Lecture 36 Luby's Algorithm

In this lecture and the next we develop a probabilistic *NC* algorithm of Luby for finding a maximal independent set in an undirected graph. Recall that a set of vertices of a graph is *independent* if the induced subgraph on those vertices has no edges. A *maximal* independent set is one contained in no larger independent set. A maximal independent set need not be of maximum cardinality among all independent sets in the graph.

There is a simple deterministic polynomial-time algorithm for finding a maximal independent set in a graph: just start with an arbitrary vertex and keep adding vertices until all remaining vertices are connected to at least one vertex already taken. Luby [76] and independently Alon, Babai, and Itai [6] showed that the problem is in random *NC* (*RNC*), which means that there is a parallel algorithm using polynomially many processors that can make calls on a random number generator such that the *expected* running time is polylogarithmic in the size of the input.

The problem is also in (deterministic) *NC*. This was first shown by Karp and Wigderson [59]. Luby [76] also gives a deterministic *NC* algorithm, but his approach has a decidedly different flavor: he gives a probabilistic algorithm first, then develops a general technique for converting probabilistic algorithms to deterministic ones under certain conditions. We will see how to do this in the next lecture.

Luby's algorithm is a good vehicle for discussing probabilistic algorithms, since it illustrates several of the most common concepts used in the analysis of such algorithms:

Law of Sum. The *law of sum* says that if $\mathcal{A}$ is a collection of pairwise disjoint events, *i.e.* if $A \cap B = \emptyset$ for all $A, B \in \mathcal{A}$, $A \neq B$, then the probability that at least one of the events in $\mathcal{A}$ occurs is the sum of the probabilities:

$$\Pr(\bigcup \mathcal{A}) \quad = \quad \sum_{A \in \mathcal{A}} Pr(A) \; .$$

Expectation. The *expected value* $\mathcal{E}X$ of a discrete random variable X is the weighted sum of its possible values, each weighted by the probability that X takes on that value:

$$\mathcal{E}X \quad = \quad \sum_n n \cdot \Pr(X = n) \; .$$

For example, consider the toss of a coin. Let

$$X \quad = \quad \begin{cases} 1 \; , & \text{if the coin turns up heads} \\ 0 \; , & \text{otherwise.} \end{cases} \tag{57}$$

Then $\mathcal{E}X = \frac{1}{2}$ if the coin is unbiased. This is the expected number of heads in one flip. Any function $f(X)$ of a discrete random variable X is a random variable with expectation

$$\begin{aligned} \mathcal{E}f(X) \quad &= \quad \sum_n n \cdot \Pr(f(X) = n) \\ &= \quad \sum_m f(m) \cdot \Pr(X = m) \; . \end{aligned}$$

It follows immediately from the definition that the expectation function $\mathcal{E}$ is linear. For example, if X_i are the random variables (57) associated with n coin flips, then

$$\mathcal{E}(X_1 + X_2 + \cdots + X_n) \quad = \quad \mathcal{E}X_1 + \mathcal{E}X_2 + \cdots + \mathcal{E}X_n \; ,$$

and this gives the expected number of heads in n flips. The X_i need not be independent; in fact, they might all be the same flip.

Conditional Probability and Conditional Expectation. The *conditional probability* $\Pr(A \mid B)$ is the probability that event A occurs given that event B occurs. Formally,

$$\Pr(A \mid B) \quad = \quad \frac{\Pr(A \cap B)}{Pr(B)} \; .$$

The conditional probability is undefined if $\Pr(B) = 0$.

The *conditional expectation* $\mathcal{E}(X \mid B)$ is the expected value of the random variable X given that event B occurs. Formally,

$$\mathcal{E}(X \mid B) \quad = \quad \sum_n n \cdot Pr(X = n \mid B) \; .$$

If the event B is that another random variable Y takes on a particular value m, then we get a real-valued function $\mathcal{E}(X \mid Y = m)$ of m. Composing this function with the random variable Y itself, we get a new random variable, denoted $\mathcal{E}(X \mid Y)$, which is a function of the random variable Y. The random variable $\mathcal{E}(X \mid Y)$ takes on value n with probability

$$\sum_{\mathcal{E}(X|Y=m)=n} \Pr(Y = m) \ ,$$

where the sum is over all m such that $\mathcal{E}(X \mid Y = m) = n$. The expected value of $\mathcal{E}(X \mid Y)$ is just $\mathcal{E}X$:

$$\begin{aligned}
\mathcal{E}(\mathcal{E}(X \mid Y)) &= \sum_m \mathcal{E}(X \mid Y = m) \cdot \Pr(Y = m) \\
&= \sum_m \sum_n n \cdot \Pr(X = n \mid Y = m) \cdot \Pr(Y = m) \\
&= \sum_n n \cdot \sum_m \Pr(X = n \wedge Y = m) \qquad (58) \\
&= \sum_n n \cdot \Pr(X = n) \\
&= \mathcal{E}X
\end{aligned}$$

(see [33, p. 223]).

Independence and Pairwise Independence. A set of events $\mathcal{A}$ are *independent* if for any subset $\mathcal{B} \subseteq \mathcal{A}$,

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

They are *pairwise independent* if for every $A, B \in \mathcal{A}$, $A \neq B$,

$$\Pr(A \cap B) \ = \ \Pr(A) \cdot \Pr(B) \ .$$

For example, the probability that two successive flips of a fair coin both come up heads is $\frac{1}{4}$. Pairwise independent events need not be independent: consider the three events

- the first flip gives heads
- the second flip gives heads
- of the two flips, one is heads and one is tails.

The probability of each pair is $\frac{1}{4}$, but the three cannot happen simultaneously.

If A and B are independent, then $\Pr(A \mid B) = \Pr(A)$.

Inclusion-Exclusion Principle. It follows from the law of sum that for any events A and B, disjoint or not,

$$\Pr(A \cup B) \;=\; \Pr(A) + \Pr(B) - \Pr(A \cap B)\ .$$

More generally, for any collection $\mathcal{A}$ of events,

$$\begin{aligned} &\Pr(\textstyle\bigcup \mathcal{A}) \\ &= \sum_{A \in \mathcal{A}} Pr(A) - \sum_{\substack{\mathcal{B} \subseteq \mathcal{A} \\ |\mathcal{B}| = 2}} \Pr(\textstyle\bigcap \mathcal{B}) + \sum_{\substack{\mathcal{B} \subseteq \mathcal{A} \\ |\mathcal{B}| = 3}} \Pr(\textstyle\bigcap \mathcal{B}) - \cdots \pm \Pr(\textstyle\bigcap \mathcal{A})\ . \end{aligned}$$

This equation is often used to estimate the probability of a join of several events. The first term alone gives an upper bound and the first two terms give a lower bound:

$$\begin{aligned} \Pr(\textstyle\bigcup \mathcal{A}) &\le \sum_{A \in \mathcal{A}} Pr(A) \\ \Pr(\textstyle\bigcup \mathcal{A}) &\ge \sum_{A \in \mathcal{A}} Pr(A) - \sum_{\substack{A, B \in \mathcal{A} \\ A \ne B}} \Pr(A \cap B)\ . \end{aligned}$$

36.1 Luby's Maximal Independent Set Algorithm

Luby's algorithm is executed in stages. Each stage finds an independent set I in parallel, using calls on a random number generator. The set I, the set $N(I)$ of neighbors of I, and all edges incident to $I \cup N(I)$ are deleted from the graph. The process is repeated until the graph is empty. The final maximal independent set is the union of all the independent sets I found in each stage. We will show that the expected number of edges deleted in each stage is at least a constant fraction of the edges remaining; this will imply that the expected number of stages is $O(\log n)$ (Homework 10, Exercise 1).

If v is a vertex and A a set of vertices, define

$$\begin{aligned} N(v) &= \{u \mid (u,v) \in E\} &&= \{\textit{neighbors} \text{ of } v\} \\ N(A) &= \bigcup_{u \in A} N(u) &&= \{\textit{neighbors} \text{ of } A\} \\ d(v) &= \text{the } \textit{degree} \text{ of } v &&= |N(v)|\ . \end{aligned}$$

Here is the algorithm to find I in each stage.

Algorithm 36.1

1. Create a set S of candidates for I as follows. For each vertex v in parallel, include $v \in S$ with probability $\frac{1}{2d(v)}$.
2. For each edge in E, if both its endpoints are in S, discard the one of lower degree; ties are resolved arbitrarily (say by vertex number). The resulting set is I.

Note that in step 1 we favor vertices with low degree and in step 2 we favor vertices of high degree.

Define a vertex to be *good* if

$$\sum_{u \in N(v)} \frac{1}{2d(u)} \;\geq\; \frac{1}{6}\,.$$

Intuitively, a vertex is good if it has lots of neighbors of low degree. This will give it a decent chance of making it into $N(I)$. Define an edge to be *good* if at least one of its endpoints is good. A vertex or edge is *bad* if it is not good. We will show that at least half of the edges are good, and each stands a decent chance of being deleted, so we will expect to delete a reasonable fraction of the good edges in each stage.

Lemma 36.2 *For all v,* $\Pr(v \in I) \geq \frac{1}{4d(v)}$.

Proof. Let $L(v) = \{u \in N(v) \mid d(u) \geq d(v)\}$. If $v \in S$, then v does not make it into I only if some element of $L(v)$ is also in S. Then

$$\begin{aligned}
\Pr(v \notin I \mid v \in S) &\leq \Pr(\exists u \in L(v) \cap S \mid v \in S) \\
&\leq \sum_{u \in L(v)} \Pr(u \in S \mid v \in S) \\
&= \sum_{u \in L(v)} \Pr(u \in S) \quad \text{(by pairwise independence)} \\
&\leq \sum_{u \in L(v)} \frac{1}{2d(u)} \\
&\leq \sum_{u \in L(v)} \frac{1}{2d(v)} \quad (\text{since } d(u) \geq d(v)) \\
&\leq \frac{d(v)}{2d(v)} \;=\; \frac{1}{2}\,.
\end{aligned}$$

Now

$$\begin{aligned}
\Pr(v \in I) &= \Pr(v \in I \mid v \in S) \cdot \Pr(v \in S) \\
&\geq \frac{1}{2} \cdot \frac{1}{2d(v)} \;=\; \frac{1}{4d(v)}\,.
\end{aligned}$$

□

Lemma 36.3 *If v is good, then* $\Pr(v \in N(I)) \geq \frac{1}{36}$.

Proof. If v has a neighbor u of degree 2 or less, then

$$\begin{aligned}
\Pr(v \in N(I)) &\geq \Pr(u \in I) \\
&\geq \frac{1}{4d(u)} \quad \text{(by Lemma 36.2)} \\
&\geq \frac{1}{8}\,.
\end{aligned}$$

Otherwise $d(u) \geq 3$ for all $u \in N(v)$. Then for all $u \in N(v)$, $\frac{1}{2d(u)} \leq \frac{1}{6}$, and since v is good,

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

There must exist a subset $M(v) \subseteq N(v)$ such that

$$\frac{1}{6} \leq \sum_{u \in M(v)} \frac{1}{2d(u)} \leq \frac{1}{3} . \tag{59}$$

Then

$$\begin{aligned}
\Pr(v \in N(I)) &\geq \Pr(\exists u \in M(v) \cap I) \\
&\geq \sum_{u \in M(v)} \Pr(u \in I) - \sum_{\substack{u, w \in M(v) \\ u \neq w}} \Pr(u \in I \wedge w \in I) \\
&\quad \text{(by inclusion-exclusion)} \\
&\geq \sum_{u \in M(v)} \frac{1}{4d(u)} - \sum_{\substack{u, w \in M(v) \\ u \neq w}} \Pr(u \in S \wedge w \in S) \\
&\geq \sum_{u \in M(v)} \frac{1}{4d(u)} - \sum_{\substack{u, w \in M(v) \\ u \neq w}} \Pr(u \in S) \cdot \Pr(w \in S) \\
&\quad \text{(by pairwise independence)} \\
&= \sum_{u \in M(v)} \frac{1}{4d(u)} - \sum_{u \in M(v)} \sum_{w \in M(v)} \frac{1}{2d(u)} \cdot \frac{1}{2d(w)} \\
&= (\sum_{u \in M(v)} \frac{1}{2d(u)}) \cdot (\frac{1}{2} - \sum_{w \in M(v)} \frac{1}{2d(w)}) \\
&\geq \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36} \text{ by (59).}
\end{aligned}$$

□

We will continue the analysis of Luby's algorithm in the next lecture.

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 $\mathcal{B} \subseteq \mathcal{A}$,

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

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 $0 \leq a_u < p$ 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:

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

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

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

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

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).