Consensus Dynamics: an Overview

Luca Becchetti[1], Andrea Clementi[2], and Emanuele Natale[3]

[1]*Sapienza* Università di Roma, becchetti@diag.uniroma1.it
[2]Università *Tor Vergata* di Roma, clementi@mat.uniroma2.it
[3]Université Côte d'Azur, CNRS, I3S, INRIA, natale@unice.fr

Contents

1 Introduction **2**
1.1 Algorithmic perspective on emergent complexity . . . 2

2 Preliminaries **4**
2.1 Distributed models . . . 4
2.2 Dynamics . . . 5
2.3 Stable almost Consensus . . . 7

3 Voter dynamics **8**
3.1 Fundamental results on the Voter dynamics . . . 9

4 Median dynamics **11**

5 Majority dynamics **14**
5.1 2-Choices dynamics . . . 14
5.2 The 3-Majority dynamics . . . 18
5.3 2-Choices vs 3-Majority dynamics . . . 20
5.4 Further results on majority dynamics . . . 21

6 Dynamics using extra states **21**
6.1 Two colors in the random sequential model . . . 22
6.2 Memory lower bounds for Majority and k-Plurality Consensus . . . 25
6.3 The parallel synchronous setting . . . 26
6.4 Dealing with more colors . . . 28

7 Averaging dynamics **29**
7.1 Averaging dynamics . . . 30
7.2 Averaging dynamics, random walks and Consensus . . . 30
7.3 Clustering properties of the Averaging dynamics . . . 33
7.3.1 Clustering properties in the $\mathcal{LOCAL}$ model . . . 34
7.3.2 Clustering properties in the random sequential model. . . . 36

1 Introduction

The term *Distributed System* typically refers to a set of entities, called *nodes*, connected by point-to-point communication links. The set of nodes together with the set of links form a network, which is usually represented by a graph. The term "system" is used to remark that nodes *evolve* over time, i.e., they change their internal states according to some local interaction-rule, which is applied in every time step (i.e. *round*).

Nowadays, many complex phenomena are studied using models that are *de facto* (dynamical) distributed systems. Examples range from physics and biology, where the study of systems of simple interacting entities (such as particles or bacteria) is an active research area, to modern social sciences, with their focus on social networks of intelligent agents.

In physics, for example, the investigation of interacting particle systems has played a central role in establishing statistical mechanics as a fundamental theory connecting the microscopic behavior of elementary entities (e.g., molecules) to emerging, macroscopic phenomena [Lig12].

In systems biology, many natural processes exhibit complex behavior and can perform non-trivial information-processing tasks, in many respects behaving as highly robust and adaptive distributed systems of simple agents [Cha12, BCE+17].

Decades ago, inherently distributed models were introduced in social sciences to describe emerging social phenomena, such as consensus and opinion formation [DeG74, FJ90]. More recently, variants of these and other models have been revisited in an algorithmic perspective, with the general intent of investigating emerging computational properties of social networks of elementary computing entities [MS10, KMM+13, MNT14]. These algorithmic models provide powerful metaphors, capable of capturing important traits of emerging, complex social behaviour. In a number of cases, they afforded a rigorous analysis of the collective ability of social networks of simple agents to perform non-trivial coordination tasks.

1.1 Algorithmic perspective on emergent complexity

It is common opinion [FK13, NBJ15] that *Theoretical Computer Science* (TCS) is in a vantage point to achieve significant advances in our understanding of key emergent properties in complex systems. Indeed, interpreting "computational entity" and "communication links" in a broad sense, the use of the *computational lens* [Kar11] to investigate the myriad of algorithmic processes evolved by such dynamical systems, currently represents one of the most significant challenges in many research areas. In particular, a very interesting and fascinating issue is the apparent difficulty to provide non-trivial mathematical characterizations of the *Complexity From Simplicity* (for short *CFS*) phenomenon, namely, the hidden interplay between simple local interactions occurring at a microscopic level and global system evolution, often characterized by complex and sometimes surprising forms of self-organizing behavior.

Distributed Computing (DC) naturally lends itself to addressing such questions, with its focus on the design and analysis of systems of computational agents that collectively achieve some global goal in an efficient and resilient way. If we set an emerging, observed "complex behaviour" as the global goal of the system, while constraining agents to "simple" communication and computational primitives that are consistent with the microscopic behaviour of a social or natural system of interest, we are in fact investigating a form of the *Complexity From Simplicity* phenomenon. The above description of the current state of affairs is not wishful thinking. From programmable matter [DDG+14, CDRR16] to chemical reaction networks [CSWB09, Dot14, CKW16, Reu16], from sensor networks [AAD+06a, AFJ06] to social insects' behaviour [FHK14, FN16], significant research efforts in distributed computing are driven by the pursuit of a theory analogous to the one developed in statistical mechanics to characterize the behavior of interacting particle systems, with the latter replaced by (simple) computing agents.

One possible, algorithmic abstraction of the interplay between the microscopic and the macroscopic scales of complex phenomena is the notion of *Dynamics*. In Distributed Computing, the term *Dynamics* usually refers to simple interaction and computational rules, *locally* implemented and applied by the nodes of an anonymous network. Currently, the study of dynamics, their global properties and their application to perform fully decentralized computational tasks is an active research area. In a sense, an entire sub-area of Distributed Computing is devoted to the investigation of problems that are inherently related to the CFS phenomenon [AAE08, BCN+16, DGM+11, FHK14, MNT14]. Originally, the study of dynamics in Distributed Computing was motivated by the quest for novel algorithmic paradigms in the design of *lightweight* decentralized algorithms, with the goal of achieving levels of simplicity comparable to those of interacting particle systems. As an early example, [HP01] analyzed the famous Voter model [Lig12] (here, Voter dynamics), as a proportionate-consensus dynamics for distributed systems.

As in other areas, computer simulations alone are unlikely to afford scientific breakthroughs in the absence of new, robust theories. Distributed computational models can help understand biological and social distributed systems. However, a long observed fact is that very few such processes seem amenable to analytical treatment [Wol02]. This naturally raises the question of whether it is possible to make significant advances towards a rigorous theory of dynamics.

The main goal of this survey is to provide an overview of a rich body of recent results, which globally provide analytical evidence of important computational and self-organizing properties of dynamics. In this survey, we mostly focus on results that address crucial aspects of these distributed processes and in particular:

- Convergence time. If allowed to evolve from an initial configuration, dynamics may or not converge to a *stationary* state (also known as steady states and absorbing configurations [LPW09]). When this is the case, a key question is the time it takes to achieve convergence. From a mathematical perspective, dynamics are typically Markov chains [LPW09]. Thus, in principle, one might leverage advanced techniques in Markov-chain theory and concentration-of-measure tools, to provide tight bounds on the number of rounds required by the system to reach a steady state in the absence of external perturbations. Convergence time is a key measure to quantitatively describe the behavior of important epidemic processes in biological systems and social networks [AAE08, Ald13, FHK15], as well as to characterize the efficiency of algorithms that solve fundamental tasks in parallel and distributed computing [BCN+16, BGPS06, DGH+87, DGM+11].

- Computational power. Another crucial and fascinating characterization of dynamics concerns their ability to somehow "compute" and reflect global properties of a starting configuration. This is extremely important, since initial configurations in many cases can be regarded as a global input, collectively "sensed" from the environment by the nodes of the system. Typical examples include the ability to efficiently converge to the global median or to the average of the initial values held by nodes of the system [BGPS06, DGM+11]. Further basic tasks (listed here in increasing order of complexity) include reaching a *valid* consensus [AAE08, BCN+16], a proportional (or fair) consensus [HP01] or, finally, plurality consensus (see following sections for formal definitions of these notions). On one hand, an algorithmic characterization of the properties above can be crucial to explain surprising forms of synchronization and coordination in agent populations. On the other hand, decentralized mechanisms performing the aforementioned tasks are fundamental building-blocks to address key problems, well beyond core primitives traditionally considered in distributed computing [AW04].

- Fault-tolerance and self-stabilization. Other interesting properties of dynamics reflect the ability of many natural systems to achieve stronger or weaker forms of *self-stabilization* [AFJ06, DGM+11, Dij74] in the presence of faulty or, even worse, malicious behavior [DH07]. Informally, a dynamic process is said to be *self-stabilizing* if, starting from any possible configuration, the system eventually reaches a *legal* configuration, i.e., one exhibiting certain properties of interest. This property should be preserved in the presence of a limited fraction of faulty/malicious nodes that depart from the dynamics' local rule. Self-stabilization is a fundamental desideratum in Distributed Computing [Dij74, Dol00a]. While self-stabilizing protocols do exist for specific problems, several impossibility results have been obtained over the last decades [AAFJ08, BBK11, Dol00a, DKS10]. In this survey, we discuss a number of examples, highlighting the somewhat surprising ability of dynamics to achieve "relaxed", yet effective forms of self-stabilization in important cases.

In the remainder, we consider the behaviour of dynamics under different models of distributed communication, ranging from asynchronous population protocols [AAFJ08] to synchronous models, such as the $\mathcal{LOCAL}$ model [CKP16, DGM+11, CIG+15].

Disclaimer. This overview is definitely not complete, nor is it "fair": Rather, it covers recent contributions that mainly appeared in traditional venues of the Theoretical Computer Science and Distributed Computing communities. One common trait of the results we discuss here is the fascination they exerted on the authors of this survey, inducing them to focus on the investigation of these distributed models over the past five years.

2 Preliminaries

As discussed in the previous section, the past few years have witnessed a surge in the design and analysis of elementary, yet fundamental synchronization and coordination primitives in distributed systems, under models that severely constrain communication and computation [AAE08, BCN+15, DGM+11]. This interest is motivated both from efficiency considerations and because such models seem to capture key aspects of the way coordination and consensus are achieved in social networks, biological systems, and other domains of interest in network science [AAD+06a, AFJ06, DKS10, CKW16, Dot14, FHK14, FPM+02].

The next subsection provides an overview of the computation and communication models we consider in the remainder of this survey. Subsection 2.2 introduces the notion of dynamics. Finally, Subsection 2.3 describes the probabilistic notion of almost-stable consensus, which plays a crucial role in essentially all results selected in this survey.

2.1 Distributed models

In the remainder of this survey, we use the terms *agent* and *node* interchangeably.

We assume an anonymous network, whose nodes/agents possess no unique IDs, nor do they have any static binding of their local link ports (i.e., nodes cannot keep track of *who sent what*). Computationally, unless otherwise specified, we assume the most restrictive setting in which each node only has $\mathcal{O}(\log|\Sigma|)$ bits of memory available, where Σ is an alphabet that describes the set of possible, fundamental states for a node in the network.[1] We further assume that this bound extends to link bandwidth available in each round. In the most restrictive case, we further assume that communication resources are severely constrained and non-deterministic:

[1] The meaning and size of Σ depend on the problem under consideration. For example, in consensus problems, Σ will denote the set of available opinions or *colors*.

Every node can communicate with at most a (small) constant number of random neighbors in each round.

These constraints are well-captured by the *uniform-gossip* communication model [DGH+87, KSSV00, KDG03]: In each round, every node can exchange a (short) message (say, $\Theta(\log(|\Sigma|))$ bits) with each of at most h random neighbors, with h a (small) absolute constant[2]. A more recent, sequential variant of the uniform-gossip model is the *(random) Population Protocol* Model [AAE08, AG15, AR07] where, in each round, a single interaction between a pair of randomly selected nodes occurs[3]. In the *asynchronous*, sequential model, one oriented edge (u, v) of the underlying graph is selected uniformly at random in each step. Upon selection, node u can "pull" information from v and perform local computations[4]. This model is inspired by network scenarios where a (random) link activation represents an opportunistic meeting that the endpoints can exploit to interact in a single time step. Observe that this process is sequential, since only one node can update its state in each round. Moreover, it is asynchronous, since nodes do not share any global clock. Finally, the system is anonymous: nodes are not aware of theirs or their neighbors' identities. This model is also known as the *(uniform) Population Protocol* Model [DEM+18].

In the parallel, synchronous model, called (uniform) $\mathcal{GOSSIP}$ nodes share a discrete-time global clock. In each round, one edge is selected for every node, independently and uniformly at random. Then, every node exchanges one message with its selected partner. Finally, each node performs a local computation, possibly resulting in an update of its internal state. Communication can occur in two ways: a node can pull information from its randomly-selected partner, or it can push information to it. We mostly consider the first variant, also known as the $\mathcal{PULL}$ model.

Finally, we also consider the $\mathcal{LOCAL}$ model [Pel00, FKP13], which is intended to capture the essential traits of locality in distributed computing. In this model, computation proceeds in fault-free synchronous rounds, during which every agent can exchange messages with each of its neighbors and perform a computational step. The averaging dynamics described in Section 7 in fact works in the $\mathcal{LOCAL}$ model. Though no assumption on the size of messages or computational power of a single node are implied by the model, the examples we consider in this survey are consistent with constraints on both.

2.2 Dynamics

Within the field of Distributed Computing, the focus of this work is on a class of distributed processes that may resemble interacting particles systems in statistical mechanics [LM10, Lig12]: the class of protocols we consider are *simple* and *lightweight* [HP01], their typical behavior strongly relies on randomness, which constitutes an essential part of the process. Commonly, they are referred to as *dynamics* [AAE08, AAB+11, Dot14, MNT14].

As in the case of *natural algorithms* and *complex networks*, the notion of dynamics is affected by a clear discrepancy between the informal consensus about the meaning of the concept within

[2]In fact, $h = 1$ in the standard uniform-gossip model. In general, it is easy to see that all results we are going to present in this survey still hold in this more restricted setting, at the cost of a constant slow-down in convergence time and local memory size.

[3]We remark that, in the original definition of Population Protocols [AAD+06b], the sequence of edge-activations is determined by a *scheduler* which does not need to be random; however, in this survey we focus on the uniformly-random scheduler, which has been the standard assumption in the literature concerned with the convergence time of the protocols.

[4]In an essentially equivalent continuous-time model, each (oriented) edge has a clock that ticks at random intervals with a Poisson distribution of average 1; when the clock ticks, the endpoints of the edge become active. For t larger than $n \log n$, the behavior of the continuous time process for t/n units of time and the behavior of the discrete-time process for t steps are roughly equivalent.

the related experts' community, and the lack of principled attempts at a rigorous definition, which would be extremely useful, especially to outsiders.

For this reason, the first definition appearing in this survey is an attempt at a first formalization[5] of the notion of *dynamics* as *simple, lightweight, natural, local, elementary rules*.

Definition 1 (Dynamics)**.** *A* dynamics *is a distributed algorithm characterized by a very simple structure. In particular, the state of a node at round t only depends on its state and a symmetric function of the multi-set of states of its neighbors at round $t-1$, while the update rule is the same for every graph and for every node and it does not change over time.*

Remark 1. *Within the constraints of the previous definition, it may still be possible to come up with computational rules that appear* cumbersome and unnatural*. We emphasize that the goal of Definition 1 is to provide a reference, not to replace reliance of the scientific community on the real world phenomena the concept is intended to capture. Definition 1 is therefore overtly provisional and open to replacement by more suitable candidates.*

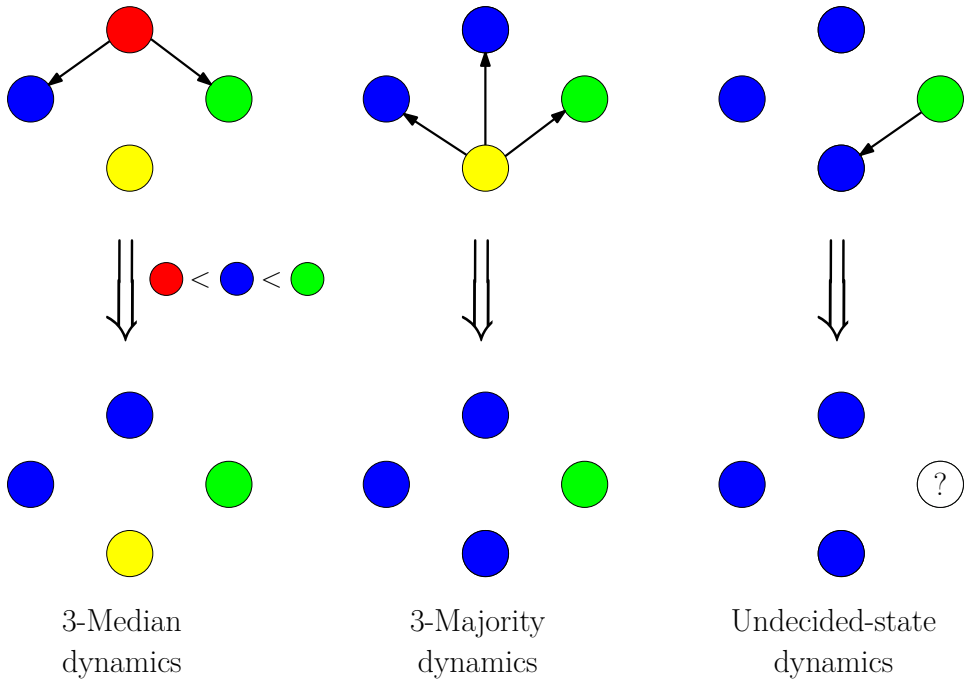

Figure 1: Illustration of the 3-Median dynamics (in which each agent samples two other agents at random and updates her color with the median of their values and her own), the 3-Majority (in which each agent samples three other agents at random and update her color with the most frequent value among those three, breaking ties arbitrarily), and Undecided-State Dynamics (in which each agent samples another agent, if their values differ she becomes *undecided*, and if she is undecided she picks the first color she sees).

Note that Definition 1 implies that the network is *anonymous*, that is, nodes do not possess distinguished identities. Examples of dynamics include update rules in which every node updates its state to the plurality or the median of the states of its neighbors[6] (see Figure 1), or which it updates to the average of the values held by its neighbors[7] (see Figure 1).

[5] The definition has already appeared in [BCN+17b].

[6] When states correspond to rational values.

[7] Actually, the averaging rule over rational numbers requires arbitrarily-high numerical precision, which often is

In contrast, an algorithm that, say, proceeds in two phases, using averaging during the first $10 \log n$ rounds and plurality from round $1 + 10 \log n$ onward, with n the number of nodes, is not a dynamics according to our definition, since its update rule depends on the size of the graph. As another example, an algorithm that starts by having the lexicographically first node elected as "leader" and then propagates her state to all other nodes again does not meet the definition of dynamics, since it assigns roles to the nodes and requires them to possess distinguishable identities.

2.3 Stable almost Consensus

Consensus and Plurality Consensus are simplified models of the way inconsistencies and disagreements are resolved in social networks, biological systems and peer-to-peer systems [Dot14, FPM+02, MNT14]. In distributed models that severely restrict the way in which nodes communicate (to model constraints that arise in peer-to-peer systems or in social or biological networks), upper and lower bounds for Consensus protocols often give insights on how to break symmetry in distributed networks in the case of balanced initial color configurations.

It is reasonable to claim that the main, initial interest of our scientific community to all dynamics described in this survey was essentially motivated by their ability to achieve some form of consensus. *Consensus* (a.k.a. *Agreement*) is a fundamental algorithmic problem that has been studied under all available models of distributed computing. Several variants of this task have been defined, depending on the underlying communication model, the presence of faults and/or Byzantine agents, the specific properties required for the final legal system configuration, and further important aspects. Its original version [Dij74, Dol00b, PSL80, Rab83] can be informally defined as follows. A collection of agents, each holding a piece of information (an element of a set Σ), interact with the goal of agreeing on one of the elements of Σ initially held by at least one agent, possibly in the presence of an adversary that is trying to disrupt the protocol.

As for the Consensus task in the presence of an adversary (a.k.a. Byzantine Agreement) [PSL80, Rab83], the goal is to design a distributed, local protocol that brings the system into a configuration that meets the following criteria:

- (Agreement). All non-corrupted nodes support the same color v;
- (Validity). The color v must be a *valid* one, i.e., a color which was initially declared by at least one (non-corrupted) node;
- (Termination). Every non-corrupted node can correctly decide to (locally) terminate the protocol at some point.

The classic notion of Consensus is typically too strong and unrealistic in distributed settings where dynamics are used as algorithmic models of natural or social phenomena. Self-organization and other important properties shown by these systems rely on *weaker* forms of consensus, which have been deeply investigated [AAE08, AFJ06, BCN+17b, DGM+11]. We thus consider a variant of the *Stable Consensus* problem [AFJ06] considered in [AAE08]: There, a solution is required to converge to a stable *regime* in which the above three properties are guaranteed in a relaxed, yet useful form[8]. We formalize these notions in the paragraphs that follow.

not a reasonable assumption in several applications where agents have limited local memory. Still, we believe this protocol should be considered a dynamics in the sense defined above, since it matches all properties required in Definition 1 and well-describes microscopic behavior of many natural and social systems [HK05, OT09, XBK07].

[8] These relaxed convergence properties are described in detail in Section 7 of [AAE08].

We consider a distributed system of n nodes/agents, interacting over an underlying graph $G = (V, E)$ according to one of the communication models described in Subsection 2.1. In each round, every node supports a color, i.e. an element $i \in \Sigma = \{1, 2, \ldots, k\}$ (with $2 \leqslant k \leqslant n$). A configuration of the system is thus a function $c : V \to \Sigma$. When G is the complete graph and nodes are anonymous, the state of the system at any round is completely specified by a *color configuration* $\mathbf{c} = \langle c_1, \ldots, c_k \rangle$ (where c_i denotes the number of agents supporting color $i \in [k]$). There can be an initial *plurality* c_1 of agents supporting some *plurality color* (wlog, we can assume that color communities are ordered, so that $c_i \geqslant c_{i+1}$ for any $i \leqslant k-1$). Initially, every node only knows its own color. In the remainder, the subset of agents supporting color i is denoted as C_i.

Definition 2 (Stable almost Consensus)**.** *A* stabilizing almost (Plurality) Consensus *protocol must ensure the following properties:*

- *(Almost agreement.) Starting from any initial configuration, in a finite number of rounds, the system must reach a* regime *of configurations where all but a* negligible *"bad" subset (i.e. having size $\mathcal{O}(n^\gamma)$ for constant $\gamma < 1$) of the nodes support the same color.*
- *(Almost validity / Almost plurality). The system is required to converge w.h.p. to an almost-agreement regime where all but a negligible bad set of nodes keep the same* valid *color. In the case of* stabilizing Almost Plurality Consensus*, the almost-validity property is replaced by the request to converge to an almost-agreement regime where all but a negligible bad set of nodes hold the color that was initially supported by the plurality of the nodes, assuming that there was some initial* bias s *w.r.t. all the other colors - see below for a definition of bias.*
- *(Stability). The convergence toward such a weaker form of agreement is only guaranteed to hold* with high probability *(in short,* w.h.p.[9]*) and only over a* long period *(i.e. for any arbitrarily-large polynomial number of rounds).*

Note that the above definition does not require the property of termination. In particular, in dynamic distributed systems, nodes represent simple and anonymous computing units that are not necessarily able to detect any global property, let alone termination.

The crucial parameters in the analysis of (Plurality) Consensus processes are the number n of nodes, the number k of colors, and, in the case of Plurality Consensus, the initial *bias* towards the plurality color. The latter is characterized in terms of a parameter that only depends on the relative magnitude: Typically, this relative magnitude is defined in terms of the absolute difference or the ratio between c_1 and c_2. Moreover, several analyses of Consensus processes we describe in the following sections assume the presence of an *F-bounded dynamic adversary* that, in each round, can arbitrarily change the color of any subset of nodes of size at most F. In this framework, we say that a protocol is *F-resilient* if it guarantees the properties of Definition 2 even in the presence of F-bounded dynamic adversaries.

3 Voter dynamics

The Voter Model, which for consistency we call here VOTER dynamics, is arguably the simplest *nontrivial* dynamics: at each round, every node copies tha state of a neighbor sampled uniformly at random. Here, by nontrivial we mean that the node is updating its state in a meaningful way with respect to the states of its neighbors.

[9] According to the standard definition, we say that a sequence of events $\mathcal{E}_n$, $n = 1, 2, \ldots$ holds *with high probability* if $\mathbf{P}(\mathcal{E}_n) = 1 - \mathcal{O}(1/n^\lambda)$ for some positive constant $\lambda > 0$.

More formally, let $G(V, E)$ be a connected graph with node set V (of cardinality n) and edge set E (of cardinality m). Let Σ be a palette of colors of size n. Let $d(u)$ be the number of neighbours of $u \in V$.

Definition 3 (VOTER dynamics)**.** *The* Voter Model $\{X_t\}_t$ *on* G *is the discrete time Markov process where the state space is the set of color functions* $c : V \mapsto \Sigma$ *and its time evolution is determined by the following elementary dynamics (denoted as* VOTER *dynamics):*

- *Each node initially supports a color in* Σ*;*
- *In each round one node is selected uniformly at random and it adopts the color of a neighbour sampled uniformly at random.*

The *asynchronous* version of the VOTER dynamics, in which at each round only one node chosen uniformly at random updates its state, has long been studied in statistical mechanics as a model for consensus formation in population of spinless individuals [KRBN10, Lig12, Ald13].

The study of the synchronous version of the model, in which all nodes update their state, has been first considered in Distributed Computing in [HP01] as a proportionate agreement protocol. The *Proportionate Agreement* problem is defined as follows. Given a network of n nodes, where each node is initially supporting a color out of a set of k possibilities, we require the system to eventually reach a configuration in which every node is supporting the same color and, moreover, the final color should equal i with a probability proportional to the volume[10] of the initial set of nodes supporting i, for each $i \in [k]$.

As usual, when the system reaches a configuration in which every node is supporting the same color, we say that the system has reached *Consensus*.

The Proportionate Agreement problem is a special case of Consensus, which the further constraint of a probability distribution on the color the system eventually agrees on.

As discussed in Section 1.1, many fundamental aspects of dynamics are often best described in the language of Markov chains. In the following sections, we present the fundamental results on the convergence probability and the convergence time of the VOTER dynamics.

3.1 Fundamental results on the Voter dynamics

It is immediate to see that *absorbing* states[11] of the VOTER dynamics are the ones in which nodes have reached consensus, i.e., those corresponding to *monochromatic configurations*. Probably the first and most natural question one may ask is with which probability each monochromatic configuration is reached, as a function of the initial color configuration.

[HP01] answered this question by providing, via an elegant martingale argument, the guarantee for employing the VOTER dynamics as a protocol for the *Proportionate Agreement* problem, in which we want the system to reach consensus on color i with probability proportional to the volume of the set of nodes initially supporting color i.

[10] The *volume* of a set of nodes S is defined as the sum of the degrees of nodes in S, in formulas $\mathrm{vol}(S) := \sum_{u \in S} d_u$. In the general case of weighted graphs, we further have that d_u equals the sum of the weights of the edges incidents to u, namely $d_u := \sum_{v \sim u} w(u, v)$.

[11] We remark that the term *state* is used in quite different ways in theory of Markov chains and in the distributed computing literature: in distributed computing the term *state* usually refers to a single node and a full description of the system at a given time is referred to as *configuration* (thus, a configuration is often the vector of each node's state); on the contrary, the term *state* in the theory of Markov chains corresponds to that of *configuration* in distributed computing. In this survey, for the sake of consistency, we employ the term *state* in the Markov-chain sense when referring to *absorbing states*.

Theorem 4 (Proportionate Agreement [HP01]). *Let i be any color and $e_i^{(0)}$ be the volume of the nodes that support color i at $t = 0$; formally, $e_i^{(0)} = \{\sum_{u:c^{(0)}(u)=i} d_u\}$. Then, $\frac{e_i^{(0)}}{2m}$ is the probability that the* VOTER *dynamics converges to the monochromatic configuration in which all nodes supports color i.*

The next natural question that arises when considering a dynamics such as the VOTER dynamics, is the speed with which consensus is reached. It turns out that also for the latter question, VOTER dynamics is amenable to an elegant mathematical treatment: its convergence time can be analyzed via a *duality* with the *coalescing random walks process* [AF95].

Definition 5. *The* Coalescing Random Walks *process (for short CRW) on G is a discrete time Markov process defined as follows:*

- *Each node initially holds one token.*
- *In each round, each node sends all its tokens to a neighbor chosen uniformly at random.*

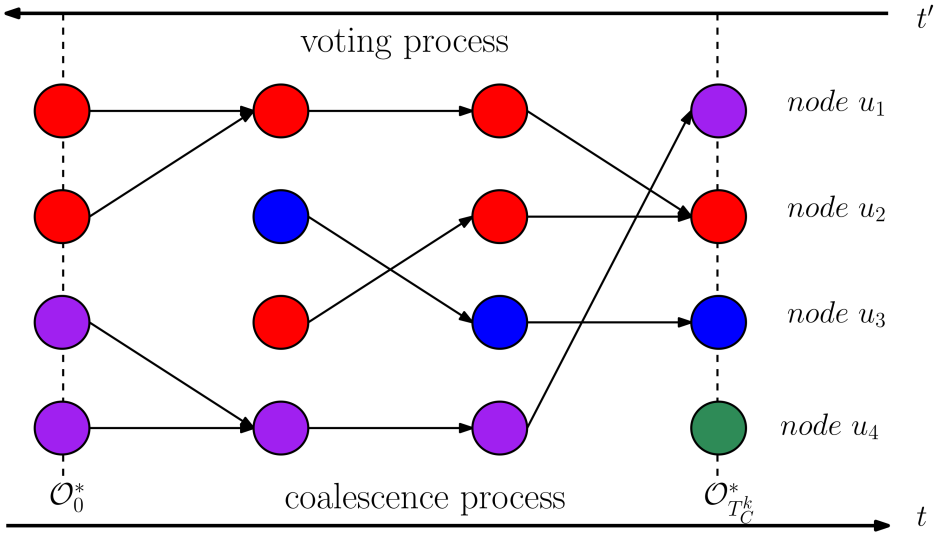

Figure 2: The diagram represent an instance of the CRW process, from left to right, in which an edge from u to v means that the token on u -if any- moves to v. In the diagram, after $T = 4$ rounds the number of random walks in the CWR process reduces to $k = 2$. The same diagram represents an instance of the VOTER dynamics, from right to left, in which an edge from u to v means that u pulls v's color. For the VOTER dynamics as well, we see that the number of colors after $T = 4$ rounds is also 2. This is no coincidence, given that the two processes share the same random choices (i.e. black arrows).

The term *coalescing* in the name of the CRW process refers to the fact that, as soon as too tokens happen to be located on the same node, they will move together in the following rounds. Hence, we can imagine that they have "coalesced".

It is easy to see that, by the process' definition, all tokens in the CRW process are eventually located on a single node. The first occurrence of the latter event is thus referred to as the *coalescing time*. In fact, the following theorem draws a direct correspondence between the convergence time of the VOTER dynamics and the coalescing time of the CRW process, allowing to study either of them by investigating the other (for a proof see e.g. Lemma 4 in [BCE+17]). An intuitive representation of the duality is also depicted in Figure 3.1.

Theorem 6 (VOTER dynamics vs CWR)**.** *Let τ be the first round such that the* VOTER *dynamics reaches consensus and let τ' be the first round such that all the tokens of the CWR process are on the same node. A coupling between the* VOTER *dynamics and the CWR process exists such that $\tau = \tau'$.*

The coalescing time of the CRW process, and thus the corresponding consensus time for the VOTER dynamics, can be studied on a variety of topologies by employing standard tools for the analysis of stochastic processes [HP01, KMTS19].

As an example, by combining Theorem 6 with the application of drift analysis techniques to the CWR process, we obtain the following theorem that provides the time after which at most k colors are still "alive" (for a proof, see e.g. Lemma 3 in [BCE+17]).

Theorem 7 (Color evolution on the complete graph)**.** *Let G be the complete graph, where each node has also a self-loop. Starting from an arbitrary initial configuration $\mathbf{c}$ on G, the* VOTER *dynamics reaches a configuration c' having at most k remaining colors w.h.p. in $\mathcal{O}(\frac{n}{k}\log n)$ rounds.*

More generally, by combining Theorem 6 with results on the *meeting time* of random walks on general graphs [TW93], we obtain the following general bound on the convergence time of the VOTER dynamics.

Theorem 8 (Convergence time on general topologies)**.** *Let G be any connected undirected graph. Starting from an arbitrary initial configuration c on G, the* VOTER *dynamics reaches consensus w.h.p. in $\mathcal{O}(n^3 \log n)$ rounds.*

4 Median dynamics

If we consider an *ordered* color set $\langle \Sigma, \leqslant \rangle$, then a simple consensus dynamics is the MIN dynamics: every node pulls a random neighbor (or a few of them) and then it recolors itself with the minimal color it sees.

Let us consider the synchronous $\mathcal{PULL}$ model on a connected graph $G(V, E)$ and assume that the dynamics starts with any color configuration with one node u having the minimal value min.[12] Then it is not hard to see that the resulting process is equivalent to the popular single-source broadcast on the $\mathcal{PULL}$ model which has been widely studied in algorithm theory [CLP11, CCD+16, Gia16]: For instance if the graph is regular and it has high conductance[13], the MIN dynamics turns out to be fast, i.e. $O(\log n)$ [CLP10].

However, this dynamics is not reliable in network scenarios where the state of some, few nodes can be corrupted by an adversary. Indeed, to just get a simple intution about this breakable behaviour, let us consider the time the system has reached the monochromatic configuration in which all nodes support the initial minimal value. Then it suffices that the adversary corrupts the state of *just one* node by setting its value to a smaller value than min and this small change immediatly causes the system to start a new process where *all* nodes will change the state again. If the adversary makes this small change frequently, it is clear that the system will never stays over an (almost-)stable regime.

In [DGM+11], Doerr et al proposed another natural dynamics on the synchronous $\mathcal{PULL}$ model that works on any ordered color set: the MEDIAN dynamics. Its local updating rule is

[12] Notice that if the minimal value is initially supported by more nodes, then, clearly, the process will be stochastically not slower than the the single-source process.

[13] Informally speaking, the *conductance* of a connected graph is a fundamental parameter $\phi \in (0, 1]$ which is large for highly-connected graphs (such as the complete graphs), while it is small when the graph contains some bottleneck.

very simple: at every round, every node u pulls the colors $x(v)$ and $x(w)$ of 2 neighbors chosen independently and uniformly at random; then, the color of node u at the next round will be the *median* among $x(u), x(v)$, and $x(w)$. Notice that, differently from getting the average of the values, the median rule does not introduce *new* values, i.e. values that were not supported by any node in the previous round.

Doerr et al provide an analysis of the MEDIAN dynamics on the complete graph for both the fault-free model and in presence of an F-bounded adversary that can corrupt the state of at most F nodes at every round.

The first result essentially shows that, if no node of the complete graph is ever corrupted, then the MEDIAN dynamics let the median value to spread fast over the network, as the minimal one in the MIN dynamics.

Theorem 9 (Median rule in the fault-free case [DGM$^+$11])**.** *Starting from any color configuration of the complete graph, w.h.p., the* MEDIAN *dynamics guarantees the following properties:*

- *(Convergence time.) It takes $O(\log n)$ rounds to converge to a stable monochromatic configuration.*
- *(Median computation.) The system converges to the configuration where all nodes support the median value.*

As remarked above, however, the main interest on this dynamics, lies in its tolerance w.r.t. node corruptions.

Doerr et al show that the MEDIAN dynamics is in fact a natural and efficient almost-stabilizing consensus protocol against $O(\sqrt{n})$-bounded adversary on the complete graph (see Definition 2). More precisely, they prove the following results.

Theorem 10 (Median rule vs Byzantine nodes [DGM$^+$11])**.** *Let the* MEDIAN *dynamics start from any color configuration of the complete graph. Assume there is an F-bounded adversary with $F = O(\sqrt{n})$. Then, the following properties hold:*

- *(Almost stability.) The system converges to an almost-stable regime where all but at most $O(F)$ nodes agree on some legal color $j \in \Sigma$ and this regime lasts for any $\mathrm{poly}(n)$ many rounds, w.h.p.*
- *(Convergence time.) The time to reach this almost-stable regime is bounded by $O(\log |\Sigma| \cdot \log\log n + \log n)$, w.h.p.*
- *(Median computation.) The legal value j "computed" by the system in the almost-stable regime is between the $(n/2 - c\sqrt{n \log n})$-largest value and the $(n/2 + c\sqrt{n \log n})$-largest value of the initial values, w.h.p, for a suitable constant $c > 0$.*

Key-ingredients of the analysis. The analysis of the MEDIAN dynamics proposed in [DGM$^+$11] considers two main cases.

They first consider the binary case, i.e., $\Sigma = \{0, 1\}$. Notice that, in this setting, the update rule of the MEDIAN dynamics turns out to be equivalent to the majority rule: every node takes the most frequent value he sees among its own value and the values of the two randomly-selected neighbors (notice that there cannot be ties). In this case, the analysis works in 3 phases depending on the magnitude of the bias $s = |c_0 - c_1|$ of the starting configuration (recall that c_i is the number of nodes supporting color i).

(Case 1.) If $s \geqslant n/4$, then the size of the minimal color is proved to decrease at a double-exponential rate, w.h.p.: this implies that after $O(\log\log n)$ rounds the minimal color will disappear, w.h.p.
(Case 2.) If the bias s is such that $\Omega(\sqrt{n\log n}) \leqslant s \leqslant n/4$, then the analysis focuses on the evolution of the random variable s over a logarithmic time-window. Indeed, it is proved that, at every round, the bias increases by a constant factor, w.h.p. and, thus, after $O(\log n)$ rounds, the system falls into the range of Case 1. In both Cases 1 and 2, the analysis essentially relies on a strong expected drift towards the majority color and some standard concentration arguments (i.e. Chernoff's bounds).
(Case 3.) If the system starts from some configuration having bias $s < \sqrt{n\log n}$ then the analysis requires different arguments since the drift of the bias is too weak for applying any step-by-step concentration argument. When $s \leqslant \sqrt{n}$, they essentially prove that, thanks to the variance of the process, at the next round there is constant probability that the bias reaches a magnitude of $\gamma\sqrt{n}$ for some suitable constant $\gamma > 0$. Once the system reaches a bias $s \geqslant \gamma\sqrt{n}$, then s is likely to increase at the next round. However, differently from the previous two cases, this event cannot be guaranteed to hold w.h.p. at every round and an amortized, more complex argument is required. To this aim, the authors make use of a useful bound on the hitting time on a Markov chain Z_t $(t = 1, 2, \ldots)$ defined on a finite set of integers $\{0, 1, \ldots, q\}$ which have some drift properties towards increasing values (see Claim 2.9 in [DGM$^+$11] for a formal statement of the bound). Setting $Z_t := \lfloor s/(\gamma\sqrt{n})\rfloor$ and $q = \lfloor (n/2)/\gamma\sqrt{n}\rfloor$, the bound on the hitting time implies that after $O(\log n)$ rounds, the Markov chain hits a state j such that $j \geqslant \alpha\log q$, for some suitable constant $\alpha > 0$. We thus have that the bias reached by the system after $O(\log n)$ rounds turns out to be $\Omega(\sqrt{n\log n})$, w.h.p. which let the analysis go back to Case 2 above.

The analysis of the multivalued case (i.e. $|\Sigma| \geqslant 3$) relies on some arguments which significantly depart from the binary case. It is important to observe that in the multivalued case, the Median dynamics has a local rule which is different from the majority one and it is not symmetric w.r.t. to the color values. Indeed, by definition, the median value plays a preferential role among the colors. Informally speaking, the analysis introduces a sort of rank of the colors: the more the color is close to the median, the more this rank is high and it is maximum for the median value. Then, it is shown that the dynamic makes colors with highest rank to increase their own supporters at every round. This process is then proved to converge within $O(\log n)$ time to a configuration with at most only two "median" colors and, thus, the analysis can go back to the binary case. The presence of the adversary further complicates the process analysis and it requires the additive factor $\log|\Sigma| \cdot \log\log n$ in the obtained convergence time claimed in Theorem 10.

On the validity property. It is important to investigate the kind of agreement which is guaranteed by Theorem 10 for the Median dynamics. The property of *validity* which is required in the standard definition of Consensus against Byzantine nodes (i.e. the Byzantine Agreement) is that the system must converge to an almost-stable configuration where all but a small fraction of node support a valid color, i.e., a color which was supported by at least one *non-corrupted* node in the initial configuration. Unfortunately, the analysis in [DGM$^+$11] leading to Theorem 10 does not guarantee this validity property: in particular, it is an interesting open issue to establish a bound on the fault tolerance parameter F for which the Median dynamics guarantees this stronger, classic validity property. We believe the real bound needs to be small, much smaller than $F = O(\sqrt{n})$. To motivate our intuition here, we invite the reader to think about the starting configuration with $n/2$ nodes colored with value 0 and $n/2$ nodes colored with value 2. Then, the adversary takes $m = \Theta(\text{polylog}\,(n))$ nodes at every round and it colors them with value 1. Non-rigorous, "mean-field" arguments lead us to guess that, with non-negligible probability, the non-valid, "median" value 1 will spread faster than any possible growth of the

other two valid colors, thus forcing the system to converge to the non valid color 1.

The MEDIAN dynamics over non-complete topologies. We are not aware of rigorous analysis of the MEDIAN dynamics that works over non-complete graphs. However, in the binary case, the MEDIAN dynamics turns out to be equivalent to the popular majority dynamics known as 2-CHOICES (see Section 5.1) and the latter has been recently analyzed over regular expander graphs.

5 Majority dynamics

The MEDIAN dynamics protocol fully relies on the presence of some total ordering on the color set Σ. In some scenario, such as the ones inspired by biological systems, this assumption might be not reasonable: Particles and bio-inspired agents may have no ordering on the possible states they can assume. Moreover, as remarked in the previous section, it is unknown (and, in our opinion, unlike) whether MEDIAN dynamics can guarantee the important *validity* property, with high probability.

A class of natural dynamics, requiring no color ordering, are those based on *majority rules*. In the synchronous $\mathcal{PULL}$ model, h-MAJORITY dynamics works as follows:

Definition 11 (h-MAJORITY dynamics)**.** *At every round, every node u samples $h \geqslant 1$ neighbors, independently and uniformly at random.*
Then, u gets the plurality *color among those in the sample; if there is no plurality in the sample, u gets the first sampled color.*[14]

An important instance of the h-MAJORITY dynamics is the 3-MAJORITY dynamics. The reason of its relevance is essentially that a sample of size 3 is the minimal one to hope for a different behaviour from the VOTER dynamics seen in Section 3. Indeed, looking at only two random nodes and breaking ties uniformly at random would yield a coloring process equivalent to the VOTER dynamics, and the latter it is known to converge to a *minority* color with constant probability even in the case of 2-color initial configurations having a large initial bias (see Section 3).

Another popular majority rule is the 2-CHOICES. It differs from the 3-MAJORITY dynamics on the fact that node u samples the colors of (only) two random neighbors. Then, it applies the plurality rule (like 3-MAJORITY dynamics) among such two random colors *and* its own color. So, in the 2-CHOICES, the updating rule deterministically depends on the current state of the node while 3-MAJORITY dynamics gives no special role to the current state of the node. As we will see later in Subsection 5.3, even though the expected behaviours of 3-MAJORITY dynamics and 2-CHOICES are equivalent, their "real" behaviours turn out to be significantly different when the number k of the initial colors is large.

5.1 2-Choices dynamics

It is easy to see that, in the binary case (i.e. $|\Sigma| = 2$), MEDIAN dynamics turns out to be equivalent to 2-CHOICES. As for the latter, we thus have the following immediate consequences of the results [DGM+11] shown in the previous section.

Corollary 12 (Two colors, [DGM+11])**.** *Let $\Sigma = \{0, 1\}$. Let the system start from any color configuration and assume there is an F-bounded adversary with $F = O(\sqrt{n})$. Then, 2-*CHOICES *is a stabilizing almost (Plurality) Consensus protocol. In more detail, it guarantees the following properties:*

[14] Or, equivalently, a randomly chosen one among those in the sample.

- *(Almost Stability.) The system converges to an almost-stable regime where all but at most $O(F)$ nodes agree on some* valid *color $j \in \Sigma$ and this regime lasts for any $\mathrm{poly}(n)$ many rounds, w.h.p.*

- *(Convergence Time.) The time to reach this almost-stable regime is bounded by $O(\log n)$, w.h.p.*

- *(Plurality Consensus.) Assume the initial configuration has bias s towards a plurality color such that $s \geqslant c\sqrt{n \log n}$, for some constant $c > 0$. Then, the color the system converges to in the almost-stable regime is the plurality one, w.h.p.*

As we will also see in further results described in this survey about dynamics for Plurality Consensus, the bias threshold $\sqrt{n \log n}$ will show up rather often. Informally speaking, this is due to the fact that the standard deviation (in the binary case) of the studied process is order of $\sqrt{n}$. This implies that if the system starts from bias "too" close to the magnitude of the standard deviation, some anti-concentration bounds say us that the system can change plurality color in one round, with non-negligible probability. Thus, in order to preserve the initial plurality and to get results in concentration, the system must start from a bias $\Omega(\sqrt{n \log n})$.

The multi-color case. A deep analysis of this dynamics for $k = O(n^{\varepsilon})$ has been recently presented by Elsässer et Al in [EFK$^+$17]. They consider initial configurations $\mathbf{c} = (c_1, c_2, \ldots, c_k)$ having some initial bias $s = c_1 - c_2$ and show that this dynamics is an efficient, fault-tolerant protocol for Plurality Consensus.

Theorem 13 (More colors, biased initial configurations, [EFK$^+$17])**.** *Let $\Sigma = \{1, 2, \ldots, k\}$ with $k = O(n^{\varepsilon})$ where ε is a su ciently small constant*[15]*. Then 2-*Choices *is a stabilizing almost Plurality Consensus protocol. In detail, let the system start from any color configuration $\mathbf{c} = (c_1, c_2, \ldots, c_k)$ with bias $s \geqslant \gamma\sqrt{n \log n}$ where $\gamma > 0$ is some positive constant and assume there is an F-bounded adversary with $F \leqslant c_1(c_1 - c_2)/(8n)$. Then,*

- *(Almost Stability.) The system converges to an almost-stable regime where all but at most $o(n)$ nodes agree on some* valid *color $j \in \Sigma$ and this regime lasts for any $\mathrm{poly}(n)$ many rounds, w.h.p.*

- *(Convergence Time.) The time to reach this almost-stable regime is bounded by $O((n/c_1)\log n)$, w.h.p. Notice that, since it always holds that $c_1 \geqslant n/k$, then the latter bound also implies the bound $O(k \log n)$.*

- *(Plurality Consensus.) The color the system converges to in the almost-stable regime is the plurality one, w.h.p.*

Moreover, if the initial configuration $\mathbf{c}$ is such that $s \geqslant \gamma\sqrt{n \log n}$ and $c_j = c_2$ for $j = 3, \ldots, k$, then the expected time required to converge to the almost-stable regime is $\Omega(n/c_1 + \log n)$.

The proof of the above theorem relies on the following arguments. By using Chernoff bounds, Elsässer et al. show that the number of nodes which change their color to 1 is larger than the number of nodes which switch to color 2. From the lower bound on the initial bias s assumed by the above theorem, the bias increases by some positive factor in the next round w.h.p, and using a union bound over an $O((n/c_1)\log n)$ sequence of rounds, they get the claimed upper bound on the convergence time. The main technical issue lies in bounding the number of nodes moving to color 1 and 2. Indeed, just applying a Chernoff bound to every single color community would

[15] The current analysis in [EFK$^+$17] does not give an explicit value on ε.

lead to much weaker results. Instead, they carefully aggregate colors when considering the nodes moving to one color to the other. Intuitively, the difficulty lies in the sheer number of initial colors the new analysis allows. Their total support may significantly exceed c_1.

Starting from unbiased configurations. A natural issue on 2-Choices concerns its behaviour when starting from balanced configurations, i.e., whenever s turns out to be relatively small, say $s = o(\sqrt{n})$.

This setting has been analyzed in a recent work by Ghaffari et Al in [GL18]. They provide a tight bound on the convergence time starting from any configuration with a polynomially-bounded number of colors, i.e., $k = O(\sqrt{n/\log n})$.

Theorem 14 (More colors, balanced intial configurations [GL18])**.** *Let $\Sigma = \{1, 2, \ldots, k\}$ with some $k = O(\sqrt{n/\log n})$. Then, 2-Choices is a stabilizing almost Consensus protocol. In more detail, let the system start from any color configuration $\mathbf{c}$ and assume there is an F-bounded adversary with $F = O(\sqrt{n}/k^{1.5})$. Then,*

- *(Almost stability.) The system converges to an almost-stable regime where all but at most $o(n)$ nodes agree on some* valid *color $j \in \Sigma$ and this regime lasts for any $\mathrm{poly}(n)$ many rounds, w.h.p.*
- *(Convergence time.) The time to reach this almost-stable regime is bounded by $\Theta(k \log n)$, w.h.p.*

The analysis in [GL18] uses and improves an approach which has been previously introduced by Becchetti et Al in [BCN+16] to analyze 3-Majority dynamics in a similar setting (see Subsection 5.2). When the process starts from an (almost-)balanced configuration, any first-moment analysis of the expected values of the color size is almost useless: in average, the system stays in the same (almost-)balanced configuration. On the other hand, the real process starts to jump randomly over several (almost-)balanced configurations thanks to its variance until the symmetry among the color sizes is broken: we can then observe the birth of some plurality color that starts to increase rapidly. It thus follows that a non-standard second-moment analysis is required that makes use of anti-concentration bounds for multinomial distributions.

Analysis over non-complete topologies. In [CER14], Cooper et al. analyze, for the first time, the random process yielded by applying 2-Choices over d-regular graphs. They consider the binary case and give different bounds on the convergence time depending on several input parameters, such as node degree d, initial bias s, and on which class of regular graphs the process runs on. We give here two of the bounds mentioned above.

The first result concerns a graph which is sampled uniformly at random from the class of d-regular graphs of n nodes.

Theorem 15 (Binary case on sparse random graphs [CER14])**.** *Let G be a d-regular random graph with $d \leqslant \sqrt{n}$ and assume the initial color configuration has bias $s \geqslant \gamma n/\sqrt{d}$, for a large enough constant $\gamma > 0$. Then, with probability $1 - o(1)$, 2-Choices converges to the initial majority within $O(\log n)$ rounds.*

The second results provided in [CER14] concerns (deterministic) regular graphs having good expansion properties. A graph G is said to be an (edge) ε-*expander* if, for any node subset S of size not larger than $n/2$, it holds $e(S, V - S) \geqslant \varepsilon|S|$, where $e(S, V - S)$ denotes the number of edges in the cut $E, V - S)$. The expansion properties of the underlying d-regular graph $G(V, E)$ can be expressed in terms of the spectral gap $\lambda_1 - \lambda_2$, where λ_1 and λ_2 are respectively the largest and the second-largest (in absolute value) eigenvalue of the transition matrix $P = (1/d) \cdot A$ of G (A denotes the symmetric adjacency matrix og G - see also Section 7.2 for further useful concepts

in spectral graph theory). If G is connected and not bipartite then it holds that $\lambda_1 = 1$, $\lambda_2 < 1$ and $|\lambda_i| \geqslant 1$, for every $i = 1, \ldots, n$. Roughly speaking, the larger is the spectral gap the larger is the (edge) expansion of the graph. This fact is formalized by the well-known Expander Mixing lemma which is a key-ingredient in Cooper et al.'s analysis:

Lemma 16 (Expander mixing lemma [AC88]). *Let G be a connected, non-bipartite d-regular graph, then, for any $S, T \subset V$, it holds*

$$\left| e(S,T) - \frac{d|S| \cdot |T|}{n} \right| \leqslant \lambda_2 d \sqrt{|S| \cdot |T|}\,.$$

Theorem 17 (Binary case on expander graphs [CER14]). *Let G be a d-regular graph with $d \leqslant \sqrt{n}$ and second largest eigenvalue $0 < \lambda_2 < 1$, and assume the initial color configuration has bias $s \geqslant \gamma \lambda_2 n$, for a large enough absolute constant $\gamma > 0$. Then, with probability $1 - o(1)$, 2-Choices converges to the initial majority within $O(\log n)$ rounds.*

We first notice that the success probability guaranteed in the above theorems is weaker than that defined in the standard notion *"w.h.p."* we adopt in this paper.

Roughly, the proofs of the above two bounds on the convergence time of the 2-Choices process rely on the following key-ingredients. Let us look at the color configuration of G at a generic round $t \geqslant 0$ and consider the sizes c_1 and c_2 of subsets C_1 and C_2 of nodes that support the majority color and the minority one, respectively. Let us also define the two random variables Δ_{12} and Δ_{21} which count the number of 1-colored (2-colored) nodes that will get color 2 (1), at round $t+1$. A first simple but crucial fact is that, if a 2-colored node changes its color at the next round, then, according to the 2-Choices, it must have sampled the other color twice from its neighborhood, i.e., from some neighbors that must belong to C_1. Then, it is easy to verify that that the expected number $\mathbf{E}[\Delta_{21}]$ of nodes that makes this change is an increasing function of the expansion of subset C_2, i.e., of the size $e(C_2, V - C_2 = C_1)$. A symmetric argument clearly holds for $\mathbf{E}[\Delta_{12}]$. Now, if G is a good expander, say $\lambda_2 < 1/3$, their analysis shows that Lemma 16 implies a positive drift for the expectation $\mathbf{E}[\Delta_{12} - \Delta_{21}]$. This analysis is organized in few consecutive phases that depend on the range the random variable c_1 (and, thus, c_2) currently lies on during the process. Several technical issues are neglected by our informal description above which are essentially due to get concentration bounds even when the considered random variables takes small values.

Further analysis of the 2-Choices dynamics on non-complete graphs consider the binary case over non-regular expander graphs [CER$^+$15] and the multi-color case on regular expander graphs [CRRS17]. The techniques introduced in these works are combined in [CNS19] with the community-sensitive labeling scheme introduced in [BCN$^+$17c] (see Section 7 for further details on the latter), in order to use the 2-Choices dynamics to perform community detection.

Finally, the 2-Choices has also been proposed in recent work by Cruciani et al. [CNNS18] as a natural, simplistic model of competition among opinions in social networks having core-periphery topologies. The author consider a network model where a densely-connected subset of agents, the core, holds a different opinion from the rest of the network, the periphery. Then, depending on the strength of the cut between the core and the periphery, they provide analytical evidence of the existence of a phase-transition phenomenon: Either the core's opinion rapidly spreads among the rest of the network, or a metastability phase takes place, in which both opinions coexist in the network for superpolynomial time. This result sheds light on the influence of the core on the rest of the network, as a function of the core's expansion over the latter. Moreover, this represents the first analytical result which shows a heterogeneous behavior of a "pure" majority dynamics as a function of structural parameters of the network; similar behaviour have been previously proved for averaging dynamics only, that is, when opinions admit numerical operations (see Section 5).

5.2 The 3-Majority dynamics

From an historical point of view, the study of the 3-MAJORITY dynamics started on the ground of the results obtained for the MEDIAN dynamics by Doerr et Al in [DGM+11] that we have described in Section 4. Indeed, as discussed in that section, the MEDIAN dynamics is a fault-tolerant, efficient dynamics which, however, does not guarantee to reach a valid consensus (see the discussion at the end of Section 4). This fact naturally leads researchers to look for efficient dynamics having this further fundamental property.

3-MAJORITY dynamics for Plurality Consensus

The 3-MAJORITY dynamics was the first dynamics which has been proved to be a fault-tolerant protocol for (valid) Plurality Consensus. Such results were proved by Becchetti et Al in [BCN+17a].

Theorem 18 (Plurality Consensus [BCN+17a]). *Let $\Sigma = \{1, 2, \ldots, k\}$ with* any *with $k \leqslant n$. Then,* 3-MAJORITY *dynamics is a stabilizing almost Plurality Consensus protocol. In details, let λ be any value such that $\lambda < \sqrt[3]{n}$, and let $\mathbf{c}$ be any initial color configuration, with $c_1 \geqslant n/\lambda$ and $s \geqslant 72\sqrt{2\lambda\, n \log n}$. Assume there is an F-bounded adversary with $F = o(s/k)$. Then, the* 3-MAJORITY *dynamics guarantees the following properties:*

- *(Almost-stable Plurality Consensus.) The system converges to an almost-stable regime where all but at most $o(n)$ nodes agree on the (*valid*) initial plurality color and this regime lasts for any* $\mathrm{poly}(n)$ *many rounds, w.h.p.*
- *(Convergence time.) The system converges to the plurality color within $\mathcal{O}(\lambda \log n)$ time, w.h.p.*

The analysis achieving the theorem above is accurate enough to get an interesting form of the upper bound that does not depend on k. Indeed, when the initial plurality size c_1 is larger than $n/\lambda(n)$ for any function $\lambda(n)$ such that $1 \leqslant \lambda(n) < \sqrt[3]{n}$ and $s \geqslant 72\sqrt{2\lambda(n)\, n \log n}$, the process converges within time $\mathcal{O}(\lambda(n) \log n)$, w.h.p., no matter how large k is. Hence, when $c_1 \geqslant n/\mathrm{polylog}(n)$ and $s \geqslant \sqrt{n\, \mathrm{polylog}(n)}$, the convergence time is polylogaritmic.

Becchetti et Al [BCN+17a] show that the upper bound on the convergence time in Theorem 18 is tight for a wide range of the input parameters. When $k \leqslant (n/\log n)^{1/4}$, they in fact prove a lower bound $\Omega(k \log n)$ on the convergence time when the 3-MAJORITY dynamics starts from some configurations with bias $s \leqslant (n/k)^{1-\varepsilon}$, for an arbitrarily small constant $\varepsilon > 0$. Observe that this range largely includes the initial bias required by the upper bound whenever $k \leqslant (n/\log n)^{1/4}$. So, the *linear-in-$k$* dependence of the convergence time cannot be removed for a wide range of the parameter k.

The analysis also provides a clear picture of the process yielded by the 3-MAJORITY dynamics. Informally speaking, the larger the initial value of c_1 is (w.r.t. n), the smaller the required initial bias s and the faster the convergence time are. On the other hand, the lower-bound argument shows, as a by-product, that, from balanced configurations, the initial plurality size c_1 needs $\Omega(k \log n)$ rounds just to increase from $n/k + o(n/k)$ to $2\, n/k$.

Another natural issue is to analyze the process under weaker assumptions on the initial bias. Becchetti et Al [BCN+17a] show that there are initial configurations with bias $s = O(\sqrt{kn})$ for which the bias decreases in a single round with constant probability. This implies that the assumption on the magnitude of the initial bias is in a sense (almost) tight if one wants to prove a monotonic increase of the bias in every round with high probability, as they did in proving the upper bound on the convergence time.

Analysis over non-complete topologies. Very recently [KR19], the 3-MAJORITY dynamics was analyzed over a class of dense graphs with minimum degree $d = \Omega(n^\alpha)$, where $\Omega((\log\log n)^{-1}) \leqslant \alpha \leqslant 1$, and starting from random, biased binary configurations . In more detail, [KR19] prove that if initially each vertex chooses red with probability greater than $1/2 + \delta$, and blue otherwise, where $\delta \geqslant (\log d)^{-\beta}$, for some $\beta > 0$, then the dynamic solves Plurality Consensus within $O(\log\log n + \log(\delta^{-1}))$ rounds, w.h.p.

3-MAJORITY dynamics for Consensus

In [BCN+16], Becchetti et al extend the analysis of 3-MAJORITY dynamics in the multi-color case from any (even balanced) initial configuration in the presence of an adaptive F-dynamic adversary and of the F-*static* adversary (in the latter, the adversary looks at the initial configuration, then changes the opinion of up to F nodes and, after that, no further adversary's actions are allowed).

Theorem 19 (Consensus [BCN+16]). *Let $k \leqslant n^\alpha$ for some constant $\alpha < 1$ and $F = \beta\sqrt{n}/(k^{\frac{5}{2}}\log n)$ for some constant $\beta > 0$. Starting from any initial configuration having k valid opinions, 3-MAJORITY dynamics reaches a stabilizing almost-consensus in the presence of any F-dynamic adversary within $O((k^2\sqrt{\log n} + k\log n)(k + \log n))$ rounds, w.h.p.*
Moreover, the same bound on the convergence time holds in the presence of any F-static adversary with a larger bound on F, i.e., $F = n/k - \sqrt{kn\log n}$.

Not assuming a large initial bias of the plurality opinion considerably complicates the analysis. Indeed, the major open issue here is the analysis from (almost) uniform configurations, where the system needs to break the initial symmetry in the absence of significant drifts towards any of the initial opinions. Notice that the phase before symmetry breaking is the one in which the adversary has more chances to cause undesired behaviours: Long delays and/or convergence towards non-valid opinions. Let us consider the non-adversarial case. When the configuration is (approximately) uniform, the process exhibits no significant drift toward any *fixed* opinion. Interestingly, things change if we consider the random variable $c_{\mathrm{m}}^{(t)}$, indicating the smallest opinion support at round t. Let $j \leqslant k$ be the number of active opinions in a given round t, it is first proved that the expected value of $c_{\mathrm{m}}^{(t)}$ always exhibits a non-negligible negative drift:

$$\mathbf{E}\left[c_{\mathrm{m}}^{(t+1)} \mid \mathbf{c}^{(t)} = \hat{\mathbf{c}}\right] \leqslant c_{\mathrm{m}} - \varepsilon\frac{\sqrt{n}}{j^{3/2}}\,, \text{ for some constant } \varepsilon > 0\,. \tag{1}$$

The analysis then proceeds along consecutive phases, each consisting of a suitable number of consecutive rounds. If the number of active opinions at the beginning of the generic phase is j, it is shown that, with positive constant probability, $c_{\mathrm{m}}^{(t)}$ vanishes within the end of the phase, so that the next phase begins with (at most) $j - 1$ active opinions.
Clearly a good bound is needed on the length of a phase beginning with at most j opinions. To this aim, the authors derive a new upper bound on the *hitting time* of stochastic processes with expected drift that are defined by finite-state Markov chains [LPW09]. Thanks to this result, they can use the negative drift in (1) to prove that, from any configuration with $j \leqslant k$ active opinions, $c_{\mathrm{m}}^{(t)}$ drops below the threshold $n/j - \sqrt{jn\log n}$ within $O(\mathrm{poly}(j, \log n))$ rounds, with constant positive probability: This "hitting" event represents the exit condition from the symmetry-breaking stage of the phase. Indeed, once it occurs, we can consider *any fixed* active opinion i having support size c_i below the above threshold (thanks to the previous stage, we know that there is a good chance this opinion exists): It is then shown that c_i has a negative drift of order $\Omega(c_i/j)$. This allows to prove that c_i drops from $n/j - \sqrt{jn\log n}$ to zero within

$O(\text{poly}(j, \log n))$ further rounds, with positive constant probability. This interval of rounds is the dropping stage of the phase.
Ideally, the process proceeds along k consecutive phases, indexed as $j = k, k-1, \ldots, 2$, such that we are left with at most $j-1$ active opinions at the end of Phase j. In practice, there is only a constant probability that at least one opinion disappears during Phase j. However, using standard probabilistic arguments, it is proved that, w.h.p., for every j, the transition from j to $j-1$ active opinions takes a constant (amortized) number of phases, each requiring $O(\text{poly}(j, \log n))$ rounds.

The upper bound in Theorem 19 was later improved in [GL18], using a more refined analysis that does not require k consecutive phases (see Theorem 14 and the discussion below it).

5.3 2-Choices vs 3-Majority dynamics

The results described in the previous two subsections naturally lead to investigating the main differences between the 2-Choices and the 3-Majority dynamics. This issue has been recently addressed by Berenbrink et Al in [BCE+17].

We notice that Theorem 14 provides an upper bound on the convergence time for the 2-Choices for a *limited* number of colors (i.e. $k = O(\sqrt{n/\log n})$). On the other hand, when k is almost linear in n, the behaviour of the 2-Choices turns out to be rather "lazy", as stated by the following result.

Theorem 20 (2-Choices: Many colors, balanced initial configurations [BCE+17])**.** *Let $\Sigma = \{1, 2, \ldots, k\}$ with some $k = \Omega(n/\log n)$. Let the system start from any color configuration $\mathbf{c} = (c_1, c_2, \ldots, c_k)$ such that $c_1 = O(\log n)$, then, the 2-Choices needs $\Omega(n/\log n)$ rounds to converge to a monochromatic configuration, w.h.p.*

The proof of the above theorem shows that, when started from an almost balanced configuration, the convergence time is dominated by the time it takes for one of the colors to gain a community of supporters of size $\omega(\log n)$. The analysis adopts the following line of reasoning. If the system lies in an almost-balanced configuration with an almost linear number of colors, then most nodes are likely to see two different colors in their respective 2 randomly-selected neighbors. This implies that, according to the 2-Choices updating rule, such nodes will keep their own color for several rounds. In other words, under this setting, the variance of the process is relatively low.

Does the 3-Majority dynamics show the same "lazy" behaviour when it starts from many-colors, unbiased configurations? The answer to the above question is given by the following result.

Theorem 21 (3-Majority dynamics: Many colors, balanced initial configurations [BCE+17])**.** *Let $\Sigma = \{1, 2, \ldots, k\}$ with* any *$k \leqslant n$. Starting from an* arbitrary *configuration, the* 3-Majority *dynamics converges to a monochromatic configuration within $O(n^{3/4} \log^{7/8} n)$ rounds, w.h.p.*

Theorems 20 and 21 together show that there can be a *polynomial* gap between the convergence time of 2-Choices and 3-Majority dynamics: one should note that this gap is in stark contrast not only to the expected behavior of both processes (which is identical - see for instance [BCN+17a, BCE+17]) but also to the setting when $k = O(n^{\varepsilon})$ (for some $\varepsilon < 1$) and a good bias towards one color: essentially, we have seen in the previous subsections that, in that case, both processes exhibit the same asymptotic runtime $O(k \cdot \log n)$.

Interestingly enough, the proof of Theorem 21 introduces (and exploits) a new coupling between the 3-Majority dynamics and the Voter dynamics (see Section 3). Berenbrink et al. show that the time needed by 3-Majority dynamics to reduce the number of colors to

a fixed value is stochastically dominated by the time VOTER dynamics needs for this. This, finally, allows to upper bound the time needed by 3-MAJORITY dynamics to go from $\Theta(n)$ to $\Theta(n^{1/4})$ colors by the time VOTER needs for this (which, in turn, they bound by $O(n/k)$ - see [BCE+17, Lemma 2 and 3]). The authors also observe that, for a large number of colors, a node executing 3-MAJORITY dynamics behaves like a node performing VOTER dynamics. Thus, it is relatively tight to bound 3-MAJORITY dynamics by VOTER dynamics in this parameter regime. The coupling works for a wide class of processes which are essentially defined by an update rule that causes each node to adopt any color i with the same probability α_i that depends only on the current frequency of colors (see [BCE+17] for a more formal description of this general result).

5.4 Further results on majority dynamics

Majority over larger random samples. In Subsection 5.2, we observed that the 3-MAJORITY dynamics takes $\Theta(k \log n)$ time steps w.h.p. to converge in the worst case: the linear-in-k dependence of the convergence time cannot be removed for a large range of k [BCN+17a].

A natural question thus arises: using h-MAJORITY dynamics, with h slightly larger than 3 (say with a sample of size $h = \log n$), is it possible to significantly (say exponentially w.r.t. to k) speed-up the process?

Becchetti et al. [BCN+17a] show this is not the case. Indeed, assuming $k/h = O(n^{(1-\varepsilon)/4})$, they prove that, for any $j \in [k]$ such that $c_j \leqslant 2n/k$, the probability that, at the next round, c_j increases by a factor $(1+\Theta(h^2/k))$ is exponentially small. This technical result is then exploited to derive the following lower bound on the convergence time of h-MAJORITY dynamics.

Theorem 22 (A lower bound for h-MAJORITY dynamics [BCN+17a])**.** *Let $k/h = O(n^{(1-\varepsilon)/4})$ for some constant $\varepsilon > 0$ and let h-MAJORITY dynamics starts from any initial configuration $\mathbf{c} = (c_1, c_2, \ldots, c_k)$ such that $c_1 \leqslant \frac{3}{2} \cdot \frac{n}{k}$. Then, the convergence time is $\Omega(k/h^2)$, w.h.p.*

Dynamics studied here, and more generally, any scalable and efficient protocol must yield low communication complexity and small node congestion at every time step. These properties are guaranteed by the h-MAJORITY dynamics only if h is small, say $h = O(\text{polylog}\,(n))$: in this case, the above lower bound says that the resulting speed-up is only polylogarithmic with respect to the 3-MAJORITY dynamics.

Negative results for other 3-inputs dynamics. A further natural question on Plurality Consensus is whether there are simple (and efficient) mechanisms, not based on majority rules, that can achieve plurality consensus. In [BCN+17a], Becchetti et al. show there is no dynamics with at most 3 inputs (but the majority one) that w.h.p. converges to plurality consensus starting from any initial bias s such that $s = o(n)$. In other words, not only there is no hope to find a 3-input dynamics that is asymptotically faster than $k \log n$ but the 3-majority dynamics is the only one preserving the initial plurality, no matter in how much time.

6 Dynamics using extra states

In the VOTER dynamics a node just gets the color it sees. This "social" behaviour is often too simplistic to model opinion dynamics. Moreover, the results of Section 3 show that VOTER dynamics does not achieve fast consensus. The local function of the majority dynamics we discussed in Section 5 have three ore more inputs (each of size $\lceil \log |\Sigma| \rceil$) and, thus, nodes requires at least $\lceil \log |\Sigma| \rceil^3$ states to implement them. A natural and important question is whether dynamics with a smaller number of states can achieve fast (Plurality) Consensus.

The issue above leads to consider a dynamics having just one extra state, called UNDECIDED-STATE dynamics: let nodes stay "undecided" whenever they see a different color from their own current belief.

Updating rule of the Undecided-State dynamics. Let us assume that the set of possible colors a node can support is $[k]$ and assume that, at a given round, the generic node u samples a random node v. Then if the color of v differs from that of u, the latter enters the *undecided* state, an extra state that an agent can support. When an agent is in the undecided state and pulls a color, it gets that color. Finally, an agent that pulls either the undecided color or its own color remains in its current state (see Table 1).

$u \backslash v$	undecided	color i	color j
undecided	undecided	i	j
i	i	i	undecided
j	j	undecided	j

Table 1: The update rule of the Undecided-State dynamics where $i, j \in [k]$ and $i \neq j$ are the colors a node can support.

We remark that, differently from majority rules we described in the previous section, this rule is not symmetric w.r.t. colors (i.e. states). As we will see this difference is a key-ingredient for fast convergence and for other important features of the Undecided-State dynamics.

We remark that this dynamics only requires $k+1$ states for the nodes. Moreover, the interest for the Undecided-State dynamics touches areas beyond the borders of computer science and it appears to play a major role in important biological processes modelled as so-called chemical reaction networks [CCN12, Dot14].

6.1 Two colors in the random sequential model

In [AAE08], Angluin et al study the Undecided-State dynamics on the complete graph of n nodes by adopting the asynchronous, sequential model[16] where, at every round, one oriented edge (u, v) is chosen uniformly at random and then node u applies the local rule of the dynamics by pulling the state of v (according to Table 1).

The authors provide an unconditional analysis for the binary case (i.e. $|\Sigma| = 2$) when the process starts from any possible color configuration, also including undecided nodes. Since the system is anonymous and the graph is complete, the configuration (or global state) of the process can be represented by a triple (c_0, c_1, q) where c_i is the number of nodes supporting color i ($i = 0, 1$) and q is the number of undecided nodes. We call $\mathcal{C}$ the set of all possible configurations.

It is easy to verify that the process yielded by Undecided-State dynamics is a reversible finite Markov chain which admits 3 absorbing monochromatic states: $(n, 0, 0), (0, n, 0), (0, 0, n)$. Then, standard arguments on Markov chains shows that, starting from any configuration, with probability 1, the system converges to one of these absorbing states in finite time.

Notice that, in the sequential model, such absorbing states also represent *valid* consensus. Indeed, since at most one node can change state in 1 round, if the initial configuration has no undecided nodes, then the process converges to a colored configuration.

The study of Angluin et al. addresses the three main issues which characterized this survey. The worst-case convergence time: starting from any fixed configuration how long does the system take to reach an absorbing state, i.e., to reach consensus? The plurality-preserving property: Assuming the system starts with a sufficiently large bias towards one of the two colors, say color 1, then does the system converge to the monochromatic configuration $(0, 1, 0)$, w.h.p.? The fault-tolerance of the protocol, or its self-stabilization properties: assuming at every round there

[16] In particular, they adopted the (uniform) *Population-Protocols model* [AAFJ08].

is a "small" set of Byzantine nodes which can behave adversarily w.r.t. the protocol, then does the system stabilize to an almost, valid consensus (i.e. to a configuration where $n - o(n)$ nodes support the same valid color), w.h.p.?

The analysis in [AAE08] consider the configuration space $\mathcal{C}$ as a 2-dimensional triangle and define a partition into four main subregions: each of the three corner regions is characterized by a "large" fraction of nodes supporting one of the three possible states while the fourth region is a large "central" one which is characterized by a balanced number of nodes assuming each possibile state.

The complex evolution of UNDECIDED-STATE dynamics is essentially due to the fact that its random behaviour is *qualitatively di erent in di erent regions of the conguration space*. In order to get the good intuitions here, we observe that when the system lies in configurations with a *large* bias $s = |c_1 - c_0|$ and $q = \Theta(n)$, then s increases, in average. On the other hand, when the system lies in almost balanced configurations with a large $q = \Theta(n)$, then the evolution is very close to a double, almost-independent information-spreading process of the two colors. Finally, when the system lies in almost balanced configurations where no color predominates, the process works like a random walk with an increasing drift toward configurations having a relatively large bias. The main technical contribution of [AAE08] is the definition of a suitable potential function over the set $\mathcal{C}$ which well-represents the critical region-transitions of the process evolution. On the other hand, they show that the random process defined by this potential function is essentially a sub-martingale and, thus, it is possible to bound its convergence time.

This important analysis provides rigorous and tight answers to all the three main issues discussed above.

Theorem 23 (Sequential UNDECIDED-STATE dynamics [AAE08])**.** *Assume the* UNDECIDED-STATE *dynamics starts from an arbitrary configuration*[17] $\mathbf{c} \neq (0, 0, n)$*. Then the following claims hold.*

- Convergence Time*. After $O(n \log n)$ rounds (i.e. edge activations), the* UNDECIDED-STATE *dynamics converges to a valid monochromatic configuration $\bar{\mathbf{c}} \neq (0, 0, n)$, w.h.p.*
- Plurality-Preserving Property*. If the initial configuration $\mathbf{c}$ has bias $s = \omega(\sqrt{n \log n})$, then the* UNDECIDED-STATE *dynamics converges to the initial plurality color, w.h.p.*
- Fault-Tolerance*. If the number of Byzantine nodes is $o(\sqrt{n})$, then, the two claims above still hold w.h.p.: the* UNDECIDED-STATE *dynamics reaches valid consensus and it preserves the initial plurality provided that $s = \omega(\sqrt{n \log n})$.*

As an important consequence of the above theorem, the UNDECIDED-STATE dynamics turns out to be a fault-tolerant protocol that achieves binary consensus and plurality consensus with a logarithmic work per node. So, similarly to the 3-MAJORITY dynamics or to the 2-CHOICES dynamics, it yields an exponential speed-up w.r.t. VOTER dynamics at the cost of just one extra state the nodes can assume.

Analysis for non-complete topologies

Recently, [MNRS16] provides an analysis of the UNDECIDED-STATE dynamics over arbitrary connected graph G starting from binary configurations. The first result concerns the case in which the initial configuration is chosen randomly.

[17] If the process starts with all nodes in the undecided state, then it clearly stays on this configuration forever.

Theorem 24 (Sequential UNDECIDED-STATE dynamics on general graphs - random initialization [MNRS16]). *Let $G(V, E)$ be any connected graph and assume each node chooses its initial color from $\Sigma = \{0, 1\}$ randomly according to any fixed distribution. Then,* UNDECIDED-STATE *dynamics converges to the initial majority with probability at least $1/2$.*

The proof of the above result makes use of a classic result in extremal combinatorics (in particular, on the Hall's marriage Theorem) and gives no interesting bounds on the completion time.

Mertzios et al [MNRS16] also consider special classes of graphs and initial configurations which are chosen in some adversarial way in order to show very strong negative results.

Theorem 25 (Sequential UNDECIDED-STATE dynamics - worst-case settings [MNRS16]). *Some infinite families of connected graphs together with some (adversarial) initial color configurations exist such that the* UNDECIDED-STATE *dynamics converges to the initial minority color with high probability. The result holds even starting with a very large bias, i.e. for $s = n - \Theta(\log n)$. Moreover, some infinite families of connected graphs together with some (adversarial) initial color configurations exist where the expected time of the* UNDECIDED-STATE *dynamics is exponential in n.*

The negative results above essentially relies on the good fault-tolerance the UNDECIDED-STATE dynamics exhibits over the complete graphs. Informally, the authors consider a lollipop topology where the core is a small clique with one of its nodes connected to a long path. If the clique is initialized with the minority color and the path with the majority one then they prove there is a high probability that the former will win.

A four-states dynamics over non-complete topologies

It is perhaps not surprising that there is no one-extra-state dynamics that have "better" performances than the UNDECIDED-STATE dynamics; this result has been formally proved in [MNRS16]. What about using two extra-states (i.e., four states for the binary case)? This issue has been investigated in a sequence of works that we summarize in this subsection.

A dynamics was introduced by Bénézit et al. in [BTV09], in which they study its properties for reaching some forms of consensus. The convergence time for the binary case was first analyzed in [DV12]. According to [DV12], the dynamics will be denoted as INTERVAL-CONSENSUS dynamics. The set of possible states is $\Sigma = \{0, 0.5^-, 0.5^+, 1\}$ and the following order relation on this set is introduced: $0 < 0.5^- < 0.5^+ < 1$. Since we are in the random sequential model, the states held by the nodes are updated at random pairwise contacts between nodes sharing a link of the supporting graph.

The idea is to "simulate" a local *quantized averaging* process between any two interacting nodes so that the total "mass" is preserved and swapped. Formally, the updating rule of INTERVAL-CONSENSUS dynamics is described below [DV12].

INTERVAL-CONSENSUS dynamics.

A. If a node in state 0 and a node in state 1 interact, they update their states to state 0.5^+ and state 0.5^-, respectively.

B. If a node in state 0.5^- and a node in state 1 interact, they update their states to state 1 and state 0.5^+, respectively.

C. If a node in state 0.5^+ and a node in state 0 interact, they update their states to state 0 and state 0.5^-, respectively.

D. If a node in state 0.5^- and a node in state 0 interact, they update their states to state 0 and 0.5^-, respectively.

E. If a node in state 0.5^+ and a node in state 1 interact, they update their states to state 1 and state 0.5^+, respectively.

F. If a node in state 0.5^- and a node in state 0.5^+ interact, they update their states to state 0.5^- and state 0.5^+, respectively.

For any other possible pair of states the two interacting nodes can assume, their states at the next round remain unchanged[18].

In [BTV09], Bénézit et al provide a first important property about this dynamics.

Theorem 26 (Interval-Consensus dynamics I [BTV09]). *Let $G(V, E)$ be any connected graph and assume each node starts with a color (state) in $\{0, 1\}$ such that there is some bias $s \geqslant 1$ towards color 0. Then with probability $1 - o(1)$, the* Interval-Consensus *dynamics converges to a regime of configurations where every node is either in state 0 or in state 0.5^-. A symmetric claim holds if the initial bias is toward color 1.*

More recently, Draief et al [DV12] analyzed its convergence time over general connected graphs and derive an upper bound which depends on the initial bias and the spectral properties of some matrices associated to the Markov chain induced by the process. The statement of this general result requires to introduce a set of notions which are far-beyond the scope of this survey, so we invite the interested reader to look at the original paper [DV12]. We instead here report the obtained bound for complete graphs.

Theorem 27 (Interval-Consensus dynamics II [DV12]). *Let us assume each node of the n-node complete graph starts with a state in $\{0, 1\}$ such that there is some bias $s \geqslant 1$ towards one of the two colors. Then, the expected convergence time of the* Interval-Consensus *dynamics is bounded by*

$$\frac{\log n}{2\alpha - 1}(1 + o(1)),$$

where $\alpha = (1/2)(1 + s/n) > 1/2$ denotes the fraction of nodes supporting the initial majority.

We observe that, on the positive side, the above results show that the Interval-Consensus dynamics somewhat computes the initial majority in any connected graph and starting from any initial bias $s > 0$. On the negative side, these results show a form of majority consensus which is not monochromatic and it is not guaranteed with high probability. Finally, the analysis leading to the two theorems above strongly rely on the fact that any fixed node, at any round, can interact with at most one of its neighbors. This makes any attempt to adapt this analysis for parallel synchronous interaction models (such as the $\mathcal{PULL}$ one) an interesting open issue. We conclude by observing that the Interval-Consensus dynamics has been independently re-discovered and analyzed also in [MNRS14].

6.2 Memory lower bounds for Majority and k-Plurality Consensus

As we mentioned in the previous section, [MNRS16] proved that no 3-state protocol can solve the majority consensus problem *exactly*. In this section we discuss further related work concerned with *exact* solutions the problem[19].

[18] For a possible instance of this process over a simple graph of 4 nodes, the reader may look at [BTV09].

[19] Even though some of the results mentioned in the following hold for the more general *fair scheduler*, here by "exactly" we mean with probability 1 under the random scheduler.

With respect to the Majority Consensus Problem, a recent line of research such as [AGV15, AAE+17, AAG18], investigates trade-offs between the convergence time and number of states which are necessary in order to solve the problem. Interestingly, the *Average&Conquer* they consider is a discretized version of the averaging process that we discuss in Section 7; their discretization of the process into a set of quantized levels is performed in order to reduce the number of states employed by the algorithm.

As for the more general k-Plurality Consensus Problem in the Population Protocol Model, following up their work on the Interval-Consensus dynamics, in [BTV11] Benezit et al. investigate the k-Plurality Consensus Problem (which they call *Multiple Voting Problem* with k colors). They proposed a 15-state protocol for the 3-color case and a 100-state protocol for the 4-color case, and left open the problem of determining the minimum number of states to solve (exactly) the general k-Plurality Consensus Problem.

In [SSNG15], the authors propose a $\mathcal{O}(k2^k)$ Plurality Consensus protocol and conjecture that such number of states is optimal. The conjecture was later disproved in [RN19], where a $\mathcal{O}(k^{11})$-state exact Plurality Consensus protocol is provided, together with a $\Omega(k^2)$-state lower bound for the problem.

6.3 The parallel synchronous setting

If we run the Undecided-State dynamics on the parallel $\mathcal{PULL}$ model, then at every round all nodes update their own state. Even though it is easy to verify that the dynamics still achieves consensus (with high probability), the evolution of the parallel process significantly departs from the sequential one described in the previous subsection.

To get just one immediate evidence of this difference, observe that, in the former model, the system can converge to the (non-valid) configuration where all nodes are undecided even if starting from a "fully-colored" configuration[20] (where all nodes are not undecided). As we observed in the previous section, it is easy to see that this evolution cannot happen in the sequential setting.

A deeper, crucial difference lies in the random number of nodes that may change color at every round: In the sequential model, this is at most one[21], while in the parallel one, *all* nodes may change state in one shot and indeed, for most phases of the process, the expected number of changes is linear in n. This crucial difference in the evolution of the two processes above is one of the main reasons why no general techniques are currently available which allow to extend any quantitative analysis for the sequential process to the corresponding parallel one (and viceversa): The latter turned out to be a non-reversible Markov chain having a state-transition di-graph of very large degree.

The analysis in [AAE08] strongly uses the fact that only one node can change state in one state to define a suitable supermartingale argument to obtain a bound on the stopping time of the process. So, the analysis of [AAE08] fully covers the case of sequential interaction models, but it is not helpful to understand the evolution of the Undecided-State dynamics process on any interaction model in which the number of nodes that may change state in one round is not bounded by some absolute constant.

Recently, an unconditional analysis of the Undecided-State dynamics on the parallel $\mathcal{PULL}$ model has been presented by Clementi et Al in [CGPS17, CGG+17] and its main consequences are stated below.

[20] This happens if every colored node pulls the other color.

[21] This number becomes 2 if the sequential communication model activate a random edge per round, rather than one single node [AAE08].

Theorem 28 (Parallel UNDECIDED-STATE Dynamics [CGPS17, CGG+17]). *Let the* UNDECIDED-STATE *dynamics start from any configuration* $\mathbf{c} \in \mathcal{C}$. *Then, the following claims hold.*

- Convergence Time. *The process converges to a (valid) monochromatic configuration within* $\mathcal{O}(\log n)$ *rounds, w.h.p. Furthermore, if the initial configuration has at least one colored node (i.e.* $q \leqslant n-1$*), then the process converges to a valid monochromatic configuration* $\bar{\mathbf{c}} \neq (0,0,n)$*, w.h.p.*

- Plurality-Preserving Property. *If the initial configuration* $\mathbf{c}$ *has bias* $s = \Omega(\sqrt{n \log n})$*, then the process converges to the plurality color, w.h.p. Furthermore, the result is almost tight in a twofold sense: (i) An initial configuration exists, with* $s = \Omega(\sqrt{n \log n})$*, such that the process requires* $\Omega(\log n)$ *rounds to converge w.h.p., and (ii) there is an initial configuration with* $s = \Theta(\sqrt{n})$ *such that the process converges to the minority color with constant probability.*

The major technical issues arise from the analysis of *balanced* initial configurations where the system "needs" to *break symmetry* without having a strong expected drift towards any color. Essentially, previous analysis of this phase consider either *sequential* processes of interacting particles that can be modeled as *birth-and-death* chains (such as that discussed in the previous section for the sequential version of the UNDECIDED-STATE dynamics [AAE08]) or as parallel processes whose local rule is fully symmetric w.r.t. the states/colors of the nodes (such as majority rules)[BCN+16, DGM+11]. The parallel UNDECIDED-STATE dynamics process falls neither in the former nor in the latter scenario: it works in parallel rounds and the role of the undecided nodes makes the local rule not symmetric. Informally speaking, Clementi et al show an "efficient" way to reduce all "critical" almost-balanced starting of the process to a specific regime along which the system keeps a number q of undecided nodes which is some suitable constant fraction of n *until* the bias s has reached an $\Omega(\sqrt{n \log n})$ magnitude: in other words, during this regime, with very high probability the system never jumps to almost-balanced configurations having either too many or too few undecided nodes. This fact is crucial essentially because of two reasons: along this regime, (i) the *variance* of the bias s is large (i.e. $\Theta(n)$) and (ii) whenever the bias s gets $\Omega(\sqrt{n})$, its drift turns out to be *exponential* with non-negligible, increasing probability (w.r.t. s itself). Then, by devising a coupling to a "simplified" pruning process, the analysis in [CGPS17, CGG+17] makes use of (a suitable version) of a general lemma [DGM+11] (see Claim 9.2 in [DGM+11]) that provides a logarithmic bound on the hitting time of some Markov chains that have Properties (i) and (ii) above.

The symmetry-breaking phase terminates when the UNDECIDED-STATE dynamics reaches some configuration having a bias $s = \Omega(\sqrt{n \log n})$. Then it is proved that, starting from *any* configuration having that bias, the process reaches consensus within $\mathcal{O}(\log n)$ rounds, w.h.p. This part still must cope with some *non-monotone* behaviour of the key random variables (such as the bias and the number of undecided nodes at the next round): again, this is due to the non-symmetric role played by the undecided nodes. The refined analysis shows that, during this majority phase, the winning color never changes and, thus, the UNDECIDED-STATE dynamics also ensures Plurality Consensus in logarithmic time whenever the initial bias is $s = \Omega(\sqrt{n \log n})$.

The analysis in [CGPS17, CGG+17] does not consider any kind of faults, so the fault-tolerance of this dynamics in the parallel model is still an open issue. However, we believe that the protocol can tolerate any adaptive, dynamic adversary that corrupts at most $o(\sqrt{n})$ nodes at every round.

6.4 Dealing with more colors

The results given in the previous subsections on the UNDECIDED-STATE dynamics concern the binary case. When the colors a node can support is larger, then the process become significantly more complex and, up to now, there are no unconditional analysis available for this general setting.

In [BCN+16], Becchetti et Al provide an analysis of the UNDECIDED-STATE dynamics for the multi-color case on the complete graph by assuming a certain initial bias toward a plurality color. This subsection gives a short overview of this work.

As described in the previous subsections, the performance of this dynamics on the complete graph has been evaluated w.r.t. the following parameters: the number n of nodes and the initial *bias* towards the plurality color, with the latter characterized in terms of a parameter that only depends on the relative magnitude[22] of $\bar{c}_1$ and $\bar{c}_2$.

However, when $k > 2$, any such measure of the initial bias is not sensitive enough to accurately capture the convergence time of a plurality protocol: a *global* measure is needed, i.e., one that reflects the whole initial color configuration. To better appreciate this issue, consider the two configurations $\bar{\mathbf{c}}$ and $\bar{\mathbf{c}}'$ in Fig. 3. Whether the absolute difference or the relative ratio is used to measure the initial bias, the color configuration $\bar{\mathbf{c}}'$ appears to be not "worse" than $\bar{\mathbf{c}}$. Still, computer simulations and intuitive arguments suggest that, under any "natural" plurality protocol, the almost-uniform color distribution $\bar{\mathbf{c}}'$ can result in much larger convergence times than the highly-concentrated color configuration $\bar{\mathbf{c}}$. We remark that the impact of the whole initial color configuration on the speed of convergence of plurality protocols has never been analyzed before.

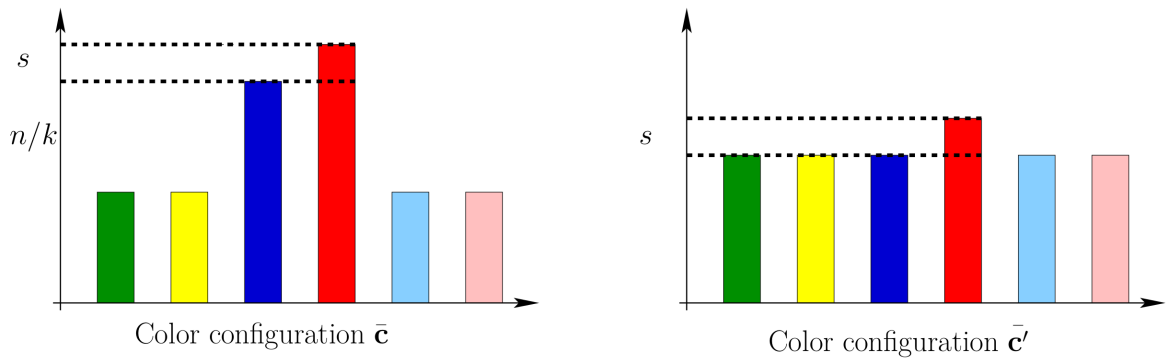

Figure 3: Two different color configurations having the same bias $s = s(c_1, c_2)$

As a first step, Becchetti et al introduce a suitable *distance* $\mathsf{d}(\cdot,\cdot)$ on the set $\mathcal{C}$ of all color configurations. It naturally induces a function $\mathsf{md}(\cdot)$, called the *monochromatic distance*, which equals the *distance* between any configuration $\mathbf{c}$ and the target configuration:

$$\mathsf{md}(\mathbf{c}) \;=\; \sum_{i=1}^{k} \left(\frac{c_i}{c_1}\right)^2$$

They use md to characterize the bias of the initial configuration. In particular, note that $\mathsf{md}(\bar{\mathbf{c}})$ measures the extent to which $\bar{\mathbf{c}}$ is "uniform": Indeed, the higher the extent of the bias towards a small subset of the colors (including the plurality one), the smaller the value of $\mathsf{md}(\bar{\mathbf{c}})$.

[22] Typically, this relative magnitude is defined in terms of the absolute difference or the ratio.

As an example, in Fig. 3, $\mathsf{md}(\bar{\mathbf{c}})$ can be substantially smaller than $\mathsf{md}(\bar{\mathbf{c}}')$. At the extremes, when there are only $O(1)$ color communities of size $\Theta(\bar{c}_1)$, we have $\mathsf{md}(\bar{\mathbf{c}}) = \Theta(1)$ while, when $\Theta(k)$ color communities have size $\Theta(n/k)$, we have $\mathsf{md}(\bar{\mathbf{c}}) = \Theta(k)$.

Notice that, in the restricted, binary case, the complex dependence of the dynamics' evolution on the overall shape of the initial color configuration is not exhibited.

The analysis introduced by Becchetti et al in [BCN+16] allows to establish the following relationship between the convergence time and the monochromatic distance.

Theorem 29 ([BCN+16]). *let $k = k(n)$ be any function such that $k = O((n/\log n)^{1/3})$, and consider any initial configuration $\bar{\mathbf{c}} \in \mathcal{S}$ such that $\bar{c}_1 \geqslant (1+\alpha)\bar{c}_2$ where $\alpha > 0$ is* any arbitrarily-small *constant*[23]*. Then, the following claims hold.*

- Convergence towards plurality. *The* Undecided-State *dynamics converges within $O(\mathsf{md}(\bar{\mathbf{c}}) \log n)$ rounds towards the plurality color, w.h.p.*
- Tightness. *This result is almost-tight in a strong sense. In particular, for $k = O((n/\log n)^{1/6})$ and* for any *initial k-colors configurations $\bar{\mathbf{c}}$,* the convergence time of the Undecided-State dynamics is linear in the monochromatic distance $\mathsf{md}(\bar{\mathbf{c}})$ w.h.p.

Comparison with the 3-Majority dynamics. We emphasize that, when k is some root of n, the above result implies that the Undecided-State dynamics is exponentially faster than the best protocol that uses polylogarithmic bounded memory [BCN+17a] on a large class of initial color configurations. In particular, as described in Section 5, in [BCN+17a] the authors analyze the 3-Majority dynamics and, when the initial bias is $s = \Omega(\sqrt{kn \log n})$, we have seen that the 3-Majority dynamics converges within $\Theta(\min\{k, n^{1/3}\} \log n)$ rounds using $\Theta(\log k)$ memory and message size. Notice that the convergence times of the 3-Majority dynamics become polylogarithmic only if $\bar{c}_1 \geqslant n/\text{polylog}\,(n)$, thus they are not polylogarithmic whenever $k = \omega(\text{polylog}\,(n))$ and $\bar{c}_1 = o(n/\text{polylog}\,(n))$. This is the parameter range where Theorem 29 shows that the Undecided-State dynamics leads to an exponential speed up w.r.t. the convergence time of the 3-Majority dynamics. For example, consider an initial "oligarchic" scenario where $k = n^{1/4}$ and a subset $\mathcal{L} = \{1, 2, ..., \text{polylog}\,(n)\}$, such that for any $i \in \mathcal{L}$, $\bar{c}_i \sim n/\sqrt{k}$, and for any $i \in [k] \setminus \mathcal{L}$, $\bar{c}_i \sim n/k$. Clearly, $1, 2 \in \mathcal{L}$ and the resulting monochromatic distance is $\mathsf{md}(\bar{\mathbf{c}}) = \text{polylog}\,(n)$. Assuming $\bar{c}_1 \geqslant (1+\alpha)\bar{c}_2$ for some $\alpha > 0$ our upper bound implies that, starting from any such configuration, the Undecided-State dynamics converges in polylogarithmic time, whereas the 3-Majority dynamics converges in $\Theta(k \log n)$ time [BCN+17a].

On the other hand, the upper bound on the convergence time for the 3-Majority dynamics (see Theorem 18) does not require the condition on the multiplicative bias in Theorem 29, Moreover, we note that the Undecided-State dynamics may fail to reach plurality consensus when $k = \omega(\sqrt{n})$. Indeed, for such parameter range there are initial configurations in which the Undecided-State dynamics makes the plurality color disappear after one round with constant probability (see, e.g., Section 3 in [BCN+15]).

7 Averaging dynamics

In this section, we discuss the interesting and at first sight surprising properties of a family of elementary network processes. At a high level, each node of an underlying graph G initially holds a possibly random value. Then, in each consecutive step, every node updates to a (possibly

[23] in the multi-color case this is a weak-bias condition that ensures the convergence of the process towards the plurality color.

weighted) average of its current value and of the current values held by one or more of its neighbours. For the scenarios of interest in this section, nodes follow a very simple random rule to initialize their values. We remark that different schemes are possible, or initialization might be the outcome of an exogenous process.

Averaging may come in different flavours, depending on the underlying communication model. In Subsection 7.1, we describe two specific algorithms that fulfill the general idea given above, respectively in the $\mathcal{LOCAL}$ and in the random sequential communication models. We always use the phrase AVERAGING dynamics in the remainder of this section, whereas the specific algorithm we are referring to each time should be clear from context.

7.1 Averaging dynamics

In the $\mathcal{LOCAL}$ communication model, the averaging dynamics can be defined as follows.

Definition 30 (AVERAGING dynamics on $\mathcal{LOCAL}$). *(Initialization). At round $t = 0$ every node $v \in V$ independently samples its value from $\{-1, +1\}$ uniformly at random;*

- *(Updating rule). In each subsequent round $t \geqslant 1$, every node $u \in V$ updates its value x_u to the average of the values of its neighbors at the end of the previous round.*

In the random sequential case, a node performs some action whenever it is one of the endpoints of the edge selected for communication in the current step.

Definition 31 (AVERAGING(δ) on the random sequential model). *Let $\delta \in (0, 1)$ be the parameter measuring the weight given to the neighbor's value.*

- *(Initialization). If it is the first time u is active, then pick $x_u \in \{-1, +1\}$ u.a.r.*
- *(Updating rule). Send x_u to the other endpoint of the active edge and then update $x_u := (1 - \delta)x_u + \delta r$, where r is the value received from the other endpoint.*

Notation and preliminaries We use the following notation in the remainder of this section. Whatever the communication model, we denote by $\mathbf{x}^{(t)}$ the global state of the AVERAGING dynamics at time t, i.e., the vector of values held by the nodes at time t. In particular, we let $\mathbf{x}^{(0)} = \mathbf{x}$ for simplicity. Unless otherwise specified, the norm of a vector $\mathbf{x}$ is its ℓ_2 norm $\|\mathbf{x}\| := \sqrt{\sum_i (\mathbf{x}(i))^2}$ and the norm of a matrix M is its spectral norm $\|M\| := \sup_{\mathbf{x}:\|\mathbf{x}\|=1} \|M\mathbf{x}\|$. For a diagonal matrix, this is the largest diagonal entry in absolute value. Finally, given an $n \times n$ matrix M, we denote by $\lambda_1(M) \geqslant \ldots \geqslant \lambda_n(M)$ its eigenvalues. For the rest of this section, we refer to an undirected graph $G = (V, E)$ with n nodes. We denote by $d(v)$ v's degree and we let $m = \sum_v d(v)$.

We next discuss the behaviour of the AVERAGING dynamics in both $\mathcal{LOCAL}$ and random sequential models. One of the most interesting aspects is the versatility and algorithmic potential of the resulting process when considered at different time scales.

7.2 AVERAGING dynamics, random walks and Consensus

In this section, we discuss relationships between the AVERAGING dynamics and random walks on undirected graphs. We further discuss their behaviour with respect to consensus. Finally, we highlight connections between convergence of the AVERAGING dynamics and spectral properties of the underlying topology, which in turn are crucial for clustering.

Random walks and Consensus. In the $\mathcal{LOCAL}$ model, the AVERAGING dynamics is perhaps one of the simplest, yet most interesting examples of a linear dynamics. In the first place, it always converges when the underlying graph G is connected and not bipartite. In more detail, it converges to the global average of the initial values if the graph is regular and to a weighted global average if it isn't, the weight of each node/value being the corresponding entry of the main left eigenvector of G's transition matrix [BGPS06, Sha09]. Seeing this is relatively simple if we consider that a single step of the AVERAGING dynamics amounts to applying the transition matrix P of G to the current global state.[24] P operates as the random walk process on G by left multiplication, and as the AVERAGING dynamics by right multiplication. So, the global state of the system after t steps of the AVERAGING dynamics is

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

If G is undirected and non-bipartite very interesting properties hold: i) the Markov chain described by P is ergodic; ii) as a consequence, as $t \to \infty$, P^t converges to a rank one matrix, whose rows are copies of P's main left eigenvector; iii) the mixing time of this Markov chain crucially depends on $\lambda_2(P)$.[25] In more detail, the following results holds:

Theorem 32 (Convergence, see Section 1 of [L+93]). *The random walk on G is ergodic, whenever G is not bipartite. Moreover, the corresponding stationary distribution is given by the vector $\boldsymbol{\pi}$ defined as:*

$$\boldsymbol{\pi}(v) = \frac{d(v)}{2m}, \forall v \in V.$$

While proving ergodicity requires some (small) extra work, one can immediately check that $\boldsymbol{\pi}$ above is a stationary distribution for P, i.e., it satisfies $\boldsymbol{\pi}^T P = \boldsymbol{\pi}^T$, which in turn implies that $\boldsymbol{\pi}$ is a left eigenvector of P, with $\lambda_1(P) = 1$ as the associated eigenvalue. On the other hand, the vector $\mathbf{1} = (1, \ldots, 1)^T$ is a trivial, right eigenvector of P associated to $\lambda_1(P) = 1$. These two facts imply that, whenever G is not bipartite, P^T converges to $\boldsymbol{\pi}^T\mathbf{1}$ as $t \to \infty$.[26] As a result, for every $\mathbf{x} \in \mathbb{R}^n$, $P^t\mathbf{x}$ converges to $\alpha\mathbf{1}$, where $\alpha = \sum_v \boldsymbol{\pi}(v)\mathbf{x}(v)$. Hence, the AVERAGING dynamics achieves consensus on the (generally weighted) average of the values they initially hold.[27]

As for the speed of convergence, this crucially depends on the spectrum of P. Note that, for every t, P^t_{uv} is the probability that a random walk started at u is at node v after t steps. The following result hold:

Theorem 33 (Convergence time, see Theorem 5.1 of [L+93]). *Let $\lambda = \max\{|\lambda_2(P)|, |\lambda_n(P)|\}$. For every $u, v \in V$ and for every $t \geqslant 0$ we have:*

$$|P^t_{uv} - \boldsymbol{\pi}(v)| \leqslant \sqrt{\frac{d(v)}{d(v)}}\lambda^t.$$

Unless G is bipartite, $\lambda < 1$. In particular, this implies exponential convergence speed, whenever λ is bounded away from 1 by a (possibly small) constant.

[24] Recall that, if $G = (V, E)$ is an undirected graph (possibly with multiple edges and self loops), A its adjacency matrix and d_i the degree of node i, the *transition matrix* of (the random walk on) G is the matrix $P = D^{-1}A$, where D is the diagonal matrix such that $D_{i,i} = d_i$. $P_{i,j} = (1/d_i) \cdot A_{i,j}$ is thus the probability of going from i to j in one-step of the random walk on G.

[25] The results we are about to state almost seamlessly extend to weighted graphs.

[26] The algebraic argument is relatively easy, but beyond the scope of this survey.

[27] We stress that this is not a valid consensus in the sense considered elsewhere in this survey, since the final, common value in general is not among those initially held by the nodes, i.e., it not one of the components of $\mathbf{x}$.

The picture changes completely in process over the random sequential model described in Definition 30. Here, the global state $\mathbf{x}^{(t)}$ at time t depends on the global state at time $t-1$ according to a non-homogeneous Markov chain. In more detail, we have:

$$\mathbf{x}^{(t)} = W(t)\mathbf{x}^{(t-1)},$$

where $W(t)$ is a matrix that corresponds to the averaging between the end nodes of the edge that is selected for communication in the t-th step. In more detail,[28] it is easy to see [BGPS06] that, if edge (i,j) is selected for communication in the t-step we have:

$$W(t) = I - \frac{(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T}{2}, \tag{2}$$

where I is the identiy matrix and $\mathbf{e}_h$ denotes the h-th canonical vector. As a result we have:

$$\mathbf{x}^{(t)} = W(t)W(t-1)\cdots W(1)\mathbf{x}^{(0)}, \tag{3}$$

where each $W(i)$ has the form (2) and corresponds to the edge sampled in the i-th step. We let $W^{(t)} = W(t)W(t-1)\cdots W(1)$ in the remainder.

A few remarks are in order: i) the $W(t)$'s are independent and identically distributed random variables,[29] their distribution corresponding to the one over the edges that defines the random sequential process; ii) each $W(t)$ is a doubly stochastic matrix; iii) it is easy to see that $W^{(t)}$ is a (doubly) stochastic matrix that defines a non-homogeneous Markov chain.

The expectation $\mathbf{E}\left[W(t)\right] = W$ of course does not depend on t, while it depends on G's transition matrix in a simple way, namely:

$$W = I - \frac{1}{2n}\bar{D} + \frac{P + P^T}{2n}. \tag{4}$$

Here, $\bar{D}$ is the diagonal matrix such that $\bar{D}_{ii} = \sum_{j=1}^{n}(P_{ij} + P_{ji})$. If G is a regular graph, the above expressione simplifies to

$$W = \left(1 - \frac{1}{n}\right) I + \frac{1}{n}P. \tag{5}$$

One first question is whether the stochastic process defined by $\mathbf{x}^{(t)}$ converges as $t \to \infty$. Loosely speaking, in [BGPS06] the authors prove that the process defined by the Averaging dynamics converges to the constant vector $\bar{x}\mathbf{1}$, where $\bar{x} = \sum_u \mathbf{x}_u^{(0)}/n$, i.e., to the vector that assigns the average of the initial values to each node of the graph. More formally, let $\bar{T}(\varepsilon)$ be defined as:

$$sup_{\mathbf{x}^{(0)}} inf \left\{ t : \mathbf{P}\left(\frac{\|\mathbf{x}^{(t)} - \bar{x}\mathbf{1}\|}{\|\mathbf{x}^{(0)}\|} \geqslant \varepsilon \right) \leqslant \varepsilon \right\}.$$

Then the following result holds:

Theorem 34 (Convergence time in the random sequential model [BGPS06])**.**

$$\bar{T}(\varepsilon) \leqslant \frac{3\log\varepsilon^{-1}}{\log\lambda_2(W)^{-1}},$$
$$\bar{T}(\varepsilon) \geqslant \frac{0.5\log\varepsilon^{-1}}{\log\lambda_2(W)^{-1}}.$$

[28] We remind the reader that we focus on the case $\delta = 1/2$.

[29] If edges are selected independently across different time steps as we assume.

Second eigenvalue and graph topology. Theorem 34 entails an implicit dependance of convergence on the underlying graph topology, which we briefly discuss below. For the sake of simplicity, we consider regular graphs. In this case we have from (5):

$$\lambda_2(W) = 1 - \frac{1}{n} + \frac{\lambda_2(P)}{n} = 1 - \frac{1 - \lambda_2(P)}{n}.$$

From this, as a corollary of Theorem 34, simple manipulations allow us to conclude that

$$\bar{T}(\varepsilon) = \Theta\left(\frac{n}{1 - \lambda_2(P)} \log \frac{1}{\varepsilon}\right).$$

On the other hand, $\lambda_2(P)$ is related to the expansion properties of G: loosely speaking, a value of $\lambda_2(P)$ close to 1 denotes the presence of (at least) one sparse cut with low conductance. This dependence is quantitatively characterized by the well-known Cheeger's inequality and can be generalized to provide a characterization of the cluster structure of a graph [LGT14]. Keeping this in mind, the above expression allows to quantitatively assess the intuition that convergence of the Averaging dynamics to the global average in the random sequential model is negatively affected by the clustering properties of the underlying graph G, in particular the presence of a sparse cut in G. This intuition is at the core of the section that follows.

7.3 Clustering properties of the Averaging dynamics

The remarks at the end of the previous section suggest that, whatever the model of communication, the evolution of the process resulting from the the Averaging dynamics might provide information about the structure of the underlying graph. To be more specific, suppose that G exhibits a *community structure*, which in the simplest case consists of two equal-sized expanders (the two "communities" of the graph) connected by a sparse cut. Assume further that we run our averaging process starting, for example, from an initial ± 1 random global state. One might reasonably expect the following high-level pattern in the evolution of the Averaging dynamics: i) a faster convergence toward a local average within each community; ii) a slower convergence toward the global average over the entire graph; iii) a transient phase in which the values held by nodes within each community are close to the local average, whereas local averages of the two communities will in general exhibit a certain gap that depends on the initial values.

If this were the case, the global state during the transient phase would be correlated with the indicator of the cut between the two communities. This intuition suggests a main question, which we present below as it is formulated in [BCN+17c, BCM+18]:

Is there *a phase in which the global state carries information about community structure*? If so, *how strong is the corresponding "signal"*? Finally, *can nodes leverage local history to uncover the underlying community structure*?

In the remainder, we analyze the clustering properties of the Averaging dynamics with respect to an important benchmark in the theory of clustering algorithms, namely, the *block reconstruction problem*. Let $G = ((V_1, V_2), E)$ be a graph with $V_1 \cap V_2 = \emptyset$. A weak (block) reconstruction is a two-coloring of the nodes that separates V_1 and V_2 up to a small fraction of the nodes. Formally, we define an *ε-weak reconstruction* as a map $f : V_1 \cup V_2 \to \{\texttt{red}, \texttt{blue}\}$ such that there are two subsets $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ with $|W_1 \cup W_2| \geqslant (1 - \varepsilon)|V_1 \cup V_2|$ and $f(W_1) \cap f(W_2) = \emptyset$. When $\varepsilon = 0$ we say that f is a *strong reconstruction*. Given a graph $G = ((V_1, V_2), E)$, the block reconstruction problem requires computing an ε-reconstruction of G. The underlying, hidden partition we intend to uncover can be described by an indicator vector. In more detail, for $i = 1, 2$, define $\mathbf{1}_{V_i}$, as the $|V|$-dimensional vector, whose j-th component is

1 if $j \in V_i$, it is 0 otherwise. If (V_1, V_2) is a bipartition of the nodes with $|V_1| = |V_2| = n$, we define the *partition indicator vector* $\boldsymbol{\chi} = \mathbf{1}_{V_1} - \mathbf{1}_{V_2}$.

In general, the reconstruction problem has been studied extensively using a multiplicity of techniques, which include combinatorial algorithms [DF89], belief propagation [DKMZ11] and variants of it [MNS14], spectral-based techniques [McS01, CO10], Metropolis approaches [JS98], and semidefinite programming [ABH14], among others. The contributions we discuss below show that the transient behaviour of an elementary and fully decentralized process as the Averaging dynamics carries information about the cluster structure of G, which in important cases can be effectively recovered.

7.3.1 Clustering properties in the $\mathcal{LOCAL}$ model

In [BCN+17b], the authors propose the following distributed protocol for block reconstruction. It is based on the Averaging dynamics and produces a coloring of the nodes at the end of every round. The authors show that, within $\mathcal{O}(\log n)$ rounds, the coloring computed by the algorithm achieves *strong reconstruction* of the two blocks in the case of clustered regular graphs and *weak reconstruction* for a large family of clustered, non-regular graphs that with high probability includes those generated from the stochastic block model [MNS14].

Averaging protocol:

Rademacher initialization: At round $t = 0$ every node $v \in V$ independently samples its value from $\{-1, +1\}$ uniformly at random;

Updating rule: At each subsequent round $t \geqslant 1$, every node $v \in V$

1. (Averaging dynamics) Updates its value $\mathbf{x}^{(t)}(v)$ to the average of the values of its neighbors at the end of the previous round,
2. (Coloring) If $\mathbf{x}^{(t)}(v) \geqslant \mathbf{x}^{(t-1)}(v)$ then v sets $\texttt{color}^{(t)}(v) = \texttt{blue}$ otherwise v sets $\texttt{color}^{(t)}(v) = \texttt{red}$.

Algorithm 1: Overall clustering algorithm for a node u in the $\mathcal{LOCAL}$ model of communication.

The choice of the above coloring rule will be motivated in the paragraphs that follow. Here, we note that the algorithm is a dynamics in the strictest sense of Definition 1. In particular, it is completely oblivious to time. Namely, after initialization the protocol iterates over and over at every node. The resulting coloring eventually stabilizes to a (possibly weak) reconstruction. This is a property of the protocol, of which nodes are not aware, it is an event that eventually occurs with high probability when certain assumptions hold (Theorems 36 and [BCN+17b, Theorems 4.1 and 4.2]). Moreover, the clustering criterion is completely *local*, in the sense that a decision is individually and independently made by each node in each round, only on the basis of its state in the current and previous rounds. This may seem counter-intuitive at first, but closer scrutiny shows that the clustering criterion implicitly leverages spectral properties of P to uncover the underlying community structure, as we outline in the paragraphs that follow.

Remark. We briefly note that, although Algorithm 1 is well-defined regardless of graph size and, as such, it meets the standards of Definition 1, the minimum amount of local memory required at each node for the results of Theorems 36 and [BCN+17b, Theorems 4.1 and 4.2] to hold might indeed depend on properties of the graph, in particular its maximum degree. The reason is in the averaging operation itself, which is a relatively complex primitive and requires each node to receive (and, therefore, temporarily store) the values held by its neighbours at the end of the previous round.

AVERAGING **dynamics and clustering.** Recall that, from Section 7.2, the behavior of the AVERAGING dynamics on a graph G is closely related to the behavior of random walks in G. Namely, if $\mathbf{x}$ is the initial vector of values, after t rounds of the AVERAGING dynamics the vector of values at time t is $\mathbf{x}^{(t)} = P^t\mathbf{x}$. The product of the power of a matrix by a vector is best understood in terms of the spectrum of the matrix, which is what we explore in the next section. In the remainder, we denote by λ the largest, in absolute value, among all but the first two eigenvalues of P, namely, $\lambda = \max\{|\lambda_i(P)| : i = 3, 4, \ldots, n\}$.[30]

Strong reconstruction for regular graphs. The simple, yet non-obvious intuition behind the AVERAGING protocol and its analysis is best captured by the case of regular graphs. Specifically, we consider the following, "well" clustered family or regular graphs.

Definition 35 (Clustered Regular Graph)**.** *A (n, d, b)-clustered regular graph $G = ((V_1, V_2), E)$ is a graph over vertex set $V_1 \cup V_2$, with $|V_1| = |V_2| = n/2$ and such that:*[31] *(i) Every node has degree d and (ii) Every node in cluster V_1 has b neighbors in cluster V_2 and every node in V_2 has b neighbors in V_1.*

The next result shows that, for this graph class, the AVERAGING protocol achieves strong reconstruction with high probability.

Theorem 36 (Strong reconstruction [BCN+17b])**.** *Let $G = ((V_1, V_2), E)$ be a connected $(2n, d, b)$-clustered regular graph with $1 - 2b/d > (1 + \delta)\lambda$ for an arbitrarily-small constant $\delta > 0$. Then the* AVERAGING *protocol produces a strong reconstruction within $\mathcal{O}(\log n)$ rounds, w.h.p.*

The intuition behind the above result is best described by an outline of the main arguments used in the proof of Theorem 36. When G is d-regular, $P = (1/d)A$ is a real symmetric matrix and P and A have the same set of eigenvectors. We know that $\mathbf{1}$ is an eigenvector of P with eigenvalue 1, and it is easy to see that the partition indicator vector $\boldsymbol{\chi}$ is an eigenvector of P with eigenvalue $1 - 2b/d$. We denote by $\mathbf{v}_1 = (1/\sqrt{2n})\mathbf{1}, \mathbf{v}_2, \ldots, \mathbf{v}_{2n}$ a basis of orthonormal eigenvectors, where each $\mathbf{v}_i$ is the eigenvector associated to eigenvalue $\lambda_i(P)$. Then, we can write a vector $\mathbf{x}$ as a linear combination $\mathbf{x} = \sum_i \alpha_i \mathbf{v}_i$ and we have:

$$P^t\mathbf{x} = \sum_i \lambda_i^t \alpha_i \mathbf{v}_i = \frac{1}{2n}\left(\sum_i \mathbf{x}(i)\right)\mathbf{1} + \sum_{i=2}^{2n} \lambda_i^t \alpha_i \mathbf{v}_i, \tag{6}$$

which implies that $\mathbf{x}^{(t)} = P^t\mathbf{x}$ tends to $\alpha_1\mathbf{v}_1$ as t tends to infinity, i.e., it converges to the vector that has the average of $\mathbf{x}$ in every coordinate. If $1 - 2b/d$ happens to be the second eigenvalue, after t rounds of the AVERAGING dynamics, the configuration $\mathbf{x}^{(t)}$ is close to a linear combination of $\mathbf{1}$ and $\boldsymbol{\chi}$. Formally, if $\lambda < 1 - 2b/d$, it is possible to prove that there are reals α_1, α_2 such that for every t:

$$\mathbf{x}^{(t)} = \alpha_1\mathbf{1} + \alpha_2(\lambda_2(P))^t\boldsymbol{\chi} + \mathbf{e}^{(t)}, \tag{7}$$

where $\left\|\mathbf{e}^{(t)}\right\|_\infty \leqslant \lambda^t\sqrt{2n}$.

The equation above naturally suggests a coloring rule in the AVERAGING protocol, as soon as one considers the difference between consecutive values at any node u, i.e.,

$$\mathbf{x}^{(t-1)}(u) - \mathbf{x}^{(t)}(u) \tag{8}$$

[30] Recall that, since P is a stochastic matrix we have $\lambda_1(P) = 1$ and $\lambda_n(P) \geqslant -1$, moreover for all graphs that are connected and not bipartite it holds that $\lambda_2(P) < 1$ and $\lambda_n(P) > -1$.

[31] We assume n even for simplicity.

$$= \alpha_2 \lambda_2^{t-1}(1-\lambda_2)\boldsymbol{\chi}(u) + \mathbf{e}^{(t-1)}(u) - \mathbf{e}^{(t)}(u)\,. \tag{9}$$

If λ is sufficiently small, $\left\|\mathbf{e}^{(t)}\right\|_\infty$ eventually becomes negligible in (7), so that, after a short initial phase, the sign of $\mathbf{x}^{(t-1)}(u) - \mathbf{x}^{(t)}(u)$ is essentially determined by $\boldsymbol{\chi}(u)$, thus by the community u belongs to. This happens w.h.p. over the randomness of $\mathbf{x}^{(0)} = \mathbf{x}$.

Weak reconstruction for non-regular graphs. If G is non-regular, $P = D^{-1}A$ is not symmetric in general and the analysis based on (7) has to be significantly revisited.[32]

On the other hand, one might argue that, if G is not too "distant" from an (n, d, b)-clustered regular graph, a weaker version of Theorem 36 might still hold. To quantitatively formalize this intuition, [BCN+17b] introduces a relaxed notion of (n, d, b)-clustered regular graph. For this graph family, which with high probability includes realizations of the stochastic block model, [BCN+17b] proves that, under reasonable assumptions, the Averaging dynamics achieves weak reconstruction within $\mathcal{O}(\log n)$ rounds w.h.p. The authors further provide a tighter analysis for the specific case of the stochastic block model. We refer the reader to Section 4 and in particular Theorems 4.1 and 4.2 of [BCN+17b] for formal statements of these results.

7.3.2 Clustering properties in the random sequential model.

The clustering properties of the Averaging(δ) dynamics in the random sequential model were investigated in [BCN+17c, BCM+18] and are considerably harder to analyze. The obvious reason is that the evolution of the global state in this case follows a non-homogeneous process (3). On the other hand, intuition suggests that the process resulting from the Averaging dynamics in the random sequential model might be a "sparsified" version of the one we obtain in the $\mathcal{LOCAL}$ model, at least in expectation. This intuition turns out to be correct and we briefly show why this is the case in the paragraphs that follow. On the other hand, achieving provable performance seems considerably more challenging, for reasons that we briefly explain in the remainder.

First moment analysis. We next discuss the expected evolution of $\mathbf{x}^{(t)}$ under the action of the Averaging(δ) dynamics. For simplicity of exposition, in the remainder we assume G is regular. Again, we set $\mathbf{x}^{(0)} = \mathbf{x}$, with $\mathbf{x}$ realized according to the initialization step in Algorithm 31. When G is regular, we have

$$\mathbf{E}\left[\mathbf{x}^{(t)}\right] = W^t \mathbf{x},$$

with W given from (5). This implies that P and W share the same eigenvectors and that the i-th largest eigenvalues of P and W are related as follows:

$$\lambda_i(W) = 1 - \frac{1-\lambda_i(P)}{n}. \tag{10}$$

Moreover, if G is (n, d, b)-clustered regular, $\boldsymbol{\chi}$ happens to be an eigenvector of W. This again follows from (5), since in this case $\boldsymbol{\chi}$ is an eigenvector of P, as we remarked in the previous section. Hence, the evolution of $\mathbf{E}\left[\mathbf{x}^{(t)}\right]$ is dictated by an equation that has the same form as (7), with $\lambda_2(W)$ replacing $\lambda_2(P)$. Proceeding like in the $\mathcal{LOCAL}$ case, with some work it is possible to draw the following conclusions:

Monotonicity in expectation. If λ (defined as in the previous section) is sufficiently small, after an initial phase of length $\mathcal{O}(n \log n)$, the sign of $\mathbf{E}\left[\mathbf{x}^{(t-1)}(u) - \mathbf{x}^{(t)}(u)\right]$ does not change

[32] In a nutshell, this happens for two reasons: i) P no longer admits an orthonormal eigenvector basis; ii) while $\mathbf{1}$ is always the main right eigenvector of any stochastic matrix, $\boldsymbol{\chi}$ is not an eigenvector in general, even in regular graphs that are not (n, d, b)-clustered.

(hence monotonicity) and it is essentially determined by $\boldsymbol{\chi}(u)$, i.e., by the community u belongs to. This happens w.h.p. over the randomness of $\mathbf{x}^{(0)} = \mathbf{x}$.

Second moment analysis for (n, d, b)-clustered regular graphs. Designing heuristics for block reconstruction in the random sequential setting using the insights from the first moment analysis outlined above poses significant technical challenges. The main reason is that, despite the "nice" behaviour of $\mathbf{E}\left[\mathbf{x}^{(t)}\right]$ described above, $\|\mathbf{x}^{(t)} - \mathbf{E}\left[\mathbf{x}^{(t)}\right]\|$ might change significantly over time. On the other hand, analyzing the evolution of $\|\mathbf{x}^{(t)} - \mathbf{E}\left[\mathbf{x}^{(t)}\right]\|$ (or, equivalently, $\mathbf{E}\left[\|\mathbf{x}^{(t)} - \mathbf{E}\left[\mathbf{x}^{(t)}\right]\|^2\right]$) entails characterizing the concentration of a product of independent, identically distributed random matrices. Unfortunately, while a theory of the concentration properties of the sum of independent, identically distributed random matrices is well-established by now [Tro12], the same cannot be said for their products, for which very little is known. Achieving this turns out to be technically challenging even when G is (n, d, b)-clustered regular.

In [BCN+17c, BCM+18], the authors propose two clustering heuristics with provable guarantees in the case of (n, d, b)-clustered regular graphs, both of which rely on Algorithm 30 as a basic building block. Though Algorithm 30 is a dynamics, the proposed clustering criteria only apply within a suitable time window, which in turn depends on the underlying graph's size. For this reason, the overall resulting clustering algorithms are not dynamics in the sense of Definition 1 and we refer the interested reader to the above references for details.

Extensions and generalizations. While the first moment analysis sketched above can be extended to more general topologies than (n, d, b)-clustered regular graphs considered in [BCN+17c, BCM+18], extending their second moment analysis seems more challenging. We would like to mention that recently, an important step in this direction was taken in [MTMM18]. There, the authors considered a distributed version of Oja's centralized method for eigenvector approximation [Oja82], which works in the random sequential model of communication. The authors considered an averaging scheme that is very similar to the one of [BCN+17c, BCM+18]. They were able to show that their decentralized scheme can compute approximations of the k main eigenvectors of matrices derived from the (possibly weighted) matrix from which edges are sampled in every step, including its Laplacian. This way, they were able to prove both weak and strong reconstruction for the stochastic block model. Though, strictly speaking, the eigenvector computation and community detection algorithms proposed in [MTMM18] are not dynamics in the sense of Definition 1,[33] the basic building block is an averaging dynamics, close to Algorithm 31.

References

[AAB+11] Y. Afek, N. Alon, O. Barad, E. Hornstein, N. Barkai, and Z. Bar-Joseph. A biological solution to a fundamental distributed computing problem. *Science*, 331(6014):183–185, 2011.

[AAD+06a] D. Angluin, J. Aspnes, Z. Diamadi, M. J. Fischer, and R. Peralta. Computation in networks of passively mobile finite–state sensors. *Distributed computing*, 18(4):235–253, 2006.

[AAD+06b] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4), January 2006.

[33] For example, their eigenvector computation algorithm has distinct averaging and orthogonalization phases.

[AAE08] D. Angluin, J. Aspnes, and D. Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, 2008. (Preliminary version in DISC'07).

[AAE+17] Dan Alistarh, James Aspnes, David Eisenstat, Rati Gelashvili, and Ronald L. Rivest. Time-space Trade-offs in Population Protocols. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 2560–2579, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics.

[AAFJ08] D. Angluin, J. Aspnes, M. J. Fischer, and H. Jiang. Self-stabilizing population protocols. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(4):13, 2008.

[AAG18] D. Alistarh, J. Aspnes, and R. Gelashvili. Space-Optimal Majority in Population Protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, Proceedings, pages 2221–2239. Society for Industrial and Applied Mathematics, January 2018.

[ABH14] Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Trans. on Information Theory*, 62(1):471–487, 2014.

[AC88] N. Alon and F.R.K. Chung. Explicit construction of linear sized tolerant networks. *Discrete Mathematics*, 72(1):15 – 19, 1988.

[AF95] David Aldous and James Fill. Reversible markov chains and random walks on graphs, 1995.

[AFJ06] D. Angluin, M. J. Fischer, and H. Jiang. Stabilizing consensus in mobile networks. In *Proceedings of Distributed Computing in Sensor Systems (DCOSS'06)*, volume 4026 of *LNCS*, pages 37–50, 2006.

[AG15] D. Alistarh and R. Gelashvili. Polylogarithmic-time leader election in population protocols. In *Proceedings, Part II, of the 42nd International Colloquium on Automata, Languages, and Programming*, volume 9135 of *ICALP'15*, pages 479–491. Springer-Verlag New York, Inc., 2015.

[AGV15] Dan Alistarh, Rati Gelashvili, and Milan Vojnović. Fast and Exact Majority in Population Protocols. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, PODC '15, New York, NY, USA, 2015. ACM.

[Ald13] David Aldous. Interacting particle systems as stochastic social dynamics. *Bernoulli*, 19(4):1122–1149, 2013.

[AR07] J. Aspnes and E. Ruppert. An introduction to population protocols. *Bulletin of the EATCS*, 93:98–117, 2007.

[AW04] Hagit Attiya and Jennifer Welch. *Distributed computing: fundamentals, simulations, and advanced topics*, volume 19. John Wiley & Sons, 2004.

[BBK11] J. Beauquier, J. Burman, and S. Kutten. A self-stabilizing transformer for population protocols with covering. *Theoretical Computer Science*, 412(33):4247–4259, 2011.

[BCE+17] Petra Berenbrink, Andrea E. F. Clementi, Robert Elsässer, Peter Kling, Frederik Mallmann-Trenn, and Emanuele Natale. Ignore or comply?: On breaking symmetry in consensus. In *Proceedings of the 36th ACM Symposium on Principles of Distributed Computing, PODC'17*, pages 335–344, 2017.

[BCM+18] Luca Becchetti, Andrea E. F. Clementi, Pasin Manurangsi, Emanuele Natale, Francesco Pasquale, Prasad Raghavendra, and Luca Trevisan. Average whenever you meet: Opportunistic protocols for community detection. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, pages 7:1–7:13, 2018.

[BCN+15] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and R. Silvestri. Plurality consensus in the gossip model. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'15)*, pages 371–390. SIAM, 2015.

[BCN+16] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan. Stabilizing consensus with many opinions. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'16)*. SIAM, 2016.

[BCN+17a] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, R. Silvestri, and L. Trevisan. Simple dynamics for plurality consensus. *Distributed Computing (Ext. Abs. in ACM SPAA'14)*, 30(4):293–306, Aug 2017.

[BCN+17b] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and L. Trevisan. Find your place: Simple distributed algorithms for community detection. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*, pages 940–959. Society for Industrial and Applied Mathematics, 2017.

[BCN+17c] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, Prasad Raghavendra, and Luca Trevisan. Friend or foe? population protocols can perform community detection. *arXiv preprint arXiv:1703.05045*, 2017.

[BGPS06] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, pages 2508–2530, 2006.

[BTV09] F. Bénézit, P. Thiran, and M. Vetterli. Interval consensus: From quantized gossip to voting. In *Proceedings of the 34th IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '09, pages 3661–3664. IEEE Computer Society, 2009.

[BTV11] F. Benezit, P. Thiran, and M. Vetterli. The Distributed Multiple Voting Problem. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):791–804, August 2011.

[CCD+16] Andrea E. F. Clementi, Pierluigi Crescenzi, Carola Doerr, Pierre Fraigniaud, Francesco Pasquale, and Riccardo Silvestri. Rumor spreading in random evolving graphs. *Random Struct. Algorithms*, 48(2):290–312, 2016. (Ext. Abs. in ESA 2013).

[CCN12] L. Cardelli and A. Csikász-Nagy. The cell cycle switch computes approximate majority. *Scientific Reports*, Vol. 2, 2012.

[CDRR16] S. Cannon, J. J. Daymude, D. Randall, and A. W. Richa. A markov chain algorithm for compression in self-organizing particle systems. In *Proceedings of the 2016 ACM*

Symposium on Principles of Distributed Computing, PODC '16, pages 279–288. ACM, 2016.

[CER14] C. Cooper, R. Elsasser, and T. Radzik. The power of two choices in distributed voting. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP'14)*, volume 8573 of *LNCS*, pages 435–446. Springer, 2014.

[CER+15] C. Cooper, R. Elsässer, T. Radzik, N. Rivera, and T. Shiraga. Fast consensus for voting on general expander graphs. In *Distributed Computing: 29th International Symposium, (DISC '15)*, pages 248–262. Springer, Springer Berlin Heidelberg, 2015.

[CGG+17] A Clementi, M. Ghaffari, L. Gualà, F. Natale, E. Pasquale, and G. Scornavacca. A tight analysis of the parallel undecided-state dynamics with two colors. *CoRR*, abs/1707.05135, 2017.

[CGPS17] A. Clementi, L. Gualà, F. Pasquale, and G. Scornavacca. Brief announcement: On the parallel undecided-state dynamics with two colors. In *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, pages 47:1–47:4, 2017.

[Cha12] Bernard Chazelle. Natural algorithms and influence systems. *Commun. ACM*, 55(12):101–110, December 2012.

[CIG+15] Andrea E. F. Clementi, Miriam Di Ianni, Giorgio Gambosi, Emanuele Natale, and Riccardo Silvestri. Distributed community detection in dynamic graphs. *Theor. Comput. Sci.*, 584:19–41, 2015.

[CKP16] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL model. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 615–624, 2016.

[CKW16] L. Cardelli, M. Z. Kwiatkowska, and M. Whitby. *Chemical Reaction Network Designs for Asynchronous Logic Circuits*, pages 67–81. Springer, 2016.

[CLP10] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Almost tight bounds for rumour spreading with conductance. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 399–408, 2010.

[CLP11] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumor spreading in social networks. *Theoretical Computer Science*, 412(24):2602–2610, 2011. (Preliminary version in ICALP'09).

[CNNS18] Emilio Cruciani, Emanuele Natale, André Nusser, and Giacomo Scornavacca. Phase transition of the 2-choices dynamics on core-periphery networks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 777–785, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[CNS19] Emilio Cruciani, Emanuele Natale, and Giacomo Scornavacca. Distributed Community Detection via Metastability of the 2-Choices Dynamics. In *AAAI 2019 - Thirty-Third AAAI Conference Association for the Advancement of Artificial Intelligence*, Honolulu, Hawaii, United States, January 2019.

[CO10] Amin Coja-Oghlan. Graph partitioning via adaptive spectral techniques. *Combinatorics, Probability and Computing*, 19(02):227–284, 2010.

[CRRS17] Colin S Cooper, Tomasz Radzik, Nicolás Rivera, and Takeharu Shiraga. Fast plurality consensus in regular expanders. In *DISC*, 2017.

[CSWB09] M. Cook, D. Soloveichik, E. Winfree, and J. Bruck. Programmability of chemical reaction networks. In *Algorithmic Bioprocesses*, pages 543–584. Springer, 2009.

[DDG+14] Z. Derakhshandeh, S. Dolev, R. Gmyr, A. W. Richa, C. Scheideler, and T. Strothmann. Brief announcement: Amoebot – a new model for programmable matter. In *Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'14)*, pages 220–222. ACM, 2014.

[DeG74] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[DEM+18] David Doty, Mahsa Eftekhari, Othon Michail, Paul G Spirakis, and Michail Theofilatos. Exact size counting in uniform population protocols in nearly logarithmic time. *arXiv preprint arXiv:1805.04832*, 2018.

[DF89] Martin E. Dyer and Alan M. Frieze. The solution of some random NP-hard problems in polynomial expected time. *Journal of Algorithms*, 10(4):451–489, 1989.

[DGH+87] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC'12)*, pages 1–12. ACM, 1987.

[DGM+11] B. Doerr, L. A. Goldberg, L. Minder, T. Sauerwald, and C. Scheideler. Stabilizing consensus with the power of two choices. In *Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'11)*, pages 149–158. ACM, 2011.

[DH07] D. Dolev and E. N. Hoch. On self-stabilizing synchronous actions despite byzantine attacks. In *Proceedings of the International Symposium on Distributed Computing (DISC'07)*, pages 193–207, 2007.

[Dij74] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM*, 17(11):643–644, 1974.

[DKMZ11] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.

[DKS10] Shlomi Dolev, Ronen I. Kat, and Elad Michael Schiller. When consensus meets self-stabilization. *J. Comput. Syst. Sci.*, 76(8):884–900, 2010.

[Dol00a] S. Dolev. *Self-stabilization*. MIT press, 2000.

[Dol00b] Shlomi Dolev. *Self-Stabilization*. MIT Press, 2000.

[Dot14] D. Doty. Timing in chemical reaction networks. In *Proceedings of 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'14)*, pages 772–784. SIAM, 2014.

[DV12] M. Draief and M. Vojnovic. Convergence speed of binary interval consensus. *SIAM Journal on Control and Optimization*, 50(3):1087–1109, 2012.

[EFK+17] Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Frederik Mallmann-Trenn, and Horst Trinker. Brief announcement: Rapid asynchronous plurality consensus. In *Proceedings of the 36th Annnual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, (PODC'17) (full version at ArXive https://arxiv.org/abs/1602.04667)*, pages 363–365, 2017.

[FHK14] O. Feinerman, B. Haeupler, and A. Korman. Breathe before speaking: Efficient information dissemination despite noisy, limited and anonymous communication. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC '14)*, pages 114–123. ACM, 2014.

[FHK15] O. Feinerman, B. Haeupler, and A. Korman. Breathe before speaking: efficient information dissemination despite noisy, limited and anonymous communication. *Distributed Computing*, pages 1–17, 2015.

[FJ90] N. E. Friedkin and E. C. Johnsen. Social influence and opinions. *The Journal of Mathematical Sociology*, 15(3-4):193–206, 1990.

[FK13] Ofer Feinerman and Amos Korman. Theoretical distributed computing meets biology: A review. In *International Conference on Distributed Computing and Internet Technology*, pages 1–18. Springer, 2013.

[FKP13] Pierre Fraigniaud, Amos Korman, and David Peleg. Towards a complexity theory for local distributed computing. *Journal of the ACM (JACM)*, 60(5):35, 2013.

[FN16] P. Fraigniaud and E. Natale. Noisy rumor spreading and plurality consensus. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC'16)*, pages 127–136. ACM, 2016.

[FPM+02] N. R. Franks, S. C. Pratt, E. B. Mallon, N. F. Britton, and D. J. T. Sumpter. Information flow, opinion polling and collective intelligence in house–hunting social insects. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 357(1427):1567–1583, 2002.

[Gia16] G. Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *Symposium on Theoretical Aspects of Computer Science (STACS'11)*, LNCS, pages 57–68,. Springer, 2016.

[GL18] Mohsen Ghaffari and Johannes Lengler. Nearly-tight analysis for 2-choice and 3-majority consensus dynamics. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, (PODC'18), pages 305–313. ACM, 2018.

[HK05] Rainer Hegselmann and Ulrich Krause. Opinion dynamics driven by various ways of averaging. *Comput. Econ.*, 25(4):381–405, June 2005.

[HP01] Yehuda Hassin and David Peleg. Distributed Probabilistic Polling and Applications to Proportionate Agreement. *Information and Computation*, 171(2):248–268, December 2001.

[JS98] Mark Jerrum and Gregory B. Sorkin. The metropolis algorithm for graph bisection. *Discrete Applied Mathematics*, 82(1):155–175, 1998.

[Kar11] Richard M Karp. Understanding science through the computational lens. *Journal of Computer Science and Technology*, 26(4):569–577, 2011.

[KDG03] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'03)*, pages 482–491. IEEE, 2003.

[KMM+13] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

[KMTS19] Varun Kanade, Frederik Mallmann-Trenn, and Thomas Sauerwald. On Coalescence Time in Graphs: When is Coalescing As Fast As Meeting? In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 956–965, Philadelphia, PA, USA, 2019. Society for Industrial and Applied Mathematics. event-place: San Diego, California.

[KR19] Nan Kang and Nicolás Rivera. Best-of-three voting on dense graphs. In *The 31st ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '19, pages 115–121, New York, NY, USA, 2019. ACM.

[KRBN10] Pavel L Krapivsky, Sidney Redner, and Eli Ben-Naim. *A Kinetic View of Statistical Physics.* Cambridge University Press, Leiden, 2010.

[KSSV00] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science (FOCS'00)*, pages 565–574. IEEE, 2000.

[L+93] László Lovász et al. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 2(1):1–46, 1993.

[LGT14] James R Lee, Shayan Oveis Gharan, and Luca Trevisan. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM (JACM)*, 61(6):37, 2014.

[Lig12] T. Liggett. *Interacting particle systems*, volume 276. Springer, 2012.

[LM10] X. Liu and T. Murata. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, 389(7):1493–1500, 2010.

[LPW09] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2009.

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *Proc. of the 42nd IEEE Symp. on Foundations of Computer Science (FOCS'01)*, pages 529–537, 2001.

[MNRS14] George B. Mertzios, Sotiris E. Nikoletseas, Christoforos Raptopoulos, and Paul G. Spirakis. Determining majority in networks with local interactions and very small local memory. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 871–882, 2014.

[MNRS16] George Mertzios, Sotiris Nikoletseas, Christoforos Raptopoulos, and Paul Spirakis. Determining majority in networks with local interactions and very small local memory. *Distributed Computing (Ext.Abs. in ICALP'14)*, 30, 07 2016.

[MNS14] E. Mossel, J. Neeman, and A. Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3-4):431–461, 2014.

[MNT14] E. Mossel, J. Neeman, and O. Tamuz. Majority dynamics and aggregation of information in social networks. *Autonomous Agents and Multi-Agent Systems*, 28(3):408–429, 2014.

[MS10] E. Mossel and G. Schoenebeck. Reaching consensus on social networks. In *Proceedings of the 2nd Innovations in Computer Science (ITCS'10)*, pages 214–229, 2010.

[MTMM18] Frederik Mallmann-Trenn, Cameron Musco, and Christopher Musco. Eigenvector computation and community detection in asynchronous gossip models. *arXiv preprint arXiv:1804.08548*, 2018.

[NBJ15] Saket Navlakha and Ziv Bar-Joseph. Distributed information processing in biological and computational systems. *Communications of the ACM*, 58(1):94–102, 2015.

[Oja82] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[OT09] A. Olshevsky and J. N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55, 2009.

[Pel00] D. Peleg. Distributed computing. *SIAM Monographs on discrete mathematics and applications*, 5, 2000.

[PSL80] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.

[Rab83] M. O. Rabin. Randomized byzantine generals. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (SFCS'83)*, pages 403–409. IEEE, 1983.

[Reu16] B. Reus. *Molecular Computing*, pages 299–316. Springer, 2016.

[RN19] Iliad Ramezani and Emanuele Natale. On the Necessary Memory to Compute the Plurality in Multi-Agent Systems. In *CIAC'19 - 11th International Conference on Algorithms and Complexity*, Rome, Italy, May 2019.

[Sha09] D. Shah. *Gossip algorithms*. Now Publishers Inc, 2009.

[SSNG15] S. Salehkaleybar, A. Sharif-Nassab, and S.J. Golestani. Distributed Voting/Ranking with Optimal Number of States per Node. *IEEE Transactions on Signal and Information Processing over Networks*, PP(99):1–1, 2015.

[Tro12] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

[TW93] Tetali, P. and Winkler, P. Simultaneous reversible Markov chains. In *Combinatorics: Paul Erdós Is Eighty,*, volume Vol. 1, pages 433–451, Budapest, 1993. János Bolyai Mathematical Society - D. Miklós, V. T. Sós, and T. Szónyi, Eds.

[Wol02] S. Wolfram. *A New Kind of Science*, volume 5. Wolfram media Champaign, 2002.

[XBK07] L. Xiao, S. Boyd, and S. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.