CCS Discrete II | Professor: Padraic Bartlett

Lecture 6: The Chromatic Number

Week 6 | *UCSB 2015*

In our discussion of bipartite graphs, we mentioned that one way to classify bipartite graphs was to think of them as graphs that are **2-colorable**: i.e. graphs in which we could color all of the vertices either red or blue, so that no edge would have two endpoints of the same color. As well, when we studied the four-color theorem, the general notion of "graph colorings" came up; the entirety of the four-color theorem was the claim that the chromatic number of any planar graph was at most 4.

So: graph colorings seem interesting. We should study them!

1 Basic Definitions

For completeness's sake, we remind the reader about how we defined "colorability:"

Definition. We say that a graph G is k**-colorable** if we can assign the colors[1] $\{1, \ldots k\}$ to the vertices in $V(G)$, in such a way that every vertex gets exactly one color and no edge in $E(G)$ has both of its endpoints colored the same color. We call such a coloring a **proper coloring**, though sometimes where it's clear what we mean we'll just call it a coloring.

Alternately, such graphs are sometimes called k-partite. For a fixed graph G, if k is the smallest number such that G admits a k-coloring, we say that the chromatic number of G is k, and write $\chi(G) = k$.

To illustrate how this definitions goes, we work a few examples:

1. K_n: The complete graph on n vertices has chromatic number n. To see that it is at least n, simply paint each of the vertices $\{v_1, \ldots v_n\}$ of $V(K_n)$ a different color (say, v_i is painted i;) then every edge trivially has two endpoints of different colors. To see that this is necessary, take any proper coloring of K_n, and look at any vertex v_i: because it's connected to every other vertex, it cannot be the same color as any other vertex (and therefore must have a different color than every other vertex, which forces n colors.)

2. Edgeless graphs: If a graph G has no edges, its chromatic number is 1; just color every vertex the same color. These are also the **only** graphs with chromatic number 1; any graph with an edge needs at least two colors to properly color it, as both endpoints of that edge cannot be the same color.

3. Bipartite graphs: By definition, every bipartite graph with at least one edge has chromatic number 2.

[1]By "color," we just mean a collection of distinct labels, like (say) natural numbers. Actual colors have the disadvantage of being finite in number, which is rather pesky.

4. The pentagon: The pentagon is an odd cycle, which we showed was not bipartite; so its chromatic number must be greater than 2. In fact, its chromatic number is 3: simply color its vertices R, G, R, G, B in order by walking around the perimeter of the pentagon. (In fact, this same idea can be used to show that any cycle of length $2k+1$ is 3-colorable: we know that these are not bipartite, and that they do admit 3-colorings via the $R, G, R, G \ldots R, G, B$-coloring described above.)

5. In our first graph theory lecture, we said that one way of phrasing the 4-color theorem was to say that all "map-graphs" could be colored with at most four colors. In the language we've described above, this is the claim that all "map-graphs" have chromatic number at most 4.

2 Properties and Examples

We developed this notion of k-chromatic graphs by generalizing the concept of bipartite graphs. A natural question to ask, then, is whether our earlier classifcation of bipartite graphs can be generalized to k-partite graphs. I.e.: we showed that a graph was bipartite if and only if it didn't contain any odd cycles. Is there a similar classification for all graphs with chromatic number (say) 3?

Surprisingly: no! While there certainly are tons of 3-chromatic graphs, there is no materially different classification of all of them beyond "there is a 3-coloring of this graph" that graph theorists have found. In fact, while graph theorists have been studying colorings pretty much since the 4-color theorem was postulated, there really is a lot that we don't know out there! (For example: consider the **unit-distance graph**, which has vertex set $\mathbb{R}^2$ and an edge between two points in the plane if and only if the distance between them is 1. On the HW, you proved that its chromatic number is between 4 and 7: to this day, these are the best known bounds.)

However, we can say a few things about how the chromatic number relates to some other properties of a graph. We state a few relevant definitions below, and then prove a few related propositions:

Definition. For a graph G and a subgraph H, we say that H is a **induced subgraph** of G if and only if whenever $u, v \in V(H)$ and $\{u, v\} \in E(G)$, we have that $\{u, v\} \in E(H)$. In other words, H is a subgraph made by picking out some vertices from within H, and then adding in **every** edge in G that connects those vertices.

Definition. For a graph G, we define the **clique number** of G, $\omega(G)$, to be the largest value of k for which K_k is an induced subgraph of G. As every nonempty graph contains (at the minimum) a K_1 as an induced subgraph, this is a well-defined quantity.

Proposition 1. *If G is a graph and H is any subgraph of G, $\chi(G) \geq \chi(H)$.*

Proof. This is remarkably trivial. If G admits a k-coloring, then simply take some proper k-coloring of G and use it to color H's vertices. Because H's edges are all in E, we know that none of these edges are monochromatic under this coloring; therefore, it is a proper k-coloring of H, and thus $\chi(H) \leq k = \chi(G)$. □

Proposition 2. *If G is a graph, $\chi(G) \geq \omega(G)$.*

Proof. Let H be an induced subgraph of G isomorphic to $K_{\omega(G)}$, which exists by definition. Then, by the above proposition, $\chi(G) \geq \chi(K_{\omega(G)}) = \omega(G)$. $\square$

This gives us a lower bound. The following definition and proposition from the HW give us an upper bound, as well:

Definition. For a graph G, let $\Delta(G)$ denote the maximum degree of any of G's vertices, and $\delta(G)$ denote the minimum degree of any of G's vertices.

Proposition 3. *For any graph G, $\chi(G) \leq \Delta(G) + 1$.*

Proof. The algorithm here is remarkably simple, but at the same time important enough that we give it a name: the **greedy algorithm**. We define it here:

- (Greedy algorithm.) As input: take in a graph G with vertex set $V(G) = \{v_1, \ldots v_n\}$, and a list of potential colors $\mathbb{N}$.
- At stage k: look at v_k, and color it the smallest color in $\mathbb{N}$ not yet used on any of v_k's neighbors.

By construction, this creates a proper coloring of G. As well, because each vertex has $\leq \Delta(G)$ neighbors, we'll always have at least one choice of a color that's less than $\Delta(G)+1$; therefore, this creates a proper coloring of G that uses $\leq \Delta(G) + 1$ colors! So $\chi(G) \leq \Delta(G) + 1$, as claimed. $\square$

To sum up: we've shown that for any graph G, we have

$$\omega(G) \leq \chi(G) \leq \Delta(G) + 1.$$

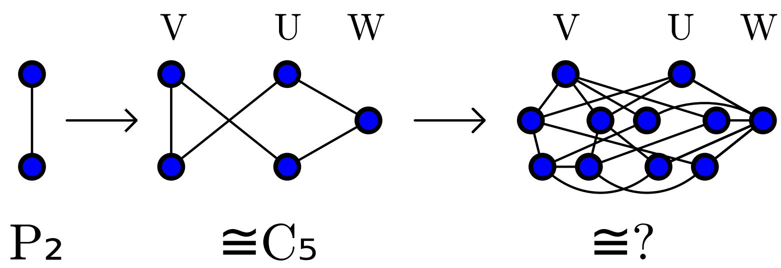
Which is something! However, as it turns out, it's not a *lot*. If you consider the **complete bipartite graph** $K_{n,n}$ formed by taking two groups of n vertices and connecting all vertices in one group to the other group, the degree of any vertex in this graph is n, while the chromatic number is 2; so there can be a massive gap between $\Delta(G)$ and $\chi(G)$. (On the HW, you will prove that this gap isn't just a accident, in the following sense; there are orderings $\{v_1, \ldots v_{2n}\}$ that you can place on the vertices of any $K_{n,n}$ that will make the greedy algorithm give such an "awful" n-coloring.)

As well, the gap between $\omega(G)$ and $\chi(G)$ can be quite large, as the following family of graphs shows:

Example. The **Mycielski** construction is a method for turning a triangle-free graph with chromatic number k into a larger triangle-free graph with chromatic number $k+1$. It works as follows:

- As input, take a triangle-free graph G with $\chi(G) = k$ and vertex set $\{v_1, \ldots v_n\}$.
- Form the graph G' as follows: let $V(G') = \{v_1, \ldots v_n\} \cup \{u_1, \ldots u_n\} \cup \{w\}$.
- Start with $E(G') = E(G)$.
- For every u_i, add edges from u_i to all of v_i's neighbors.
- Finally, attach an edge from w to every vertex $\{u_1, \ldots u_n\}$.

Starting from the triangle-free 2-chromatic graph K_2, here are two consecutive applications of the above process:



Proposition 4. *The above process does what it claims: i.e. given a triangle-free graph with chromatic number k, it returns a larger triangle-free graph with chromatic number $k+1$.*

Proof. Let G, G' be as described above. For convenience, let's refer to $\{v_1, \ldots v_n\}$ as V and $\{u_1, \ldots u_n\}$ as U. First, notice that there are no edges between any of the elements in U in G'; therefore, any triangle could not involve two elements from U. Because G was triangle-free, it also could not consist of three elements from V; finally, because w is not connected to any elements in V, no triangle can involve w. So, if a triangle exists, it must consist of two elements v_i, v_j in V and an element u_l in U; however, we know that u_l's only neighbors in V are the neighbors of v_l. Therefore, if (v_i, v_j, u_l) was a triangle, (v_i, v_j, v_l) would also be a triangle; but this would mean that G contained a triangle, which contradicts our choice of G.

Therefore, G' is triangle-free; it suffices to show that G' has chromatic number $k+1$.

To create a proper $k+1$-coloring of G': take a proper coloring $f : V(G) \to \{1, \ldots k\}$ and create a new coloring map $f' : V(G') \to \{1, \ldots k+1\}$ by setting

- $f'(v_i) = f(v_i)$,
- $f'(u_i) = f(v_i)$, and
- $f(w) = k+1$.

Because each u_i is connected to all of v_i's neighbors, none of which are colored $f(v_i)$, we know that no conflicts come up there; as well, because $f(w) = k+1$, no conflicts can arise there. So this is a proper coloring.

Now, take any k-coloring g of G': we seek to show that this coloring must be improper, which would prove that G' is $k+1$-chromatic. Doing this is on the homework for this week! □

As the example above illustrates, our bounds can (unfortunately) be rather loose: the Mycielskians, for example, have $\omega(M) = 2$ (because they don't even contain a triangle, K_3!), and yet have arbitrarily high chromatic number. Conversely, as noted before, the complete bipartite graphs $K_{n,n}$ all have chromatic number 2, and yet have $\Delta(G) = n$; so our upper bound of $\Delta(G) + 1$ can also be rather misleading!

We might hope that we can do better with some more machinery. As it turns out, we can!

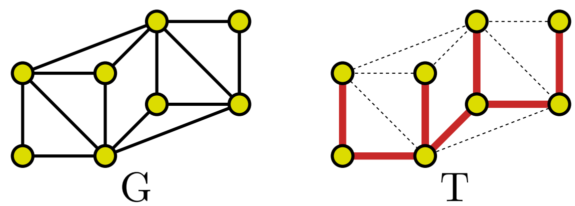
Theorem. (Brooks's theorem.) Suppose that G is a connected graph that is neither a complete graph nor an odd cycle. Then $\chi(G) \leq \Delta(G)$.

(We didn't say we could do a **lot** better.)

Proof. We need one key ingredient for this proof: last Friday's quiz! Recall the concept of a **spanning tree**, which we defined there:

Definition. Take any connected graph G. A **spanning tree** is a subgraph T of G with the following two properties:

- T is a tree.
- T contains every vertex of G.



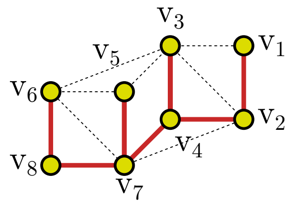
The quiz claimed that every connected graph on finitely many vertices contains a spanning tree. There are many ways to prove this; perhaps the simplest is to simply take our connected graph G and see if it has any cycles. If it does not, then it is a tree by definition; otherwise, it contains a cycle $C = (v_1, \{v_1, v_2\}, \ldots v_n, \{v_n, v_1\}, v_1)$. Pick any one edge in that cycle — say, $\{v_1, v_2\}$ — and delete it from our graph. Notice that this does not disconnect the graph, as any path that used that edge can simply go the "long way" around (i.e. replace $\{v_1, v_2\}$ in any walk with $(\{v_1, v_n\}, v_n, \{v_n, v_{n-1}\}, \ldots \{v_3, v_2\}, v_2)$.) Iterate this process, which must eventually halt as our graph can only have finitely many edges; the result is a connected graph on the same vertices with no cycles, i.e. a spanning tree.

We will use spanning trees to color graphs as follows: take any spanning tree T of a graph G. We can use this tree to make a notion of "distance" on our graph G as follows: for any two vertices $v_1, v_2 \in G$, we set $d_T(v_1, v_2)$ to be the length of the shortest path in T that connects v_1, v_2.

Take any vertex $v_n \in T$; call this the "root" vertex. Collect all of the vertices of G into groups sorted by their distance from v_n; say $A_0, A_1, A_2, \ldots A_l$, where the elements of A_i are precisely those vertices that are distance i from v_n under the d_T-metric, and l is the maximum such distance.

Use this grouping to create an "ordering" on the vertices of G as follows: let the elements of A_l be the "first" elements in our ordering, the elements of A_{l-1} be the "next" elements, and so on/so forth until we get to A_0, which contains v_n (the only element distance 0 from v_n), which is the "last" element in our ordering. In the event that there are multiple elements in an A_i, resolve ties between elements in any matter you like (it won't matter.)

For example, if we take our graph G and spanning tree T from our example on the previous page, and pick the bottom-left vertex to be the "root" vertex v_n (for $n = 8$, as we're on an eight-vertex graph), one ordering we could get is the following:



The reason we care about this is the following observation:

Observation. Suppose that G is a graph and v_n is a vertex with degree strictly less than $\Delta(G)$. Then running the greedy coloring algorithm to color G's vertices, using the "tree-ordering" rooted at v_n, colors G with at most $\Delta(G)$ colors.

Proof. Take any vertex v_i, for $i \neq n$. When we go to color v_i using the greedy algorithm, we will have never colored any of v_i's neighbors that are closer to v_n than v_i under the d_T-metric. In particular, this means that there will always be at least one neighbor of v_i that is not yet colored!

Consequently, there are at most $\Delta(G) - 1$ neighbors of v_i that are currently colored, and therefore we will never need more than $\Delta(G)$ colors to color v_i as well.

The only vertex that we could conceivably have a problem with, then, is v_n; but the degree of v_n is strictly less than $\Delta(G)$, so it too will never have more than $\Delta(G) - 1$ already-colored neighbors. So we're done! $\square$

Great! We've proven Brooks's theorem for all **non-regular** graphs[2] For k-regular graphs, we do not need to worry about the case where $k = 2$, as those graphs are precisely the cycle graphs C_n (and thus are either bipartite or odd cycles, and we assumed that we're not working with odd cycles in our theorem statement.) So we only need to consider k-regular graphs for k at least 3.

We can get another case for "free" by using induction. Suppose that our graph G has a **cut-vertex**[3] v_n. Then, if we were to delete v_n from our graph, we would separate our graph into k pieces $G_1, \ldots G_l$, all of which would be disconnected from each other.

We prove our claim (that any such cut-vertex graph is $\Delta(G)$-colorable) by induction on the number of vertices in G. If our graph has one vertex, we are trivially set. Inductively, assume that our claim holds on all cut-vertex graphs with no more than n vertices, and take any $n+1$-vertex graph with a cut-vertex. Let $G_1, \ldots G_l$ be the components that deleting v_n splits our graph up into. Add v_n back into each component to get graphs $G'_1, \ldots G'_l$; these are all graphs on at most n vertices, and therefore have $\Delta(G)$-colorings. Pick such a coloring for each graph; furthermore, because the names of the colors are arbitrary, pick colorings so that v_n gets the same color across all colorings.

But the only place where these colorings overlap is on the single vertex v_n; so the union of these colorings is again a proper coloring!

This leaves us with the following kinds of graphs G to consider: namely, those that

- are not complete graphs,

[2] A graph is called k-regular if the degree of every vertex is k, and regular in general if there is some k such that it is k-regular.

[3] A **cut-vertex** in a graph G is a vertex that if you delete that vertex, you will separate your graph into multiple pieces. This is analogous to the idea of a cut-edge.

- have no cut-vertices, and
- are k-regular for some $k \geq 3$.

We make the following claim:

Claim. For any such graph G, there must be some vertex v_n with neighbors v_1, v_2 so that the following holds:

- v_1 and v_2 are not connected.
- $G - \{v_1, v_2\}$ is connected.

If this holds, then we can use the greedy algorithm as follows: let v_1, v_2 be the first two vertices in our ordering, and order the remaining vertices using the d_T algorithm on a spanning tree in the graph $G - \{v_1, v_2\}$ from before. This process will insure that v_1, v_2 both get the same color; this is because we use the smallest color available when possible, and there is no edge $v_1 \leftrightarrow v_2$. It will insure that every vertex other than v_n gets a color from $\{1, \ldots \Delta(G)\}$; this is because they all have a neighbor that is currently uncolored when they are colored. Finally, this will insure that v_n also can have a color from $\{1, \ldots \Delta(G)\}$; this holds because we've made sure that two of its neighbors share the same color, meaning there must be at least one color left over. This finishes our proof; so we would be done if we can prove this claim!

We consider two cases.

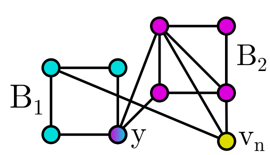
1. G is a "3-vertex-connected" graph[4] In this case, take any vertex $v_1 \in V(G)$. Because G is connected and is not the complete graph, there is some vertex v_2 that is exactly two edges away from v_1 (connected implies that there are paths to every vertex, and not complete implies at least one path is longer than one edge.)

   Let v_n be the vertex in between v_1, v_2 on that two-edge path. Notice that by assumption $v_1 \not\leftrightarrow v_2$ and by 3-vertex-connectivity deleting v_1, v_2 cannot disconnect our graph; so we're done!

2. Otherwise, our graph is 2-vertex-connected, with some cutset $\{v_n, x\}$ of size two.

   Here is where we do strange things[5].

   Define a **block** subgraph B to be any subgraph that is 2-vertex connected and maximal: that is, there is no larger subgraph that is two-connected and contains B.



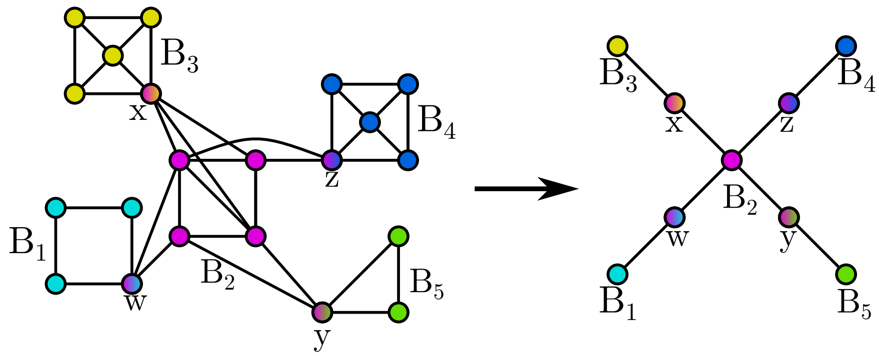
A 2-vertex connected graph G. Deleting v_n yields a 1-vertex-connected graph, which has two blocks B_1, B_2. Notice that two blocks can overlap; for instance, the vertex y is in both blocks.

[4]A connected graph G is called k-vertex-connected if there is no way to delete $k-1$ vertices from G that disconnects G. We define the **vertex connectivity** of a graph G as the largest value of k for which the graph is connected, and write this as $\kappa(G)$.

[5]Well, stranger things. We always do strange things.

Take any 1-connected graph G. We can turn this into a **tree** T_B as follows:

- Vertices: create one vertex for each block B_i of our graph, and one vertex of our graph for each cut-vertex x_j in our graph.
- Edges: connect a block B_i to a cut-vertex x_j if and only if $x_j \in B_i$.



A graph G along with its "block-tree" T_B.

Things that will be on Friday's homework:

- Any two blocks share at most one cut-vertex in common.
- If e is an edge linking two distinct blocks, then at least one endpoint of e is a cut-vertex.
- This is indeed a tree!

As a result, we know that in any block-graph there must be two "leaf-blocks:" that is, two blocks that are leaves in the block-tree. By the above, these leaf-blocks are connected to the entire rest of the graph by precisely one cut-vertex; deleting this cut-vertex separates this leaf-block from all of the other blocks.

What can we do with this? Well: let's return to our original problem setup, where we had a 2-connected graph G with some cutset $\{v_n, x\}$ of size two. Delete v_n; we get a graph $G - \{v_n\}$ that is 1-connected. Label its blocks $B_1, \ldots B_n$. Take any two distinct leaf-blocks B_1, B_2. We know that v_n must have a non-cut-vertex neighbor in each of B_1, B_2, as otherwise the single cut-vertex that connects each leaf to the rest of the graph would be a cut-vertex in G itself! Call its neighbor in B_1 v_1, and its neighbor in B_2 v_2.

Because our graph G is regular of degree at least 3, v_n has at least one other neighbor in our graph. Because of this, we can see that $G - \{v_1, v_2\}$ is connected:

- Deleting v_1, v_2 does not disconnect any individual block, as there is at most one of these vertices in any block. So we can travel within blocks.
- Each block is still connected to the other blocks by the cut-vertices, as we didn't delete any cut-vertices. So we can get from any block to any other block.
- Finally, we can also still get to v_n, as it is degree at least 3 (and thus has a third neighbor in some block.

Finally, we simply need to verify that there is no edge $v_1 \leftrightarrow v_2$. But this is not hard to see; if there were such an edge, then either v_1, v_2 would have to be a cut-vertex in $G - \{v_n\}$. But we picked v_1, v_2 to not be cut-vertices!

This completes our claim, and thus our proof.

$\square$

Yay! Improving bounds by 1! (The real lesson here: coloring is weird.)