Coloring

Lecture 5 – Graph Theory 2016 – EPFL – Frank de Zeeuw

1 Vertex coloring

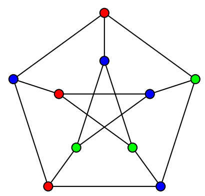
Definition. *A* vertex coloring *of a graph G is a map $c : V(G) \to \mathbb{N}$ such that $c(x) \neq c(y)$ whenever $xy \in E(G)$. The* chromatic number $\chi(G)$ *of G is the minimum image size of a vertex coloring of G; in other words, it is the minimum number of colors that $V(G)$ can be colored with.*

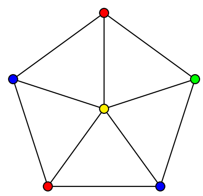
Given a vertex coloring, each color class (the set of vertices with the same color) is an independent set. So a vertex coloring is the same thing as a partition of the vertex set into independent sets.

A graph G has $\chi(G) = 1$ if and only if it has no edges. Having $\chi(G) = 2$ is the same as being bipartite (assuming the graph has at least one edge). Even cycles are bipartite and thus have chromatic number 2; odd cycles have chromatic number 3.

A complete graph K_n has chromatic number n. If G contains H, then $\chi(G) \geq \chi(H)$. Thus, if a graph G contains a K_k, then $\chi(G) \geq k$. But it is not true that if $\chi(G) \geq k$, then G must contain a K_k; we have $\chi(C_5) = 3$, but C_5 contains no K_3.

Below are two graphs with minimum colorings. The one on the left is the Petersen graph and has chromatic number 3 (it cannot be 2 since the graph contains a C_5). The one on the right is a *wheel* and has chromatic number 4 (it cannot be 3, since we need at least three colors on the outer C_5, which forces the middle vertex to have a fourth color).





Note that for both depicted graphs, the chromatic number is at most the maximum degree of a vertex. On the other hand, for complete graphs and odd cycles we have $\chi(G) = \Delta(G)+1$.

Lemma 1.1. *For any graph G we have $\chi(G) \leq \Delta(G) + 1$.*

Proof. We use the following greedy algorithm, which consecutively colors the vertices in an arbitrary order. For each vertex, we look at the neighbors that have already been colored, and we give the vertex the smallest color that is not used on those neighbors. Since any vertex has at most $\Delta(G)$ neighbors, we need at most $\Delta(G) + 1$ colors to color all the vertices. □

The greedy algorithm can be refined somewhat, by choosing the ordering in a certain way, so that it does better on some graphs. However, the problem of determining the chromatic number of a graph is NP-hard, so we do not know any fast algorithm that always works. We do have an algorithm for determining whether or not a graph has chromatic number two, since we have seen an algorithm for determining if a graph is bipartite.

As mentioned, complete graphs and odd cycles have $\chi(G) = \Delta(G) + 1$, so the inequality in Lemma 1.1 cannot be improved in general. In fact, complete graphs and odd cycles are the *only* connected graphs that attain equality, but we will not prove this until later. For now, we prove a weaker statement by choosing a specific ordering for the greedy algorithm.

Lemma 1.2. *If a connected graph G has $\chi(G) = \Delta(G) + 1$, then G is regular.*

Proof. We show that if G is not regular, then we can choose an ordering of the vertices so that the greedy algorithm from the proof of Lemma 1.1 needs only $\Delta(G)$ colors. Set $n = |V(G)|$. Assume G is not regular, so there is a vertex of degree at most $\Delta(G) - 1$, which we label x_n.

We grow a spanning tree from x_n: Starting with T consisting only of the vertex x_n, we repeatedly add to T an edge from $\partial(T)$. Since G is connected, this will indeed terminate with a spanning tree. While doing this, we label the vertices $x_{n-1}, x_{n-2}, \ldots, x_1$ in decreasing order as we add them. We claim that the resulting sequence $x_1, \ldots, x_n$ has the property that each vertex has at most $\Delta(G) - 1$ neighbors in the sequence before it. Indeed, any x_i with $i < n$ has a unique path to x_n in the tree, and by construction the labels on the vertices of this path increase from x_i to x_n. Thus x_i has at least one neighbor x_j with $j > i$. The claim also holds for x_n, because we chose x_n to have degree at most $\Delta(G) - 1$.

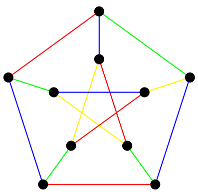
We apply the greedy algorithm from the proof of Lemma 1.1, with the vertices in the ordering $x_1, \ldots, x_n$. When we are about to color x_i, at most $\Delta(G) - 1$ of its neighbors are already colored, so $\Delta(G)$ colors suffice. $\square$

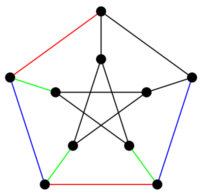
2 Edge coloring

Definition. *An* edge coloring *of a graph G is a map $c : E(G) \to \mathbb{N}$ such that $c(e) \neq c(e')$ whenever e, e' are distinct edges that share a vertex. The* edge-chromatic number *$\chi_e(G)$ of G is the minimum image size of an edge coloring of G; in other words, it is the minimum number of colors that $E(G)$ can be colored with.*

Each color class in an edge coloring is a matching, so an edge coloring is partition of the graph into matchings.

An even cycle has edge-chromatic number 2 and an odd cycle has edge-chromatic number 3. The picture on the left below is an edge coloring of the Petersen graph with four colors. We now prove that four is the minimum number (the picture on the right illustrates the proof). Suppose we could color it with three colors. The outside C_5 needs three colors, and there must be three consecutive vertices on that C_5 that are incident with only two colors (here red and blue). Then the third edge of each of those three vertices must have the third color (here green). But then this third color cannot be used on the inside C_5.





Lemma 2.1. *For any graph G with at least one edge we have $\Delta(G) \leq \chi_e(G) \leq 2\Delta(G) - 1$.*

Proof. There must a vertex of degree $\Delta(G)$, and an edge coloring must give a different color to each of the $\Delta(G)$ edges at that vertex. This implies $\chi_e(G) \geq \Delta(G)$.

The upper bound follows by a greedy algorithm just like that in the proof of Lemma 1.1. Take an arbitrary ordering of the edges, and consecutively color each edge with the smallest color that is not yet used on the colored edges that it shares a vertex with. Since an edge shares a vertex with at most $2(\Delta(G) - 1)$ edges, it follows that $2\Delta(G) - 1$ colors suffice. $\square$

Unlike for vertex coloring, the greedy algorithm for edge coloring can be significantly improved on. This is shown in the proof of the following theorem, which tells us that any graph G has edge-chromatic number either $\Delta(G)$ or $\Delta(G)+1$. Both are possible, since even cycles have $\chi_e(G) = \Delta(G)$ and odd cycles have $\chi_e(G) = \Delta(G)+1$. However, this algorithm still does not always give the exact number, and in fact it is NP-hard to determine which of the two values is the edge-chromatic number of a given graph.

Theorem 2.2 (Vizing). *For any graph G we have $\chi_e(G) \leq \Delta(G)+1$.*

Proof. We use induction on the number of edges; the statement clearly holds for a graph without edges. Given an edge $xy \in E(G)$, we describe an algorithm that, given an edge coloring of $G - xy$ with at most $\Delta(G)+1$ colors, produces an edge coloring of G with the same number of colors. To find a color for xy, the algorithm may have to change the colors of other edges, so it is not a greedy algorithm.

We first give some ad hoc definitions.

- If no edge incident with vertex v has color c, then we say that c is *free at* v, or that v is *c-free*.
- Given two colors c, d, a *cd-path* is a path in $G - xy$ whose edges are colored c and d. If a cd-path is maximal, then we can *invert* it by switching the colors c and d along the path. The result will still be an edge coloring, because if there were a conflict, then the path would not be maximal.
- A *fan* consists of a vertex x, a sequence $y_0, \ldots, y_k$ of distinct neighbors of x, and a sequence $c_1, \ldots, c_k$ of distinct colors, such that xy_0 is uncolored, and for $1 \leq i \leq k$, xy_i has color c_i and y_{i-1} is c_i-free. We can *rotate* a fan by recoloring edge xy_{i-1} with color c_i for $i = 1, \ldots, k$, and leaving xy_k uncolored; the result is still an edge coloring (except for xy_k).

Given an edge coloring of $G - xy$ with $\Delta(G)+1$ colors, we begin by constructing a fan based at x with $y_0 = y$ as follows. Since we have $\Delta(G)+1$ colors and y_0 has degree at most $\Delta(G)$, there is a color c_1 that is free at y_0. If there is an edge incident to x with color c_1, then we label its other endpoint y_1. We continue like this: We pick a color c_{i+1} that is free at y_i, and if possible we pick a c_{i+1}-colored edge xy_{i+1} that is not yet in the fan. This terminates when we have a vertex y_k that is c-free for some color c, but there is no new edge incident to x with color c.

Given this fan, there is a color c that is free at y_k, and there is a color d that is free at x. We take a maximal cd-alternating path containing x. Such a path either consists of x, or it starts at x with a c-edge, possibly followed by a d-edge, etc. We invert the path, which gives a new edge coloring.

After this inversion x is c-free. We claim that for some $1 \leq \ell \leq k$, y_ℓ is c-free, and $x, y_1, \ldots, y_\ell$ still forms a fan. If the alternating path was just x, and the inversion did nothing, then we can take the whole fan. Otherwise, the path started with some c-edge xy_{i+1}, and y_i must have been c-free before the inversion; in other words, $c_i = c$. If the path did not contain y_i, then $x, y_1, \ldots, y_i$ is still a fan. Indeed, the colors $c_1, \ldots, c_k$ are distinct, so for $j \leq i$ the inversion did not affect the fact that xy_j has color c_j or the fact that y_{j-1} is c_j-free. If the path did somehow reach y_i, then the inversion made it d-free, and it colored xy_{i+1} with d. In this case, $x, y_1, \ldots, y_k$ is still a fan, with the only change within the fan being that c_i is replaced with d.

In all cases we have a fan $x, y_1, \ldots, y_\ell$ in which x and y_ℓ are c-free. Now we can rotate the fan, so that the uncolored edge $xy_0 = xy$ becomes colored, and xy_ℓ becomes uncolored. Then we can color xy_ℓ with c. $\square$

3 Line graphs

Definition. *The* line graph *of a graph G is the graph $L(G)$ with vertex set $V(L(G)) = E(G)$, and with $e, e' \in V(L(G))$ adjacent if e and e' share a vertex in G.*

Line graphs explain some similarities that you may have noticed, between edge coloring and vertex coloring, and matchings and independent sets. From the definitions it should be clear that

$$\chi_e(G) = \chi(L(G)),$$

and similarly that

$$\mathrm{m}(G) = \alpha(L(G)).$$

The cycle of a line graph is the cycle itself, which explains why the chromatic number of a cycle equals its edge-chromatic number (if that needed any explanation).

We have $\Delta(L(G)) \leq 2(\Delta(G) - 1)$, since if two vertices of degree $\Delta(G)$ are adjacent, then their edge corresponds to a vertex of degree $2(\Delta(G) - 1)$ in the line graph. Thus the upper bound $\chi_e(G) \leq 2\Delta(G) - 1$ actually follows from the upper bound $\chi(G) \leq \Delta(G) + 1$ (and indeed, the proofs were essentially the same).

Here is another example where this connection is useful.

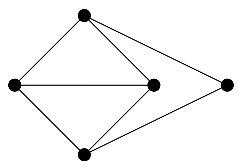
Lemma 3.1. *For any graph G we have $\chi(G) \geq \frac{|V(G)|}{\alpha(G)}$ and $\chi_e(G) \geq \frac{|E(G)|}{\mathrm{m}(G)}$.*

Proof. First we prove that $\chi(G)\alpha(G) \geq |V(G)|$. Given a coloring with $\chi(G)$ colors, the color classes (label them $S_1, \ldots, S_{\chi(G)}$) are independent sets, and thus have size at most $\alpha(G)$. Hence we have

$$|V(G)| = \sum_{i=1}^{\chi(G)} |S_i| \leq \sum_{i=1}^{\chi(G)} \alpha(G) = \chi(G)\alpha(G).$$

The second inequality follows directly by applying the first one to the line graph of G. □

Not every graph is a line graph. Suppose for instance that the following graph H is a line graph $L(G)$. The pair of vertices of H connected by the horizontal edge would correspond to a path xyz in G. The edges in G corresponding to the top and bottom vertex of H have to touch both edges of the path xyz, so one should be incident only with y and one should connect x and z. But then the edge of G corresponding to the rightmost vertex of H cannot touch the edge xz without touching xy or yz, which is a contradiction.



We can now look back at the algorithmic facts that we have seen. In general, there is no good algorithm for finding independent sets. But, for graphs that are line graphs, finding an independent set in $L(G)$ is equivalent to finding a matching in G, for which there is a good algorithm. Similarly, there is no good algorithm for determining the chromatic number of a graph, and the greedy algorithm is basically the best we have. But, for line graphs, determining the chromatic number of $L(G)$ is the same as determining the edge-chromatic number of G. For that there is a non-greedy algorithm which does a lot better, since it gives an edge coloring of G (and thus a vertex coloring of $L(G)$) that is either minimal or has one color too much.