

CS 599 (Spring 2009) — Assignment 5

Due: 04/30/2009

Read Chapters 6.1–6.3 and 6.5 from the textbook (the rest of Chapter 6 should also be interesting). Also, read the paper by Fakcharoenphol, Rao, and Talwar (I recommend the journal version), and optionally the two papers by Bartal. Here are the homework problems (as usual, even though the textbook gives references where these problems are solved, it is *not* appropriate to seek out these references as hints for solutions):

- (1) Problem 6.5 from the textbook.
- (2) Problem 6.16 from the textbook.
- (3) As we mentioned in class, the 2-SAT algorithm from Section 6.1 can be naturally generalized to 3-SAT (or k -SAT). Since 3-SAT is NP-hard, a polynomial number of steps will not usually suffice. It turns out to be better to randomly restart the algorithm periodically if it hasn't found a solution yet.

Here's the new algorithm:

```
1: for  $g(n)$  iterations do
2:   Let  $x$  be an assignment where each  $x_i$  is independently true/false with probability  $\frac{1}{2}$ .
3:   while  $x$  is not a satisfying assignment, and the number of update steps has been less than  $f(n)$  do
4:     Let  $C$  be a uniformly randomly chosen unsatisfied clause.
5:     Let  $x_i$  be a uniformly random variable in  $C$ .
6:     Update  $x$  by flipping the value of  $x_i$ .
7:   if no solution has been found then
8:     Output "Probably Unsatisfiable".
9:   else
10:    Output the satisfying solution.
```

Here, we will focus only on 3-SAT, and prove that we can set the parameters such that if the formula is satisfiable, we will find a satisfying assignment in time $O((4/3)^n \text{poly}(n))$. We will derive this in several steps. Notice that even if you don't succeed in proving one part, you can still use it in the next part of the problem.

For the analysis, assume that the formula is satisfiable. As in class, we will consider a suitable (now biased) Markov Chain on the interval of numbers $\{0, \dots, n\}$.

- (a) Derive the probabilities for the resulting Markov Chain.
- (b) We will now be more pessimistic, and consider the natural generalization of the Markov Chain to the infinite line $\{0, 1, 2, \dots\}$.

Let $p(d)$ be the probability that starting at d , this new random walk will *ever* reach 0. Using that $p(0) = 1$ and $p(\infty) = 0$, prove that $p(d) = 2^{-d}$ for all d .

- (c) In the same version as above, prove that the probability of being at 0 after *exactly* $3d$ steps, starting from d is at least $q(d) = \frac{1}{2^d \text{poly}(d)}$.

(This result says that choosing $f(n) = 3n$ is a safe choice. The reason is that in the pessimistic version, our probability of ever getting to the origin will be at best polynomially higher than our probability of getting there within at most $3n$ steps.)

- (d) Prove that the probability that one whole iteration of the outer loop (starting from a random assignment) finds a satisfying assignment in at most $f(n)$ updates is at least $(3/4)^n \cdot \frac{1}{\text{poly}(n)}$.

- (e) Determine a suitable function $g(n)$, and prove that with high probability, the resulting algorithm will, in time $O((4/3)^n \cdot \text{poly}(n))$, find a satisfying assignment (when there is one).
- (4) We are given n points in the unit square $[0, 1]^2$. We want to partition these points into “clusters centered at the corners”, so that nearby points usually end up in the same cluster. Here is an algorithm for doing that.

First, we choose a uniformly random permutation of the four corners of the unit square, and a radius r uniformly randomly between 0 and 1. Then, we go through the four corners in the uniformly random order. Let's call them C_1, C_2, C_3, C_4 , which are $(0, 0), (0, 1), (1, 0), (1, 1)$ in some order. For each $C_i, i \leq 3$, assign to S_i all points within distance at most r of C_i . Let S_4 be the cluster of all points not assigned yet.

Prove that there are constants $0 < \alpha \leq \beta$ with the following property: For each pair of points x, y at distance $d = d(x, y)$ (the Euclidean distance; if you prefer, you can use the Manhattan distance, though), the probability that x and y are in the same cluster is between αd and βd .