Lecture 37 Analysis of Luby's Algorithm

In the previous lecture we proved that for each good vertex v, the probability that v is deleted in the current stage is at least $\frac{1}{36}$. Recall that a vertex v is *good* if

$$\sum_{u \in N(v)} \frac{1}{2d(u)} \geq \frac{1}{6} \tag{60}$$

(intuitively, if it has lots of neighbors of low degree), and that an edge is *good* if it is incident to at least one good vertex. Since the probability that a good edge is deleted is at least as great as the probability that its good endpoint is deleted (if both its endpoints are good, so much the better), a good edge is deleted with probability at least $\frac{1}{36}$.

Lemma 37.1 *At least half the edges in the graph are good.*

Proof. Direct each edge toward its endpoint of higher degree, breaking ties arbitrarily. Then each bad vertex has at least twice as many edges going out as coming in, since if not then at least a third of the vertices adjacent to v would have degree $d(v)$ or lower, and this would imply (60).

Using this fact, we can assign to each bad edge e directed into a bad vertex v a pair of edges (bad or good) directed out of v so that each bad edge is assigned a unique pair. This implies that there are at least twice as many edges in all as bad edges. Equivalently, at least half the edges are good. □

We can now argue that the expected number of edges removed at a given stage is at least a constant fraction of the number of edges present.

Theorem 37.2 *Let the random variable X represent the number of edges deleted in the current stage. Then*

$$\mathcal{E}X \geq \frac{|E|}{72} .$$

Proof. Let G denote the set of good edges. For $e \in E$, define the random variable

$$X_e = \begin{cases} 1 , & \text{if } e \text{ is deleted} \\ 0 , & \text{otherwise.} \end{cases}$$

Then $X = \sum_{e \in E} X_e$, and by linearity of expectation,

$$\begin{aligned} \mathcal{E}X &= \sum_{e \in E} \mathcal{E}X_e \\ &\geq \sum_{e \in G} \mathcal{E}X_e \\ &\geq \sum_{e \in G} \frac{1}{36} \quad \text{(by Lemma 36.3)} \\ &= \frac{|G|}{36} \\ &\geq \frac{|E|}{72} \quad \text{(by Lemma 37.1).} \end{aligned}$$

□

We have shown that we can expect to delete at least a fixed fraction of the remaining edges at each stage. This implies that the expected number of stages required until all m edges are deleted is $O(\log m)$. We leave this argument as a homework exercise (Homework 10, Exercise 1).

37.1 Making Luby's Algorithm Deterministic

As described in the last lecture, each stage of Luby's algorithm makes n independent calls on a random number generator, one for each vertex. We can think of the call for vertex u as a flip of a biased coin with $\Pr(\text{heads}) = \frac{1}{2d(u)}$ and $\Pr(\text{tails}) = 1 - \frac{1}{2d(u)}$. It can be shown that $\Omega(n)$ truly random bits (independent flips of a fair coin) are necessary to generate these n independent biased coin flips.

However, a quick check reveals that the analysis of Luby's algorithm never used the independence of the biased coin flips, but only the weaker condition

of *pairwise independence*. Recall from the last lecture that a collection of events $\mathcal{A}$ are *independent* if for all subsets $\mathcal{B} \subseteq \mathcal{A}$,

$$\Pr(\bigcap \mathcal{B}) \;=\; \prod_{A \in \mathcal{B}} \Pr(A) \;;$$

for *pairwise independence*, this only has to hold for subsets $\mathcal{B}$ of size two.

After observing that only pairwise independence was necessary for the analysis, Luby made the beautiful observation that only $O(\log n)$ truly random bits are needed to generate the n pairwise independent biased coin flips. This leads to a deterministic NC algorithm: in parallel, consider all possible bit strings of length $O(\log n)$ representing all possible outcomes of $O(\log n)$ flips of a fair coin (there are only $2^{O(\log n)} = n^{O(1)}$ of them). Use each such bit string to generate the n pairwise independent biased coin flips as if that string were obtained from a random number generator, and carry on with the algorithm. Since we expect to delete at least a constant fraction of the edges, one of the deterministic simulations must delete at least that many edges. Pick the one that discards the most edges and throw the other parallel computations out, then repeat the whole process. Everything is deterministic and at least a constant fraction of the edges are removed at each stage.

Here is how to simulate the n pairwise independent biased coin flips with $O(\log n)$ independent fair coin flips. Let p be a prime number in the range n to $2n$ (such a prime exists by *Bertrand's postulate*; see [49, p. 343]). Assume the vertices of the graph are elements of the finite field $\mathcal{Z}_p$. For each vertex u, let a_u be an integer in the range $0 \le a_u < p$ such that the fraction $\frac{a_u}{p}$ is as close as possible to the desired bias $\frac{1}{2d(u)}$. (We will not get the exact bias $\frac{1}{2d(u)}$, but only the approximation $\frac{a_u}{p}$. This will be close enough for our analysis.)

Let A_u be any subset of $\mathcal{Z}_p$ of size a_u. To simulate the biased coin flips, choose elements x and y uniformly at random from $\mathcal{Z}_p$ and calculate $x + uy$ in $\mathcal{Z}_p$ for each vertex u. Declare the flip for vertex u to be heads if $x + uy \in A_u$, tails otherwise.

Note that the random selection of x and y, since they are chosen with uniform probability from a set of size p, requires $2 \log p = O(\log n)$ truly random bits.

For each $z, y \in \mathcal{Z}_p$, there is exactly one $x \in \mathcal{Z}_p$ such that $x + uy = z$, namely $x = z - uy$. Using this fact at the critical step, we calculate the probability of heads for the vertex u:

$$\begin{aligned}
\Pr(x + uy \in A_u) &= \frac{1}{p^2} \left|\{(x, y) \mid x + uy \in A_u\}\right| \\
&= \frac{1}{p^2} \sum_{z \in A_u} \left|\{(x, y) \mid x + uy = z\}\right| \\
&= \frac{1}{p^2} \sum_{z \in A_u} p
\end{aligned}$$

$$= \frac{a_u}{p} .$$

Finally, we show pairwise independence. For any $u, v, z, w \in \mathcal{Z}_p$, $u \neq v$, there is exactly one solution x, y to the linear system

$$\begin{bmatrix} 1 & u \\ 1 & v \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} z \\ w \end{bmatrix}$$

over $\mathcal{Z}_p$, since the matrix is nonsingular. Thus

$$\begin{aligned}
&\Pr(x + uy \in A_u \wedge x + vy \in A_v) \\
&= \frac{1}{p^2} \left|\{(x, y) \mid x + uy \in A_u \wedge x + vy \in A_v\}\right| \\
&= \frac{1}{p^2} \sum_{z \in A_u} \sum_{w \in A_v} \left|\{(x, y) \mid x + uy = z \wedge x + vy = w\}\right| \\
&= \frac{1}{p^2} \sum_{z \in A_u} \sum_{w \in A_v} 1 \\
&= \frac{a_u a_v}{p^2} \\
&= \Pr(x + uy \in A_u) \cdot \Pr(x + vy \in A_v) .
\end{aligned}$$

We have seen how to generate up to p pairwise independent events with only $2 \log p$ truly random bits. A generalization of this technique allows us to generate up to p d-wise independent events with only $d \log p$ truly random bits: pick $x_0, \ldots, x_{d-1} \in \mathcal{Z}_p$ uniformly at random; the u^{th} event is

$$x_0 + x_1 u + x_2 u^2 + \cdots + x_{d-1} u^{d-1} \in A_u .$$

The analysis of this generalization is left as an exercise (Homework 10, Exercise 2).