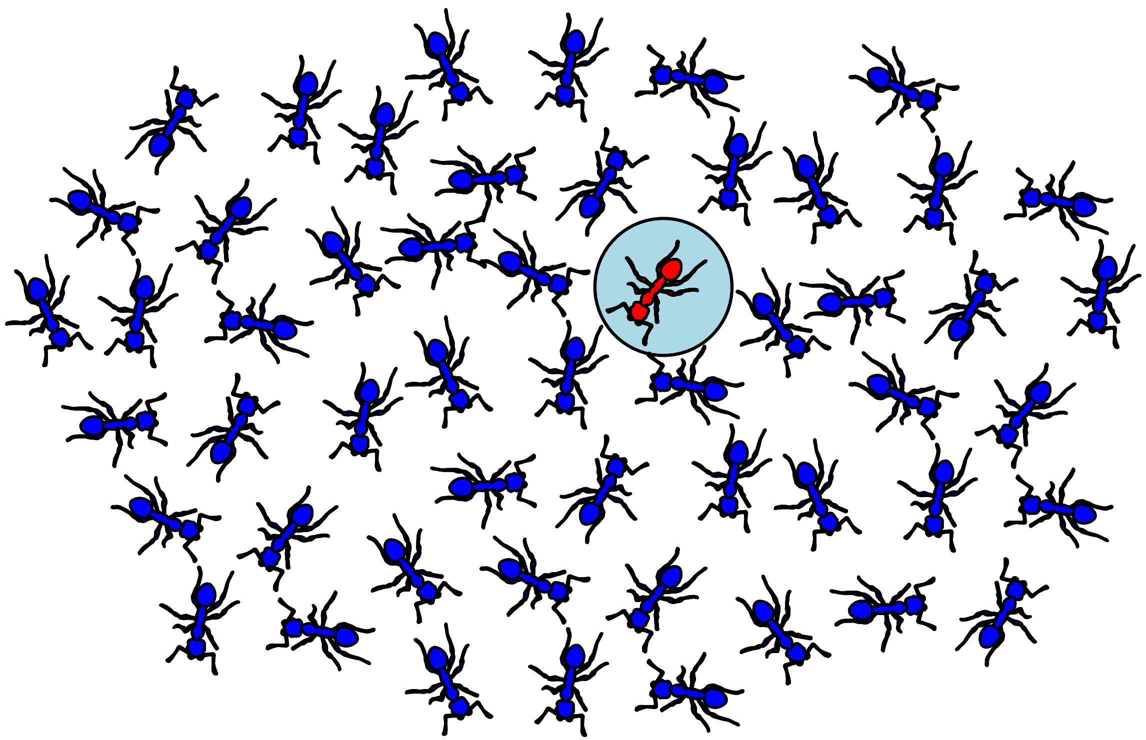
Self-stabilizing Information Spreading

Sources' bits (and other agents' states) may change in response to *external environment*.



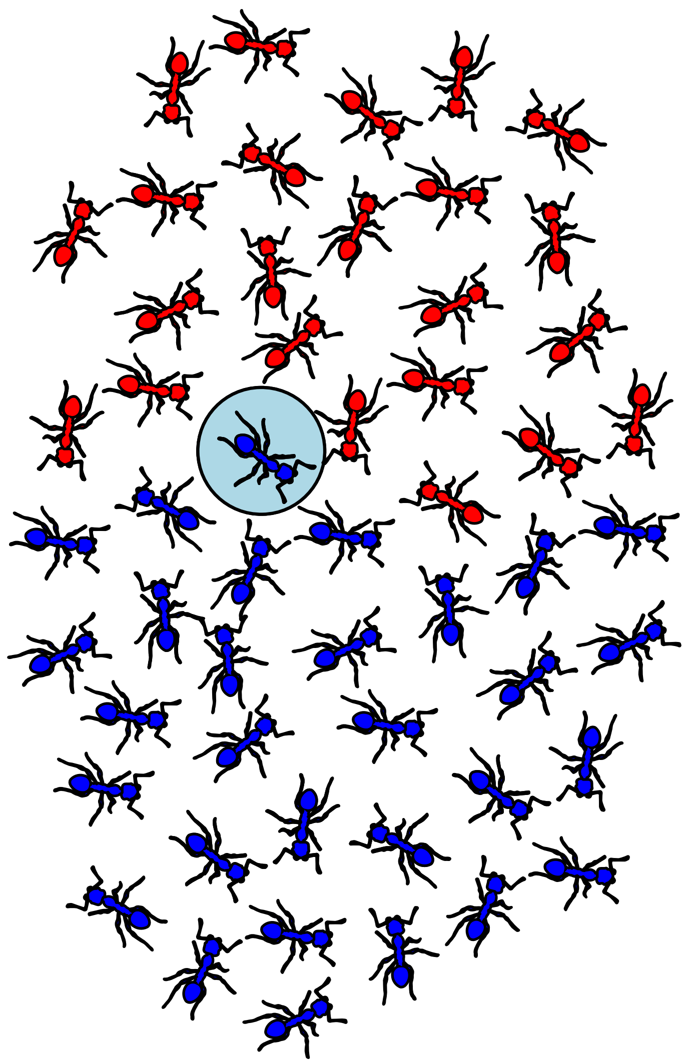
Self-stabilizing Information Spreading

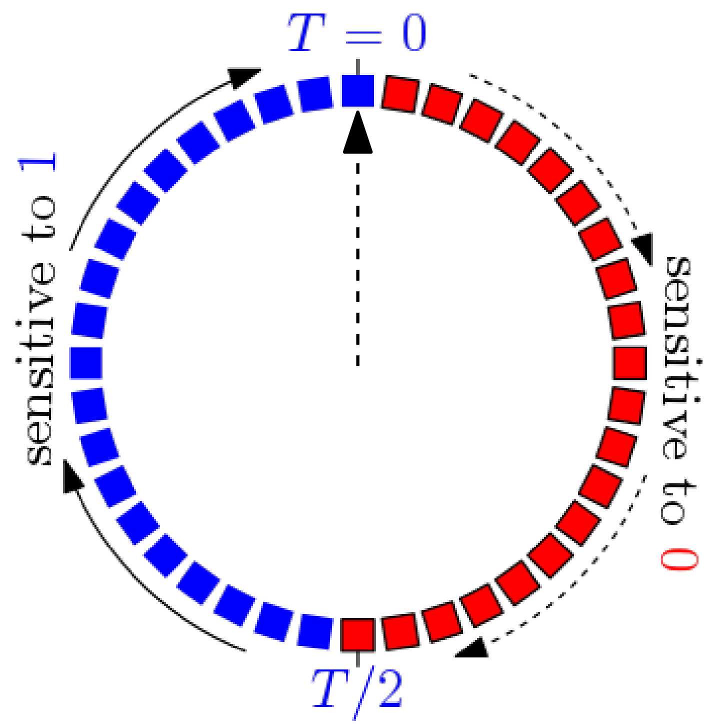
Sources' bits (and other agents' states) may change in response to *external environment*.



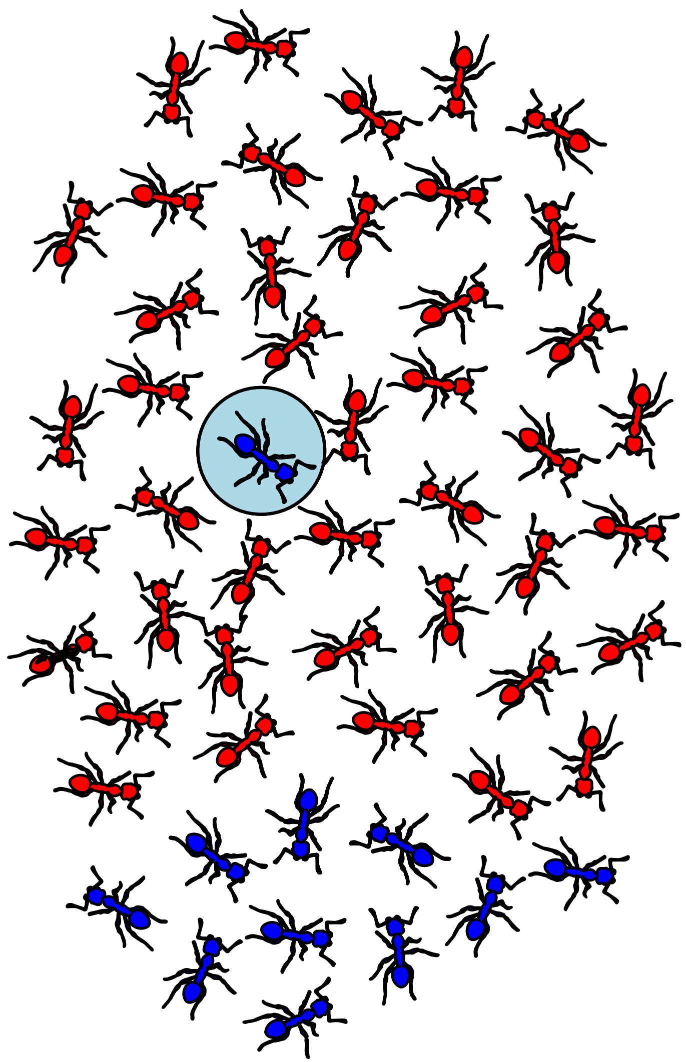
More generally, system is initialized in *arbitrary state* (self-stabilization).

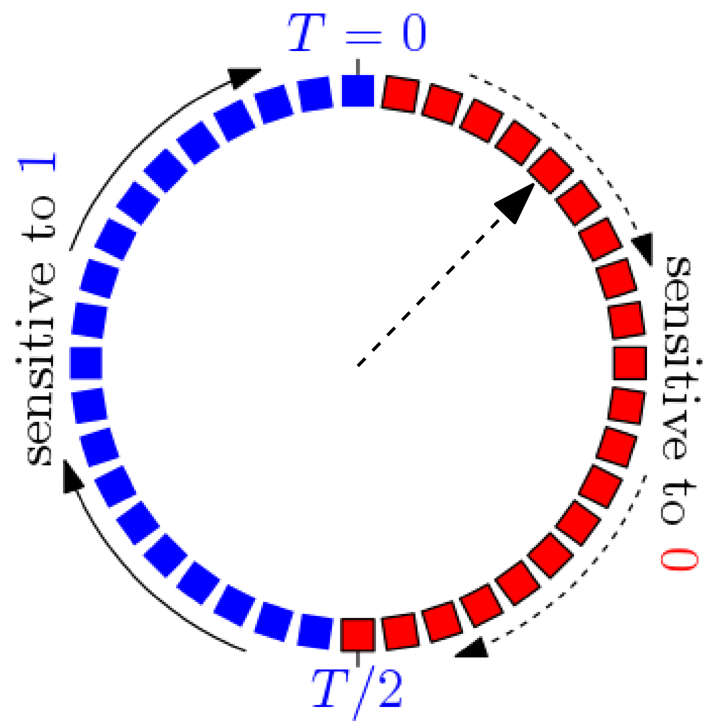
(Self-Stab.) Inf. Spreading vs Synchronization



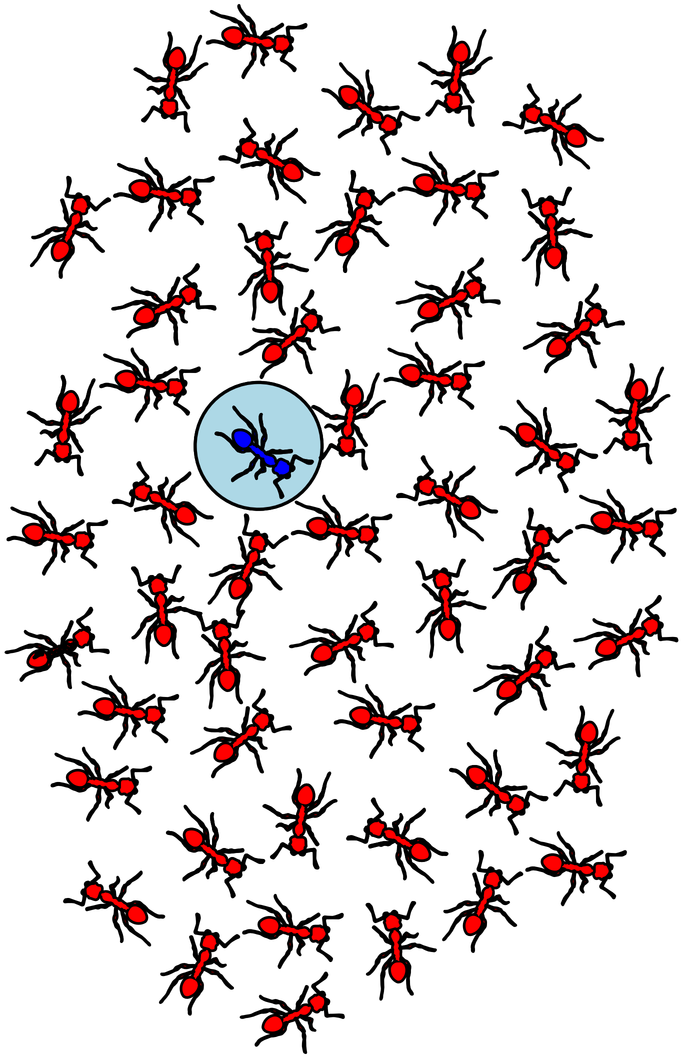


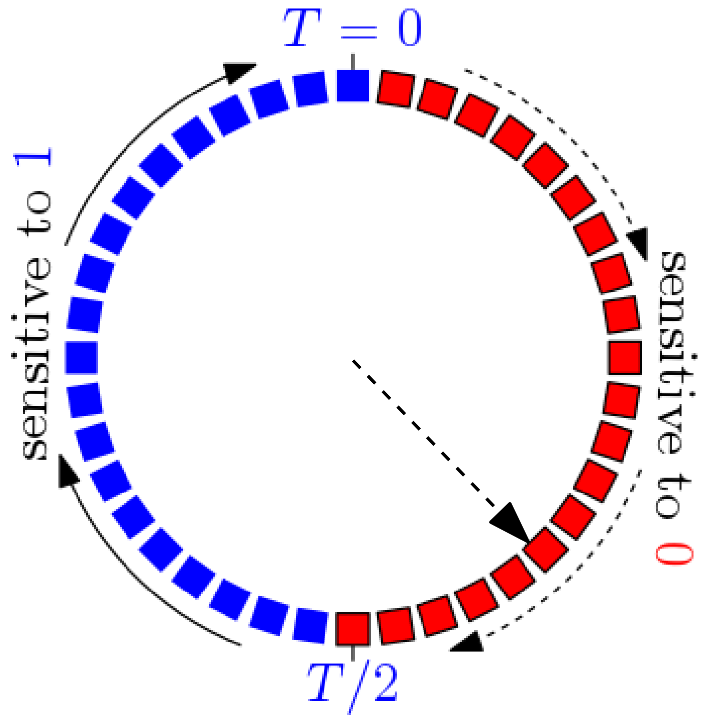
(Self-Stab.) Inf. Spreading vs Synchronization



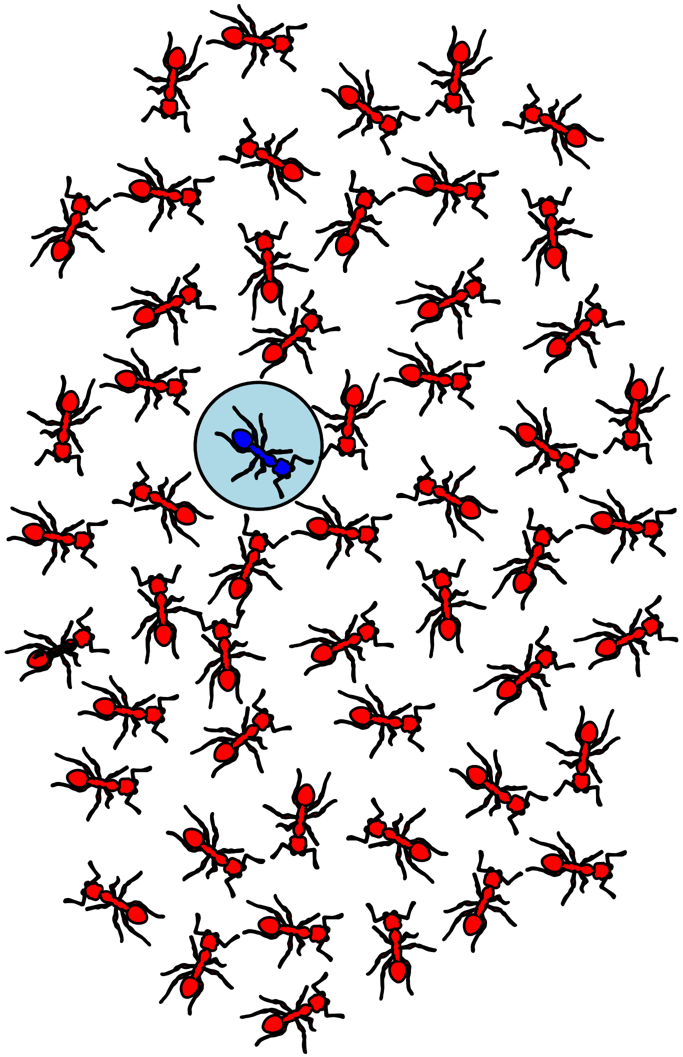


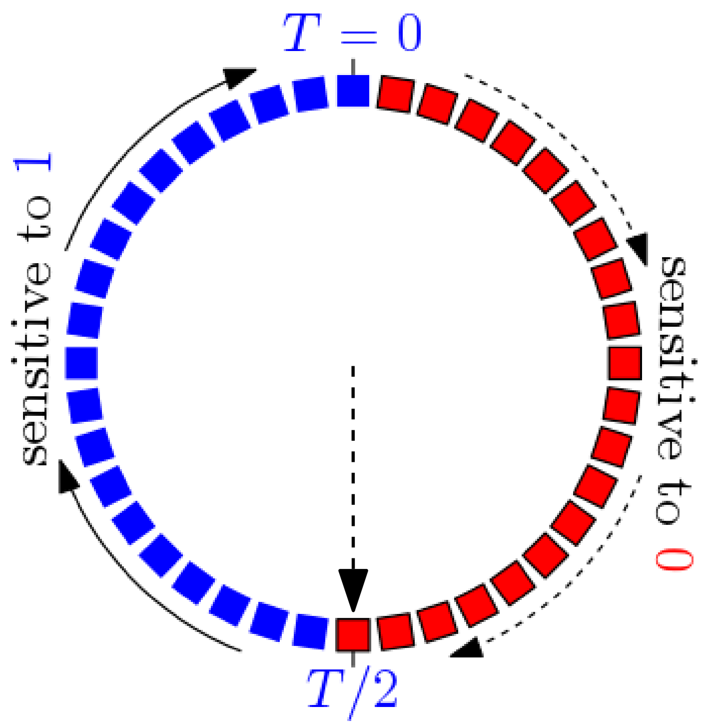
(Self-Stab.) Inf. Spreading vs Synchronization



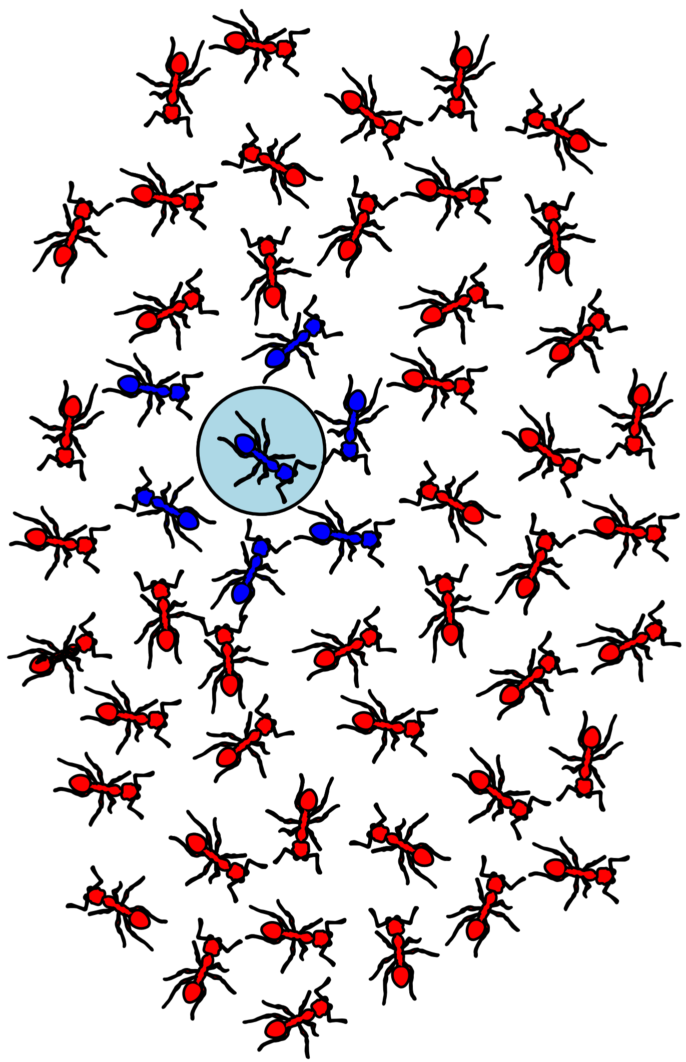


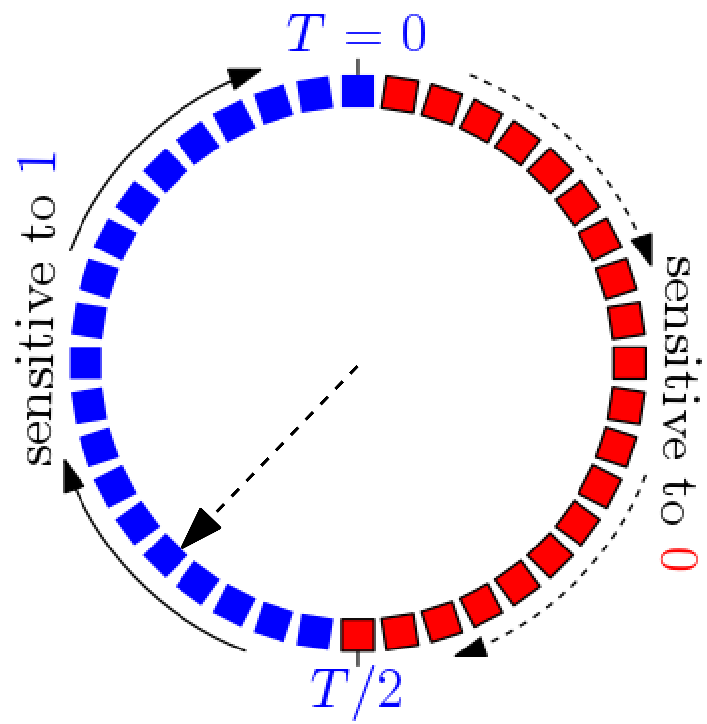
(Self-Stab.) Inf. Spreading vs Synchronization



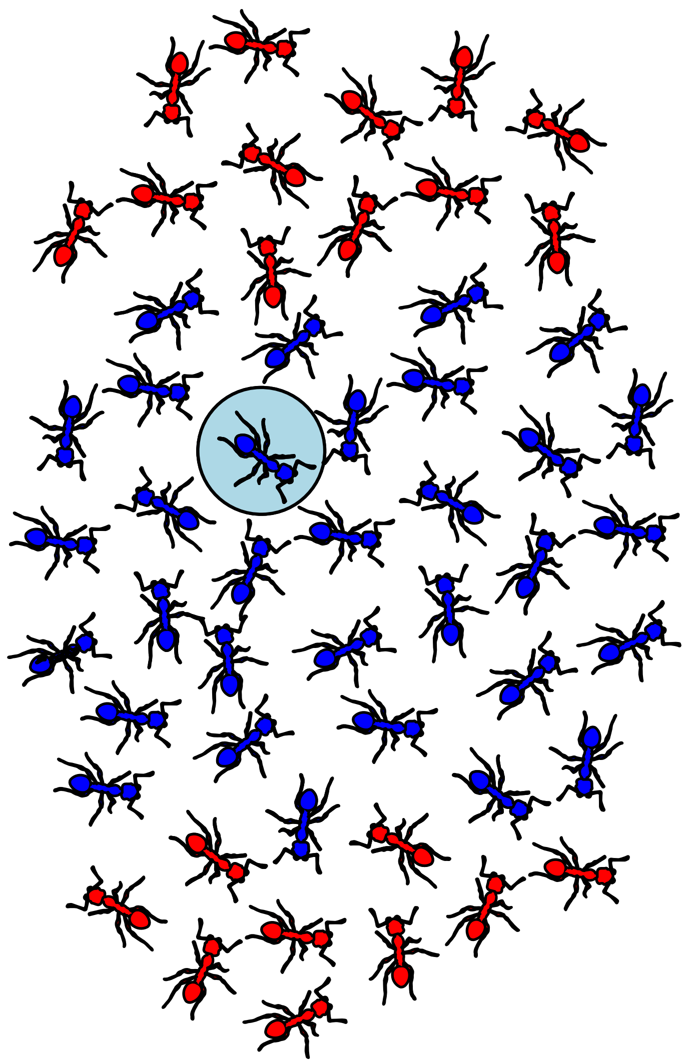


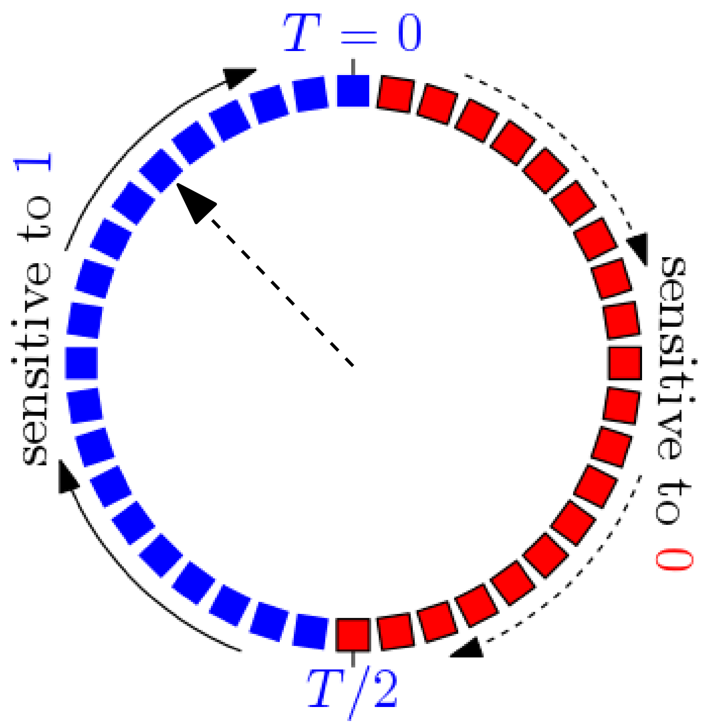
(Self-Stab.) Inf. Spreading vs Synchronization



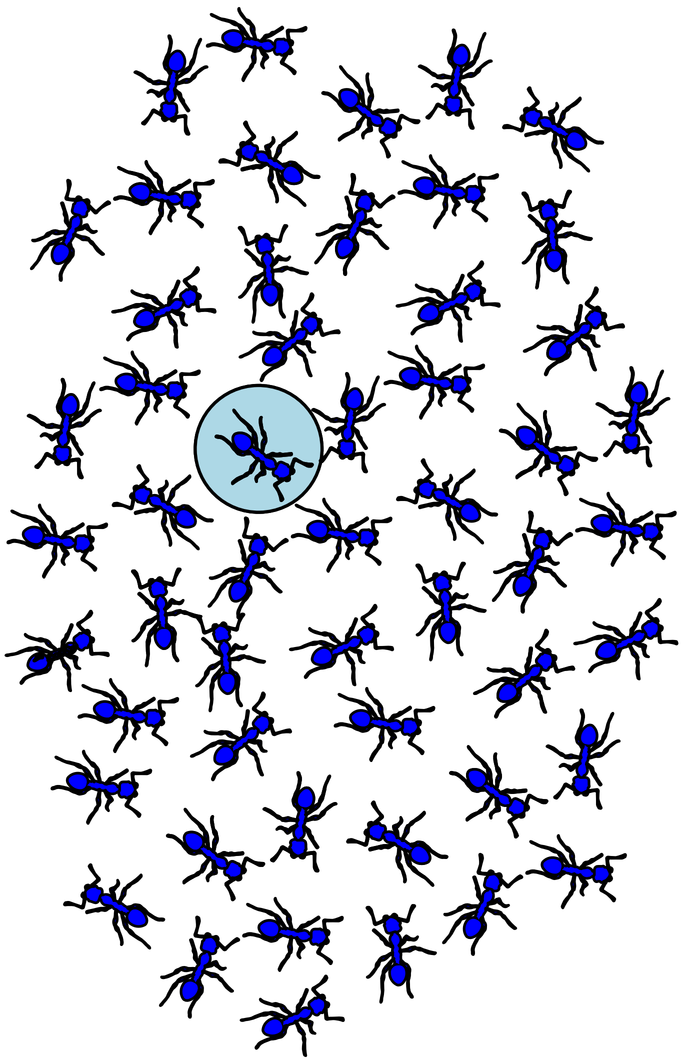


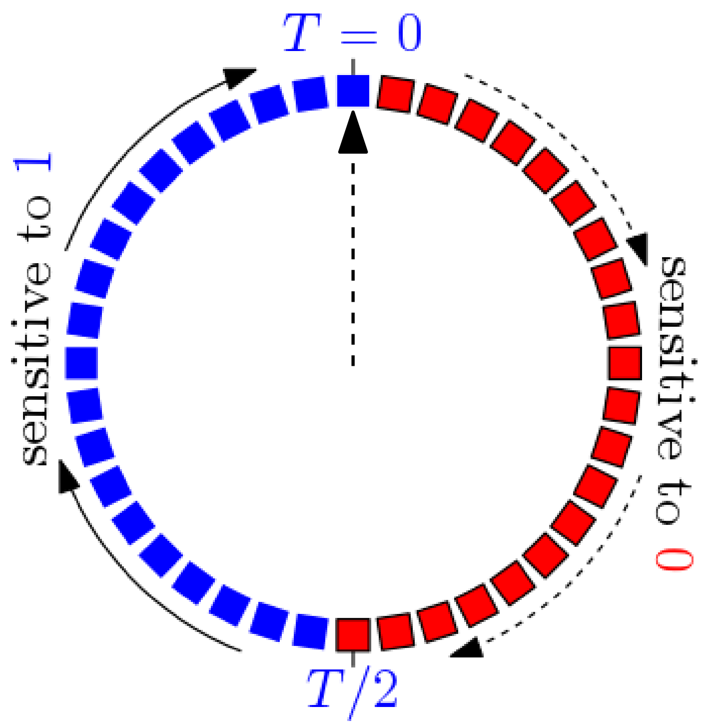
(Self-Stab.) Inf. Spreading vs Synchronization





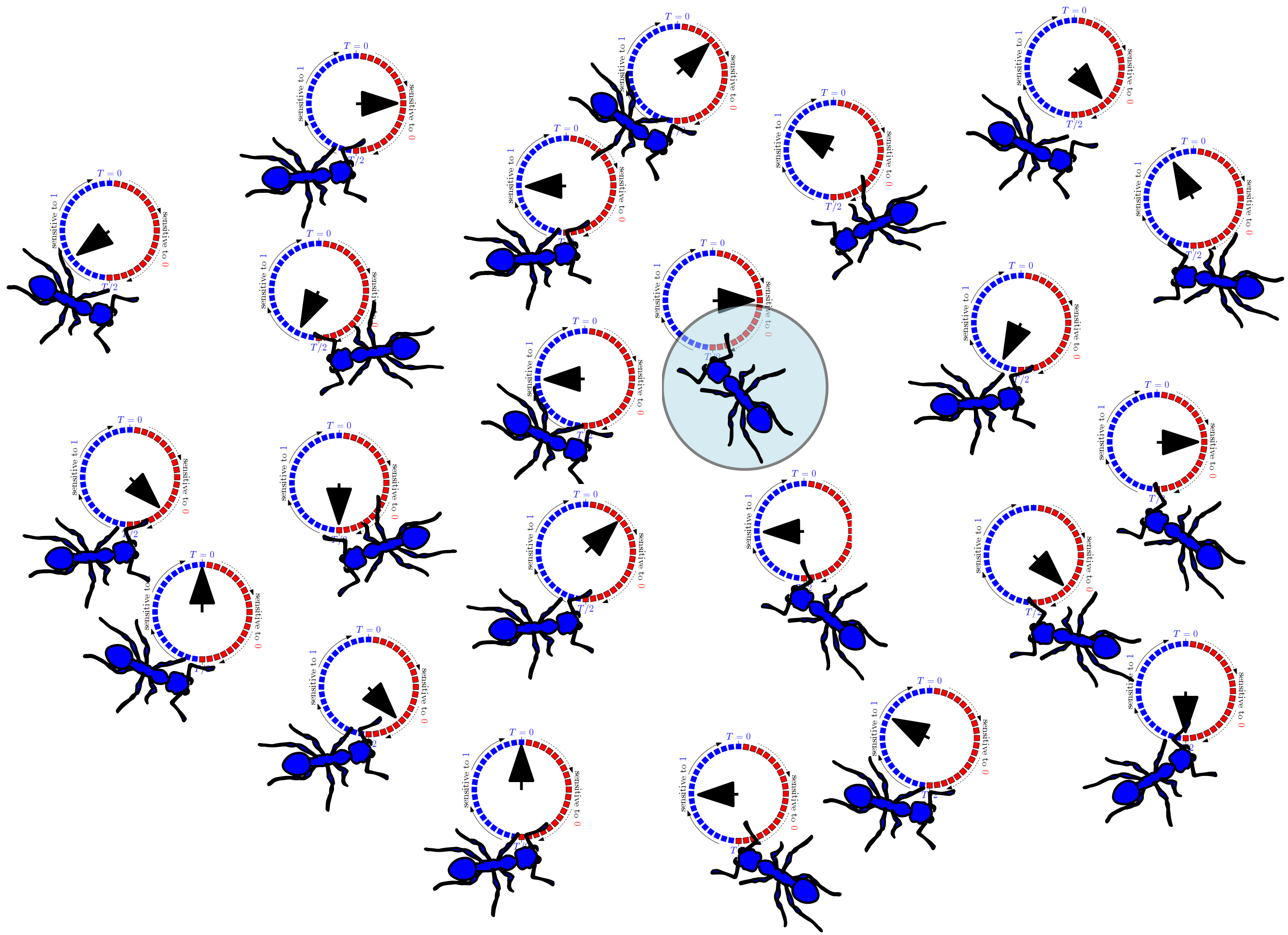
(Self-Stab.) Inf. Spreading vs Synchronization



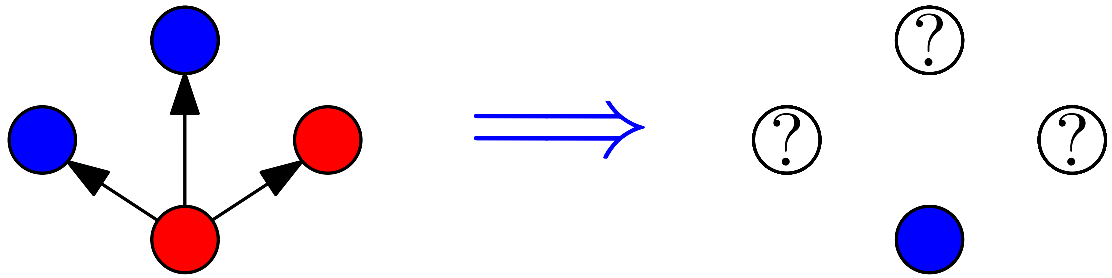


(Self-Stab.) Inf. Spreading vs Synchronization

Self-stablizing algorithms converge from
any initial configuration

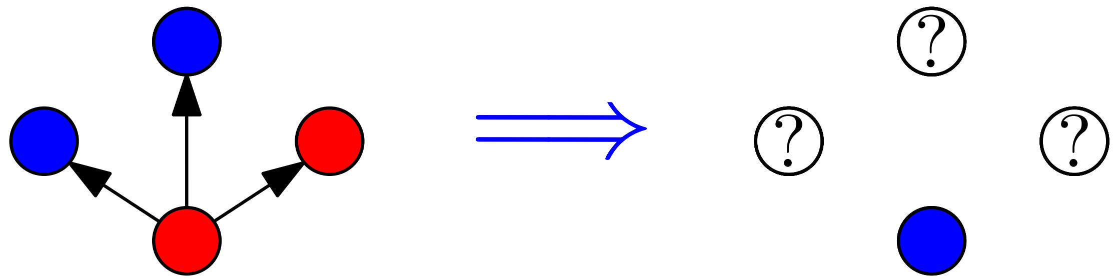


S.-Stab. Sync. in $\mathcal{PULL}$ with Small Messages?

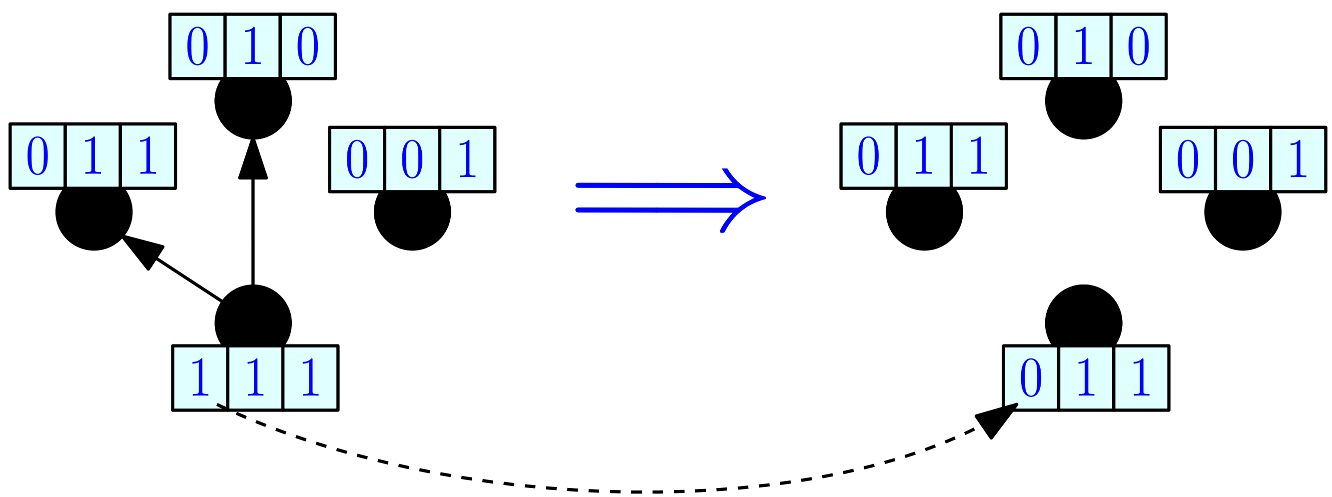


2-Choices dynamics. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.

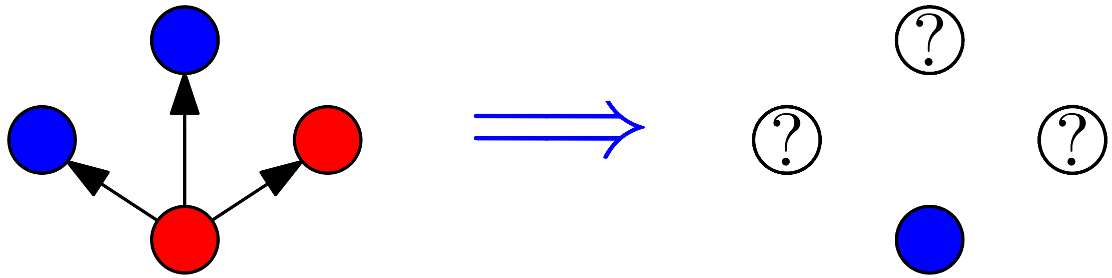
S.-Stab. Sync. in $\mathcal{PULL}$ with Small Messages?



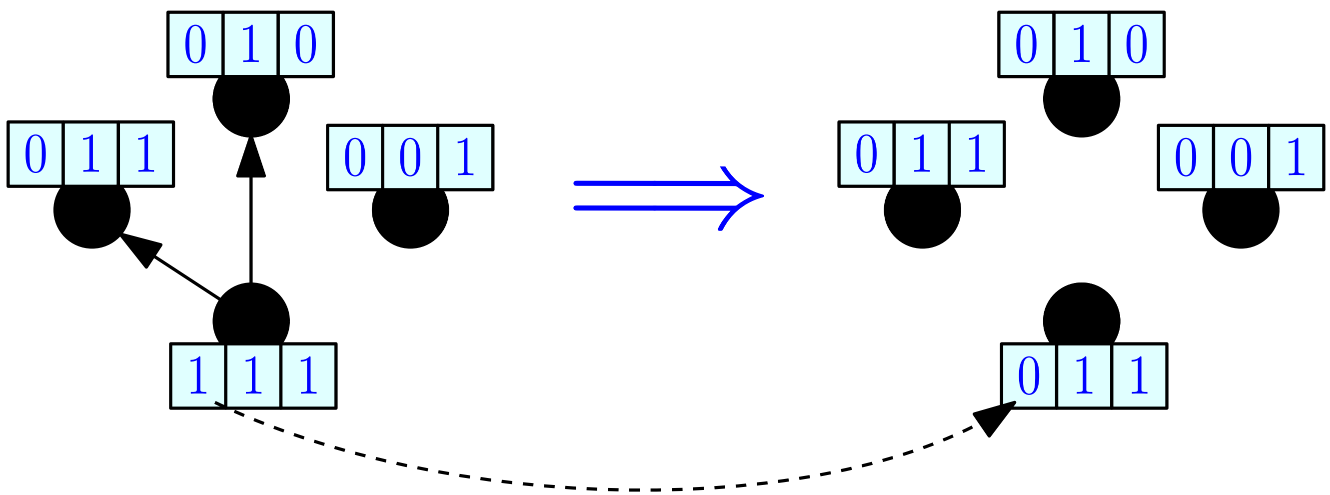
2-Choices dynamics. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.



S.-Stab. Sync. in $\mathcal{PULL}$ with Small Messages?

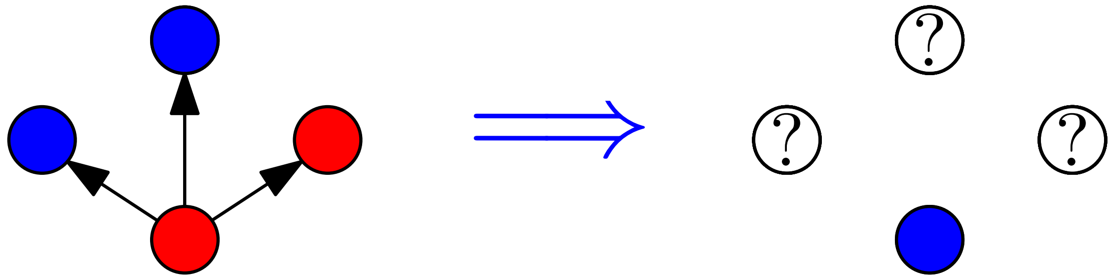


2-Choices dynamics. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.

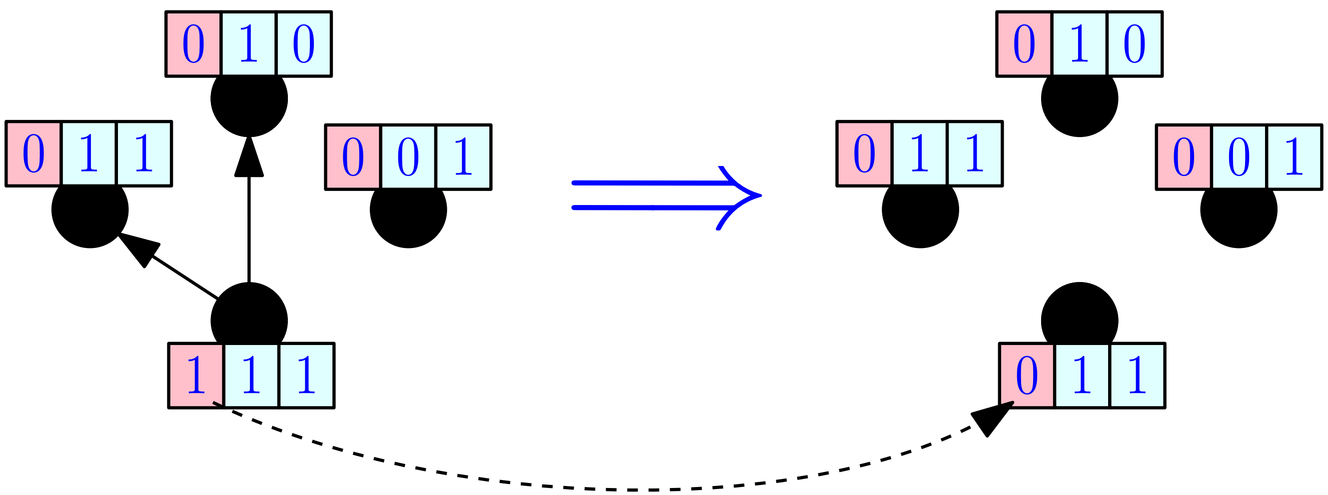


T-clock can be sync. in $\mathcal{O}(\log n \log T)$ rounds w.h.p. using $\log T$ *bits*.
But Binary Information Spreading can be done in 1-*bit* $\mathcal{PULL}$...

S.-Stab. Sync. in $\mathcal{PULL}$ with Small Messages?

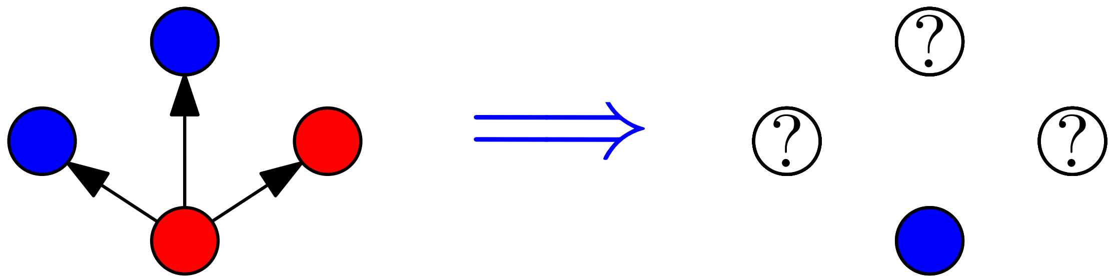


2-Choices dynamics. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.



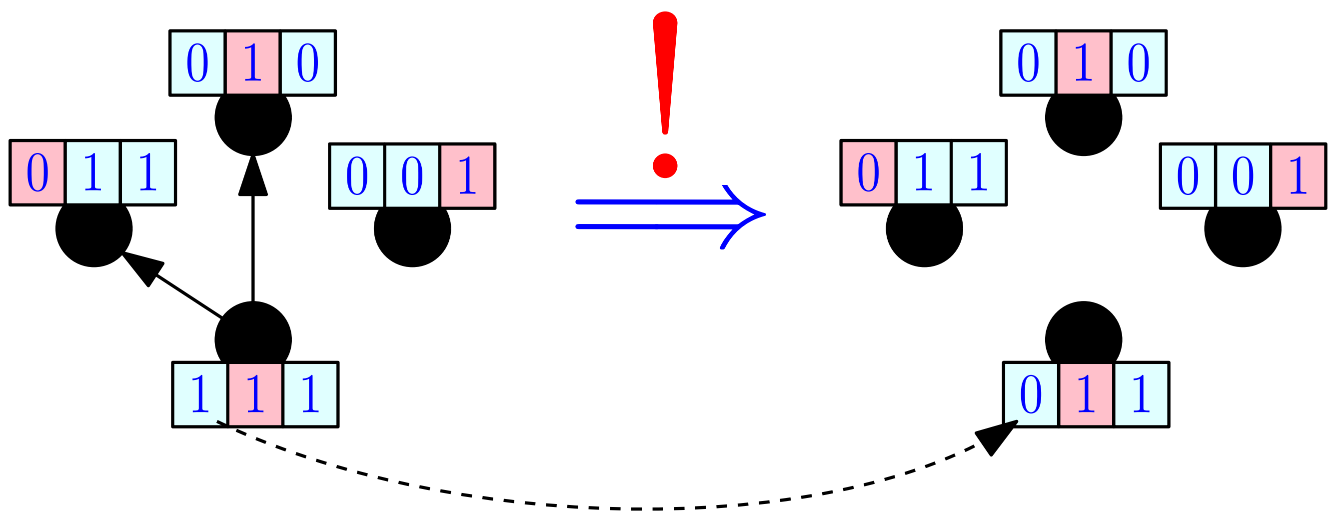
T-clock can be sync. in $\mathcal{O}(\log n \log T)$ rounds w.h.p. using $\log T$ *bits*.
But Binary Information Spreading can be done in 1-*bit* $\mathcal{PULL}$...

S.-Stab. Sync. in $\mathcal{PULL}$ with Small Messages?



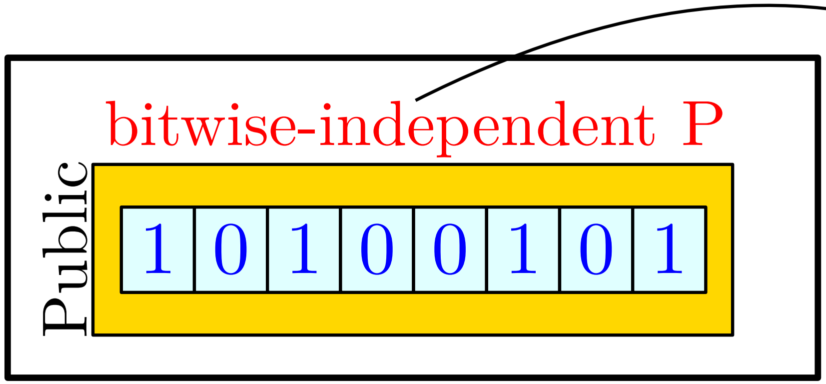
2-Choices dynamics. Converge to consensus in $\mathcal{O}(\log n)$ rounds with high probability.

Ok only if indices are already sync!

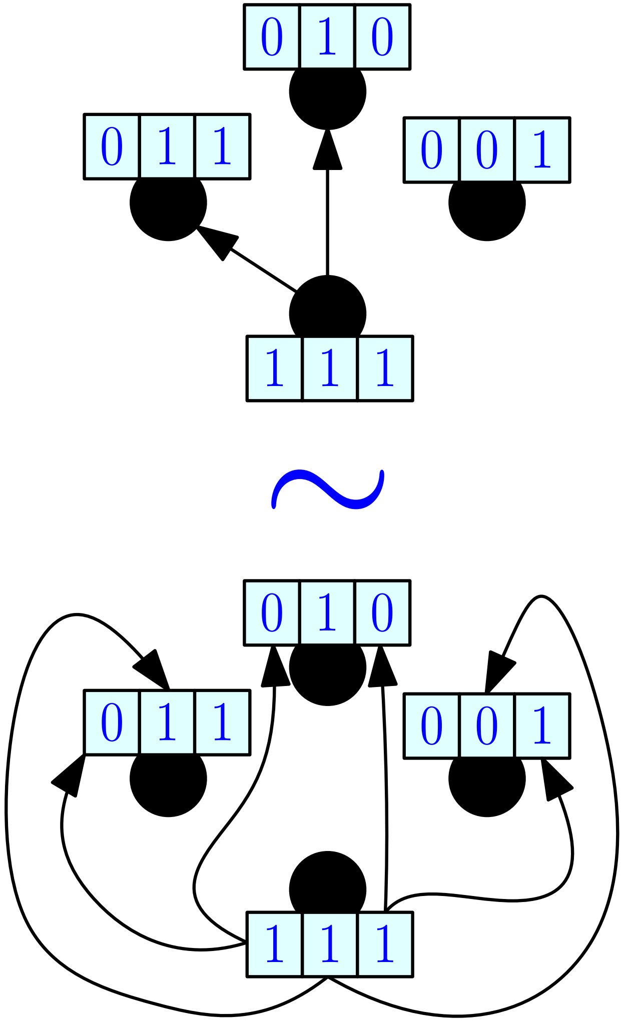


T-clock can be sync. in $\mathcal{O}(\log n \log T)$ rounds w.h.p. using $\log T$ *bits*.
But Binary Information Spreading can be done in 1-*bit* $\mathcal{PULL}$...

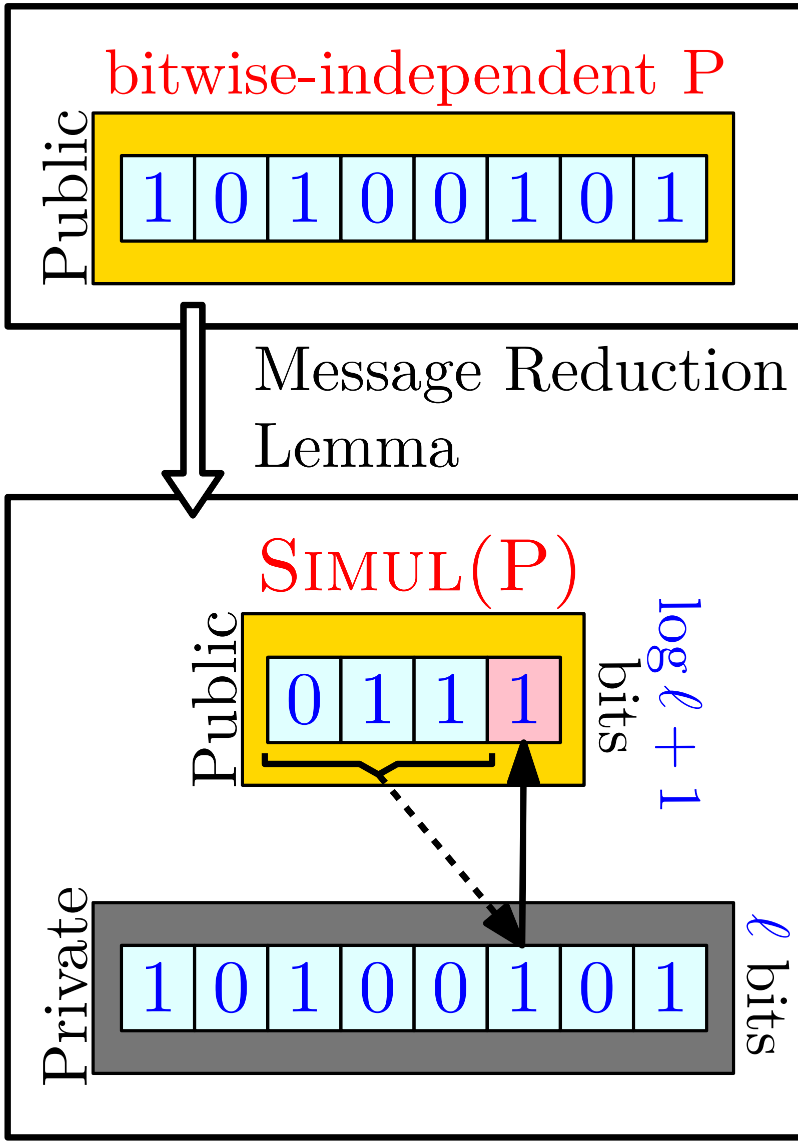
The Message Reduction Lemma



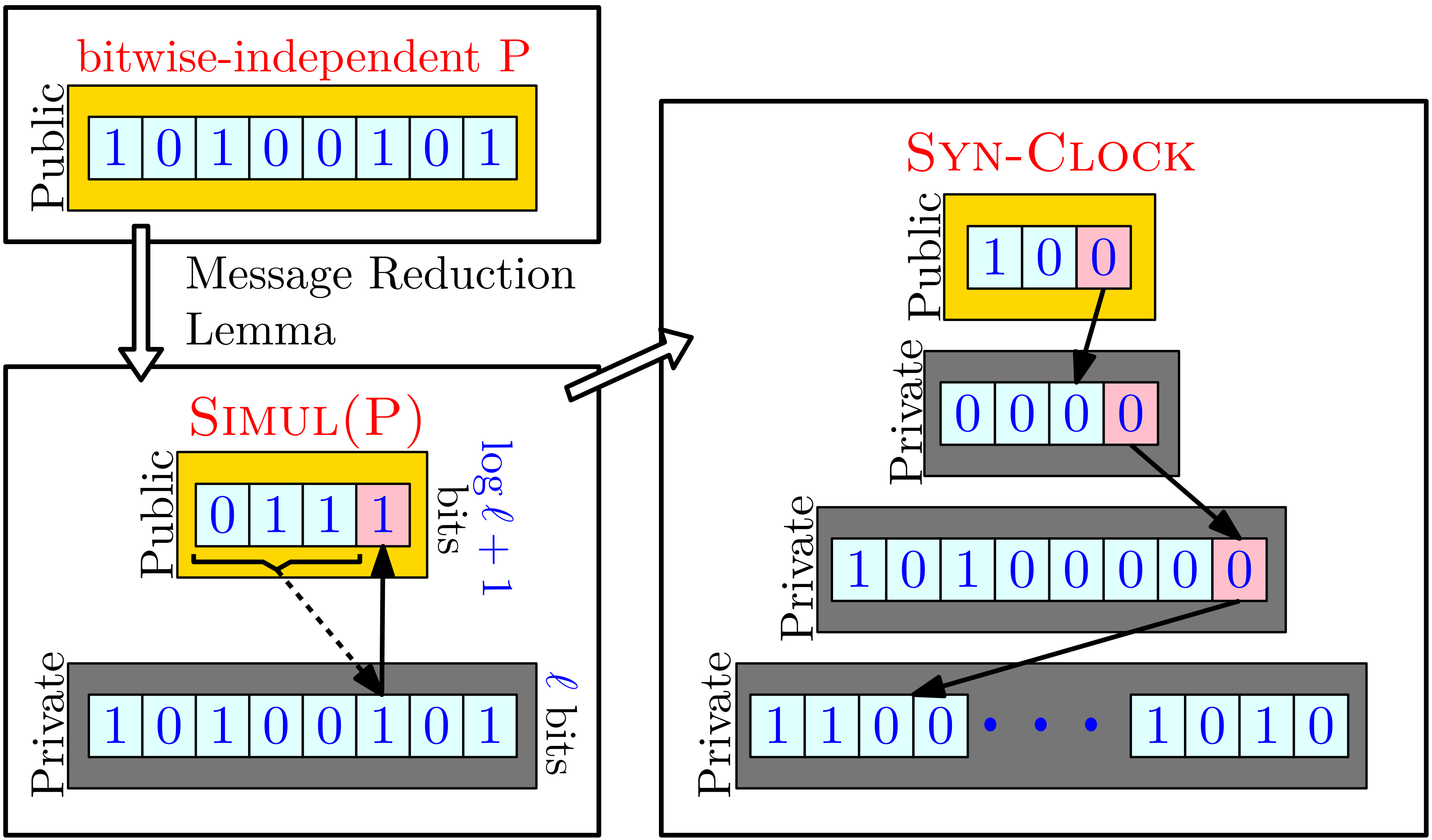
Parts of message can come from different agents:



The Message Reduction Lemma



The Message Reduction Lemma



Results: 3 Bits suffice...

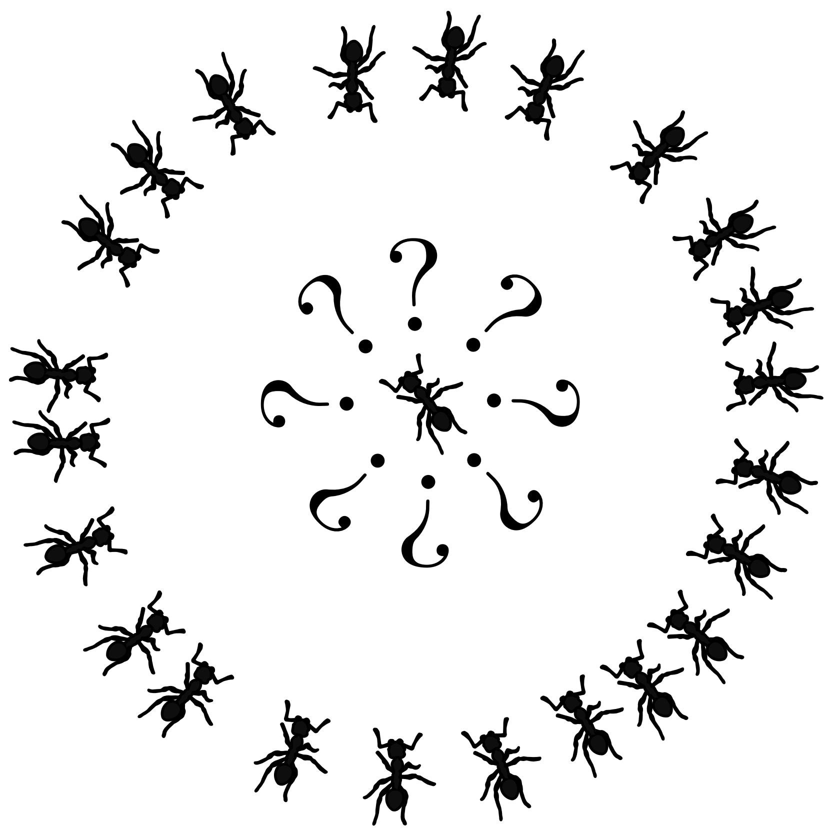
Theorem (Clock Syncronization) [1]. There is a *self-stabilizing* clock synchronization protocol which synchronizes a clock modulo T in $\tilde{\mathcal{O}}(\log n \log T)$ rounds w.h.p. using **3-bit messages**.

Corollary (Self-stabilizing Majority Infromation Spreading) [1]. There is a *self-stabilizing* Majority Information Spreading protocol which converges in $\tilde{\mathcal{O}}(\log n)$ rounds w.h.p using **3-bit messages**, provided majority is supported by $(\frac{1}{2} + \epsilon)$-fraction of source agents.

[1] L. Boczkowski, A. Korman, and E. Natale, "Minimizing Message Size in Stochastic Communication Patterns: Fast Self-Stabilizing Protocols with 3 bits," in Proc. of 28th ACM-SIAM SODA, 2017.

Noisy Information Spreading

Communication model: $\mathcal{PUSH}$ model [1]:
at each round each agent can **send** a bit to another one chosen uniformly at random.



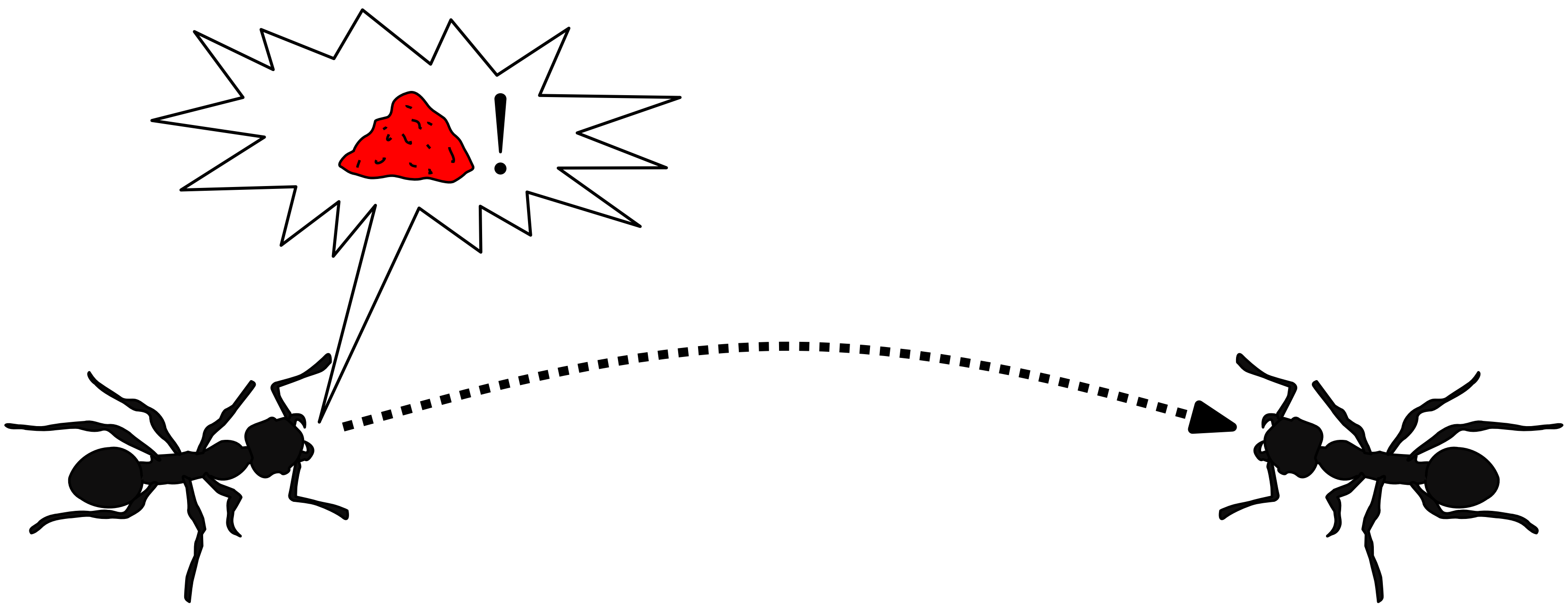
[1] B. Pittel, "On Spreading a Rumor," SIAM J. Appl. Math., vol. 47, no. 1, pp. 213–223, Mar. 1987.

Noisy Information Spreading

Communication model: $\mathcal{PUSH}$ model [1]:
at each round each agent can **send** a bit to another one chosen uniformly at random.

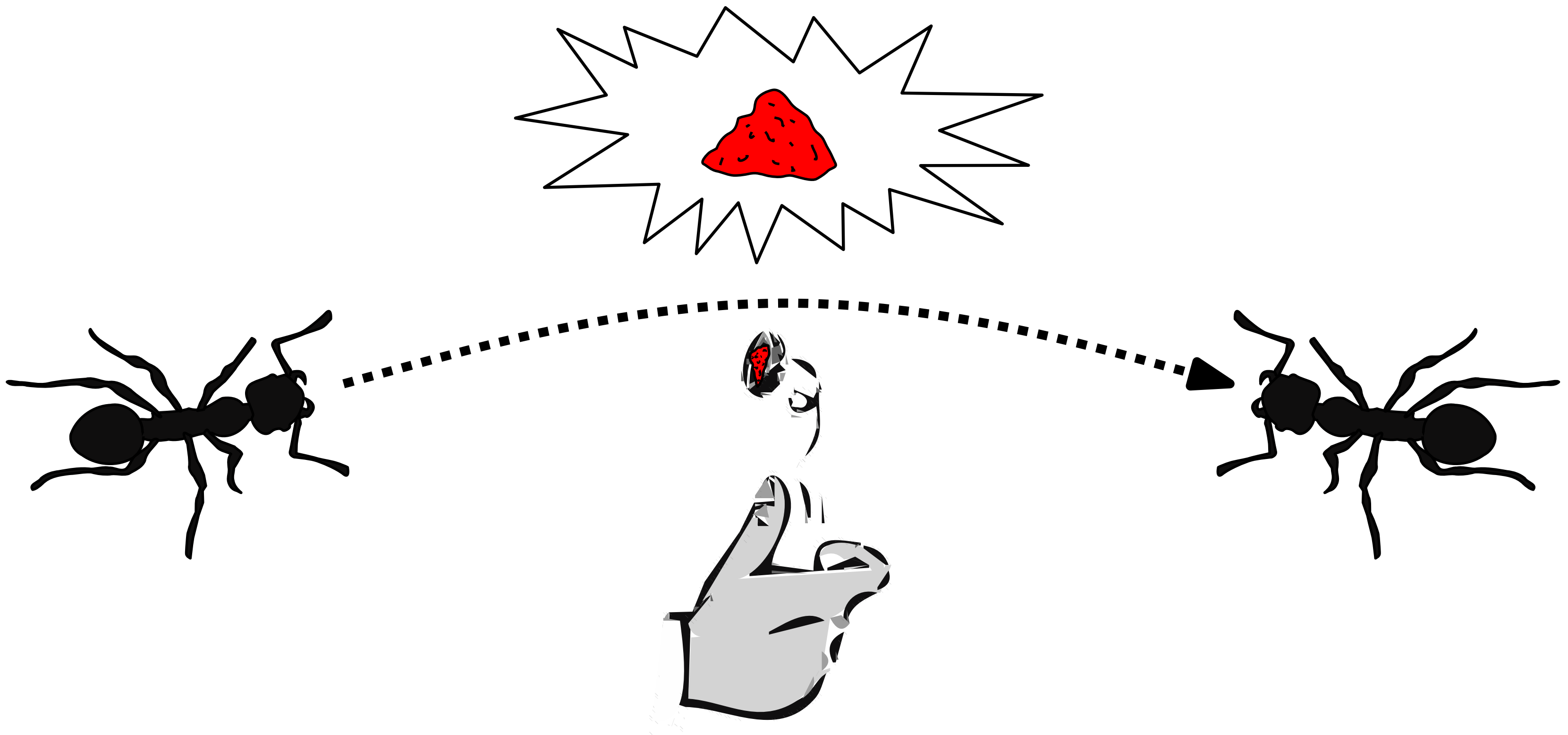
Noisy Information Spreading

Noise: before being received, each bit is flipped with probability $1/2 - \epsilon$ ($\epsilon = n^{-const}$).



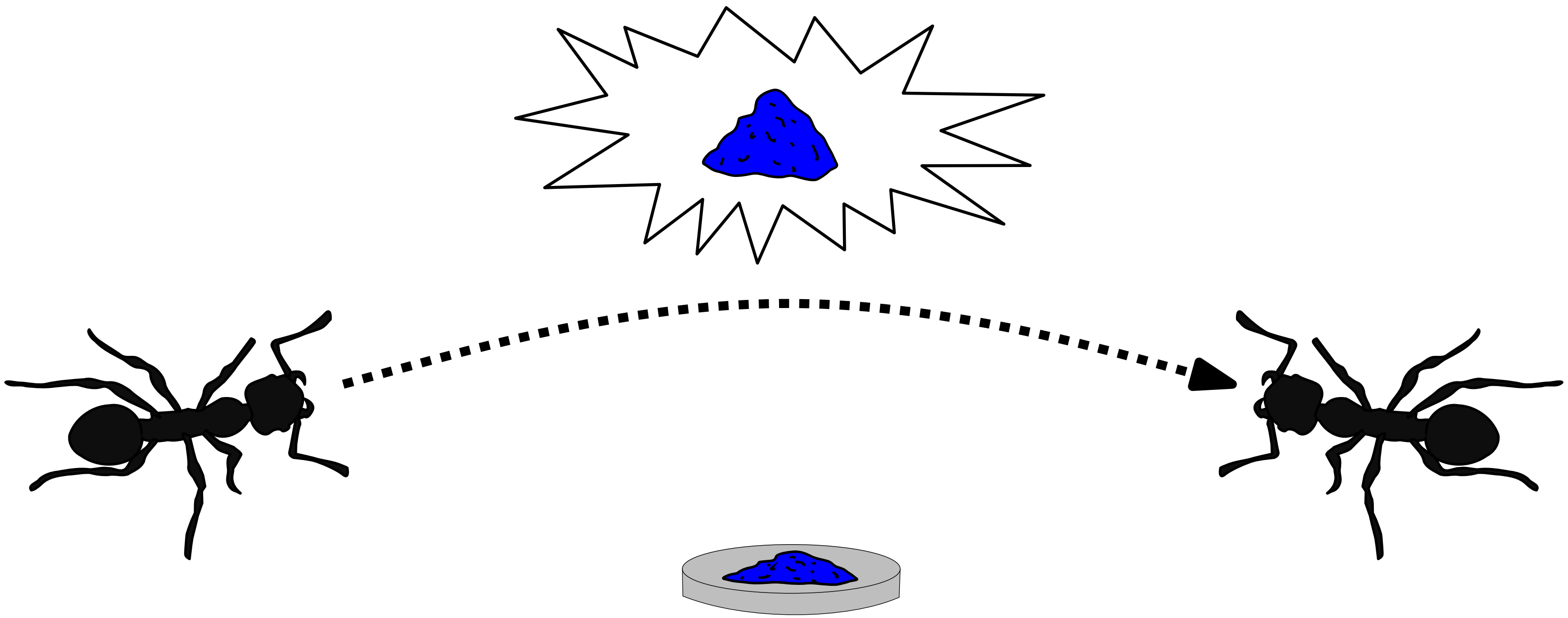
Noisy Information Spreading

Noise: before being received, each bit is flipped with probability $1/2 - \epsilon$ ($\epsilon = n^{-const}$).



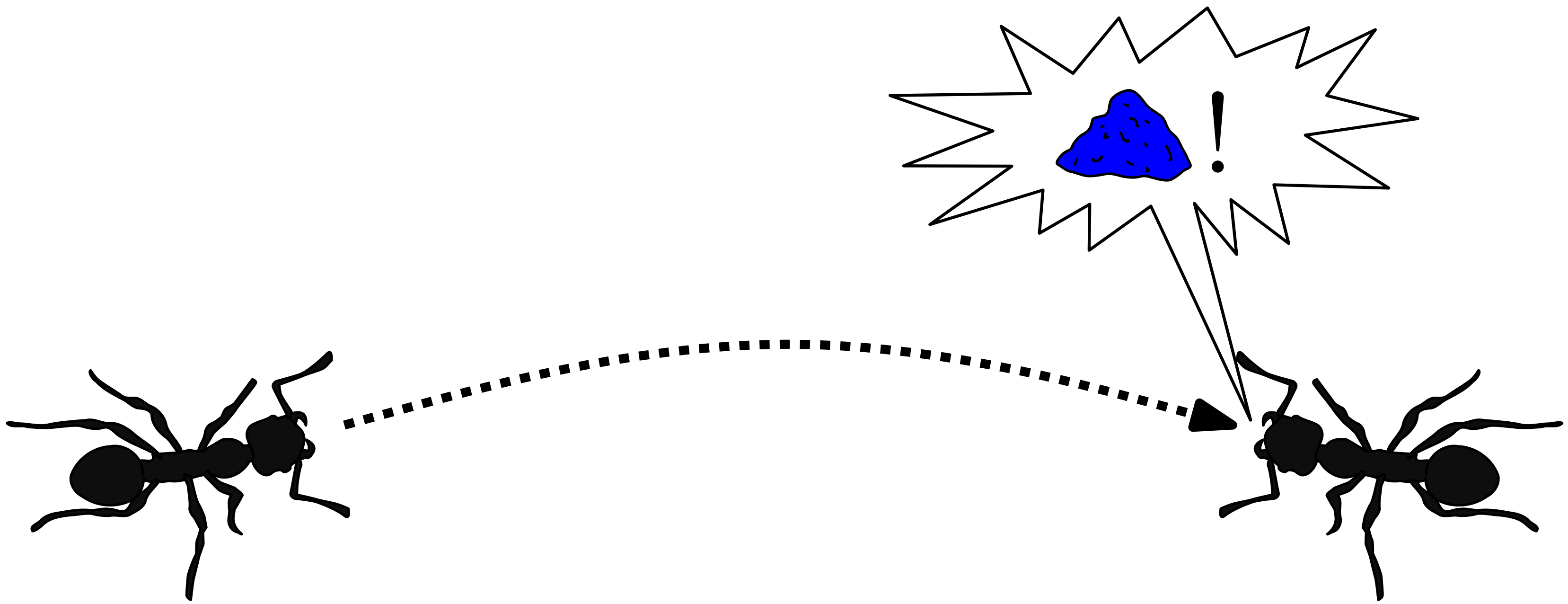
Noisy Information Spreading

Noise: before being received, each bit is flipped with probability $1/2 - \epsilon$ ($\epsilon = n^{-const}$).

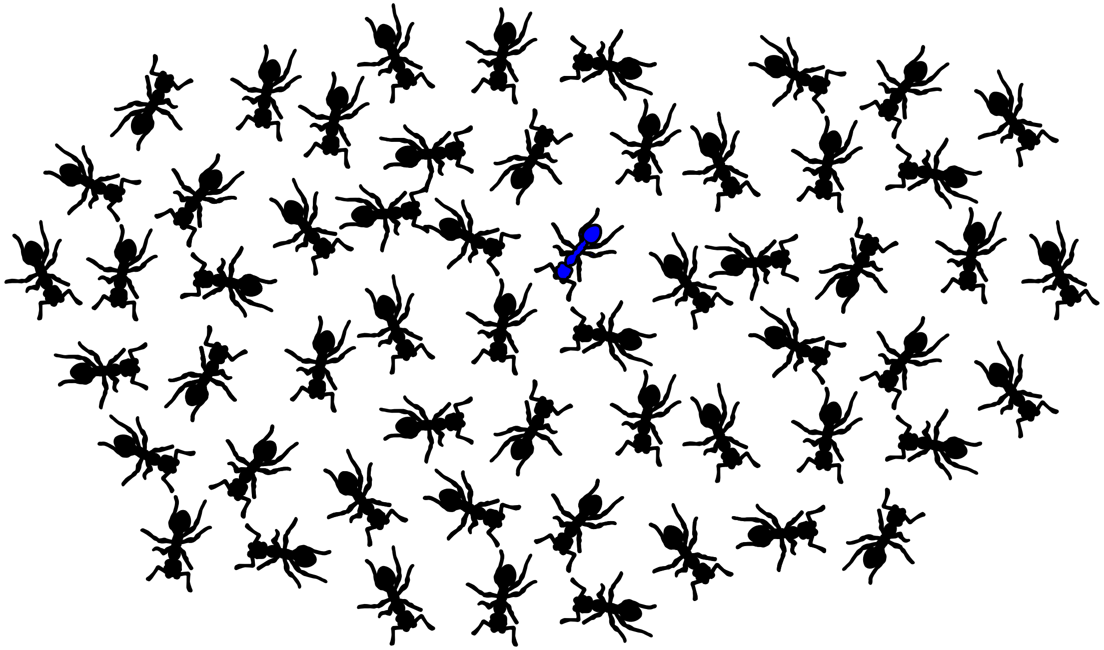


Noisy Information Spreading

Noise: before being received, each bit is flipped with probability $1/2 - \epsilon$ ($\epsilon = n^{-const}$).



Previous Work: Breathe Before Speaking [1]

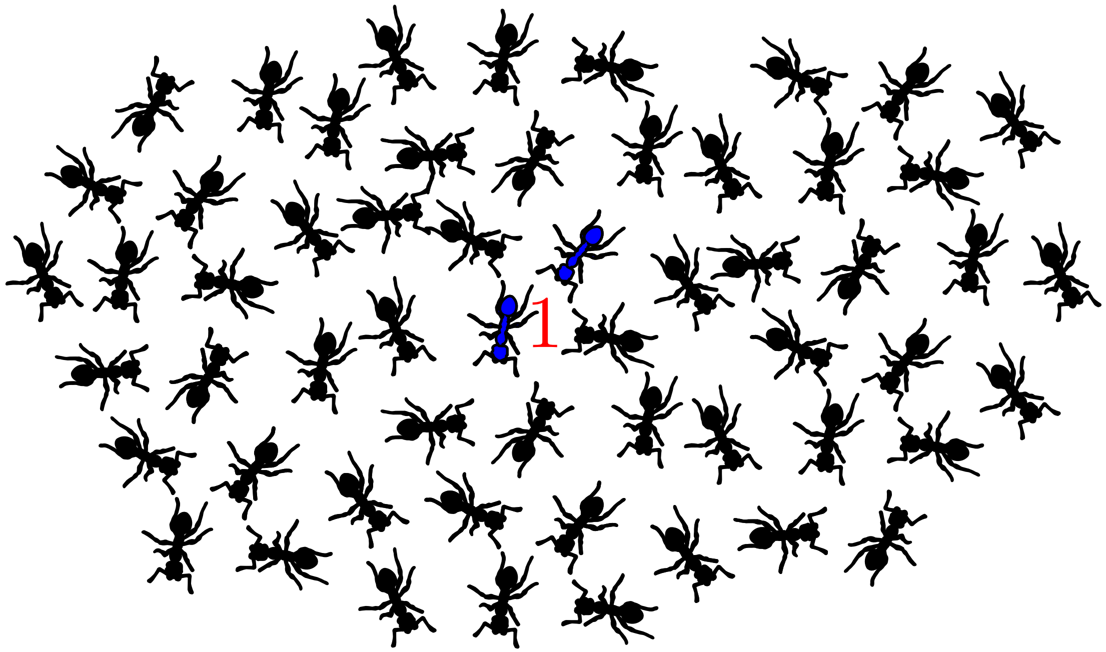


trivial
strategy

blue vs red:
1/0

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]

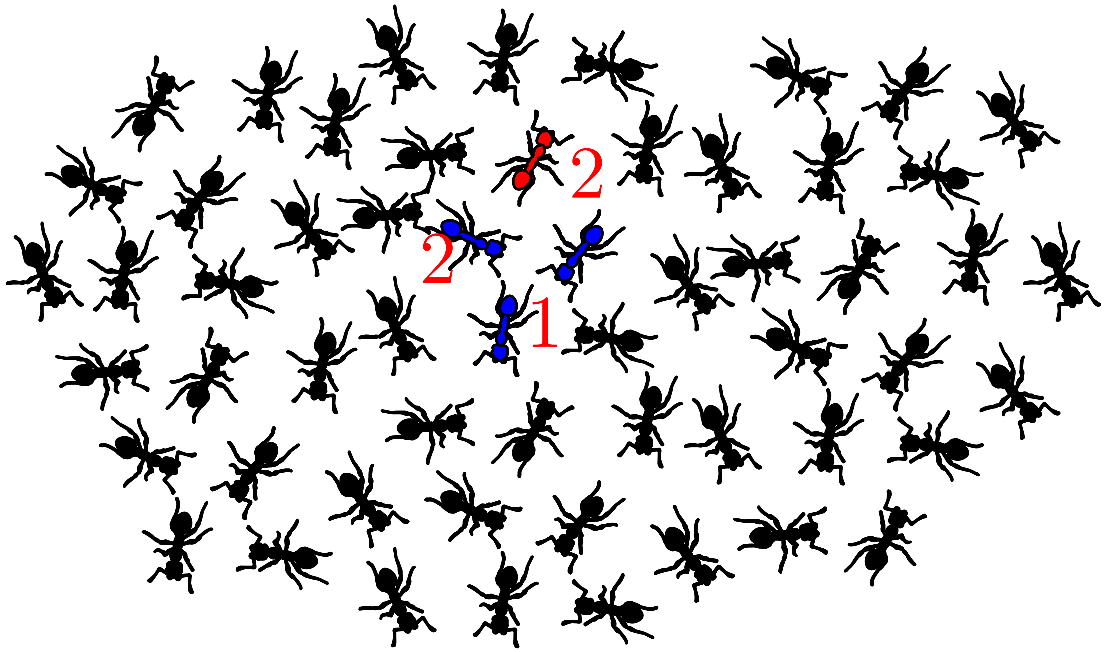


trivial
strategy

blue vs red:
2/0

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]

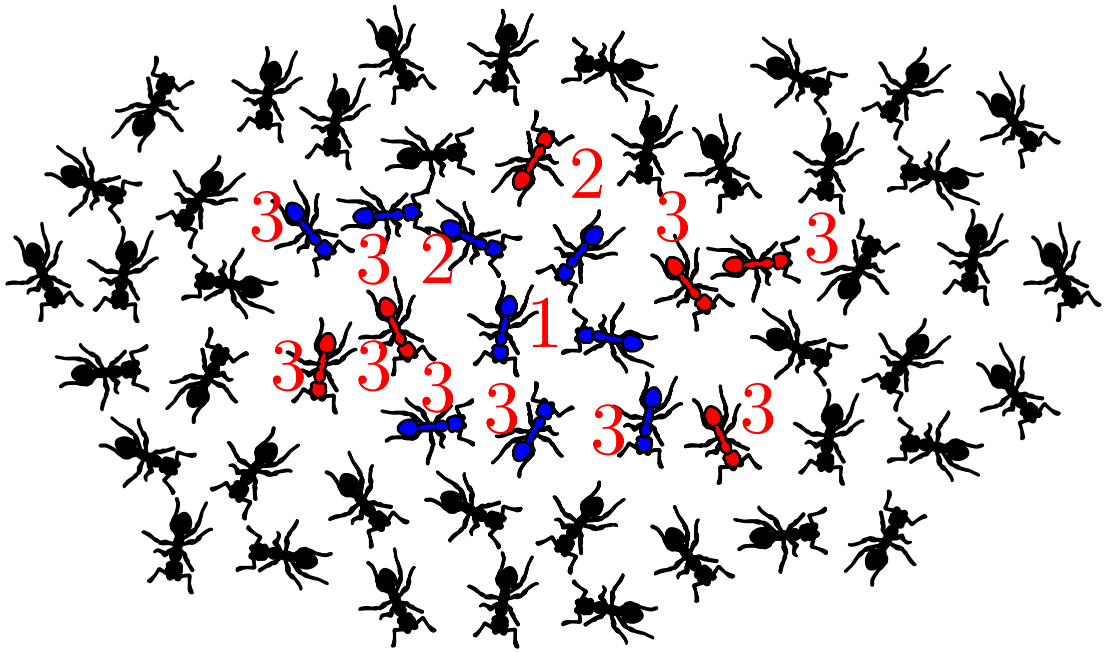


trivial
strategy

blue vs red:
3/1

[1] O. Feinerman, B. Haeupler, and A. Korman, “Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication,” Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]

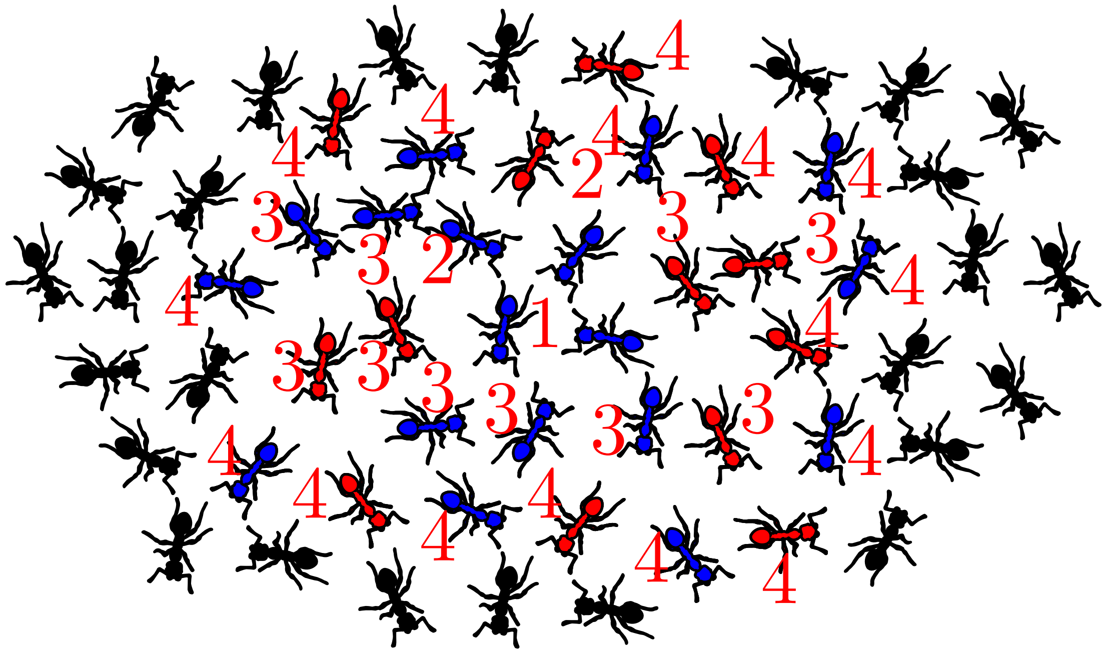


trivial
strategy

blue vs red:
$9/6 = 1.5$

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]

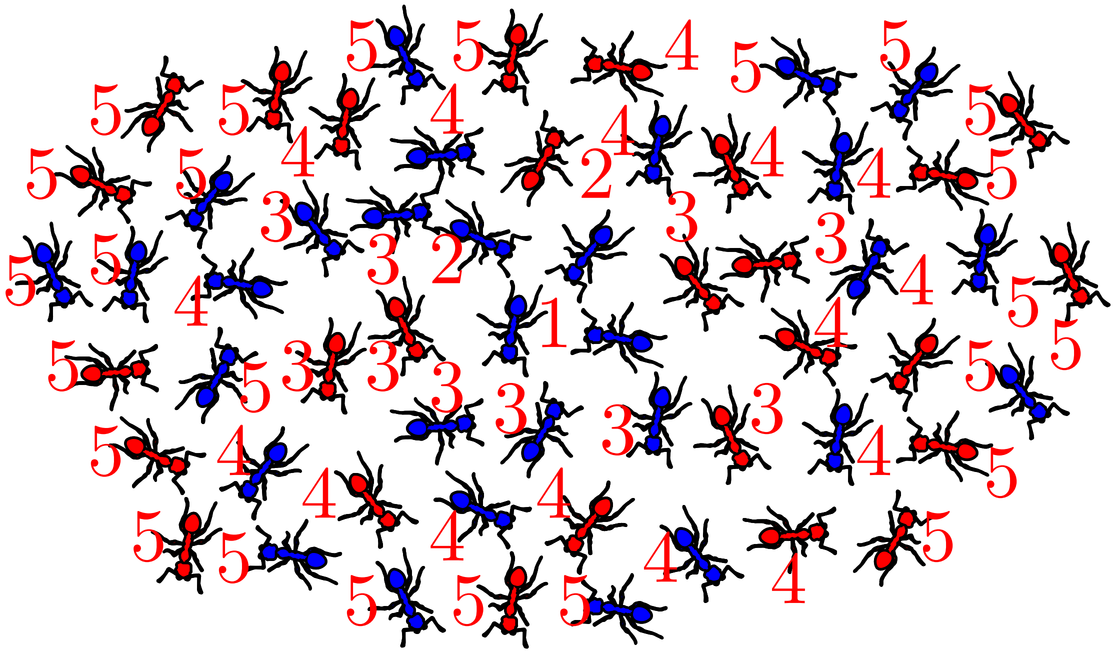


trivial
strategy

blue vs red:
$18/13 \approx 1.4$

[1] O. Feinerman, B. Haeupler, and A. Korman, “Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication,” Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]

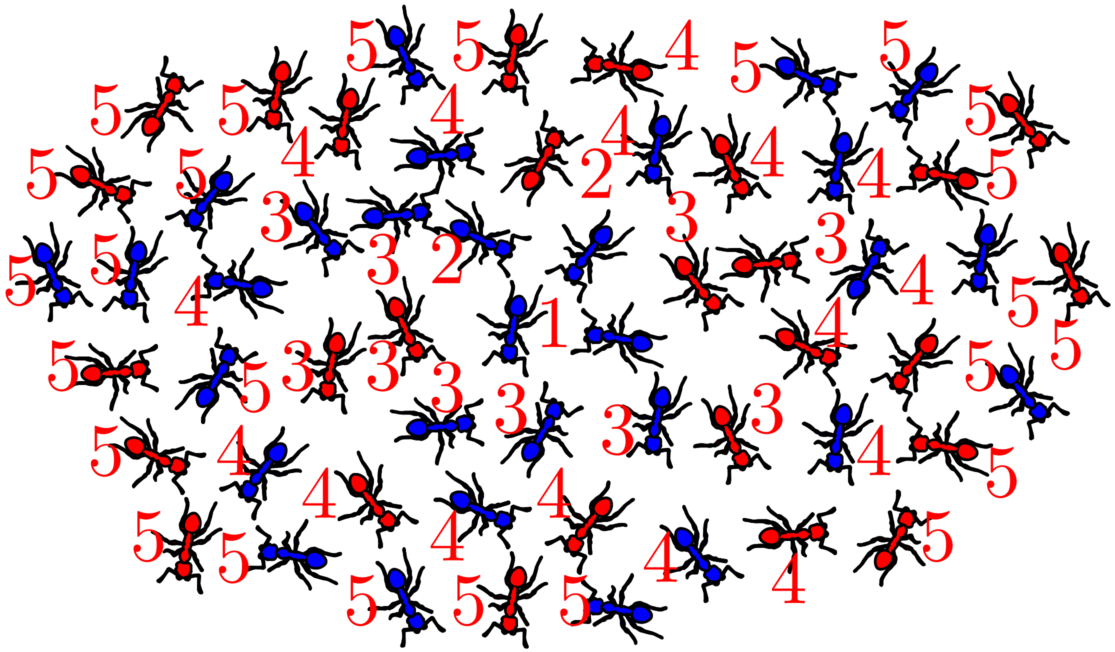


trivial
strategy

blue vs red:
$35/29 \approx 1.2$

[1] O. Feinerman, B. Haeupler, and A. Korman, “Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication,” Distrib. Comput., pp. 1–17, Jun. 2015.

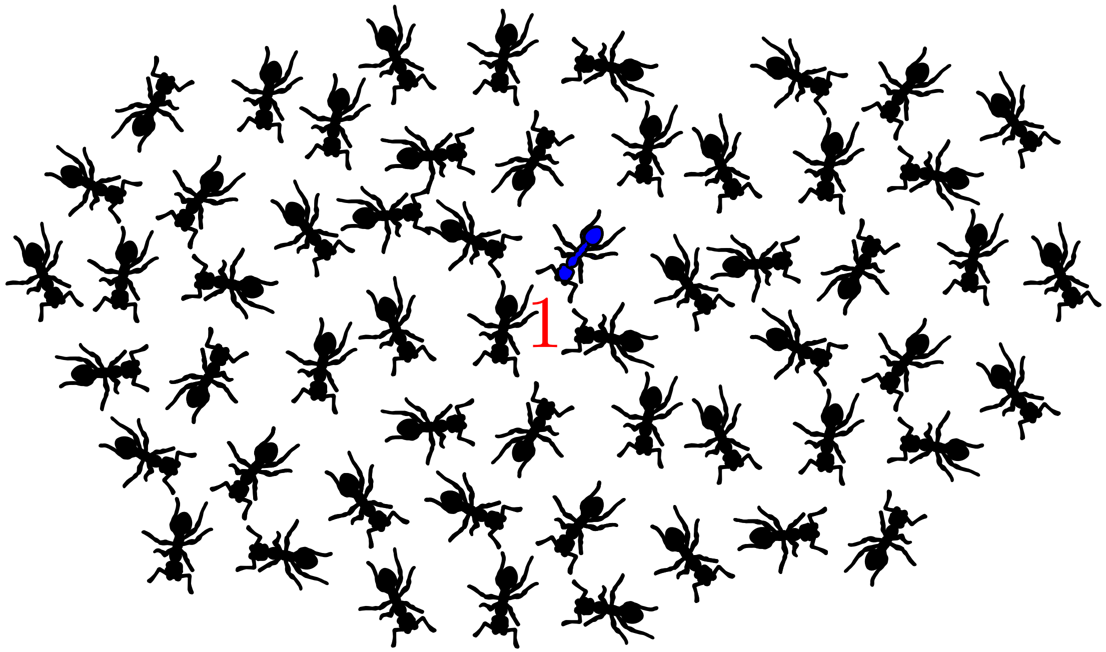
Previous Work: Breathe Before Speaking [1]



blue vs red:
$35/29 \approx 1.2$

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



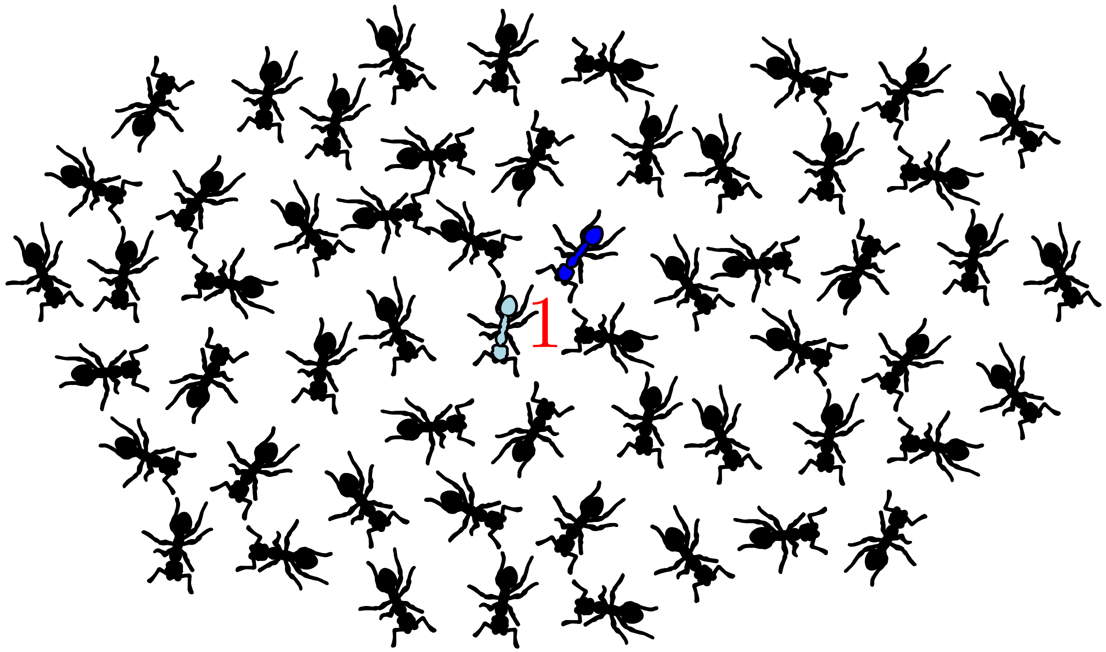
Stage 1: Spreading

blue vs red:
1/0

Idea: the "hops" a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



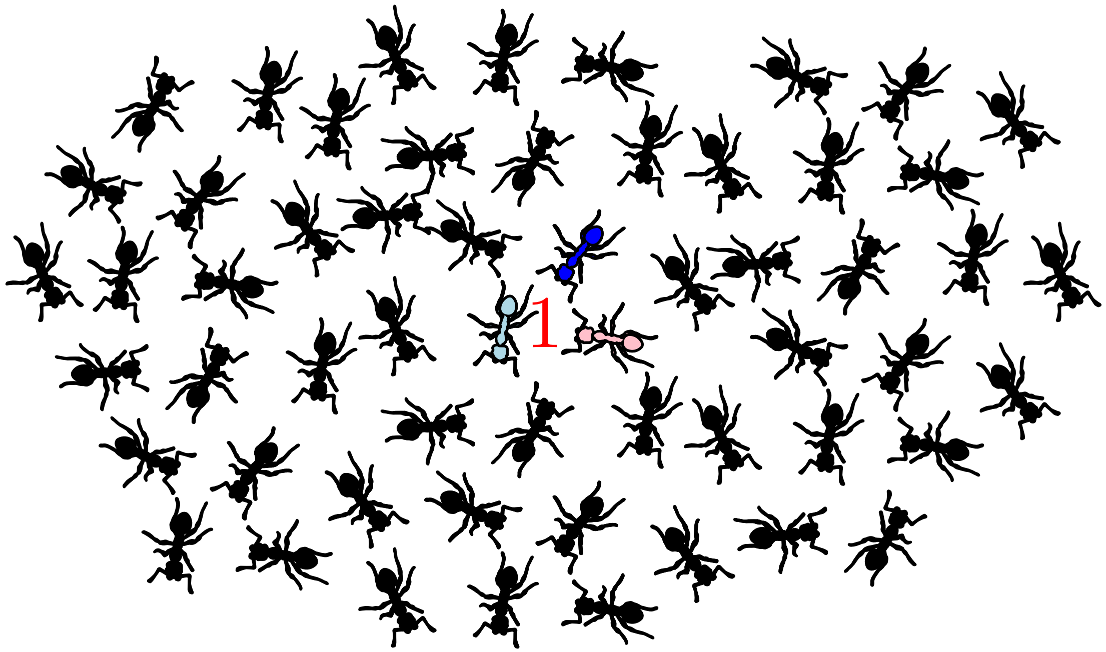
Stage 1: Spreading

blue vs red:
1/0

Idea: the “hops” a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, “Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication,” Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



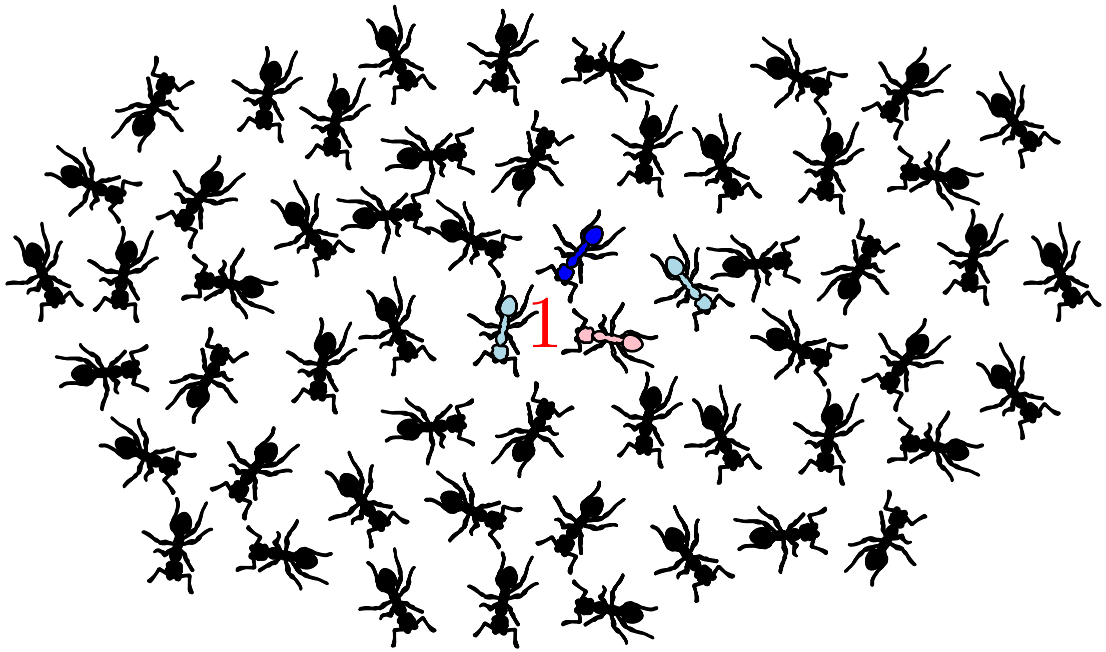
Stage 1: Spreading

blue vs red:
1/0

Idea: the “hops” a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, “Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication,” Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



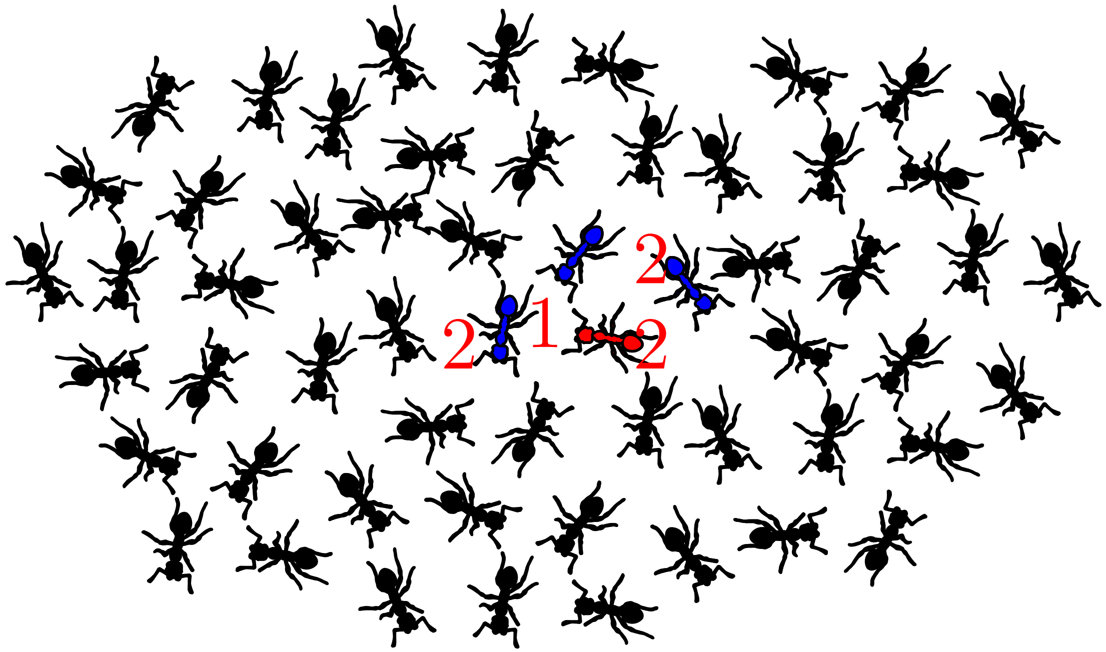
Stage 1: Spreading

blue vs red:
1/0

Idea: the "hops" a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



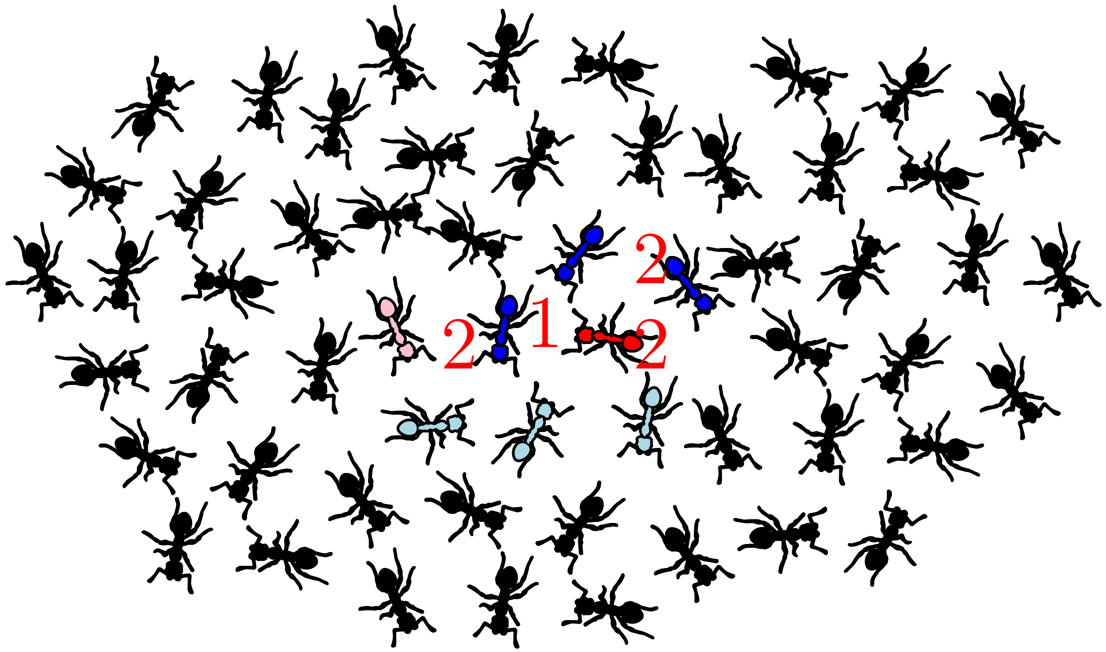
Stage 1: Spreading

blue vs red:
3/1

Idea: the "hops" a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



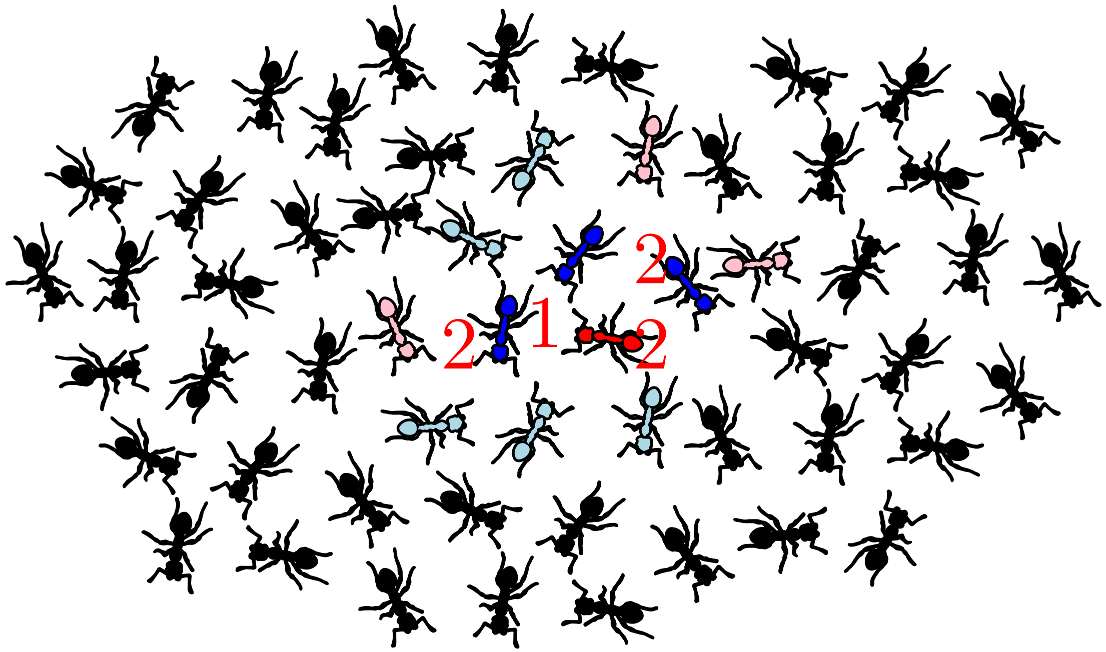
Stage 1: Spreading

blue vs red:
3/1

Idea: the "hops" a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



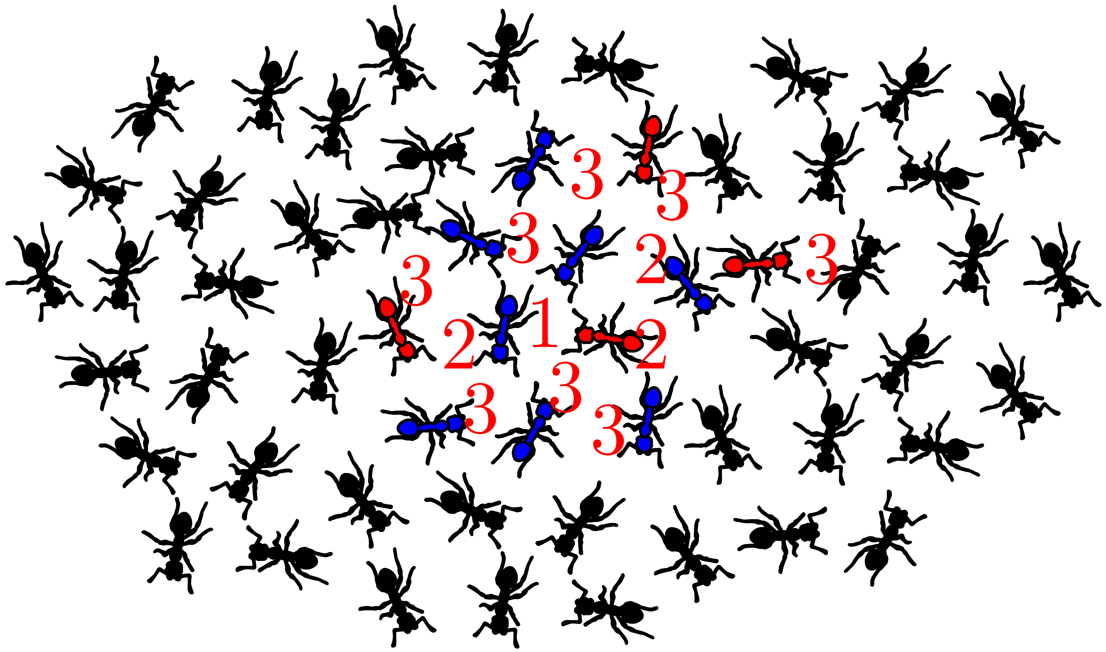
Stage 1: Spreading

blue vs red:
3/1

Idea: the “hops” a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, “Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication,” Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



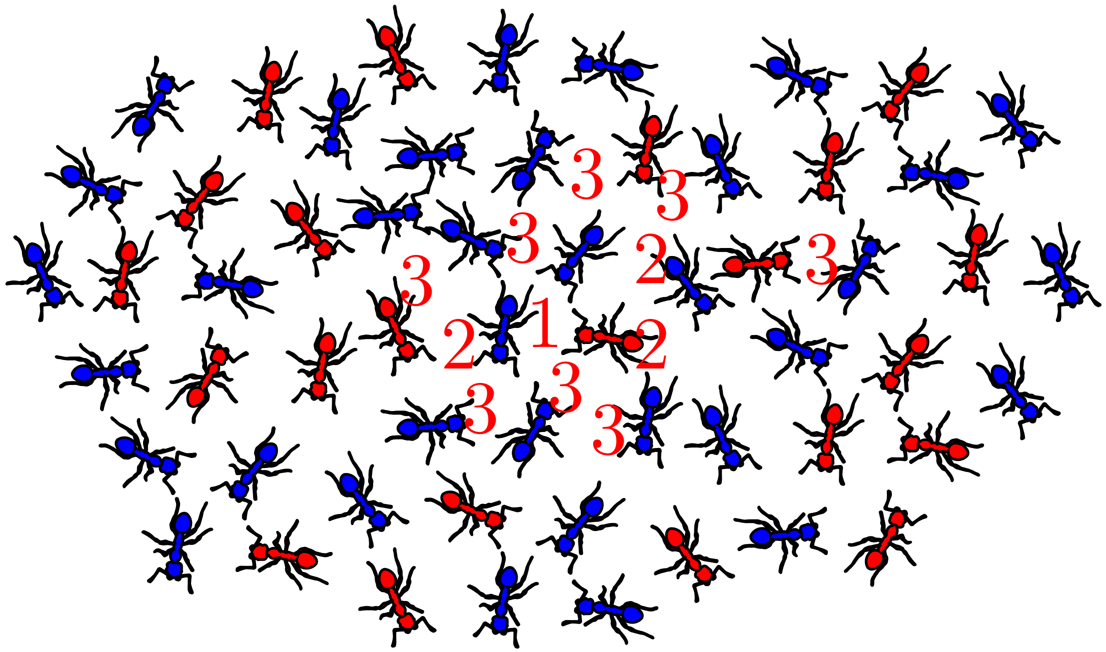
Stage 1: Spreading

blue vs red:
8/4

Idea: the "hops" a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Previous Work: Breathe Before Speaking [1]



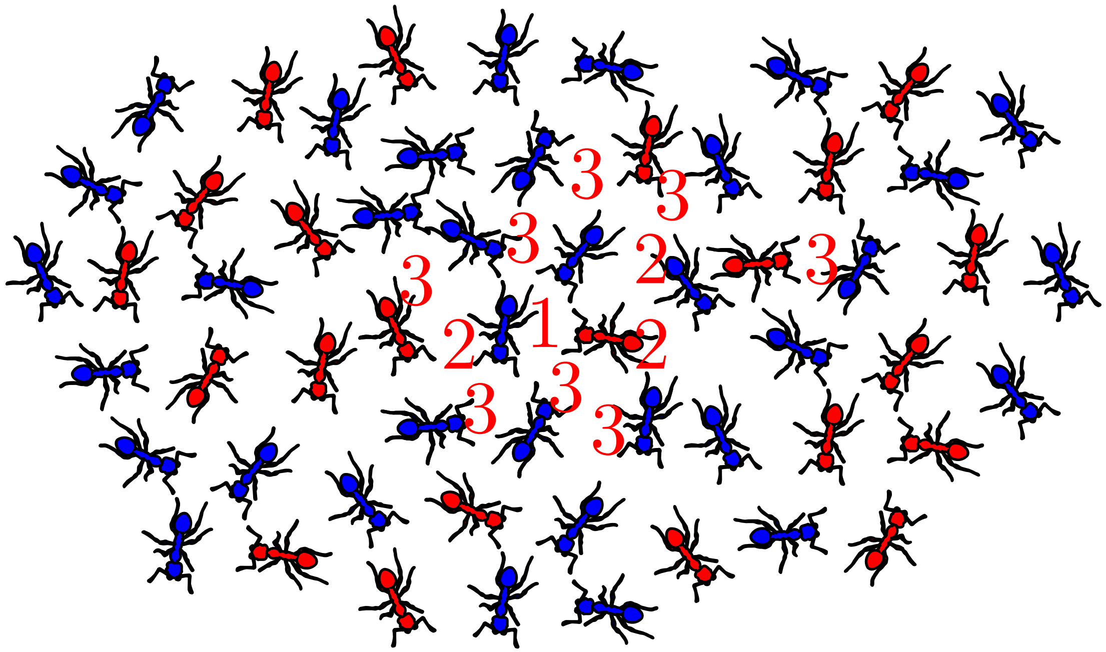
Stage 1: Spreading

blue vs red:
$40/24 \approx 1.7$

Idea: the “hops” a message does from source to agent deteriorate it; number of hops can be reduced with phases of waiting before spreading.

[1] O. Feinerman, B. Haeupler, and A. Korman, “Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication,” Distrib. Comput., pp. 1–17, Jun. 2015.

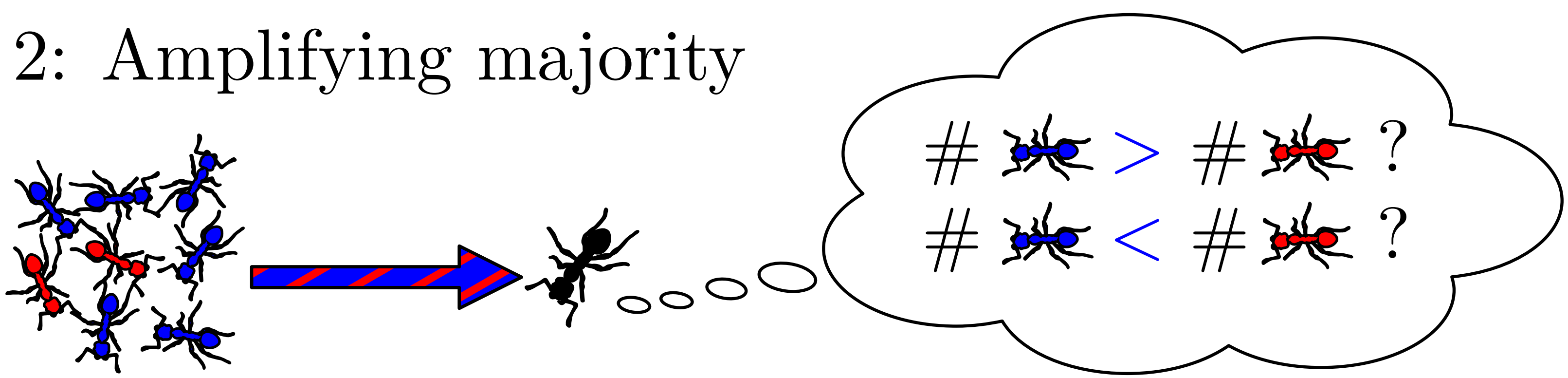
Previous Work: Breathe Before Speaking [1]



Stage 1: Spreading

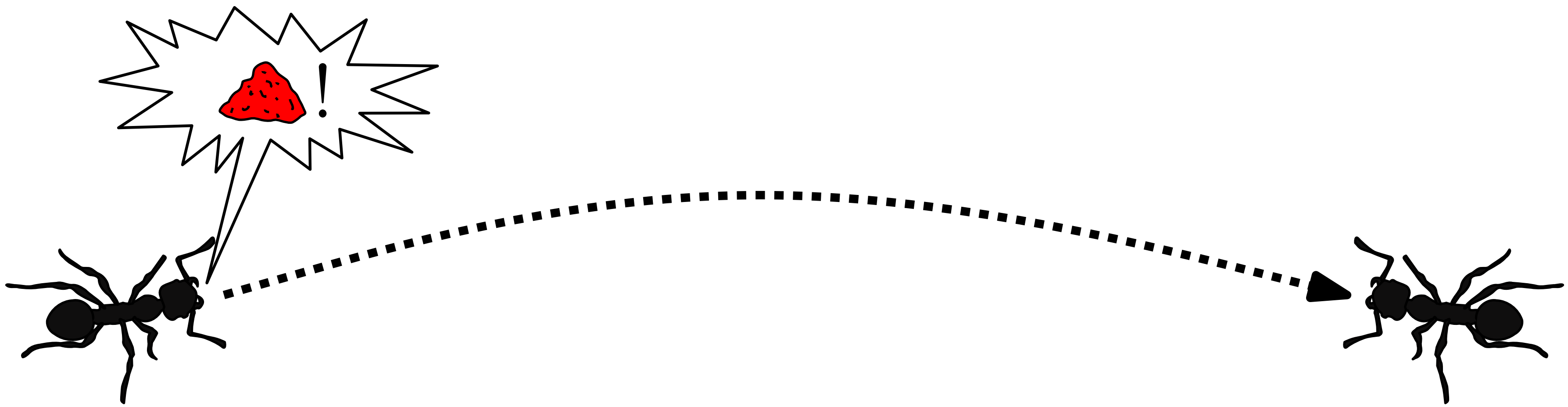
blue vs red:
$40/24 \approx 1.7$

Stage 2: Amplifying majority



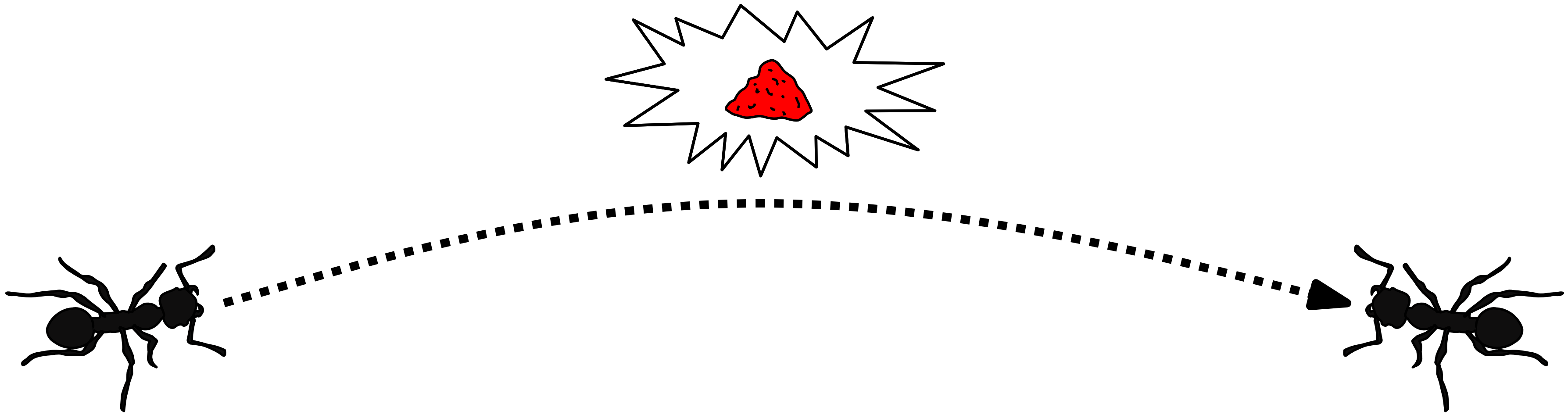
[1] O. Feinerman, B. Haeupler, and A. Korman, "Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication," Distrib. Comput., pp. 1–17, Jun. 2015.

Our Generalization: Multivalued Case [1]



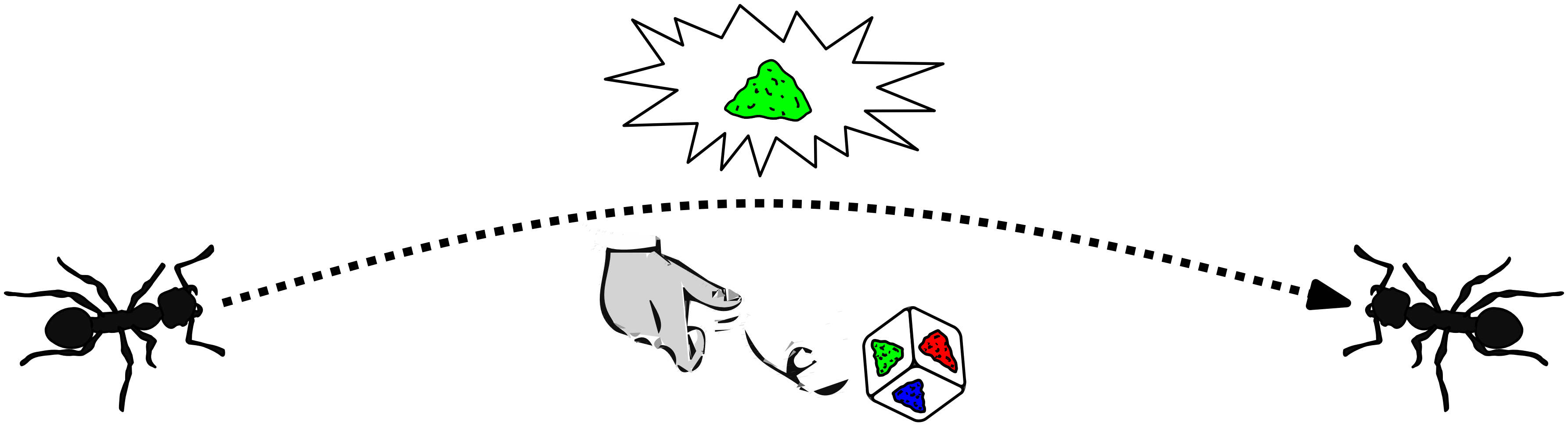
[1] P. Fraigniaud and E. Natale, “Noisy Rumor Spreading and Plurality Consensus,” in Proc. of ACM PODC, 2016.

Our Generalization: Multivalued Case [1]



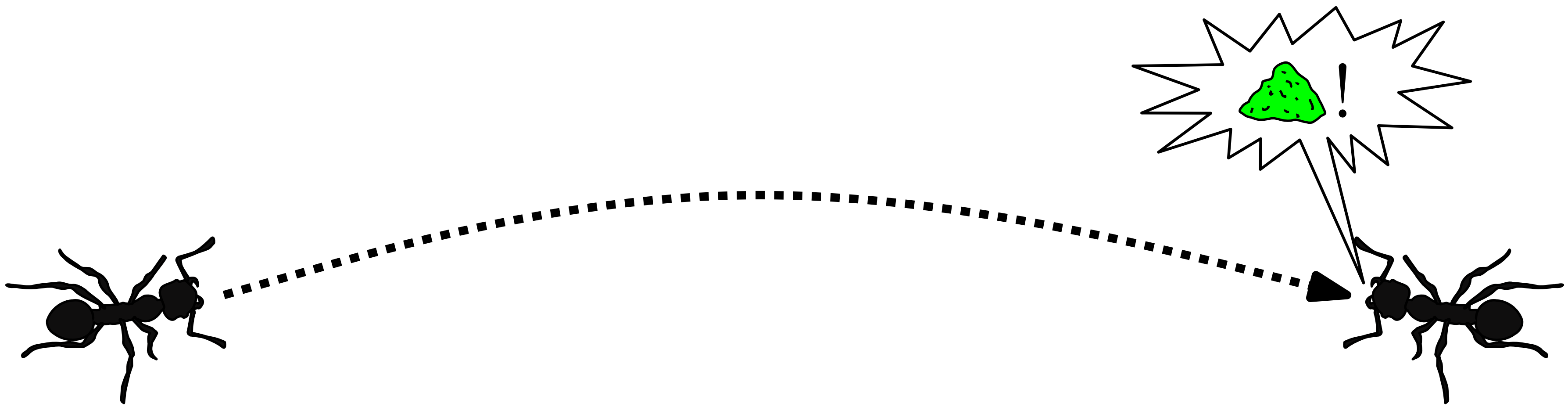
[1] P. Fraigniaud and E. Natale, "Noisy Rumor Spreading and Plurality Consensus," in Proc. of ACM PODC, 2016.

Our Generalization: Multivalued Case [1]



[1] P. Fraigniaud and E. Natale, “Noisy Rumor Spreading and Plurality Consensus,” in Proc. of ACM PODC, 2016.

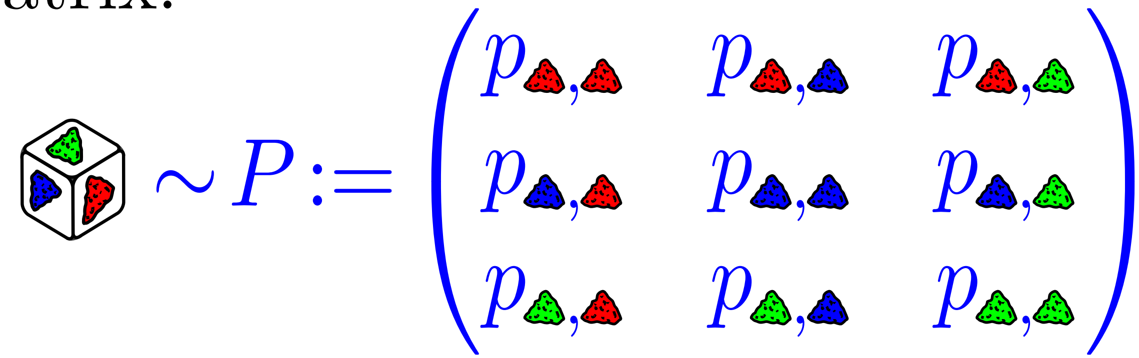
Our Generalization: Multivalued Case [1]

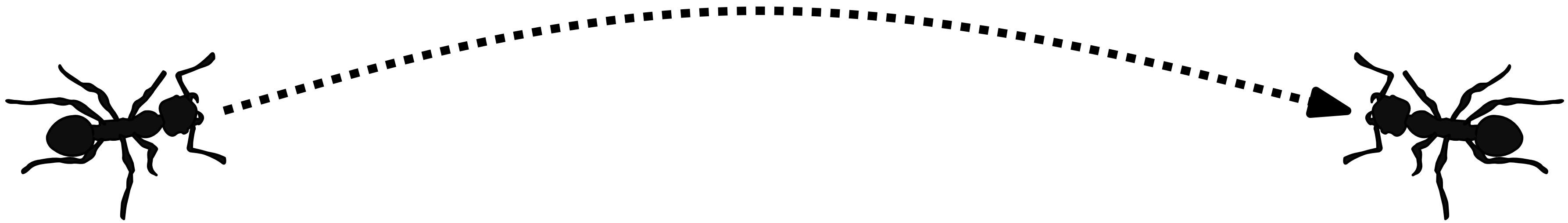


[1] P. Fraigniaud and E. Natale, "Noisy Rumor Spreading and Plurality Consensus," in Proc. of ACM PODC, 2016.

Our Generalization: Multivalued Case [1]

Noise Matrix:

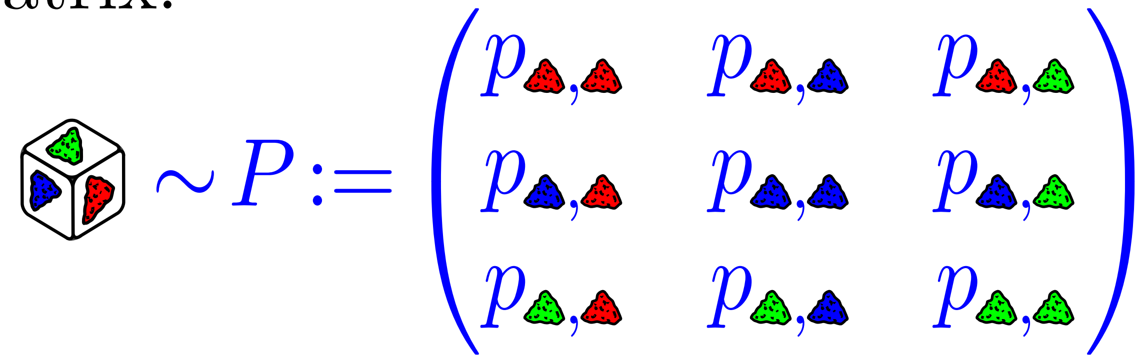


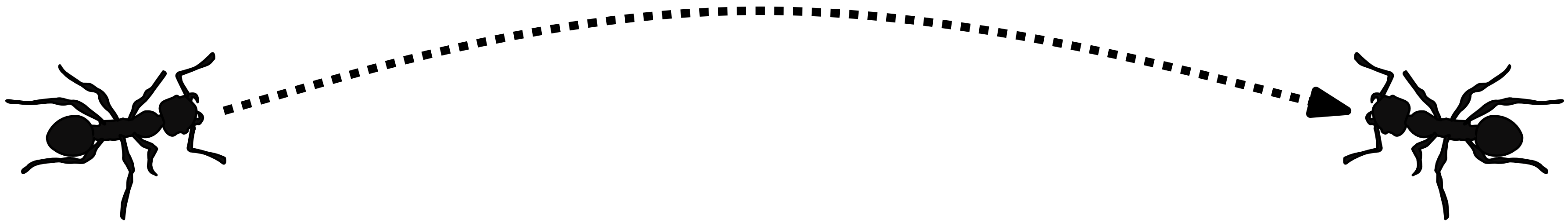


[1] P. Fraigniaud and E. Natale, "Noisy Rumor Spreading and Plurality Consensus," in Proc. of ACM PODC, 2016.

Our Generalization: Multivalued Case [1]

Noise Matrix:





Configuration **c** $:= (\#/n, \#/n, \#/n)$

δ-majority-biased configuration w.r.t. :

$$\#/n - \#/n > \delta$$
$$\#/n - \#/n > \delta$$

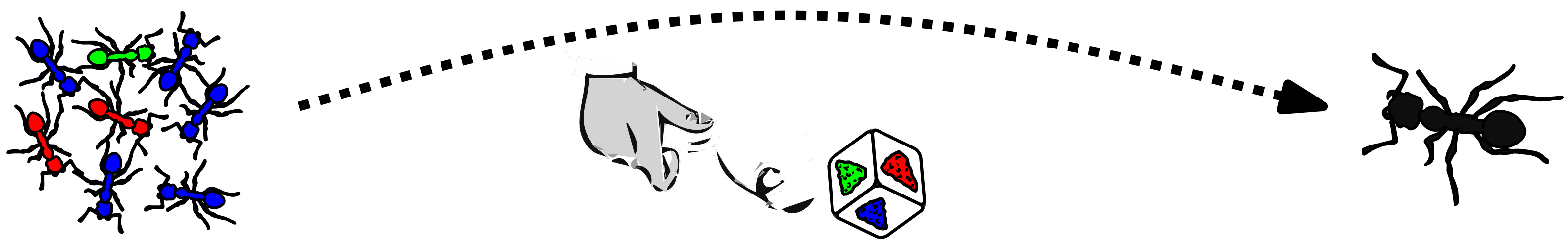
[1] P. Fraigniaud and E. Natale, "Noisy Rumor Spreading and Plurality Consensus," in Proc. of ACM PODC, 2016.

Majority-Preserving Matrix

Random sender in conf. $\mathbf{c}$

Noise acting according to matrix P

Message distributed as $c \cdot P$

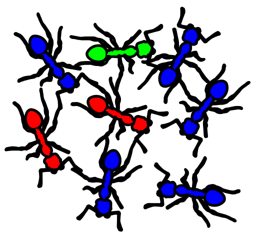


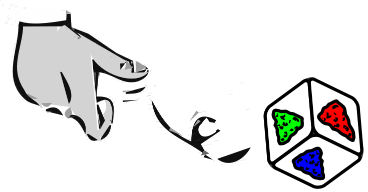
Majority-Preserving Matrix

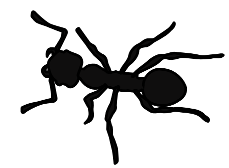
Random sender in conf. $\mathbf{c}$

Noise acting according to matrix P

Message distributed as $c \cdot P$







(ε, δ)-majority-preserving noise matrix:

$$(\mathbf{c}P)_{\color{blue}\blacktriangle} - (\mathbf{c}P)_{\color{red}\blacktriangle} > \varepsilon\delta$$

$$(\mathbf{c}P)_{\color{blue}\blacktriangle} - (\mathbf{c}P)_{\color{green}\blacktriangle} > \varepsilon\delta$$

Main Result

Theorem [1]. Let S be the initial set of agents with opinions in $[k]$. Suppose that S is $\delta = \Omega(\sqrt{\log n / |S|})$-majority-biased with $|S| = \Omega(\frac{\log n}{\epsilon^2})$ and the noise matrix P is (ϵ, δ)-majority-preserving. Then the plurality consensus problem can be solved in $O(\frac{\log n}{\epsilon^2})$ rounds w.h.p., with $O(\log \log n + \log \frac{1}{\epsilon})$ memory per node.

[1] P. Fraigniaud and E. Natale, "Noisy Rumor Spreading and Plurality Consensus," in Proc. of ACM PODC, 2016.

Main Result

Theorem [1]. Let S be the initial set of agents with opinions in $[k]$. Suppose that S is $\delta = \Omega(\sqrt{\log n/|S|})$-majority-biased with $|S| = \Omega(\frac{\log n}{\epsilon^2})$ and the noise matrix P is (ϵ, δ)-majority-preserving. Then the plurality consensus problem can be solved in $O(\frac{\log n}{\epsilon^2})$ rounds w.h.p., with $O(\log\log n + \log\frac{1}{\epsilon})$ memory per node.

$|S| = 1 \implies$ rumor spreading in $O(\frac{\log n}{\epsilon^2})$ rounds

[1] P. Fraigniaud and E. Natale, "Noisy Rumor Spreading and Plurality Consensus," in Proc. of ACM PODC, 2016.

Main Result

Theorem [1]. Let S be the initial set of agents with opinions in $[k]$. Suppose that S is $\delta = \Omega(\sqrt{\log n / |S|})$-majority-biased with $|S| = \Omega(\frac{\log n}{\epsilon^2})$ and the noise matrix P is (ϵ, δ)-majority-preserving. Then the plurality consensus problem can be solved in $O(\frac{\log n}{\epsilon^2})$ rounds w.h.p., with $O(\log \log n + \log \frac{1}{\epsilon})$ memory per node.

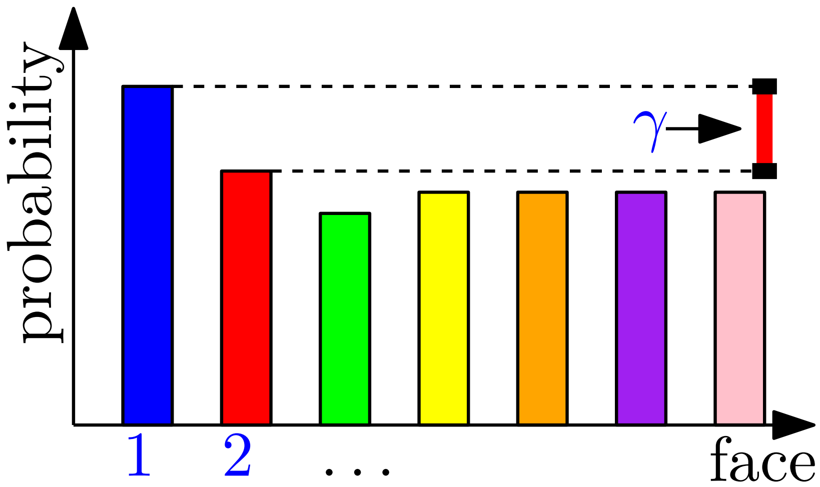
$|S| = 1 \implies$ rumor spreading in $O(\frac{\log n}{\epsilon^2})$ rounds

$$P = \begin{pmatrix} 1/2 + \varepsilon & 1/2 - \varepsilon \\ 1/2 - \varepsilon & 1/2 + \varepsilon \end{pmatrix} \implies \text{Feinerman et al.}$$

[1] P. Fraigniaud and E. Natale, "Noisy Rumor Spreading and Plurality Consensus," in Proc. of ACM PODC, 2016.

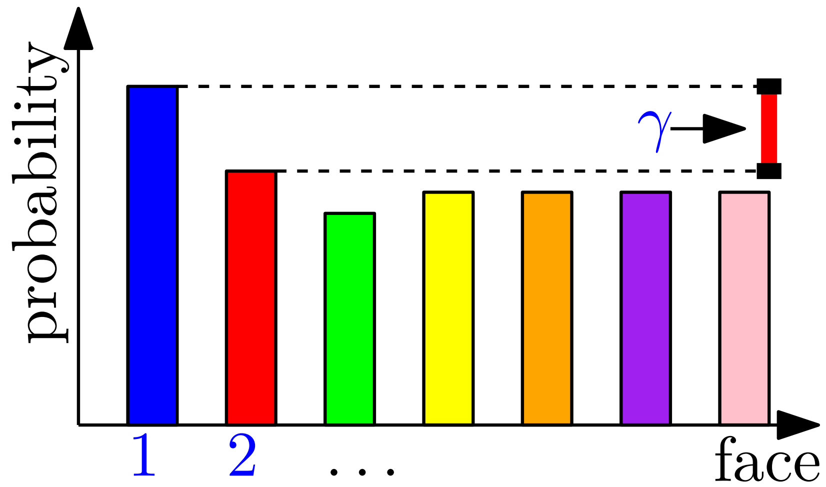
Probability Amplification: Binomial vs Beta

A dice with k faces is thrown ℓ times.



Probability Amplification: Binomial vs Beta

A dice with k faces is thrown ℓ times.



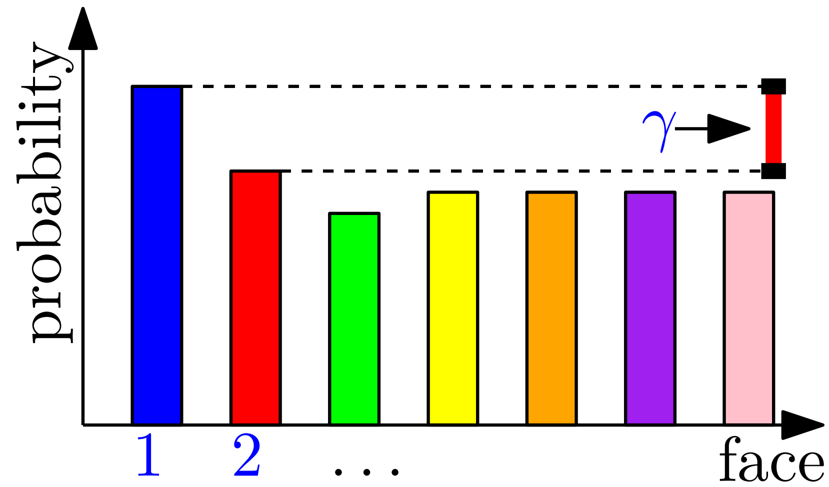
$\mathcal{M}$:= most frequent face in the ℓ throws (breaking ties at random).

For any $j \neq 1$

$$\Pr(\mathcal{M} = 1) - \Pr(\mathcal{M} = j) \geq \text{const} \cdot \sqrt{\ell}\,\gamma(1 - \gamma^2)^{\frac{\ell-1}{2}}$$

Probability Amplification: Binomial vs Beta

A dice with k faces is thrown ℓ times.



$\mathcal{M}$:= most frequent face in the ℓ throws (breaking ties at random).

For any $j \neq 1$

$$\Pr(\mathcal{M} = 1) - \Pr(\mathcal{M} = j) \geq \text{const} \cdot \sqrt{\ell}\,\gamma(1-\gamma^2)^{\frac{\ell-1}{2}}$$

Given $p \in (0,1)$ and $0 \leq j \leq \ell$ it holds:

$$\Pr(Bin(n,p) \leq j) = \sum_{j<i\leq\ell} \binom{\ell}{i} p^i (1-p)^{\ell-i}$$

$$= \binom{\ell}{j+1}(j+1)\int_0^p z^j (1-z)^{\ell-j-1}\,dz = \Pr(Beta(n-k, k+1) < 1 -$$

Open Problem: Multinomial vs Dirichlet?

Noisy $\mathcal{PUSH}$: ✓. Noisy $\mathcal{PULL}$?

δ-uniform noise criterion. Any time some agent u observes an agent v holding some message $m \in \Sigma$, the probability that u actually receives a message m' is at least δ, for any $m' \in \Sigma$.

Theorem [1]. For any rumor spreading protocol in the Noisy $\mathcal{PULL}$ model with δ-uniform noise, no agent can have a guess on the source's opinion which is correct with probability $\geq \frac{2}{3}$ in less than $\Omega(\frac{n\delta}{(1-2\delta)^2})$ rounds.

Ideas: Pearson's Lemma + Pinsker's inequality + chain rule for KL div. = hypothesis testing bounds for *adaptive* coin tossing

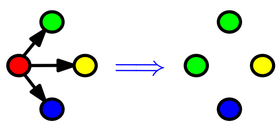
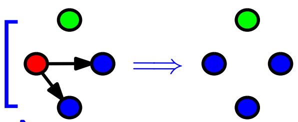
[1] L. Boczkowski, O. Feinerman, A. Korman, and E. Natale, "Limits for Rumor Spreading in stochastic populations," in Proc. of 9th ITCS, 2018.

Question 3/4

The techniques to study dynamics are ad-hoc arguments which do not generalize.

Can we perahps develop techniques to *compare* dynamics?

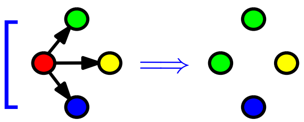
Voter vs 2-Choice vs 3-Majority

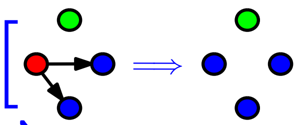
$\mathbb{E}\Big[$  $\Big]$ $=$ $\mathbb{E}\Big[$  $\Big]$ $= c_{red}\left(1+\frac{c_{red}}{n} - \sum_j \frac{c_j^2}{n^2}\right)$

3-Majority — 2-Choice

[1] P. Berenbrink, A. Clementi, R. Elsässer, P. Kling, F. Mallmann-Trenn, and E. Natale, "Ignore or Comply?: On Breaking Symmetry in Consensus," in Proc. of ACM PODC, 2017.

Voter vs 2-Choice vs 3-Majority

$\mathbb{E}\Big[$ 

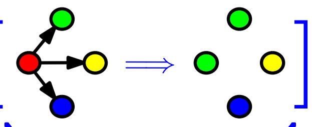
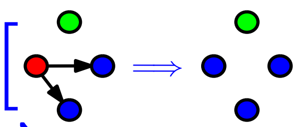
$\Big] = \mathbb{E}\Big[$ 

$\Big] = c_{red}\left(1+\frac{c_{red}}{n}-\sum_j \frac{c_j^2}{n^2}\right)$

Theorem (simplified) [1] **.** In the 2-Choice process, from the n-color conf., w.h.p. no color has support larger than $\gamma \log n$ for $\frac{n}{\gamma^2 \log n}$ rounds. Starting from *any* conf. $c \in \mathcal{C}$, 3-Majority reaches consensus w.h.p. in $\mathcal{O}(n^{3/4} \log^{7/8} n)$ rounds.

[1] P. Berenbrink, A. Clementi, R. Elsässer, P. Kling, F. Mallmann-Trenn, and E. Natale, "Ignore or Comply?: On Breaking Symmetry in Consensus," in Proc. of ACM PODC, 2017.

Voter vs 2-Choice vs 3-Majority

$\mathbb{E}[$  $]$ $=$ $\mathbb{E}[$  $]$ $= c_{red}\left(1 + \frac{c_{red}}{n} - \sum_j \frac{c_j^2}{n^2}\right)$

3-Majority 2-Choice

Theorem (simplified) [1] . In the 2-Choice process, from the n-color conf., w.h.p. no color has support larger than $\gamma \log n$ for $\frac{n}{\gamma^2 \log n}$ rounds. Starting from *any* conf. $c \in \mathcal{C}$, 3-Majority reaches consensus w.h.p. in $\mathcal{O}(n^{3/4} \log^{7/8} n)$ rounds.

Key theorem. Consider Voter and 3-Majority dynamics started from same initial conf c. There is a coupling s.t., after any round, the number of colors in Voter is at least that of 3-Majority.

[1] P. Berenbrink, A. Clementi, R. Elsässer, P. Kling, F. Mallmann-Trenn, and E. Natale, "Ignore or Comply?: On Breaking Symmetry in Consensus," in Proc. of ACM PODC, 2017.

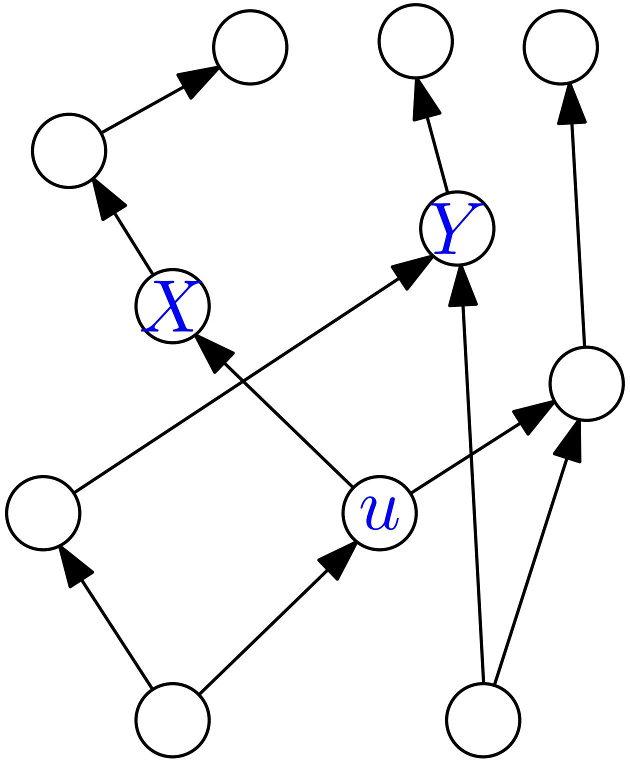
Majorization Theory and Strassen's Theorem

Folklore:
$\Pr(X > t) \geq \Pr(Y > t)$ then there is a coupling s.t. $\Pr(X \geq Y) = 1$.

Majorization Theory and Strassen's Theorem

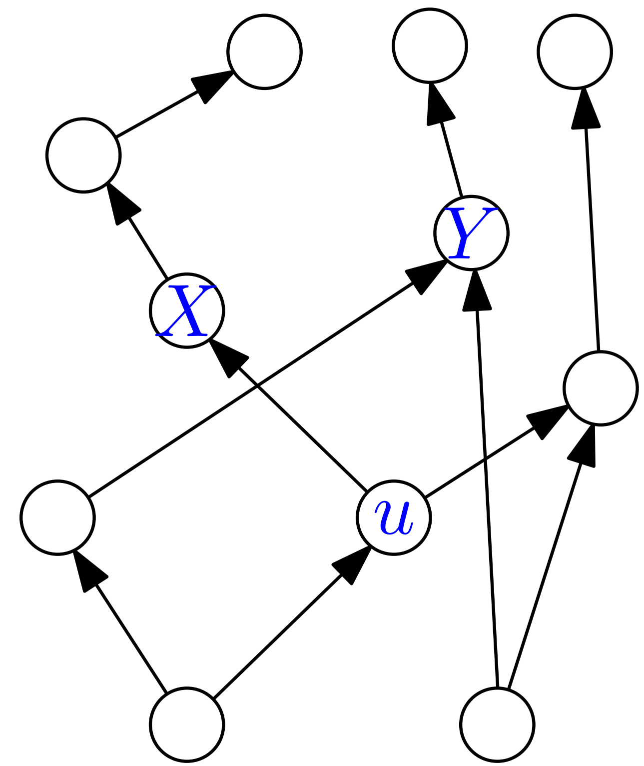
Folklore:
$\Pr(X > t) \geq \Pr(Y > t)$ then there is a coupling s.t. $\Pr(X \geq Y) = 1$.

Strassen's Theorem (finite case).
Given a DAG G and $X, Y \in V$ r.v.s, if
$\Pr($$X$ descendant of $u$$) \geq$
$\Pr($$Y$ descendant of $u$$)$ for each $u \in V$,
then there is a coupling s.t.
$\Pr($$X$ descendant of $Y$$) = 1$.



Majorization Theory and Strassen's Theorem

Folklore:
$\Pr(X > t) \geq \Pr(Y > t)$ then there is a coupling s.t. $\Pr(X \geq Y) = 1$.



Strassen's Theorem (finite case).
Given a DAG G and $X, Y \in V$ r.v.s, if
$\Pr(X$ descendant of $u) \geq$
$\Pr(Y$ descendant of $u)$ for each $u \in V$,
then there is a coupling s.t.
$\Pr(X$ descendant of $Y) = 1$.

Using tools from Majorization Theory: $\forall$conf c,
Pr(Conf. c' given by 3-Majority majorizes c) $\geq$
Pr(Conf. c' given by Voter majorizes c)
where majorize means, $\forall i$, $\sum_j^i {c'}_j \geq \sum_j c_j$ with colors in c' ordered decreasingly.

Question 4/4

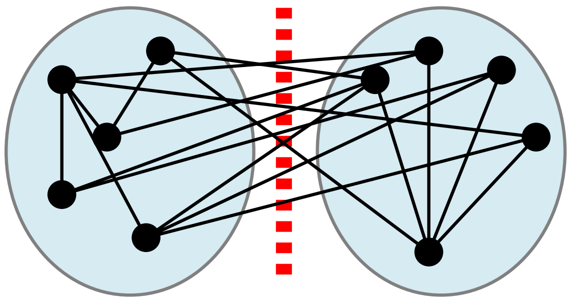
Dynamics can solve Consensus, Median, Majority, in a robust way, but this is trivial in centralized setting..

Can dynamics solve a problem non-trivial in centralized setting?

Community Detection

Min. Bisection Problem.
Given a graph G with $2n$ nodes. Find

$$S = \arg\min_{\substack{S \subset V \\ |S| = n}} E(S, V - S).$$



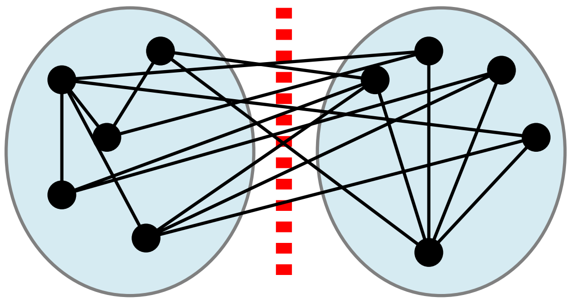
Min. Bisection is *NP-Complete* [1].

[1] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," Theoretical Computer Science, vol. 1, no. 3, pp. 237–267, Feb. 1976.

Community Detection

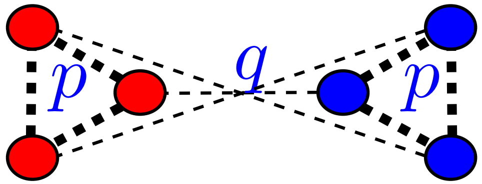
Min. Bisection Problem.
Given a graph G with $2n$ nodes. Find

$$S = \arg\min_{\substack{S \subset V \\ |S| = n}} E(S, V - S).$$



Min. Bisection is *NP-Complete* [1].

Stochastic Block Model. Two "communities" of equal size V_1 and V_2, each edge inside a community included with probability p, each edge across communities included with probability $q < p$.



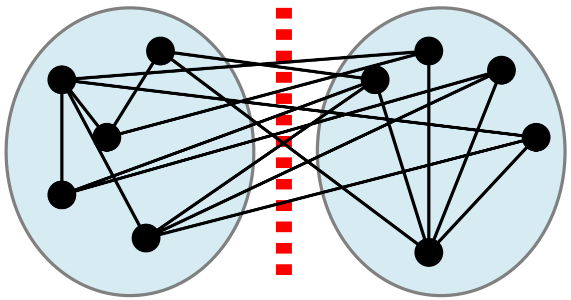
[1] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," Theoretical Computer Science, vol. 1, no. 3, pp. 237–267, Feb. 1976.

Community Detection

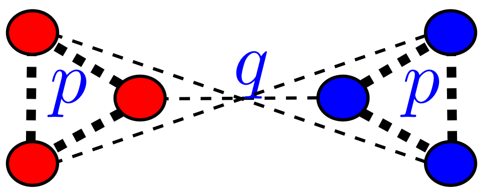
Min. Bisection Problem.
Given a graph G with $2n$ nodes. Find

$$S = \arg \min_{\substack{S \subset V \\ |S| = n}} E(S, V - S).$$

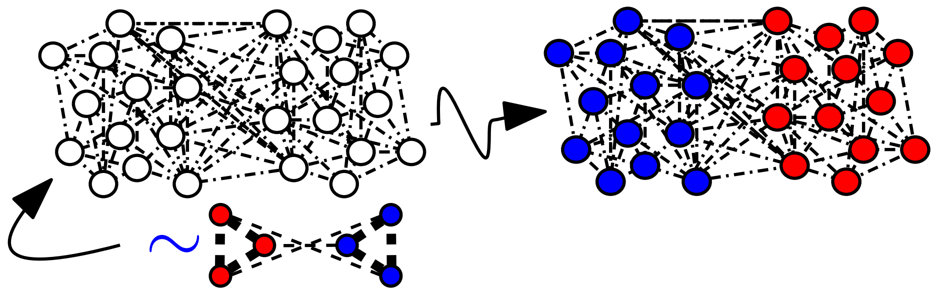
Min. Bisection is *NP-Complete* [1].



Stochastic Block Model. Two "communities" of equal size V_1 and V_2, each edge inside a community included with probability p, each edge across communities included with probability $q < p$.



Reconstruction problem. Given graph generated by SBM, find original partition.

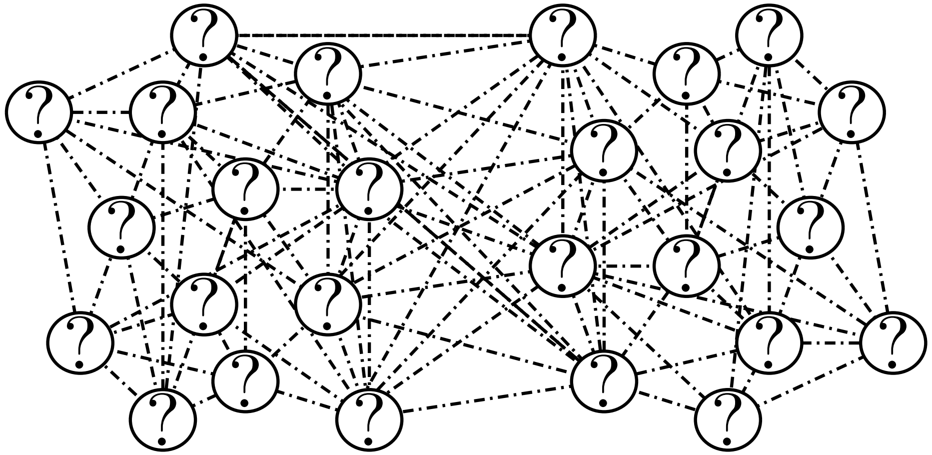


[1] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," Theoretical Computer Science, vol. 1, no. 3, pp. 237–267, Feb. 1976.

The Averaging Dynamics

Asynchronous Averaging Protocol:
At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.



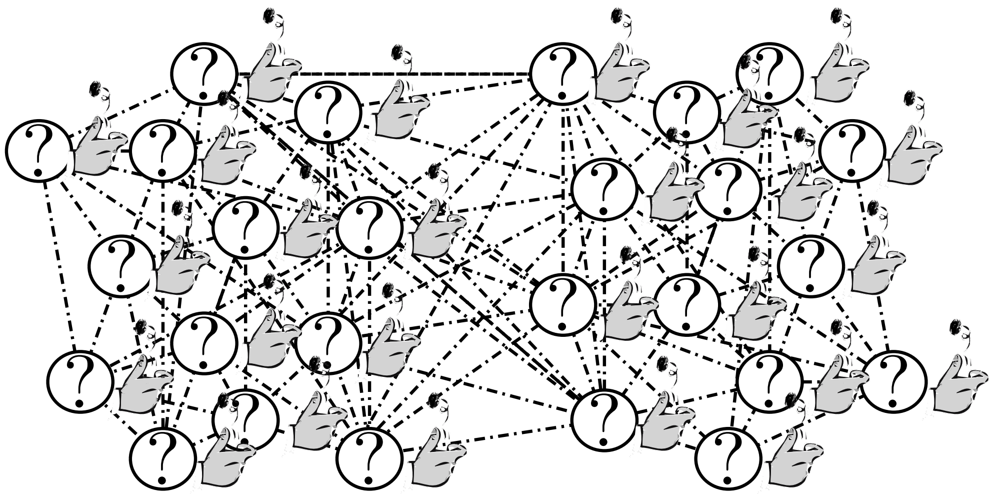
[1]L. Becchetti, A. Clementi, P. Manurangsi, E. Natale, F. Pasquale, P. Raghavendra, L. Trevisan, "Distributed Asynchronous Averaging for Community Detection," arXiv:1703.05045 [cs], Mar. 2017.

The Averaging Dynamics

Asynchronous Averaging Protocol:

At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.



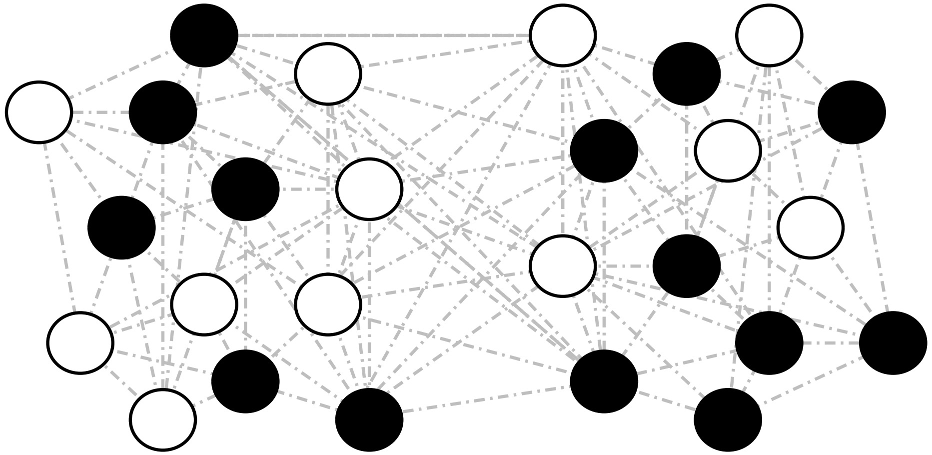
[1]L. Becchetti, A. Clementi, P. Manurangsi, E. Natale, F. Pasquale, P. Raghavendra, L. Trevisan, "Distributed Asynchronous Averaging for Community Detection," arXiv:1703.05045 [cs], Mar. 2017.

The Averaging Dynamics

Asynchronous Averaging Protocol:

At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.



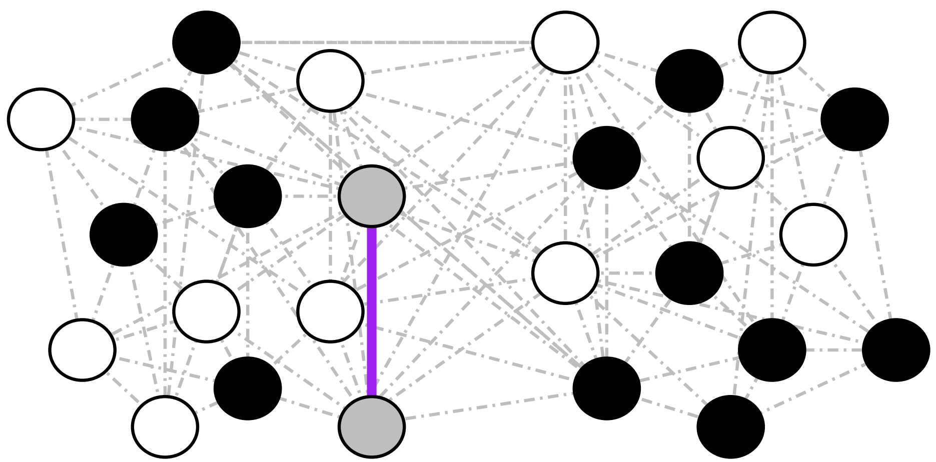
[1]L. Becchetti, A. Clementi, P. Manurangsi, E. Natale, F. Pasquale, P. Raghavendra, L. Trevisan, "Distributed Asynchronous Averaging for Community Detection," arXiv:1703.05045 [cs], Mar. 2017.

The Averaging Dynamics

Asynchronous Averaging Protocol:

At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.

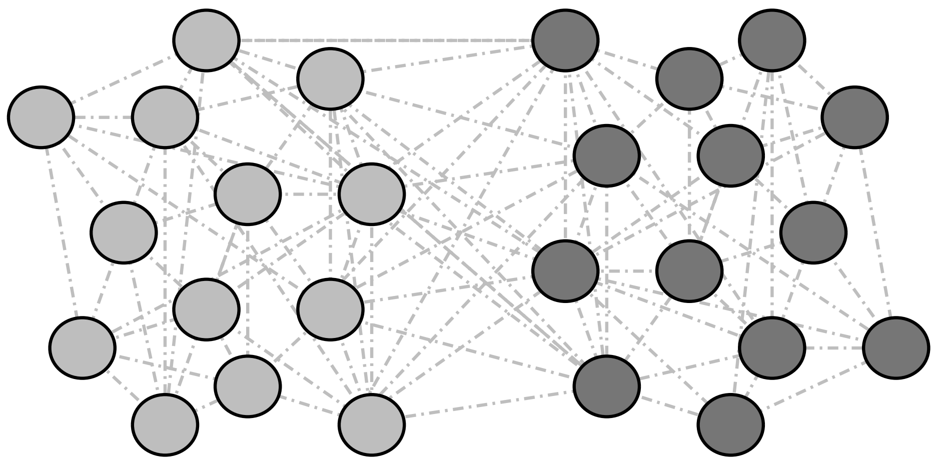


[1]L. Becchetti, A. Clementi, P. Manurangsi, E. Natale, F. Pasquale, P. Raghavendra, L. Trevisan, "Distributed Asynchronous Averaging for Community Detection," arXiv:1703.05045 [cs], Mar. 2017.

The Averaging Dynamics

Asynchronous Averaging Protocol:
At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.

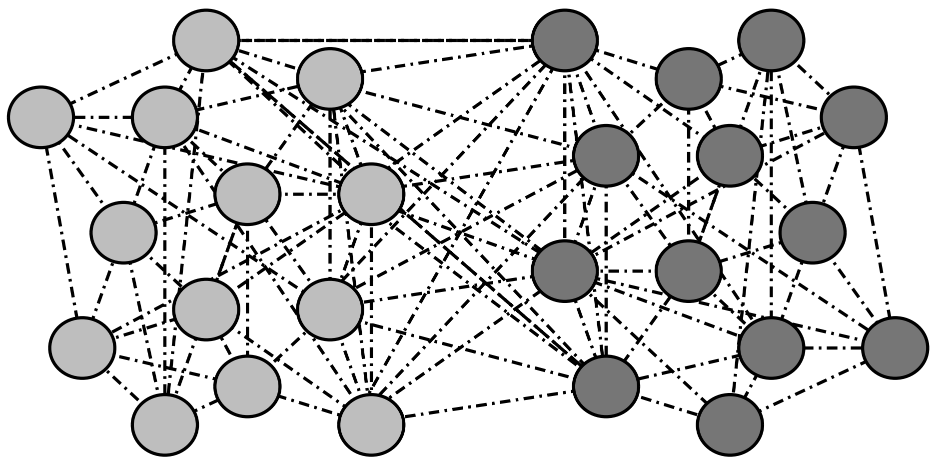


[1]L. Becchetti, A. Clementi, P. Manurangsi, E. Natale, F. Pasquale, P. Raghavendra, L. Trevisan, "Distributed Asynchronous Averaging for Community Detection," arXiv:1703.05045 [cs], Mar. 2017.

The Averaging Dynamics

Asynchronous Averaging Protocol:
At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.



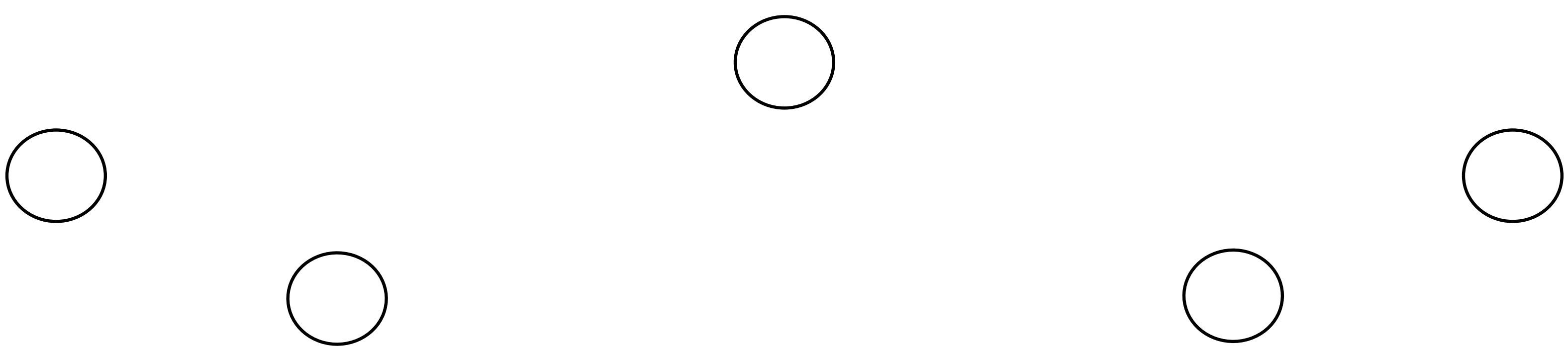
Theorem (Corollary of [1]). There exist τ_1, τ_2 s.t., if each node labels itself with the sign of the difference of its value at two activation times τ_1 and τ_2, then with prob. $1 - \epsilon$, after $O_\varepsilon(n \log n + \frac{n}{\lambda_2})$ rounds, we get a correct reconstruction up to an ϵ-fraction of nodes.

[1]L. Becchetti, A. Clementi, P. Manurangsi, E. Natale, F. Pasquale, P. Raghavendra, L. Trevisan, "Distributed Asynchronous Averaging for Community Detection," arXiv:1703.05045 [cs], Mar. 2017.

"$\mathbb{E}$[Averaging Dynamics]" ([1])

Al nodes at the same time:

- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{\text{-}1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

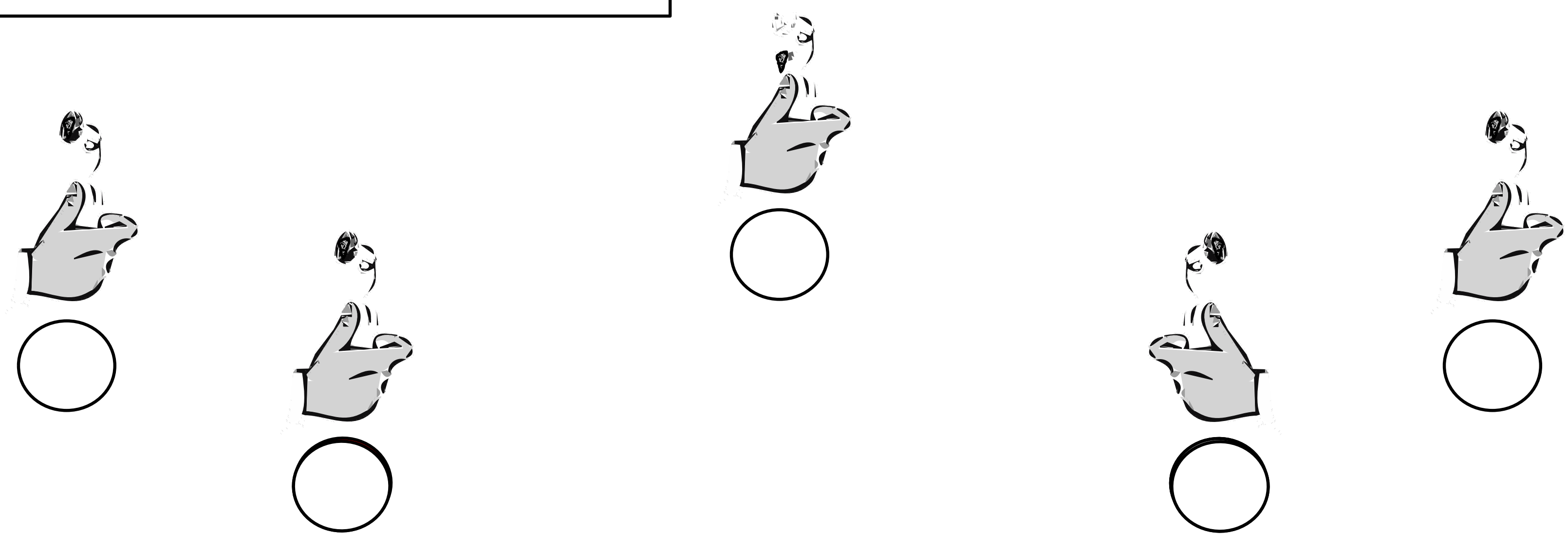


[1]L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, "Find Your Place: Simple Distributed Algorithms for Community Detection," in Proc. of 28th ACM-SIAM SODA, 2017.

"$\mathbb{E}$[Averaging Dynamics]" ([1])

Al nodes at the same time:

- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

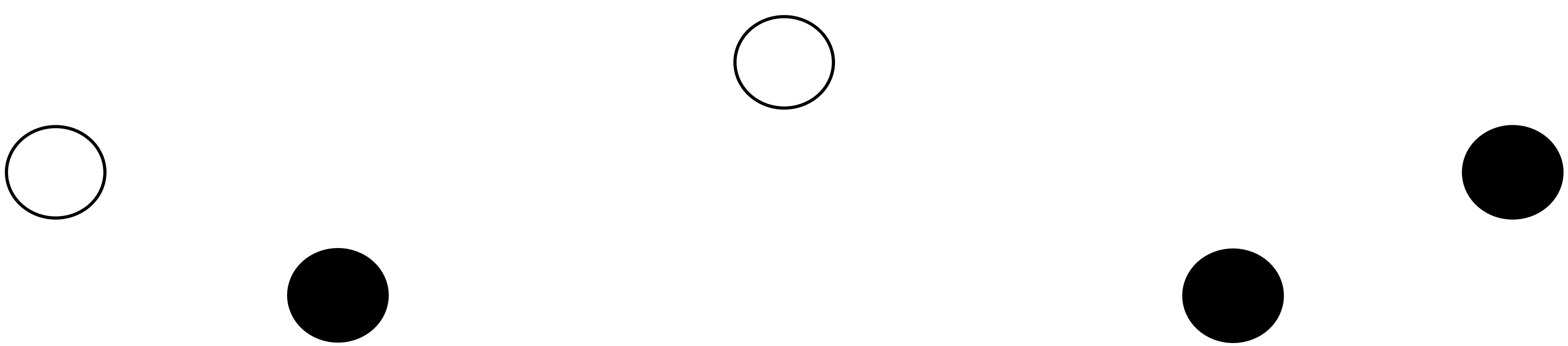


[1]L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, "Find Your Place: Simple Distributed Algorithms for Community Detection," in Proc. of 28th ACM-SIAM SODA, 2017.

“$\mathbb{E}$[Averaging Dynamics]” ([1])

Al nodes at the same time:

- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

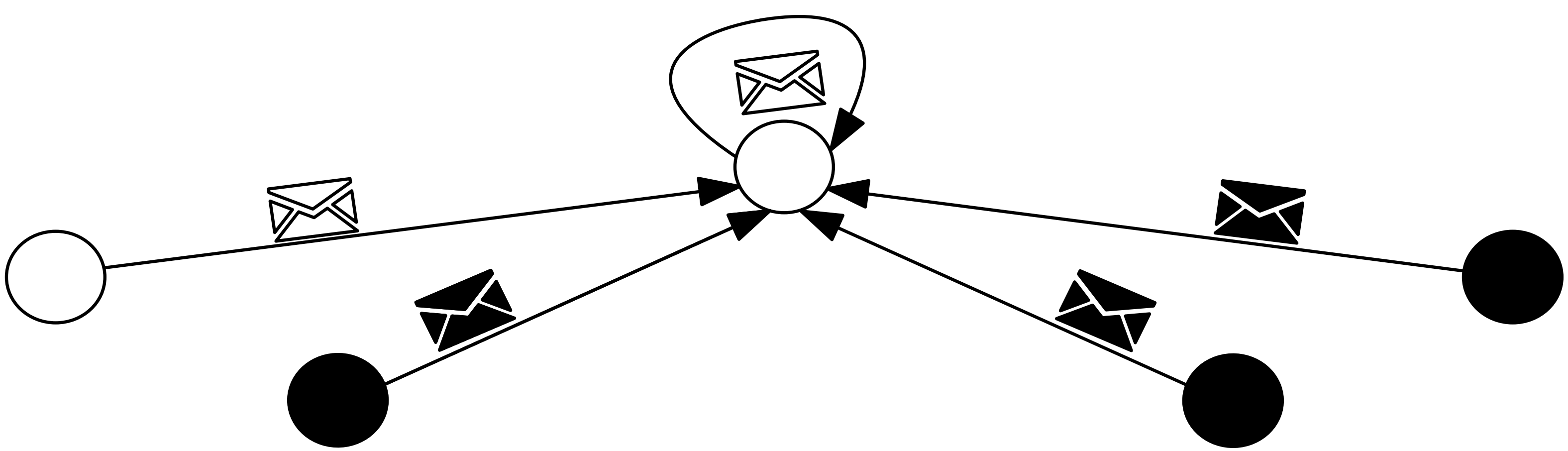


[1]L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, “Find Your Place: Simple Distributed Algorithms for Community Detection,” in Proc. of 28th ACM-SIAM SODA, 2017.

"$\mathbb{E}$[Averaging Dynamics]" ([1])

Al nodes at the same time:

- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

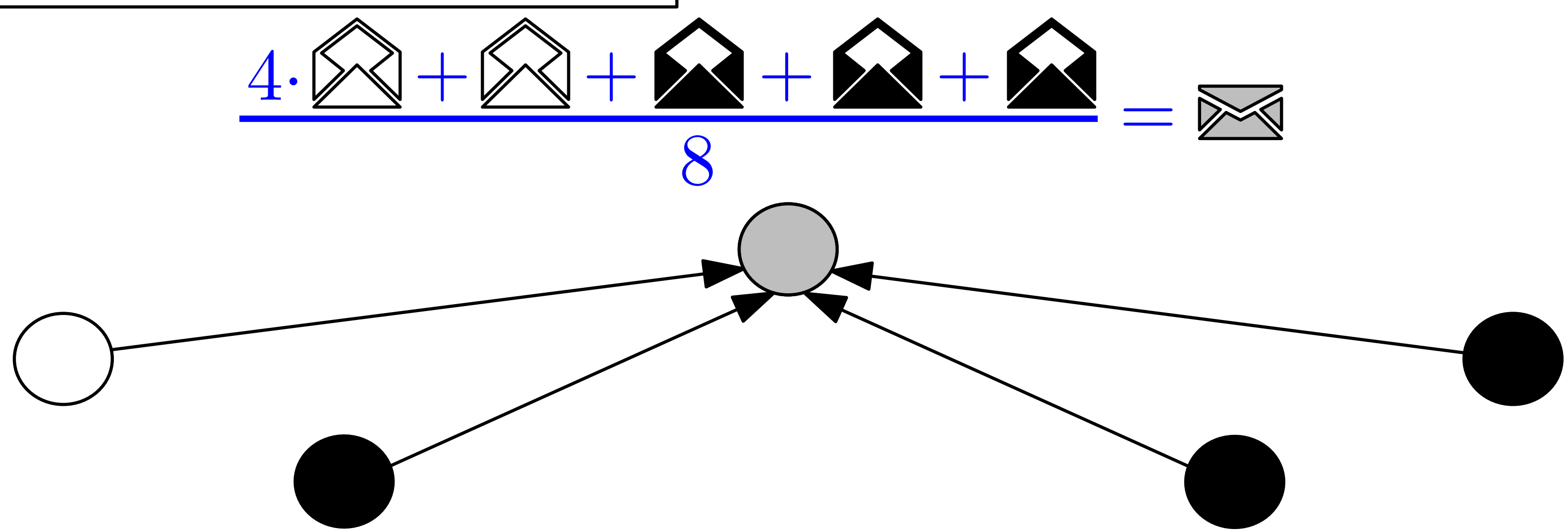


[1]L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, "Find Your Place: Simple Distributed Algorithms for Community Detection," in Proc. of 28th ACM-SIAM SODA, 2017.

"$\mathbb{E}$[Averaging Dynamics]" ([1])

Al nodes at the same time:

- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

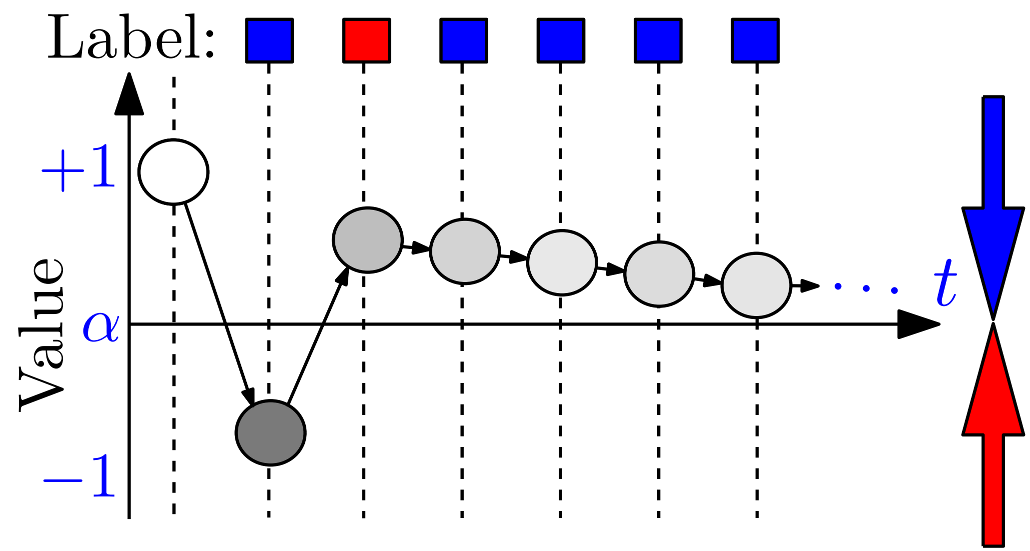


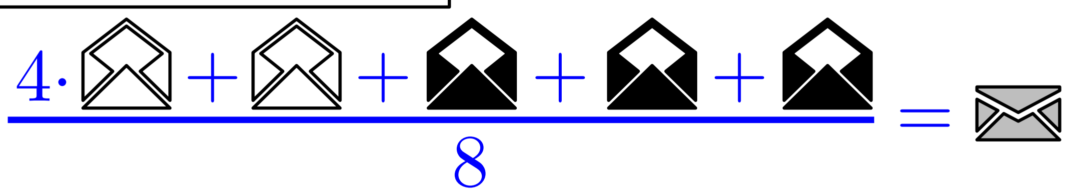
[1]L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, "Find Your Place: Simple Distributed Algorithms for Community Detection," in Proc. of 28th ACM-SIAM SODA, 2017.

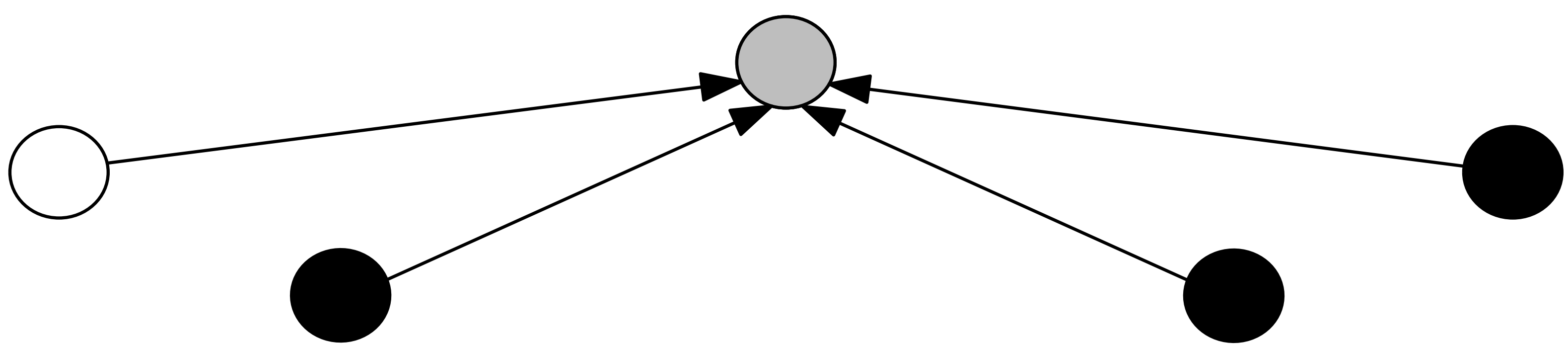
“$\mathbb{E}$[Averaging Dynamics]” ([1])

Al nodes at the same time:

- At $t = 0$, randomly pick value $x^{(t)} \in \{\textbf{+1}, \textbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

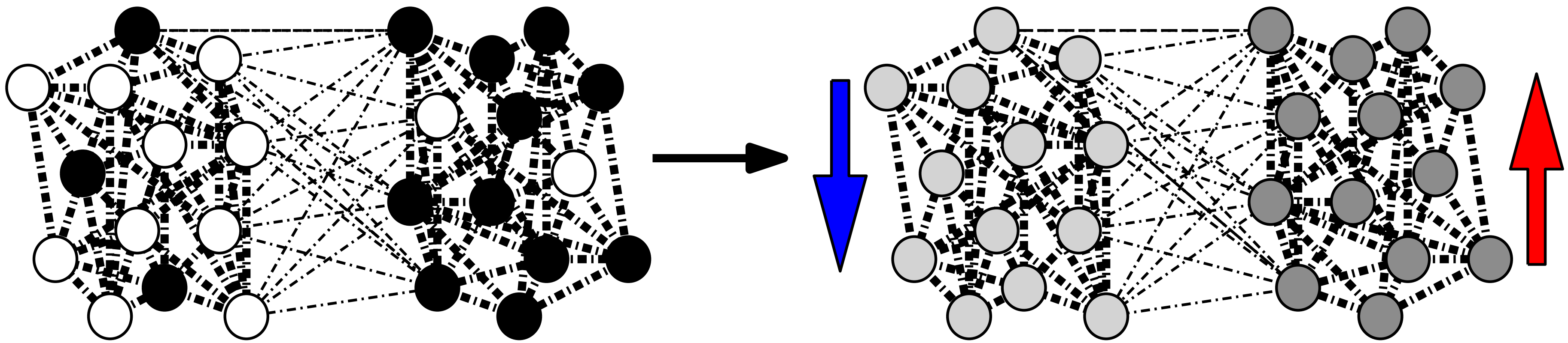




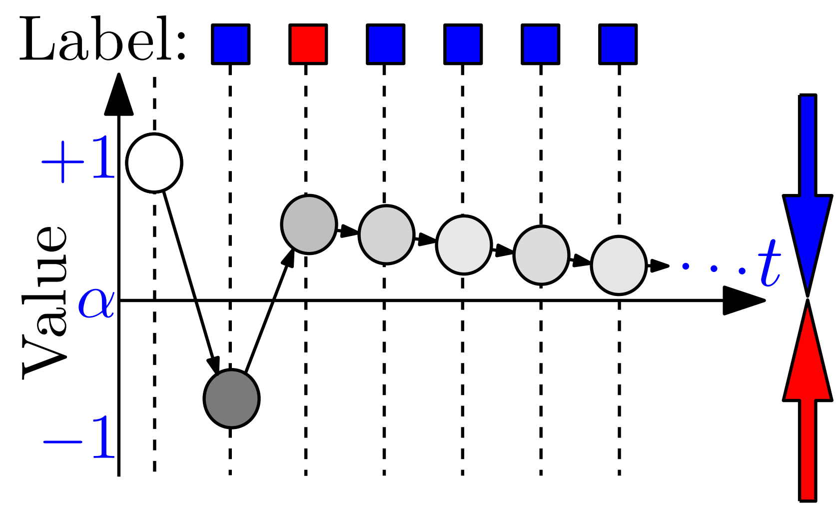


[1]L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, “Find Your Place: Simple Distributed Algorithms for Community Detection,” in Proc. of 28th ACM-SIAM SODA, 2017.

Community Detection via (Parallel) Averaging



Theorem (Informal) [1].
$G = (V_1 \dot{\bigcup} V_2, E)$ s.t.
i) $\chi = \mathbf{1}_{V_1} - \mathbf{1}_{V_2}$ close to right-eigenvector of eigenvalue λ_2 of transition matrix of G, and
ii) gap between λ_2 and λ_3 sufficiently large,
then Averaging (approximately) identifies (V_1, V_2).



[1]L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan, "Find Your Place: Simple Distributed Algorithms for Community Detection," in Proc. of 28th ACM-SIAM SODA, 2017.

We provide 4 Answers

- Can dynamics be used to perform algorithmically-interesting tasks?

  They can efficiently compute median, majority, average. (Problem: quantiles?)

- What are the minimal model requirements which allow effective information spreading?

  Self-stabilizing scenarios can allow very small messages. When noisy, active or passive communication is a big deal.

- Can we develop a *comparative* approach to dynamics?

  We can ensure the existence of a coupling among some dynamics. Work in progress on generalizing techniques.

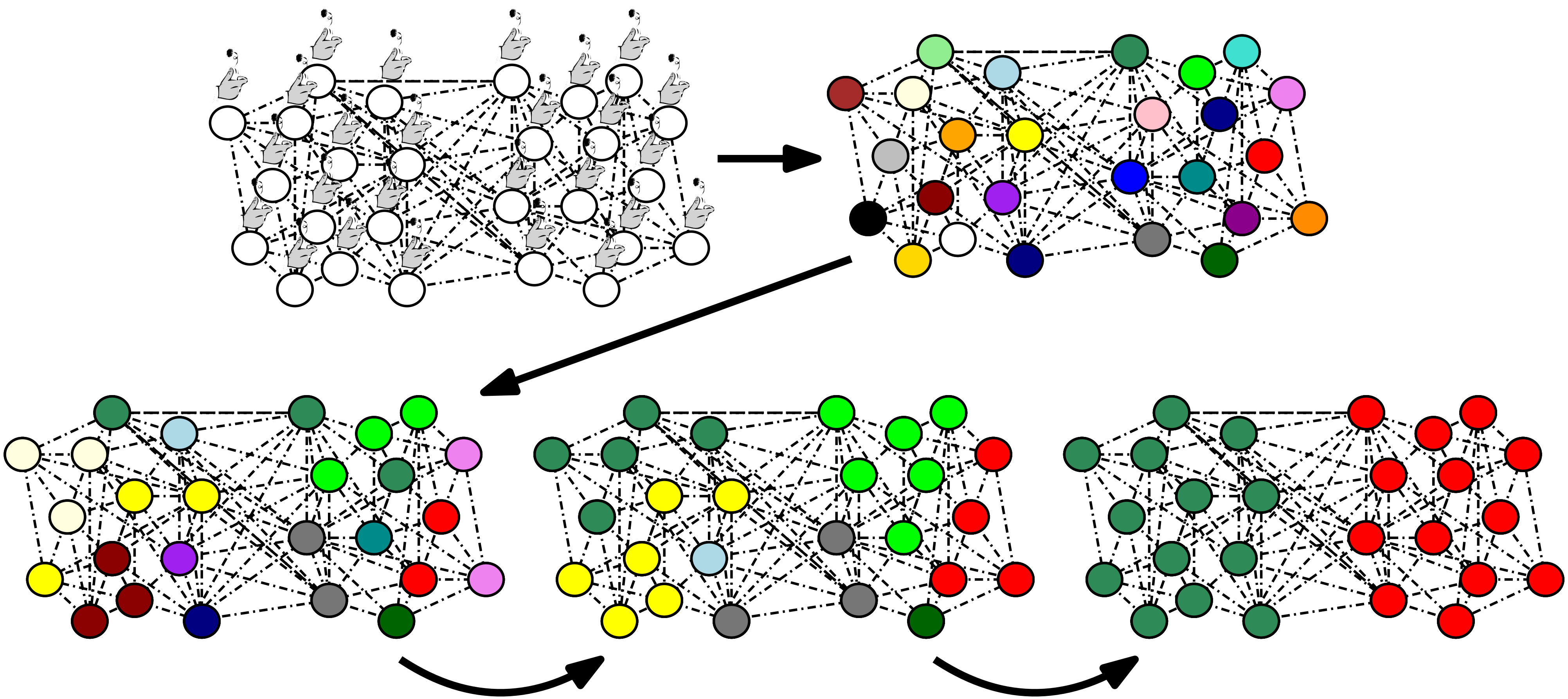
- Can dynamics solve problems which are *non-trivial* even in centralized setting?

  The averaging dynamics *shows* denser clusters. Doing the same for 3-Majority would be the first rigorous result on Label Propagation Algorithms.

(More on analyzing LPAs)

Averagins is a "linearization" of Label Propagation Algorithms:

- Each node initially sample a random color, then
- at each round, each node switch to the majority label of a sample of neighbors.



Conclusions

It is important to study systems
in-between interacting-particle systems
and human-made ones.

TCS can analyze dynamics,
helping to understand principles behind
complex systems' ability to compute in
simple chaotic ways.

Thank You!