# The Distributed Computing Column

Seth Gilbert
National University of Singapore
seth.gilbert@comp.nus.edu.sg

This month, the Distributed Computing Column is featuring Robin Vacus, winner of the 2024 Principles of Distributed Computing Doctoral Dissertation Award. His thesis on "Algorithmic Perspectives to Collective Natural Phenomena" explores how distributed computing techniques can help to understand biological processes found in the real world. It examines a variety of different phenomena, including problems in synchronization (i.e., agreement), problems in task selection, and problems in cooperative behavior. Quoting from the award citation: "Dr. Vacus' thesis provides an inspiring overview of the questions studied, and employs a wide range of tools and techniques, involving probabilistic analysis, control theory, statistics and game theory, and computer simulations."

This column focuses on the question of "bit dissemination," a basic agreement problem in which a single source wants to disseminate one bit information to all the agents in the system. Specifically, it discusses several different self-stabilizing protocols for achieving dissemination based only on passive communication, and it explores issues of limited memory and limited communication (i.e., sample size).

*The Distributed Computing Column is particularly interested in contributions that propose interesting new directions and summarize important open problems in areas of interest. If you would like to write such a column, please contact me.*

# Minimal Requirements for Bit-Dissemination with Passive Communication

Robin Vacus

Multi-agent systems often face two closely related challenges: achieving consensus among agents (i.e., making sure they take consistent decisions), and aggregating local information about their environment. Indeed, reaching approximate agreement is often a prerequisite of collective decision-making, while the overall performance of the group depends on how effectively local information is processed. For example, consider a group of ants carrying a heavy load. In the first place, in order to prevent forces from canceling each-other and successfully move the object, almost all ants in the group must push in the same direction – which requires a degree of consensus. In addition, information about an optimal route to the nest, which may be distributed unevenly among the ants, must be processed in order to enable the group to reach its destination quickly [11, 8]. These tasks become especially difficult when the system is susceptible to faults or when interactions between agents are noisy and constrained. This is particularly evident in biological ensembles, where individuals may have a wide range of internal computational abilities, but often limited communication capacity. Addressing these challenges and limitations in a biologically relevant way necessitates specific modeling approaches, which is why the bit-dissemination problem was introduced [4].

## 1 The Bit-Dissemination Problem

**Problem definition.** We consider a discrete time process over a fully-connected network of $n$ agents. In round $t$, each agent $i$ holds an *opinion* $X_t^{(i)} \in \{0, 1\}$. In each execution, one of these opinions is called *correct* and denoted $z$. The group contains one *source agent* $i^\star$ which knows which opinion is correct, and must always remain with it: for every $t$, $X_t^{(i^\star)} = z$.

We adopt the basic $\mathcal{PULL}$ model of communication: each time a non-source agent is *activated*, it obtains a random vector $S \in \{0, 1\}^\ell$, in which every element is equal to 1 with probability $x_t := \frac{1}{n} \sum_i X_t^{(i)}$ (the proportion of agents with opinion 1 in that round) and 0 otherwise. This is equivalent to sampling the opinions of $\ell$ other agents taken uniformly at random. Typically, the *sample size* $\ell$ shall be negligible compared to the size of the group $n$. After receiving $S$ and depending on the protocol, the activated agent may decide to update its opinion $X_t^{(i)}$ and memory state. We distinguish between two different activation settings: the *parallel* setting, where all agents are activated simultaneously in every round, and the *sequential* setting, where only one non-source agent selected uniformly at random is activated.

Relying only on the passive presence of the source, the goal of non-source agents is to reach a consensus on the correct opinion as fast as possible, and remain with it forever. Moreover, we restrict attention to *self-stabilizing* protocols, able to converge fast regardless of the initial configuration of the system. Specifically, a protocol is considered correct if, starting from any initial distribution of opinions (including the correct opinion $z$) and any initial memory states of the agents, it reaches a configuration where all $X_t^{(i)}$ are equal to $z$ within poly-logarithmic time, with high probability.

**Key quantities.** Our objective is to pinpoint the minimal requirements for solving the bit-dissemination problem. We focus on the two following resources:

- The memory of the agents, i.e., the amount of information about past observations (measured in bits) that they are allowed to use in order to make a decision.
- The sample size $\ell$. Intuitively, the larger $\ell$ is, the more accurately the agents can estimate $x_t$ (the proportion of 1-opinions in the population).

For convenience, we express the convergence time of protocols in both activation settings using *parallel rounds*. One parallel round corresponds to $n$ activations, which equals 1 round in the parallel setting and $n$ rounds in the sequential setting. Keep in mind that while this allows for qualitative comparisons, the two settings are not directly equivalent.

**Hardness induced by self-stabilization.** By definition of self-stabilization, protocols cannot specify the initialization of the memory of the agents, which should be thought of as being set up by an adversary. As a consequence, agents cannot rely on having a shared clock modulo-$T$ from the beginning of execution, or even distinct identifiers. Instead, a protocol must be able to create and maintain such objects from any arbitrary state in order to use them.

**Hardness induced by passive communications.** The fact that sampling happens over the opinions of other agents is often referred to as *passive communication*. This assumption is inspired by natural scenario where information can be gained only by observing the behavior of other agents, which, in principle, may not even intend to communicate [5]. Not only does it restrict each observation to a single bit in our model, but it also leaves the agents essentially unable to communicate when it comes to reaching consensus, since they are then forced to display the correct opinion $z$.

In contrast, a distributed system is said to exhibit *active communications* if the opinion $X_t^{(i)}$ of each agent $i$ differs from the 1-bit message displayed to other agents. Consensus would then be defined over the opinion $X_t^{(i)}$, while sampling would occur separately, based on the arbitrary 1-bit message. Results from [1], allowing agents to synchronize a clock in a self-stabilizing way, can be used to solve the bit-dissemination problem with active communications. Moreover, a solution to an equivalent problem (called "bit-broadcast") was identified for population protocols [7]. However, neither of these algorithms can be adapted to the framework of passive communications, which appears to be significantly weaker. These considerations are discussed in more detail in [9, Section 1.4].

## 2 Fast Dissemination with Memory

In this section, we investigate what happens when agents are allowed to use a moderate amount of memory. We describe a simple algorithm, called *Follow the Trend (FtT)*, that efficiently solves the bit-dissemination problem while being self-stabilizing.

The protocol is based on letting agents estimate the current tendency direction of the dynamics, and then adapt to the emerging trend. Informally, each non-source agent compares the number of 1-opinions that it observes in the current round, with the number observed in the previous round. If more 1's are observed now, then the agent adopts the opinion 1, and similarly, if more 0's are observed now, then it adopts the opinion 0. If the same number of 1's is observed in both rounds, then the agent does not change its opinion. Formally, our algorithm is defined as Algorithm 1 (the number of 1-opinions observed in the last round is stored in a variable named $\sigma_t$).

As illustrated on Figure 1, this behaviour creates a persistent movement of the average opinion of non-source agents towards either 0 or 1, which "bounces" back when hitting the wrong opinion.

Up to modifying it slightly (to account for technical difficulties), we can show that Algorithm 1 solves the bit-dissemination problem efficiently when all agents are activated simultaneously, as long as the sample size $\ell$ is at least poly-logarithmic in $n$.

**Theorem 1** (Theorem 1 in [9])**.** *There exists a protocol based on Algorithm 1 that solves the bit-dissemination problem in the parallel setting in* $O(\log^{5/2} n)$

**Algorithm 1:** Follow the Trend (sketch)

**Input**: Current opinion $X_t^{(i)} \in \{0, 1\}$, memory state $\sigma_t \in \{0, \dots, \ell\}$, opinion sample $S \in \{0, 1\}^\ell$

$\sigma_{t+1} \leftarrow$ number of 1-opinions in $S$ ;

**if** $\sigma_{t+1} > \sigma_t$ **then** $X_{t+1}^{(i)} \leftarrow 1$ ;

**else if** $\sigma_{t+1} < \sigma_t$ **then** $X_{t+1}^{(i)} \leftarrow 0$ ;

**else** $X_{t+1}^{(i)} \leftarrow X_t^{(i)}$ ;

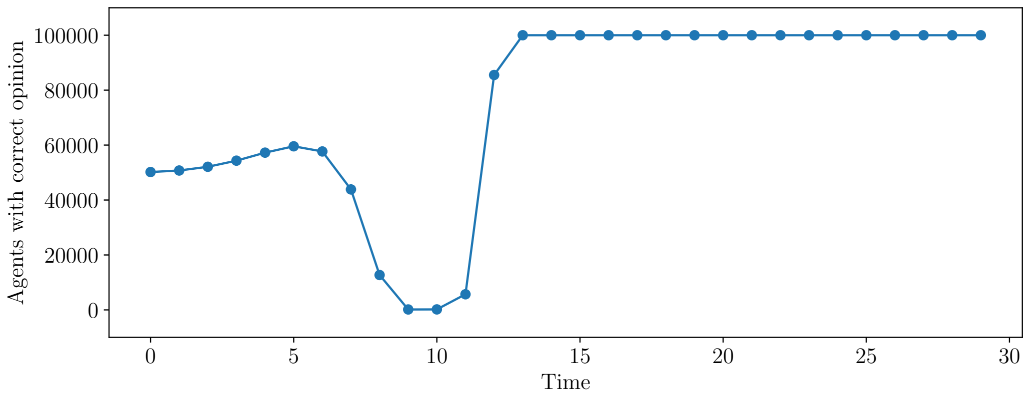


Figure 1: One execution of Algorithm 1 in the parallel setting, with $n = 10^5$ and $\ell = 34$. The graph shows the number of agents holding the correct opinion as a function of the round number.

*parallel rounds with high probability, while relying on* $\ell = \Theta(\log n)$ *samples and* $\Theta(\log \log n)$ *bits of memory.*

To prove the correctness of a self-stabilizing algorithm, there are generally 2 methods. Ideally, one is able to exhibit a scalar quantity (a "potential") that contains enough relevant information about the system, while being simpler to analyze. For example, if this quantity is simultaneously decreasing at a fast rate and bounded from below, then it directly implies an upper bound on the convergence time of the algorithm. Another, more tedious approach is to partition the configuration space into as many subsets as necessary, and then characterize the algorithm's behavior on each of the subsets. Unfortunately, our proof of Theorem 1 belongs to the latter category. When Algorithm 1 is used, each configuration is fully characterized by the proportion of 1-opinions in the previous round and in the current round, i.e., by the couple $(x_t, x_{t+1})$. The resulting 2-dimensional configuration space is depicted in Figure 2, together with the partition used in the proof of Theorem 1.

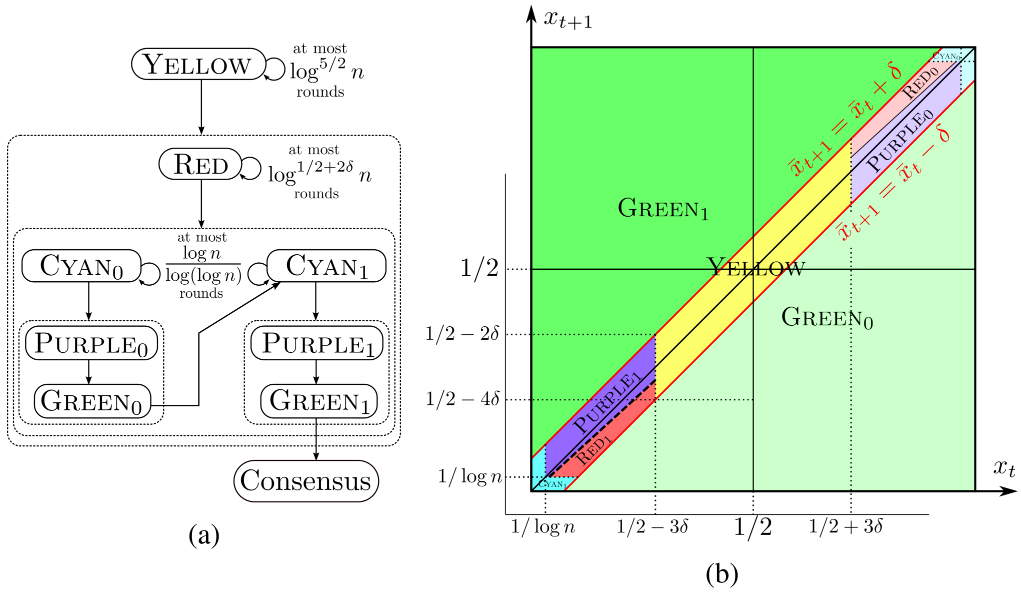


Figure 2: **(a)** Sketch of the proof of Theorem 1. Self-loops indicate the number of rounds spent by the process in corresponding areas. **(b)** An illustration of the areas (see [9, Section 2.1] for an exact definition).

Although Theorem 1 only holds in the parallel setting, it seems that the "Follow the Trend" strategy can be adapted to the sequential setting (where only one random agents is activated at a time). The trick is to let agents execute one step of Algorithm 1 every $\log n$ activations. This technique essentially limits the number of time that an agent $i$ can be executed between two activations of another agent $j$, at the cost of slowing down the process by a logarithmic factor. While the resulting protocol was never rigorously analyzed, a detailed description and some empirical evidence about its correctness can be found in [2, Section 5].

# 3 The Need for Memory in the Sequential Setting

In this section, we restrict attention to *memory-less* algorithms, in order to further characterize the requirements of the problem. More specifically, we consider update rules that only depend on the agent's current opinion $X_t^{(i)}$, and on the last sample $S$. In particular, this assumption precludes the possibility to maintain clocks and counters, or to estimate the tendency of the dynamics as in Algorithm 1.

Since agents having the same opinion are indistinguishable in absence of memory, the configuration of the system in round $t$ is fully described by the proportion $x_t$ of agents with opinion 1. Moreover, the distribution of $x_{t+1}$ depends only on $x_t$, and hence the process is always a Markov chain on $\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1\}$. In the sequential setting, the number of agents with opinion 1 may only vary by at most one unit in every round. The Markov chain is even simpler in this case, as its underlying graph is just a path of length $n + 1$. These are known as "birth-death" chains [10, Section 2.5]. The exact transition probabilities of the birth-death chain induced by a given protocol $\mathcal{P}$ does not only depend on $\mathcal{P}$, but also on the opinion of the source (which is not running the protocol). For example, the state $x_t = 0$, corresponding to a consensus on opinion 0, cannot be reached when the source has opinion 1. However, since the source itself is sampled with probability only $1/n$ by other agents, its impact on most of the transition probabilities is expected to be quite limited.

In summary, a protocol $\mathcal{P}$ solving the bit-dissemination problem in the sequential setting implies the existence of two nearly identical birth-death chains $C_1$ and $C_0$ (corresponding to the case that the correct opinion is 1 or 0 respectively). Due to their similarity, any tendency of $C_1$ to move towards state 1 (a consensus on opinion 1) implies an almost equivalent tendency of $C_0$ to move away from state 0 (a consensus on opinion 0). This observation can be leveraged to obtain the following lower-bound, which interestingly, holds regardless of the sample size.

**Theorem 2** (Theorem 3 in [2])**.** *The expected convergence time of any memory-less dynamics for the bit-dissemination problem in the sequential setting is at least* $\Omega(n)$ *parallel rounds, or equivalently* $\Omega(n^2)$ *activations, even when* $\ell \geq n$.

The same reasoning implies that the lower bound is reached when the induced birth-death chains are unbiased random walks. The corresponding algorithm turns out to be the *voter* dynamics (Algorithm 2), in which activated agents simply copy the opinion of another agent chosen uniformly at random. Indeed, an upper bound based on Algorithm 2 and almost matching Theorem 2 follows from classical arguments.

**Algorithm 2:** Voter dynamics ($\ell = 1$)

**Input**: Opinion sample $S \in \{0, 1\}$

$X_{t+1}^{(i)} \leftarrow S$ ;

**Theorem 3** (Theorem 4 in [2])**.** *The voter dynamics solves the bit-dissemination problem in the sequential setting in $O(n \log n)$ parallel rounds in expectation.*

Since the voter dynamics only uses $\ell = 1$, we conclude that the sample size is of no importance in the sequential setting. On the other hand, together with the insights from Section 2, our results imply that memory is a critical requirement.

# 4 The Minority Dynamics

In contrast to the sequential setting, memory-less dynamics in the parallel setting may jump from any configuration to any other in just one round – albeit with extremely small probability. Therefore, the ideas behind Theorem 2 are not applicable, which suggests that the lower bound could be beaten. In this section, we show that fast convergence is indeed possible, by considering the *minority* dynamics (Algorithm 3). Introduced by Amos Korman, this fascinating protocol has its own interest beyond the bit-dissemination problem.

The minority rule is defined as follows: if an agent sees unanimity among the $\ell$ elements of the sample $S$, it adopts this unanimous opinion; otherwise, it adopts the opinion corresponding to fewer samples.

**Algorithm 3:** Minority dynamics

**Input**: Opinion sample $S \in \{0, 1\}^{\ell}$

**if** *all opinions in $S$ are equal to $x$* **then**
  $X_{t+1}^{(i)} \leftarrow x$ ;
**else**
  $X_{t+1}^{(i)} \leftarrow$ minority opinion in $S$ *(breaking ties randomly)* ;

When all non-source agents follow Algorithm 3, the proportion of agents with opinion 1 exhibits chaotic oscillations with no apparent pattern, as illustrated on

Figure 3. As long as the sample size is large enough, the oscillations eventually stop abruptly and the group reaches consensus in just a few rounds. However, when $\ell$ is too small compared to $n$, simulations depict seemingly endless oscillations. Identifying the value of $\ell$ required for the rapid convergence of the dynamics is a challenging task, as the chaotic nature of the process complicates the analysis. To this day, the only existing upper bound on the convergence time relies on a relatively large sample size ($\ell \geq \sqrt{n}$).

**Theorem 4** (Theorem 1.3 in [3]). *If $\ell = \Omega(\sqrt{n \log n})$, then the minority dynamics solves the bit-dissemination problem in the parallel setting in $O(\log n)$ parallel rounds in expectation.*

Under the minority dynamics, the configuration of the system in round $t$ is fully characterized by the number $m_t$ of agents holding the minority opinion, and hence the configuration space is simply $\{0, \ldots, \lfloor n/2 \rfloor\}$. As for Theorem 1, the proof of Theorem 4 proceeds by partitioning this one-dimensional space into several areas, in which the dynamics' behavior is easier to predict. It then consists in bounding the probability that the process remains stuck in the same area for a long time, and in showing that it eventually reaches the "green" area leading to consensus with constant probability (see Figure 4 for an illustration).

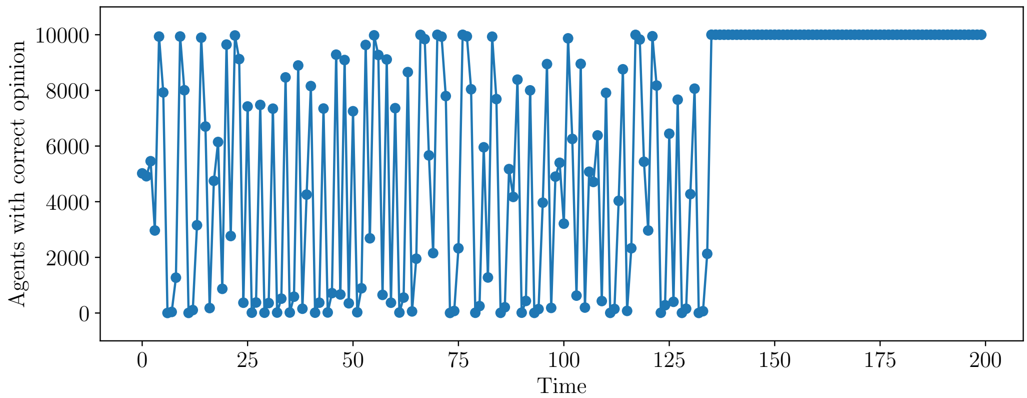


Figure 3: One execution of the minority dynamics in the parallel setting, with $n = 10^4$ and $\ell = 36$. The graph shows the number of agents holding the correct opinion as a function of the round number.

# 5 Open Questions

We conclude by listing the most interesting open problems arising from this line of works. We leave aside natural generalizations of the results described above,

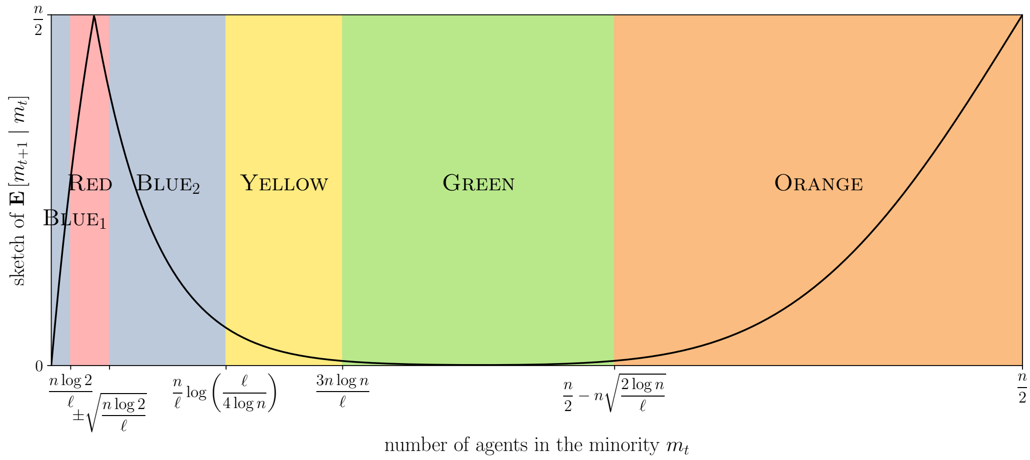


Figure 4: Partition of the configuration space into 6 areas used in the proof of Theorem 4. The black line indicates how the random variable $m_t$, corresponding to the number of agents with the minority opinion, is expected to vary in the next round.

such as the case where agents have access to more than 2 opinions, or the case where the communication network is not fully connected.

**Q1)** *What is the smallest sample size $\ell^\star$ allowing the minority dynamics to converge fast in the parallel setting?*

This question is interesting even in the absence of a source agent. It is possible to show that the convergence time of the minority dynamics is $\Omega(e^n)$ when $\ell = O(1)$, which together with Theorem 4, implies that $\ell^\star$ must satisfy $1 \ll \ell^\star \leq \sqrt{n \log n}$ – leaving a huge gap for future works.

**Q2)** *Is there a memory-less algorithm solving the bit-dissemination with fewer samples than the minority dynamics in the parallel setting?*

As a first step towards answering this question, a general lower bound is given in [6]. More specifically, it is shown that when $\ell = O(1)$, any memory-less dynamics needs almost linear time to solve the bit-dissemination problem, that is, at least $\Omega(n^{1-\varepsilon})$ rounds for every $\varepsilon > 0$. However, the arguments therein are not applicable even when $\ell$ is only $\Omega(\log n)$.

**Q3)** *Is there an algorithm solving the bit-dissemination problem when $\ell = O(1)$?*

Because of the aforementioned lower bound in [6], such algorithm would necessarily rely on memory. A promising candidate, called “BSF”, is mentioned in [12, Chapter 10], but was never successfully analyzed.

## Acknowledgments

The results presented in this column were obtained as part of the doctoral thesis "Algorithmic Perspectives to Collective Natural Phenomena", conducted under the supervision of Amos Korman and Pierre Fraigniaud.

## References

[1] Paul Bastide, George Giakkoupis, and Hayk Saribekyan. Self-stabilizing clock synchronization with 1-bit messages. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 2154–2173. Society for Industrial and Applied Mathematics, January 2021.

[2] Luca Becchetti, Andrea Clementi, Amos Korman, Francesco Pasquale, Luca Trevisan, and Robin Vacus. On the role of memory in robust opinion dynamics. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, {IJCAI-23}*, volume 1, pages 29–37, Macao, August 2023. International Joint Conferences on Artificial Intelligence Organization.

[3] Luca Becchetti, Andrea Clementi, Francesco Pasquale, Luca Trevisan, Robin Vacus, and Isabella Ziccardi. The minority dynamics and the power of synchronicity. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 4155–4176. Society for Industrial and Applied Mathematics, January 2024.

[4] Lucas Boczkowski, Amos Korman, and Emanuele Natale. Minimizing message size in stochastic communication patterns: Fast self-stabilizing protocols with 3 bits. In *Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 2540–2559. Society for Industrial and Applied Mathematics, January 2017.

[5] Étienne Danchin, Luc-Alain Giraldeau, Thomas J. Valone, and Richard H. Wagner. Public information: From nosy neighbors to cultural evolution. *Science*, 305(5683):487–491, July 2004.

[6] Niccolò D'Archivio and Robin Vacus. On the limits of information spread by memory-less agents. In *38th International Symposium on Distributed Computing (DISC 2024)*, Madrid, October 2024. Schloss-Dagstuhl - Leibniz Zentrum für Informatik.

[7] Bartłomiej Dudek and Adrian Kosowski. Universal protocols for information dissemination using emergent signals. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 87–99, New York, NY, USA, June 2018. Association for Computing Machinery.

[8] Aviram Gelblum, Itai Pinkoviezky, Ehud Fonio, Abhijit Ghosh, Nir Gov, and Ofer Feinerman. Ant groups optimally amplify the effect of transiently informed individuals. *Nature Communications*, 6(1):7729, July 2015.

[9] Amos Korman and Robin Vacus. Early adapting to trends: Self-stabilizing information spread using passive communication. *Distributed Computing*, February 2024.

[10] David A Levin and Yuval Peres. *Markov Chains and Mixing Times, Second Edition*, volume 107. American Mathematical Soc., 2017.

[11] H. F. McCreery and M. D. Breed. Cooperative transport in ants: A review of proximate mechanisms. *Insectes Sociaux*, 61(2):99–110, May 2014.

[12] Emanuele Natale. *On the Computational Power of Simple Dynamics*. PhD thesis, 2016.