

Lecture 28: Bounds on Chromatic Number

December 2, 2021

Kennesaw State University

1 Coloring random graphs

We have two interesting lower bounds on the chromatic number $\chi(G)$ of a graph G :

1. $\chi(G) \geq \omega(G)$, where $\omega(G)$ is the clique number;
2. $\chi(G) \geq \frac{n}{\alpha(G)}$, where $\alpha(G)$ is the independence number, and n is the number of vertices.

It is actually quite hard to come up with examples where the first bound is far from the truth, and the second bound is more useful. This is not what typical small graphs with nice structure look like.

It is hard to do this... for humans. In reality, when the number of vertices is large, almost all graphs behave in precisely the opposite way: the first bound is far from the truth, and the second bound is much more useful.

How are we going to prove something like this? Well, to be concrete, let's consider graphs with $n = 1000000$ (a million) vertices. There are many such graphs. To pick one of them at random, we flip a coin for each pair of vertices to decide if there is an edge between them. There is a mindbogglingly large number of graphs like this; every time you go to the trouble of flipping all $\binom{1000000}{2}$ coins to get a random 1000000-vertex graph, it is overwhelmingly likely that you are looking at a graph no human has ever seen before. But in some ways, these graphs are very predictable. They all have about $\frac{1}{2} \binom{1000000}{2}$ edges, give or take a few million. And all of them have fairly small clique number and independence number.

Claim 1.1. *If G is a random graph chosen in this way, it is very unlikely that $\omega(G) \geq 40$.*

Proof. There are $\binom{1000000}{40} \approx 1.22 \times 10^{192}$ ways to choose 40 of the vertices of G . Each one of those 40-vertex sets *could* be a 40-vertex clique, if we're lucky.

It *could* be a 40-vertex clique, but that's very unlikely. There are $\binom{40}{2} = 780$ edges between those 40 vertices, so we flip 780 coins to decide which edges between the vertices are present. In order for us to get a clique, all the coinflips have to go one way, which has a probability of $2^{-780} \approx 1.57 \times 10^{-235}$.

As an upper bound on the probability of getting a 40-vertex clique, it's enough to just multiply these two numbers together. If there are about 1.22×10^{192} potential 40-vertex cliques, and each one of them has about a 1.57×10^{-235} chance of actually being a clique, then *even if those are all disjoint events*, the total probability of a 40-vertex clique is only $(1.22 \times 10^{192}) \cdot (1.57 \times 10^{-235})$ or about 1.93×10^{-43} .

¹This document comes from the Math 3322 course webpage: <http://facultyweb.kennesaw.edu/mlavrov/courses/3322-fall-2021.php>

That's a tiny probability: we're looking at an event that almost never happens. Our graph on 1000000 vertices almost never has any 40-vertex cliques. \square

So if we're using the lower bound $\chi(G) \geq \omega(G)$, it usually won't even be able to tell us that $\chi(G) \geq 40$.

Okay, but what if this is accurate? What if the chromatic number really is that small? To argue against that, we use the second bound.

Claim 1.2. *If G is a random graph chosen in this way, it is very unlikely that $\alpha(G) \geq 40$.*

Proof. The probability here is exactly the same as for cliques. For any given 40-vertex set, if we are flipping coins for each of the 780 edges, there is (again) a 2^{-780} probability that *none* of the edges are present.

So once again, there is at most a 1.93×10^{-43} probability of getting a 40-vertex independent set. \square

By using the second of our bounds of chromatic number, we know that $\chi(G)$ is almost always at least $\frac{1000000}{40} = 25000$. That's much larger than the bound we got from the clique number!

2 The Mycielski construction

Using a random graph feels unsatisfying: we can't actually point to a specific example of a graph G where $\chi(G)$ is much bigger than $\omega(G)$, we just know that randomly picking one will always work.

In fact, it is an open problem for how to come up with a deterministic rule for large graphs G with small $\omega(G)$ and $\alpha(G)$. Even though almost every large graph will work, any specific large graph we can construct seems to behave very differently!

But there is one construction that we can be very specific about, called the Mycielski construction, which graphs G in which $\chi(G)$ is as large as we want, but $\omega(G) = 2$: G does not even have any copies of K_3 . When $\omega(G) = 2$, we say that G is **triangle-free**.

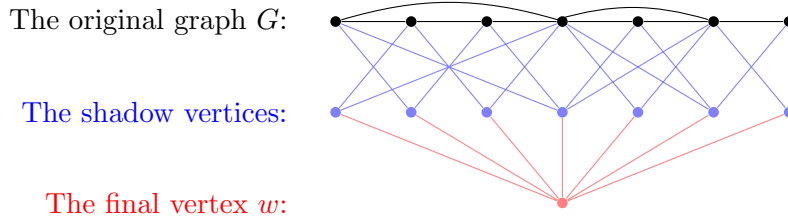
To find these graphs, we first define a new operation on graphs. Given a graph G , the **Mycielskian of G** is a graph $M(G)$ constructed as follows.

1. Start with a copy of G , with vertices named v_1, v_2, \dots, v_n .
2. For each vertex v_i , add a "shadow vertex" u_i adjacent to all of v_i 's neighbors in the copy of G .

(We never add edges between two different shadow vertices.)

3. Finally, add a vertex w adjacent to all the shadow vertices u_1, u_2, \dots, u_n .

On the next page is an example of this construction in action, with an arbitrary starting graph G . This is not the graph G we'll ultimately want to apply the construction to.



There are two important facts about the Mycielskian.

Claim 2.1. *If G is triangle-free, then so is $M(G)$.*

Proof. Where could we try to find a triangle (copy of K_3) in $M(G)$, when there are no triangles in G ?

If one vertex of the triangle were the final vertex w , then the other two vertices would both have to be shadow vertices u_i, u_j . This does not work, because the shadow vertices are not adjacent to each other.

Similarly, we cannot have a triangle with more than one shadow vertex in it. So the only hope for creating a triangle is to take one shadow vertex u_i , and two original vertices v_j, v_k . (You can see some triangles like this in the diagram above.)

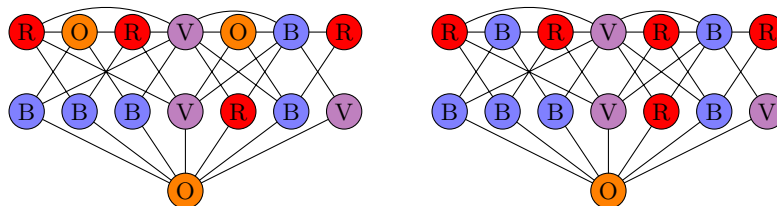
But if all three of these vertices are adjacent, then v_i is also adjacent to v_j and v_k , so v_i, v_j, v_k form a triangle as well! Therefore this cannot happen if G is triangle-free: $M(G)$ must also be triangle-free. \square

Claim 2.2. $\chi(M(G)) = \chi(G) + 1$: *the Mycielskian operation increases chromatic number by 1.*

Proof. One direction of this claim (which we don't particularly need in the end, but which we'll do anyway because it's easy) is to show that $\chi(M(G)) \leq \chi(G) + 1$: if we can color G with $\chi(G)$ colors, we can color $M(G)$ with $\chi(G) + 1$ colors.

To do this, just take a proper k -coloring of G and apply it to the copy of G inside $M(G)$. Then, give every shadow vertex u_i the same color as the color of v_i : this works fine, because it has all the same neighbors as v_i . Finally, give w a new color we did not use in G . This also cannot create any conflicts, so we get a proper $(k + 1)$ -coloring of $M(G)$.

The other direction is harder: given a proper k -coloring of $M(G)$, we must construct a proper $(k - 1)$ -coloring of G . The algorithm to do so is this: given a proper k -coloring of $M(G)$, if any vertex v_i has the same color as w , change it to have the same color as u_i (which must be different from w 's color, because u_i is adjacent to w). Here is an example:



You'll notice that when we do this, the new coloring is no longer proper: the new color of v_i might conflict with some of its shadow vertex neighbors. However:

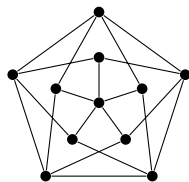
If v_i is adjacent to v_j and we recolor v_i , then v_i and v_j are different colors.

That's because u_i was also adjacent to v_j in the proper coloring we started with, so the color of u_i (which is the new color of v_i) is different from the color of v_j . Also, none of the vertices we recolored were adjacent (because they were all the same color as w), so we don't get any conflicts between two vertices that change colors.

Another way to say the bolded statement is that if we take the new coloring of $M(G)$ and only look at the copy of G inside it, we get a proper coloring of G . That proper coloring uses one fewer color, because the color of w no longer appears on any of v_1, v_2, \dots, v_n . This gives us the $(k - 1)$ -coloring of G we wanted. \square

As a consequence, if we start with an arbitrary triangle-free graph, and apply the Mycielski construction over and over and over, we get a sequence of triangle-free graphs with growing chromatic number. This is an other way to see that $\chi(G)$ might be much larger than $\omega(G)$.

Traditionally, we start with K_2 : two vertices with an edge. Then $M(K_2) = C_5$: the 5-cycle. $M(C_5)$ is the Grötzsch graph, shown below:

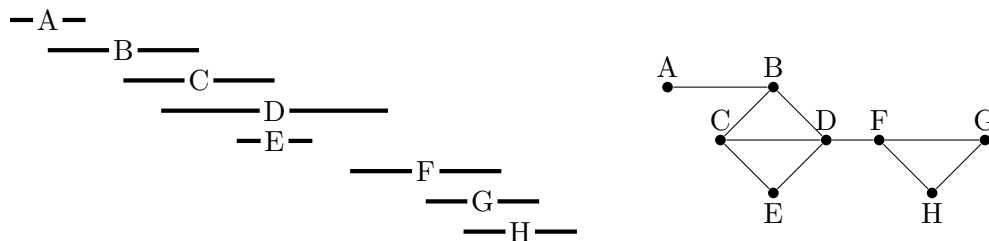


The Grötzsch graph is the smallest triangle-free graph with chromatic number 4.

3 Interval graphs

We've been trash-talking clique number the whole lecture, so let's do something to make it feel better.

An **interval graph** is a graph defined by a set of intervals $\{[a_1, b_1], [a_2, b_2], \dots, [a_n, b_n]\}$. For each interval $[a_i, b_i]$, there is a vertex v_i ; two vertices are adjacent whenever the intervals overlap. Here is an example of a set of intervals and their interval graph:



Many graph coloring problems are actually about coloring interval graphs. Two important applications are:

- Event scheduling. Here, we imagine that each interval $[a_i, b_i]$ represents an event starting at time a_i and ending at time b_i . We want to put the events in rooms, but we have a limited supply of rooms available, so we want to use as few as possible. However, if two events are happening at the same time, they must be in different rooms.
- Register allocation: an application to computer science. Here, each interval $[a_i, b_i]$ represents a variable in a computer program, which is initialized at step a_i in the code, and last used at step b_i . There are special parts of computer memory called *registers* which are particularly easy to access, but their number is limited. We want to see if the variables can be stored in registers, so that when variables are in use simultaneously, they are in different registers.

Here's the good news:

Theorem 3.1. *If G is an interval graph, then $\chi(G) = \omega(G)$.*

Proof. This will be a greedy coloring proof. We sort the intervals $[a_i, b_i]$ by their starting point, so that $a_1 < a_2 < \dots < a_n$. Then, we color the vertices v_1, v_2, \dots, v_n in this order. We want to show that this never uses more than $\omega(G)$ colors.

Suppose we are coloring vertex v_i . Which vertices among v_1, \dots, v_{i-1} are adjacent to this vertex: which of the intervals that come before $[a_i, b_i]$ overlap with it?

Well, they all start before a_i , so to overlap with $[a_i, b_i]$, they must end after a_i . This means that they all contain the point a_i itself. But if we're looking at a set of intervals that all contain a_i , then they form a clique, because they all overlap! So, together with v_i , we are looking at $\omega(G)$ or fewer vertices: v_i has at most $\omega(G) - 1$ neighbors that come before it.

Therefore the greedy algorithm will never use more than $\omega(G)$ colors. It will use at least that many because we know $\chi(G) \geq \omega(G)$ for all graphs. Therefore the greedy algorithm always perfectly covers interval graphs. \square