Innovations in Computer Science 2010

Analytical Tools for Natural Algorithms

Bernard Chazelle
Princeton University, USA
chazelle@cs.princeton.edu

Abstract: We introduce an analytical tool to study the convergence of bidirectional multiagent agreement systems and use it to sharpen the analysis of various natural algorithms, including flocking, opinion consensus, and synchronization systems. We also improve classic bounds about colored random walks and discuss the usefulness of *algorithmic proofs.*

Keywords: Natural algorithms; algorithmic proofs; colored random walks; total s-energy.

1 Introduction

We introduce an analytical tool to study the convergence of certain multiagent agreement systems and use it to sharpen the analysis of various natural algorithms, including flocking, opinion consensus, and synchronization systems. We also improve classic bounds about colored random walks. Before we go into any of the details, we wish to explain the motivation behind this work.

Nonlinear dynamics counts as one of the great scientific advances of the last century: chaos, fractals, strange attractors, emergence, and "small worlds" have all been the focus of public attention and the ferment of first-rate science. As these subjects mature, however, the limitations of classical mathematics are being felt. No one today seriously believes that to understand ecologies, immune systems, markets, or social networks is just a matter of finding the right differential equations and shaping them into predictive tools. Mathematics thrives on symmetry and physics on invariance. This perfect match largely accounts for the amazing success of 20th-century science. But, after what Wigner called the "unreasonable effectiveness of mathematics in the natural sciences," we are beginning to witness its reasonable ineffectiveness in coping with complex systems short on symmetry and regularity.

This is where computer science comes in. Numerical simulations, data mining, machine learning, and other data-centric applications of computing have moved center stage. Yet the question remains: Can natural algorithms be analyzed *directly*, without relying primarily on the data they generate? To appreciate this point, try a thought experiment. Suppose no one knew that a square matrix M could be diagonalized or put in Jordan normal form. To analyze the dynamical system

$$u_{t+1} = Mu_t,$$

we would then probably gather statistics on its iterates and search for numerical patterns. We would observe that the system seems to stretch the initial states in some directions, compress them in others, and sometimes produce oscillations. We would further learn to classify these oscillations into two types: periodic and quasirandom. Meanwhile, we would be making all these inferences *with no theory to explain them.* Unfortunately, this situation is all too common in the study of multiagent systems. Does it need to be so?

We wish to suggest that algorithms themselves should be harnessed as analytical tools *to study other algorithms.* A good example of such an inward-looking approach is mathematics itself. Although the motivation might often come from the outside (especially the physical sciences), most mathematical tools are in fact invented for internal purposes: determinants for matrices, resultants for polynomials, groups for algebraic equations, etc. Can the same be true of algorithms?

1.1 Results

We focus on the use of *algorithmic proofs* for the analysis of complex systems. The idea could not be simpler. Theorems often have proofs that look like algorithms. But theorems are hard to generalize whereas algorithms are easy to modify. Therefore, if a complex system is too ill-structured to satisfy the requirements of a specific theorem, why not *algorithmicize* its proof and retool it as a suitable analytical

device? We illustrate this idea with three examples related to agreement systems.

• We give a short algorithmic proof that a lazy random walk on a connected graph mixes in polynomial time. The result and the ideas behind our proof are well known, but the perspective is different. Our proof algorithmicizes the concept of reversibility in a Markov chain. Think of it as a warm-up exercise.

• Lorenz [27] and, independently, Hendrickx and Blondel [14], proved a counterintuitive bound on the nonzero probabilities occurring in colored random walks. We improve the bound to its optimal asymptotic value and prove a general ergodicity result for colored random walks. The proof is a flow algorithm tailored to mimic an eigenvalue computation. Specifically, it algorithmicizes the proof of Schur's Lemma. One might sense a bit of a paradox here, as eigenvalues are notoriously inadequate for tackling products of noncommuting matrices—the kind that occurs in colored walks. This illustrates an intriguing aspect of algorithmicized proofs, which is that the mathematical object upon which the original proof bears might become entirely irrelevant in the new proof.

• We introduce the *total s-energy* of a multiagent agreement systems. This is a generating function (specifically, a Dirichlet series) that partly captures the dynamics of the system. We show how to derive good convergence bounds from the order of its pole. We apply this to (i) Vicsek-Cucker-Smale flocking; (ii) Hegselmann-Krause opinion dynamics; and (iii) Kuramoto synchronization.

1.2 Discussion

Aren't most mathematical proofs algorithmic anyway? Computer science proofs, in particular, often track the flow of execution so closely as to resemble a rewriting of the code at a different level of abstraction. The literature in program verification, distributed computing, and proof-carrying code is full of such examples. So our basic point may seem at best unoriginal and at worst meaningless. After all, most proofs consist of discrete steps with variable names, conditionals, and the occasional recursive (ie, inductive) calls. So what's new? Our purpose here is not to introduce a formal concept but to appeal to an intuition that algorithms themselves must join the analytical arsenal of a theory of natural algorithms. The reason is that, in a practical sense (if not a formal one), *algorithms are more expressive than formulas and equations*. And so, to analyze natural algorithms, it is sometimes useful, or perhaps even indispensable, to think of the proof itself as an abstraction of the original algorithm. In particular, the standard method of, first, exhibiting forbidden structures and, second, drawing the combinatorial consequences might not always be suitable. (We discuss this point further in §2.3.) Our three examples point to the rich potential of an *algorithmic calculus* for dynamical systems. This work is a small contribution to this larger project.

Note that nonconstructivity is not necessarily an issue. The standard proof of König's Lemma, which says that an infinite connected bounded-degree graph has an infinite path, is nonconstructive. Yet, with the proper oracle in place, it is algorithmic and fits our model. On the other hand, consider the theorem stating the equality of the row and colum ranks of a matrix M. A nonalgorithmic proof will argue that both ranks must solve the feasible system,

$$\min \left\{ k \,|\, M = \textstyle\sum_k u_k v_k^T \right\},$$

and hence be equal. One can prove the same result algorithmically via Gaussian elimination. The proof is longer and less elegant, but it has two major advantages: first, it actually gives us the rank; second, it can be adapted to other purposes. For example, a few changes will show us how to invert a nonsingular matrix; a few more will prove that the determinant is multiplicative; further alterations will take us all the way to the simplex algorithm. Both proofs express duality in its simplest form. The difference is that Gaussian elimination does it *algorithmically*, via pivoting, whereas the structural proof appeals to symmetry: specifically, the equivalence of $(\exists u_k \exists v_k)$ and $(\exists v_k \exists u_k)$. Google's *PageRank* follows the same idea but takes it one step further: it converts the proof of the spectral theorem into an algorithm.

As we said earlier, an algorithmic approach to complex systems is nothing new. The works of Kleinberg [21, 22] and others to model social processes in an algorithmic language and integrate temporal dynamics fit that mold. Henzinger et al [15] have used model checking to automate the subdivision of the phase space of hybrid systems into coarse-grained classes and build approximate variants of Markov partitions. What distinguishes our approach from the latter is its emphasis on asymptotic analysis: if the system is scale-free, for example, we want to know its power-law.

The complexity of systems prediction has received considerable attention (eg, [3, 4, 20, 31, 38]). A limitation of these results is that they tend to zoom in on a corner of computational hardness that, most likely, evolution has kept at bay. The intractability of protein folding, for example, might simply mean that hard instances did not make it down the tree

of life [2, 36]. How adaptation navigates its way across the problem instance hardness spectrum is one of the most exciting open questions in science today. For example, there is a well-studied evolutionary arms race among species engaged in pursuit-evasion contests [7]. This has produced birds with astonishing powers to predict the short-term behavior of their (Turing-complete) predators. This is algorithmic magic and a reminder that *natural selection is the ultimate software optimizer*. It would be a pity if algorithms research had nothing to say on the matter.

Natural algorithms shine especially as out-of-equilibrium systems. Emergence in ant colonies or fish schooling arises as heat is evacuated and low-entropy energy is absorbed. This crucial point explains why classical thermodynamics, despite its recognized role in computer science [24], is probably ill-suited for natural algorithms. Jerrum and Sinclair [19] pioneered an algorithmic approach to studying the complexity of approximating the partition function of random fields in statistical physics. Although it bears relevance to phase transitions, this line of work is inherently about equilibrium (as is the present paper, we should add). This is not to say that statistical physics is irrelevant. In fact, it may well be that one of the most promising source of inspiration at this point is renormalization group theory [16], which allows for multiscale analysis of (self-similar) physical systems. The technique has been extremely useful in the study of phase transitions. To adapt it to natural algorithms is a challenging undertaking for future work. (Our third example takes a baby step in that direction.)

2 Three Algorithmic Proofs

When used as proofs, algorithms are to be granted more expressive power than usual: they may use oracles and infinite loops; they may be nondeterministic; they may break open a closed-loop dynamical system and feed it an adversarial signal; etc. Our proofs do all of that liberally.

2.1 Markov Chain Mixing

Let P be the transition matrix of a random walk on a connected graph G with n vertices. We add a self-loop to each vertex and let d_i denote the degree of vertex i (counting the self-loop): the walk at i hops to any neighbor with probability $1/d_i$. We give an algorithmic proof that the walk mixes rapidly, ie, that for any initially distribution π_0, the linear system

$$\mathcal{S} : \pi_{t+1}^T = \pi_t^T P$$

converges exponentially close to the distribution $\mathbf{p}$ in polynomial time, where $\mathbf{p}_i = d_i / \sum d_l$. The idea of the proof is to modify the algorithm $\mathcal{S}$ until the result essentially tumbles out. The pseudocode below does not describe one algorithm but a sequence of them defined by using operations from a simple algorithmic calculus. For example, the difference between two algorithms A, B is understood as the algorithm defined by subtracting the outputs of A and B. The same proof works unchanged for your favorite definition of a lazy walk. In fact, it applies to general aperiodic reversible Markov chains. Not only that, but we can even change the chain at each time step and, as long as we keep the stationary distribution invariant, it will still work. This invariance is required because time-dependent Markov chains may otherwise take exponential time to mix.

PROOF

[1] Let $\overleftarrow{S}(\mathbf{x})$ be the dual system: $\mathbf{x}_{t+1} = P\mathbf{x}_t$.
[2] Let $\overleftarrow{S}(\mathbf{x}) \odot \overleftarrow{S}(\mathbf{y})$ be the joint algorithm formed by running $\overleftarrow{S}(\mathbf{x})$ and $\overleftarrow{S}(\mathbf{y})$ simultaneously and, at each step t, picking a random vertex i with probability $\mathbf{p}_i$ and returning the product $(\mathbf{x}_t)_i(\mathbf{y}_t)_i$.
[3] Let $R(\mathbf{x}, \mathbf{y})$ denote the algorithm: $\overleftarrow{S}(P\mathbf{x}) \odot \overleftarrow{S}(\mathbf{y}) - \overleftarrow{S}(\mathbf{x}) \odot \overleftarrow{S}(P\mathbf{y})$.
[4] Run $R(\mathbf{x}, P\mathbf{x})$.

Pick a random i with probability $\mathbf{p}_i$ and, for any $k \geq 0$, let χ_k denote the i-th coordinate of $\mathbf{x}_{t+k}$. We easily check that $\mathbf{p}_i p_{ij}$ remains unchanged if we permute i and j; therefore,

$$\sum_{i,j} \mathbf{p}_i p_{ij} x_j y_i - \sum_{i,j} \mathbf{p}_i p_{ij} x_i y_j$$

is identically zero. This implies that algorithm $R(\mathbf{x}, \mathbf{y})$, and hence step [4], return only unbiased random variables; therefore, $\mathbf{E}\,\chi_1^2 = \mathbf{E}\,\chi_0\chi_2$ and the difference $\mathbf{var}\,\chi_0 - \mathbf{var}\,\chi_1$ is equal to

$$\begin{aligned} \mathbf{E}\,(\chi_0^2 - \chi_1^2) &= \mathbf{E}\,\chi_0^2 - \mathbf{E}\,\chi_0\chi_2 \\ &= \sum_{i<j} \mathbf{p}_i (P^2)_{ij} ((\mathbf{x}_t)_i - (\mathbf{x}_t)_j)^2 \qquad (1) \\ &\geq m^{-3} ((\mathbf{x}_t)_i - (\mathbf{x}_t)_j)^2 , \end{aligned}$$

for any (i, j) out of the m edges of G; both laziness and reversibility are used in (1). Because G is connected, by the pigeonhole principle, following a path

from the highest coordinate of $\mathbf{x}_t$ to the lowest one leads to an edge i, j for which $((\mathbf{x}_t)_i - (\mathbf{x}_t)_j)^2$ is at least $n^{-2}\,\mathbf{var}\,\chi_0$. It follows by induction that

$$\mathbf{var}\,\chi_k \leq (1 - n^{-O(1)})^k\,\mathbf{var}\,\chi_0\,.$$

The variance of χ_k thus decays exponentially in k. Since the initial vector $\mathbf{x}_0$ is arbitrary, this implies that the matrix P^t has rows that have ℓ_2 differences at most $e^{-tn^{-O(1)}}$, which proves rapid mixing.

Discussion. Step [1] introduces the time-reversal chain (hence the reverse arrow): we still use P because the chain is reversible. Steps [3,4] output only unbiased random variables: this is the algorithmicization of reversibility: technically, self-adjointness over $L^2(\mathbf{p})$ (which is visually apparent in step [3]). Informally, the algorithm expresses the fact that the correlation between yesterday and tomorrow can be inferred today. This is the key to rapid mixing and the basis, implicit or not, of every known proof. Recall that Perron-Frobenius alone cannot prove better than exponential-time mixing, and the crux of any eigenvalue proof is a bound on the spectral gap. Our proof does this indirectly. Its use of the pigeonhole principle along a path mimics the Landau-Odlyzko spectral gap proof [23], while the Dirichlet form in (1) follows its use by Mihail [30].

2.2 Colored Random Walks

Colored random walks are an interesting variant of the standard sort: each step brings in a new graph to walk on. It was introduced to computer science in the context of interactive proof systems [8, 9]. Being all about inhomogeneous products of stochastic matrices, however, the notion has been investigated in many other areas [37]. Let $\mathcal{G}$ be a set of connected graphs over the same set of n vertices: each graph is assigned a random walk with every positive transition probability bounded below by some parameter $\delta > 0$. We also assume that the walk is *bidirectional*, meaning that if the transition probability from i to j is nonzero then the same is true from j to i. (Note that reversibility implies bidirectionality but not the other way around.) Since graphs are thus annotated with probability distributions at the vertices, the set $\mathcal{G}$ is possibly infinite. A colored random walk is defined by a starting vertex and a word $G_0G_1\cdots G_N$, where the G_is belong in $\mathcal{G}$ but need not be distinct. At step t, the walk takes place in G_t.

Can the probability of hitting a certain vertex at time t be exponentially small in t for arbitrarily large t? If $\mathcal{G}$ consists of only one graph, then the answer is clearly no. But what if $|\mathcal{G}| > 1$? Then, indeed, some nonzero probabilities might decay exponentially in t; see [6] for an example with $|\mathcal{G}| = 2$. Colored random walks are, indeed, different. Yet Lorenz [27] and, independently, Hendrickx and Blondel [14], proved the surprising result that nonzero probabilities can be bounded from below *uniformly* as long as the walk is lazy. In other words, assuming that the transition matrix of each G_t has a nonzero diagonal, then, after t steps of a colored random walk, the probability of hitting a vertex is either 0 or at least $\delta^{O(n^2)}$, as opposed to, say, $\delta^{\Theta(t)}$. We improve the exponent to linear. This is obviously optimal: consider the probability of reaching one end of a chain from the other one in time equal to the number of edges between them.

Theorem 2.1. *Any lazy bidirectional colored random walk whose nonzero transition probabilities are bounded below by $\delta > 0$ hits any vertex with probability either 0 or at least δ^{n-1}. This holds at any time uniformly.*

The proof of this result is quite easy. We enhance it a little to establish a general result about ergodicity. With each vertex v, we define an infinite sequence of sets

$$S^v(t_1) \supseteq S^v(t_2) \supseteq \cdots S^v(\infty) \supseteq \{v\} \qquad (2)$$

such that, if the walk begins at $t = 0$ in $S^v(t_k)$, for $k = \delta^{-cn}$ with constant c large enough, and hits v at time t_k, then there is no way to tell which vertex of $S^v(t_k)$ the walk started from other than by a random guess (with exponentially small deviation). These sets, called *stabilizers* in [6], play an important role in products of inhomogeneous stochastic matrices [14, 27]. What makes them highly useful is that they depend only the communication process generated by the graph sequence and not on the random walk itself.

First, we dualize the problem to view it as a deterministic communication process. We consider the graphs $G_0\cdots G_N$ in reverse order, ie, $H_0\cdots H_N$, where $H_i = G_{N-i}$. Each vertex v holds a water reservoir with an amount R_v. To H_t*-average* the vertices is to replace each R_v by $\sum_w p_{vw}R_w$ at time t, where p_{vw} is the probability of going from v to w. It is obvious that, starting the walk at u at time 0, the probability of being at vertex v at time $N+1$ is precisely equal to the reservoir amount R_u after the vertices have been H_t-averaged, for $t = 0, 1, \ldots, N$, assuming that all reservoirs are empty at time $t = 0$, except for $R_v = 1$. Any vertex w with an empty reservoir is called *dry*; otherwise ($R_w > 0$) it is *wet*. We make the graph sequence $H_0\cdots H_N$ infinite by repeating it forever if need be.

PROOF

[1] Initialize t_v and all reservoirs to 0, except for $R_v = 1$. Set $S^v(0) = \{v\}$.
[2] Repeat forever:
[2.1] For $t = t_v, t_v + 1, \ldots, \infty$:
- H_t-average all the vertices.
- $S^v(t+1) \leftarrow S^v(t) \cup$ { all wet vertices dry at time t }.
- If $S^v(t+1) \supset S^v(t)$, then set t_v to $t+1$ and make a record of all the reservoirs.

[2.2] Restore all reservoirs to their values at t_v.
Let $m = \min R_u$ and $M = \max R_u$ over all $u \in S^v(t_v)$.
[2.3] If $R_v < \frac{1}{2}(M+m)$, then set $R_u \leftarrow M - R_u$ for all $u \in S^v(t_v)$ else set $R_u \leftarrow R_u - m$ for all $u \in S^v(t_v)$.

Whenever $S^v(t+1) = S^v(t)$, no new vertex is made wet; so, by directionality, no wet vertex is averaged with dry ones. This implies that the minimum reservoir amount cannot decrease. Suppose now that $S^v(t+1) \supset S^v(t)$ in step [2.1]. Then each newly wet vertex inherits at least a fraction δ of the minimum nonzero reservoir amount. By laziness, a wet vertex H_t-averages itself with its neighbors (all of which could be dry prior to t), and so its reservoir level drops by at most a factor of δ. But this can happen at most $n-1$ times, so in step [2.2], $m \geq \delta^{n-1}$; hence Theorem 2.1.

To prove the ergodicity claim, let $t_1 < t_2 < \cdots$ be the values of t_v initializing the for-loop of step [2.1]. Since water flows between adjacent vertices and is occasionally removed, (2) trivially holds. The transformation $R_u \leftarrow M - R_u$ ensures that, at the next round in the infinite loop [2], the amount of water at v is in the upper half of the range formed by the wet vertices. This in turn guarantees that in the next iteration the next value of m will always be at least $m' \geq \delta^{n-1}(M-m)/2$; so M decreases by a factor of $1 - \frac{1}{2}\delta^{n-1}$. Note that the amount of water in the entire system may sometimes increase: the decay is only observable in the maximum reservoir level M. This proves our claim that the colored random walk is ergodic when starting in $S^v(t_k)$ at $t = 0$, for $k = \delta^{-\Theta(n)}$, in that all starting points have roughly the same probability of leading to v at time t_k.

DISCUSSION. The algorithm has two infinite loops. The outer loop models the infinite (time-reversed) random walk while the inner one goes through time looking for stabilizers: their appearance can take arbitrarily long. Indeed, an adversary can drive each t_i as high as it wants. The proof is an algorithmicization of Schur's Lemma. The flip/shift [2.3] is intended to empty at least one reservoir and bring the water supply at v into the upper half of the range. This is the "Gram-Schmidt" part of the algorithm: the shift corresponds to a (partial) projection of the water vector along the principal right eigenvector. We use the fact that all matrices share this eigenvector (but possibly none others) without attempting to capitalize on individual spectral gaps, which would be futile. As in Schur's Lemma, the algorithm identifies an eigenvector and factors it out of the system by projection. The main difference is that the dimension of that vector may keep shrinking.

2.3 Multiagent Agreement Systems

Moreau introduced a general model for agreement systems [32] and established several convergence criteria, but with no quantitative analysis to go along. We introduce an analytical tool for that purpose, which we call the *total s-energy*. Moreau's model consists of n agents represented at time t by points $x_1(t), \ldots, x_n(t)$ in $\mathbb{R}^d$, together with an infinite sequence of undirected n-vertex graphs G_0, G_1, etc. At time $t \geq 0$, each agent is free to move anywhere in the relative interior of the set consisting of its own position and those of its adjacent agents in G_t. As observed in [1], there is nothing special about the convex hull and the model can be generalized to other regions. We use enclosing boxes for simplicity. Although this gives more room for the agents to move about, all of our upper and lower bounds apply to Moreau's model just the same.

We capture the dynamics of the system by defining a generating function for the edge lengths. By analogy with the Riesz s-energy of points on a sphere, we define the *total s-energy* of the system as

$$E(s) = \sum_{t \geq 0} \sum_{(i,j) \in G_t} \|x_i(t) - x_j(t)\|_2^s. \qquad (3)$$

This sum is a general Dirichlet series. To see why, assume that $E(s)$ converges. All the terms are nonnegative, so they can be rearranged in nonincreasing order. This allows us to match the standard definition of the general Dirichlet series [12]:

$$E(s) = \sum_{n \geq 1} a_n e^{-\lambda_n s},$$

where a_n is the number of edges (i,j) such that $\|x_i(t) - x_j(t)\|_2 = d_n$, and $\lambda_n = -\ln d_n$. Note that

if each G_t were to consist of a single edge of length $\frac{1}{t+1}$ then the total s-energy would be the Riemann zeta function. We show that $E(s)$ converges for all $s > 0$ and diverges for $s \leq 0$. By classical results in complex analysis [12], it then follows that the series converges uniformly over any finite region D of the complex plane within $\Re(s) \geq \sigma > 0$, for any $\sigma > 0$; furthermore $E(s)$ defines an analytic function over D. We prove that $s = 0$ is a pole of order $n - 1$. We use this fact to bound the convergence time of multiagent agreement systems.

Let $u_1, \ldots, u_d$ be an arbitrary unit coordinate system in $\mathbb{R}^d$, for constant $d > 0$; the vectors need not be orthogonal. Given a set S, let $\mathtt{pp}(S)$ denote the smallest parallelepiped $\sum_i [a_i, b_i]u_i \supseteq S$, and let $\|S\|_p$ be the ℓ_p-norm of the vector $\sum_i (b_i - a_i)u_i$. Fix arbitrary $\rho > 0$ smaller than a suitable constant and shrink $\mathtt{pp}(S)$ ever so slightly by defining

$$\overline{\mathtt{pp}}(S) = (1 - \rho)\mathtt{pp}(S) + \rho c(S),$$

where $c(S)$ is the mass center of $\mathtt{pp}(S)$. Note that $\overline{\mathtt{pp}}(S) \subseteq \mathtt{pp}(S)$. Let $X(0) = \{x_1(0), \ldots, x_n(0)\}$ be n points in $\mathbb{R}^d$ specifying the locations of the agents at time 0, and let $G_0, G_1, \ldots$ be an infinite sequence of undirected graphs over n vertices $v_1, \ldots, v_n$ (the agents). We define the following adversarial process. At each step t, the adversary moves each $x_i(t)$ anywhere inside the smallest (perturbed) parallepiped enclosing its neighbors and itself; specifically, if we define

$$\mathcal{N}_{t,i} = \bigcup \{\, x_j(t) \,|\, (v_i, v_j) \in G_t \text{ or } j = i \,\},$$

then $x_i(t+1)$ can be anywhere in $\overline{\mathtt{pp}}(\mathcal{N}_{t,i})$. If $x_i(t)$ is already inside $\overline{\mathtt{pp}}(\mathcal{N}_{t,i})$, note, of course, that motion is allowed but not required. Since the sequence G_t is infinite, the process goes on forever.

Previous convergence results makes various connectivity assumptions [1, 5, 14, 17, 25, 27, 32, 33, 41]. These assumptions are often necessary for global consensus (eg, infinitely recurring all-pair message passing) but they cannot be checked ahead of time: the only way to tell if they hold is to run the system until it converges. Our model avoids any such assumptions. We restrict ourselves to undirected graphs and positive ρ because both are necessary for *guaranteed* convergence. Relaxing either one opens the door to perpetual macro-oscillations (easy exercise). This is not to dismiss the nonbidirectional case, which is actually highly interesting, but to say that different techniques are likely to be required.

Let $E_n(s)$ be the maximum total s-energy for a system of n agents with coordinates between 0 and L. Since $E_n(s)$ scales with L as L^s, we can set $L = 1$ once and for all. Note that the quantity $E_n(s)$ is defined adversarially by choosing both the infinite graph sequence and the agent motion so as to maximize the total s-energy.

THEOREM 2.2. *For any* $0 < s < 0.99$,

$$s^{1-n}\rho^{-\Omega(n)} \leq E_n(s) \leq s^{1-n}\rho^{-n^2(1+o(1))}.$$

Given $\varepsilon > 0$, let T_ε be the number of times t the graph G_t contains at least one edge of length no shorter than ε. This quantity plays an essential role in bounding the convergence time of agreement systems.

THEOREM 2.3. *Given any* $\varepsilon > 0$,

$$T_\varepsilon \leq (\log \tfrac{1}{\varepsilon})^{n-1}\rho^{-n^2(1+o(1))}.$$

Proof. Viewed as an infinite series, the total s-energy adds ε^s or more to the sum for every graph G_t with at least one edge of length at least equal to ε. It follows then that $T_\varepsilon \leq \varepsilon^{-s} E_n(s)$. Setting $s = \min\{0.99, (1 - n)/\ln \varepsilon\}$ in the upper bound of Theorem 2.2 gives us the desired result.

We now prove Theorem 2.2. We show that the total s-energy satisfies the recurrence: $E_1(s) = 0$ and, for $n \geq 2$,

$$E_n(s) \leq 2nE_{n-1}(s) + (1 - \rho^{2n})^s E_n(s) + d^{s/2}n^3. \quad (4)$$

As in the case of colored random walks, all agents are initially dry, except for a selected agent v_1, which will spread "wetness" from one agent to the next, causing the geometry to change in the process. Once wet, an agent always remains so.

PROOF

[1] Initially, all agents are dry except for v_1. Set $S(0) = \{x_1(0)\}$.
[2] For $t = 0, 1, \ldots, \infty$:
  [2.1] Declare wet any agent adjacent to a wet agent in G_t.
  [2.2] $S^*(t) \leftarrow S(t) \cup \{$ positions at time t of dry agents just turned wet $\}$.
  [2.3] Move every agent adversarially with respect to G_t. If no newly wet agent, then we may carry all motion within $S(t)$ in isolation from the $n - |S(t)|$ other agents.
  [2.4] $S(t+1) \leftarrow \{$ positions at time $t+1$ of agents in $S^*(t)$ $\}$.

Let $\{t_k\}$ be the times t at which $|S^*(t)| > |S(t)|$. The sets $S^*(t_k)$ track wetness propagation. We interpret both $S(t)$ and $S^*(t)$ as multisets. No interesting geometry can be inferred from the latter but the same is not true of $S(t_k)$. We can show that

$$\|S(t_k)\|_\infty \le 1 - \rho^{2k}. \tag{5}$$

Consider the case $d = 1$. Let $[a, b] \subseteq [0, 1]$ be the smallest interval enclosing $S(t_k)$. By flipping the interval if necessary, we can assume that $a + b \ge 1$. By induction, it follows that $a \ge \frac{1}{2}\rho^{2k}$. Since $\|S(t)\|_1$ can increase only at times of the form $t = t_l$, we can prove (5) for t_{k+1} by showing that $[0, \frac{1}{2}a\rho) \cap S(t_k + 1) = \emptyset$. We proceed by contradiction. Consider an agent v_i contributing to $S(t_k + 1)$ with $x_i(t_k + 1) < \frac{1}{2}a\rho$. We distinguish between two cases:

- If $x_i(t_k)$ is dry at time t_k, then G_{t_k} has at least one edge (v_i, v_j) with v_j wet, ie, $x_j(t_k) \ge a$. Since $x_i(t_k + 1) < a$, it lies in an interval
$$(1 - \rho)[\alpha, \beta] + \tfrac{1}{2}(\alpha + \beta)\rho,$$
where $\beta \ge a$. It follows that
$$x_i(t_k + 1) \ge \tfrac{1}{2}\beta\rho \ge \tfrac{1}{2}a\rho.$$
- If $x_i(t_k)$ is wet at time t_k, $x_i(t_k) \ge a$ and $x_i(t_k + 1)$ again lies in an interval
$$(1 - \rho)[\alpha, \beta] + \tfrac{1}{2}(\alpha + \beta)\rho,$$
where $a \le x_i(t_k) \le \beta$. It follows that
$$x_i(t_k + 1) \ge \tfrac{1}{2}a\rho.$$

We get a contradiction in both cases, which proves (5). The same argument can be repeated along each dimension, so (5) holds for arbitrary d. Note that the set $S(t_k)$ can only gain agents, as k grows, but the set may stop growing before it absorbs all of them. When t is not of the form t_k, step [2.3] indicates that the adversary can act on $S(t)$ in isolation from the rest. It follows that the s-energy between t_{k-1} and t_k is bounded by $E_{|S(t_k)|}(s) + E_{n-|S(t_k)|}(s)$. At time t_k, the extra energy involved is

$$\sum_{(i,j)\in G_{t_k}} \|x_i(t) - x_j(t)\|_2^s \le \binom{n}{2} d^{s/2}.$$

Using obvious monotonicity properties, it follows that, up to the highest value of t_k, the s-energy is bounded by

$$\sum_{l=1}^{n-1} \left\{ E_l(s) + E_{n-l}(s) + \binom{n}{2} d^{s/2} \right\} \le 2nE_{n-1}(s) + d^{s/2}n^3.$$

When t_k reaches its highest point t, if $|S(t + 1)| < n$ then all the energy has been accounted for above. Otherwise, we must add the future energy of the n agents in $X(t + 1)$. By (5), however, their ℓ_∞-norm has been reduced: $\|X(t + 1)\|_\infty \le 1 - \rho^{2n}$. So, all we need to do is add $(1 - \rho^{2n})^s E_n(s)$ inductively to the s-energy; hence (4). We may assume that $0 < s < 1$ for the purpose of the upper bound proof.

The case $n = 2$ is worth special attention. The problem is inherently one-dimensional, so we can assume that the two agents start at 0 and 1, respectively, and move toward each other by the minimum allowed distance of $\rho/2$. This gives us the equation

$$E_2(s) = 1 + (1 - \rho)^s E_2(s).$$

Scaling up to d dimensions gives us:

$$E_2(s) = \frac{d^{s/2}}{1 - (1 - \rho)^s} \le \frac{2d^{s/2}}{s\rho}. \tag{6}$$

We now consider the case $n > 2$. For $s \le 1$, $(1 - \rho^{2n})^s \le 1 - \frac{1}{2}s\rho^{2n}$; it then follows from (4) that

$$E_n(s) \le \frac{4nE_{n-1}(s) + 2n^3 d^{s/2}}{s\rho^{2n}}.$$

We verify that the numerator is at most $2n^3 E_{n-1}(s)$; therefore,

$$E_n(s) \le \frac{2n^3 E_{n-1}(s)}{s\rho^{2n}} \le s^{1-n}\rho^{-n^2(1+o(1))}.$$

This proves the upper bound of Theorem 2.2. A much better asymptotic bound can be derived for the special case $s = 1$ (which, analytically, is indeed very special). But our proof is long and complicated and it will be reported elsewhere. It is easy to find forbidden structures and exploit them to derive upper bounds for general s. Briefly, we can show the existence of regions that can never be occupied by any agents and that can be "crossed" only in a special direction. This alone allows us to bound the total s-energy, but the result is not as tight and so far we have not been able to beat the algorithmic proof of Theorem 2.2.

To establish the lower bound, we show that the pole at $s = 0$ is, indeed, of order $n - 1$. We describe an algorithm $\mathcal{A}$ that moves n agents initially within the smallest enclosing interval of $[0, 1]$ toward a single point $x(n)$ while producing a total s-energy $S(n)$. Place $n - 1$ agents at position 0 and one at position 1. The graph G_0 consists of a single edge between the agent i at 1 and any one, j, of those at 0. At time 0, agent j moves to position $\alpha \stackrel{\text{def}}{=} \rho/2$ while i shifts to $1 - \alpha$. The $n - 2$ other agents stay put.

Next, apply $\mathcal{A}$ to the set of all agents but i. This brings them to position $\alpha x(n-1)$. Finally, apply $\mathcal{A}$ to all the agents. The attractor point $x(n)$ satisfies the recurrence $x(1) = 1$ and

$$x(n) = \alpha x(n-1) + (1 - \alpha x(n-1) - \alpha)x(n).$$

This implies that

$$\frac{1}{x(n)} = 1 + \frac{1}{x(n-1)};$$

therefore $x(n) = 1/n$. It should not be a surprise that $x(n)$ does not depend on ρ. The operations of $\mathcal{A}$ leave the center of mass invariant, so if $x(n)$ exists it must be $1/n$. The total s-energy $S(n)$ satisfies the relation: $S(1) = 0$; and, for $n > 1$, by (6),

$$\begin{aligned} S(n) &= \alpha^s S(n-1) \\ &\qquad + (1 - \alpha x(n-1) - \alpha)^s S(n) + 1 \\ &\geq \frac{\alpha^s S(n-1) + 1}{1 - (1-2\alpha)^s} \geq \frac{\alpha^{(n-2)s}}{(1-(1-2\alpha)^s)^{n-1}}. \end{aligned}$$

Since $\alpha = \frac{\rho}{2}$ is small enough, $(1-2\alpha)^s \geq 1 - 3\alpha s$ and

$$S(n) \geq s^{1-n}\rho^{-\Omega(n)}.$$

We observe that Algorithm $\mathcal{A}$ cannot start the second recursive call before the first one is finished. Of course, that takes an infinite amount of time. This technicality is easy to overcome, however. This completes the proof of Theorem 2.2.

Discussion. Self-confidence usually plays a crucial role in the convergence of agreement systems. This is the requirement that any agent should include itself in the averaging. What our result shows is that this condition can be relaxed. The shrinking by $1 - \rho$, which is necessary, has less to do with self-confidence than with the necessity of not following extremes (for the example, the boundary of the enclosing box). Of course, if one's opinion is itself extreme then by that same logic one must move away from it at least a little bit.

2.4 Flocking, Opinion Consensus, and Synchronization

Theorem 2.2 leads to new, or exponentially improved, bounds for flocking, consensus dynamics, and synchronization. We give a quick summary, leaving the technical details for the full version of the paper. Following Vicsek et al [42] and Cucker & Smale [10], the dynamics of bird flocking is expressed in [6] by:

$$\begin{cases} x(t) = x(t-1) + v(t); \\ v(t+1) = (P_t \otimes I_d)v(t). \end{cases} \tag{7}$$

The vectors $x(t), v(t)$ encode the positions and velocities of the n birds in dimension $d > 0$. The consensus matrix P_t is the stochastic transition matrix of the flocking network, which links any two birds within a fixed distance of each other.[1]

Instead of an adversary, the system has a closed loop determined by the (intricate) geometric dynamics of the flocks. The evolution of the velocity vector fits our model, however, with $\rho = n^{-O(1)}$. Let $2 \uparrow\uparrow n$ denote the tower-of-twos of height n. We proved that n birds may require as many as $2 \uparrow\uparrow \log \frac{n}{2}$ steps before reaching steady state and never more than $2 \uparrow\uparrow 4 \log n$ [6]. We also showed that the maximum number of times the flocking network can change is $n^{O(n^3)}$. Theorem 2.3 improves this bound to $n^{O(n^2)}$. (Obviously, this cannot have any incidence on the asymptotic convergence time, which is already optimal.) A similar improvement applies to the time for convergence in the Hegselmann-Krause opinion dynamics model [13]: this is a popular model in sociology to measure polarization in political opinions in a population. (Technically, this is not an improvement but a new result, since we are not aware of any previous asymptotic bound.)

Theorem 2.3 does not require linearity—both flocking and opinion dynamics are piecewise linear systems. It can therefore be used for collective synchronization. The *Kuramoto model* is a general framework for coupled oscillators with so many applications it is worth a brief mention here. Examples include flashing fireflies, chirping crickets, microwave oscillators, yeast cell suspensions, circadian neurons, and pacemaker cells in the heart (which keep our heart beating at roughly the same pace). After Winfree's pioneering work, Kuramoto set out to explain how huge systems of coupled oscillators can reach synchrony with no centralized control [39, 44]. He introduced a hugely influential model that is easy to describe. The system consists of n oscillators: the i-th one has phase θ_i and natural frequency ω_i. Kuramoto followed the same mean-field approximation as Winfree's and assumed that all pairs of oscillators were coupled. This gives the set of differential equations (for $1 \leq i \leq n$):

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{n}\sum_{j=1}^{n} \sin(\theta_j - \theta_i).$$

Many authors have considered a more realistic refinement of the model where the sum applies only to the neighbors of each agent in a network and where delays

[1] The tensor notation $\otimes$ distributes the averaging over each coordinate. We skip the details of the model (including noise and hysteresis) to keep the discussion simple.

might occur [11, 18, 26, 28, 29, 32, 34, 35, 43, 45]. Further work introduced a discrete version of the model, again out of concern for realism [28, 32, 40]. Assuming all oscillators share the same natural frequency, a fixed phase shift gives the dynamics:

$$\theta_i(t+1) = \theta_i(t) + \frac{K\Delta T}{|N_i(t)|+1} \sum_{j \in N_i(t)} \sin(\theta_j(t) - \theta_i(t)),$$

where $N_i(t)$ is set of neighbors of vertex i in G_t. Convergence to synchrony, when it happens, can be bounded by applying Theorem 2.3.

Acknowledgments

I wish to thank Peter Sarnak for helpful suggestions. This work was supported in part by NSF grants CCF-0634958 and NSF CCF-0832797.

References

[1] Angeli, D., Bliman, P.-A. *Stability of leaderless multi-agent systems: extension of a result by Moreau*, Mathematics of Control, Signals & Systems 18 (2006), 293–322.

[2] Berger, B., Leighton, F.T. *Protein folding in the hydrophobic-hydrophilic (HP) model is NP-Complete*, J. Computational Biology 5 (1998), 27–40.

[3] Blondel, V., Bournez, O., Koiran, P., Papadimitriou, C., Tsitsiklis, J. *Deciding stability and mortality of piecewise affine dynamical systems*, Theoretical Computer Science 255 (2001), 687–696.

[4] Buss, S., Papadimitriou, C.H., Tsitsiklis, J.N. *On the predictability of coupled automata: an allegory about chaos*, FOCS 1990, 788-793.

[5] Cao, M., Spielman, D.A. Morse, A.S. *A lower bound on convergence of a distributed network consensus algorithm*, 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain, 2005.

[6] Chazelle, B. *The convergence of bird flocking*, arXiv:0905.4241v1, 2009. Prelim. version in 20th SODA 2009, 422–431.

[7] Cliff, D., Miller, G.F. *Co-evolution of pursuit and evasion II: Simulation methods and results*, in P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, S. W. Wilson (eds), From Animals to Animats 4: Proc. 4th Int. Conf. Simulation Adaptive Behavior(1996), MIT Press, 506–515.

[8] Condon, A., Hernek, D. *Random walks on colored graphs*, Random Structures and Algorithms 5 (1994), 285–303.

[9] Condon, A., Lipton, R.J. *On the complexity of space bounded interactive proofs*, Proc. 30th IEEE Symp. Found. Comput. Sci. (1989), 462–267.

[10] Cucker, F., Smale, S. *Emergent behavior in flocks*, IEEE Trans. Automatic Control 52 (2007), 852–862.

[11] Earl, M.G., Strogatz, S.H. *Synchronization in oscillator networks with delayed coupling: a stability criterion*, Phys. Rev. E, 67 (2003), 036204(1–4).

[12] Hardy, G.H., Riesz, M. *The General Theory of Dirichlet's Series*, Cambridge University Press, 1915.

[13] Hegselmann, R., Krause, U. *Opinion dynamics and bounded confidence models, analysis and simulation*, J. Artificial Societies and Social Simulations 5 (2002).

[14] Hendrickx, J.M., Blondel, V.D. *Convergence of different linear and non-linear Vicsek models*, Proc. 17th International Symposium on Mathematical Theory of Networks and Systems (MTNS2006), Kyoto (Japan), July 2006, 1229–1240.

[15] Henzinger, T.A., Ho, P.H., Wong-Toi, H. *Algorithmic analysis of nonlinear hybrid systems*, IEEE Transactions on Automatic Control 43 (1998), 540–554.

[16] Honerkamp, J. Statistical Physics, Springer, 2nd ed., 2004.

[17] Jadbabaie, A., Lin, J., Morse, A.S. *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, IEEE Trans. Automatic Control 48 (2003), 988–1001.

[18] Jadbabaie, A., Motee, N., Barahona, M. *On the stability of the Kuramoto model of coupled nonlinear oscillators*, Proc. American Control Conference (ACC 2004).

[19] Jerrum, M., Sinclair, A. *Polynomial-time approximation algorithms for the Ising model*, SIAM J. Comput. 22 (1993), 1087–1116.

[20] Kilian, J., Siegelmann, H.T. *On the power of sigmoid neural networks*, Information and Computation 128 (1996), 48–56.

[21] Kleinberg, J. *Temporal dynamics of on-line information streams*, In Data Stream Management: Processing High-Speed Data Streams, (M. Garofalakis, J. Gehrke, R. Rastogi, eds.), Springer, 2004.

[22] Kleinberg, J. *The convergence of social and technological networks*, C.ACM 51 (2008), 66–72.

[23] Landau, H.J., Odlyzko, A.M. *Bounds for eigenvalues of certain stochastic matrices*, Linear algebra and Its Applications 38 (1981), 5–15.

[24] Li, M., Vitanyi, P. *Introduction to Kolmogorov complexity and its applications*, Springer-Verlag, New York, 3rd ed, 2008.

[25] Li, S., Wang, H. *Multi-agent coordination using nearest neighbor rules: revisiting the Vicsek model*, 2004, arXiv:cs/0407021v2

[26] Li, C., Xu, H., Liao, X., Yu, J. *Synchronization in small-world oscillator networks with coupling delays*, Physica A 335 (2004), 359.

[27] Lorenz, J. *A stabilization theorem for dynamics of continuous opinions*, Physica A: Statistical Mechanics and its Applications 355 (2005), 217–223.

[28] Martinez, S. *A convergence result for multiagent systems subject to noise*, Proc. American Control Conference (2009), St. Louis, Missouri.

[29] McGraw, P.N., Menzinger, M. *Analysis of nonlinear synchronization dynamics of oscillator networks by Laplacian spectral methods*, Phys. Rev. E 75 (2007), 027104.

[30] Mihail, M. *Conductance and convergence of Markov chains—a combinatorial treatment of expanders*, Proc. 30th Annual Symposium on Foundations of Computer Science (1989), 526–531.

[31] Moore, C. *Unpredictability and undecidability in dynamical systems*, Phys. Rev. Lett. 64 (1990), 2354–2357.

[32] Moreau, L. *Stability of multiagent systems with time-dependent communication links*, IEEE Transactions on Automatic Control 50 (2005), 169–182.

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

[34] Papachristodoulou, A., Jadbabaie, A. *Synchonization in oscillator networks: switching topologies and nonhomogeneous delays*, Proceedings of ECC-CDC (2005), Seville, Spain.

[35] Papachristodoulou, A., Jadbabaie, A. *Synchonization in oscillator networks with heterogeneous delays, Switching Topologies and Nonlinear Dynamics*, IEEE Conf. Decision and Control (2006), San Diego, CA.

[36] Papadimitriou, C.H., Sideri, M. *On the evolution of easy instances*, manuscript, 1998.

[37] Seneta, E. Non-Negative Matrices and Markov Chains, Springer, 2nd ed., 2006.

[38] Siegelmann, H.T., Sontag, E.D. *On the computational power of neural nets*, J. Comput. System Sci. 50 (1995), 132–150.

[39] Strogatz, S. *From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators*, Physica D 143 (2000).

[40] Triplett, B.I., Klein, D.J., Morgansen, K.A. *Discrete time Kuramoto models with delay*, Lecture Notes in Control and Information Sciences, Springer, 2005.

[41] Tsitsiklis, J.N., *Problems in decentralized decision making and computation*, PhD thesis, MIT, 1984.

[42] Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O. *Novel type of phase transition in a system of self-driven particles*, Physical Review Letters 75 (1995), 1226–1229.

[43] Wang, Q., Duan, Z., Chen, G., Feng, Z. *Synchronization in a class of weighted complex networks with coupling delays*, Physica A 387 (2008), 5616–5622.

[44] Winfree, A.T. *Biological rhythms and the behavior of coupled oscillators*, J. Theoret. Bio. 16 (1967), 15–42.

[45] Zanette, D.H. *Propagation of small perturbations in synchronized oscillator networks*, Europhys. Lett. 68 (2004), 356.