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 $L_{v,i}$ be the the color list for vertex v at time i. 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 the tth time step of the algorithm

(1) Choose uniformly at random v from vertices smallest list of size greater than zero.
(2) Pick a color ζ uniformly at random from $L_{v,t}$.
(3) Fix v with color ζ.
(4) For each vertex u not in the set of fixed color vertices:
   - If u is a neighbor of v remove ζ from its color list. $L_u = L_u \setminus \{\zeta\}$

If at any time during the execution any vertex's color list is empty, the algorithm has failed to find a valid coloring. If at time n all vertices have a fixed color, then the algorithm has succeeded.

Let $S_3(t)$ denote the number of vertices at time t with color list $\{A, B, C\}$, $X_\alpha(t)$ denotes the number of vertices at time t with 2-color list $\alpha \in \{\{A, B\}, \{A, C\}, \{B, C\}\}$, $Y_\beta(t)$ denotes the number of vertices at

Date: September 3, 2007.

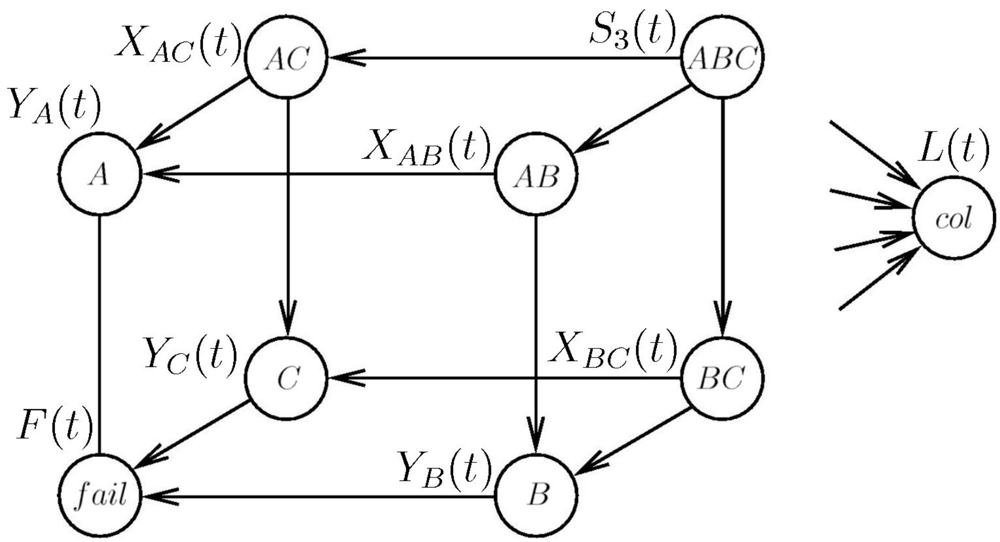

FIGURE 1. A Visualization for the List Counts

time t with 1-color list $\beta \in \{\{A\}, \{B\}, \{C\}\}$, $F(t)$ denotes the number of vertices at time t with empty color lists, and $L(t)$ denotes the number of vertices at time t which are colored.

The progress of the algorithm is shown in Figure (1). It should be noted that for all vertices with color list size greater than 0, there is a positive probability of moving to $L(t)$.

The decision to analyze Erdös-Rényi graphs is motivated by the fact that the occurrence of a specific graph structures (which may be used as counter examples to the algorithm succeeding) occur with small probability. Thus, by performing the analysis on Erdös-Rényi graphs, we are, in a sense, looking at an ”average” class of graphs that the algorithm may be used on.

The evolution of the algorithm on an Erdös-Rényi graph can be describe by a Markov Chain where the variables are the list counts defined above. Both $F(t)$ and $L(t)$ are absorbing states. At each iteration of the algorithm a vertex can be given a fixed color, it can lose a color from its color list, or its color list can remain unchanged.

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. In the Erdös-Rényi graph 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 $n^{1/3}$

Define

$$S_1(t) = \sum_{\beta \in \{\{A\},\{B\},\{C\}\}} Y_\beta(t)$$

that is, the number of vertices in the 1-stacks. 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

$$p_1(t) = \binom{t}{2} \left(\frac{c}{n}\right)^2 \left(1 - \frac{c}{n}\right)^{t-2}. \tag{3.1}$$

This implies that the number of 1-stack vertices up to time m is stochastically dominated by

$$\sum_{t=1}^{m} np_1(t) \leq \sum_{t=1}^{m} \frac{t^2}{2}\frac{c^2}{n} = \frac{c^2}{6n}m^3 + o(n)$$

By the Markov Inequality this implies that

$$\mathbb{P}\{S_1(\varepsilon_1 n^{2/3}) \geq \varepsilon_1 n^{1/3}\} \leq \frac{c^2}{6}\frac{\varepsilon_1^2}{n^{1/3}} \tag{3.2}$$

Therefore, up to a time of order $n^{1/3}$ the expected size of the 1-stacks goes to zero as n gets large; $S_1(t) = o_p(n^{1/3})$.

4. The Expected Size of $S_2(t)$

Let $b_t \sim \text{Bin}(n, c/n)$. As shown in Section (2), b_t stochastically dominates the number of vertices going from the 3-stack to a 2-stack at time t. Then the size of the 2-stack at time t is stochastically dominated by the sum of the binomial increments

$$\mathbb{P}S_2(t) \leq \mathbb{P}\sum_{i=1}^{t} b_i = ct. \tag{4.1}$$

5. The First Time the 1-Stack is Cleared Out

In Section (3) it is shown that until a time of order $n^{1/3}$ the 1-stacks remain empty. Let $t_0 = \varepsilon_1 n^{1/3}$ be the first time the 1-stack becomes non-empty. Equation (4.1) implies that expected size of $S_2(t)$ at this time is of order $n^{1/3}$. 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. This section provides some justification for this intuition.

Let

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

This is the first time a 1-stack becomes empty after it has received a vertex. Let $m > 1$ 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 $t_1 = t_0 + m$. Then

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

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.

The expected number of vertices moving to a 1-stack between time t_0 and t_1 is at most the size of the 3-stack times the probability of moving to a 1-stack during this time plus the size of the 2-stack times the probability of moving to to a 1-stack during this time.

The size of the 3-stack at any time is less than n and, by Equation (3.1) the probability of a vertex going from a 3-stack to a 1-stack in m time steps is given by $p_1(m)$. Therefore the expected number of 3-stack vertices moving to 1-stack vertices is less than $(mc)^2/(2n)$.

For a single time step t where $t_0 \leq t \leq t_1$

$$\mathbb{P}S_2(t) \leq cm,$$

and the probability of a vertex going from a 2-stack to a 1-stack is at most (c/n). Then for m time steps the expected number of 2-stack vertices moving to 1-stack vertices is less than $(c^2m^2)/n^2$

$$\mathbb{P}N \leq m^2c^2\left(\frac{1}{2n}+\frac{1}{n^2}\right)$$

Using this result and the result from Equation (3.2) it follows

$$\mathbb{P}\{\tau_1 > t_0 + m\} \leq m^2c^2\left(\frac{1}{2n}+\frac{1}{n^2}\right)+\frac{c^2\varepsilon_1^3}{6m}$$

When m is small the function is close to zero. This implies that the 1-stacks will become empty shortly after the first time it becomes non-empty.

6. 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 stochastically dominated by

$$n\binom{t}{3}\left(\frac{c}{n}\right)^3 \leq \frac{t^3}{6}\frac{c^3}{n^2}.$$

By the Markov Inequality this implies

$$\mathbb{P}\{F(\varepsilon_f n^{2/3}) \geq \varepsilon_f n^{2/3}\} \leq \frac{\varepsilon_f^2 c^3}{6n^{2/3}}$$

This means that up to a time of order $n^{2/3}$ the probability that the algorithm will fail goes to zero as n gets large; $F(t) = o_p(n^{2/3})$.

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.

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

Define b_t as before. Using the fact that the number of vertices going from the 3-stack to a 2-stack is stochastically dominated by b_t,

$$\max_{1 \le t \le n\varepsilon} S_2(t) \le \sum_{t=1}^{n\varepsilon} b_t$$

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 (Pollard 2001, Chapter 11)

$$\begin{aligned} \mathbb{P}\left\{\sum_{t=1}^{n\varepsilon} b_t \ge \nu\right\} &= \mathbb{P}\left\{\sum_{t=1}^{n\varepsilon} b_t - n\varepsilon c \ge \nu - n\varepsilon c\right\} \\ &\le \exp\left(-\frac{(\nu - n\varepsilon c)^2}{2c(1-c/n)}\psi\left(\frac{n(\nu - n\varepsilon c)}{c(1-c/n)}\right)\right) \end{aligned}$$

where

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

As ν gets larger than the expected value of the sum of b_t's, that is $\nu > n\varepsilon c$, there is a sharp decrease in the probability of getting an $S_2(t)$ larger than ν. Therefore, we expect that $S_2(t)$ will not be much bigger than the sum of the expected values of the binomial increments, b_t.

9. The Restricted Model

Until now, only 3-stack, 2-stack, 1-stack and fail stack counts have been analyzed. In this an subsequent sections, the behavior of specific color-list counts will be studied. It is not immediately clear how to analyze all list counts in the algorithm, and managing all of these variables may become unwieldy. Therefore, as a first-pass, consider a simplified Markov chain similar to that in Figure 1 consisting of only $S_3(t)$, $X_\alpha(t)$ for $\alpha \in \{A, B, C\}$, and Y_β for $\beta \in \{\{A, B\}, \{A, C\}, \{B, C\}\}$. This means that a given vertex may go from the 3-color stack to a 2-color stack, the 3-color stack to the fixed color stack, or a 2-color stack to the fixed color stack. This will be referred to as the restricted model. To begin with, the added simplification of no fixed-color stack will be imposed. In this case a "colored" vertex will retain its color list and the algorithm will proceed. It will not be added to the fixed-color stack. After the simplified model is well understood, the fixed-color stack will be analyzed.

In the first step of the algorithm, $t = 1$ in the restricted model with no fixed-color stack, one of the n vertices is chosen uniformly at random and assigned a color which is chosen uniformly at random from 1 of its 3 colors. This color is removed from the color-lists of each of its adjacent vertices. For $t > 1$, a vertex with color lists $\{A, B\}$, $\{A, C\}$, or $\{B, C\}$ is picked to be colored proportional to its respective list count. In other words, for vertices with color list $\{A, B\}$ at time t, denoted $V_{AB}(t)$,

(9.1)

$$\mathbb{P}\{\text{a vertex in } V_{AB}(t) \text{ is colored at } t{+}1 | X_{AB}(t), X_{AC}(t), X_{BC}(t)\} = \frac{X_{AB}(t)}{S_2(t)}.$$

As before, let b_{t+1} be the number of vertices that go from 3-color lists to 2-color lists at time $t + 1$. The probability that $X_{AC}(t + 1)$ is selected to receive b_{t+1} vertices from $S_3(t)$ is the probability that a vertex with color list $\{A, B\}$ or $\{B, C\}$ is picked to be colored times the probability that the color picked is B. Write $\mathbb{P}_t$ for the conditional probability conditioned on information through time t. Then

$$\begin{aligned}\mathbb{P}_t\{X_{AC}(t + 1) = X_{AC}(t) + b_{t+1}\} &= \frac{1}{2}\left(\frac{X_{AB}(t) + X_{BC}(t)}{S_2(t)}\right) \\ &= \frac{1}{2}\left(1 - \frac{X_{AB}(t)}{S_2(t)}\right).\end{aligned}$$

Let X_t be $(X_{AB}(t), X_{AC}(t), X_{BC}(t))$ and let δ_{t+1} indicate which stack get b_{t+1} items at $t+1$. Then

$$\pi_t = \mathbb{P}_t \delta_{t+1} = \frac{1}{2}.\left(1 - X_t/S_2(t)\right)$$

For the restricted model with no fixed-color stack, the transition to a state at time $t+1$ is controlled by two conditionally independent quantities b_{t+1} and δ_{t+1} Where

$$\begin{aligned}\pi_t &= (\pi_{AB}(t), \pi_{AC}(t), \pi_{BC}(t)) \\ &= \begin{cases} (1/3, 1/3, 1/3) \text{ if } t = 0 \\ \frac{1}{2}\left(1 - X_t/S_2(t)\right) \text{ if } t \geq 1. \end{cases}\end{aligned}$$

If at a given time t, $X_{AB}(t)$ is much larger than both $X_{AC}(t)$ and $X_{BC}(t)$, then a vertex with color list $\{A, B\}$ is more likely to be selected for coloring. Therefore, for some $i > 0$, $X_{AB}(t+i)$ is unlikely to receive more items until its size is closer to $X_{AC}(t+i)$ or $X_{BC}(t+i)$. After the distributions of $S_3(t)$, $S_2(t)$, and b_t are established in the next subsection a more formal justification of this balancing mechanism is provided in Section (9.2).

9.1. **The Distribution of $S_3(t)$, $S_2(t)$, and b_t.** To find the distribution of $S_3(t)$, the 3-stack size, start by looking at the probability a vertex remains in the 3-stack through time t. For this to happen it must survive t independent attempts at removing it. For each of these attempts a 3-stack vertex has probability $(1 - c/n)$ of of remaining in the 3-stack. Therefore the size of the 3-stack is

$$S_3(t+1) \sim \text{Bin}(n, (1 - c/n)^t).$$

In the simplified, no fixed-color state, model

$$S_2(t+1) = n - S_3(t+1)$$

and therefore, distribution of $S_2(t+1)$ is

$$S_2(t+1) \sim \text{Bin}(n, 1 - (1 - c/n)^t)$$

Until now the number of vertices moving from the 3-stack to the 2-stack, b_{t+1} has been approximated as $\text{Bin}(n, c/n)$ since at each step of the algorithm, there is a c/n chance that a vertex in a 3-stack will move to a 2-stack, and there are at least n 3-stack vertices. A more accurate estimate is comes from realizing that at time t there are $S_3(t)$ possible 3-stack vertices. Therefore, a more accurate estimate of b_{t+1} is given by

$$b_{t+1} \sim \text{Bin}(S_3(t), c/n).$$

This approximation will be used for the remainder of this paper.

9.2. $X_1(t)$, $X_2(t)$, and $X_3(t)$ Remain Balanced when Colors are Not Fixed.

Define

$$\Delta_{t+1}X = X_{t+1} - X_t = b_{t+1}\delta_{t+1}.$$

In order to show that $X_{AB}(t) \approx X_{AC}(t) \approx X_{BC}(t)$ with high probability, consider the differences between 2-color stack sizes. The behavior of

$$Z_t = X_{AB}(t) - X_{BC}(t)$$

is typical.

Let

$$\gamma_{t+1} = \delta_{AB}(t+1) - \delta_{BC}(t+1)$$

and

$$\Delta_{t+1}Z = Z_{t+1} - Z_t.$$

The conditional probability of Z_t by $\Delta_{t+1}Z$ is then

$$\begin{aligned}
\mathbb{P}_t Z_t \Delta_{t+1} Z &= Z_t \mathbb{P}_t \Delta_{t+1} Z \\
&= Z_t \mathbb{P}_t b_{t+1} \mathbb{P}_t \gamma_{t+1} \\
&= Z_t S_3(t) \frac{c}{n} \frac{1}{2} (1 - X_1(t)/S_2(t) - 1 + X_3(t)/S_2(t)) \\
&= -\frac{1}{2} \frac{c}{n} \frac{S_3(t)}{S_2(t)} Z_t^2 \\
&\leq 0
\end{aligned}$$

This implies that if, at some time, the difference $|X_{AB}(t) - X_{BC}(t)|$ gets large, the stack selection process ensures that it will not remain large for long since subsequent iterations of the algorithm will tend to reduce this difference. Also, since the choice of stacks used to define Z_t was arbitrary, this negative feedback mechanism exists between all two-color each pair of elements in X_t. Now, if it can be shown that the expected conditional differences $\mathbb{P}_t Z_{t+1}^2$ cannot grow too quickly, it can be concluded that the Z_t tends to be around zero.

9.3. $\mathbb{P}_t Z_{t+1}^2$ **is Bounded for by a Constant When Colors are Not Fixed.** We would like to show the expected conditional differences $\mathbb{P}_t Z_{t+1}^2$ cannot grow too quickly in t

$$\begin{aligned}
\mathbb{P}_t Z_{t+1}^2 &= \mathbb{P}_t (Z_t + \Delta_{t+1} Z)^2 \\
&= Z_t^2 + Z_t \mathbb{P}_t \Delta_{t+1} Z + \mathbb{P}_t \Delta_{t+1} Z^2 \\
&\leq Z_t^2 + \mathbb{P}_t b_{t+1}^2 \mathbb{P}_t \gamma_{t+1}^2 \\
&\leq Z_t^2 + \mathbb{P}_t b_{t+1}^2 \text{ since } \gamma_{t+1}^2 \leq 1 \\
&\leq Z_t^2 + (c + c^2) \text{ since } \mathbb{P}_t b_{t+1} \leq c + c^2.
\end{aligned}$$

From this we can conclude

$$\mathbb{P} Z_{t+1}^2 \leq \mathbb{P} Z_0^2 + t(c + c^2).$$

9.4. **Balance in the Fixed Color Case.** The previous 2 sections show that a balance mechanism exists in the 2-stacks which keeps their sizes comparable when colored vertices are not removed from the system. We would like to extend this to the case where colored vertices are moved to a fixed color state and removed from the 2 and 3-stack counts.

From Equation 9.1 we know that a vertex in a 2-stack is colored is proportional to its size. A new $\Delta_{t+1} X$ can then be defined as

$$\Delta_{t+1} X = X_{t+1} - X_t = b_{t+1} \delta_{t+1} - X_t / S_2(t)$$

Defining γ_{t+1} as before

$$\begin{aligned}
\mathbb{P}_t Z_t \Delta_{t+1} Z &= Z_t (\mathbb{P}_t b_{t+1} \gamma_{t+1} - Z_t / S_2(t)) \\
&= \frac{1}{2} \frac{c}{n} \frac{S_3(t)}{S_2(t)} Z_t^2 - \frac{1}{S_2(t)} Z_t^2
\end{aligned}$$

The first term is the same as when the the fixed-color stack was not included and along with this negative term there is another negative term. This implies that the feedback is stronger in the model including the fixed-color stack.

Unfortunately, we cannot recursively bound the second moment of Z_{t+1} when the fixed-color stack is introduced. The problem is that the conditional second moment of Z_{t+1} is a function of $Z_t / S_2(t)$ and since the 2-stack size of zero may occur with positive probability we cannot give an upper bound for the $1/S_2(t)$ term.

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.