Quasirandom Walk with Rotor-Routers

James Propp (U. Wisconsin)

September 22, 2004

(based on an article in progress with Ander Holroyd;

with thanks also to Hal Canary, Matt Cook, Dan Hoey, Michael Kleber, Yuval Peres, Oded Schramm, and Peter Winkler)

For most of this talk, I will be using the sharpest definition of quasirandom, namely, discrepancy $= O(1)$.

That is: a sequence $x_1, x_2, \ldots$ taking its values in some set A will be called quasirandom if for each element $a \in A$ there is a probability $p_a \in [0,1]$ such that the discrepancy

$$|\#\{1 \le k \le N : x_k = a\} - p_a N|$$

is bounded (independently of N).

In particular, every periodic sequence is quasirandom in this sense.

(So are Sturmian sequences and the like, but I'll focus on periodic sequences.)

An $a : b$ rotor has as its internal state a number between 1 and $a + b$ inclusive and as its output either 1 or 2.

Let x be the previous state of the rotor. The current output o and new state x' are determined as follows:

If $x > b$,
  $x' = x - b$ and $o = 1$;
otherwise,
  $x' = x + a$ and $o = 2$.

The output sequence is quasirandom with $p_1 = a/(a + b)$, $p_2 = b/(a + b)$.

Example: $a = 1, b = 2$.

```
States: 1,2,3,1,2,3,..
Outputs: 2 2 1 2 2 1 ...
```

A recurrent quasirandom walk

After I gave an earlier version of my talk, I asked David Griffeath for a good example to try my methods on. He replied:

`For a simple but somewhat subtle positive recurrent chain with computable equilibrium, how about the birth-death process on {0,1,2,...} with reflection at 0 and probabilities (n+4)/(2n+4) and n/(2n+4) of jumps to the left and right, respectively, from any state other than 0 ?'`

The steady-state probability of 0 is exactly 3/8.

Quasirandomize: route the bugs using rotors (one per vertex) instead of random choices.

Start the chain from 0.

Let s_N be the number of times state 0 has been visited by time N.

It does not appear that $|s_N - 3N/8|$ is bounded. But it does appear that $|s_N - 3N/8|$ is $O(N^{.4})$.

Moreover, .4 seems to be the exact right exponent.

Evidence: Let

$p_N^+ = \max\{s_M/M : N/2 \leq M \leq N\}$

and

$p_N^- = \min\{s_M/M : N/2 \leq M \leq N\}$.

Here are the values of

$$\log(p_N^+ - .375)/\log(N)$$

with $N = 10, 100, 1000, \ldots, 10^9$:

```
-0.7433880751
-0.7651331620
-0.6967433616
-0.6579282788
-0.6439365375
-0.6335571521
-0.6252501014
-0.6198259145
-0.6136531941
```

And here are the values of

$$\log(p^-_N - .375)/\log(N)$$

with $N = 10, 100, 1000, \ldots, 10^9$:

```
-0.5351127052
-0.6170428439
-0.5850101801
-0.5900288146
-0.5910136907
-0.5896003336
-0.5926773786
-0.5930909369
-0.5997711121
```

(What we would get from true random walk?)

A 1-dimensional walk involving pi

Here's a 2-stage random walk on $\{1, 3, 5, ...\}$ for which the stationary probability measure at k is $(1/k^2)/(\pi^2/8)$.

At the first time-step, re-randomize the sets $\{1, 3\}$, $\{5, 7\}$, $\{9, 11\}$, ..., using rotors of appropriate bias ($3^2 : 1^2$, $7^2 : 5^2$, $11^2 : 9^2$, ...)

At the next time-step, re-randomize the sets $\{3, 5\}$, $\{7, 9\}$, $\{11, 13\}$, ..., again using rotors of appropriate bias.

Etc. (alternating).

Let's relabel the sites as 0,1,2,3,4,...; the stationary measure of state i is then proportional to $1/(2i+1)^2$. In particular, the stationary measure at 0 is $p = 8/\pi^2$.

At odd time-steps, we re-randomize $\{0, 1\}$, $\{2, 3\}$, $\{4, 5\}$, ...; at even time-steps, we re-randomize $\{1, 2\}$, $\{3, 4\}$, $\{5, 6\}$, ...

If time and position are congruent mod 2, we consider replacing k by $k-1$ using one set of rotors; otherwise we consider replacing k by $k+1$, using another set of rotors.

This time, $s_N - Np$ seems to be $O(N^c)$. where c lies between .634 and .647.

What is c?

A transient quasirandom walk

Start by considering a transient random walk.

States: $0; 1, 2, 3, \ldots$

Transitions: 1 step to the right or 1 step to the left (the former is twice as likely)

Start bug at 1

If we put a bug at 1, what's the probability that it eventually hits 0?

1/2.

Quasirandom version: Each site (other than 0) has one of three colors (green, yellow, red).

If a bug is at a green vertex, the vertex becomes yellow and the bug moves one step to the right.

If a bug is at a yellow vertex, the vertex becomes red and the bug moves one step to the right.

If a bug is at a red vertex, the vertex becomes green and the bug moves one step to the left.

We start a bug at site 1 and see where it goes.

Either the bug reaches vertex 0 after a finite number of steps *or* the bug visits each vertex at most finitely many times. In the latter case, the state of the system "at time infinity" can be defined "by continuity", by saying that the state of each rotor is the state that it had the last time a bug left it. Then the process can start again with a fresh bug at vertex 1.

Let s_N be the number of successes after N trials (where a bug "succeeds" if it eventually reaches 0).

Then $|s_N - N/2|$ is $O(1)$.

Proof idea (words adapted from Winkler's write-up of the Holroyd-Propp proof): The states of the rotors encode a number between 0 and 1 in base 2, using the digits

0 (green light),
$\frac{1}{2}$ (yellow light),
and
1 (red light).

The value of vertex i is 2^{-i} times the digit associated with the color of that vertex.

The bug itself has value 2^{-i} when at position i.

The value of the rotors plus the value of the bug is invariant under updating-the-rotor-and-sliding-the-bug.

When the bug moves to the right from point i, the digit upon which it sat goes up in value by $\frac{1}{2}$; therefore, the value of the rotors increases by $(\frac{1}{2})^{i+1}$, but the bug's own value diminishes by the same amount.

If the bug moves to the left from i it gains in value by $(\frac{1}{2})^{i}$, but the value of the rotors decreases by a whole digit in the ith place to compensate.

The exception is when the bug falls off to the left, in which case both the value of the rotors and the value of the bug drop by $\frac{1}{2}$, for a loss of 1 overall.

When the next bug is added, the total value of the system goes up by $\frac{1}{2}$.

To put it another way, the value of the rotors goes up by $\frac{1}{2}$ if a bug is introduced and disappears to the right; and drops by $\frac{1}{2}$ if a bug is introduced and falls off to the left.

Of course, the total value of the rotors must always lie in the unit interval. If its initial value lies strictly between 0 and $\frac{1}{2}$, the bugs must alternate right, left, right, left; if between $\frac{1}{2}$ and 1, the alternation will be left, right, left, right.

In fact, the state of the system itself alternates: adding three bugs to the system leaves the rotors in the same state as adding one bug.

Let's look at this proof again from a more sophisticated perspective.

For each vertex i, let p_i be the probability that a bug that does random walk starting from i will eventually hit 0.

It can be shown that $p_i = (1/2)^i$.

Consistency check:

$$p_i = (1/3)p_{i-1} + (2/3)p_{i+1}.$$

In fact, the only solutions to this relation are of the form $A(1/2)^i + B$. And once we know that $p_0 = 1$ and $p_i \to 0$ as $i \to \infty$, we can deduce that $A = 1$ and $B = 0$.

The key properties of our scheme for assigning values to rotors and to the bug are:

(1) when the bug is at i, its value is p_i; and

(2) when the rotor at i switches from pointing towards j to pointing towards k, the value of the rotor increases by $p_i - p_k$, to compensate for the change in the value of the bug.

Consistency check: As the rotor makes one complete revolution, the sum of these changes should be zero.

Think of these changes as being arranged in a circle.

Note that (1) and (2) don't determine the values of rotors; only the difference between the values of two different rotor-directions at a vertex.

If we're unwise in the way we assign values to the rotors, the values of the rotors won't be summable. But that's not what really matters, because what's "physically meaningful" are the *differences* between the values of the system. E.g., the difference between the value of the system at the start and the value of the system after some bugs have been added and removed (maybe removed at infinity).

For each vertex i there is a range of values taken on by the rotor. The width of this range is well-defined. If the sum over i of these widths is finite, then everything works fine. (This is sufficient; is it necessary?)

The range of values taken on by the rotor at i is the maximum of $\Sigma\ (p_i - p_k)$, as k ranges over some arc of the circle.

E.g., in our example, at location i, the three numbers in the circle are

$p_i - p_{i+1} = (1/2)^{i+1}$,
$p_i - p_{i+1} = (1/2)^{i+1}$, and
$p_i - p_{i-1} = -(1/2)^i$,
So this range is $(1/2)^i$.

The range of values for the system as a whole is $\Sigma_i\ (1/2)^i = 1$, a finite number.

General construction

Given: Markov chain with discrete state space (i.e., random walk on a graph) with transition probabilities $q_{i,j}$.

Fix source x and targets $y \neq z$. (x is a genuine vertex; y and z can be at infinity.) Assume that with probability 1, random walk from x "hits" either y or z.

For each vertex v, let p_v be the probability that random walk started at v hits y before it hits z.

Thus, p_x is what we want to compute (or estimate); $p_y = 1$, $p_z = 0$.

p is a *harmonic function*: for all v other than y and z,

$$p_v = \Sigma_{v \to w}\, q_{v,w} p_w.$$

Assume that there are finitely many arcs from each vertex and that all transition probabilities are rational.

Then one can create a rotor-router mechanism where bugs get added at x and removed at y and z.

Let s_N be the number of bugs among the first N inserted at x that get removed at y.

Theorem (Holroyd and Propp): If

$$\Sigma_{v \to w} \; |p_v - p_w|$$

is finite (where the sum is over all directed edges), the discrepancy

$$|s_N - Np_x|$$

is $O(1)$, i.e., bounded.

Hitting-time

Theorem (Holroyd and Propp): Suppose the Markov chain has finite state-space. Suppose there is a target y which a bug started at x almost surely hits (under random walk). Let t_N be the total time taken for N bugs, added one at a time at x, to get to y, under rotor-router walk. Then the discrepancy

$$|t_N - NE_x(\text{time to hit } y)|$$

is bounded.

Proof idea: For simplicity, take $x \neq y$. For each vertex v, let h_v be the expected time it takes for random walk started at v to hit y. Then, $h_y = 0$, and for all v other than y,

$$h_v = 1 + \Sigma_{v \to w} \; q_{v,w} h_w.$$

One can use this to devise a rotor-value argument.

What if the state space is infinite? Then we know of cases where boundedness fails, but we suspect that (at least under reasonable hypotheses) the discrepancy is $O(\log N)$.

Schramm's theorem

Theorem (Schramm): Assume the same hypotheses as in the first Holroyd-Propp theorem, but instead of assuming

$$\Sigma_{v \to w} |p_v - p_w|,$$

merely assume that the walk is recurrent. Then $s_N/N \to p_x$; i.e., the discrepancy $|s_N - Np_x|$ is $o(N)$.

This applies to the two-dimensional quasirandom walk I showed you yesterday (the one that computes $\pi/8$).

Before I show you the proof of Schramm's result, I'll need to digress and talk about chip-firing.

Chip-firing, aka sandpile avalanches

In a chip-configuration, each vertex of a directed graph has some number of "chips" (a non-negative integer).

If the number of chips at v is less than the outdegree of v, we say the configuration is stable at v; otherwise, we say that it is unstable at v.

If a chip-configuration is unstable at v, we may "fire" v by sending 1 chip along each of the directed edges emanating from v.

As long as a chip-configuration is unstable somewhere, we can fire it.

There may also be "sinks" that absorb chips but cannot fire.

The strong convergence property

The choices you make (choosing which vertex to fire next) don't matter. That is:

EITHER, no matter what you do, you cannot make the configuration stable,

OR, no matter what you do, the configuration will reach a stable state, and this stable state is independent of the choices you made along the way.

The same is true for rotor-router walk when multiple identical bugs are allowed on the graph. If we want each bug to have a chance to wander until it hits some target (possibly at infinity), we can advance one bug at a time, ignoring all the other bugs while we do so. The resulting configuration of bugs-and-rotors does not depend on the order in which we advance the bugs.

Important caveat: It is vital that the bugs are indistinguishable *and* that we allow each bug to advance until it hits one of the targets. E.g., the strong convergence property does NOT apply in Cooper and Spencer's work, because there each bug advances some fixed number of steps.

Application of strong convergence: To find out what the $\pi/8$-machine will look like after a million trials without running a million trials in succession, put a million bugs at (0,0) at the start. Let each of them move one step. Repeat (absorbing bugs at (0,0) and (1,1) when needed).

When there are four or more bugs at a site, and we send out four bugs in succession, there is no net effect on the rotor. So we can think of this as a chip-firing operation. If there are $n = 4k + r$ bugs at the site (with $0 \leq r \leq 3$), routing them all to neighboring vertices can be thought of as k chip-firing operations (which don't affect the rotor) followed by r rotor-router operations.

Proof of Schramm's theorem

(adapted from Schramm's email):

Consider a recurrent directed graph G (Markov chain) with a starting node s and terminal nodes $t_1, ..., t_k$. Suppose that you want to estimate the various exit probabilities up to an error of ϵ. Let n be sufficiently large so that the probability for the random walk to terminate after n steps is at least $1 - \epsilon/2$. Let m be the product of all numbers j that are out-degrees of vertices in G at distance at most n from s.

Consider dropping m^n particles at s, and letting them go. Whenever a particle already made n steps (without terminating) we freeze it. The other particles keep going. If we do the analogue of the sandpile avalanche process, then at each stage the number of non-frozen particles at a vertex is divisible by the out-degree of that vertex, and so is split equally among the out-edges. Thus, at the end, the number of particles at node t_1 is precisely m^n times the probability that a single particle gets to node t_1 in n or less steps. We have no control over where the remaining frozen particles go when we unfreeze them, but that's at most $\epsilon/2$ times m^n.

Now, if M is a number of the form $km^n + i$, where $i < m^n$ and k is large, then the km^n particles will behave well, and the i remaining particles can be treated as the frozen particles in the previous argument. So, as long as $k > 2/\epsilon$, we gain an additional error in the probability of $\epsilon/2$, which is fine.

Eulerian walkers model

This is the physicists' name for rotor-router walk with random settings of the rotors.

Empirical data (see `cond-mat/9611019`) suggest that EWM is like ordinary random walk in 3 dimensions, and possibly higher dimensions as well, but not in 2 dimensions.

In particular, in two dimensions, the distance of the walker from the origin at time T seems to grow like $T^{1/3}$ rather than $T^{1/2}$ (although, in contrast, the time it takes for a walker to escape from a square region of side L is claimed to grow like L^2).

Is this true?

Whirling tours

From “On Playing Golf With Two Balls” by Dumitriu, Tetali, and Winkler:

“Let T be any tree, possibly with loops. Fix a target vertex t, and let v be any other vertex. Order the edges (including loops) incident to each $u \neq t$ arbitrarily subject to the edge on the path from u to t being last. Now walk from v by choosing each exiting edge in round-robin fashion, in accordance with the edge-order at the current vertex, until t is reached.”

“For example, if the edges incident to some degree-3 vertex u are ordered e_1, e_2, e_3, then the first time u is reached it is exited via e_1, the second time by e_2, the fourth time by e_1 again, etc. We call such a walk a ‘whirling tour’; an example is provided in Figure 3.

Theorem: In any finite tree (possibly with some loops) the length of any whirling tour from v to t is exactly the expected hitting time from v to t.”

Interestingly, this shows that on trees, hitting times are always integers.

Using fewer rotors

Consider random walk on Z^2 where the two equally likely successors of (i, j) are $(i \pm 1, j)$ if $i + j$ is even and $(i, j \pm 1)$ if $i + j$ is odd.

I'm certain we can quasirandomize this walk using rotors $r_{i,j}$ (one rotor for each vertex).

Can we quasirandomize using rotors r_i and r'_j (i.e., a 1-dimensional family of rotors shared by all the sites in the same row, and a 1-dimensional family of rotors shared by all the sites in the same column)?

(Possibly the right thing to do is use rotors r_{i-j} and r'_{i+j} instead.)

Beyond rotors

What's beyond rotors if you want something that's more like simple random walk?

One candidate is the Ehrenfeucht-Mycielski sequence 0, 1, 0, 0, 1, 1, 0, 1, $\ldots$.

To find the next term of the sequence, find the longest suffix s of the current sequence that has occurred earlier. The next term is whichever bit (0,1) did *not* occur following the most recent previous occurrence of s.

This sequence is highly patterned, but it passes many tests for randomness.

Do 0 and 1 each occur with density 1/2?