Voting Models in Random Networks

Mehmet E. Yildiz
Cornell University
mey7@cornell.edu

Roberto Pagliari
Cornell University
rp294@cornell.edu

Asuman Ozdaglar
MIT
asuman@mit.edu

Anna Scaglione
UC Davis
anna@ucdavis.edu

***Abstract*—A crucial problem of Social Sciences is under what conditions agreement, or disagreement, emerge in a network of interacting agents. This topic has application in many contexts, including business and marketing decisions, with potential impact on information and technological networks. In this paper we consider a particular model of interaction between a group of individuals connected through a network of acquaintances.**

In the first model, a node waits an exponentially time with parameter one, and when it expires it chooses one of its neighbors' at random and adopts its decision. In the second one, the node chooses the opinion which is the most adopted by its neighbors (hence, *majority rule*). We show how different updating rule of the agent' state lead to different emerging patterns, namely, agreement and disagreement. In addition, in the case of agreement, we provide bounds on the time to convergence for various types of graphs.

I. Introduction

Networks of interacting agents play a fundamental role in modeling physical structures as well as social, information, technological and biological phenomena. In Social Sciences, a considerable effort has been made in deriving models aimed at describing the statistical properties and emergence of large networks, from both an empirical and a theoretical point of view. One of the first studies in this arena, by Watts and Strogatz [1], showcased common features observed in neural networks (of the worm *Caenorhabditis Elegans*), man-made networks (power grids), and social networks (film actors). The similarities in the characteristic path length, clustering coefficient (network topology), and the existence of critical points for cascading behavior are not a coincidence. Such properties, along with the degree distribution and maximum degree, the maximum separation among nodes and many other features, are strictly related to these systems growth mechanisms over time (see [2] and references therein). Some of the implications of such models have been studied in the context of routing data over large networks [3], and to analyze the diffusion of innovations in social networks [4], [5], [6]. These insights can potentially have implications on economics issues related to technological networks, and marketing of new products as well of ideas. For example, suppose that a group of individuals interconnected together by a network of acquaintances[1] are using different cell phones. All different devices have different properties, and it is not clear whether one would be preferred over the other ones. Can social interaction eventually convince the society to reach a consensus on a single product? If so how fast is the process of convincing the others? Obviously, the answer depends on the way the individuals interact, which is the main issue of modeling in social sciences. In general, there is no consensus on what is the most appropriate choice, and to what level one can derive quantitative answers.

[1]Either in the real world, or via *virtual* connections, such as friends on Facebook and similar web-based services.

Contribution

In this paper, we consider a particular model of interaction, known as *the voter algorithm* [7], in which a node in the network chooses to adopt the same behavior of one of its neighbors taken uniformly at random in the neighborhood. We focus on the case where the possible behaviors form a discrete set. We note that in the original voter algorithm, there are only two possible behaviors, *i.e.*, binary opinions [**?**]. In that sense, our analysis is more general since we do not make that particular assumption. We prove convergence to a consensus, and provide an upper-bound on the time to convergence in both deterministic and random topologies. We note that it has been already proved that the voter algorithm converges to a network wide consensus in finite networks and $1-2$ dimensional infinite lattices [7], [8]. Moreover, similar algorithms with uncountable behavior sets have also been studied in the literature.

We then compare this interaction approach with another model: this is the same as the previous one, except for the fact that a node chooses the opinion corresponding to the most adopted in its neighborhood (*majority rule*). We show that the outcome changes completely: in particular, the emerging pattern is the division of the nodes according to the communities they belong to rather than consensus. We report an example performed on the European Power Grid.

II. Voting on a Graph: Agreement

We consider a network $G(\mathcal{V},\mathcal{E})$ where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of edges. We focus on undirected simple graphs, *i.e.*, no self-loops and multiple edges. Thus, the neighborhood of a node in the network, say node $i \in \mathcal{V}$, is defined as $\mathcal{N}_i = \{j|(i,j) \in \mathcal{E}\}$. The degree of node i is then the cardinality of the set $\mathcal{N}_i$, that we indicate with $d_i \equiv |\mathcal{N}_i|$. The degree matrix $\mathbf{D}$ is a diagonal matrix whose i-th element is the degree of node i, i.e., $D_{i,i} = d_i$. The adjacency matrix $\mathbf{A}$ is a matrix that contains information about the connections among the nodes. Specifically, $A_{i,j} = A_{j,i} = 1$ if there is a link between nodes i and j, and zero otherwise.

A. General model

The algorithms we study are asynchronous. Each node schedule the next update of its own opinion according to a Poisson process with rate 1. Therefore, we can use a discrete index k to denote the updating events (each of which involve only one node) and follow the evolution of the system at discrete times.

The state of a node is indicated by $c_i[k]$, where i is the node index. Opinions, or nodes' states, are chosen among a finite set of elements. In particular, we assume that initially, at iteration $k = 0$, all nodes have different opinions that we can think of as colors. Hence, node 1 is colored with 1, node 2 with 2 and so on and so forth, so that the initial set of opinions is $\mathcal{S} = \{1, 2, \ldots, N\}$. Of note is that the voter model with binary opinions is a special of this setup since in that particular case $\mathcal{S} = \{1, 2\}$. Our case is the most general one since there may be at most N different opinions in a network with N agents. We use an index function $\mathcal{I} : \mathbb{N}^+ \to \mathcal{V}$ that associates any iteration k to the nodes currently performing the update.

We formulate the algorithm in the most general case. At iteration $k+1$, node $\mathcal{I}[k+1]$ updates its state as follows:

$$c_{\mathcal{I}[k+1]}[k+1] = f(c_{i,1}[k], c_{i,2}[k], \ldots, c_{i,d_i}[k]) \qquad \text{(II.1)}$$

where $f(\cdot)$ is the updating function (equal for all nodes), and $c_{i,j}[k]$ is the color (at iteration k) of the jth neighbor of node i. The other nodes, instead, do not change their opinion, i.e., $c_j[k+1] = c_j[k]$ for all $j \neq \mathcal{I}[k+1]$. In the rest of the paper we assume that the following conditions hold true:

1) The number of nodes in the network $|\mathcal{V}| = N$ is finite.
2) The underlying topology G is strongly connected, *i.e.*, there exists a path (not necessarily single hop) between any given node pairs.

B. Randomization and Agreement

The algorithm we consider for reaching agreement is as follows. Assume node i wakes up at iteration k, i.e., $\mathcal{I}[k] = i$. Let us define as $Q_{i,c}[k]$, the number of neighbors of node i colored with c at iteration k. The updating rule used by node i, as defined in II.1, becomes:

$$c_i[k+1] = X, \; X \sim \frac{1}{d_i} \sum_{c \in \{1,2,\ldots,N\}} Q_{i,c}[k] \delta(x - c) \qquad \text{(II.2)}$$

and $c_j[k+1] = c_j[k]$ for all $j \neq i$. Basically, node i constructs a histogram where for each colors there are as many units as the number of its neighbors currently adopting that color, and chooses the new state based on the probability mass function induced by the histogram. We note that (II.2) is equivalent to saying that node i copies the opinion of one of its neighbors uniformly at random, i.e.

$$c_i[k+1] = X, \; X \sim \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \delta(x - c_j[k]).$$

The algorithm constructed above is called the voter algorithm.

C. Convergence Properties

Since the the updating sequence is random, and the evolution only depends on the most recent update, the entire system can be viewed as a Markov chain. The state of the network is then given by an N-dimensional vector, given by the nodes' states. Clearly, the condition where all nodes have the same color is an absorbing state, since a node cannot choose a color that does not appear in the network.

Theorem 2.1: Consider a connected network composed of N nodes performing the voter algorithm. For any iteration $k > 0$, condition

$$\Pr\left(c_i[k+N] = c_j[k+N], \; \forall i, j \in \{1, 2, \ldots, N\}\right) > 0$$

holds true.

Proof: We need to prove that there always exists at least one sequence of N consecutive updates leading to an absorbing state. We pick the node who performed the kth iteration and call it v. Consider the set of nodes defined by the following recursion

$$\mathcal{T}[k+l] = \mathcal{T}[k+l-1] \cup u \qquad \text{(II.3)}$$

subject to the constraint

$$u \notin \mathcal{T}[k+l-1] : \exists \; e(u, v), v \in \mathcal{T}[k+l-1]$$

with $l = 0, 1, \ldots, N-1$ and $\mathcal{T}[k] = \{v\}$. Equation (II.3) states that, at each step, we choose a node that has at least one edge connected to at least one node in $\mathcal{T}[k+l-1]$. If there are, let us say, $1 \leq m < N$ such nodes, the probability that this situation actually happens is $m/N \geq 1/N > 0$. At iteration $k+1$, the probability that the new node u, added to $\mathcal{T}[k] = \{v\}$, chooses the same color as v is at least $1/Q_u = (\sum_m Q_{um})^{-1} > 0$, where Q_u is the total number of colors used by node u. Conditioned to this event, the probability that the second node, say z, added to $\mathcal{T}[k+1] = \{v, u\}$, chooses the same color is at least $1/Q_z = (\sum_m Q_{zm})^{-1} > 0$, and so on and so forth, for $N-1$ times. Therefore, given any initial color of v, the probability that after $N-1$ consecutive updates all nodes have that particular color is at least

$$\prod_{u \in \mathcal{N} - \{v\}} \frac{1}{N Q_u} > 0$$

which is always positive, so long as $N < \infty$. Since node v and iteration k are arbitrary, the color of node v could be any color in the set $\mathcal{S}$, and, thus, the result follows. ■

We can now use the result provided by Theorem 2.1 to establish the following convergence result.

Theorem 2.2: Consider a connected network composed of N nodes performing the voter algorithm, with initial conditions $\{c_1[0], c_2[0], \ldots, c_N[0]\}$ and $c_i[0] \in \mathcal{S}, \forall i$. The event

$$\exists \; k, c \in \mathcal{S} : \; \forall k' > k, \; c_i[k'] = c \; \forall i$$

occurs almost surely.

Proof: As described above, the state of the network is given by an N-dimensional vector containing the n colors of the nodes. Since this process is a Markov chain, and

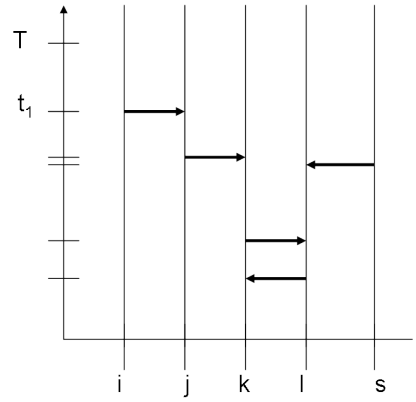

Fig. 1. A sample update path for the voter algorithm.

the absorbing set is a condition where all nodes have the same color, a sufficient condition for convergence is that there always exists a strictly positive probability of reaching the absorbing set. From Theorem 2.1 we know that this situation always occurs, and, therefore, the result follows. ■

III. Bounds on the Time to Convergence

One of the major approaches in the analysis of the voter model is so called the *dual approach*, where the voting problem is being mapped into *coalescing random walks* on the graph G. While the reader may refer to [7] for detailed discussions and proofs of the duality, we will discuss them briefly in the following for the sake of completeness and clarity of our proofs.

In the coalescing random walks process, there is a single particle at each vertex of the graph G at time $t = 0$. These particles perform independent continuous random walks on the graph, *i.e.*, each particle jumps to one of its neighbors independently according to a rate 1 Poisson process. When two or more particles meet on a vertex, they coalesce and form a single particle. Thereafter, these particles perform a single random walk on G, possible colliding with other particles.

Assuming nodes have been initialized by different types (opinions) in the voter model, the convergence of time of the voter model (to a consensus) has the same distribution as the convergence time of the coalescing random walk process (to a single particle). While reader may refer to [8], [7] for rigorous proofs, the idea is based on the following: for each vertex $i \in \mathcal{V}$ and time $T \geq 0$, one can define a dual process $\{y_i^T(s) : 0 \leq s \leq T\}$, which traces the origin of the opinion at node i at time $T \geq 0$. The primal and the dual processes will have the property:

$$c_i(T) = c_{y_i^T(T-s)}(s). \tag{III.1}$$

In other words, the origin of the opinion at node i at time $T \geq 0$, is the opinion at node $y_i^T(T-s)$ at time s. We note that $y_i^T(0) = i$, and the process $\{y_i^T(s) : 0 \leq s \leq T\}$ will jump according to the arrival times of the primal process $c_i(t)$. For instance, if $t_1 \leq T$ is the last time before (and including) T that node i has copied one of its neighbors decision (say neighbor j), then y_i^T will jump to the neighbor j at time $T - t_1$. A sample update path is given in Fig. 1. In this example, the x-axis denotes different nodes in the network, and the y-axis denotes continuous time interval. Whenever a node copies its neighbor's decision, it will be denoted by an arrow from the node itself to its neighbor. As we have discussed above, node i has copied node j's decision at time t_1, and there is a corresponding arrow in the figure. We note that one can track the origin of the node i's decision at time T by following arrows backwards in time, *i.e.*, by the dual process. In our example, node i's decision at time T is node k's initial decision.

Moreover, if we want to track the origins of opinions at more than one location, *i.e.*, at $\mathcal{B} \subset \mathcal{V}$, then we can define an independent dual process for each element $i \in \mathcal{B}$. The crucial point is that if two or more dual processes reside at the same location at the same time, they will collide and move together, since the origin of the decision at that point and time will be exactly the same for both processes. Thus, these dual processes form a coalescing random walk on G.

At this point, it should be clear to our readers that the dual process for the voter model is indeed a coalescing random walks process, and assuming nodes have been initialized by different types, voter model will converge to a consensus when coalescing random walks process initialized at all nodes on G collapses to a single particle.

The following result is an upper-bound of the expected convergence time $E[T_{\text{crw}}]$ for a coalescing random walk process on a given graph.

Lemma 3.1 (Aldous [8]): Assume $E_i[T_j]$ is the mean hitting time of a regular random walk to node j on graph G, given that the walk is initialized at node i. Then

$$E[T_{\text{crw}}] \leq e \log(N+2) \max_{i,j} E_i[T_j].$$

The mean hitting time is given by the following lemma.

Lemma 3.2 (Lovasz [9]): The mean hitting time of a regular random walk to node j on graph G, when the walk is initialized at node i, is equal to

$$E_i[T_j] = 2|\mathcal{E}| \sum_{k=2}^{N} \frac{1}{1-\lambda_k(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})} \left(\frac{v_{kj}^2}{D_{jj}} - \frac{v_{kj}v_{ki}}{\sqrt{D_{jj}D_{ii}}} \right)$$

where v_k is the eigenvector of $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ corresponding to eigenvalue λ_k.

Hence, based on Lemmas 3.1 and 3.2, we are able to bound the convergence time of the *voter model*.

Theorem 3.3: Given a network $G(\mathcal{V}, \mathcal{E})$, its adjacency and degree matrices A and D respectively, the expected time for the voter model to convergence is upper bounded by:

$$E[T] \leq \frac{4e \log(N+2)|\mathcal{E}|}{1-\lambda_2(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})} \max_j D_{jj}^{-1}. \tag{III.2}$$

Proof: Since the coalescing random walk process is the dual of the voter model, by Lemma 3.1,

$$E[T] = E[T_{crw}] \leq e \log(N+2) \max_{i,j} E_i T_j.$$

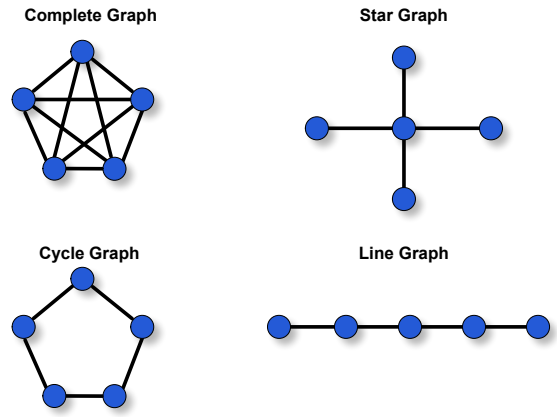

Fig. 2. The deterministic topologies we consider are 1) the complete graph, 2) the star graph, 3) the cycle graph and 4) the line graph.

Moreover, denoting $\mathbf{M} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and by Lemma 3.2,

$$\begin{aligned}
\max_{i,j} E_i T_j &= \max_{i,j} 2|\mathcal{E}| \sum_{k=2}^{N} \frac{1}{1-\lambda_k(\mathbf{M})} \left(\frac{v_{kj}^2}{D_{jj}} - \frac{v_{kj}v_{ki}}{\sqrt{D_{jj}D_{ii}}} \right) \\
&\leq^{(a)} \frac{2|\mathcal{E}|}{1-\lambda_2(\mathbf{M})} \max_{i,j} \sum_{k=2}^{N} \left(\frac{v_{kj}^2}{D_{jj}} - \frac{v_{kj}v_{ki}}{\sqrt{D_{jj}D_{ii}}} \right) \\
&\leq^{(b)} \frac{2|\mathcal{E}|}{1-\lambda_2(\mathbf{M})} \left(\max_j \frac{1}{D_{jj}} + \max_{i,j} \sum_{k=2}^{N} \frac{-v_{kj}v_{ki}}{\sqrt{D_{jj}D_{ii}}} \right), \\
&=^{(c)} \frac{2|\mathcal{E}|}{1-\lambda_2(\mathbf{M})} \left(\max_j \frac{1}{D_{jj}} + \max_{i,j} \frac{v_{1j}v_{1i}}{\sqrt{D_{jj}D_{ii}}} \right), \\
&\leq^{(d)} \frac{4|\mathcal{E}|}{1-\lambda_2(\mathbf{M})} \max_j \frac{1}{D_{jj}},
\end{aligned}$$

where (a) follows from the fact that $\lambda_2(\mathbf{M})$ is the second largest eigenvalue, (b) is due to $\sum_{k=1}^{N} v_{kj}^2 = 1$, (c) follows from $\sum_{k=1}^{N} v_{kj}v_{ki} = 0$ and $v_{1j} \geq 0$, and (d) is since $v_{1j}v_{1i} \leq 1$. If we combine the upper bound with Lemma 3.1, we obtain the final bound:

$$E[T] \leq \frac{4e\log(N+2)|\mathcal{E}|}{1-\lambda_2(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})} \max_j D_{jj}^{-1}$$

Finally, we note that $\lambda_k(.)$'s and v_k's are well defined since $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is a real symmetric matrix. ■

IV. Scaling Laws for Different Graphs

We now derive the scaling laws for various types of graphs, from deterministic graph topologies, to 2-dimensional random graphs, Erdos-Renyi and small-world graphs. We show how the connectivity impacts the final result on the time to convergence.

A. Deterministic Network Topologies

In the context of deterministic network topologies we consider five types of graph, namely, 1) the complete graph, 2) the star graph, 3) the cycle graph and 4) the line graph, sketched in Fig. 2. Consider, for example, the complete graph. In this case, the number of edges is equal to $|\mathcal{E}| = N(N-1)/2$. The degree is equal to $N-1$ for all nodes, and $1-\lambda_2(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = N/(N-1)$. By using the result in Theorem 3.3, we obtain

$$E[T] \leq e(N-1)\log(N+2).$$

In a similar fashion one can compute the bound on the time to convergence for the other cases, that are reported in Table I.

B. Random Geometric Graph

Let us consider the case of 2-dimensional random geometric graphs (which is the case of interest), where N nodes are randomly placed (w.l.o.g.) in a 1-by-1 square, and two nodes communicate if their distance is less than r. The following result holds true.

Theorem 4.1: Given a 2-dimensional random geometric graph with $r \geq \sqrt{\frac{4\log N}{N}}$, there exists a finite $N' \geq 0$ such that for all $N \geq N'$, the expected time to reach consensus is upper-bounded by

$$E[T] \leq \frac{2e\log(N+2)N}{r^2}. \tag{IV.1}$$

Proof: We first note that for $r \geq (4\log(N)/N)^{1/2}$ and large enough N, the graph will be connected with high probability (w.h.p.). Moreover, there exists $n' \geq 0$ such that for all $N \in [n', \infty)$ the degree of every node in the network is equal to some $\alpha(N)$ w.h.p. [10]. Thus, the ratio of $|\mathcal{E}|/\min_{jj} D_{jj}$ is equal to N w.h.p. Moreover, the term $[1-\lambda_2(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})]^{-1}$ upper bounded by is $1/r^2$ for large enough N [10]. Combining these results with Theorem 3.3, the result follows. ■

In order to see how the performances of this algorithm scale as the network grows, we may set the transmission radius to $r = 2\sqrt{\frac{\log N}{N}}$; that is, as the network becomes large, the connectivity remains constant [10]. By substitution, from (IV.4) we obtain

$$E[T] \leq \frac{e}{2}N^2 = O(N^2).$$

In Figure 3 we report, as an example, the evolution of the network state (the nodes' colors) over time, with a network composed of $N = 100$ nodes and normalized communication radius $r = 0.2$.

C. Erdos-Renyi Graphs

The Erdos-Renyi graph (also known as ER graph) is one of the most famous, and still used, random graph used to model social networks, because of its simplicity of analysis due to the

TABLE I
Time to Convergence

Graph	Bounds on Expected Time to Convergence
Complete Graph	$e(N-1)\log(N+2)$
Star Graph	$e(2N-2)\log(N+2)$
Cycle Graph	$eN^2\log(N+2)$
Line Graph	$e(N-1)^2\log(N+2)$

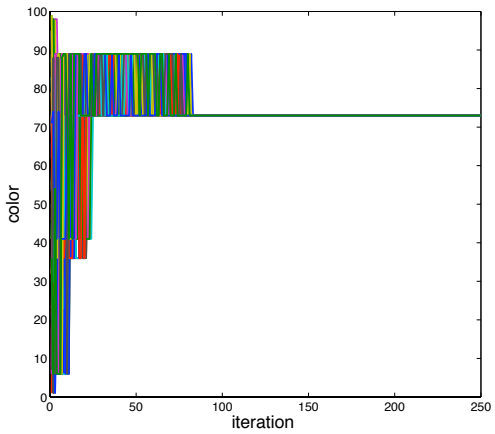

Fig. 3. Voting algorithm performed on a network composed of $N = 100$ nodes, with normalized transmission radius $r = 0.2$.

independence of the links between the nodes. The construction of an ER graph is simple: pick a set of N nodes with no edges. For each pair of nodes, out of $N(N-1)/2$ possible combinations, establish a link with probability p (coin flip). In this context, we wish to derive a bound of the time to convergence for the voter model when the network grows, but the connectivity remains constant. To accomplish this task we introduce a dependency of p on the network size N. In order to bound the convergence time for ER graphs, we first introduce the following result.

Lemma 4.2: Assume random variables $X_1, X_2, \ldots, X_N$ are i.i.d with $X_i \in \{0, 1\}$. Then,

$$P\left(\sum_{i=1}^{N} X_i > (1+\delta)NE[X_i]\right) < e^{\frac{NE[X_i]\delta^2}{3}}, \tag{IV.2}$$

$$P\left(\sum_{i=1}^{N} X_i < (1-\delta)NE[X_i]\right) < e^{\frac{NE[X_i]\delta^2}{2}}, \tag{IV.3}$$

where $0 < \delta < 1$.
Of note is that the above lemma gives exponentially decaying bounds on the tail distributions of sums of independent binary random variables.

For a given node i, the number of neighbors of that particular node is equal to:

$$d_i = \sum_{j=1}^{N} A_{ij}.$$

We note that above definition introduces self-loops, *i.e.*, each node is neighbor to itself. Such an assumption is for the mathematical brevity, and does not change the validity of our analysis. We note that due to the independent nature of the link generations, A_{ij}'s are i.i.d. binomial with mean p for a given i.

We can now apply the result of Theorem 3.3 to bound the expected time to consensus for ER graphs.

Theorem 4.3: Given an Erdos-Renyi graph with $p >> \log^2(N)/N$, the expected time to reach consensus scales as:

$$E[T] \sim O\left(\frac{4e\log(N+2)N\ (1+2\sqrt{\log(N)/Np})}{1-\frac{8}{Np}-\frac{g(n)\log^2(N)}{Np}}\right), \tag{IV.4}$$

where $g(n)$ is a function tending to infinity arbitrary slowly.

Proof: Using (IV.2), one can show that

$$P\left(d_i \leq Np + \delta Np\right) \geq 1 - e^{\frac{Np\delta^2}{3}}.$$

Moreover, if we choose $\delta = 3\log(N^2)/Np$:

$$P\left(d_i \leq Np + \sqrt{3Np\log(N)}\right) \geq 1 - \frac{1}{N^2}.$$

Since above equation holds for all $i \in \mathcal{V}$, then:

$$P\left(|\mathcal{E}| \leq N^2p + N\sqrt{3Np\log(N)}\right) \geq 1 - \frac{1}{N^2}. \tag{IV.5}$$

To bound $\max_j D_{jj}^{-1}$, we first note that $D_{jj} = |\mathcal{N}_j| = d_j$. Then, by (IV.3):

$$P\left(d_j \geq Np + \delta Np\right) \geq 1 - \frac{1}{N^2}.$$

By choosing $\delta = 2\log(N^2)/Np$:

$$P\left(\min_j D_{jj}^{-1} \leq \left(Np - \sqrt{2Np\log(N)}\right)^{-1}\right) \geq 1 - \frac{1}{N^2} \tag{IV.6}$$

To bound $1 - \lambda_2(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})$, we utilize Theorem 3.6 in [11]. By noting that $w_{min} = \bar{w} = np$ in our case and assuming $Np >> \log^2(N)$, the bound simplifies to:

$$|1 - \lambda_{N-1}(I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})| \leq (1+o(1))\frac{4}{Np} + g(n)\frac{\log^2(N)}{Np},$$

where $g(n)$ is an a function tending to infinity arbitrary slowly. Since $\lambda_2 = 1 - \lambda_{N-1}(I - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})$, we can bound the last term in III.2 as:

$$\frac{1}{1-\lambda_2(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})} \leq \left(1 - \frac{8}{Np} - \frac{g(n)\log^2(N)}{Np}\right)^{-1}. \tag{IV.7}$$

Combining (IV.5), (IV.6) and (IV.7), our result follows. ■

V. Voting on a Graph: Disagreement

As we have seen from the specifications of the model described in Section II, the nodes are always driven to consensus by means of local interactions. One might ask what happens if the updating rule is changed. After all, equation (II.3) simply means that the updating node copies the color of one of its neighbors, chosen at random. The vote is biased towards becoming the vote of the majority, but is not necessarily that of the majority. Hence, a straightforward modification of such dynamics is the majority rule, also known as Label Propagation Algorithm (LPA) [12]. This particular scheme is somewhat similar to the one we discussed above. Each node

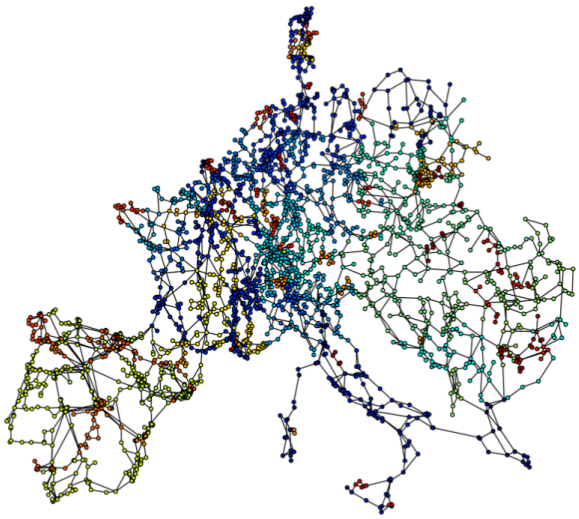

Fig. 4. Label Propagation Algorithm (LPA) applied to the European Power Grid.

wakes-up and collect the colors of its neighbors. Instead of picking one at random, the node chooses the color with the most number of occurrences (hence, *majority* rule). Suppose node i wakes up at iteration k. Recalling that $Q_{i,c}[k]$ is the number of neighbors of node i colored with c, the new state of node i is given by the following updating rule

$$c_i[k+1] = \arg\max_c Q_{i,c}[k]. \tag{V.1}$$

If two or more colors satisfy Eq. (V.1), node i chooses one of such colors at random. We can see that, as opposed to Eq. (II.3), the updating rule is deterministic, since the node chooses the opinion adopted by the majority of its neighbors. The LPA algorithm does not lead the nodes to consensus, nor (in general) guarantees convergence. However, an interesting pattern emerges as the node keep updating their colors using Eq. (V.1). The number of colors reduces, and nodes within a community end up sharing the same color. By community in a graph, we mean a subgraph of G such that the number of edges within the subgraph is much bigger than the number of edges connecting the subgraph to the rest of the network. Alternatively, one may define the concept of community in a graph as a region where the number of links is much bigger with respect to a reference graph (the so called *null model*), corresponding to what we would expect by adding edges between nodes at random [2].

In Figure 4 we show the result of the LPA algorithm applied to a large graph, the European Power Grid. We can see that "dense" regions of nodes are identified by one color. Although there are not many theoretical results about the properties of LPA, the algorithm seems to be working quite well with many graphs of interest. Potential applications of this scheme include self-organization for mobile robots and power control in wireless networks.

The LPA algorithm has been improved quite significantly since its discovery. In [13] an modified version of the LPA for finding communities of similar size has been proposed. In [14] the Authors designed an LPA-based algorithm to detect the extent to which a node belongs to one of the surrounding communities. An interesting feature offered by that alternative is this weighted and bipartite networks are handled as well. In [15] it has been shown that the LPA algorithm is equivalent to finding the local minima of a simple Potts model.

The Label Propagation Algorithm has also potential applications in the engineering panorama. This scheme can, in fact, be used for formation control problems. In this context, one of the main problems is keeping the network "well" connected over the time; this means that the formation of communities is not desirable. If the nodes perform the LPA scheme, some of them will know that they lie over the borders between different communities, allowing them to inform the others to get closer. The same idea can be employed to improve the connectivity of low-density networks.

VI. Conclusions

In this paper we have discussed two voting algorithms, both based on voting mechanisms. In one case, a node adopts the opinion of one of its neighbors; in the other, the majority rule is applied. Although both these two algorithms make use of local information only, their behavior is completely different, and this is determined by the specific updating rule they use. In the first case we have consensus: as the time evolves, the nodes are driven to the same color, meaning that they have the same opinion. In the second one there is no consensus, and we call this situation *disagreement*. The number of colors reduces, and they distribute in a way such that nodes belonging to the same community share the same opinion. These coloring games have potential applications in the engineering arena for coordinating mobile agents interconnected through a communication networks, to modeling collective behavior in human societies.

References

[1] D. Watts and S. Strogatz, "Collective dynamics of 'small world' networks," *Nature*, pp. 440–42, 1998.

[2] M. Newman, A.-L. Barbasi, and D. Watts, "The structure and dynamics of networks," *Princeton University Press*, 2006.

[3] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proc. 32nd ACM Symposium on Theory of Computing*, 2000.

[4] M. Granovetter, "Threshold models of collective behavior," *American Journal of Socioligy*.

[5] J. Goldberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, 2001.

[6] J. Kleinberg, "Cascading behavior in networks: Algorithmic and economic issues," in *Algorithmic Game Theory*. Cambridge University Press, 2007.

[7] T. M. Liggett, *Interacting Particle Systems*. Springer Berlin Heidelberg, 2005.

[8] D. Aldous, "Random walks on finite groups and rapidly mixing markov chains," in *Seminaire de Probabilities XVII*. Springer, 1983.

[9] L. Lovasz, "Random walks on graphs: A survey," *Combinatorics*, 1993.

[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *INFOCOM*, 2005.

[11] F. Chung, L. Lu, and V. Vu, "Spectra of random graphs with given expected degrees," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, May 2003.

[12] U. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *arXiv:0709.2938v1 [physics.soc-ph]*, 2007.

[13] X. Liu and T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," *arXiv:0910.1154v2*, 2009.

[14] S. Gregory, "Finding overlapping communities in networks by label propagation," *arXiv:0910.5516v1*, (Submitted) 2009.

[15] G. Tibely and J. Kertesz, "On the equivalence of the label propagation method of community detection and a potts model approach," *Physica A*, 2008.