Collisions of Multiple Tokens Taking Random Walks on a Graph

Keith Carlson *

March 5, 2013

Abstract

Given a graph $G = (V, E)$ we consider the following problem. Suppose that two distinct tokens are placed on vertices x, $y \in V$. At each step in time some adversarial demon selects one of the tokens to move. The selected token moves according to the rules of a random walk. If the demon's goal is to keep the tokens from colliding for as many steps as possible, what is the expected time $M(x, y)$ before the tokens finally meet?

This problem is relevant to the study of self-stabilizing systems in distributed computing. In the worst case $M(x, y) = (\frac{4}{27} + o(1))n^3$ which implies that a certain algorithm for token management in distributed computing system will self-stabilize in polynomial time.

1 Introduction

Let $G = (V, E)$ be a connected graph with n vertices. Suppose that two tokens are placed on vertices $x, y \in V$. At each step some demon selects one of the tokens to move. The selected token then moves according to the rules of a random walk. If the demon's goal is to keep the tokens apart for as many steps as possible and he is clever enough to play optimally, then how long do we expect it to take for the tokens to meet? We will call this value meeting time and denote it by $M(x, y)$. An upper bound for meeting time will clearly imply an upper bound for the case where the token which moves at a given step is chosen randomly.

This model arises from the study of self-stabilizing token management schemes. In distributed computing *self-stabilizing* means that no matter what state a system finds itself in, it will always return to some acceptable configuration and continue its operation. In a token

*Don Coppersmith, Prasad Tetali, and Peter Winkler produced the results covered in this paper management scheme, there is only one active processor at any given time. The active processor can do some computation, and then designate another processor as active. This can be thought of as passing a token. We will design our system such that it can be thought of as a graph. Each processor will have some set of neighboring processors such that if processor A is a neighbor of B, then B is also a neighbor of A. The token management algorithm we will consider will just pass the token according to the rules of a random walk on this graph.

In order for a token management scheme to be self-stabilizing, it must be able to correct situations where there are no tokens in the system, and situations where there are 2 or more tokens. Collisions among random walks play a role only in recovery from multiple tokens, so we will discuss only this case. If a processor is ever active and receives a token from a neighbor then we just say that the two tokens are combined, and we have successfully reduced the number of active processors in the system. An upper bound for meeting time against an adversarial demon is, then, an upper bound for self-stabilization time of our token management scheme.

We will show that meeting time is bounded polynomially in n. Specifically we show that for any graph, $M(x, y) \leq (\frac{4}{27})n^3$ plus some lower order terms.

2 Preliminaries, Notation, Examples

Let hitting time from x to y be denoted by $H(x, y)$. Hitting time on a graph implies a lower bound for meeting time, because the demon could simply choose to move the token which started at vertex x until it reaches y. Since this is a valid strategy, and meeting time is defined as the optimal strategy for the demon, we have that

$$M(x, y) \geq \max(H(x, y), H(y, x)).$$

We know that the the maximum hitting time on a graph occurs on a "lollipop" graph [2] and is

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

where $c = 0, -\frac{2}{27}$, or $-\frac{2}{9}n + \frac{14}{27}$ depending on whether $n \equiv 0$, 1, or 2 modulo 3. This shows that the worst meeting time is at least cubic in n. On a lollipop graph the demon's optimal strategy is, indeed, to only ever move the token which started in the clique, but is it always the case that the demon's optimal strategy only moves one token? Let's look at some examples.

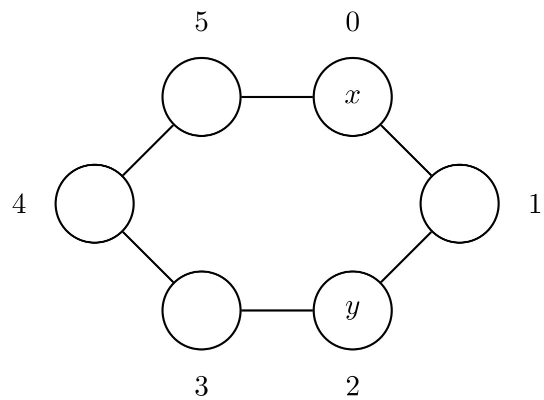

Figure 1: Meeting time on a cycle

In Figure 1 we see a cycle on 6 vertices with 2 tokens starting at vertices x and y. What is $M(x, y)$? We can calculate that $H(x, y) = H(y, x) = 8$ and we have already seen that $M(x, y)$ is at least the larger of these two hitting times. Is there any strategy the demon can employ to make the expected collision time on this graph larger than 8? It turns out that in this example there is not. The intuition here is that it makes no difference which of the two tokens on a cycle is moved. They will always make the distance in one direction smaller by 1 and the distance in the other direction greater by 1. Since it doesn't matter which token moves at each step we can just say that we always move the token which started at x, which, in this case, means that $M(x, y) = H(x, y) = 8$.

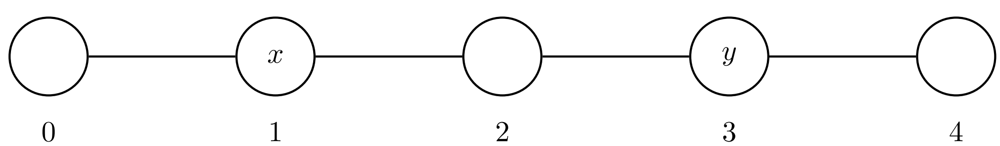

Figure 2: Meeting time on a path

In Figure 2 we see a path on 5 vertices with 2 tokens starting at vertices x and y. We can calculate the hitting times to find that $H(x, y) = H(y, x) = 8$, so $M(x, y)$ is at least 8. In this case the demon can actually achieve a meeting time higher than 8. We will calculate the expected collision time if the demon chooses the strategy to move x until it hits either y or the left endpoint of the path, then moves only y until a collision occurs.

Token x has probability $\frac{1}{3}$ of reaching y first and $\frac{2}{3}$ of reaching 0 first. The expected time for it to reach one of these points is $2 = (3-1)(1)$. Now, if we reached y we are done, but if we reached 0, then the demon now moves token y until it also reaches 0. This takes $15 = 4^2 - 1^2$ steps in expectation. The expected time before a collision occurs if the demon uses this strategy is then equal to $2 + \frac{2}{3}(15) = 12$. Clearly the demon has a strategy which makes $M(x,y) > \max(H(x,y), H(y,x))$.

This result shows us that although on the lollipop and the cycle the demon's strategy is irrelevant, there are cases where he can increase meeting time by playing cleverly. The strategy described above for the path is in fact optimal for the demon, and we will show this later on. In general however, we are interested in finding an upper bound for $M(x,y)$ on any graph.

3 Result

Let us call a strategy that may be adopted by a demon S. We will let $M^S(x,y)$ be the expected meeting time if the demon uses strategy S. Note that for all S we have that $M^S(x,y) \leq M(x,y)$ by definition. We will call S optimal if $\forall x, y\ M^S(x,y) = M(x,y)$.

Let us call a strategy S pure if the demon's choice of which token to move depends only on the positions of the two tokens. We notice that a pure strategy implies a tournament of all vertices where the winner is the vertex from which the demon will move a token. You can think of this tournament as an n by n matrix which defines the winner between any pairing of vertices.

For a function f on a vertex v of a graph, we will let $f(\overline{v})$ be the average $f(u)$ over all neighbors u of v. It is simple to see that $H(x,y) = 1 + H(\overline{x}, y)$.

Lemma 3.1. *On any graph the demon has a pure optimal strategy.*

Proof. Let $S(x,y)$ be a strategy that maximizes $M^{S(x,y)}(x,y)$. This means that $S(x,y)$ is optimal if the tokens are placed at vertices x and y, but makes no guarantee about optimality in other situations. We will define a tournament T by saying that x beats y if $S(x,y)$ will move x. We claim that the strategy S which is defined by T is optimal. Since S is pure, proof of this claim will prove our lemma.

Assume to reach a contradiction that S is not optimal. Let $\alpha > 0$ be the maximum value of $M(x,y) - M^S(x,y)$ over all $x, y \in V$, and of all the x and y which attain this α we will choose the pair of minimum distance on the graph. We assume without loss of generality that x beats y in T. Since $S(x,y)$ is optimal at (x,y),

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

It appears that we have shown the unsurprising result that $M^{S(x,y)}(x,y) \leq M^{S(x,y)}(x,y)$, however the inequality in the second line is actually strict because at least one neighbor of x is closer to y than x itself is. Since we selected the x and y with minimum distance that achieved α, this particular neighbor of x must differ from the optimal strategy by less than α. This means we have actually shown that $M^{S(x,y)}(x,y) < M^{S(x,y)}(x,y)$, which is a clear contradiction and proves our lemma. $\square$

Lemma 3.2. *H(x,y) + H(y,z) + H(z,x) = H(x,z) + H(z,y) + H(y,x)*

Proof. We have seen this proof idea in class. $\square$

Lemma 3.3. *The relation given by*

$$x \leq y \iff H(x,y) \leq H(y,x)$$

is transitive (it defines a pre-order of the vertices).

Proof. Suppose $H(x,y) \geq H(y,x)$ and $H(z,x) \geq H(x,z)$.

$$H(x,y) + H(y,z) + H(z,x) = H(x,z) + H(z,y) + H(y,x)$$

$$H(y,z) + H(z,x) \leq H(x,z) + H(z,y)$$

$$H(y,z) \leq H(z,y)$$

$\square$

We can now give an ordering to the vertices of a graph according to this relation. We are concerned with the smallest vertex according to this relation and will call such a vertex *hidden*. If a vertex t is hidden, then by our definition of this relation we know that $H(v,t) \geq H(t,v)$ for every other vertex v in our graph.

Let us call a strategy a *degree strategy* if whenever the tokens are on vertices of different degrees, the demon moves the token on the larger degree vertex. The strategy we previously considered on a path was in fact a degree strategy. We now present a relevant theorem.

Theorem 3.4. *On a path a strategy for the demon is optimal if and only if it is a degree strategy.*

Proof. With tokens at x and y on a path we consider at all times the distance between x and y and will call this distance d. Clearly a collision has occured and our walk is complete only when $d = 0$. Let us then imagine letting a single token walk our path whose position is equal to d. It starts at position $|x - y|$, and 0 is our absorbing state. d is only on vertex n when x and y are at opposite endpoints, and in this case clearly the next move of either x or y will decrease d by one. When d is not at vertex n then either x or y must be on a vertex of degree 2. Moving a token on a vertex of degree 2 either increases or decreases d by one, each with equal probability. Moving a token on a vertex of degree 1, however, always decreases d by one. By our previous knowledge of hitting time on a path it is clear that when $d \neq n$, $H(d,0) > 1 + H(d-1,0)$. The optimal strategy for the demon then on a path must always be to move a token from a vertex of higher degree. □

Theorem 3.5. *Let t be a hidden vertex in G. $\forall x, y \in V$*

$$M(x,y) \leq H(x,y) + H(y,t) - H(t,y)$$

Since hitting time is always non-negative, this theorem will immediately imply that for any graph, the meeting time will be no more than twice the maximum hitting time.

Proof. Define $\Phi(x,y) = H(x,y) + H(y,t) - H(t,y)$. By Lemma 3.2 this is equal to:

$$H(y,x) + H(x,t) - H(t,x)$$

$$\implies \Phi(x,y) = \Phi(y,x)$$

$$\implies 1 + \Phi(\overline{x}, y) = 1 + \Phi(x, \overline{y})$$

Assume to reach a contradiction that Theorem 3.5 is false. Let β be the maximum value of $M(x,y) - \Phi(x,y)$. Amongst all x and y achieving this β choose the pair of minimum distance.

Assume w.l.o.g. that the demon moves the token on vertex x.

$$\begin{aligned} M(x,y) &= \Phi(x,y) + \beta \\ &= 1 + \Phi(\overline{x}, y) + \beta \\ &\geq 1 + M(\overline{x}, y) \\ &= M(x,y) \end{aligned}$$

As in the proof of Lemma 3.1, this appears to just show the simple result that $M(x,y) \geq M(x,y)$. However, we have once again selected x and y to be of minimum distance achieving our β difference. This means that one of the neighbors of x is closer to y than x was which means that it differs by less than β. The inequality is then strict and we have reached the absurd claim that $M(x,y) > M(x,y)$. $\square$

This theorem implies that a graph can never have a meeting time of more than twice the maximum hitting time. Since we know that the maximum hitting time on a graph to be $\frac{4}{27}n^3$ plus some lower order terms [2], we can immediately say that $M(x,y) \leq \frac{8}{27}n^3$ plus some lower order terms. As we will soon see, this is not the best upper bound available.

Let us denote commute time between x and y by $C(x,y)$.

$$C(x,y) = H(x,y) + H(y,x)$$

In [1] they prove the following lemma.

Lemma 3.6. *In a graph on n vertices, $n \geq 13$, for any three distinct vertices (x, y, x)*

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

We use this result to obtain a better upper bound on $M(x,y)$.

Theorem 3.7. *Maximum meetng time for $n \geq 13$ is bounded by*

$$\frac{4}{27}n^3 + \frac{4}{3}n^2 + \frac{2}{9}n - \frac{296}{27}$$

Proof.

$$C(x,y) = H(x,y) + H(y,x), \qquad \text{(by definition)}$$

$$M(x,y) \le H(x,y) + H(y,z) - H(z,y) = H(y,x) + H(x,z) - H(z,x). \qquad \text{(using Theorem 3.5, Lemma 3.2)}$$

Since hitting time is always non-negative, it follows that

$$M(x,y) \le H(x,y) + H(y,z) + H(z,y),$$

$$M(x,y) \le H(y,x) + H(x,z) + H(z,x).$$

Therefore, adding the above two inequalities

$$\begin{aligned} 2M(x,y) &\le H(x,y) + H(y,x) + H(z,y) + H(y,x) + H(x,z) + H(z,x), \\ &= C(x,y) + C(y,z) + C(z,x). \end{aligned}$$

By selecting z to be any vertex other than x or y and applying Lemma 3.6, we now conclude that

$$M(x,y) \le \frac{4}{27}n^3 + \frac{4}{3}n^2 + \frac{2}{9}n - \frac{296}{27}$$

□

4 Recap

I will now, for ease of recollection, attempt to summarize the portions of this paper that are likely to be applicable in your studies or research.

We saw in Lemma 3.1 that on any graph the demon has a pure optimal strategy. This means that when you observe the demon's choice of whether to move x or y in on the graph, it makes no difference whether this is his first move or his hundredth move.

We saw in Lemma 3.3 that hitting time defines a pre-order of any graph. This result is useful as it does not require any concept of meeting time or collisions. For any given graph, we can always order the vertices by the relation of their hitting times. Specifically there always exists at least one vertex t such that hitting time from any other vertex u to t is always greater than the hitting time from t to u. That is

$$H(t,u) \le H(u,t).$$

In Theorem 3.5 we saw that for a hidden vertex t

$$M(x,y) \leq H(x,y) + H(y,t) - H(t,y).$$

This can be used on specific instances of a graph where you are able to compute hitting times and identify a hidden vertex to calculate an upper bound on meeting time. Alternatively, and more generally, if you are able to only calculate the maximum hitting time on a graph, the meeting time is no more than twice this quantity.

Finally, we saw in Theorem 3.7 that for any graph on 13 or more vertices

$$M(x,y) \leq \frac{4}{27}n^3 + \frac{4}{3}n^2 + \frac{2}{9}n - \frac{296}{27}.$$

This is useful because it gives you an upper bound on meeting time without knowing anything at all about the graph's structure.

All of the results discussed here are for the case when moves are made by an intelligent adversarial demon, so they also apply to situations where the token which moves at each step is decided randomly.

As discussed, these results apply to self-stabilization of token management schemes in distributed computing. Our polynomial bound on meeting time implies that distributed computing systems can always self-stabilize in polynomial time.

References

[1] D. Coppersmith, P. Tetali, and P. Winkler, Collisions among Random Walks on a Graph, *SIAM J. on Discrete Mathematics* **6** No. 3 (1993), 363–374.

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