Brief Announcement: Using Read-k Inequalities to Analyze a Distributed MIS Algorithm

Sriram V. Pemmaraju
Department of Computer Science
The University of Iowa
Iowa City, IA 52242-1419, USA
sriram-pemmaraju@uiowa.edu

Talal Riaz
Department of Computer Science
The University of Iowa
Iowa City, IA 52242-1419, USA
talal-riaz@uiowa.edu

ABSTRACT

Until recently, the fastest distributed MIS algorithm, even for simple graphs, e.g., unoriented trees, has been the simple randomized algorithm discovered in the 80s. This algorithm (commonly called Luby’s algorithm) computes an MIS in $O(\log n)$ rounds (with high probability). This situation changed when Lenzen and Wattenhofer (PODC 2011) presented a randomized $O(\sqrt{\log n \cdot \log \log n})$-round MIS algorithm for unoriented trees. This algorithm was improved by Barenboim et al. (FOCS 2012), resulting in an MIS algorithm running in $O(\sqrt{\log n \cdot \log \log n})$ rounds.

The analyses of these tree MIS algorithms depend on “near independence” of probabilistic events, a feature of the tree structure of the network. In their paper, Lenzen and Wattenhofer hope that their algorithm and analysis could be extended to graphs with bounded arboricity. We show how to do this in this note. By using a new tail inequality for read-k families of random variables due to Gavin-sky et al. (Random Struct Algorithms, 2015), we show how to deal with dependencies induced by the recent tree MIS algorithms when they are executed on bounded arboricity graphs. Specifically, we analyze a version of the tree MIS algorithm of Barenboim et al. and show that it runs in $O(poly(\alpha) \cdot \sqrt{\log n \cdot \log \log n})$ rounds in the CONGEST model for arboricity-$\alpha$ graphs.

Keywords

Bounded Arboricity Graphs, CONGEST model, Luby’s Algorithm, Maximal Independent Set, Read-k Inequality

1. INTRODUCTION

Computing a maximal independent set (MIS) is a fundamental problem in distributed computing because it nicely captures the essential challenge of distributed symmetry breaking. The fastest algorithm for MIS, commonly called Luby’s algorithm [1, 7, 5], is randomized and it computes an MIS of an $n$-node graph in $O(\log n)$ rounds, with high probability (whp), i.e., with probability at least $1 - 1/n$. In PODC 2011, Lenzen and Wattenhofer [6] showed that an MIS in an $n$-node unoriented tree can be computed in $O(\sqrt{\log n \cdot \log \log n})$ rounds whp. More recently in FOCS 2012, Barenboim et al. [2, 3] presented a slightly faster tree MIS algorithm (similar to the Lenzen-Wattenhofer algorithm) that runs in $O(\sqrt{\log n \cdot \log \log n})$ rounds whp.

A natural question that arises (and asked by Lenzen and Wattenhofer [6]) is whether these tree MIS algorithms and analyses can be extended to bounded arboricity graphs. A graph $G$ is said to have arboricity $\alpha$ if $\alpha$ is the minimum number of forests that the edges of $G$ can be partitioned into. From this it follows that the edges of an arboricity-$\alpha$ graph can be oriented in such a manner that each node has at most $\alpha$ outgoing edges. Clearly, forests have arboricity 1, and the family of graphs with constant arboricity includes planar graphs, graphs with constant treewidth, graphs with constant genus, family of graphs that exclude a fixed minor, etc. Unfortunately, the Lenzen-Wattenhofer analysis and the Barenboim et al. analysis runs into trouble for graphs with even constant arboricity because of the nature of dependencies between probabilistic events in the algorithm.

The source of the difficulty can be explained as follows. Even though these algorithms [6, 2, 3] run on unoriented trees, for the purposes of analysis it can be assumed that the input tree is rooted at an arbitrary node. Because the graph is a tree, probabilistic events at children of a node $v$ are essentially independent, the only slight dependency being caused by the interaction via their parent, namely $v$. For graphs with arboricity greater than 1 the dependency structure among the probabilistic events can be much more complicated. Suppose (for the purposes of the analysis) that we orient the edges of an arboricity-$\alpha$ graph such that each node has at most $\alpha$ out-neighbors. Let us call the out-neighbors of a node $v$ its parents (denoted Parent($v$)) and the in-neighbors, its children (denoted Child($v$)). For a node $v$, consider the set Child($v$) and the dependencies among probabilistic events at nodes in Child($v$). The events we are referring to are of the type “$w$ joins the MIS” or “a neighbor of $w$ joins the MIS” for $w \in$ Child($v$). Even though each node has at most $\alpha$ parents, a node $w \in$ Child($v$) may share children with every other node in Child($v$) and as a result there could be dependencies between events at $w$ and events at any of the other nodes in Child($v$). Thus it is not clear how to take advantage of the structure of bounded arboricity graphs in order to mimic the analysis in [6, 2, 3].

The main purpose of this note is to show that recent
results on \textit{read-k families of random variables} deal with roughly this type of dependency structure and therefore provide a new approach to analyzing randomized distributed algorithms with more complicated dependency structure. Using analysis based on \textit{read-k inequalities}, we show that the tree MIS algorithms of Lenzen and Wattenhofer \cite{6} and Barenboim et al. \cite{3, 2} work for bounded arboricity graphs as well.

\subsection{1.1 Read-k Inequalities}

Let \( \{Y_j \mid 1 \leq j \leq n\} \) be a set of random variables such that each random variable \( Y_j \) is a function of some subset of the set of independent random variables \( \{X_i \mid 1 \leq i \leq m\} \). For each \( 1 \leq j \leq n \), let \( P_j \subseteq \{1, 2, \ldots, m\} \), let \( f_j \) be a boolean function of \( \{X_i \mid i \in P_j\} \), and define \( Y_j := f_j((X_i)_{i \in P_j}) \). The collection of random variables \( Y_j \) is called a \textit{read-k family} if every \( Y_j \) is present in at most \( k \) of the \( P_j \)'s. In other words, \( \mathcal{X} \) is allowed to influence at most \( k \) of the \( Y_j \)'s. Note that the \( Y_j \)'s have a complicated dependency structure amongst themselves – it is their dependency graph on the \( X_i \)'s that is bounded. For example, the dependency graph of the \( Y_j \)'s can even be a clique! The first of the two inequalities from Gavinsky et al. \cite{4} is the following.

\textbf{Theorem 1.1 (Theorem 1.2, \cite{4}).} Let \( Y_1, Y_2, \ldots, Y_n \) be a family of \textit{read-k indicator variables} with \( \Pr[Y_1 = 1] = p \). Then, \( \Pr[Y_1 = Y_2 = \cdots = Y_n = 1] \leq p^{n/k} \).

If the \( Y_j \)'s are independent, then the probability that \( Y_1 = Y_2 = \cdots = Y_n = 1 \) would simply be \( p^n \). Thus Theorem 1.1 is essentially saying that the \textit{read-k family} structure of the dependencies among the \( Y_j \)'s allows us to obtain an upper bound on the probability that is an exponential factor \( 1/k \) worse than what is possible had the \( Y_j \)'s been independent.

Gavinsky et al. \cite{4} use Theorem 1.1 and information-theoretic arguments to derive the following tail inequality on the sum of indicator random variables that form a \textit{read-k} family.

\textbf{Theorem 1.2 (Theorem 1.1, \cite{4}).} Let \( Y_1, \ldots, Y_n \) be a family of \textit{read-k indicator variables} with \( \Pr[Y_1 = 1] = p \). Define \( p := \frac{1}{k} \sum_{i=1}^n p_i \) and \( Y := \sum_{i=1}^n Y_i \). Then for any \( \epsilon, \delta > 0 \),

\[ \Pr(Y \leq (p - \epsilon) n) \leq \exp \left(-2\frac{\epsilon^2 n^2}{k}\right). \]

As in Theorem 1.1, these tail inequalities are also an exponential \( 1/k \) factor worse than corresponding Chernoff bounds that we might have used, had the \( Y_j \)'s been independent.

To see that the above tools are well-suited for analyzing tree MIS algorithms \cite{6, 3, 2}, consider a graph \( G \) with arboricity \( \alpha \), fix a node \( u \) in \( G \), and consider the set \( \text{Child}(u) \). For a node \( w \in \text{Child}(v) \), let \( Y_w \) be an indicator variable for a probabilistic event at node \( w \). Now suppose that \( Y_w \) depends on independent random choices made by \( w \) and its children. The structure of an arboricity-\( \alpha \) graph and the associated edge-orientation ensures that each node has at most \( \alpha \) parents and therefore the random choice at each node can influence at most \( \alpha \) of the \( Y_w \)'s. Thus the set \( \{Y_w \mid w \in \text{Child}(v)\} \) forms a \textit{read-\( \alpha \)} family and we can apply Theorems 1.1 and 1.2 to bound \( \Pr(\cap_w Y_w = 1) \) and to show that \( \sum Y_w \) is concentrated about its expectation.

The above example illustrates the simplest application of \textit{read-k inequalities} in our analysis. We use \textit{read-k inequalities} to evaluate probabilistic interactions between a node and its parents also. This may seem difficult to do given that a parent can have arbitrarily many children and thus a random choice at a parent can influence events at arbitrarily many children. However, in our algorithm nodes with extremely high degree opt out of the competition (temporarily) and this turns out to be sufficient for the use of \textit{read-k inequalities}, with appropriate \( k \), to analyze the interaction between nodes and their parents. Finally, our analysis also relies on interactions between a node and its grandchildren, leading to our use of \textit{read-\( \Theta(\alpha^2) \)} families as well.

\subsection{1.2 Our Result}

We apply \textit{read-k-inequality-based} analysis to the execution of the tree MIS algorithm of Barenboim et al. \cite{3, 2} on bounded arboricity graphs and obtain the following result.

\textbf{Theorem 1.3.} The tree MIS algorithm of Barenboim et al. \cite{3, 2} (with appropriate parameter values) can be used to compute an MIS in the CONGEST model on the family of graphs with arboricity \( \alpha \) in \( O(\text{poly}(\alpha) \cdot \sqrt{\log n \cdot \log \log n}) \) rounds, whp.

This result can also be seen as an improvement over the MIS result on bounded arboricity graphs due to Barenboim et al. \cite{3, 2}. In their paper, Barenboim et al. have a separate algorithm (distinct from their tree MIS) algorithm that computes an MIS on graphs with arboricity \( \alpha \) in \( O(\log^2 \alpha + \log^2/3 n) \) rounds. The dependency on \( n \) of the running time of our algorithm is asymptotically better, implying that for small \( \alpha \) (i.e., \( \alpha = O(\log n) \) for a small enough constant \( c \)) our algorithm is asymptotically faster. In this context, it should be noted that recently (in SODA 2016) Gafﬁeri presented a novel distributed MIS algorithm for general graphs that runs in \( O(\log \Delta) + 2^{O(\sqrt{\log \log \log \log n})} \) rounds and a corollary of this algorithm is an \( O(\log \alpha + \sqrt{\log n}) \)-round MIS algorithm on graphs with arboricity \( \alpha \).

\section{MIS on Bounded Arboricity Graphs}

We present an algorithm that we call \textsc{BoundedArbDependentSet}, which is essentially identical to the \textsc{TreeIndependentSet} algorithm of Barenboim et al. \cite{8, 3}, except for parameter values \( (\Theta, \Lambda, \rho_c) \) which now depend on \( \alpha \) as well.

\( I \) denotes the set of nodes which have joined the MIS and \( B \) stores a set of so-called “bad” nodes. As nodes join \( I \), they and their neighbors become inactive; additionally, nodes in \( B \) also become inactive. We use \( V_B \) to denote the set of nodes which are currently active, \( \Gamma_{IB}(u) \) to denote the neighborhood of node \( u \), restricted to nodes in \( V_B \), and \( \text{deg}_{IB}(u) := |\Gamma_{IB}(u)| \). The algorithm proceeds in \( \Theta := |\log \left(17\Delta / 16\log \log \Delta^2\right)| \) scales. For any scale \( k \), \( 1 \leq k \leq \Theta \), a node in \( V_B \) that has degree more than \( \Delta / 2^k + \alpha \) is called a \textit{high degree} node for that scale. In each scale, we start by performing \( O(\alpha^4(\log \alpha + \log \log \Delta)) \) iterations of Luby’s algorithm \cite{7}. The exact number of iterations (denoted by \( \Delta \)) is \( \lfloor p - 8\alpha^2(32\alpha^6 + 1) \cdot \ln(209\alpha^4 \ln^2 \Delta) \rfloor \). Here \( p \) is a large enough constant.

In a single iteration, every node \( v \in V_B \) chooses a random number \( r(v) \in [0, 1) \) called a \textit{priority}. If \( r(v) \) has more than \( \rho_c := 8 \ln \Delta \cdot \Delta / 2^{k+1} \) neighbors in any iteration, its priority is (deterministically) set to 0, otherwise, it chooses a priority uniformly at random in \( (0, 1) \) (denoted \( UAR(0, 1) \) in the pseudocode). After each iteration, nodes in \( I \) and neighbors of these nodes (i.e., \( \Gamma_{IB}(I) \)) are removed from \( V_B \). If, after
Algorithm 1: BoundedArbIndependentSet(G):

1: Initialize sets \( I, B \subseteq V(G) \); \( I \leftarrow \emptyset \); \( B \leftarrow \emptyset \)
2: for each scale \( k \leftarrow 1 \) to \( \Theta := \left( \log \left( \frac{\Delta}{100 + \log \Delta} \right) \right) \) do
   \( \rho_k \leftarrow 8 \ln \Delta \cdot \Delta/2^{k+1} \)
   \( \Lambda \leftarrow \left[ \rho \cdot 8 \cdot a(2(\frac{\alpha}{3})^2 + 1) \cdot \ln(260a^4 \ln^2 \Delta) \right] \)
2(a) Repeat \( \Lambda \) times
   - Each node \( v \in V_{IB} \) chooses a priority \( r(v) \):
     \[ r(v) \left\{ \begin{array}{ll} 0, & \text{if deg}_{IB}(v) > \rho_k \\ \text{UAR}(0,1), & \text{otherwise} \end{array} \] \( I \leftarrow I \cup \{ v \in V_{IB} | v \} \)
2(b) Each node \( v \) is labeled “bad” if \( |\{ w \in \Gamma_{IB}(v) | \text{deg}_{IB}(w) > \Delta/2^k + \alpha \} | > \Delta/2^{k+2} \)
   \( B \leftarrow B \cup \{ v \in V_{IB} | v \text{ is labeled “bad”} \} \)
   \( V_{IB} \leftarrow V_{IB} \setminus B \)
end
3: return \( (I, B, V_{IB}) \)

all \( \Lambda \) iterations in the current scale, a node \( v \in V_{IB} \) has more than \( \Delta/2^{k+2} \) high-degree neighbors then it is designated a “bad” node and added to the set \( B \). It is worth emphasizing that this algorithm has no access to an edge-orientation or a forest-decomposition of the given \( \alpha \)-arboricity graph.

Theorem 2.1. Using BoundedArbIndependentSet in \( O \left( \alpha^5(\log \alpha + \log \log \Delta) \cdot \log \Delta + \log \log n \cdot \alpha \right) \) rounds whp we can compute an MIS on a graph with arboricity \( \alpha \). This leads to an algorithm that computes an MIS on a graph with arboricity \( \alpha \) in \( O (\alpha^5 \log \alpha \log \Delta) \) rounds whp.

2.1 Finishing Up

Algorithm BoundedArbIndependentSet returns an independent set \( I \) (which need not be maximal), a set \( B \) of “bad” nodes, and a set \( V_{IB} \), that need not be empty. So after the algorithm has completed, we still need to process the sets \( B \) and \( V_{IB} \). The “finishing up” algorithm is aided by two results shown by our analysis: (i) the subgraph induced by \( B \) has small connected components and (ii) the graph induced by \( V_{IB} \) has the property that no node has too many high degree neighbors (otherwise, they would have been placed in \( B \)). For details of the “finishing up” see the full paper [8].

3. Overview of Analysis

The heart of the analysis shows that the following invariant is maintained (at the end of each scale) at every active node, with sufficiently high probability.

**Invariant:** At the end of scale \( k \), for all \( v \in V_{IB} \),
\[
|\{ w \in \Gamma_{IB}(v) | \text{deg}_{IB}(w) > \Delta/2^k + \alpha \} | \leq \Delta/2^{k+2}
\]

The Invariant bounds the number of high degree neighbors a node has after \( k \) scales of the algorithm. Let \( N \) denote the set on the left-hand side of the Invariant above. The goal is to show that, with probability at least \( 1 - 1/\Delta^{2p} \), in Scale \( k \), (i) either \( v \) becomes inactive in Step 2(a) because it or its neighbor joins the MIS or (ii) \( |N| \) falls to \( \Delta/2^{k+2} \)

below by the end of Scale \( k \). Showing this leads to the result that after Scale \( k \), each active node satisfies the invariant with probability at least \( 1 - 1/\Delta^{2p} \) and is therefore placed in \( B \) with probability at most \( 1/\Delta^{2p} \). Using the proof in Barenboim et al. [2, 3], we conclude from this result that all connected components in the graph induced by \( B \) are small.

We partition the analysis needed to show (i) and (ii) above into the analysis of three key probabilistic events. The first event concerns the interaction between nodes and their children and the second concerns the interaction between nodes and their parents. The third event is more complicated and it concerns the interaction between nodes and their children, their grandchildren and their children’s other parents. Let us fix a Scale \( k \) and an iteration within that scale. Let \( M \subseteq V_{IB} \) be an active subset of nodes just before the start of this iteration. The three probabilistic events we analyze can be informally described as follows. For Events (1) and (2), we assume that all nodes in \( M \) have degree at most \( \rho_k \) and are therefore competitive. The analysis of these three events makes critical use of read-\( k \) inequalities, for different values of \( k \) and leads to the claimed result. See [8] for details.

Event (1) Among the set of nodes \( M \), there exists a node whose priority is larger than the priority of all its children.

Event (2) Suppose that \( M \) is sufficiently large. Then a large fraction of the nodes in \( M \) have priority greater than priorities of all their parents.

Event (3) Suppose that every node in \( M \) has sufficiently high degree. Then a large fraction of the nodes in \( M \) become inactive due to their children joining the MIS.

4. References

[1] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986.

[2] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider. The locality of distributed symmetry breaking. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 321–330, 2012.

[3] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider. The locality of distributed symmetry breaking. *CoRR*, abs/1202.1983, 2015.

[4] D. Gavinsky, S. Lovett, M. Saks, and S. Srinivasan. A tail bound for read-\( k \) families of functions. *Random Structures & Algorithms*, 47:99–108, 2015.

[5] A. Israeli and A. Itai. A fast and simple randomized parallel algorithm for maximal matching. *Inf. Process. Lett.*, 22(2):77–80, 1986.

[6] C. Lenzen and R. Wattenhofer. MIS on trees. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing, PODC ’11*, pages 41–48, New York, NY, USA, 2011. ACM.

[7] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15:1036–1053, 1986.

[8] S. V. Pemmaraju and T. Riaz. Using Read-\( k \) Inequalities to Analyze a Distributed MIS Algorithm. *ArXiv e-prints*, May 2016.