ANALYSIS OF THE GREEDY LIST COLORING ALGORITHM ON A RANDOM GRAPH

MICHAEL KANE

As a special project for Spring and Summer 2007 David Pollard and I have been studying a graph coloring algorithm proposed by Achlioptas and Molloy (1997). By relying on heuristic arguments, they claimed to be able to find the probability of success for a proposed graph coloring algorithm on Erdös-Rényi graphs. This paper is a progress report detailing our attempts to rigorously derive similar success-probabilities in the special case of the 3-color problem.

1. The Algorithm and its Representation

For a graph with n vertices, the Achlioptas and Molloy algorithm attempts to color each vertex such that no vertex has the same color of any as any of its neighbors. The algorithm begins by associating each vertex with its own list of possible colorings $\{A, B, C\}$. Let L_v be the the color list for vertex v. When a vertex is colored it will be referred to as having a fixed color. When a vertex has a fixed color its list size can be regarded as zero. This will allow us to distinguish between a vertex with 1 possible coloring and a vertex that is assigned a color. For a graph with n vertices, the algorithm is defined as:

Date: October 15, 2008.

```
for n iterations do
    Choose uniformly at random a vertex v from vertices with smallest
        list size greater than zero;
    Pick a color ζ uniformly at random from L_v;
    Fix v with color ζ;
    foreach vertex u not in the set of fixed-color vertices do
        if u is a neighbor of v then
            Remove ζ from u's color list, that is,
                change L_u to L_u \ {ζ};
            if L_u is empty then
                The algorithm fails;
            end
        end
    end
end
```

Algorithm 1: The Greedy Graph Coloring Algorithm

If the algorithm executes for all n iterations and does not reach the fail state, the algorithm succeeds. It should be noted that for $1 \leq t \leq n$, $S_L(t) = t$.

Let $\mathcal{L}_\alpha$ denote the set of α-color combinations. For example, $\mathcal{L}_2 = \{\{A, B\}, \{A, C\}, \{B, C\}\}$. Define $\mathcal{L}_1$ and $\mathcal{L}_3$ similarly. Then, let $S_\beta(t)$, with $\beta \in \mathcal{L}_2$, denote the number of vertices at time t with 2-color list β. Denote the size of individual 1-color lists similarly. Employing a slight abuse of notation, let $S_3(t)$, $S_2(t)$, and $S_1(t)$ denote the number of vertices with, 3-color, 2-color, and 1-color lists respectively. $S_F(t)$ denotes the number of vertices with empty color list and $S_L(t)$ denotes the number of vertices with fixed color.

It is difficult to determine the probablity that the algorithm will succeed for any particular graph. However, Achlioptas and Molloy were able to make probability assertions when the algorithm is run on a particular type of random graph, the Erdös-Rényi graph.

For the ER graph on n vertices, each $\binom{n}{2}$ possible edges is included with probability c/n. We can delay the selection of edges connected to a vertex v until the neighbors of v are found in the algorithm. In this way the algorithm becomes a Markov chain with state space given by the set of vertex counts. The rest of this report assumes an ER graph in the analysis.

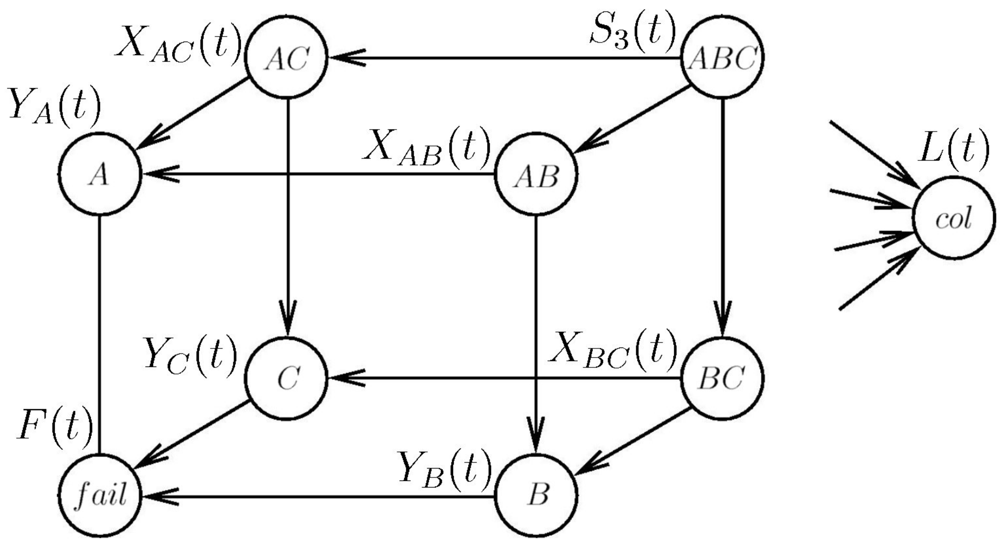

FIGURE 1. A Representation of the Progress of the Algorithm

2. WHEN $c < 1$ THE ALGORITHM SUCCEEDS WITH HIGH PROBABILITY

The algorithm starts by selecting a vertex in the set of vertices with color-list of size 3. This set will be referred to as the 3-stack. Since the fixed-color vertex's neighbors lose a possible fixed color, they move to a 2-stack. There is a c/n probability of a vertex being a neighbor of another vertex in the graph. Since there are always at most n uncolored vertices, at time t the number of 3-color vertices that become 2-color vertices is stochastically dominated by a random variable with Bin$(n, c/n)$ distribution. For the case where $c < 1$, the the expected number of vertices which go from the 3-stack to a 2-stack is less than 1 at each time step. Since the algorithm moves 1 vertex to the fixed color stack at any time, when a vertex is moved to the 2-color stack it is quickly moved to the fixed color stack in the next time step. This implies that, with high probability, when $c < 1$ the algorithm will succeed.

3. 1-STACKS STAY SMALL UNTIL ORDER $n^{1/3}$

For a vertex to be a member of a 1-stack at time t, it must have been chosen twice to have a color from its color-list removed. Each of the other $t-2$ times it was not selected to be colored and was not adjacent to a vertex being colored. This means that the probability of a given vertex being in a

1-stack at time t is at least

$$p_1(t) = \binom{t}{2}\left(\frac{c}{n}\right)^2 \tag{3.1}$$

This implies that the expected number of 1-stack vertices up to time m_1 is less than

$$\sum_{t=1}^{m_1} np_1(t) \leq \sum_{t=1}^{m_1} \frac{t^2}{2}\frac{c^2}{n} = \frac{c^2}{6n}m_1^3 \tag{3.2}$$

For a given $\varepsilon_1 > 0$, if an m_1 is chosen as a suitably small multiple of $n^{1/3}$, we can ensure that $\mathbb{P}\{S_1(m_1) > 0\} < \varepsilon_1$.

4. The Probability of Failure is Small Until Order $n^{2/3}$

For a vertex to reach the fail state at time t, it must have been chosen three times to have a color-list removed. This means that the probability of a vertex reaching the fail state at time t is at least

$$p_f(t) = \binom{t}{3}\left(\frac{c}{n}\right)^3.$$

Then, the expected number of vertices in the fail state at time t is dominated by

$$\mathbb{P}S_F(t) \leq n\binom{t}{3}\left(\frac{c}{n}\right)^3 \leq \frac{t^3}{6}\frac{c^3}{n^2}. \tag{4.1}$$

For a given $\varepsilon_F > 0$, if an m_F is chosen as a suitably small multiple of $n^{2/3}$, we can ensure that $\mathbb{P}\{S_F(m_f) > 0\} < \varepsilon_F$

5. An Upper Bound on the Maximum $S_2(t)$

Using the fact that the number of vertices going from the 3-stack to a 2-stack is stochastically dominated by $b_t \sim \text{Bin}(n, c/n)$,

$$\max_{1\leq t\leq \tilde{t}} S_2(t) \leq \sum_{t=1}^{\tilde{t}} b_t \tag{5.1}$$

and therefore to get an upper bound on the biggest $S_2(t)$ it is sufficient to get a bound on the size of the sum of b_t. This can be done by centering each of the b_t values and then applying the Bennett Inequality.

Following the justification for the Bennet inequality given in (Pollard 2001, Chapter 11), if $X \sim \text{Bin}(n, p)$, then

$$\mathbb{P}\{X \geq \varepsilon\} \leq \exp\left(-\frac{(\varepsilon - np)^2}{2np(1-p)}\psi\left(\frac{n(\varepsilon - np)}{np(1-p)}\right)\right) \tag{5.2}$$

where

$$\psi(x) := \begin{cases} 2\left((1+x)\log(1+x) - x\right)/x^2 & \text{for } x \geq -1 \text{ and } x \neq 0 \\ 1 & \text{for } x = 0. \end{cases}$$

If $x \leq 17$ then $\psi(x) > 1/4$. For the binomial case this implies that when $\varepsilon + np \leq 17$, Equation (5.2) becomes

$$\mathbb{P}\{X \geq \varepsilon + np\} \leq \exp\left(-\frac{(\varepsilon - np)^2}{4np(1-p)}\right)$$

between $\varepsilon = np$ and $\varepsilon - np \leq 17$ there is a sharp decrease in the probability that X will be larger than ε. In otherwords, the probablity that X will is larger than its expected becomes small quickly in ε.

Getting back to the case of $S_2(t)$, it can be seen that the sum of binomial term in Equation (5.1) is distributed as $\text{Bin}(n\tilde{t}, c/n)$. This implies that at time $\tilde{t}$ there is, at most, only a small probability that $S_2(t)$ will be greater than $\tilde{t}c$.

6. Clearing out 1-Stacks

From the last two sections we see that at early stages of the algorithm, all vertices have color list sizes of either 2 or 3. Equation (3.2) indicates that at some time of order $n^{1/3}$, some vertices begin to have color list sizes of 1. Since a vertex gets to a 1-stack from a 2-stack and the size of a 2-stack is relatively small at this time, it seems reasonable that for the first few times the 1-stack becomes non-empty, the number of vertices going from a 2-stack to a 1-stack is relatively small. By a time of order $n^{2/3}$ there is a danger that a vertex's color list will become empty and the algorithm will fail. This section examines the behavior of the 1-stacks at times of order $n^{2/3}$.

Let $t_0 = \varepsilon_c n^{2/3}$ be a random time. Let

$$\tau_1 = \min\{t \geq t_0 : S_1(t) = 0\}.$$

This is the first time a 1-stack becomes empty after a random time of order $n^{2/3}$. Let $s > 1$ be a number of steps after t_0. Let N be the number of vertices that drop to a 1-stack between time t_0 and $t_1 = t_0 + s$. Then

$$\{\tau_1 > t_0 + s\} \leq \{N \geq 1\} + \{S_1(t_0) \geq s\}.$$

The number of time steps to clear out the 1-stack is less than the size of the 1-stack at time t_0 plus the number of vertices that go to the 1-stack between time t_0 and t_1.

By Equation (3.1), the probability a vertex ends up in a 1-stack at time t_0 is less than $p_1(t_0)$. Since, at time t_0 there are less than n nodes that have not been colored, the distribution of $S_1(t_0)$ is stochastically dominated by a $\mathrm{Bin}(n, (ct_0)^2/2n)$ distribution. Using the results Bennet bound results form the previous section, $\mathbb{P}\{S_1(t_0) \geq s\}$ will be close to 1 in s until

$$s = (ct_0)^2/2$$

at which point the probability will decrease quickly.

From time t_0 to t_1, vertices can get to the 1-stack from either the 3-stack or the 2-stack. Let $\Delta_{t_0}B_{v,t_1}$ be the number of times vertex v is chosen to lose a color from its color from t to t_1. Let V, $V_3(t)$ and $V_2(t)$ be the set of all vertices, and the set of vertices with color list sizes 3 and 2 respectively. Then,

$$\begin{aligned}
N &= \sum_{t=t_0}^{t_1}\left(\sum_{v\in V}\{v \in V_3(t)\}\{\Delta_{t_0}B_{v,t_1} = 2\} + \{v \in V_2(t)\}\{\Delta_{t_0}B_{v,t_1} = 1\}\right)\\
&\leq \sum_{t=t_0}^{t_1}\frac{c}{n}S_3(t) + \left(\frac{c}{n}\right)^2 S_2(t)\\
&\leq \frac{sc}{n}\left(c + S_2(t_2)\right).
\end{aligned}$$

This means that

$$\mathbb{P}\{N \geq 1\} \leq \mathbb{P}\left\{S_2(t_2) \geq \frac{n}{sc} - c\right\}. \tag{6.1}$$

Again, from the Bennet bound found in the last section, when the expected value of $\sum b_t$ gets more than its mean, this probability goes to zero quickly. This means Equation (6.1) decreases sharply at

$$\frac{ct_0^2}{2} = \frac{n}{sc} - c$$

or

$$s = \frac{n}{c^2}\left(\frac{t_0^2}{2} + 1\right). \tag{6.2}$$

It may be noted that in general the bound found in Equation (6) is smaller than Equation (6.2). This means that there is a sharp decrease in the the probility of hitting τ after $t_0 + s$ when $s = (ct_0)/2$.

7. A Summary and Justification for Subsequent Sections

The previous sections it has been shown that the Achlioptas and Molloy algorithm will have empty 1-stacks up to a time of order $n^{1/3}$. At this time a 1-stack will become non-empty, but it will return to being empty after a short amount of time. It seems likely that the 1-stack is emptied because the number of vertices entering from the 2-stacks is small. We suspect that similar behavior will continue but, as time progresses, the number of vertices going from 2-stack to 1-stacks will increase. As a result it will take longer to empty the 1-stacks. At a time of order $n^{2/3}$ it is possible for the algorithm to fail. Failure will occur because as the number of vertices in 1-stacks increases it becomes more likely that a vertex will be chosen to move to the fail state.

If the algorithm does not fail, it is because the size of the 1-stack does not get too big. In the success case, the size of the 3-stacks will eventually become small and as a result, the number of vertices going to 2-stacks and 1-stacks will become negligible. If the algorithm successfully executes to this time the failure probability will go to zero.

The analysis performed in subsequent sections is motivated by the idea that to understand the probability of the algorithm failing we need to understand how the 1-stacks behave during the time interval where the algorithm is in danger of failing.

REFERENCES

Achlioptas, D. and M. Molloy (1997). The analysis of a list-coloring algorithm on a random graph (extended abstract). *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, 204–212.

Pollard, D. (2001). *A User's Guide to Measure Theoretic Probability*. Cambridge University Press.