6.5240 Sublinear Time Algorithms November 9, 2022

Lecture 17

Lecturer: Ronitt Rubinfeld *Scribe: Shriya Rangaswamy*

1 Introduction

1.1 Maximal Independent Set (MIS)

Let $G = (V, E)$ be a graph of maximum degree d, with vertex set V and edge set E. We say $U \subseteq V$ is a Maximal Independent Set (MIS) if the following are true:

1. $\forall\ u_1, u_2 \in U$, $(u_1, u_2) \notin E$
2. There exists no such $w \in V \setminus U$ such that $U \cup \{w\}$ is independent

This is not to be confused with Maximum Independent Set, which is in fact NP-Hard. MIS can actually be solved greedily in polynomial time. This MIS however is not necessarily unique, as a graph can have multiple Maximum Independent Sets. This lecture will focus on how to make use of Local Computation Algorithms to output an MIS in sublinear time.

1.2 Local Computation Algorithms (LCA)

Here, we look at a new form of computation which computes only specific parts of the output at any given time. Several problems, including locally decodable codes, local property reconstruction, and local decompression, employ LCAs. For the problem of Maximal Independent Set (MIS), the model supports queries from the user that are of the form "is vertex i in the MIS?" and the model will probe the input graph and output $y_i = 1$ if vertex i is in the MIS and $y_i = 0$ otherwise.

Additionally. several LCAs can act on the graph in parallel. The difficulty is that the answer that each LCA outputs all of its queries must be consistent with the output of the other LCAs. That is, they must all work in tandem to construct parts of the same MIS. The way in which this is done is the following:

- Initially the LCAs share a short, random string
- Each LCA computes independently its output based on the queries it receives
- The union of the output of all the LCAs provides the illusion of a fully constructed MIS

1.3 LCAs to Distributed Algorithms

If there is a k-round distributed algorithms for MIS, the Parnas-Ron reduction tells us that:

- A vertex v's output only depends on on the inputs and computation of the k-radius ball around v
- Can convert a k-round distributed algorithm into a d^k time sequential algorithm

2 Distributed Algorithm for MIS

The following is actually a variant of Luby's algorithm, but the same probabilistic analysis applies:

Algorithm 1:

1. MIS $= \emptyset$
2. Set the status of all nodes to "live"

3. Repeat until graph is "dead":
   - $\forall\ v \in V$, color self red with probability $\frac{1}{2d}$, else color self blue. Send color to all neighbors
   - If v colors self red and no other neighbor of v colors themselves red, then:
     (a) Add v to MIS
     (b) Remove v and its neighbors from the graph (set their status to "dead")

Note: setting a node's status to "dead" means that the node is no longer in contention to be added to the MIS (its status has already been decided). However, for the analysis of the sublinear version of this algorithm, we will assume that dead nodes can still randomly color themselves in each round, but cannot be added to the MIS.

Theorem 1: Let X be the number of rounds that Luby runs until graph is empty. Then, $P(X \geq 8d \log n) \leq \frac{1}{n}$
Corollary 1: $E[X] = d \log n$

This seems to be a problem for us though. If a distributed algorithm takes $\log n$ rounds to terminate, then simulating the distributed algorithm sequentially will not result in a sublinear time algorithm.

Theorem 2: $P(v$ live and added to MIS in this round$) \geq \frac{1}{4d}$

Proof of Theorem 2:

1. $P(v$ colors self red$) = \frac{1}{2d}$
2. By Union Bound $P(w \in N(v)$ colors themselves red$) \leq \sum_{w \in N(v)} \frac{1}{2d} \leq \frac{1}{2}$
3. Therefore, $P(v$ is live and added to MIS this round$) \geq \frac{1}{2d} * \frac{1}{2} = \frac{1}{4d}$

Corollary to Theorem 2: $P(v$ live after $4kd$ rounds$) \leq (1 - \frac{1}{4d})^{4kd} = e^{-k}$. Setting $k = O(\log n)$, we get that $P(v$ is still live after k rounds$) \leq \frac{1}{n^c}$

New Idea: Run Luby's algorithm for only a constant number of rounds. Hopefully with high probability, most nodes will be "dead" and the remaining "live" nodes will lie in small connected components.

3 Sublinear Local Computation Algorithm for MIS

The LCA for this problem will be motivated by considering the behavior of Luby's Algorithm after $k = O(d \log d)$ rounds.

There are 3 options for the status of every node v after $O(d \log d)$ rounds:

- v is still "live"
- v is in the MIS
- v is not in the MIS

Using the Parnas-Ron reduction, we can simulate v's view in a sequential manner in $d^k = d^{O(d \log d)}$ time. If v has either been added to the MIS or one of its neighbors has been added to the MIS, we are done and know the status of v.

What if v is still "live" though? $P(v$ is still "live" after $O(d \log d)$ rounds$) \leq \frac{1}{d^c}$. Then we do the following:

1. Start a BFS from v to find its connected component (CC) of "live" nodes by running the sequential version of Luby's algorithm on each node in CC: This takes $d^{O(d \log d)} \cdot$ (size of connected component) time
2. Compute lexicographically the MIS M' for this CC: Can do this in $d^{O(d \log d)}$ time
3. Output whether v is in/out of M'