Rotor Router Walks

Alexander E. Holroyd[*] and James Propp

January 20, 2006

NOTE: This is essentially a place-holder for the article “Rotor Router Walks” that is in the process of being written by Ander Holroyd and myself. As of today the actual writing has not gotten very far, and what appears below is a stitching together of various fragments of the article that are not entirely consistent in their nomenclature and notation. I would normally not put something this rough on the web, but since my proposal on quasirandomness is currently under review, and since none of my own writings on this subject have been published yet or even released in preprint form, I felt I owed it to the reviewers to offer at least some indication of what I am proposing to do, above and beyond what appears in the research proposal.

As the writing progresses, this document will mutate into something more closely resembling a first draft. Until then, I hope that it at least conveys the flavor of the work and the outlines of the main results — and I hope that those who read this document will (a) excuse its infelicities and (b) not blame their frustrations on my co-author and (c) not inflict this document on others by passing on the PDF file or its URL.

(Jim Propp, January 20, 2006)

1 Introduction

[Where do we introduce the word “quasirandom”?]

This article is concerned with discrete Markov chains $X_0, X_1, X_2, \dots$, where the random variables X_n take their values in some finite or countable set S, and the joint probability distribution is determined by transition probabilities $p(x, y)$ in the usual fashion, so that the conditional probability

[*] Funded in part by an NSERC (Canada) grant.

$\mathrm{Prob}(X_n = x_n \mid X_{n-1} = x_{n-1}, \ldots, X_0 = x_0)$ is equal to $p(x_{n-1}, x_n)$ for all $x_0, \ldots, x_n$ (see [StandardReference] or section X of this article). We often think of the elements of S as the vertices of a directed graph G, where there is a directed edge from x to y whenever $p(x, y) > 0$, and we call $p(x, y)$ the weight of the directed edge; hence we may describe the Markov chain as random walk on the weighted directed graph G, and we imagine a walker or particle that successively visits vertex X_0, vertex X_1, vertex X_2, etc.

The quantities associated with Markov chains that will interest us are:

- **hitting probability**: How likely is it that random walk started from x will hit y before it hits z? We call x the source and call y and z sinks, and we assume that $p(y, x) = p(z, x) = 1$, so that after the walk hits a sink it returns to the source for another trip through the graph. The notion of hitting probability can be defined when y and z are not vertices but ends of the graph (ways for the walker to go to infinity) when G is infinite.

- **expected hitting time**: How long on average does it take for random walk started from x to hit y? Of course this will be infinite unless we assume that the probability that random walk started from x never hits y is 0.

- **stationary probability**: In the long run, what fraction of the time does the walker spend at vertex x?

There are a number of commonly used approaches for answering such questions (either exactly or approximately):

- **direct solution**: The answers to these questions satisfy linear relations that permit one to solve for them exactly or approximately via standard algorithms for solving systems of linear equations. If the graph is small, or if it is large (or infinite) but has a high degree of degeneracy or symmetry or other structure, exact solution is feasible. Otherwise, one may have to settle for approximations.

- **relaxation**: One can converge numerically to the exact solution to the aforementioned linear system by a simple iterative procedure.

- **pseudorandom simulation, aka Monte Carlo**: One can simulate the random walk on a computer using a pseudorandom number generator for some large finite number of steps and/or some large finite number of runs, and then use empirical data to estimate the quantities being considered (estimating probabilities by frequencies and expected values by averages), with errors that typically fall like $O(1/\sqrt{T})$ where T is time spent on simulation.

In this article we will describe a new method that has elements in common with all three approaches. In some cases it can be used to give exact solutions; in others, it can be used to give approximate solutions, with errors that are quite small (and in particular often fall like $O(1/T)$ rather than $O(1/\sqrt{T})$). The method is is based on simulation, but the simulation is not random; rather, it is deterministic and is designed to minimize a natural notion of discrepancy. The simulation scheme is quite interesting as an example of an exactly solvable cellular automaton model, quite aside from its possible uses in computation. Furthermore, the simulation scheme is parallelizable, and when it is run in parallel it is essentially a variant of the method of relaxation that uses extra variables to reduce error via a novel rounding protocol. As of this writing we do not claim that the method is suitable for real-world applications, but its extremely good performance for toy problems suggests that it might be useful. [Mention Owen? Mention Engel?]

Our model of simulation is simplest to describe when the graph G is finite and all transition probabilities are rational; for simplicity, assume that all transition probabilities are multiples of $1/d$. By introducing multiple arcs, we can ensure that all transition probabilities are exactly $1/d$, with d arcs emanating from each vertex. (If we are working with hitting probabilities or expected hitting times, we introduce d arcs of weight $1/d$ from each sink back to the source.) We have assumed that our Markov chain is irreducible, so the directed graph G is strongly connected, that is, that it is possible to get from any vertex to any other.

We introduce at each vertex v a cyclic ordering of the d arcs emanating from the vertex. We then conceive of a routing mechanism at v that accepts particle and passes them along to neighboring vertices according to the following rule: when a particle arrives at v, the mechanism remembers which way it sent the particle that most recently left v (that is, which arc that particle traversed) and routes the new particle along the next arc, where "next" means "next in the cyclic ordering of arcs emanating from v". Thus, in the figure below, if the neighbors of v are given the cyclic ordering $x, y, z, x, y, z \ldots$, and if the first particle that arrives at v gets routed to x, then the second particle that arrives at v gets routed to y, the third gets routed to z, the fourth gets routed to x, and so on.

Note that this use-the-next-arc rule does not specify what happens when a vertex is visited for the first time. Hence we adopt an alternative description of the routing mechanism in which every vertex v is assigned a "rotor" that at each time has a state, namely, a specific arc emanating from v; the dynamical rule is that when a particle arrives at v, the state of the rotor at v updates (i.e., it become the next arc in the cyclic ordering) and the particle gets routed along the new arc. Hence, in the case where a vertex v has been visited one or more times, the state of the rotor at v is nothing other than the arc from v that was traversed by the particle that most recently exited from v.

Thus, if the rotor at v in the figure starts by pointing at z, then the first time a particle arrives at v, it will be routed to x (since the rotor gets updated before the particle gets routed). The next particle that arrives at v will be routed to y, and the next to z, and so on, as above.

We call this the **rotor-router mechanism**. The two-part slogan for describing the mechanism is "The rotor rotates" (to its next position) "and the particle moves" (in the direction specified by the rotor's new position). For those who would prefer a mechanism that first routes the particle and then updates the rotor, so that the state of the rotor indicates the direction in which the *next* particle that visits a site will be routed — i.e., those who prefer rotors that are "prospective" rather than "retrospective"' — see the discussion of this issue in subsection X. There we show that the distinction is cosmetic, and that for at least some purposes, the convention we have chosen here is superior.)

Note that the above rule leaves unspecified the initial settings of the rotors. As we will see, many of our theorems apply regardless of how the rotors are set at the start, and in particular, would apply if one chose the initial settings randomly. However, it is crucial that the rotors not be interfered with once they have been initialized, except insofar as they are updated by the rotor-router rule.

Rotor-router walk is the operation of putting a particle at a vertex x and letting it walk in the graph G, getting routed by all the rotor-routers associated with the vertices that it meets in turn. Thus, if our graph G has vertices $\{x, y, z\}$ with two arcs at each vertex, one pointing to each neighbor of the vertex, and if our initial setting of the rotors consists of the arcs (x, y), (y, z), and (z, x), then rotor-walk starting from x would visit the vertices $x, z, y, x, y, z, x, z, y, x, y, ...$ This rotor-walk is eventually periodic.

It is easy to see that, in the current context (finite graphs with rational transition probabilities), every rotor walk must be eventually periodic. For, define a (global) *state* as an ordered pair consisting of a vertex of a graph (the location of the current particle) and an n-tuple (where n is the number of vertices of G) that gives the current setting of each rotor. Then the rotor-router update rule is a mapping from the set of global states to itself. Since the set of global states is finite, iteration of the update rule must eventually lead into a loop that is traversed forever.

From this observation, it follows that a question such as "What fraction of the time does a particle doing rotor-walk spend at location x?" must be rational (although it is not a priori evident that the rational number will be independent of the initial settings of the rotors). The happy fact is that the answers to such questions are the same for rotor-walk as they are for random walk. Specifically:

(1) Assume that there is one source x and two sinks y and z. The proportion of the time that the rotor-walking particle departing from x arrives at y (rather than z) is exactly the corresponding hitting probability for random walk.

(2) Assume that there is one source x and one sink y. The average time it takes for a rotor-walking particle departing from x to arrive at y is exactly the corresponding expected hitting time for random walk.

(3) The proportion of the time that the rotor-walking particle spends at a vertex v is exactly the stationary probability of v for random walk.

Note that, since the rotor-walking particle's trajectory is eventually periodic, terms like "average" and "proportion" mean "with respect to one period". (One could ignore periodicity and talk about the asymptotic average or asymptotic proportion; these would converge to the average or proportion over a single period as the length of the walk goes to infinity, since the effect of the initial transient becomes arbitrarily small, as does the effect of the incomplete period at the end of the simulation. For now, we will ignore this and focus on the behavior during a single period.) Thus, rotor walk gives a (not very efficient) procedure for exactly calculating these probabilities and expected values; one simply waits until the system commences periodic behavior and then counts or averages over the period. We are not claiming that this is a practical scheme, but these theorems should give a taste for the enterprise, and in particular, show the surprising computational power of the simple rotor-router mechanism.

For applications to hitting probabilities and expected hitting times, one might be tempted to reset the rotors after the particle has arrived at one of the sinks. This temptation may be especially strong if the initial setting of the rotors was chosen randomly, and one's mindset is still governed by ordinary Monte Carlo. "After all," one might reason, "if we don't reset the rotors, our next run will be correlated with the preceding run". In fact, these correlations are a good thing, and account for the superior performance of rotor-based simulation in comparison with random simulation. These correlations are exactly the sort of negative correlations that researchers call "antithetic"; in this case they come for free as a consequence of the rotor setup. We will see that this compensatory mechanism is present whether the initial settings of the rotors were chosen randomly or deterministically, but in the latter case, one cannot speak of "negative correlations" literally since everything is completely deterministic.

Here are proofs of the three preceding assertions:

(1) Create a modified version of the Markov chain with probability-1 transitions leading from the sinks y and z to the source x. Then the ratio of the hitting probabilities for y and z in the original Markov chain is equal to the ratio of the stationary probabilities of y and z for the modified Markov chain. We can conclude that the original Markov chain satisfies (1) if we can show that the modified Markov chain satisfies (3).

(2) We create a modified Markov chain with a probability-1 transition leading from the sink y to the source x. Then the expected hitting time from x to y in the original Markov chain is equal to 1 less than the reciprocal of the stationary probability of y for the modified Markov chain. We can conclude that the original Markov chain satisfies (2) if we can show that the modified Markov chain satisfies (3).

(3) For each vertex v, let $\pi(v)$ be the fraction of the time that the particle spends at v, so that $\sum_v \pi(v) = 1$. The rotor-mechanism insures that, for all v and w, the relative fraction of the time that the particle, upon exiting v, goes to w, is $p(v,w)$. Hence $\pi(w) = \sum_v p(v,w)\pi(v)$. Hence π coincides with the stationary measure for the random walk.

The rotor-router mechanism has been studied before, under the name "Eulerian walkers model". [Name the authors.] As these earlier authors noticed, the dynamics have the "abelian" property (or, in the terminology of Eriksson, exhibit "strong convergence). That is, if one has multiple particles moving through the system, and therefore has a choice of which particle to move next, the choice does not affect the final outcome, as long as the particles are indistinguishable. We will give a more careful statement of this property later; for now, it suffices to give examples.

(1') Suppose one is using rotor-routers to study hitting probabilities, and that in particular, one has sent a particle through the system n times, with the result that on a of its passages through the system the particle has hit y and on b of its passages through the system the particle has hit z, with $a + b = n$. Clearly it would be equivalent to send n particles through the system, one at a time (that is, requiring the first particle to arrive at y or z before sending the second particle out from x, and so on). But the strong convergence property says that we would get the same result if we put all n particles through the system together, regardless of the choices we make concerning which particle to move next. We could, for instance, take one of the n particles that start at x and move it repeatedly until it arrives at a sink, then take a different particle that starts at x and move it repeatedly until it arrives at a sink, and so on, until all n particles initially at y have arrived at a sink; this is essentially the same as sending n particles through the system one at a time. On the other hand, we could take each of the n particles that start at x, move each of them one step, move each of them another step (except for the ones that by this time have gotten absorbed by a sink), and so on, until all n of the particles have been absorbed at a sink. Regardless of what we do, a of the particles will wind up at y and b of the particles will wind up at z.

(2') Suppose one is using rotor-routers to study expected hitting times, and that in particular, one has sent a particle through the system n times, with the result that the average time it takes for the particle to reach the sink is T. Then one could instead send n particles through the system with moves taken in any order, starting from the same initial settings of the rotors, and one would still find that the average time it takes the particles to reach the sink is T.

(3') Suppose one is using rotor-routers to study stationary probabilities, Instead of sending a single particle through G for some large number of steps, one could do it with a larger number of particles. It is still the case that average number of particles occupying a given site (over time), divided by the total number of particles, equals the stationary probability of that site.

An example will clarify how we do rotor-router updates when there are multiple particles per site. If we are in the situation of Figure 1, with 100 particles starting at v, and the rotor at v initially pointing to z, then if we let each of the particles at x advance one step, the rotor at v will make thirty-three and a third complete revolutions, sending 34 particles to x, 33 particles to y, and 33 particles to z, with the rotor ending up pointing to x. That is, we have divided the 100 particles into three more or less equal batches, and since 3 does not equally divide 100, we have used the rotor to deal with the round-off error and decide which neighbors of v should receive fewer than 100/3 particles and which neighbors of v should receive more.

For a discussion of multi-particle rotor-walk on the (infinite) d-dimensional integer lattice $\mathbf{Z}^d$, see [Cooper and Spencer]. It should be mentioned that the results of Cooper and Spencer give $O(1)$ error-bounds in which the implied constant does not depend on the number of particles moving through the lattice or the number of time-steps they are being run forward. Nor does the bound depend on the initial settings of the rotors. (Here, as in other applications, it is possible that correctly chosen initial settings of the rotors may permit some increase in accuracy, but we won't explore this here.) One technical caveat in Cooper and Spencer's result is that the particles are required to start at vertices of the same parity.

Proving claims 1', 2', and 3' is fairly simple, as was the case for claims 1, 2, and 3. All the heavy lifting is done by the periodicity of the system; once one knows that the asymptotic averages exist, it is a simple consequence of the nature of the rotor mechanism that these averages satisfy the same linear relation and boundary conditions as the quantities that pertain to random walk. But these claims remain true for many infinite graphs, where the rotor system does not behave periodically. One still finds that the asymptotic averages associated with rotor-walk exist and coincide with the corresponding averages for random walk. Moreover, one finds that finite averages often converge quite rapidly to their asymptotic values, with errors that fall off like $O(1/n)$ where n is the number of simulation steps. (Compare this with the error-rate $O(1/\sqrt{n})$ for the ordinary Monte Carlo method of estimating these quantities, that is, direct simulation of the random walk itself rather than its deterministic rotor-router analogue.) In the finite case treated above, the $O(1/n)$ bound is a trivial consequence of convergence plus periodicity; in the infinite case, the fast convergence is the result of something a bit deeper.

Furthermore, a more general version of the rotor-router construction can be applied to finite and infinite graphs in which transition probabilities can be irrational, and for many of these cases one finds (where "finds" in some cases means "rigorously proves" and in other cases means "empirically observes") that the rate of convergence is quite rapid.

In fact, these results can be put in a general framework that applies to both random walk and rotor-walk. Consider the transition kernel $p(\cdot,\cdot)$ associated with an irreducible Markov chain with stationary probability distribution $\pi(\cdot)$. Let $x_1, x_2, x_3, \ldots$ be a sequence of states in the state space of the Markov chain. For each positive integer n and any two states y, z, let $\tilde{p}_n(y,z)$ be $N_p(y,z)/N_p(y)$, where $N_n(y)$ is the number of $1 \leq i \leq n-1$ with $x_i = y$ and $N_n(y,z)$ is the number of $1 \leq i \leq n-1$ with $x_i = y$ and $x_{i+1} = z$ (so that $\sum_z \tilde{p}_n(y,z) = 1$), and likewise let $\tilde{\pi}_n(y)$ be $N_p(y)/n$ (so that $\sum_y \tilde{\pi}_n(y) = 1$). Then one can prove (see Theorem X) a general result that says that if $\tilde{p}_n(y,z) \to p(y,z)$ for all y, z, then $\tilde{\pi}_n(y) \to \pi(y)$ for all y, and more strongly, one can prove a strengthened result in which the rate of convergence of $\tilde{\pi}_n$ is controlled by the rate of convergence of $\tilde{p}_n$. E.g., if the "discrepancy" (to be defined later) between $\tilde{p}_n$ and p goes down like $O(1/n)$ (the rotor-router case), then the discrepancy between $\tilde{\pi}_n$ and π goes down like $O(1/n)$ as well. Or, if the discrepancy between $\tilde{p}_n$ and p goes down like $O(1/\sqrt{n})$ (as would be the case with probability 1 for a sequence $x_1, x_2, x_3, \ldots$ given by random walk), then the discrepancy between $\tilde{\pi}_n$ and π goes down like $O(1/\sqrt{n})$ as well. [Ander: should we phrase this so that we are being honest but don't have to worry about iterated logarithms?] Similar remarks apply to the estimation of hitting probabilities and expected hitting times.

Before embarking on a discussion of these more general results, we will first take a detour through discrepancy theory. (A good source of background information is [Chazelle].) This provides some of the motivation for our definitions of rotors, and makes it clear how the notion should be generalized to allow for irrational transition probabilities. We also will introduce a "random stacks" picture (following work of Diaconis and Fulton and subsequent work of Wilson) that facilities a unified view that handles both deterministic and stochastic scenarios. Another detour will prove the aforementioned "abelian" or "strongly convergent" property of rotor-walk. Then we will give several applications of the general theory to special cases.

2 Antecedents and recent literature

The computer scientist Arthur Engel, in a remarkably prescient and neglected pair of articles, developed ideas rather similar to ours. His notion was that finite-state Markov chains could be deterministically simulated by chip-firing. We will give a proper definition of chip-firing later. The essential property to focus on now is that these are purely deterministic evolution-rules that mimic properties of a specified random evolution-rule. If you want to know the probability that a particle that starts at vertex v_1 in some graph and does random walk thereafter will arrive at vertex v_2 before it ever hits vertex v_3, Engel can describe for you a deterministic surrogate for your process that will let you estimate this answer by simulation. (Indeed, in the sorts of problems Engel considers, the answer is a rational number, and Engel's method does not merely let you estimate it to any desired level of accuracy; it lets you compute the probability exactly.) Likewise, it you want to know the expected number of steps it will take for a random walker that starts at v_1 to reach v_2, Engel's sort of derandomized walk can tell you.

Engel's work was neglected (probably in part because his articles were published in a mathematics education journal) and his notion of chip-firing (which he called "the probabilistic abacus") was reinvented many times by other people, such as Joel Spencer, Laszlo Lovasz [did Lovasz et al. come up with it independently of Spencer et al.?], Norman Biggs, and physicists, who use the term "abelian sandpile model". [Give references for all articles mentioned in this paragraph.]

Similarly, the rotor-router mechanism has been discovered independently a number of times. Workers in computer science ran across this mechanism in their study of balancing circuits; physicists came up with it as a variation on the sandpile model (which they dubbed the "Eulerian walker model"); and Dumitriu, Tetali, and Winkler came up with it as a by-product of their work on playing games on graphs, and more specifically, as the basis for an algorithm for computing the expected hitting time on a tree. We will see later that the "whirling tours" algorithm of Dumitriu, Tetali, and Winkler is a special case of the rotor-router method of computing hitting-times. [Give references for all articles mentioned in this paragraph.]

There is a somewhat more covert precedent for rotor-routers, to be found in work of Biggs and Winkler; indeed, going even further back, to work on Eulerian circuits. But to our knowledge, earlier work on Eulerian walkers (under its varied names) has not brought to the fore the connection with random walk, or exploited this connection as we do here.

Lastly, Diaconis and Fulton's notion of the smash product is essentially a random version of rotor-walk; they present it as a random procedure, but their stacks picture hints that randomness is inessential.

The present line of work began in 2001. as an outgrowth of the second author's long-standing interest in chip-firing and Diaconis and Fulton's work (both of which he learned about from David Aldous). After a few general results had been proved, and preliminary simulations of rotor-router aggregation (discussed in Section X) looked promising, he told Lionel Levine about the problem. Levine preceded to prove some theorems about rotor-router aggregation in one and two dimensions, as part of his senior honors thesis in 2002. The second author went on to present these results in 2003 at a workshop attended by the second author. The two of them proved the basic results that are the core of Section X. Later work by Cooper and Spencer and Levine and Peres was based on this work, which was unpublished but had been circulated in preliminary form by email.

3 Discrepancy and Rotors

Classical discrepancy theory of the combinatorial kind (as in the Beck-Fiala theorem; see [Chazelle]) assumes that we have a finite set-system (that is, a ground-set together with a collection of subsets of the ground-set; this is also called a hypergraph) and that we have a finite sample of elements of the ground-set that is supposed to be well-dispersed relative to all the chosen subsets. That is, the number of elements in our sample that fall within a particular set A, divided by the cardinality of A, should be approximately equal to the total number of elements in the sample divided by the cardinality of the ground-set, for all sets A in our set-system. Our set-up is different: the ground-set is quite small, and we are looking at an infinite sample (with repetition) whose finite truncations are all supposed to be well-dispersed with respect to *all* of the subsets of the ground-set, where the notion of "well-dispersed" now depends on a specified probability measure on the ground-set.

Recall that the total variation distance between two probability measures p and p' on some finite set S is $\frac{1}{2}\sum_s |p(s)-p'(s)|$, or equivalently, $\sup_A |p(A)-p'(A)|$, where the supremum is taken over all subset of S.

Every infinite sequence $x_1, x_2, \ldots$ of elements of S determines a sequence of probability measures p_N on S, where $p_N(s) = |\{k : 1 \leq k \leq N \text{ and } x_k = s\}|/N$. Suppose that each element $s \in S$ occurs with well-defined limiting frequency $p(s)$, so that the measures p_N converge to p in the total variation metric. We define the (unnormalized) discrepancy D_N as N times the total variation distance between p_N and p, and we say that the sequence $x_1, x_2, \ldots$ has bounded discrepancy iff the D_N's are bounded, that is, if there exists a constant C such that for all $N \geq 1$ and all $A \subseteq S$, $p(A)$ differs from

$|\{k : 1 \leq k \leq N \text{ and } x_k \in A\}|/N$ by at most C/N.

We write S as $\{s_1, s_2, \dots\}$ and write $p(s_i)$ as p_i. Let $n_k(N)$ denote the number of occurrences of s_k in the finite sequence $x_1, x_2, \dots, x_N$. Note that $x_1, x_2, \dots$ can be reconstructed from these numbers. We will call $n_k(N)$ ($1 \leq k \leq |S|$, $1 \leq N \leq \infty$) a rotor-scheme.

Claim: For any probability measure p on a finite set S, there exists an infinite sequence $x_1, x_2, \dots$ of elements of S that approximate p with bounded discrepancy, indeed, with discrepancy at most $(|S|-1)/2$.

Proof: For a probability vector $\mathbf{p} = (p_1, \dots, p_k)$ and a rotor scheme $\mathbf{n} = (n_1(N), \dots n_k(N))_{N=0}^{\infty}$ define the discrepancy

$$D(\mathbf{p}, \mathbf{n}) = \sup_N \sum_i |n_i(N) - p_i N|,$$

and let

$$D(\mathbf{p}) = \inf_{\mathbf{n}} D(\mathbf{p}, \mathbf{n}).$$

We need to show that $D(p_1, \dots, p_k) \leq k - 1$. For $|S| = 1$ this is trivial, and for $|S| = 2$ this is easy, since we can achieve $D(p, 1-p) \leq 1$ by taking $n_1(N)$ to be the closest integer to pN. The general case will follow by induction, once we prove the inequality

$$D(p_1 q_1, \dots, p_1 q_\ell, p_2, \dots, p_k) \leq D(p_1, \dots, p_k) + D(q_1, \dots, q_\ell).$$

To verify this inequality, note that given schemes $\mathbf{m}, \mathbf{n}$ for $\mathbf{p}, \mathbf{q}$ respectively, we can use the scheme $(n_1(m_1(N)), \dots, n_\ell(m_1(N)), m_2(N), \dots, m_k(N))$. By the triangle inequality,

$$|n_i(m_1(N)) - p_1 q_i N| \leq |n_i(m_1(N)) - q_i m_1(N)| + q_i |m_1(N) - p_1 N|.$$

On summing we obtain

$$D(p_1 q_1, \dots, p_1 q_\ell, p_2, \dots, p_k) \leq D(\mathbf{q}, \mathbf{n}) + D(\mathbf{p}, \mathbf{m}).$$

(It is not hard to check that our estimate $D(p_1, \dots, p_k) \leq k-1$ is optimal to within a factor of 2. For, in the case where k is even and $\mathbf{p}$ is $(1/k, \dots, 1/k)$, the best any rotor-scheme can do for the case $N = k/2$ is to have $k/2$ of the n_i's equal to 1 and the other $k/2$ of them equal to 0, yielding discrepancy $(k/2)|1 - (1/k)(k/2)| + (k/2)|0 - (1/k)(k/2)| = k/2$.)

When all of the probabilities p_i are rational, it is clear that we can find a finite-discrepancy rotor-scheme that is periodic. We will not explore here the issue of minimizing the discrepancy in this case, though it seems clear that a technology of rotor-router simulation would need to include a complete analysis of the issue as part of its infrastructure.

As we mentioned in the preceding proof, when the set S has two elements there is an especially simple way to minimize discrepancy: writing S as $\{0, 1\}$, $\pi(1)$ as p, and $\pi(0)$ as $1 - p$, we may take

$$(*) \qquad x_n = \text{the integer closest to } pn. \tag{1}$$

The infinite sequence $x_1, x_2, x_3, \ldots$ has discrepancy $\frac{1}{2}$ relative to π. This is not, however, the only good sequence to choose. More generally, given any initial value r_0 in $[-\frac{1}{2}, \frac{1}{2}]$, iteratively define

$$(**) \qquad (r_n, x_n) = \begin{cases} (r_{n-1} + p, 0) & \text{if } -\frac{1}{2} \leq r_{n-1} + p \leq \frac{1}{2}, \\ (r_{n-1} + p - 1, 1) & \text{if } -\frac{1}{2} \leq r_{n-1} + p - 1 \leq \frac{1}{2}, \end{cases} \tag{2}$$

(in the borderline case where $r_{n-1}+p = \frac{1}{2}$ and $r_{n-1}+p-1 = -\frac{1}{2}$, either choice is acceptable). Note that at least one of the numbers $r_{n-1} + p$, $r_{n-1} + p - 1$ must lie in the interval $[-\frac{1}{2}, \frac{1}{2}]$, since the two numbers differ by the width of the interval and since they lie on opposite sides of a number that lies in the interval (namely r_{n-1}). So, by induction, $-\frac{1}{2} \leq r_n \leq \frac{1}{2}$ for all n. The resulting sequence $x_1, x_2, x_3, \ldots$ has discrepancy at most 1 [check!]; in the particular case $r_0 = 0$, we get the specific sequence given by $(*)$ discussed in the previous paragraph.

In the case where the p is rational with denominator n, the sequence given by $(*)$ is periodic [deal correctly with the ambiguous case]. In this situation, where $p = \frac{a}{n}$ and $1 - p = \frac{b}{n}$ with a, b relatively prime positive integers summing to n, we can avoid the ambiguity in our mechanism by using instead the state-space $\{1, 2, \ldots, n\}$, with r_0 in this set and with $r_1, x_1, r_2, x_2, \ldots$ iteratively determined by

$$(r_n, x_n) = \begin{cases} (r_{n-1} + a, 0) & \text{if } 1 \leq r_{n-1} + a \leq n, \\ (r_{n-1} - b, 1)\text{if } 1 \leq r_{n-1} - b \leq n, \end{cases}$$

where now exactly one of the two conditions must hold.

In some cases, one might like to do rotor-router simulation of a one-parameter family of Markov chains, and not just an individual Markov chain; for instance, one might be doing some sort of sensitivity analysis, to assess how a change in the transition probabilities (mediated by some real parameter

t) affects hitting probabilities, expected hitting-times, or stationary probabilities. Rotor-walk, on account of its low discrepancy, gives good estimates of these quantities, so it is well-suited to estimation of differences between these quantities at nearby values of t (t_1 and t_2, say). To make the best use of the rotor-router simulation, one would want to use rotor-schemes for $\mathbf{p}_1$ and $\mathbf{p}_2$ that are "highly correlated" (although we stress as before that the notion of correlation is only metaphorical, since there is no randomness). One way of doing this would be to use a single universal scheme for quasirandomly sampling real numbers from the unit interval, and then map these numbers to the set S in a fashion that varies on the parameter t. E.g., consider the van der Corput sequence $1/2, 1/4, 3/4, 1/8, 5/8, \ldots$ of continuous discrepancy theory, whose nth term is gotten by writing the bits in the binary expansion of n in reverse order and preprending a decimal point (more properly speaking, a "binary point"). If one wanted to use rotors to simulate a fair coin of variable bias t, with t taking on a continuum range of values, one would map every term in the interval $(0, t)$ to Heads and every term in the interval $[t, 1)$ to Tails. (We will see later that the van der Corput sequence need not be "imported" into our theory from without, but can be built within it, via rotor walk on the infinite binary tree.) This sort of approach is akin to Arthur Owen's work on quasirandom simulation. However, it is worth noting that one pays a slight price for the flexibility of this variant; rotor-schemes on a set S that come from such an approach typically do not have bounded discrepancy, but rather discrepancy that grows like the logarithm of the sample-size. This price will sometimes, but not always, be worth paying.

4 Transfinite simulation time

Imagine a particle that does rotor-walk on a strongly connected graph G for an infinite number of steps, starting from some source-vertex x. If some vertex v is visited by the particle infinitely often during the walk, then every neighbor of v is also visited infinitely often. Hence every site in G will be visited infinitely often by the walk, and there is no obvious sense in which one could define the position of the particle or the state of the system "at time infinity". If, however, there is no vertex v that is visited infinitely often, then there is a well-defined limiting state for each rotor (since each rotor is changed only a finite number of times), and we imagine that the particle itself has vanished. Hence we may at that point re-start the simulation by placing a new particle at the source x and letting it do rotor-walk, starting at x at times $\infty+0$ and proceeding to other vertices at times $\infty+1$, $\infty+2$, etc. If there is no vertex that gets visited infinitely often by this second particle, then one can define the state at time 2∞ as the state in which each rotor has as its direction the limit of its direction at time $\infty+N$ as N goes to infinity. And so on, with more particles. (One can make this notion of transfinite time more formal with ordinals $a\omega+b$, but we will use the notation $a\infty+b$.)

5 Stacks of Rotors

So far, we have used rotors that record which way a particle went the last time a given site was exited. This is useful for many purposes. For instance, it will give us a clean criterion for recognizing when a rotor-setting is "recurrent" (namely, when it is acyclic); see section X. However, in some cases it is better to use rotors that present which way the next position that exits a vertex should go. (For one thing, the last-exit interpretation doesn't make sense when a site hasn't even been visited yet!)

We will call rotors of the former sort *retrospective rotors*, and rotors of the latter sort *prospective rotors*.

In the case where the rotor at each vertex just cycles through the integers mod n_v, it's easy to switch between prospective and retrospective pictures, by adding or subtracting 1 mod n_v.

A useful mental picture that unifies the prospective and retrospective pictures, and even the random and deterministic realms, is the model, introduced by Diaconis and Fulton and rediscovered independently by Wilson. Suppose we are observing some process in which a single particle moves through a graph (possibly in "transfinite time"). Then for each vertex v, we can record for each $n \geq 1$ the vertex that the particle went to after it exited v for the nth time (which may be undefined if the particle does not exit v n times); call this $s(v,n)$. Then I claim that one can reconstruct the walk-process just from the function $s(\cdot,\cdot)$, as long as one knows where to start the process, and where to re-start it if the particle leaves the system. It does no harm to define $s(v,n)$ for all n, even if v doesn't get exited n times. In this way, each vertex v gets associated with the list $s(v,1), s(v,2), s(v,3), \ldots$. When we execute the walk associated with the assemblage of lists, we need only keep track of the current position of the particle and where we are within each list (that is, which steps have been taken and which haven't). We may imagine having a kind of bookmark in each list, namely, a pointer that points between two adjoining entries in the list (or possibly right before the first entry in the list or right after the last entry in the list); the entry preceding the pointer (if it exists) is the *last* exit direction from v and the entry following the pointer (if it exists) is the *next* exit direction from v. If we focus on the entry before the pointer, then we are looking at the retrospective rotor; if we focus on the entry after the pointer, then we are looking at the prospective rotor. In both cases, we can choose to discard all elements of the list preceding the one we're focussing on; then instead of a list with a pointer, we have a stack whose elements (from top to bottom) are the undiscarded elements of the list, and every time we exit a site we pop another entry off its stack.

If we start from a random walk and turn it into a field of stacks, then the probability distribution governing the stack at v is just the initial distribution on the set of successors of v, and moreover, each such stack is independent; that is, all the random variables $s(v,n)$ are independent. Turning this around: if we take $s(v,n)$ to be independent random variables of the specified kind with $\mathrm{Prob}(s(v,n) = v') = p(v,v')$, then we get a construction of the random walk process.

On the other hand, if the $p(v,v')$ are all rational and we take the $s(v,n)$ deterministically so that the sequences $s(v,1), s(v,2), \ldots$ are low-discrepancy periodic sequences approximating the probability distribution $p(v,\cdot)$, then we get a construction of the rotor-walk process.

We find it helpful to adopt a viewpoint in which one is agnostic about whether the stacks at the vertices are deterministic stacks or random stacks that have been randomized ahead of time. Either way, the particle or particles that move through the field of stacks will have no freedom; its behavior is completely determined by the contents of the stacks it encounters. We call this *stack-walk*.

6 Strong convergence

As we mentioned earlier, when there are several particles doing rotor-walk on a graph, the order in which the particles move does not matter, as long as the particles are indistinguishable. (Note that the sort of strong convergence being discussed here applies when one is parallelizing random walk until absorption by a sink, *not* random walk for a fixed number of steps. If a multiple particles are moving on a graph, each of them taking a fixed number of steps, it is essential that when two of them temporarily occupy the same site, each particle remembers how many steps it has already taken, and this is inconsistent with the stricture that the particles be indistinguishable at every stage of the multi-particle rotor-walk process.)

The right setting in which to view strong convergence is not rotor-walk, but stack-walk more generally. E.g., if two particles do random walk until they are absorbed, the law governing where they are absorbed is the same regardless of whether they advance in tandem or not. Of course this particular claim about random walk is trivial, but others that are less trivial (such as some properties of loop-erased walk) turn out to be consequences of invariance properties of stack-walk.

In the simplest case (which nonetheless conveys the essential idea), we have two particles doing stack-walk on G. We will take a prospective rotors point of view, so we imagine that two walkers are moving through G, and each time a walker moves to a new vertex, she reads the top instruction in the stack at that vertex (which tells her where to go next), and follows that instruction after destroying the message (as in the TV series "Mission Impossible"). Each walker moves until absorbed by some sink. We want to show that the final outcome (where the walkers get absorbed, and what the stacks look like) is independent of the order in which the walkers take their steps. That is, we imagine two joint histories, each of which gives a way for the two walkers to walk until absorption, and we want to show that the end results look the same, except that it's possible that the first joint history has walker #1 ending up at sink y and walker #2 ending up at sink z while the second joint history has walker #1 ending up at sink z and walker #2 ending up at sink y, for some pair of sinks y and z.

There are two cases to consider. If there is no non-sink vertex that gets visited by both walkers, then the claimed independence is clear. Otherwise, there is some non-sink vertex v that is visited by both walkers. We may then imagine that each walker is doing a two-stage process: first stack-walk until absorption at v, then stack-walk until absorption at the original sinks of G. If we compare the histories of the two particles, we see that the only possible difference between them, aside from the order in which the different steps occur, can be compensated for by having them swap identities when they reach v. [Actually, this is not quite right, since there could be vertices that are visited by both walkers after they leave v. But we can fix the write-up later.]

Indeed, Michael Kleber and Matt Cook have independently pointed out a stronger invariance property. Suppose we have a finite set of identical walkers dispersed throughout a graph G (not necessarily originating from a single source). These walkers do stack-walk-with-absorption, resulting in a certain set of instructions being followed. (Specifically, at each vertex v, the top k_v instructions are followed, for some k that depends on v.) Suppose that we now restart the process with the walkers dispersed throughout G as they were before, but with the top k_v instructions in each stack permuted in some arbitrary fashion. Then the walkers will end up at the same sinks as before (possibly permuted), and the stacks will end up looking exactly as they did before (that is, for each v, the top k_v instructions on the stack will have been read, followed, and destroyed). To prove this, note that (in view of the fact that the walkers are indistinguishable) the history of a multi-walker stack-walk is determined by a list of all the instructions that are followed, in the order that they are followed. Suppose we have two such lists, L_1 and L_2. We wish to show that the two lists are permutations of each other.

Let us call an instruction on a list *active* when it is the top instruction in its stack and there is at least one walker at that site, or more generally, when it is among the top i instructions in its stack and there are at least i walkers at that site. A site that has one or more active instructions will be called active. The stack-walk process may be described as follows: At some active vertex, the top instruction is deployed and removed from its stack, where deploying an active vertex means activating the topmost inactive instruction in some particular adjacent stack (as indicated by the instruction). Note that deploying an active vertex can also render it inactive, depending on how many walkers were at that site just before the site was deployed: if there was just one walker, then the site becomes inactive, but if there were two or more walkers, the site remains active, and the number of walkers there decreases by one. Note that we can define a function that records the number of walkers that are at a site and update it as instructions are deployed, where deploying an instruction decreases the number of walkers at one (active) vertex by 1 and decreases the number of walkers at another vertex by 1.

Suppose now that the list L_1 contains m instructions from the stack of some vertex v, whereas the list L_2 contains fewer than m instructions from the stack at v. Of those m instructions in the list L_1, let I be the last of them (that is, the one that occurs lowest in the stack at v, before the entries in the stack get permuted). ... Suppose now that there is some instruction on the list L_1 that is not on the list L_2. (If two different entries in some stack say the same thing, i.e., point to the same neighboring vertex, we regard them as different instructions, for purposes of the present argument.) That is, there is some vertex v such that the number of instructions at v that get Suppose the mth instruction in L_1 is the first instruction in L_1 that is not in L_2; call this instruction I. Then all the instructions that precede I in L_1 eventually occur in L_2, and when they have all occurred, the vertex at which I sits will necessarily be active. [This is seriously messed up. Fix!] So at some later stage in L_2, the instruction I must occur, contradicting our choice of I.

The strong convergence property has many implications. For one thing, in the case where the stacks are given by rotors, strong convergence implies that there is a natural abelian group on the set of states. For now, we content ourselves with discussing the notion of recurrence.

ADD MATERIAL ON RECURRENCE

7 Examples

Example 1: Consider the directed graph on the set of dyadic rationals in $(0, 1)$ with root at $\frac{1}{2}$ and edges from $k/2^n$ to $(2k-1)/2^{n+1}$ and $(2k+1)/2^{n+1}$ for $n \geq 1$ and for k an odd number between 1 and $2^n - 1$. Give each directed edge probability $\frac{1}{2}$. Each end of the tree can be identified with an infinite sequence of 0's and 1's, where a 0 represents a leftward arc $k/2^n \mapsto (2k-1)/2^{n+1}$ and a 1 represents a rightward arc $k/2^n \mapsto (2k+1)/2^{n+1}$, and the sequence $b_1, b_2, b_3, \ldots$ encodes the number $b_1/2 + b_2/4 + b_3/8 + \ldots$. In this way, the ends of the tree map into $[0, 1]$, and the hitting measure on the ends of the tree (product-measure of the two-point measure $\pi(0) = \pi(1) = \frac{1}{2}$) maps to Lebesgue measure on $[0, 1]$. If we quasirandomize the system with rotors, and set the initial state with all (retrospective) rotors pointing to the right, then successive walks through the directed graph from the root hit the ends

$$\begin{array}{ccccc}
0 & 0 & 0 & 0 & \ldots \\
1 & 0 & 0 & 0 & \ldots \\
0 & 1 & 0 & 0 & \ldots \\
1 & 1 & 0 & 0 & \ldots \\
0 & 0 & 1 & 0 & \ldots \\
1 & 0 & 1 & 0 & \ldots
\end{array}$$

which, when mapped into $[0, 1]$, are the dyadic rationals $0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \ldots$, that is, the van der Corput sequence.

Example 2: Consider random walk on $\{-1, 0, 1, 2, \dots\}$ with a source at 1 and sinks at -1 and 0, with transition probabilities $p(n, n+1) = p(n, n-2) = \frac{1}{2}$ for $n \geq 1$. Since the expected change in position is $-\frac{1}{2}$ at each step, the walker gets absorbed at -1 or 0 with probability 1. Let q be the probability that the walker gets absorbed at -1. Then $q = (\frac{1}{2})(1) + (\frac{1}{2})(r)$, where r is the probability that a walker that starts at 2 gets absorbed at -1. Consider the first time that the walker that starts at 2 visits a site to the left of 2; with probability q (resp. $1 - q$), the walker is at 0 (resp. 1) at this time (because of the translation-invariance of the transition probabilities). Hence $r = (q)(0) + (1 - q)(q)$. Combining these two equations, we get $2q - 1 = r = q - q^2$, so $q^2 + q - 1 = 0$, and $q = (-1 + \sqrt{5})/2$, the reciprocal of the golden ratio. Alternatively, we may argue thus: the function $h(x)$, defined as the probability that a walker starting from x eventually lands at -1, is harmonic on $1, 2, 3, \dots$, i.e., it satisfies $h(x) = \frac{1}{2}h(x-2) + \frac{1}{2}h(x+1)$. Using the theory of linear recurrence relations, we see that the characteristic equation $\lambda^2 = \frac{1}{2} + \frac{1}{2}\lambda^3$ has roots $(-1 \pm \sqrt{5})/2$ and 1, so we must have $h(x) = A + B((1+\sqrt{5})/2)^x + C((1-\sqrt{5})/2)^x$ for all $x \geq -1$, for suitable coefficients A, B, C. Since $h(x)$ is a probability, $0 \leq h(x) \leq 1$ for all x. Hence B must equal 0 (since $(1 + \sqrt{5})/2$ is the only root of the characteristic equation with magnitude > 1). Then the two initial conditions $h(-1) = 1$ and $h(0) = 0$ suffice to determine A and C, from which we deduce $h(1) = (-1 + \sqrt{5})/2$.

We can derandomize this walk using rotor-routers, as described in [Kleber]. Leaving aside the trivial invariant (associated with the constant harmonic function $h(x) = 1$ for all x), there are essentially just two invariants, associated with $h_+(x) = ((1 + \sqrt{5})/2)^x$ and $h_-(x) = ((1 - \sqrt{5})/2)^x$. The former does not satisfy our summability constraint, so we will not apply Theorem X to it. [ANDER: Our article needs to talk about harmonic functions that aren't summable, since in at least this one case, it's a tool that lets one prove a not-wandering-to-infinity result!]

[insert stuff about h_+]

With h_-, we find that every particle-free rotor state has value between $-\alpha$ and β, that the process of adding a particle at 1, letting it walk until it hits $\{-1, 0\}$, and then removing the particle increments the total value of the rotors by β or $-\alpha$, according to whether the particle hit -1 or 0. [Fix details.] It follows that if we define y_n to be 1 or 0 according to whether the nth run has the particle that started at 1 arriving at -1 or 0, then $y_1, y_2, y_3, \dots$ has discrepancy $\frac{1}{2}$ (the smallest value possible) relative to the asymptotic distribution $\pi(1) = (-1+\sqrt{5})/2$, $\pi(0) = (3-\sqrt{5})/2$. That is, a mechanism made of $\frac{1}{2}, \frac{1}{2}$ rotors effectively simulates a $(-1+\sqrt{5})/2, (3-\sqrt{5})/2$ rotor.

Furthermore, if we define r_n as the value of the rotors after the nth particle has passed through the system, then

$$(r_{n+1}, y_{n+1}) = \{\text{fill in details}$$

exactly as in $(**)$. Thus we are simulating a rotor.

We pause here to note that the pairs (r_n, y_n) are exactly what one would get if one used quasi-Monte Carlo to estimate the integral of the indicator function

$$f(t) = \begin{cases} 1 & \text{if ...,} \\ 0 & \text{otherwise.} \end{cases}$$

on the interval $[0,1)$' using the fractional part of $n\frac{1}{\tau}$ as one's nth sample-point for all $n \geq 1$. Specifically, ...[insert details]. Thus, in this case, using rotors to estimate the probability of some event occurring in random walk is in a deep sense equivalent to using a low-discrepancy sequence of points to quasi-randomly estimate the integral of a certain function on a continuous measure space, For that reason if no other, we feel justified in using the term "quasirandom" to describe our construction, despite the fact that Chung and Graham and others have used the term in a somewhat different sense in discrete probability. However, lest one conclude from this analogy that quasirandom discrete simulation via rotors is merely a matter of arranging to sample a function over trajectories that are dense in some way, the next example shows us otherwise.

Example 3: Consider random walk on $\{0,1,2,3,\dots\}$ with a source at 1 and sinks at 0 and ∞, with transition probabilities $p(n,n-1) = \frac{1}{3}$, $p(n,n+1) = \frac{2}{3}$ for $n \geq 1$. Since the expected change in position is $+\frac{1}{3}$ at each step, the walker that starts at 1 has a positive probability of wandering off to infinity without ever getting absorbed at 0. Let q be the probability that the walker that starts at 1 gets absorbed at 0. Then $q = (\frac{1}{3})(1) + (\frac{2}{3})(r)$, where r is the probability that a walker that starts at 2 gets absorbed at 0. We also have $r = (q)(q) + (1-q)(0)$ (the first term corresponds to the case in which the walker that starts at 2 eventually hits 1, and the second term corresponds to the complementary case). Combining these two equations, we get $q = \frac{1}{3} + \frac{2}{3}q^2$, so $q = 1$ or $q = \frac{1}{2}$. [How do we rule out $q = 1$, Ander?] Alternatively, we may argue that the function $h(x)$, defined as the probability that a walker starting from x eventually lands at 0, is harmonic on $1, 2, 3, \ldots$, i.e., it satisfies $h(x) = \frac{2}{3}h(x-1) + \frac{1}{3}h(x+1)$. The characteristic equation $\lambda^2 = \frac{1}{3} + \frac{2}{3}\lambda^2$ has roots 1 and $\frac{1}{2}$, so we must have $h(x) = A + B(1/2)^x$ for all $x \geq 0$, for suitable coefficients A, B. Since $h(x) \to 0$ as $x \to \infty$ [how do we know this, Ander?], we must have $A = 0$, and since $h(0) = 1$, we must have $B = 1$, whence $h(x) = (\frac{1}{2})^x$ for all $x \geq 1$, and in particular, $h(1) = \frac{1}{2}$.

Derandomizing the walk with three-state rotors, we obtain the deterministic bugs-on-a-line scenario of [Winkler]. If we let $s^{(n)}$ denote the state of the entire system (that is, the settings of all the rotors) after the nth bug has passed through the system, we find that $s^{(1)} = s^{(3)} = s^{(5)} = \ldots$ and $s^{(2)} = s^{(4)} = s^{(6)} = \ldots$ (though the states $s^{(2)}, s^{(4)}, s^{(6)}, \ldots$ need not coincide with the initial state $s^{(0)}$).

Note that in no sense is the sequence $s^{(0)}, s^{(1)}, s^{(2)}, \ldots$ "dense" in the space of all rotor-settings, relative to the obvious compact product topology or any other non-trivial topology. So in this case, the fact that our sequence $y_1, y_2, y_3, \ldots$ has discrepancy $\frac{1}{2}$ from its limiting distribution is not a consequence of our settings $r_1, r_2, r_3, \ldots$ being dense. Rather, the settings are tailored to the function we are trying to integrate (or, to use probabilistic language, the random variable whose expected value we are trying to compute). Indeed, if one asks "How can this quasirandom scheme be assured of giving the right answer if it doesn't visit every part of the space?", the answer is "If the parts of the space that aren't visited didn't have the right properties (notably, the fact that this function is harmonic there), that led us to not go there, we would have *had* to go there." [Fix this.] Cf. the fact that if you want to know the value of a harmonic function at a point, you need only look at the values of that function on the boundary. We would be interested in knowing whether this has been exploited in the Monte Carlo integration world in the continuum realm. [This isn't quite right. We don't even visit a dense subset of the boundary. So what gives?] From a dynamical systems point of view, this is a bit strange. On the one hand, our tour of the state-space, being periodic, is manifestly not ergodic. On the other hand, the average of the function f over each orbit is equal to the average value of f on the measure space as a whole. So, with respect to the function f, our periodic dynamical system behaves as if it were ergodic. Of course, this is only for a particular function f, whose definition hinges upon a choice of harmonic function which itself was used in constructing the rotor-mechanism that gives us the walk (that is, the dynamics), so this is not as paradoxical as it might initially seem.

Example 4: Consider ordinary random walk on a finite tree G, with a source at x and a sink at y (which may as well be a leaf, since vertices that are separated from x by y will never be reached by the walk). As usual, we add a transition $p(y,x) = 1$ to allow for repeated walks. We quasirandomize the walk by creating rotors at each vertex $v \neq y$ that cycle through the neighbors of v in some order. Suppose we initialize all rotors so that they point in the direction of y. We claim that when the walker reaches y, every vertex will have been visited. For, consider a vertex $z \neq y$ that is visited as the walker moves from x to y. Since the rotor at z starts by pointing towards y and ends by pointing the same way, and since it changes at least once (the first time z is visited), the rotor at z must make one or more full turns, so that all neighbors of z get visited as well. This shows that if a vertex z is visited by the walker, so is every neighbor of z; hence, by induction on distance from x, every vertex in G gets visited. Hence, when the walker hits y, every vertex will have been visited, and every vertex will have its rotor pointing toward y — just as at the start. Hence, if we re-start the process by putting the walker at y back at x and letting it do rotor-walk, the walker will take the same tour as before, and reach y in exactly the same number of steps (call it n) as before. And so on, ad infinitum — each time we place the walker at x, it reaches y after exactly n steps. Hence by Theorem X, the expected hitting time from x to y (under random walk) is n. This is precisely the "whirling tours" algorithm of [DTW]. What makes the argument work is the fact that each rotor has one and only state that points toward y. The result holds more generally for any Markov chain on a tree that has the property that for each vertex $z \neq y$, the probability of the arc from z in the direction of y has probability of the form $1/m_z$ for some integer m_z. Indeed, the whirling tour algorithm, generalized to this case, shows that the expected hitting time from x to y must be an integer, namely, the number of steps of rotor-walk required to get from x to y when each rotor starts by pointing the direction of y. The simplest example is the two-state Markov chain with state-space $\{x,y\}$ with $p(x,y) = 1/m$, with expected hitting time from x to y equal to m.

Furthermore, in the case where the transition probabilities from vertex z are all multiples of $1/m_z$, our methods let us conclude that the expected hitting time n is equal to the number of spanning arborescences of the directed graph whose vertices are the states of the Markov chain and whose directed arcs correspond to allowed transitions, with the proviso that a transition from v to v' with $p(v,v') = k/m_v$ corresponds to an arc from v to v' with multiplicity k, that is, k arcs from v to v'. To see this, note first that since at each step in the n-step rotor-walk, the orientations of the rotors constitute such an arborescence. No two of these arborescences can be the same, since the arborescence determines the position of the particle (at the root), so if two of the arborescences were the same, they would correspond to the same exact state of the rotors-and-particle system, which would mean that the system's evolution had entered into a cycle, contradicting the fact that the period of the system is n. Hence the number of arborescences is greater than or equal to n. On the other hand, consider a spanning arborescence A, and let A_0 be the spanning arborescence rooted at y (unique by hypothesis). If we evolve A for enough steps (with a particle that starts at the root of A), the particle must eventually reach y, at which point the rotor-state will be A_0. Hence A leads to A_0 under rotor-evolution. Since A is recurrent, A must be in the forward history of A_0. That is, A is one of the n arborescences that comes from the rotor-walk. Hence the number of such arborescences is exactly equal to n, the hitting time from x to y.

8 Main results

Markov chain. Let $X_0, X_1, \ldots$ be a Markov chain on a countable state space V with transition probabilities $p : V \times V \to [0,1]$. We call the elements of V **vertices**. We write $\mathbb{P}_u$ for the law of the Markov chain started at vertex u (so $\mathbb{P}_u$-a.s. we have $X_0 = u$). Let $T_v = \min\{t \geq 0 : X_t = v\}$ be the first hitting time of vertex v (where $\min \emptyset = \infty$). As usual we say that p is **irreducible** if for any $u, v \in V$ there exists a finite sequence of vertices $u = u_0, u_1, u_2, \ldots, u_n = v$ with $p(u_0, u_1)p(u_1, u_2) \cdots p(u_{n-1}, u_n) > 0$.

Rotor walk. The following deterministic cellular automaton can be used to approximate quantities associated with the Markov chain. Assume that all transition probabilities $p(u, v)$ are rational and that for each u there are only finitely many v such that $p(u, v) > 0$. To each vertex u we associate a finite sequence of $d(u)$ (not necessarily distinct) vertices $s_1 u, \ldots, s_{d(u)} u$ [ARE $s_1, s_2, \ldots$ FUNCTIONS? SHOULD WE PUT PARENTHESES AROUND u?] (called the **successors** of u) in such a way that

$$p(u, v) = \frac{\#\{i : s_i u = v\}}{d(u)} \quad \text{for all } u, v \in V. \tag{3}$$

(This is clearly possible under the given assumptions). A **rotor configuration** is a map r which assigns to each vertex v an integer $r(v) \in \{1, \ldots d(v)\}$. (We think of an arrow or "rotor" located at each vertex, where the rotor at v points to vertex $s_{r(v)}v$). A **rotor walk** associated with p [I THINK WE SHOULD PRECEDE THIS BY AN INFORMAL DESCRIPTION; OTHERWISE THE NOTATION MAY BE HARD FOR READERS TO ASSIMILATE] is a sequence of **particle locations** $x_0, x_1, \ldots \in V$ together with rotor configurations $r_0, r_1, \ldots$ constructed inductively as follows. Given x_t and r_t at time t we set:

(i) $r_{t+1}(x_t) = (r_t + 1) \mod d(x_t)$
and $r_{t+1}(v) = r_t(v)$ for $v \neq x_t$
(increment the rotor at the current particle location and leave all other rotors unchanged); and

(ii) $x_{t+1} = s_{r_{t+1}(x_t)}(x_t)$ (move the particle in the new rotor direction).

Given a rotor walk, write

$$n_t(v) = \#\{i \in [0, t-1] : x_i = v\}$$

for the number of times the particle visits vertex v before time t.

Hitting probabilities. Fix two vertices b, c and consider the hitting probability

$$h(v) = \mathbb{P}_v(T_b < T_c).$$

In order to estimate this using a rotor walk, fix also an initial vertex a and assume that $p(b, a) = p(c, a) = 1$ (so that when the particle reaches b or c it is immediately returned to a) — this does not change the function h. Let $x_0, x_1, \ldots$ be an associated rotor walk.

[EXPLAIN HOW TO DEAL WITH HITTING ONE SET VERSUS ANOTHER, OR HITTING ONE VERSUS WANDERING OFF.]

Theorem 4 *Under the above assumptions, suppose that*

$$K := 1 + \frac{1}{2} \sum_{u \in V \setminus \{b,c\},\ v \in V} d(u)p(u,v)|h(u) - h(v)| < \infty.$$

Then for all t,

$$\left| h(a) - \frac{n_t(b)}{n_t(b) + n_t(c)} \right| \leq \frac{K}{n_t(b) + n_t(c)}.$$

Thus the proportion of times when the rotor walk hits b as opposed to c approximates the Markov chain hitting probability $h(a)$. Note that the condition $K < \infty$ holds in particular whenever V is finite. [CASE OF SYMMETRIC RANDOM WALK.]

Continuing beyond infinity. The above result is only useful for estimating $h(a)$ if $n_t(b) + n_t(c) \to \infty$ as $t \to \infty$, and we now investigate when this holds, and what to do if it does not. We say that a rotor walk is **recurrent** if it visits every vertex infinitely often, and **transient** if it visits every vertex only finitely often.

Lemma 5 *Any rotor walk $x_0, x_1, \ldots$ associated with an irreducible Markov chain is either recurrent or transient.*

Note in particular that if V is finite and p is irreducible then any rotor walk is recurrent.

Suppose that $x_0 = a$ and that the rotor walk $x_0, x_1, \ldots$ is transient. Then we can define a configuration r_∞ by $r_\infty(v) = \lim_{t\to\infty} r_t(v)$. Now restart the particle at a by setting $x_\infty = a$, and define a rotor walk $x_\infty, x_{\infty+1}, x_{\infty+2}, \ldots$ according to the usual rules. If this is again transient we can set $r_{2\infty} = \lim_{t\to\infty} r_{\infty+t}$ and restart at $x_{2\infty} = a$ and so on. Continue in this way up to the first k for which the walk $x_{k\infty}, x_{k\infty+1}, \ldots$ is recurrent, or indefinitely if it is transient for all k. Call this sequence of walks a **transfinite rotor walk** started at a.

[ADD COMMENTS ABOUT COMPUTATIONAL FEASIBILITY.]

Now consider a transfinite rotor walk started at a and let $T = I\infty + t$ be a generalized time. Let $n_T(v) = \#\{i < T : x_i = v\}$ be the number of visits to v before time T. Also let $n_{\infty^2}(v)$ be the total number of visits ever to v. [WE SHOULD BRIEFLY SAY WHAT OUR PARTIAL ORDERING ON GENERALIZED TIMES IS]

Lemma 6 *For an irreducible Markov chain and a transfinite rotor walk started at a, for any $T = I\infty + t$, either $n_T(v)$ is finite for all v or $n_T(v)$ is infinite for all v. Furthermore, $n_{\infty^2}(v)$ is infinite for all v.*

Theorem 7 *Under the assumptions of Theorem 4, suppose further that p is irreducible, and that*

$$\limsup_{v\in V} h(v) = 0.$$

Then for any generalized time $T = I\infty + t$ at which all vertices have been visited finitely often,

$$\left| h(a) - \frac{n_T(b)}{n_T(b) + n_T(c) + I} \right| \le \frac{K}{n_T(b) + n_T(c) + I}.$$

[REMOVE c FROM THIS? - JUST HITTING ONE VERTEX? (says Ander) I AGREE! (says Jim)]

Thus the proportion of times the particle hits b as opposed to hitting c or "going to infinity" approximates $h(a)$.

Hitting times. Fix a vertex b and let

$$k(v) = \mathbb{E}_v T_b$$

be its expected hitting time. To estimate this using rotors fix also an initial state a and assume $p(b, a) = 1$. Let $x_0, x_1, \ldots$ be an associated rotor walk.

Theorem 8 *Under the above assumptions, suppose that*

$$K' := \sup_{v \in V} k(v) + \frac{1}{2} \sum_{u \in V \setminus \{b\},\, v \in V} d(u) p(u, v) |k(u) - k(v) - 1| < \infty.$$

Then for all t,

$$\left| k(a) + 1 - \frac{t}{n_t(b)} \right| \le \frac{K'}{n_t(b)}.$$

Thus the average time to get from a to b approximates the expected hitting time.

Stationary probabilities. Suppose that the Markov chain is irreducible and positive recurrent with stationary distribution $\pi : V \to [0, 1]$. Let $k(v) = \mathbb{E}_v T_b$ as before. Let $x_0, x_1, \ldots$ be an associated rotor walk.

Theorem 9 *Under the above assumptions, suppose that*

$$K'' := K' + 1 + \frac{1}{2} \sum_{v \in V} d(b) p(b, v) |1/\pi(b) - k(v) - 1| < \infty,$$

where K' is as in Theorem 8. Then for all t,

$$\left| \pi(b) - \frac{n_t(b)}{t} \right| \le \frac{K''}{t} \pi(b).$$

Thus, the fraction of time spent in b approximates $\pi(b)$. [FIND A CLEANER K". EXPLAIN HOW TO GET $\int f d\pi$.]

9 Proofs of main results

For a function $f : D \to \mathbb{R}$ write $\mathrm{span}_{i \in D} = \sup_{i \in D} f(i) - \inf_{i \in D} f(i)$.

Lemma 10 *If* $\sum_{i=1}^{n} a_i = 0$ *then* $\mathrm{span}_{j=1}^{n} \sum_{i=1}^{j} a_i \leq \frac{1}{2} \sum_{i=1}^{n} |a_i|$.

Proof. For any $j \leq j'$, $\left| \sum_{i=1}^{j} a_i - \sum_{i=1}^{j'} a_i \right| = \left| \sum_{i=j}^{j'} a_i \right| \leq \sum_{i:a_i>0} a_i = \frac{1}{2}(\sum_{i:a_i>0} a_i - \sum_{i:a_i<0} a_i) = \frac{1}{2} \sum_{i=1}^{n} |a_i|$. □

Proof of Theorem 4. Note that $h(b) = 1$, $h(c) = 0$, while by the Markov property, h enjoys the harmonicity property:

$$h(u) = \sum_{v \in V} p(u,v) h(v) = \frac{1}{d(u)} \sum_{i=1}^{d(u)} h(s_i u) \quad \text{for } u \notin \{b,c\}, \tag{11}$$

hence

$$\sum_{i=1}^{d(u)} [h(u) - h(s_i u)] = 0 \quad \text{for } u \notin \{b,c\}. \tag{12}$$

Given a vertex x and a rotor configuration r, consider the quantity

$$\Phi(x,r) := h(x) + \sum_{u \in V \setminus \{b,c\}} \phi(u, r(u))$$

where

$$\phi(u,j) := \sum_{i=1}^{j} [h(u) - h(s_i u)].$$

We claim that

$$\mathrm{span}_{x,r} \Phi(x,r) \leq K \tag{13}$$

where the span is over all vertices and all possible rotor configurations, and K is as in the statement of the theorem. To check this, note firstly that $h(x) \in [0,1]$, and secondly, for $u \notin \{b,c\}$, (12), Lemma 10 and (3) imply

$$\mathrm{span}_j \phi(u,j) \leq \frac{1}{2} \sum_{i=1}^{d(u)} |h(u) - h(s_i u)| = \frac{1}{2} \sum_{v \in V} d(u) p(u,v) |h(u) - h(v)|. \tag{14}$$

Summing over u gives (13).

Now consider the change in Φ produced by a step of a rotor walk from (x_t, r_t) to (x_{t+1}, r_{t+1}). The only term in the sum over u which changes is the one corresponding to $u = x_t$. If $x_t \notin \{b, c\}$ then

$$\Phi(x_{t+1}, r_{t+1}) - \Phi(x_t, r_t) = h(x_{t+1}) - h(x_t) + [h(x_t) - h(s_{r_{t+1}} x_t)] = 0,$$

(where we have used (12) in the case when $r_{t+1}(x_t) = 1$). If $x_t \in \{b, c\}$ then

$$\Phi(x_{t+1}, r_{t+1}) - \Phi(x_t, r_t) = h(a) - h(x_t).$$

Hence, summing over t we obtain

$$\Phi(x_t, r_t) - \Phi(x_0, r_0) = n_t(b)[h(a)-1] + n_t(c)[h(a)-0] = [n_t(b)+n_t(c)]h(a) - n_t(b).$$

Combining this with (13) yields

$$\Big|[n_t(b) + n_t(c)]h(a) - n_t(b)\Big| \leq K,$$

and dividing by $n_t(b) + n_t(c)$ yields the result. □

Proof of Lemma 5. It is enough to show that if u is visited infinitely often and $p(u, v) > 0$ then v is visited infinitely often. This is immediate since $s_i u = v$ for some i, so the rotor at u will be incremented to point to v infinitely often. □

Proof of Lemma 6. As in the preceding proof, if u is visited infinitely often and $p(u, v) > 0$ then v is visited infinitely often. For the second assertion note that a is visited infinitely often. □

Proof of Theorem 7. The above proof adapts to the transfinite case. Suppose $x_0, x_1, \ldots$ is a transient rotor walk. We claim that

$$\Phi(x_\infty, r_\infty) - \lim_{t\to\infty} \Phi(x_t, r_t) = h(a).$$

The claim is proved as follows. The assumption of the theorem and the fact that the walk is transient imply that $\lim_{t\to\infty} h(x_t) = 0$. We clearly have $\lim_{t\to\infty} \phi(u, r_t(u)) = \phi(u, r_\infty(u))$, and by (14) we have for all u and j that $|\phi(u, j)| \leq g(u)$ where $\sum_{u \in V \setminus \{b,c\}} g(u) \leq 2(K-1) < \infty$. Hence by the dominated convergence theorem,

$$\lim_{t\to\infty} \Phi(x_t, r_t) = 0 + \sum_{u \in V\setminus\{b,c\}} \phi(u, r_\infty(u)) = \Phi(x_\infty, r_\infty) - h(a).$$

We have proved the above claim; thus whenever we "restart from infinity to a", the quantity Φ increases by $h(a)$. Hence the previous argument gives

$$\Big|[n_T(b)+n_T(c)+I]h(a)-n_T(b)\Big| \le K.$$

□

Proof of Theorem 8. We have $k(b)=0$, and

$$k(u)=1+\sum_{v\in V} p(u,v)k(v) \quad \text{for } u\neq b,$$

hence

$$\sum_{i=1}^{d(u)}[k(u)-k(s_iu)-1]=0 \quad \text{for } u\neq b. \tag{15}$$

Consider the quantity

$$\Psi(x,r):=k(x)+\sum_{u\in V\setminus\{b\}}\psi(u,r(u)),$$

where

$$\psi(u,j):=\sum_{i=1}^{j}[k(u)-k(s_iu)-1].$$

Using (15) and Lemma 10, we have $\operatorname{span}_{x,r}\Psi(x,r)\le K'$.

The change $\Psi(x_{t+1},r_{t+1})-\Psi(x_t,r_t)$ produced by a step of the rotor walk equals $k(x_{t+1})-k(x_t)+[k(x_t)-k(x_{t+1})-1]=-1$ if $x_t\neq b$, or $k(x_{t+1})-k(x_t)=k(a)$ if $x_t=b$. Hence

$$K'\ge|\Psi(x_t,r_t)-\Psi(x_0,r_0)|=|n_t(b)k(a)+n_t(b)-t|,$$

and dividing by $n_t(b)$ yields the result. □

Proof of Theorem 9. Denote the expected return time

$$m(v):=\mathbb{E}_v\min\{t\ge 1: X_t=b\},$$

so that $m(v)=k(v)$ for $v\neq b$, while $m(b)=1/\pi(b)$, and

$$\sum_{i=1}^{d(u)}[m(u)-k(s_iu)-1]=0 \quad \text{for all } u\in V. \tag{16}$$

Consider the quantity

$$\Gamma(x, r) := k(x) + \sum_{u \in V} \gamma(u, r(u)),$$

where

$$\gamma(u, j) := \sum_{i=1}^{j} [m(u) - k(s_i u) - 1].$$

Using (16) and Lemma 10, we have $\operatorname{span}_{x,r} \gamma(x, r) \leq 1 + \frac{1}{2} \sum_{u,v \in V} d(u) p(u, v) |m(u) - k(v) - 1| \leq K''$. ??? [WHY "???"]

The change produced by a step of the rotor walk is

$$\Gamma(x_{t+1}, r_{t+1}) - \Gamma(x_t, r_t) = k(x_{t+1}) - k(x_t) + [m(x_t) - k(x_{t+1}) - 1],$$

which equals $m(b) - 1$ if $x_t = b$ and -1 otherwise. Thus

$$K'' \geq |\Gamma(x_t, r_t) - \Gamma(x_0, r_0)| = |n_t(b) m(b) - t|,$$

and dividing by $m(b)t$ gives the result. □