Voter Model Meets Rumour Spreading: A Study of Consensus Protocols on Graphs with Agnostic Nodes

Marcelo Matheus Gauy
São Paulo State University
Itapeva, Brazil
marcelo.gauy@unesp.br

Anna Abramishvili
King's College London
London, United Kingdom
anna.abramishvili@kcl.ac.uk

Eduardo Colli
University of São Paulo
São Paulo, Brazil
colli@ime.usp.br

Tiago Madeira
University of São Paulo
São Paulo, Brazil
tmadeira@alumni.usp.br

Frederik Mallmann-Trenn
King's College London
London, United Kingdom
frederik.mallmann-trenn@kcl.ac.uk

Vinícius Franco Vasconcelos
University of São Paulo
São Paulo, Brazil
vfvmat@ime.usp.br

David Kohan Marzagão
King's College London
London, United Kingdom
david.kohan@kcl.ac.uk

ABSTRACT

Problems of consensus in multi-agent systems are often viewed as a series of independent, simultaneous local decisions made between a limited set of options, all aimed at reaching a global agreement. Key challenges in these protocols include estimating the likelihood of various outcomes and finding bounds for how long it may take to achieve consensus, if it occurs at all.

To date, little attention has been given to the case where some agents have no initial opinion. In this paper, we introduce a variant of the consensus problem which includes what we call 'agnostic' nodes and frame it as a combination of two known and well-studied processes: voter model and rumour spreading. We show (1) a martingale that describes the probability of consensus for a given colour, (2) bounds on the number of steps for the process to end using results from rumour spreading and voter models, (3) closed formulas for the probability of consensus in a few special cases, and (4) that the computational complexity of estimating the probability with a Markov chain Monte Carlo process is $O(n^2 \log n)$ for general graphs and $O(n \log n)$ for Erdős-Rényi graphs, which makes it an efficient method for estimating probabilities of consensus. Furthermore, we present experimental results suggesting that the number of runs needed for a given standard error decreases when the number of nodes increases.

KEYWORDS

Consensus processes; voter model; rumour spreading; multi-agent consensus

ACM Reference Format:
Marcelo Matheus Gauy, Anna Abramishvili, Eduardo Colli, Tiago Madeira, Frederik Mallmann-Trenn, Vinícius Franco Vasconcelos, and David Kohan Marzagão. 2025. Voter Model Meets Rumour Spreading: A Study of Consensus Protocols on Graphs with Agnostic Nodes. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 10 pages.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

In multi-agent consensus problems, agents make a sequence of independent and autonomous choices from a finite set based on their local information. Agents have a shared goal of reaching a consensus state, in which they all represent the same choice. The process is often abstracted as a graph in which nodes represent agents, their colour represent their current choice (or opinion, or state), and edges of the graph represent visibility or influence between agents.

Multi-agent processes on graphs have been shown to have several applications, including autonomous robots or drones [25, 44], electrical flow estimation [5, 15], mutation fixation in biology [29, 34], among others. In the simplest of such processes, the *voter model*, at each point in time (called 'round') nodes may change their colour based on the opinion of their neighbours until consensus is reached (e.g. the case where all nodes share the same colour). More formally, this process can be either synchronous or asynchronous. In the asynchronous case, a node is chosen uniformly at random and selects a neighbour proportional to the weight of the edge between then. It then adopts the colour (or opinion) of the chosen neighbour. In the synchronous case, all nodes act simultaneously and independently (i.e., the choice of one does not affect the choice of the other in the same round). Consensus processes in multi-agent systems have been extensively studied (e.g. [8, 27, 30, 31, 36]).

Given an initial colour configuration, the probability of consensus and time-bounds for the number of rounds before such consensus is achieved for a given colour are some of the core problems studied in this domain. Extensive results have been found for both synchronous [24] and asynchronous [14] process, as well as for processes with undecided states ([2, 3, 11, 40]). In these, nodes do not change directly from one colour to another but transition via an 'undecided' state in between them.

One of the features of the classical voter model is that it assumes all nodes start off with a colour/opinion. In some scenarios, we may want to model a process in which, at the start, some nodes do not have an opinion at all, which may be different from being 'undecided' after been given a set of option, as they have not been in contact with any of the opinions in this process. One can think of examples related to election scenarios in which voters do not yet know the candidates and may be therefore influenced by the first contact with a candidate. Or in a process on a blockchain, in which new blocks are mined and the information of new mined blocks traverse the network, possibly competing with other new blocks mined at a similar time.

In this paper, we introduce and study a variant of the voter model in which nodes can have an extra state, which we call 'agnostic' (represented by the colour, say, white). We call it *voter model with agnostic nodes*. Agnostic nodes become gnostic if they choose a gnostic neighbour to copy its colour. Once gnostic, they can never become agnostic again, i.e., if a gnostic node chooses an agnostic one, the gnostic node keeps its current colour and nothing happens. For a precise definition of the problem, see Section 2.3. The main challenge posed when studying this variant is that there is an asymmetry between states, in that agnostic states can become gnostic but not vice versa. We provide an efficient Monte Carlo algorithm for estimating the probability of consensus in this voter model with agnostic states for any graph and any initial configuration. Furthermore, we provide time-bounds for consensus to be achieved.

The variant we study can also be seen as a generalisation of the rumour spreading model [1, 19, 20, 39]. In it, there are nodes that are 'informed' and nodes that are 'uninformed' and the process studies the time bounds until all nodes become informed. Like in our variant, an 'informed' node cannot become 'uninformed'. Our voter model with agnostic nodes is analogous to two or more rumours that compete not only to gather more agnostic nodes but also to flip the opinion of other gnostic nodes. Alternatively, one can also see the voter model with agnostic nodes as a combination of two models happening simultaneously: the classical voter model with a rumour spreading process.

The following motivates the problem with a toy example.

Example 1.1. Consider the graph and initial configuration depicted in Figure 1. In this example, each node chooses a neighbour with uniform probability. For example, v_2 has 50% chance of choosing v_1 and thus becoming blue at round S_1 and 50% chance of choosing v_3 and thus becoming red at round S_1. If a node chooses an agnostic node, their colour does not change. For example, if v_1 chooses v_2, v_1 will stay blue.

With that in mind, what is the probability that there will be, say, a red consensus? What can we say about the expected number of rounds until that happens?

We will return to Example 1.1 later in this paper. Our main contributions of this paper are as follows:

(1) We obtain a martingale for the voter model with agnostic states for the case in which the underlying (weighted) graph represents a reversible Markov chain (Theorem 3.1).
(2) Although the martingale obtained may not be efficiently computed in the general case, we prove a closed formula for the consensus probability in the case of a complete graph with an asynchronous process (Corollary 3.3).
(3) We show that the existence of agnostic nodes does not affect the complexity bounds for the expected time of achieving consensus, as the agnostic nodes typically disappear faster than consensus is achieved (Lemma 4.1 and Propositions 4.2, 4.3). This comes from standard known results for rumour spreading processes.
(4) We present a Markov chain Monte Carlo (MCMC) algorithm (Section 3.2) to efficiently compute the consensus probability based on the fact that agnostic nodes disappear quickly (Section 4). This is an efficient algorithm that provides an unbiased estimate.
(5) We present an experimental analysis to support results using our MCMC algorithm, showing that few runs are necessary to obtain good probability estimates (Section 5). It also suggests estimates get better as graph sizes increase.

2 BACKGROUND AND MAIN DEFINITIONS

In this section, we present concepts and results from the literature that will be used in subsequent sections. We first introduce the classical version of consensus protocol used in this paper, also known as voter model [14, 18, 24, 26, 35, 37], in which all vertices have an initial opinion. We then propose a variant of the voter model in which some vertices do not have an initial opinion, which we label agnostic vertices. The voter model has been widely studied in the context of multi-agent systems. The winning probabilities of each colour and bounds on the convergence time were obtained for undirected graphs by Hassin & Peleg [24]. Cooper & Rivera extended this work to the linear voting model, which captures digraphs as well as several similar consensus processes [14].

2.1 Classical Voter Model

The (pull) **voter model** defines a round-based consensus process on a strongly connected directed graph $G = (V, E)$.[1] In such processes agents are represented by nodes in this graph. At each round, each node has a colour associated to it, representing the respective agent's current state (or opinion). Their goal is to reach consensus, i.e., a situation where every agent is in the same state. To that end, at each round, all agents update their state synchronously

[1] Henceforth, we assume all graphs are strongly connected unless stated otherwise.

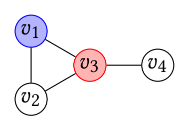


Figure 1: A motivational example of an undirected graph with an initial configuration $S_0 = s_0$ consisting of one blue node (v_1), one red node (v_3), and two agnostic nodes (v_2 and v_4). Transition probabilities are uniform, i.e., v_3 has $\frac{1}{3}$ chance of choosing a given neighbour, whereas v_4 chooses v_3 and becomes red with probability 1. What are the probabilities of consensus in this case?

based on the colour of their out-neighbours.[2] The probability that v copies colour of node u in a given round is represented by the weight of edge (v, u). The weights of edges starting at a given node are assumed to be positive and to sum to 1. We collate all these probabilities in an out-matrix H, which can be also seen as the adjacency matrix of G where entry $H(v, u)$ represents the weight of edge (v, u). We adopt the notation $H(v, u) = 0$ if $(v, u) \notin E$, and note that self loops are allowed and thus v may adopt its own colour. Once reached, a consensus is stable.

Let $X = \{c_1, \ldots, c_k\}$ be the set of all possible colours on a consensus process. A **configuration** on a graph $G = (V, E)$ is a function $s \in X^V$ that associates each node $v \in V$ with a colour $c \in X$, i.e., $s(v)$ represents v's colour in configuration s. More formally, a process is a sequence of random variables $\{S_t\}_{t\geq 0}$, with $S_{t+1} \in X^V$ being a configuration generated based on S_t. We say colour i **wins** the process if a configuration $S_t = s$, such that $s(v) = i$ for all v, is reached. Here, we assume processes converge with probability 1. For discussion of fringe cases and work related to graphs in which processes may not converge, see, e.g., [32].

Observe that the out-matrix H of the graph G can be seen as the transition matrix of a time homogeneous Markov chain (e.g., see Chapter 6, Grimmett et al. (2001)) representing the probabilities of one round in the consensus process [14]. If G is strongly connected, this Markov chain is irreducible and finite, so there exists a unique stationary distribution μ of H, that is, there is a row vector μ such that $\mu H = \mu$. We call the values $\mu(v)$ the influence of the vertex v in the consensus protocol. In this context, previous work [14], show that the winning probabilities of each colour can be determined by the initial configuration only and are given by the following proposition.

Proposition 2.1 (Cooper and Rivera 2016). *Consider a consensus process on a strongly connected graph G (further, we assume G is such that consensus is always achieved for all initial configurations), with associated adjacency matrix H and μ its unique stationary distribution. Assume the initial configuration is given by $s \in \{c_1, \ldots, c_k\}^V$. Then, we have that the winning probability of colour c_i is:*

$$\mathbb{P}(\text{colour } c_i \text{ wins} \mid S_0 = s) = \sum_{v \in V, S(v) = c_i} \mu(v)$$

We can see as a corollary, that for (non-bipartite, connected) undirected graphs, the probability of a given colour winning is simply the number of incident edges in nodes of that colour divided by the $2E$, where E is the number of edges in the graph [24, Corollary 2.2 and Section 2.3]. Example 2.2 discusses the idea applied to our motivating example.

Example 2.2. Consider the modified version of Example 1.1 with a different initial configuration: agnostic nodes are instead gnostic and coloured, say, orange. In other words, assume we are under the assumptions of the classical voter model with 3 colours. We have $\mu = \frac{1}{8}(2, 2, 3, 1)$. Then, from the we have the probability of red winning being $\frac{3}{8}$, of blue winning being $\frac{2}{8}$ and of orange winning being $\frac{3}{8}$.

[2]For precision, we consider that agents change their state at the end of each round, after all nodes have made their decisions.

2.2 Rumour Spreading Process

A rumour spreading process on a graph represents the process of information 'travelling' across the edges to eventually reach all nodes on a graph. More formally, and using the notation for the voter model, we would have two colours, one being 'red' and the other representing a node being 'uninformed'. Many strategies of information transmission were designed, such as push, pull and push-and-pull. In this work, we will concentrate, as mentioned in Section 2.1, on the pull protocol. In it, if node v selects u, then v becomes informed if u is informed, otherwise v is unchanged (informed or uninformed). The process can be synchronous or asynchronous.

For the push protocol, node v selects u and pushes its state towards u. If v is agnostic, u retains its state, otherwise u adopts the state of v. Observe that the push protocol cannot be done synchronously for consensus problems as it is not clear how to resolve the possible ambiguity (nodes v_1 and v_2 with different gnostic states, both select the same node u). Asynchronous push-and-pull would be defined as is standard in rumour spreading theory. A random agnostic node is chosen and performs a pull, and a random gnostic node is chosen and performs a push. Observe that the asynchronous push-and-pull would eventually become equivalent to asynchronous push as agnostic vertices disappear. Lastly, observe that, just like the push protocol, push-and-pull cannot be done synchronously.

We will use standard techniques and proof ideas of the rumour spreading literature to show that the rumour spreading process is fast in general for the pull version. These are done in Propositions 4.2 and 4.3 where we use the results from the literature to obtain an $O(n \log(n))$ bound for general graphs and a $O(\log(n))$ bound for random graphs (in the synchronous case, the asynchronous case adds an extra factor n multiplying both). For other known bounds from the literature, see Sections 6 and 8.

2.3 Voter Model with agnostic Nodes

We now introduce the main concept to be explored in this work. The main difference of the process with agnostic nodes is that there is an asymmetry between gnostic and agnostic nodes: an agnostic node can become gnostic but gnostic nodes cannot become agnostic. A more precise definition is given as follows.

Definition 2.3 (Voter Model with agnostic Nodes). A (pull) voter model with agnostic nodes generalises the notion of (pull) voter model by changing the rule with which nodes update their colour. As before, at each round t, each node v chooses a one of its out-neighbours u proportionally to the weight of the edge in G. However,

(1) If u is agnostic, v does not change its colour.
(2) If u is gnostic, v copies colour of u.

At the same time that Definition 2.3 can be seen as a generalisation of the voter model, it can also be seen as a generalisation of the rumour spreading process. A node that is 'uninformed' behaves equivalently to an 'agnostic' node. The difference being, of course, that we consider more than one rumour spreading in the network, and competing with each other at the same time it influences agnostic (or uninformed) nodes.

We now go back to Example 1.1 and solve it by simply accounting for all possible states and their probabilities.

Example 2.4 (Example 1.1 continued). Recall Example 1.1. Here we solve it 'by hand' to motivate the introduction of a martingale property for this model. Figure 2 shows all possible configurations for S_1, the probabilities of reaching them (a_i) and the probability that red wins from each of them (b_i) calculated by applying Proposition 2.1. Note that $\mu = \frac{1}{8}(2, 2, 3, 1)$. We have that $\mathbb{P}(\text{red wins} \mid S_0 = s) = \frac{5}{8}$.

Remark. *Example 2.4 highlights the following: if we know the probabilities that agnostic nodes become, say, red, once they become gnostic, can we amend formula in Proposition 2.1 so that it is valid for the voter model with agnostic nodes? In Figure 1, such probabilities for nodes v_2 and v_4 are, respectively, $\frac{1}{2}$ and 1. Multiplying these values with the importance of each node and summing them up gives us, $0\mu(v_1) + \frac{1}{2}\mu(v_2) + 1\mu(v_3) + 1\mu(v_4) = \frac{5}{8}$, which coincides with the probability of red winning. We will show that this is not a coincidence in Theorem 3.1.*

3 PROBABILITY OF CONSENSUS

In this section, we will provide a general study on the probability of consensus for the (pull) voter model with agnostic nodes. Our theorems deal with the case with only two colours (red and blue) for gnostic nodes. It is easy to see that the generalisation for more colours is immediate.

3.1 Martingale Property

The following theorem provides a martingale (see, e.g., [23, Chapter 12]) associated with the voter model with agnostic nodes, which will then immediately give a formula such as in Proposition 2.1.

Theorem 3.1 (Martingale Property). *Let G be a graph with vertex set V. Suppose we have a (pull) voter model with agnostic states (either synchronous or asynchronous) with associated matrix H on G, and μ such that $\mu H = \mu$. Suppose, furthermore, that the voter model is a reversible Markov chain, or in other words that, $H(v, w)\mu(v) = H(w, v)\mu(w)$. Let S_t be the state of each vertex at time t. Moreover, define $R(v)$ as the event where v is coloured red once it is gnostic. Then the following sequence is a Martingale with respect to S_t:*

$$X_t = \sum_{v\in V} \mu(v)\mathbb{P}(R(v) \mid S_t) \tag{1}$$

As a result, if G is such that the process always converges [3] on G, then

$$\mathbb{P}(\text{red wins}) = X_0. \tag{2}$$

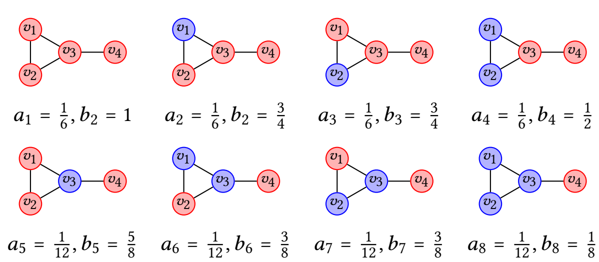


Figure 2: Every configuration s_{1i} which can be reached from S_0 in Example 1.1 after one round, i.e. $a_i := \mathbb{P}(S_1 = s_{1i}|S_0) > 0$. The probability of a red consensus in each case is denoted by b_i and can be calculated by applying Proposition 2.1. Therefore, the probability of red to win in Example 1.1 is $\sum_{i=1}^{8} a_i b_i = \frac{5}{8}$.

Proof. First note that if v is already red in S_t, then $\mathbb{P}(R(v) \mid S_t) = 1$ and if already blue, $\mathbb{P}(R(v) \mid S_t) = 0$. Let $S_t(v) = \{0, 1, 2\}$ denote the state of each vertex v at time t, with 0 representing color white, 1 representing red, and 2 representing blue. We want to show that

$$\mathbb{E}(X_{t+1}|S_t) = X_t. \tag{3}$$

Observe that $X_{t+1} = \sum_{v\in V} \mu(v)\mathbb{P}(R(v)|S_{t+1})$, and we have, by linearity of expectation:

$$\mathbb{E}(X_{t+1} \mid S_t) = \sum_{v\in V} \mu(v)\mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t).$$

We distinguish two cases: 1) $S_t(v)$ gnostic (that is equal 1 or 2) and 2) $S_t(v)$ agnostic (equal 0). For the second case observe that

$$\mathbb{P}(R(v) \mid S_t) = \sum_{S_{t+1}} \mathbb{P}(R(v) \mid S_{t+1})\mathbb{P}(S_{t+1} \mid S_t)$$

by law of total probability. The second expression is the definition of the conditional expectation so we have

$$\mathbb{P}(R(v) \mid S_t) = \mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t)$$

for the case where we assume $S_t(v) = 0$. We have

$$\mathbb{E}(X_{t+1} \mid S_t) = \sum_{v\in V} \mu(v)\mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t).$$

Since $X_t = \sum_{v\in V} \mu(v)\mathbb{P}(R(v) \mid S_t)$, the goal is then to show that:

$$\sum_{v\in V} \mu(v)\mathbb{P}(R(v) \mid S_t) = \sum_{v\in V} \mu(v)\mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t)$$

For that we split both left and right side in two sums:

$$\sum_{S_t(v)=0} \mu(v)\mathbb{P}(R(v) \mid S_t) = \sum_{S_t(v)=0} \mu(v)\mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t)$$

and:

$$\sum_{S_t(v)\in\{1,2\}} \mu(v)\mathbb{P}(R(v) \mid S_t) = \sum_{S_t(v)\in\{1,2\}} \mu(v)\mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t)$$

that is, we look at the graph at time t and split the vertices between agnostic and gnostic and we will show that the sum is equal in each part. The first case of only agnostic vertices follows from $\mathbb{P}(R(v) \mid S_t) = \mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t)$ which is why have shown this equality. The second case is done in the rest of the text and requires reversibility.

The equation $\mathbb{E}(X_{t+1}|S_t) = X_t$ then becomes equivalent to showing that:

$$\sum_{S_t(v)=1} \mu(v) = \sum_{S_t(v)\in\{1,2\}} \mu(v)\mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t)$$

because we have already shown that the terms on the left and right side where $S_t(v) = 0$ are equal.

We now turn to the case where $S_t(v) \neq 0$. Observe that

$$\mathbb{E}(\mathbb{P}(R(v) \mid S_{t+1}) \mid S_t) = \mathbb{P}(S_{t+1}(v) = 1 \mid S_t),$$

[3] For some G, such as bipartite graphs, the process may never converge. For a general result on convergence, see [32].

as if $S_t(v)$ is gnostic, then $S_{t+1}(v)$ is also gnostic. Now, in terms of H, we can write $\mathbb{P}(S_{t+1}(v) = 1|S_t)$ equal to the sum $\sum_{S_t(w)\in\{0,1\}} H(v, w)$ if $S_t(v) = 1$ and equal to $\sum_{S_t(w)=1} H(v, w)$ if $S_t(v) = 2$.[4] Thus, we just need to verify the equality:

$$\sum_{S_t(v)=1} \mu(v) = \sum_{S_t(v)=1} \mu(v) \sum_{S_t(w)\in\{0,1\}} H(v, w) + \sum_{S_t(v)=2} \mu(v) \sum_{S_t(w)=1} H(v, w).$$

Using the fact that $\sum_w H(v, w) = 1$ we can rewrite the last equality as being equivalent to:

$$\sum_{S_t(v)=1} \mu(v) \sum_{S_t(w)=2} H(v, w) = \sum_{S_t(v)=2} \mu(v) \sum_{S_t(w)=1} H(v, w)$$

This last equality is equivalent[5] to $H(v, w)\mu(v) = H(w, v)\mu(w)$ for every pair of neighbours w, v. This holds for example when the graphs are undirected and the choice of neighbour is uniform. Thus, (3) follows.

Now we assume G is such that every process converges and prove Equation 2. It is a known technique to use Doob's Stopping Theorem to go from a martingale to the probability of consensus, and that is what we use to show Equation 2. Doob's Stopping Theorem guarantees that $\mathbb{E}(X_\tau) = \mathbb{E}(X_0) = X_0$, where τ is the time where consensus is reached. Together with the fact that $\mathbb{E}(X_\tau) = \mathbb{P}(\text{red wins})\mathbb{E}(X_\tau \mid \text{red wins}) + \mathbb{P}(\text{blue wins})E(X_\tau \mid \text{blue wins})$ (due to the process converging to either a complete red or blue state), and noting that $\mathbb{E}(X_\tau \mid \text{red wins}) = 1$ and $\mathbb{E}(X_\tau \mid \text{blue wins}) = 0$, we have finally that $\mathbb{P}(\text{red wins}) = X_0$. □

3.1.1 Counterexample indicating that Reversibility is Required. One may wonder whether the property $H(v, w)\mu(v) = H(w, v)\mu(w)$ of reversibility is required for the martingale to hold, as this is not a requirement in the generalised voter model (Proposition 2.1). Here, we give a simple (counter-)example of a graph with 3 nodes such that the voter model on it is not a reversible Markov chain.

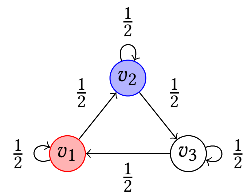
Example 3.2 (Counterexample for non-reversible chains). Consider the initial configuration $S_0 = s_0$ depicted in Figure 3a in a graph with matrix H given by:

$$H = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \tag{4}$$

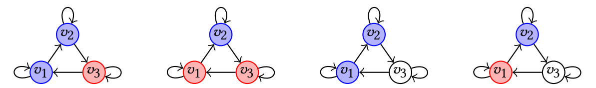
Solving $\mu H = \mu$, we get $\mu = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Note that the chain is not reversible because, e.g., $H(v_1, v_2)\mu(v_1) = \frac{1}{6} \neq 0 = H(v_2, v_1)\mu(v_2)$. We can determine $\mathbb{P}(R(v_3) \mid S_0)$ by solving the equation putting together S_0 (Figure 3a) and the four possible states for S_1 (Figures 3b, 3c, 3d, 3e) also using that $\mathbb{P}(R(v_3) \mid S_0 = s_0) = \mathbb{P}(R(v_3) \mid S_1 = s_{14})$, i.e,

$$\mathbb{P}(R(v_3) \mid S_0) = s_0) = \frac{1}{4}\left(1 + 1 + 0 + \mathbb{P}(R(v_3) \mid S_0)\right) \tag{5}$$

We get that $\mathbb{P}(R(v_3) \mid S_0 = s_0) = \frac{2}{3}$. From that, we get that $X_0 = \frac{5}{9}$ and that, on average, X_1 is given by $\mathbb{E}(X_1 \mid S_0) = \frac{1}{4}\left(\frac{1}{3} + \frac{2}{3} + 0 + \frac{5}{9}\right) = \frac{7}{18} \neq X_0$. Another way to see the martingale property does not hold is to evaluate the probability of red winning using a similar technique as in Equation 5 to get $\mathbb{P}(\text{red wins} \mid S_0) = \frac{1}{3} \neq X_0$.



(a) Initial Config. $S_0 = s_0$. We have $X_0 = \frac{5}{9}$, but $\mathbb{P}(\text{red wins} \mid S_0) = \frac{1}{3}$.



(b) Config. s_{11} with $X_1 = \frac{1}{3}$. **(c) Config. s_{12} with $X_1 = \frac{2}{3}$.** **(d) Config. s_{13} with $X_1 = 0$.** **(e) Config. s_{14} with $X_1 = \frac{5}{9}$.**

Figure 3: Counterexample for the conjecture that the Martingale property (Theorem 3.1) is valid for non-reversible chains. Note that edge weights were omitted from Figures 3b,3c, 3d, and 3e for readability.

Theorem 3.1 is not easy to use for computing the probabilities in the general case. One simple use case is when there are no edges connecting agnostic vertices. It is clear in this case that $\mathbb{P}(R(v) \mid S_0)$ equals the proportion of red nodes among the neighbours (as we have seen in Remark 2.3). There is an additional case where one can say something about the consensus probability as presented in the following corollary.

Corollary 3.3 (Solution for Complete Graph). *Let G be a complete graph and consider the asynchronous (pull) voter model with agnostic states, with initial state S_0. Let γ denote the proportion of red nodes among the gnostic ones. Then:*

$$\mathbb{P}(R(u) \mid S_0) = \gamma \text{ for all } u \in I \tag{6}$$

As a consequence,

$$\mathbb{P}(\text{red wins} \mid S_0) = \gamma \tag{7}$$

Proof. See proof in extended version of the paper [22]. □

3.2 Estimating probabilities with Markov chain Monte Carlo

While Theorem 3.1 gives us a potential means of computing the exact probability of achieving consensus with a certain colour, a generalization of Proposition 2.1, i.e., in general solution for $\mathbb{P}(R(v) \mid S_t)$ may not be efficient to compute. With that in mind, we propose evaluating the probabilities of consensus of the voter model with agnostic nodes on general graphs by:

By using known results from rumour spreading, Section 4 shows that the simulation time required until all nodes are gnostic is at most $O(n^2 \log(n))$ for general graphs and only $O(n \log(n))$ for Erdös Rényi random graphs with high probability. As usual, 'with high probability' means with probability going to 1 as the graph size n goes to infinity. The probability space is the product space of the random graph and the consensus process. In particular, this implies that running the simulation once is not substantially slower than

[4] These specific formulas assume that we are dealing with a synchronous case. The asynchronous case would have the probability be $\frac{n-1}{n} + \frac{1}{n}\sum_{S_t(w)\in\{0,1\}} H(v, w)$ when $S_t(v) = 1$ and $\frac{1}{n}\sum_{S_t(w)=1} H(v, w)$ when $S_t(v) = 2$. The equation that we need to verify stays unchanged after simple manipulation.

[5] Note that the expression has to hold for the case where there is a single blue and red nodes which shows one side. The other side is similarly simple

Algorithm 1 Estimating the Probability of Red Consensus

1: **Simulate** the process until all nodes are gnostic.
2: **Apply** Proposition 2.1 once all nodes are gnostic to obtain a probability of red consensus p.
3: **Repeat** the simulation multiple times and compute the average of the probabilities of red consensus p from each run as an unbiased estimate of the true probability.

computing the terms $\mu(v)$ (at least on the worst case). In Section 5, we present experiments showing that only a few runs (typically less than a 200) are required to obtain a good estimate (standard error below 0.01) of the consensus probability. In particular, not only running the process until all nodes are gnostic is much faster than waiting until consensus is reached, the error of the estimate obtained from multiple runs is also much lower when we end the process at the point where all vertices are gnostic.

4 CONVERGENCE TIME BOUNDS FOR CONSENSUS

In addition to the question of what are the probabilities of achieving consensus with a particular colour, the community has also focused on determining the time it takes for consensus to be achieved on general graphs. Lemma 4.1 shows that, when agnostic vertices are present, expected consensus times are bounded by the expected time it takes for agnostic vertices to disappear plus whatever bounds one can prove for the classical voter model.

Lemma 4.1 (Time bounds for consensus). *Given a graph G and a (pull) voter model on G with agnostic states (either synchronous or asynchronous). Let T_c denote the time it takes for consensus to be achieved and let T_a denote the time it takes for the agnostic vertices to disappear. Let S_0 be the starting configuration. Let $f(n)$ be a bound on the expected time of consensus being reached on G in the classical voter model (without any agnostic vertices) and any initial configuration. Then $\mathbb{E}(T_c|S_0) \leq \mathbb{E}(T_a|S_0) + f(n)$.*

Proof. For proof, see extended version of the paper [22]. □

Note that Lemma 4.1 is only useful for graphs G in which consensus is guaranteed. Otherwise, $f(n)$ may be infinite, in which case, although true, the result does not give us any useful information. As a result of Lemma 4.1, all bounds which hold for the standard voter model will also hold for the case where agnostic vertices are present with an extra time for the agnostic vertices to disappear. In practice, the time for agnostic vertices to become gnostic is typically much shorter than the time it takes to reach consensus as a simple consequence of rumour spreading results. Indeed, it is easy to see that the process of agnostic vertices disappearing is analogous to that of standard rumour spreading. However, the typical results in the literature for the rumour spreading process deal only with the push model, or sometimes the push-pull model. We, on the other hand, want results for the pull model. This is only a minor inconvenience though, as the proof ideas from previous works can also be used for the pull model. Proposition 4.2 shows that for general graphs, the expected time it takes in the pull model for the agnostic vertices to disappear is $O(n^2 \log(n))$. Observe that a star graph, with a gnostic node not on the center, as well as a path, would take an expected time of order n^2 for the agnostic vertices to become gnostic. Thus, Proposition 4.2 is almost tight. We defer the proof of Proposition 4.2 to the Appendix as it is just reusing known ideas from the rumour spreading model for the push model in the pull case.

Proposition 4.2 (rumour spreading bounds for general graphs). *Given a regular graph G (potentially with loops) and a (pull) voter model with agnostic states (either synchronous or asynchronous), where the vertices choose their neighbours (including potentially itself) with equal probability. Let T_a denote the number of rounds it takes for the agnostic vertices to disappear from the graph. Then $\mathbb{E}[T_a] = O(f(n))$ where $f(n) = n\log(n)$ for the synchronous case and $f(n) = n^2\log(n)$ for the asynchronous case.*

While Proposition 4.2 gives a bound for general graphs, it is likely that for a typical graph, the rumour spreading process is significantly faster. Indeed, it has been shown by [20, 39] that in the case of random graphs, the time it takes for agnostic vertices to disappear is of order $n\log(n)$. As their results are for the push model, we state Proposition 4.3 which is for the pull model. We again defer the proof to the appendix as we are just reusing previously known ideas for the push model of the rumour spreading process in the pull case.

Proposition 4.3 (rumour spreading bounds for random graphs). *Let $p >> \log(n)/n$ and $G \sim G(n,p)$ be an Erdös Rényi random graph. Consider a (pull) voter model on G with agnostic states (either synchronous or asynchronous). Let T_a denote the number of rounds it takes for the agnostic vertices to disappear. Then, with high probability we have that $\mathbb{E}[T_a] = O(f(n)\log(n))$ where $f(n) = 1$ in the synchronous case and $f(n) = n$ in the asynchronous case.*

Lastly, as a consequence of the above results, one can easily see that a single run of our Markov Chain Monte Carlo algorithm is relatively fast, as it is the time to simulate the process until no agnostic vertex remains plus the computation of the influences $\mu(v)$ (which only needs to be done once for the graph). Apart from the number of runs necessary to obtain a good estimate (which is analyzed in Section 5), both propositions show that a single run of the MCMC algorithm proposed takes time $O(n^2\log(n))$ in the worst case and $O(n\log(n))$ in the typical case which means MCMC is an efficient way of estimating the probabilities of consensus.

5 EXPERIMENTAL ANALYSIS WITH MCMC

Our key approach is to estimate probabilities by using MCMC, performing the simulations only until the point where all nodes are gnostic and using Proposition 2.1 to get probability values that can later be averaged over many runs to obtain an unbiased estimator for the consensus probability of a certain colour. Results from the rumour spreading theory ensure that each single run is fast. In this section, we aim to argue that not many runs are necessary to obtain good probability estimates by making a few experiments for certain graph families.

Before we describe the experiments in detail, note that an upper bound on the number of experiments required can be trivially obtained by considering the case where we do not stop the algorithm when all nodes become gnostic but instead go all the way until consensus is reached. This will give us a series of zeros and ones (corresponding to the target colour being the consensus one or not), which can then be averaged to get a probability. It is easy to see that each run is distributed as a Bernoulli and therefore their sum is distributed as a Binomial with the number of iterations it and the target consensus probability p as parameters. The average will then have variance equal to $\frac{p(1-p)}{it}$ which implies that the standard error (which is a proxy of the true error in the estimate) will scale as the inverse of the square root of the number of iterations. It is also easy to see that this approach of running the algorithm until consensus is reached gives worse estimates than the one where we stop the algorithm when agnostic nodes disappear.

Our first experiment will show that using our approach is substantially better than the error estimate from the previous paragraph. We run our algorithm for cliques and cycles with 1001 nodes for a varying number of runs and plot the standard error $\sigma_{\bar{x}}$ of the estimate for it.[6] We contrast those results with the standard error of the algorithm, where each run goes all the way until consensus is reached (described in the previous paragraph). Observe that Figure 4 shows that the error rates are substantially lower for the case where we use our algorithm versus the case where one runs until consensus is reached [7]. In fact, even as little as 40 runs are enough to be below 0.01 error in both cases.

Figure 4 also contains an additional line showing the standard error of our algorithm using a connected subgraph with 1001 nodes and 1925 edges of the graph representing a social network in Slovakia (Pokec from [42]). This subgraph was generated by selecting an initial random vertex v and performing a random walk (depth first search) until 1001 vertices were selected. Naturally, the line corresponding to the algorithm where we wait for consensus is very close to the result for cliques and cycles as should be expected (since it approximates the standard deviation of a Binomial distribution with number of iterations and target probability p as parameters divided by the number of iterations). Moreover, note that our algorithm is again substantially better than waiting until consensus is reached, just like for cliques and cycles, suggesting that our approach is also very effective for a typical social network.

As an additional experiment, we wonder whether there is a positive effect of increasing the graph sizes on $\sigma_{\bar{x}}$. It turns out that the answer is yes as Figure 5 shows. There, we run our algorithm for graph sizes varying from 300 to 3000, with a fixed number of iterations (400).[8] We perform the experiment on cliques and cycles again and vary the proportion of gnostic nodes as well.

Lastly, we add that we also performed both experiments on Erdős-Rényi random graphs with the same values of n and $p = 0.05$ and found that the $\sigma_{\bar{x}}$ line for the random graphs essentially stays superposed with the complete graph so we did not find it necessary to include it in the figure to avoid pollution.

For the code repository for these experiments, see [21] or access https://github.com/tmadeira/vmmrs.

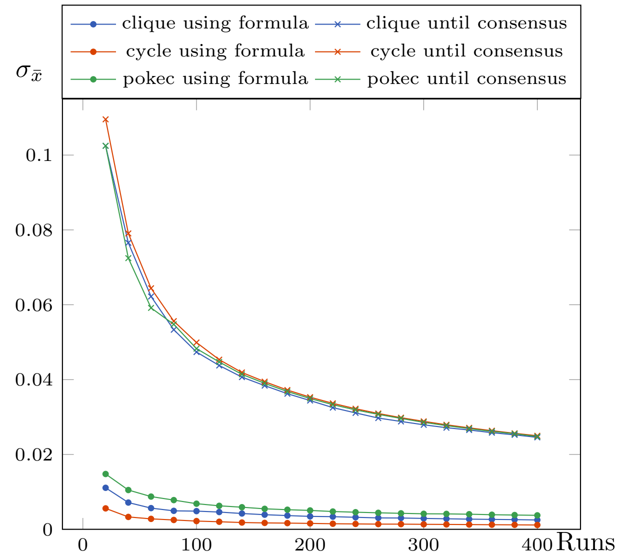


Figure 4: A comparison of the cumulative standard error of the probability of red consensus after all nodes are gnostic and the actual consensus until the simulation finishes. Each simulation ran 400 times on cliques, cycles and a connected subgraph of the Pokec social network with 1001 nodes (5% red, 5% blue, 90% agnostic). For cycles, the initial configuration has all red nodes side by side follow by all blue nodes side by side. For the Pokec subgraph, red and blue nodes were assigned at random, with the rest being agnostic.

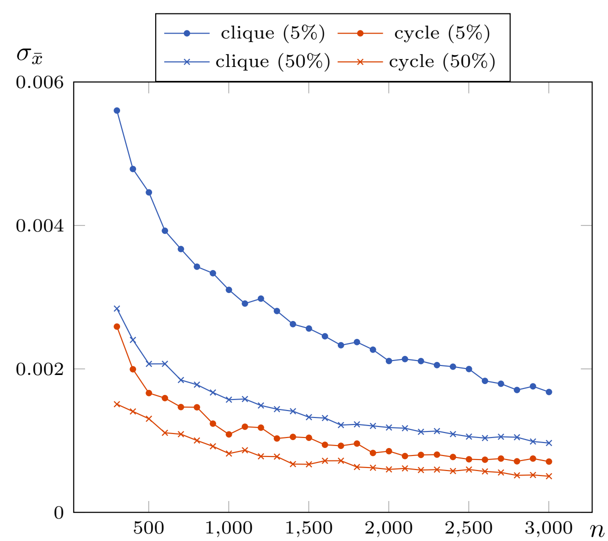


Figure 5: A comparison of the standard error of the probability of red consensus running the simulation 400 times for cliques and cycles of different sizes starting with different proportions of gnostic nodes (5%, 50%). Again, the initial configuration for cycles has all red nodes side by side follow by all blue nodes side by side.

[6] Note that having an odd number of nodes on cycles guarantees the process always converges

[7] Note that, additionally, each individual run takes substantially longer to complete if we wait until consensus is reached

[8] Using less than 400 iterations yields a graph with more variance as expected, but the same type of decay.

6 DISCUSSION

For simplicity, in this work, we focused on the pull model as that allows us to analyse both synchronous and asynchronous protocols. However, it is natural to wonder what happens when other strategies are used to transmit information. While synchronous push does not make sense for the voter model, asynchronous push and asynchronous push-and-pull could be used and Proposition 2.1 shows that the martingale for the voter model holds for asynchronous push. It is likely possible to use our martingale from Theorem 3.1 for those strategies as well, requiring only a little extra work, much like the distinction between the asynchronous and synchronous pull protocols.

Much more importantly though, the Markov chain Monte Carlo method of estimating the probabilities of consensus would function in exactly the same way. We could make use of the many known results from the rumour spreading literature to give us guarantees of fast runtime of a single run of the MCMC algorithm on many different types of graphs for those protocols, such as: bounds for the case of general graphs [1, 19], random graphs [1, 20, 39], preferential attachment graphs [17], graphs with good conductance [9] and social networks [10]. In general, our work implies that one can efficiently estimate probabilities of consensus for a given colour even in the case where agnostic nodes are present as long as a protocol that allows for fast rumour spreading is used.

7 CONCLUSION AND FUTURE WORK

Here, we introduce a variant of the voter model in which nodes can be agnostic, i.e., have no opinion or colour. Once gnostic, nodes cannot return to being agnostic. This can therefore been seen as a merge between two well-studied processes: the classical voter model and the rumour spreading process. Our approach allows for efficient estimation of the consensus probabilities for many different information transmission protocols (such as, synchronous pull, asynchronous pull, asynchronous push and asynchronous push-and-pull). We also provide a martingale akin to the one from the classical voter model, and use it to compute exact probabilities of consensus for complete graphs, and initial configurations in general graphs but where there are no edges between agnostic nodes.

In future work, we consider attempting exact computation of the consensus probabilities for other graph families, like d-regular graphs. Moreover, as observed in Section 8 there is an information transmission protocol that involves nodes transitioning into an undecided state. That protocol has the advantage of guaranteeing (with probability converging to 1) that a majority opinion achieves consensus. We find it an interesting question whether the results with that protocol also hold in the case where agnostic nodes are present and how the agnostic nodes influence the consensus times. Additionally, it may be of interest to study continuous consensus protocols [33] in the presence of initially agnostic processes.

8 RELATED WORK

The notion of reversibility in the context of the voter model has previously been used by Hassin & Peleg [24] to provide winning probabilities on a class of dynamic networks called 'stabilising dynamic graphs'. In these networks, until a given round, edges may disconnect and nodes attempting to copy a disconnected neighbour keep their own colour, similarly to gnostic nodes choosing agnostic ones in our protocol. The authors showed that their previous results hold for networks with reversible Markov chains, i.e. the total influence of the nodes of a given colour remains a martingale in the new process.

Several works studied agents that can be 'undecided' as an intermediate state, with nodes transitioning to this state when they select a differently coloured neighbour ([2, 3, 11, 40]). For a complete graph with binary opinions, the synchronous variant of this protocol has been shown to converge to the most common (plurality) colour in $O(\log n)$ rounds with high probability, assuming there is an initial difference of $\Omega(\sqrt{n \log n})$ in the numbers of agents with each colour [12]. Similar results have been obtained for the asynchronous protocol in the context of chemical reaction networks [13]. Additionally, for the consensus problem with $k > 2$ opinions, Becchetti et al. defined a 'monochromatic distance' function which measures the distance between any colour configuration and consensus, and used this to bound the convergence time of the synchronous process by $O(k \log n)$ [6].

On the other hand, Demers et al. proposed rumour-spreading protocols to aid the maintenance of distributed databases; these include push, pull, and push-pull transmissions [16]. For the synchronous push model, it is known that the number of rounds required to broadcast the rumour to all nodes is at most $O(n \log n)$, which is tight for the star graph [19]. This process has also been analysed for several other topologies, including complete graphs, hypercubes, bounded-degree graphs, and random graphs [19].

Our model also has similarities to the biased voter model proposed in [7], where one colour (corresponding to the agnostic state) has a bias of 0. However, their results do not apply in our setting since they assume that one colour has a strictly higher preference than all other colours. See also [28] for the biased voter model with 2 opinions in the continuous-time model. Our work is also related to [45]. In it, the authors study the evolution of a process with agnostic nodes, where the key difference is that gnostic nodes can never change colour.

Previous works on opinion diffusion have also studied related concepts to agnostic nodes, such as stubbornness. Those are mostly in the context of the majority model [4, 38] and the related Friedkin-Johnson model [41, 43]. The main difference between these models and ours is that the process dynamics is deterministic in the majority and Friedkin-Johnson models, whereas in the voter model, the process dynamics are randomised.

9 CODE REPOSITORY AND EXTENDED PAPER

For the code repository for these experiments, see [21] or access https://github.com/tmadeira/vmmrs. For the extended version of the paper with complete proofs and additional experiments, see [22].

ACKNOWLEDGMENTS

This work was supported by FAPESP grant number 2022/16374-6 (MMG) and EPSRC grant EP/W005573/1 (FMT).

REFERENCES

[1] Huseyin Acan, Andrea Collevecchio, Abbas Mehrabian, and Nick Wormald. 2015. On the push&pull protocol for rumour spreading. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*. ACM, 405–412.

[2] Talley Amir, James Aspnes, Petra Berenbrink, Felix Biermeier, Christopher Hahn, Dominik Kaaser, and John Lazarsfeld. 2023. Fast Convergence of k-Opinion Undecided State Dynamics in the Population Protocol Model. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing, PODC 2023, Orlando, FL, USA, June 19-23, 2023*, Rotem Oshman, Alexandre Nolin, Magnús M. Halldórsson, and Alkida Balliu (Eds.). ACM, 13–23. https://doi.org/10.1145/3583668.3594589

[3] Dana Angluin, James Aspnes, and David Eisenstat. 2008. A simple population protocol for fast robust approximate majority. *Distributed Computing* 21, 2 (2008), 87–102.

[4] Vincenzo Auletta, Ioannis Caragiannis, Diodato Ferraioli, Clemente Galdi, and Giuseppe Persiano. 2017. Information retention in heterogeneous majority dynamics. In *Web and Internet Economics: 13th International Conference, WINE 2017, Bangalore, India, December 17–20, 2017, Proceedings 13*. Springer, 30–43.

[5] Luca Becchetti, Vincenzo Bonifaci, and Emanuele Natale. 2018. Pooling or Sampling: Collective Dynamics for Electrical Flow Estimation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, 1576–1584.

[6] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, and R. Silvestri. 2015. Plurality consensus in the gossip model. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms* (San Diego, California) *(SODA '15)*. Society for Industrial and Applied Mathematics, USA, 371–390.

[7] Petra Berenbrink, George Giakkoupis, Anne-Marie Kermarrec, and Frederik Mallmann-Trenn. 2016. Bounds on the Voter Model in Dynamic Networks. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy (LIPIcs, Vol. 55)*, Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 146:1–146:15. https://doi.org/10.4230/LIPICS.ICALP.2016.146

[8] Mengtao Cao, Feng Xiao, and Long Wang. 2015. Event-based second-order consensus control for multi-agent systems via synchronous periodic event detection. *IEEE Trans. Automat. Control* 60, 9 (2015), 2452–2457.

[9] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2010. Rumour spreading and graph conductance. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, SIAM, 1657–1663.

[10] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2011. Rumor spreading in social networks. *Theoretical Computer Science* 412, 24 (2011), 2602–2610.

[11] Andrea Clementi, Mohsen Ghaffari, Luciano Gualà, Emanuele Natale, Francesco Pasquale, and Giacomo Scornavacca. 2018. A Tight Analysis of the Parallel Undecided-State Dynamics with Two Colors. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 117)*, Igor Potapov, Paul Spirakis, and James Worrell (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 28:1–28:15. https://doi.org/10.4230/LIPIcs.MFCS.2018.28

[12] Andrea Clementi, Mohsen Ghaffari, Luciano Gualà, Emanuele Natale, Francesco Pasquale, and Giacomo Scornavacca. 2018. A Tight Analysis of the Parallel Undecided-State Dynamics with Two Colors. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 117)*, Igor Potapov, Paul Spirakis, and James Worrell (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 28:1–28:15. https://doi.org/10.4230/LIPIcs.MFCS.2018.28

[13] Anne Condon, Monir Hajiaghayi, David Kirkpatrick, and Ján Maňuch. 2020. Approximate majority analyses using tri-molecular chemical reaction networks. *Natural Computing* 19, 1 (01 Mar 2020), 249–270. https://doi.org/10.1007/s11047-019-09756-4

[14] Colin Cooper and Nicolás Rivera. 2016. The Linear Voting Model. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 55)*, Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 144:1–144:12. https://doi.org/10.4230/LIPIcs.ICALP.2016.144

[15] Emilio Cruciani, Emanuele Natale, André Nusser, and Giacomo Scornavacca. 2021. Phase transition of the 2-Choices dynamics on core–periphery networks. *Distributed Computing* 34, 3 (2021), 207–225.

[16] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. 1987. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing* (Vancouver, British Columbia, Canada) *(PODC '87)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/41840.41841

[17] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. 2012. Asynchronous rumor spreading in preferential attachment graphs. In *Scandinavian Workshop on Algorithm Theory*. Springer, Springer, 307–315.

[18] Peter Donnelly and Dominic Welsh. 1983. Finite particle systems and infection models. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 94. Cambridge University Press, Cambridge University Press, 167–182.

[19] Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. 1990. Randomized broadcast in networks. *Random Structures & Algorithms* 1, 4 (1990), 447–460. https://doi.org/10.1002/rsa.3240010406

[20] Nikolaos Fountoulakis, Anna Huber, and Konstantinos Panagiotou. 2010. Reliable broadcasting in random networks and the effect of density. In *2010 Proceedings IEEE INFOCOM*. IEEE, IEEE, 1–9.

[21] Marcelo Matheus Gauy, Anna Abramishvili, Eduardo Colli, Tiago Madeira, Frederik Mallmann-Trenn, Vinícius Franco Vasconcelos, and David Kohan Marzagão. 2025. Voter Model Meets Rumour Spreading. https://github.com/tmadeira/vmmrs. GitHub repository, accessed: February 20, 2025.

[22] Marcelo Matheus Gauy, Anna Abramishvili, Eduardo Colli, Tiago Madeira, Frederik Mallmann-Trenn, Vinícius Franco Vasconcelos, and David Kohan Marzagão. 2025. Voter Model Meets Rumour Spreading: A Study of Consensus Protocols on Graphs with Agnostic Nodes [Extended Version]. *arXiv preprint* (2025).

[23] Geoffrey Grimmett, Geoffrey R Grimmett, David Stirzaker, et al. 2001. *Probability and Random Processes*. Oxford University Press.

[24] Yehuda Hassin and David Peleg. 2001. Distributed Probabilistic Polling and Applications to Proportionate Agreement. *Information and Computation* 171, 2 (2001), 248–268. https://doi.org/10.1006/inco.2001.3088

[25] Zool Hilmi Ismail and Nohaidda Sariff. 2018. A survey and analysis of cooperative multi-agent robot systems: challenges and directions. In *Applications of Mobile Robots*. IntechOpen.

[26] Varun Kanade, Frederik Mallmann-Trenn, and Thomas Sauerwald. 2023. On Coalescence Time in Graphs: When Is Coalescing as Fast as Meeting? *ACM Trans. Algorithms* 19, 2 (2023), 18:1–18:46. https://doi.org/10.1145/3576900

[27] David Kohan Marzagão, Nicolás Rivera, Colin Cooper, Peter McBurney, and Kathleen Steinhöfel. 2017. Multi-agent flag coordination games.. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, 1442–1450.

[28] N. Lanchier and C. Neuhauser. 2007. Voter Model and Biased Voter Model in Heterogeneous Environments. *Journal of Applied Probability* 44, 3 (2007), 770–787. https://doi.org/10.1239/jap/1189717544

[29] Erez Lieberman, Christoph Hauert, and Martin A Nowak. 2005. Evolutionary dynamics on graphs. *Nature* 433, 7023 (2005), 312.

[30] Nancy A Lynch. 1996. *Distributed algorithms*. Elsevier.

[31] Sonia Martinez, Francesco Bullo, Jorge Cortes, and Emilio Frazzoli. 2005. On synchronous robotic networks Part I: Models, tasks and complexity notions. In *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, IEEE, 2847–2852.

[32] David Kohan Marzagão, Luciana Basualdo Bonatto, Tiago Madeira, Marcelo Matheus Gauy, and Peter McBurney. 2021. The Influence of Memory in Multi-Agent Consensus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. AAAI Press, 11254–11262.

[33] Tal Mizrahi and Yoram Moses. 2008. Continuous consensus via common knowledge. *Distributed Computing* 20, 5 (2008), 305–321.

[34] Patrick Alfred Pierce Moran. 1958. Random processes in genetics. In *Mathematical Proceedings of the Cambridge Philosophical Society*. Cambridge University Press, Cambridge University Press, 60–71.

[35] Toshio Nakata, Hiroshi Imahayashi, and Masafumi Yamashita. 1999. Probabilistic local majority voting for the agreement problem on finite graphs. In *International Computing and Combinatorics Conference*. Springer, Springer, 330–338.

[36] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. 2007. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* 95, 1 (2007), 215–233.

[37] Roberto I. Oliveira and Yuval Peres. 2019. Random walks on graphs: new bounds on hitting, meeting, coalescing and returning. In *Proceedings of the Sixteenth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2019, San Diego, CA, USA, January 6, 2019*, Marni Mishna and J. Ian Munro (Eds.). SIAM, 119–126. https://doi.org/10.1137/1.9781611975505.13

[38] Charlotte Out and Ahad N Zehmakan. 2021. Majority vote in social networks: Make random friends or be stubborn to overpower elites. *arXiv preprint arXiv:2109.14265* (2021).

[39] Konstantinos Panagiotou and Leo Speidel. 2017. Asynchronous rumor spreading on random graphs. *Algorithmica* 78 (2017), 968–989.

[40] Etienne Perron, Dinkar Vasudevan, and Milan Vojnovic. 2009. Using three states for binary consensus on complete graphs. In *IEEE INFOCOM 2009*. IEEE, 2527–2535.

[41] Mohammad Shirzadi and Ahad N Zehmakan. 2024. Do stubborn users always cause more polarization and disagreement? a mathematical study. *arXiv preprint arXiv:2410.22577* (2024).

[42] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International scientific conference and international workshop present day trends*

of innovations, Vol. 1.

[43] Wanyue Xu, Liwang Zhu, Jiale Guan, Zuobai Zhang, and Zhongzhi Zhang. 2022. Effects of stubbornness on opinion dynamics. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM, 2321–2330.

[44] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. 2013. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems* 10, 12 (2013), 399.

[45] Ahad N. Zehmakan, Xiaotian Zhou, and Zhongzhi Zhang. 2024. Viral Marketing in Social Networks with Competing Products. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2047–2056.