1 Overview

In this lecture, we:

1. Prove that the Maximal Independent Set (MIS) algorithm from last lectures terminates in $O(\log n)$ rounds with high probability.

2. Apply the MIS algorithm to $d$-clustering and coloring.

2 Luby’s MIS Algorithm

Recall Luby’s algorithm from last lecture:

\begin{algorithm}
\caption{Luby’s MIS Algorithm}
\begin{algorithmic}
\REPEAT
\STATE Find Independent Set $S$
\STATE Each node $v$ picks priority $p_v$ from $[n^{c+2}]$ uniformly at random
\STATE $v \in S$ if $\forall w \in \Gamma(v). p_v > p_w$
\STATE $M = M \cup S$
\STATE $V = V \setminus \Gamma^+(S)$
\UNTIL{$V \neq \emptyset$}
\end{algorithmic}
\end{algorithm}

Lemma 1 $E[\text{# of edges removed per iteration}] \geq \frac{m}{2}$

Proof We say that $v$ kills $w$ if and only if $(v, w) \in E$ and $\forall u \in \Gamma(v) \cup \Gamma(w) \setminus \{v\}, p_v > p_u$. When a node $w$ is killed, $d(w)$ edges get removed. An edge can be removed if one (or both) of its ends get removed. Therefore,

$$E[\text{# edges removed}] \geq \frac{1}{2} \sum_{w \in V} P(w \text{ gets killed})d(w)$$

We showed last lecture that at most one node can kill a particular node. So we have

$$E[\text{# edges removed}] \geq \frac{1}{2} \sum_{w \in V} \sum_{u \in \Gamma(w)} P(w \text{ is killed by } u) d(w)$$
We also proved in the last lecture that \( P(w \text{ killed by } u) \geq \frac{1}{d(w) + d(u)} \). Therefore,

\[
E[\# \text{ edges removed}] \geq \frac{1}{2} \sum_{w \in V} \sum_{u \in \Gamma(w)} \frac{d(w)}{d(w) + d(u)} \\
\geq \frac{1}{2} \sum_{(u, w) \in E} \frac{d(w)}{d(w) + d(u)} + \frac{d(u)}{d(w) + d(u)} \\
\geq \frac{1}{2} |E|
\]

Lemma 2 shows that the algorithm will terminate in \( O(\log n) \) rounds in expectation. We still need to show concentration.

**Lemma 2** Luby’s algorithm finishes in \( O(\log n) \) rounds with high probability.

**Proof** Suppose \( P[\# \text{ edges removed} > \frac{n}{4}] < 0.1 \). Then we have:

\[
E[\# \text{ edges removed}] < 0.1m + \frac{0.9}{4}m < \frac{m}{2}
\]

This contradicts Lemma 2, therefore \( P[\# \text{ edges removed} > \frac{m}{4}] \geq 0.1 \).

Let’s call a round with \( \geq \frac{n}{4} \) removed edges a good round. Over \( 10c \log n \) rounds, \( E[\# \text{ good rounds}] \geq c \log n \). Because we choose different priorities uniformly at random for each round, the rounds are independent, and we can apply Chernoff Bounds:

\[
P[\# \text{ good rounds} \leq (1 - \delta)c \log n] \leq e^{-\delta^2c \log n} \leq e^{-c' \log n} \leq \frac{1}{n^{c'}}
\]

**3 Application**

**3.1 d-clustering**

d-clustering clusters nodes into centered areas such that:

1. each node is in a area
2. each node is not too far from its center (\( \leq d \))
3. no two centers are too close from each other (≥ d)

We compute a d-clustering by running MIS on Gd (each node is connected to nodes that are ≤ d away). The MIS nodes are the centers of the d-clustering and we can assign other nodes to the center that is closest.

Is this a valid d-clustering? Obviously, every node is in a cluster by construction. No two centers are ≤ d from each other as they would have been neighbors in Gd otherwise. Moreover, if a node was ≥ d away from a center, then it would not be a neighbor to any other centers in Gd, and could be added to the MIS. That is impossible, since the set is maximal. It follows that the d-clustering is valid.

How to do it distributed? We cannot add edges to get Gd. Instead, for d rounds, each node broadcasts highest priority it received. This takes d log n rounds using Luby’s algorithm.

3.2 Coloring

Claim 3 c-coloring in T rounds ⇒ MIS in c + T rounds

Proof Use coloring as the independent sets. Each iteration will select one color. It takes T rounds to get the coloring, and c iterations of Luby to get a MIS. ■

Claim 4 MIS in T rounds ⇒ (∆ + 1)-coloring in (∆ + 1)T rounds, where ∆ is the max degree.

Proof Take ∆ + 1 copies of the original graph, and connect nodes which correspond to the same node in the original graph. Each node in the original graph is a ∆ + 1 clique in the new graph. Therefore, at most one of them will be in a MIS. Consequently, neighboring cliques have at most one node selected, and there are at most ∆ of them. It follows that MIS contains exactly one node per clique. We assign colors based on which copy the selected node belongs to.

To distribute it, each node computes its clique. Each node needs to send ∆ + 1 priorities per rounds (one per node in the clique). For the LOCAL model (no limit on the message size), the algorithm would take only T rounds. ■