**Lecture Notes for Randomized Algorithms**

# Luby's Alg. for Maximal Independent Sets using Pairwise Independence

*Last Updated by Eric Vigoda on February 2, 2006*

## 8.1 Maximal Independent Sets

For a graph $G = (V, E)$, an independent set is a set $S \subset V$ which contains no edges of $G$, i.e., for all $(u, v) \in E$ either $u \notin S$ and/or $v \notin S$. The independent set $S$ is a *maximal independent set* if for all $v \in V$, either $v \in S$ or $N(v) \cap S \neq \emptyset$ where $N(v)$ denotes the neighbors of $v$.

It's easy to find a maximal independent set. For example, the following algorithm works:

1. $I = \emptyset$, $V' = V$.
2. While ($V' \neq \emptyset$) do
   (a) Choose any $v \in V'$.
   (b) Set $I = I \cup v$.
   (c) Set $V' = V' \setminus (v \cup N(v))$.
3. Output $I$.

Our focus is finding an independent set using a parallel algorithm. The idea is that in every round we find a set $S$ which is an independent set. Then we add $S$ to our current independent set $I$, and we remove $S \cup N(S)$ from the current graph $V'$. If $S \cup N(S)$ is a constant fraction of $|V'|$, then we will only need $O(\log |V|)$ rounds. We will instead ensure that by removing $S \cup N(S)$ from the graph, we remove a constant fraction of the edges.

To choose $S$ in parallel, each vertex $v$ independently adds themselves to $S$ with a well chosen probability $p(v)$. We want to avoid adding adjacent vertices to $S$. Hence, we will prefer to add low degree vertices. But, if for some edge $(u, v)$, both endpoints were added to $S$, then we keep the higher degree vertex.

Here's the algorithm:

### The Algorithm

**Problem** : Given a graph find *a* maximal independent set.

1. $I = \emptyset$, $G' = G$.
2. While ($G'$ is not the empty graph) do
   (a) Choose a random set of vertices $S \in G'$ by selecting each vertex $v$ independently with probability $1/(2d(v))$.

(b) For every edge $(u, v) \in E(G')$ if both endpoints are in S then remove the vertex of lower degree from $S$ (Break ties arbitrarily). Call this new set $S'$.

(c) $I = I \bigcup S'$. $G' = G' \setminus (S' \bigcup N(S'))$, i.e., $G'$ is the induced subgraph on $V' \setminus (S' \bigcup N(S'))$ where $V'$ is the previous vertex set.

3. output I

**Fig 1: The algorithm**

**Correctness** : We see that at each stage the set $S'$ that is added is an independent set. Moreover since we remove, at each stage, $S' \bigcup N(S')$ the set $I$ remains an independent set. Also note that all the vertices removed from $G'$ at a particular stage are either vertices in $I$ or neighbours of some vertex in $I$. So the algorithm always outputs a maximal independent set. We also note that it can be easily parallelized on a CREW PRAM.

## 8.2 Expected Running Time

In this section we bound the expected running time of the algorithm and in the next section we derandomize it. Let $G_j = (V_j, E_j)$ denote the graph after stage $j$.

**Main Lemma** : For some $c < 1$,

$$\mathrm{E}\,(\ |E_j|\ \mid\ E_{j-1}\ ) < c|E_{j-1}|.$$

Hence, in expectation, only $O(\log m)$ rounds will be required, where $m = |E_0|$.

We say vertex $v$ is BAD if more than $2/3$ of the neighbors of $v$ are of higher degree than $v$. We say an edge is BAD if both of its endpoints are bad; otherwise the edge is GOOD.

The key claims are that at least half the edges are GOOD, and each GOOD edge is deleted with a constant probability. The main lemma then follows immediately.

**Lemma 8.1** *At least half the edges are GOOD.*

**Proof:** Denote the set of bad edges by $E_B$. We will define $f : E_B \to \binom{E}{2}$ so that for all $e_1 \neq e_2 \in E_B$, $f(e_1) \cap f(e_2) = \emptyset$. This proves $|E_B| \leq |E|/2$, and we're done.

The function $f$ is defined as follows. For each $(u, v) \in E$, direct it to the higher degree vertex. Break ties as in the algorithm. Now, suppose $(u, v) \in E_B$, and is directed towards $v$. Since $v$ is BAD, it has at least twice as many edges out as in. Hence we can pair two edges out of $v$ with every edge into $v$. This gives our mapping. ■

**Lemma 8.2** *If $v$ is GOOD then* $\Pr(\ N(v) \cap S \neq \emptyset\ ) \geq 2\alpha$*, where* $\alpha := \frac{1}{2}(1 - e^{-1/6})$.

**Proof:** Define $L(v) := \{w \in N(v)\ \mid\ d(w) \leq d(v)\}$.

By definition, $|L(v)| \geq \frac{d(v)}{3}$ if $v$ is a GOOD vertex.

$$\begin{aligned}
\Pr(\ N(v) \cap S \neq \emptyset\ ) &= 1 - \Pr(\ N(v) \cap S = \emptyset\ ) \\
&= 1 - \prod_{w \in N(v)} \Pr(\ w \notin S\ ) \quad \text{using full independence} \\
&\geq 1 - \prod_{w \in L(v)} \Pr(\ w \notin S\ ) \\
&= 1 - \prod_{w \in L(v)} \left(1 - \frac{1}{2d(w)}\right) \\
&\geq 1 - \prod_{w \in L(v)} \left(1 - \frac{1}{2d(v)}\right) \\
&\geq 1 - \exp(-|L(v)|/2d(v)) \\
&\geq 1 - \exp(-1/6),
\end{aligned}$$

■

Note, the above lemma is using full independence in its proof.

**Lemma 8.3** $\Pr(\ w \notin S' \ \mid\ w \in S\ ) \leq 1/2$.

**Proof:** Let $H(w) = N(w) \setminus L(w) = \{z \in N(w) : d(z) > d(w)\}$.

$$\begin{aligned}
\Pr(\ w \notin S' \ \mid\ w \in S\ ) &= \Pr(\ H(w) \cap S \neq \emptyset \ \mid\ w \in S\ ) \\
&\leq \sum_{z \in H(w)} \Pr(\ z \in S \ \mid\ w \in S\ ) \\
&\leq \sum_{z \in H(w)} \frac{\Pr(\ z \in S, w \in S\ )}{\Pr(\ w \in S\ )} \\
&= \sum_{z \in H(w)} \frac{\Pr(\ z \in S\ ) \Pr(\ w \in S\ )}{\Pr(\ w \in S\ )} \quad \text{using pairwise independence} \\
&= \sum_{z \in H(w)} \Pr(\ z \in S\ ) \\
&= \sum_{z \in H(w)} \frac{1}{2d(z)} \\
&\leq \sum_{z \in H(w)} \frac{1}{2d(v)} \\
&\leq \frac{1}{2}.
\end{aligned}$$

■

**Lemma 8.4** *If $v$ is GOOD then* $\Pr(\ v \in N(S')\ ) \geq \alpha$

**Proof:** Let $V_G$ denote the GOOD vertices. We have

$$\begin{aligned}
\Pr(\ v \in N(S') \mid v \in V_G\ ) &= \Pr(\ N(v) \cap S' \neq \emptyset \mid v \in V_G\ ) \\
&= \Pr(\ N(v) \cap S' \neq \emptyset \mid N(v) \cap S \neq \emptyset, v \in V_G\ ) \Pr(\ N(v) \cap S \neq \emptyset \mid v \in V_G\ ) \\
&\geq \Pr(\ w \in S' \mid w \in N(v) \cap S, v \in V_G\ ) \Pr(\ N(v) \cap S \neq \emptyset \mid v \in V_G\ ) \\
&\geq (1/2)(2\alpha) \\
&= \alpha
\end{aligned}$$

■

**Corollary 8.5** *If $v$ is GOOD then the probability that $v$ gets deleted is at least $\alpha$.*

**Corollary 8.6** *If an edge $e$ is GOOD then the probability that it gets deleted is at least $\alpha$.*

**Proof:**

$$\Pr(\ e = (u,v) \in E_{j-1} \setminus E_j\ ) \geq \Pr(\ v \text{ gets deleted}\ ).$$

■

We now re-state the main lemma :

**Main Lemma** :

$$\mathrm{E}(\ |E_j| \mid E_{j-1}\ ) \leq |E_{j-1}|(1 - \alpha/2).$$

**Proof:**

$$\begin{aligned}
\mathrm{E}(\ |E_j| \mid E_{j-1}\ ) &= \sum_{e \in E_{j-1}} 1 - \Pr(\ e \text{ gets deleted}\ ) \\
&\leq |E_{j-1}| - \alpha |GOOD \text{ edges}| \\
&\leq |E_{j-1}|(1 - \alpha/2).
\end{aligned}$$

■

The constant $\alpha$ is approximately 0.07676.

Thus,

$$\mathrm{E}(\ |E_j|\ ) \leq |E_0|(1 - \frac{\alpha}{2})^j \leq m \exp(-j\alpha/2) < 1,$$

for $j > \frac{2}{\alpha} \log m$. Therefore, the expected number of rounds required is $\leq 4m = O(\log m)$.

## 8.3 Derandomizing MIS

The only step where we use full independence is in Lemma 8.2 for lower bounding the probability that a *GOOD* vertex gets picked.The argument we used was essentially the following :

**Lemma 8.7** *Let $X_i$, $1 \leq i \leq n$ be $\{0,1\}$ random variables and $p_i := \Pr(\ X_i = 1\ )$. If the $X_i$ are fully independent then*

$$\Pr\left( \sum_1^n X_i > 0 \right) \geq 1 - \prod_1^n (1 - p_i)$$

Here is the corresponding bound if the variables are pairwise independent

**Lemma 8.8** *Let $X_i$, $1 \leq i \leq n$ be $\{0,1\}$ random variables and $p_i := \Pr(\ X_i = 1\ )$. If the $X_i$ are pairwise independent then*

$$\Pr\left(\ \sum_1^n X_i > 0\ \right) \geq \frac{1}{2}\min\{\frac{1}{2}, \sum_i p_i\}$$

**Proof:** Suppose $\sum_i p_i \leq 1/2$. Then we have the following, (the condition $\sum_i p_i \leq 1/2$ will only come into some algebra at the end)

$$\begin{aligned}
\Pr\left(\ \sum_i X_i > 0\ \right) &\geq \sum_i \Pr(\ X_i = 1\ ) - \frac{1}{2}\sum_{i\neq j} \Pr(\ X_i = 1, X_j = 1\ ) \\
&= \sum_i p_i - \frac{1}{2}\sum_{i\neq j} p_i p_j \\
&\geq \sum_i p_i - \frac{1}{2}\left(\sum_i p_i\right)^2 \\
&= \sum_i p_i \left(1 - \frac{1}{2}\sum_i p_i\right) \\
&\geq \frac{1}{2}\sum_i p_i \quad \text{when } \sum_i p_i \leq 1.
\end{aligned}$$

If $\sum_i p_i > 1/2$, then we restrict our index of summation to a set $S \subseteq [n]$ such that $1/2 \leq \sum_{i\in S} p_i \leq 1$, and the same basic argument works. Note, if $\sum_i p_i > 1$, there always must exist a subset $S \subseteq [n]$ where $1/2 \leq \sum_{i\in S} p_i \leq 1$ ■

Using the construction described earlier we can now derandomize to get an algorithm that runs in $O(mn/\alpha)$ time (this is asymptotically as good as the sequential algorithm). The advantage of this method is that it can be easily parallelized to give an $NC^2$ algorithm (using $O(m)$ processors).

## 8.4 History and Open Questions

The $k-$wise independence derandomization approach was developed in [CG], [ABI] and [L85]. The maximal independence problem was first shown to be in NC in [KW].They showed that MIS in is $NC^4$. Subsequently, improvements on this were found by [ABI] and [L85]. The version described in these notes is the latter.

The question of whether MIS is in $NC^1$ is still open.

## References

[1] N Alon, L. Babai, A. Itai, "A fast and simple randomized parallel algorithm for the Maximal Independent Set Problem", *Journal of Algorithms*, Vol. 7, 1986, pp. 567-583.

[2] B. Chor, O. Goldreich, "On the power of two-point sampling", *Journal of Complexity*, Vol. 5, 1989, pp.96-106.

[3] R.M. Karp, A. Wigderson, *A fast parallel algorithm for the maximal independent set problem*, Proc. 16th ACM Symposium on Theory of Computing, 1984, pp. 266-272.

[4] M. Luby, *A simple parallel algorithm for the maximal independent set problem*, Proc, 17th ACM Symposium on Theory of Computing, 1985, pp. 1-10.