Note Title

6.S899 Lecture 2 04/12/2014

Previous Lecture:

✓ $O(1)$ – coloring for graphs with $\Delta = O(1)$ in $\log^* n + O(1)$ rounds.

✓ Optimality of the round complexity $\log^* n$

? But we used $\Delta^{O(\Delta)}$ colors

This Lecture:

(A) $O(\Delta^2)$ coloring in $\log^* n + O(1)$ rounds

$\Longrightarrow$ $\Delta+1$ Coloring in $O(\Delta^2) + \log^* n$ rounds

(B) $\Delta+1$ coloring in $O(\Delta \log \Delta) + \log^* n$ rounds

(C) $\Delta+1$ coloring in $O(\Delta) + \log^* n$ rounds

We start with an algorithm that finds an $O(\Delta^2 \log n)$ coloring in a single round.

Intuitive Outline: We will find a function $f = [n] \to 2^{[m]}$ that assigns to each node a subset of colors $\{1, 2, \cdots, m\}$ where $m = \alpha \Delta^2 \log n$, for a sufficiently large constant α to be fixed. These subsets will have the following special property:

$\star$ Each node has a color that none of its neighbors have.

Then each node will check with its neighbors and pick one color that the neighbors don't have and that gives our coloring.

All that's left is to show that such a function f actually <u>exists</u>. Formally, achieving the following property is sufficient:

$\forall\ x_0, x_1, \cdots, x_\Delta \in \{1, \cdots, n\}$,

$\star$ $\exists\ c \in \{1, 2, \cdots, m\}$ s.t.: (1) $c \in f(x_0)$

(2) $c \notin \bigcup_{j=1}^{\Delta} f(x_j)$

We will use the probabilistic method to show that such a function f exists. That is, we present a random construction of function f and show that with nonzero probability (in fact with very high probability), it satisfies the property $\star$.

The random construction is simple. For each $x \in \{1, \dots, n\}$, add each color $c \in \{1, \dots, m\}$ to the subset $f(x)$ randomly and independently with probability $p = 1/\Delta$.

For a given set of a node x_0 & its Δ neighbors $x_1, \dots, x_\Delta$, we say color c is good if $c \in f(x_0)$ & $\forall\, 1 \le j \le \Delta$, $c \notin f(x_j)$. Note that the probability that a fixed color c is good for $x_0, x_1, \dots, x_\Delta$ is $\frac{1}{\Delta} \cdot \left(1 - \frac{1}{\Delta}\right)^{\Delta} \geq \frac{1}{4\Delta}$

Thus, the probability that there is no color $c \in \{1, \dots, m\}$ that is good for $x_0, \dots, x_\Delta$ is at most $\left(1 - \frac{1}{4\Delta}\right)^m \le e^{-m/4\Delta} = e^{-\frac{\alpha}{4}\Delta \log n}$

Now, this was for a fixed $x_0, \dots, x_\Delta$. What is the probability that there is some $x_0, \dots, x_\Delta$, that don't have a good color?

Well, using a Union bound, we know that the probability that there is some $x_0, \dots, x_\Delta$ that don't have a good color is at most

$$\binom{n}{\Delta+1} e^{-\frac{\alpha}{4}\Delta \log n}$$

which using Stirling's inequality is at most

$$\left(\frac{en}{\Delta+1}\right)^{\Delta+1} e^{-\frac{\alpha}{4}\Delta \log n}$$

$$\leq \exp\left(\Delta+1 + (\Delta+1)\log\frac{n}{\Delta+1} - \frac{\alpha}{4}\Delta\log n\right).$$

Picking the constant α in the definition of m sufficiently large, we can make this expression strictly less than 1. Thus, there exists a function f with the desired property! ∎

The above scheme gives us an algorithm to find an $O(\Delta^2 \log n)$ coloring in a single round.

We can improve this to an $O(\Delta^2 \log \Delta)$ coloring by repeating the same idea for $\log^* n$ iterations, thus an algorithm with round complexity $\log^* n$.

The key point is, in the scheme described above, we effectively treated the ids as a coloring with palette size n, and applying the scheme for one 1 round, we obtained a palette size $O(\Delta^2 \log n)$.

Hence, if we start from a palette size m, we go to one of size $O(\Delta^2 \log m)$ in 1 round.

Therefore,

$$n \longrightarrow O(\Delta^2 \log n)$$

$$O(\Delta^2 \log n) \longrightarrow O(\Delta^2 \log \Delta) + O(\Delta^2 \log\log n)$$

$$\longrightarrow O(\Delta^2 \log \Delta) + O(\Delta^2 \log\log\log n)$$

$$\vdots$$

$$\longrightarrow O(\Delta^2 \log \Delta)$$

$\log^* n$ iterations

Now we have an $O(\Delta^2 \log \Delta)$-coloring algorithm that uses $\log^* n$ rounds. There is a neat algebraic method to make this an $O(\Delta^2)$-coloring, using one more round, but we won't get to cover it, due to the time restrictions.

$(\Delta+1)$ - Coloring

For the rest of this lecture, our benchmark will be $(\Delta+1)$-coloring. We first see a very simple method that transforms the $\log^* n$-round $O(\Delta^2 \log \Delta)$-coloring of the previous section to an $O(\Delta^2 \log \Delta) + \log^* n$ round $(\Delta+1)$-coloring. Then we will see other techniques that achieve a $(\Delta+1)$-coloring in respectively $O(\Delta \log \Delta) + \log^* n$ and $O(\Delta) + \log^* n$ rounds.

Color by Color Reduction

Given an m-coloring, we find a $(\Delta+1)$-coloring in $m-(\Delta+1)$ rounds.

one round per loop

For $c = m \to \Delta+2$

if $c_v = c$ then

$$c_v = \min \{1, \dots, \Delta+1\} \setminus \bigcup_{w \in N(v)} c_w$$

Lem: In each round, the coloring stays legal.

Proof Sketch: In each round, if a node v has $c_v = c$ & it's going to change its color, none of its neighbors changes color in that round (because the old coloring was legal). Furthermore, v will find a new color in $\{1, \ldots, \Delta+1\}$ because its neighbors occupy at most Δ colors.

Applying this trick to the above $O(\Delta^2 \log \Delta)$ alg., we get an $O(\Delta^2 \log \Delta) + \log^* n$ round $(\Delta+1)$-coloring.

$O(\Delta \log \Delta) + \log^* n$ round $(\Delta+1)$-coloring:

Above, we saw a color by color reduction trick which gets rid of one color per round. The main new idea will be to reduce colors faster. We explain an approach that transforms an m coloring, for $m \geq 2\Delta+2$, to an $m/2$ coloring in $O(\Delta)$ rounds. Applying this recursively and then applying the color-by-color reduction gives us the following result:

Given an m-coloring, we can transform it to a $(\Delta+1)$-coloring in $O(\Delta \log \frac{m}{\Delta})$ rounds.

Break the m colors into buckets with $2\Delta+2$ colors and one final bucket that has somewhere between $2\Delta+2$ and $4\Delta+3$ colors. In each bucket, use the color-by-color reduction trick to reduce the number of colors to $\Delta+1$. Furthermore, do this for all the buckets in parallel (Why can we do that?) As a result, after spending at most $3\Delta+2$ rounds, each bucket has at most $\Delta+1$ colors, which means in total we have at most $m/2$ colors. That is, a reduction of 2-factor in the palette size in $O(\Delta)$ rounds. Repeating this idea for about $\log\left(\frac{m}{\Delta}\right)$ times gets us to a $(\Delta+1)$-Coloring, thus achieving the claim.

Applying the above technique to the $O(\Delta^2 \log \Delta)$-coloring explained above gets us to an $O\left(\Delta \log \frac{\Delta^2 \log \Delta}{\Delta}\right) + \log^* n = O(\Delta \log \Delta) + \log^* n$ round $(\Delta+1)$-coloring.

What's left from this lecture is to get a $(\Delta+1)$-coloring in $O(\Delta) + \log^* n$ rounds.

$O(\Delta) + \log^* n$ round $(\Delta+1)$-coloring

The approach is very close to the one that we used for $O(\Delta^2 \log n)$-coloring in 1-round. The key insight is that, insted of going for a true legal coloring (where adjacent nodes must have different colors) from the start, we work with defective coloring but eventually decrease the defect to zero.

Defective Coloring: A coloring $c: V \to \{1, \ldots, m\}$ is called d-defective if for each node $v \in V$, the number of neighbors of v that have a coloring equal to v is at most d. That is, the subgraph induced by nodes of each color has max degree at most d.

Note that with this definition, a legal coloring is one with defect $d=0$, generally coloring with smaller defect is better, and closer to our goal. Moreover, any coloring has defect at most Δ.

The main technical component of the algorithm will be a scheme that given an m_1-coloring with defect d_1, in one round, it computes an m_2-coloring with defect d_2, for any $d_2 \geq d_1$, and where $m_2 = O\left(\left(\frac{\Delta - d_1}{d_2 - d_1 + 1}\right)^2 \log m_1\right)$.

The point is to decrease the palette size rapidly, while sacrificing a bit in the defect of the coloring.

We first prove that such a 1-round scheme exists. Then, we see how to use this scheme to obtain our desired $(\Delta+1)$-coloring in $O(\Delta) + \log^* n$ rounds.

From a d_1-defective m_1-coloring to a d_2-defective m_2-coloring where $d_2 \geq d_1$ & $m_2 = O\left(\left(\frac{\Delta - d_1}{d_2 - d_1 + 1}\right)^2 \log m_1\right)$:

We will use an approach close to the one we used at the start of the lecture to get an $O(\Delta^2 \log n)$ coloring.

We will find a function $f : \{1, \dots, m_1\} \rightarrow 2^{\{1, \dots, m_2\}}$

such that

(1) $\forall x_0 \in \{1, \dots, m_1\}$, $|f(x_0)| = \Omega\left(\left(\frac{\Delta - d_1}{d_2 - d_1 + 1}\right) \log m_1 \right)$

(2) $\forall x_0, x_1 \in \{1, \dots, m_1\}$, $x_0 \neq x_1$, $|f(x_0) \cap f(x_1)| = O(\log m_1)$

Before proving that such a function exists, let us first see why this function would immediately imply our desired 1-round coloring scheme:

Consider a node v and its neighbors $u_1, u_2, \dots, u_\delta$, where $\delta \leq \Delta$. At most d_1 of these neighbors have the same initial color as v. We will not insist on v getting a new color different than those, but we will make sure that there are at most $d_2 - d_1$ other neighbors that get the same new color as v, hence the new defect is at most $d_1 + (d_2 - d_1) = d_2$.

For that, suppose $c_v^{old} = x_0$. We pick a color c in $f(x_0)$ such that there are at most $d_2 - d_1$ neighbors u_i of v that have this color c in $f(c_{u_i}^{old})$.

The reason that such a color c exists is there $\Omega\left(\frac{\Delta - d_1}{d_2 - d_1 + 1} \log m_1\right)$ colors in $f(x_0)$ and for each neighbor

u_i that for a color $x_i \neq x_0$, $|f(x_i) \cap f(x_0)| = O(\log m_1)$.

Note that there are $\Delta - d_1$ neighbors u_i with a color different than x_0. Each can occupy at most $O(\log m_1)$ colors of $f(x_0)$ & $f(x_0)$ has $\Omega(\frac{\Delta - d_1}{d_2 - d_1 + 1} \log m_1)$ colors. So, there is one color that is not occupied more than $d_2 - d_1 + 1$ times. ∎

Now we prove that a function $f : \{1, \dots, m_1\} \to 2^{\{1, \dots, m_2\}}$ exists s.t.:

(1) $\forall x_0 \in \{1, \dots, m_1\}$, $|f(x_0)| = \Omega\left(\left(\frac{\Delta - d_1}{d_2 - d_1 + 1}\right) \log m_1\right)$

(2) $\forall x_0, x_1 \in \{1, \dots, m_1\}$, $x_0 \neq x_1$, $|f(x_0) \cap f(x_1)| = O(\log m_1)$

Again we use the probabilistic method.

Let $p = \frac{d_2 - d_1 + 1}{\alpha(\Delta - d_1)}$, for a sufficiently large constant α, and let $m_2 = \frac{100}{p^2} \log m_1$. Then, for each $x_i \in \{1, \dots, m_1\}$, define $f(x_i)$ by including each $c \in \{1, \dots, m_2\}$ in $f(x_i)$ randomly & independently with probability p.

We have: $E[|f(x_i)|] = m_2 p = \frac{100 \log m_1}{p}$.

Using a Chernoff bound, we get

$$Pr\left[|f(x_i)| \leq 50 \log m_1 / p \right] \leq e^{-5 \log m_1}.$$

Hence, using a union bound over all choices of x_i, we see that the probability that we satisfy property (1) is at least $1 - 1/m_i$.

Similarly, $\mathbb{E}\left[|f(x_i) \cap f(x_j)|\right] = 100 \log m_i$, for each fixed x_i & x_j, $x_i \neq x_j$. Thus,

$$\Pr\left[|f(x_i) \cap f(x_j)| \geq 200 \log m_i\right] \leq e^{-5 \log m_i}.$$

Again, a union bound over all $\binom{m_i}{2}$ choices of x_i & x_j shows that we satisfy property (2) with probability at least $1 - 1/m_i$.

Overall, we satisfy both properties with probability at least $1 - 2/m_i$, which finishes the proof of existence of the function f.

All that is left is to actually use 1-round defective coloring scheme to obtain an $O(\Delta) + \log^* n$ round $(\Delta+1)$-coloring.

In the following, for simplicity, we only explain how to get a $O(\Delta \underbrace{\log \log \log \cdots \log}_{5 \text{ times}} \Delta) + \log^* n$-round $(\Delta+1)$-coloring.

Actually getting the $O(\Delta) + \log^* n$ round algorithm extends the same idea but has more detailed parameter setting steps.

$O(\Delta \underbrace{\log\log\cdots\log}_{5\text{ times}} \Delta) + \log^* n$ round $(\Delta+1)$-coloring:

We start from the $\log^* n + O(1)$ round $O(\Delta^2 \log \Delta)$-coloring that we obtained at the start of this lecture. Note that this a 0-defective coloring.

Then, we turn this into a $\frac{\Delta}{10}$-defective coloring with $O\left(\left(\frac{\Delta - 0}{\frac{\Delta}{10} - 0 + 1}\right)^2 \log\left(\Delta^2 \log \Delta\right)\right) = O(\log \Delta)$ colors in one round. Then, in one more round, we get to a $\frac{2\Delta}{10}$-defective coloring with $O\left(\left(\frac{\Delta - \Delta/10}{\frac{2\Delta}{10} - \frac{\Delta}{10} + 1}\right)^2 \log(\log \Delta)\right) = O(\log\log \Delta)$ colors.

Repeating the same idea for 5 times, we get to a $\frac{\Delta}{2}$-defective coloring with $O(\underbrace{\log\log\cdots\log}_{5\text{ times}} \Delta)$ colors, in $\log^* n + O(1)$ rounds.

Now, for each color, the nodes of that color induce a coloring with maximum degree at most $\frac{\Delta}{2}$. Hence, by recursion, we can get a $\frac{\Delta}{2}+1$ coloring for them in $T(\Delta/2)$ rounds.

Moreover, we can do this for all the $O(\log\log\cdots\log\Delta)$ colors in parallel. Thus, after $T(\Delta/2)$ rounds, we get a coloring with a total of $O(\log\cdots\log\Delta)\cdot(\frac{\Delta}{2}+1) = O(\Delta\log\cdots\log\Delta)$ colors. Using the color by color reduction we can turn this in a $\Delta+1$ coloring in $O(\Delta\log\cdots\log\Delta)$ rounds. Thus, the recursion is

$$T(\Delta) = O(1) + T(\Delta/2) + O(\Delta\underbrace{\log\cdots\log}_{5\text{ times}}\Delta)$$

$$\rightarrow T(\Delta) = O(\Delta\underbrace{\log\cdots\log}_{5\text{ times}}\Delta).$$

To get an $O(\Delta)+\log^* n$ round algorithm, we need to do the defective color reduction for $\log^*\Delta$ iterations but have to be careful how to pick the defect in different iterations.