day 23 4/8/98

1 Luby Algorithm for Maximal Independent Set

1.1 definitions

1. $G = (V, E)$ an undirected graph
2. $I \subseteq V$ is an independent set if no pairs of elements in I share an edge
3. I is a maximal independent set if $\forall w \in V - I$, $\{w\} \cup I$ is not independent
4. I is a maximum independent set if $\forall I'$, $|I| \geq |I'|$. This problem is NP-hard or complete depending on how you phrase the problem.

1.2 algorithms

1.2.1 greedy lexicographic MIS

A maximal independent set algorithm is simple: grab the first vertex and run through the other vertices adding them to I if possible. This algorithm is $O(m + n)$.

This algorithm is P-complete, which means if this algorithm can be parallelized, every other parallel algorithm can be parallelized.

1.2.2 randomized parallel MIS

$MIS(G)$

1. Each vertex calculates it's own degree $d(v)$ and tries to join S with probability $\frac{1}{2d(v)}$
2. if $v, w \in S$ and share an edge, discard the one with a smaller degree. Break ties randomly, producing set I.
3. let $V = V - I - N(I)$ where $N(I) =$ the neighbor vertices in I and $E = E$ minus all edges into I.
4. return $I \cup MIS(G)$

1.2.3 Yet another MIS

$MIS2(G)$

1. Each vertex picks a random value, $v \in [0, 1]$
2. v joins if it's value is less than the $v's$ of it's neighbors, $N(v)$
3. let $V = V - I - N(I)$ where $N(I) =$ the neighbor vertices in I and $E = E$ minus all edges into I.

4. return $I \cup MIS2(G)$

The probability of joining is $\frac{1}{d+1}$ which is a bit more aggresive than the previous algorithm.

1.2.4 Analysis

On a CRCW (concurrent read/concurrent write) architecture, steps 1 and 2 take constant time.

Unfortunately, it isn't true that a constant proportion of the vertices are lost each round. Instead, we can prove:

The expected number of edges decreases by a constant fraction each round.

proof:

- v is 'good' if $\sum_{u \in N(v)} \frac{1}{2d(u)} \geq \frac{1}{6}$

For the graph with n nodes completely connected to $\sqrt{n}$ nodes, $K_{n,\sqrt{n}}$, each of the n nodes is 'bad' and the $\sqrt{n}$ nodes are 'good'.

- $e = (v, w)$ is 'good' if v or w is 'good'.

lemma: $P(v \in I) \geq \frac{1}{4d(v)}$

- $L(v) = \{u \in N(v) | d(u) \geq d(v)\}$

$$P(v \notin I | v \in S)$$

$$\leq p(\exists u \in L(v) \bigcap S | v \in S)$$

$$\leq \sum_{u \in L(v)} P(u \in S | v \in S)$$

$$= \sum_{u \in L(v)} P(u \in S)$$

$$= \sum_{u \in L(v)} \frac{1}{2 * d(u)}$$

$$\leq \sum_{u \in L(v)} \frac{1}{2 * d(v)}$$

$$\leq \frac{1}{2}$$

The lemma follows pretty imediately.

Lemma: v is good $\Rightarrow P(v \in N(I)) \geq \frac{1}{36}$

Lemma: at least $\frac{1}{2}$ the edges are good In G direct all edges from low to high degree, then

if v is 'bad' then $\leq \frac{1}{3}$ of v's edges are directed towards v. proof by contradicition:

$$\sum_{u \in N(v)} \frac{1}{2d(u)} \geq \sum_{u \in In(v)} \frac{1}{2d(u)} \geq \frac{\frac{1}{3}d(v)}{2d(v)} \geq \frac{1}{6}$$

pulling it together: Now, we can construct a binomial tree in the graph where every leaf is 'good'. This implies the number of good edges satisfies: $\#bad + 1 \leq \#good$.