6.856 — Randomized Algorithms

David Karger

Handout #25, December 5th, 2002 — Homework 12 Solutions

Problem 1

(a) Consider the vector $\pi = (1, 1, \ldots, 1)$. Then πP is just the sum of the rows of P, which is just the vector of column sums, each of which is 1 by assumption. So $\pi P = \pi$. Thus, π, normalized to unit sum, is a stationary distribution for P.

(b) If $p_{ij} = p_{ji}$, then the i-th column sum is equal to the i-th row sum, which is 1 (since P is stochastic). Thus the matrix is doubly stochastic, so by the previous part its stationary distribution is uniform.

(c) Suppose $\pi_i p_{ij} = \pi_j p_{ji}$ for each i and j. Then the k^{th} coordinate of πP,

$$\begin{aligned}(\pi P)_j &= \sum_i \pi_i p_{ij} \\ &= \sum_i \pi_j p_{ji} \\ &= \pi_j \sum_i p_{ji} \\ &= \pi_j\end{aligned}$$

since $\sum_i p_{ji}$ is just the probability of transitioning out of j, which is 1. Thus π is stationary.

Problem 2

Our algorithm is the following: color the graph arbitrarily with two colors. Then, as long as there is a mono-chromatic triangle (all three points the same color), flip the color of a random vertex in that triangle.

We will now show that this actually converges to a stable configuration in polynomial time. For this purpose, fix any legal three-coloring $\mathcal{C}$ of the graph (say using colors red, green and blue). Suppose that we use the colors red and green in our two-coloring. Let k be the number of nodes which have the same colors in $\mathcal{C}$ and our current coloring. Suppose that we do not yet have a legal two-coloring. How does k change in this case?

If we have a mono-chromatic triangle, say all nodes red, then flipping a random node to green can have the following effect (recall that in $\mathcal{C}$ all three nodes have different colors!).

With probability 1/3 we flip the node which is red in $\mathcal{C}$, and k decreases by one. With probability 1/3 the green node in $\mathcal{C}$ gets flipped, and k increases by one. And, finally, with probability 1/3, we flip the blue node, and k does not change.

This corresponds to a Markov process that traverses a line graph of length n (k is always between 0 and n), going left and right (and staying still) with equal probability. In class we showed that the cover time for such a line graph is $O(n^2)$, and hence our algorithm terminates in polynomial time.

Problem 3

Suppose our graph is $G = (V, E)$. Now consider the graph $G' = (V \times V, E')$, where

$$E' := \{ \ ((a, b), (c, d)) \mid (a, c), (b, d) \in E \ \}.$$

First we show that if G is connected and non-bipartite, then so is G'. If G' is non-bipartite, then it has an odd cycle $v_1, v_2, \ldots, v_k$. But then so has G' with $(v_1, v_1), (v_2, v_2), \ldots, (v_k, v_k)$, and is therefore also non-bipartite. As for connectivity, consider two vertices (a, b) and (c, d) in G'. There are paths in G from a to c, and from b to d. Moreover, these paths can be made to have the same length (although they are not necessarily simple then). This can done as follows:

- if the two path lengths have different parity (i.e. one has even length, the other has odd length), then do the following: consider a path from a to c that goes through v_1 (since G is connected, this – not necessarily simple – path exists). If this path's length does not have the same parity as the length of the path from b to d, then add the odd cycle $v_1, v_2, \ldots, v_k, v_1$ to the path from a to c. Since the path visits v_1 this cycle can just be inserted. After this, both path lengths have the same parity.

- if they still have different lengths, make the shorter of the two paths longer by adding 2-cycles at its end. E.g., if the path from a to c is shorter, and x is some neighbor of c, then add a sequence $c, x, c, x, \ldots, x, c$ to the end of the path to make it have the same length as the other path.

Given two paths of the same length from a to c, and b to d, we can easily construct a path in G' from (a, b) to (c, d), by following the path from a to c in the first component, and the path from b to d in the second component. So G' is connected.

Now back to the problem of parallel Markov processes. A random walk G' has the same behavior as two independent random walks on the graph G. This is because we have $\deg'_G((u, v)) = \deg_G(u) \deg_G(v)$, so the probability to go from a node (a, b) to a node (c, d) in G' is the same as independently going from a to c, and from b to d in two random walks on G.

In class we proved a cubic cover time for any undirected graph, so in particular G' has a cover time of $O((n^2)^3) = O(n^6)$. This implies that all nodes of the form (a, a) in G' are reached in polynomial time, which means that the two random walks meet in polynomial time.

In other words, if all drivers in Boston were driving randomly (as observations would seem to suggest), any pair of them will meet each other in polynomial time.

Point distribution: 1 point for defining G', 2 points for arguing that the transition probabilities modeled two independent Markov processes, 2 points for proving G' non-bipartite, 3 points for proving G' connected, 2 points for showing a $O(n^6)$ cover time.

Problem 4

Consider the directed graph on nodes $1, \ldots, n$ which has directed edges $(i, i+1)$ and $(i, 1)$ for all i. A sequence of transitions that gets from vertex 1 to vertex n must have an uninterrupted sequence of n transitions along edges $(i, i+1)$. The probability of such a subsequence occurring in a given sequence of n tries is 2^{-n}, so in a sequence of length (say) 1.9^n, such a subsequence is exponentially unlikely to occur. Thus, with high probability it takes exponential time to cover the graph.