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). Relying on heuristic arguments, they claimed to be able to find the probability of success for their 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 of its neighbors. The algorithm begins by associating each vertex with its own list of possible colorings $\{A, B, C\}$. 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. Let $L_{v,i)}$ be the the color list for vertex v at time i. 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.

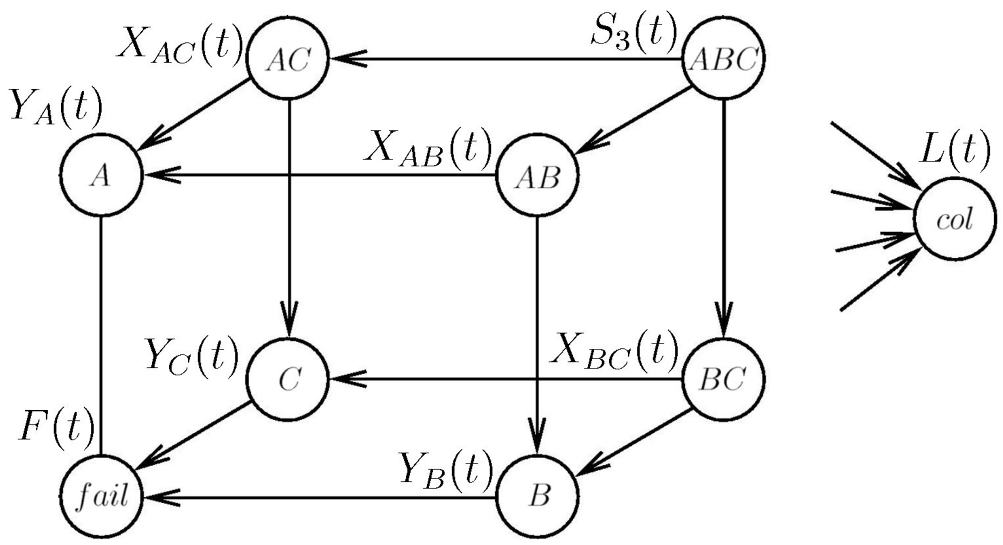

FIGURE 1. A Visualization for the List Counts

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 $L(t)$.

For Erdös-Rényi graphs there is an averaging effect such that the occurence 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 and at each iteration of the algorithm a vertex can be colored, 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 fixing the color of a vertex in the 3-stack. Since the colored vertices 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 least 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 with color-lists of size 1. These will be referred to as 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

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

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

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

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 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; $F(t) = o_p(n^{2/3})$.

5. 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 stochasitcally dominated by the sum of the binomial increments

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

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

In Section (3) it is clear that until a time of order $n^{1/3}$ the 1-stacks remain empty. In the previous section it is shown 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 it is reasonable to assume that for the first few times, the number of vertices going from a 2-stack to a 1-stack is relatively small. Therefore, it seems helpful to analyze the first time a 1-stack becomes empty after it receives a vertex.

Let $t_0 = \varepsilon_1 n^{1/3}$ be the first time the 1-stack becomes non-empty. 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 probability that a vertex moves to a 1-stack between time t_0 and t_1 is at most the size of the 3-stack times 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, the probability of a vertex going from a 3-stack to a 1-stack in m time steps is $p_1(m)$. 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 Section (3) 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 t_0.

7. A Summary and Justification for Subsequent Sections

The previous sections have shown that the Achlioptas and Molloy algorithm will run with 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 so quickly because the number of vertices entering the 1-stack 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. This will happen because as the number of vertices in 1-stacks increases it becomes more likely that the algorithm will fail.

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

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.