[8] D. Coppersmith, P. Doyle, P. Raghavan and M. Snir, Random walks on weighted graphs, and applications to on-line algorithms, *Proc. 22nd Annual ACM Symposium on Theory of Computing* (1990), 369-378.

[9] A.K. Chandra, P. Raghavan, W.L. Ruzzo, R. Smolensky, and P. Tiwari, The Electrical Resistance of a Graph Captures its Commute and Cover Times, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, May (1989).

[10] P. Dagum, M. Luby, M. Mihail, and U. Vazirani, Polytopes, permanents and graphs with large factors, *29th Annual IEEE Symposium on Foundations of Computer Science* (1989), 412-421.

[11] C. Derman, *Finite State Markovian Decision Processes*, Academic Press, New York NY 1970.

[12] P.G. Doyle and J.L. Snell, *Random Walks and Electric Networks*, Mathematical Assoc. of America, Washington, DC 1984.

[13] M. Dyer, A. Frieze and R. Kannan, A random polynomial time algorithm for estimating volumes of convex bodies, *Proc. 21st Annual ACM Symposium on the Theory of Computing* (1989), 375-381.

[14] A. Israeli and M. Jalfon, Token management schemes and random walks yield self stabilizing mutual exclusion, *Proc. 9th Annual ACM Symposium on Principles of Distributed Computing*, Quebec City, Canada (1990), 119-131.

[15] M. Jerrum and A. Sinclair, Conductance and the rapid mixing property for Markov chains: the approximation of the permanent resolved, *Proc. 20nd Annual ACM Symposium on Theory of Computing* (1988), 235-243.

[16] A. Karzanov and L. Khachiyan, On the conductance of order Markov chains, Technical Report DCS TR 268, Rutgers University, June 1990.

[17] P. Tetali, Random walks and effective resistance of networks, *J. Theoretical Prob.* No. 1 (1991), 101-109.

There is one further consideration, which leads perhaps to the most intriguing conjecture of all. Let us put two tokens on a graph and let them take random walks, as before; but now suppose the schedule demon is *clairvoyant*—that is, he can see where each token will go, infinitely far into the future. The question is, with this advantage, can he now keep the tokens apart *forever* ? Of course we must ask the question in a probabilistic context, since the tokens may e.g. be headed toward each other along the same path, at the outset. And in some graphs (e.g. a tree or a cycle) it is not hard to see that the demon will come to grief with probability 1. However, we think that on sufficiently large complete graphs the demon will win with probability greater than 0 (in fact K_4 may be big enough). Hence,

Conjecture 3 *Let two tokens begin random walks on a large complete graph. Then with probability* > 0*, the clairvoyant schedule demon can keep them apart forever.*

Acknowledgments. The authors have enjoyed the benefit of useful conversations on this topic with Stuart Haber, Shlomo Halfin, Marc Jalfon, Michal Karonski, and Milena Mihail.

References

[1] D.J. Aldous, Applications of random walks on graphs, *preprint* (1989).

[2] D.J. Aldous, Bibliography: random walks on graphs, *preprint* (1989).

[3] D.J. Aldous, Meeting times for independent Markov chains, *Stochastic Processes and Applications* (1991), to appear.

[4] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász, and C. Rackoff, Random walks, universal traversal sequences, and the complexity of maze problems, *20th Annual Symposium on Foundations of Computer Science*, San Juan, Puerto Rico (October 1979), 218-223.

[5] K. Borre and P. Meissl, Strength Analysis of Leveling-type Networks, *Geodaetisk Institut*, Copenhagen (1974), vol. 50.

[6] G. Brightwell and P. Winkler, Maximum hitting time for random walks on graphs, *J. Random Structures and Algorithms* No. 3 (1990), 263-276.

[7] A.Z. Broder, How hard is it to marry at random? (on the approximation of the permanent), *18th Annual ACM Symposium on Theory of Computing* (1986), 50-58.

which occurs for $|I|(k-|I|) \geq k-1$ of the terms, while the other terms remain constant. Hence the expected value of Φ_k diminishes by at least 1 at every step, as desired.

Note that $M_G(x_i, x_j)$ remains at zero when i and j are both in I, since tokens i and j continue to travel to the same vertex. It is for this reason that we use M_G rather than our original Φ in the definition of Φ_k; the expected value of $\Phi(v,v)$ can in some circumstances jump as v moves, e.g. when v is a hidden vertex.

Now the proof of Theorem 2, suitably amended, shows that the expected time before reducing to a single token is at most the maximum value of Φ_k on G, which is in turn bounded by k times the maximum 2-token meeting time. Thus we have

Theorem 4 *The expected number of steps before tokens on k of the n vertices of a graph reduce to one is at most $(4/27)kn^3$.*

In the worst case for the Israeli-Jalfon protocol we may as well have a token on every vertex, in which case our bound is of order n^4. Indeed, the value of Φ_n can reach order n^4, as in the barbell graph; nonetheless the "total meeting time" in that case is still only order n^3. We do not currently know of any sequence of graphs for which the total meeting time is asymptotically more than the lollipop's $(4/27)n^3$. Hence,

Conjecture 1 *The total expected time for tokens at every vertex of an n-vertex graph to reduce to one is $O(n^3)$.*

Note that it may in some graphs take more time to reduce 3 tokens (say) to 2, in the worst case, than 2 to 1. Define the *collision time* for a collection of tokens on a graph to be the expected number of steps before any two tokens collide, with the schedule demon trying to keep them apart as usual. As an example, let G consist of several copies of a long path with their right-hand endpoints identified; starting with tokens at three left-hand endpoints, the demon moves one until it becomes adjacent to a second, then moves the third exclusively. Asymptotically this takes 50% longer than the largest 2-token meeting time, obtained by hitting one endpoint with a token from another.

However, addition of more tokens, in this case, reduces collision time. We cannot find any case where this is not so, hence

Conjecture 2 *The maximum collision time on any graph is achieved either with 2 or 3 tokens.*

6 Additional Remarks

Our methods may be applied also to meeting times for two tokens when the choice of which token is to move is not made by an adversary. For example, the choice may be random; it may alternate; it might be made by an "angel" who is trying to *minimize* the expected time to collision; the tokens might move simultaneously. In all these cases our upper bound still applies; moreover it is still tight to within a multiplicative constant, since the meeting times for points on opposite sides of the barbell graph depicted in Fig. 3 are always cubic. Throughout

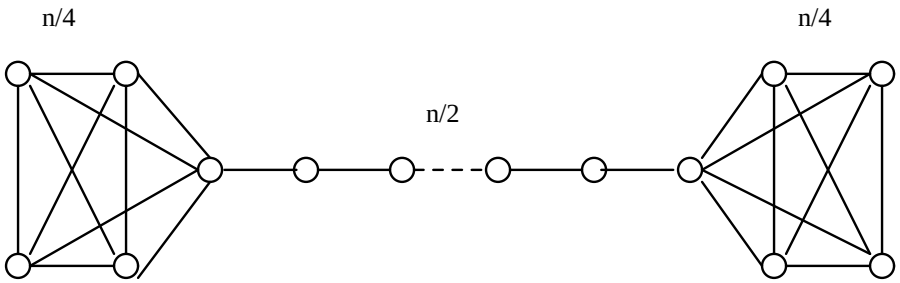

Figure 3: cubic meeting times

this paper we assumed that a token moved from a vertex to any of the neighboring vertices with *equal probability*. We must remark that our main theorem (Theorem 2), in fact, holds true in a more general case – a token, when instructed to move, takes one step of a "general random walk" prescribed by the transition probabilities of a reversible markov chain. Thus the meeting time in this case is still upper bounded by twice the hitting time.

Suppose we reinstate the demon but give him more than two tokens to work with; as in the description of Israeli and Jalfon's self-stabilizing token management scheme above, the rule is that whenever two tokens meet one is eliminated. However, for our purposes it is more convenient to think of tokens not being eliminated but simply "glued together" when they meet; then when there are k tokens at the beginning, we can employ throughout the multi-variable potential

$$\Phi_k(x_1, \ldots, x_k) = \frac{1}{k-1} \sum_{i<j} M_G(x_i, x_j)$$

where the x_i's will cease to be all distinct once collisions begin. Suppose, for example, that tokens corresponding to indices $i \in I$ are currently on vertex v and are designated for movement by the demon; then the expected value of $M_G(x_i, x_j)$ drops by at least 1 when $|\{i, j\} \cup I| = 1$,

This is maximized when $k = \lfloor \frac{n+3}{3} \rfloor$, if n is at least 9. Once again depending on the value of n we get the following bounds:

$n \equiv 0 \pmod 3, k = (n+3)/3$, bound $= (8/27)n^3 + (20/9)n^2 - 18$

$n \equiv 1 \pmod 3, k = (n+2)/3$, bound $= (8/27)n^3 + (20/9)n^2 - 392/27$

$n \equiv 2 \pmod 3, k = (n+1)/3$, bound $= (8/27)n^3 + (20/9)n^2 - 4n/9 - 280/27$.

It follows that irrespective of whether k is even or odd, when $n \geq 13$ we obtain an upper bound of at most

$$\frac{8}{27}n^3 + \frac{8}{3}n^2 + \frac{4}{9}n - \frac{592}{27}.$$

□

Theorem 3 *Maximum meeting time (for $n \geq 13$) is bounded by*

$$(4/27)n^3 + (4/3)n^2 + (2/9)n - 296/27$$

Proof.

$$C_G(x,y) = H_G(x,y) + H_G(y,x)$$

$$M_G(x,y) \leq H_G(x,y) + H_G(y,z) - H_G(z,y) = H_G(y,x) + H_G(x,z) - H_G(z,x)$$

A fortiori,

$$M_G(x,y) \leq H_G(x,y) + H_G(y,z) + H_G(z,y)$$

$$M_G(x,y) \leq H_G(y,x) + H_G(x,z) + H_G(z,x)$$

Thus

$$\begin{aligned} 2M_G(x,y) &\leq H_G(x,y) + H_G(y,z) + H_G(z,y) + H_G(y,x) + H_G(x,z) + H_G(z,x) \\ &= C_G(x,y) + C_G(y,z) + C_G(z,x) \end{aligned}$$

If z is distinct from x and y, then the Lemma gives the Theorem. If $z = x$, then

$$M_G(x,y) \leq H_G(x,y) + H_G(y,x) - H_G(x,y) = H_G(y,x) < C_G(x,y).$$

Select z' different from x and y, and note $C_G(y,z') + C_G(z',x) \geq C_G(x,y)$. So $2M_G(x,y) \leq C_G(x,y) + C_G(y,z') + C_G(z',x)$. Then the Lemma again suffices. □

What is the precise maximum meeting time on an n-vertex graph? Although we have seen that for a given G maximum meeting time may exceed maximum hitting time, we strongly suspect that the extremal case for meeting time is the Brightwell-Winkler lollipop of [6], where maximum meeting time and maximum hitting time are the same (the vertex at the end of the "stick" is hidden).

and by [9], the sum of the round trip times is bounded by

$$2 \times 2k \times [k + (n-k-1)(n-k-2)/2 + 4(n-k-1)].$$

If n is at least 13, this expression is maximized at $k = \lceil \frac{n+3}{3} \rceil$

Thus depending on n we get the following bounds:

$n \equiv 0 \pmod 3, k = (n+3)/3$, bound $= (8/27)n^3 + (8/3)n^2 - 16$

$n \equiv 1 \pmod 3, k = (n+5)/3$, bound $= (8/27)n^3 + (8/3)n^2 + 4n/9 - 740/27$

$n \equiv 2 \pmod 3, k = (n+4)/3$, bound $= (8/27)n^3 + (8/3)n^2 + 4n/9 - 592/27$.

Case II. Suppose $\rho_G(x,y) + \rho_G(y,z) + \rho_G(z,x) = 2k+1$ is odd. Then we can find nonnegative integers a, b, c such that

$$\rho_G(x,y) = a+b+1, \rho_G(y,z) = b+c+1, \rho_G(z,x) = c+a+1, \text{ and } k = a+b+c+1$$

Now construct $k+2$ disjoint nonempty subsets, along with a Residue which may or may not be empty:

$$D(x,0), D(x,1), ..., D(x,a),$$
$$D(y,0), D(y,1), ..., D(y,b),$$
$$D(z,0), D(z,1), ..., D(z,c),$$
$$\text{and Residue} = \text{everything else (may be empty)}.$$

Let F be composed of one vertex from each $D(x,i), D(y,j), D(z,k)$, and let H be everything else (including Residue). The only edges in F are between $D(x,i)$ and $D(x,i+1)$, between $D(x,a)$ and $D(y,b)$, and others by symmetry, totalling $a+b+c+1+1+1 = a+b+c+3 = k+2$. There are at most $(n-k-2)(n-k-3)/2$ edges in H. A vertex in H can be adjacent to at most 4 vertices in F. (Otherwise it is adjacent to at least four subsets among $D(x,i)$ and $D(y,j)$, and thus forms a shortcut between x and y, or similarly between y and z, or z and x.) So at most $4(n-k-2)$ edges exist between H and F.

The total number of edges is at most

$$k + 2 + (n-k-2)(n-k-3)/2 + 4(n-k-2).$$

The sum of the round trips is bounded by

$$2 \times (2k+1) \times [k + 2 + (n-k-2)(n-k-3)/2 + 4(n-k-2)].$$

where m is the number of edges of G and where $R_G(x,y)$ is the effective resistance between x and y (see *Remark* above).

Let $\rho_G(x,y)$ denote the distance between x and y in G, that is the number of edges in a shortest path between x and y. Then $R_G(x,y) \leq \rho_G(x,y)$, so

$$C_G(x,y) + C_G(y,z) + C_G(z,x) \leq 2m[\rho_G(x,y) + \rho_G(y,z) + \rho_G(z,x)]$$

Case I. Let $\rho_G(x,y) + \rho_G(y,z) + \rho_G(z,x) = 2k$ be even.
Then there are nonnegative integers a, b, c such that

$$\rho_G(x,y) = a+b,\ \rho_G(y,z) = b+c,\ \rho_G(z,x) = c+a,\ \text{and}\ k = a+b+c$$

(note that a, b, c are nonnegative, owing to the triangle inequality for distances).
Partition the n vertices into the following $a+b+c+1$ nonempty subsets:

$$D(x,0), D(x,1), ..., D(x,a-1),$$
$$D(y,0), D(y,1), ..., D(y,b-1),$$
$$D(z,0), D(z,1), ..., D(z,c-1),$$
$$\text{and Residue} = \text{everything else},$$

where $D(x,i)$ is the set of vertices t with $\rho_G(x,t) = i$. Disjointness follows from the distances: if $D(x,i) \cap D(y,j)$ is nonempty for $i < a, j < b$, then $\rho_G(x,y) \leq i+j < a+b$. Further, $D(x,i)$ and $D(y,j)$ must be nonadjacent, since otherwise $\rho_G(x,y) \leq i+j+1 < a+b$. Since G is connected, this implies that Residue is nonempty.

The only edges that can exist are between $D(x,i)$ and $D(x,i+1)$, between $D(x,a-1)$ and Residue, similarly for y and z, and within the subsets themselves.

Let F be composed of one vertex from each of the $k+1 = a+b+c+1$ subsets, and let $H = G - F$ be the remaining $n-k-1$ vertices. The number of edges among F is at most k. The number of edges among H is at most $(n-k-1)(n-k-2)/2$. The number of edges between H and F is at most $4(n-k-1)$. (Each member of H can be adjacent to at most 4 members of F, and can achieve 4 only if it is in Residue. If it is adjacent to 5 members of F, it is adjacent to 4 among $D(x,i), D(y,j)$, and Residue, and thus gives a shortcut between x and y.)

Thus the number of edges is bounded by

$$k + (n-k-1)(n-k-2)/2 + 4(n-k-1),$$

course be 0 since $\Phi(x,x) \geq 0 = M_G(x,x)$. We may assume the demon's strategy with tokens on x and y is to move x. Then

$$\begin{aligned} M_G(x,y) &= \Phi(x,y) + \beta \\ &= 1 + \Phi(\bar{x},y) + \beta \\ &\geq 1 + M_G(\bar{x},y) = M_G(x,y); \end{aligned}$$

but again at least one neighbor of x must be closer to y than x was, so the inequality is strict and the contradiction proves the theorem. □

In view of the Brightwell-Winkler bound on hitting time [6] we can now readily obtain a cubic bound of $(8/27)n^3$ for $M_G(x,y)$. However, more careful analysis (below) will obtain the right constant. Before proceeding, however, let us rescue some of the virtue of the hitting time strategies, by showing that their behavior with respect to hidden vertices is correct.

Corollary 1 *For any hidden vertex t of G, the demon has a (pure) optimal strategy which never moves a token off t.*

Proof. For any other vertex x, we have

$$M_G(x,t) \leq \Phi(x,t) = H_G(x,t)$$

so the demon cannot achieve a higher expected time to finish than he gets by merely waiting for the token from x to hit t. □

The *commute time* $C_G(x,y)$ between vertices x and y of a graph G is defined simply by

$$C_G(x,y) = H_G(x,y) + H_G(y,x).$$

Lemma 4 *In a graph G on n vertices, $n \geq 13$, for any three distinct vertices (x,y,z), we have*

$$C_G(x,y) + C_G(y,z) + C_G(z,x) \leq \frac{8}{27}n^3 + \frac{8}{3}n^2 + \frac{4}{9}n - \frac{592}{27}$$

Proof. By Chandra et al. [9] we know that

$$C_G(x,y) = 2mR_G(x,y)$$

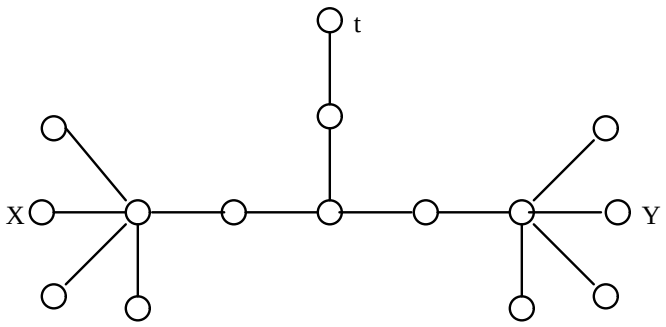

Figure 2: meeting time more than max. hitting time

twice the maximum hitting time.

5 The Main Theorem

Theorem 2 *Let G be any connected, undirected graph, and let t be a hidden vertex of G. Then for every pair x, y of vertices of G,*

$$M_G(x,y) \leq H_G(x,y) + H_G(y,t) - H_G(t,y).$$

Proof. We define a *potential function* Φ in accordance with the right-hand side of the above inequality; thus

$$\begin{aligned} \Phi(x,y) &= H_G(x,y) + H_G(y,t) - H_G(t,y) \\ &= H_G(y,x) + H_G(x,t) - H_G(t,x) \end{aligned}$$

on account of Lemma 2. Thus Φ is symmetric and no matter which token the demon decides to move, its expected value after the move will decline by 1:

$$\Phi(x,y) = 1 + \Phi(\bar{x},y) = 1 + \Phi(x,\bar{y}).$$

Since Φ is non-negative (on account of t being a hidden vertex) the statement of the theorem is already plausible, but to make it rigorous an argument similar to that used in Lemma 1 seems to be the most elementary route.

Assume, accordingly, that the theorem is false and let β be the maximum value of $M_G(x,y) - \Phi(x,y)$. Among all pairs x, y realizing β choose one of minimum distance, which cannot of

If the demon follows a degree strategy, then the distance token takes a completely normal random walk on G', with absorbing state v'_0 and reflecting state v'_{n-1}. The expected duration of the game is precisely the absorbing time from the distance token's initial position. If, however, at any pair v_0, v_i or v_i, v_{n-1} the demon has positive probability of moving the token at the endpoint, then the probabilities along the edges leading from the corresponding vertex on G' are skewed toward v'_0, decreasing the expected time to finish. □

For our next example we make only a small departure from the path; let G be the graph depicted in Fig. 1, consisting of a path on nine vertices with a pendant edge attached to the middle vertex v_5.

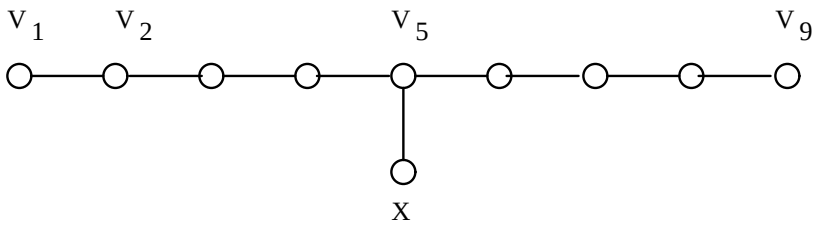

Figure 1:

In this graph $H_G(v_2, x) = 32$ and $H_G(x, v_2) = 40$ so if the demon follows a hitting time strategy, he will move from x when the tokens sit at x and v_2. This strategy will net him an expected meeting time of $M(x, v_2) = 46\frac{6}{7}$ steps from that position.

However, it turns out that the demon's optimal strategy never moves a token at x unless the other token is on v_1 or v_9. If the demon follows this rule, and otherwise moves the token which is closer to v_5, he achieves $M(x, v_2) = 52$.

For our last example we consider the tree T pictured in Fig. 2. Here we have the following hitting times: $H_T(x, y) = 84$; $H_T(y, t) = 73$; and $H_T(t, y) = 67$. Note that t is the unique hidden vertex here. Although the maximum hitting time in T is indeed 84, we can easily verify that the maximum *meeting* time from x to y is more than 87.

To see this, let the demon proceed as follows: move the token which begins at x and continue moving it until it hits y or t. If it hits t first—which happens with probability $> 50\%$—the demon switches horses by moving the token at y, thus gaining the difference $73 - 67 = 6$ between $H_T(y, t)$ and $H_T(t, y)$.

We shall see, however, that the maximum meeting time in a graph can never be more than

looks reasonable to guess that the demon always has a hitting time strategy which is optimal, but this turns out not to be the case. However, from the next lemma we can deduce the fact that there is a pure hitting time strategy whose tournament is transitive.

Lemma 3 *On any graph G the vertex-relation given by*

$$u \leq v \text{ iff } H_G(u,v) \leq H_G(v,u)$$

is transitive, i.e. constitutes a pre-order on the vertices of G.

Proof. Immediate from the equality of Lemma 2. For us the important consequence of Lemma 3 is that there is always a vertex t which is *minimal* in this pre-order, and thus satisfies $H_G(v,t) \geq H_G(t,v)$ for every other vertex v of G. Such a vertex t will be called *hidden*. (As an example, the reader may verify that a vertex of a *tree* is *hidden* just if its average distance to other vertices of the tree is maximum.)

4 Examples

In tuning one's intuition with regard to hitting times and meeting times, it is helpful to look at some examples. In the following cases hitting times can be verified by the reader using Tetali's formula or by solving simple equations.

Let us begin with a simple but already counterintuitive case: what should the demon's strategy be on a path? It seems perhaps that he should always move the token nearest the center, but in fact it makes no difference what he does as long as he never moves a token from an end-point unnecessarily.

A strategy for the demon is termed a *degree strategy* if whenever the tokens are on vertices of different degrees, he moves the token from the vertex of larger degree. Thus on the path, a degree strategy never moves a token off an endpoint unless it has to.

Theorem 1 *On a path a strategy for the demon is optimal iff it is a degree strategy.*

Proof. Let G be the path on vertices $v_0, \ldots, v_{n-1}$. As a pair of tokens moves within G, let us imagine a single "distance token" moving around a second copy G' of G according to the following rule: if the tokens on G are at vertices v_i and v_j, then the location v'_k of the distance token is given by $k' = |i - j|$.

The next lemma establishes a critical relation among hitting times.

Lemma 2 *Let x, y and z be vertices of a connected, undirected graph G. Then*

$$H_G(x,y) + H_G(y,z) + H_G(z,x) = H_G(x,z) + H_G(z,y) + H_G(y,x).$$

Proof. Essentially, this equality is a consequence of the reversibility of the Markov chain for random walks on an undirected graph. Note that the left-hand side of the equation in the lemma is the expected time for a random walk to go from x to y, then to z and back to x, and similarly for the right.

Now fix a number r and begin a random walk at x, ending when x is reached again for the rth time. Let $x, v_1, v_2, \ldots, v_k, x$ be the outcome of the walk; its probability is

$$\frac{1}{d(x)} \prod_{i=1}^{k} \frac{1}{d(v_i)}$$

where $d(u)$ is the degree of the vertex u, i.e. the number of edges incident to u. But this value is of course the same as the probability of the reverse walk $x, v_k, v_{k-1}, \ldots, v_1, x$. Now, we claim that the number of x-to-y-to-z-to-x tours in one of these walks is the same as the number of x-to-z-to-y-to-x tours in its reverse; to see this, note that the greedy algorithm for finding such tours starting from the left is optimal and thus yields at least as many such tours as one can find by listing x-to-z-to-y-to-x tours from the right; the symmetric argument establishes equality. It follows that the expected lengths of the two types of tours from x to x are the same, proving the lemma. □

Remark. For those readers who are accustomed to thinking of random walks in terms of electrical circuits, we note that the lemma follows also from the following formula which appears in Tetali [17]:

$$H_G(x,y) = mR_G(x,y) + \frac{1}{2}\sum_z d(z)[R_G(y,z) - R_G(x,z)]$$

in which m is the number of edges of G, and $R_G(u,v)$ is the effective resistance between u and v when G is regarded as an electrical network with a unit resistor on each edge.

A strategy S for the demon will be called a *hitting time strategy* if whenever the tokens are on x and y with $H_G(x,y) > H_G(y,x)$, S requires that the demon move the token from x. It

For convenience we adopt the following convention: if f is any real-valued function on the vertices of a graph, then $f(\bar{v})$ is defined to be the average of $f(u)$ over all neighbors u of v. Thus, for example, hitting time satisfies $H_G(x,y) = 1 + H_G(\bar{x},y)$ for all distinct x and y.

3 Lemmas

Lemma 1 *On any graph G the demon has a pure optimal strategy.*

Proof. For any two distinct vertices x and y, let $S(x,y)$ be a strategy maximizing $M^{S(x,y)}(x,y)$. Define a tournament T by letting x be the winner over y when, given a starting position with tokens at x and y, $S(x,y)$ moves the token at x; similarly for y. If either token may be moved, we assign a winner arbitrarily. We claim that the pure strategy S corresponding to the tournament T is optimal.

If not, let $\alpha > 0$ be the maximum value of $M_G(x,y) - M^S(x,y)$ and of all pairs x, y attaining this discrepancy, choose one of minimum distance. Assume x beats y in T and that with tokens starting at x and y, $S(x,y)$ moves x with probability $p > 0$. Then

$$M^{S(x,y)}(x,y) = 1 + pM_G(\bar{x},y) + (1-p)M_G(x,\bar{y})$$

since $S(x,y)$ is supposed to be optimal at x, y; further we must have

$$M^{S(x,y)}(x,y) = 1 + M_G(\bar{x},y)$$

else moving the token at y would be a superior strategy. But then

$$\begin{aligned} M^{S(x,y)}(x,y) &= 1 + M_G(\bar{x},y) \\ &\leq 1 + M^S(\bar{x},y) + \alpha \\ &= M^S(x,y) + \alpha \\ &= M_G(x,y) \\ &= M^{S(x,y)}(x,y). \end{aligned}$$

The catch is that the inequality in the middle is strict, since one of the neighbors z of x must be at smaller distance to y than x was; thus $M_G(z,y) - M^S(z,y) < \alpha$, or $z = y$ so that $M_G(z,y) = 0$. Either way, the contradiction proves the lemma. □

We remark that Lemma 1 can also be proved by formulating it in terms of Markov decision processes and applying the general theory, as found, e.g., in [11].

2 Notation and Preliminaries

The *hitting time* $H_G(x,y)$ from x to y is defined to be the expected number of steps for a random walk on G beginning at vertex x to reach vertex y for the first time. Thus, for example, if x and y are at opposite ends of a path on n vertices then we have a "standard" random walk with reflecting barrier, and any of a number of arguments shows that the hitting time from x to y is precisely $(n-1)^2$.

One might at first be tempted to guess that this represents the largest possible hitting time in an n-vertex graph, but in fact it has been known for many years that there are n-vertex graphs G (barbells, for example) with vertices x, y such that $H_G(x,y)$ is $\Omega(n^3)$. The precise upper bound for $H_G(x,y)$ was found by Brightwell and Winkler [6]; the unique extremal graph is a "lollipop" consisting of a clique on $\lfloor (2n+1)/3 \rfloor$ vertices with a path on the remaining $n-m$ vertices attached at one end. The start vertex x is in the clique, and the end vertex y is of course at the far end of the path. The hitting time turns out to be precisely

$$\frac{4}{27}n^3 - \frac{1}{9}n^2 + \frac{2}{3}n - 1 + c$$

where $c = 0, -\frac{2}{27}$, or $-\frac{2}{9}n + \frac{14}{27}$ according as $n \equiv 0, 1$ or $2 \bmod 3$ respectively.

This same value is thus a *lower* bound for the worst-case value of $M_G(x,y)$, obtainable if the tokens begin as described above and the demon moves only the token which started at x (this is in fact the demon's best strategy on the lollipop). However, as we shall see it is *not* generally the case that on a fixed graph G, the maximum meeting time is bounded by the maximum hitting time.

Let us now fix an n-vertex graph G. A (possibly randomized) *strategy* S for the demon on G consists of instructions which tell the demon, for each possible starting position and progress of the game so far, and each possible value of some random variable, which token he should move. We let $M^S(x,y)$ be the expected number of moves until the tokens meet, when the game is begun with tokens at x and y, and the demon follows strategy S.

An *optimal* strategy is one for which $M^S(x,y)$ is maximal, that is, equal to $M_G(x,y)$, for every x and y; a *pure* strategy is one in which the demon's choice at each turn depends only on the current locations of the tokens. Note that a pure strategy is equivalent to a *tournament* on the vertices of G; from each pair u, v of vertices the "winner" is the vertex from which the demon will move the token when the tokens are located on u and v.

becomes: what is the *expected* number of such activations before the tokens have collapsed to one?

If said time is polynomial in n then it follows from Markov's inequality that in polynomial time one can make the probability of stabilization as close to 1 as desired; furthermore, it clearly suffices to consider the case where the system begins with just two tokens. Accordingly, we define the *meeting time* $M_G(u,v)$ to be the expected number of moves before tokens placed initially at vertices u and v of G collide, given optimal (delaying) play by the demon in deciding at each step which token shall move.

Israeli and Jalfon noted that if G is an n-cycle, that is, if the processors are placed in a ring, then it makes no difference which of the two tokens is moved; hence $M_G(u,v)$ is bounded by the "hitting time" (definitions below) across the cycle, which is about $n^2/4$.

However, in a general connected, undirected, n-vertex graph G they were able only to get the exponential upper bound

$$M_G(u,v) = \mathrm{O}\left((\Delta-1)^{D-1}\right)$$

where Δ is the maximum degree of G and D is its diameter.

Our main contribution has been to obtain a polynomial upper bound in the general case, namely

$$M_G(u,v) \leq \frac{4}{27}n^3 \quad \text{plus lower order terms .}$$

Our primary technique is the use of a somewhat odd-looking potential function to bound the meeting time; the function is designed to drop by one in expected value no matter which token is moved. Using distance and electrical resistance arguments, we are then able to reduce the constant to $4/27$ which is best possible.

Since the demon is an adversary, our results apply also to the case where the decision as to which token moves next is random, and to the case where the tokens move simultaneously. A recent result of Aldous [3] implies that in the latter case meeting time is bounded by a (large) constant times the maximum hitting time. Thus our results generalize Aldous' to cover the adversarial case, and it happens that we also lower Aldous' constant to 2. However, it should be noted that we do *not* determine the adversary's optimal strategy in the general case.

approximate differences. Broder [7] and Jerrum and Sinclair [15] made use of random walks on graphs to obtain the first randomized polynomial-time algorithm for approximating the value of the permanent of a matrix; see also Dagum, Luby, Mihail and Vazirani [10]. Random walks were used by Dyer, Frieze and Kannan [13] to estimate the volume of a convex body, and by Karzanov and Khachiyan [16] to sort partial orders when comparisons are expensive. Recently, Coppersmith et al. [8] have found an application of random walks to on-line algorithms.

Aldous [1] gives many other contexts in which random walks on graphs arise, and a valuable bibliography [2] compiled by the same author lists numerous additional references on the subject.

In this work we are motivated by the work of Israeli and Jalfon [14] on self-stabilizing token management schemes. A protocol for a distributed computing network is said to be *self-stabilizing* if, no matter what state it is begun in (or perturbed to), it eventually enters a "legal" state and resumes normal operation.

In a token management scheme, only one processor at a time is supposed to be "enabled" to change state or perform some particular task. This processor is said to possess the token, the token being an abstract object which is passed from processor to processor like the baton in a relay race.

The obvious problems in designing a self-stabilizing token management scheme are in recovering from (a) a situation in which no token is present, and (b) a situation in which several tokens are present. The former can be neatly avoided by allowing a processor to use information from its neighbors to determine whether it has a token. For example, in Israeli and Jalfon's scheme each processor has a special token-management register and a processor deems itself to possess the token just when the value in its register is at least as great as the value in any of its neighbor's.

Then of course if x is the highest value in any processor's token management register, all processors holding x possess a token—as do other processors which hold "local" maxima.

To reduce the number of tokens to one, the tokens are passed randomly to neighbors and whenever two or more collide, they merge to become a single token. The tokens do not, however, move simultaneously; the processors are not synchronized but are fired at the will of a "demon" who, in the worst case, is trying to delay stabilization. Naturally we must assume the demon has to activate token-possessing processors from time to time, so the question

Collisions among Random Walks on a Graph

Don Coppersmith[1], Prasad Tetali[2] and Peter Winkler[3]

Abstract

A token located at some vertex v of a connected, undirected graph G on n vertices is said to be taking a "random walk" on G if, whenever it is instructed to move, it moves with equal probability to any of the neighbors of v. We consider the following problem: suppose that *two* tokens are placed on G, and at each tick of the clock a certain demon decides which of them is to make the next move. The demon is trying to keep the tokens apart as long as possible. What is the expected time M before they meet?

The problem arises in the study of self-stabilizing systems, a topic of recent interest in distributed computing. Since previous upper bounds for M were exponential in n, the issue was to obtain a polynomial bound. We use a novel potential function argument to show that in the worst case $M = \left(\frac{4}{27} + o(1)\right) n^3$.

1 Introduction

Let G be a connected graph on n vertices, and let v be a fixed vertex of G. A *random walk* on G, beginning at v, is a stochastic process whose state at any time t is given by a vertex of G; at time 0 it is at vertex v, and if at time t it is at vertex u, then at time $t+1$ it will be at one of the neighbors of u, each neighbor having been chosen with equal probability.

The random walk thus constitutes a Markov chain, with state transition probability $p_{x,y} = 0$ if y is not adjacent to x and $p_{x,y} = 1/d(x)$ if y is adjacent to x and x has degree $d(x)$. The Markov chain will be irreducible (unless G is bipartite) and it is easily verified that its stationary distribution π satisfies $\pi_x = d(x)/2m$ where m is the number of edges of G.

Thus, we have that in the limit, the probability of being at any particular vertex is proportional to its degree—regardless of the structure of G. This remarkable fact is the key to numerous applications.

In Aleliunas et al. [4], random walks are used to establish the existence of short universal sequences for traversing graphs; in Doyle and Snell [12] they are elegantly associated with electrical networks; in Borre and Meissl [5] they are employed to estimate measurements given by

[1]T.J. Watson Research Center, IBM, Yorktown Heights, NY 10598
[2]AT&T Bell Labs, Murray Hill, NJ 07974
[3]Bellcore, Morristown, NJ 07962