Lecture 5: Sept 15, 2003

Lecturer: Yuval Rabani *Scribe: Alexa Sharp*

1 Chromatic Index

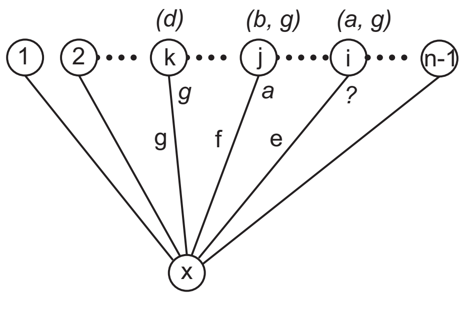
Theorem 1.1 (Shannon's Theorem, 1949 [1]) $\chi'(G) \leq \lfloor \frac{3}{2}\Delta \rfloor$. *Furthermore, there is a polynomial time algorithm to colour G with this many colours.*

Proof: If Δ is even then all odd degree nodes have degree strictly less than Δ. We add a matching of the odd degree nodes to G so that now all nodes have even degree at most Δ. We can now decompose the graph into $\frac{\Delta}{2}$ 2-factors, and each 2-factor can be coloured with ≤ 3 new colours. This uses at most $3 \cdot \frac{\Delta}{2} = \lfloor \frac{3}{2}\Delta \rfloor$ colours.

If Δ is odd then we prove the result by induction on $|V|$. The base case of no nodes is trivial. For the inductive step, suppose we add a new node x to a graph with $n-1$ nodes. This node is possibly adjacent to each of the previously added nodes, possibly with multiple edges. Let p_i be the multiplicity of the edge from x to vertex i, for $i = 1, 2, \ldots, n-1$.

Before we added the new node, each vertex i had $deg(i) \leq \Delta - p_i$. By the induction hypothesis we had coloured the graph (without x) with at most $\frac{3}{2}\Delta$ colours. Therefore vertex i must have at least $\frac{3}{2}\Delta - (\Delta - p_i) = \frac{\Delta}{2} + p_i$ colours still available to it at the point in the algorithm when we added vertex x.

If as we colour the edges adjacent to x, we come to some point where there is no available colour to use on an edge $e = (x, i)$ then it must be the case that we have coloured at least $\frac{\Delta}{2} + p_i$ edges, thus using all the colours available to i by edges between x and other vertices. At most $p_i - 1$ of these colours were used by the other edges touching i. So there are at least $\frac{\Delta}{2} + p_i - (p_i - 1) = \frac{\Delta}{2} + 1$ colours used by edges between x and other nodes that are still available at i. Consider an edge $f = (x, j), j < i$ that uses one of these colours, α say.



If j has an available colour β not used at x then we can switch f's colour from α to β, making α available at x. α was also available at i by assumption, making it possible for us to colour e with colour α and proceed with the algorithm.

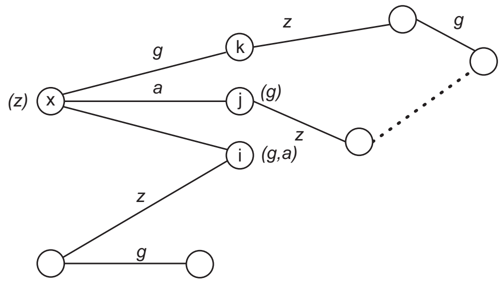
If no such colour exists then for every such vertex $j < i$, all of j's available colours are used at x. Consider one of these vertices $j < i$. We know the degree of j is at most Δ, and so at most Δ colours can be used at j. By the induction hypothesis, at most $\frac{3}{2}\Delta$ colours were used in total, leaving j with at least $\frac{\Delta}{2}$ available colours, each of which is used at x (and thus not available at x). Recall that at least $\frac{\Delta}{2}+1$ colours were available at i but used by x. Furthermore, x's degree is at most Δ. By the Pigeon Hole principle, there must be at least one colour used by x that is available at both vertex i and vertex j. Suppose this colour, γ say, is used on the edge $g = (x, k)$, $k < i$. There are two cases:

(i) If k has an available colour δ not yet used by x, change the colour of the edge g from γ to δ, which will make γ available at x. Now γ is available both at x and at i, so we can colour the edge $e = (x, i)$ with γ and continue the algorithm.

(ii) Otherwise, all of k's available colours have been used by x. Similarly, recall that all of i and j's available colours have been used by x. Viewed in the contrapositive this says that all of x's available colours are used at i, j, and k. Pick one of these colours ζ used at all of i, j and k but not at x.

Consider the γ-ζ chain starting from the edge $g = (x, k)$. There are 3 possibilities for the end of the chain:

(1) The chain ends at j. Then consider the alternating chain out of i, starting on the ζ edge. We know the chain cannot end at j or k (or else some node would either two adjacent ζ edges, or two adjacent γ edges), so we can swap the colours γ and ζ along the chain and colour the edge $e = (x, i)$ with ζ since it is now available both at x and at i.



(2) The chain ends at i. Then consider the alternating chain out of j, starting on the ζ edge. With a symmetric argument to case (1), we can swap the γ and ζ colours along the chain, and colour $f = (x, j)$ with ζ. Now α is available at both x and i, so we can colour e with α.

(3) The chain neither ends at i nor at j. Then we can swap γ and ζ along this chain to make γ available at x. It was already available at i, which allows us to colour (x, i) with γ.

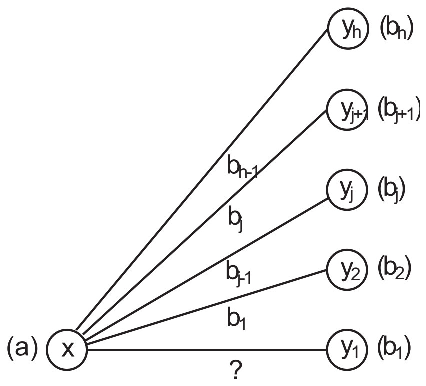
$\square$

Theorem 1.2 (Vizing [2]) *If G is simple then $\chi'(G) \le \Delta + 1$.*

Proof: (By induction on $|E|$). The base case of no edges is trivial. For the inductive step, consider some edge (x, y_1), and the graph $G - \{x, y_1\}$. By the induction hypothesis, $G - \{(x, y_1)\}$ can be coloured with $\Delta + 1$ colours. Since $deg(x), deg(y_1) \le \Delta$, they are each missing at least one of the used colours. Let α be a colour missing at x and β_1 a colour missing at y_1. If β_1 is missing at x or α is missing at y then we colour (x, y_1) with the missing colour and we're done.

Otherwise, there is an edge adjacent to x, (x, y_2) say, coloured β_1. Since $deg(y_2) \le \Delta$, there is a colour β_2 missing at y_2. If β_2 is also missing at x then we can colour (x, y_2) with β_2 and (x, y_1) with β_1 and we're done.

So let us assume there is a β_2-coloured edge (x, y_3) adjacent to x, and this vertex y_3 must be missing some colour β_3. We continue in this fashion, with edge (x, y_j) coloured β_{j-1} and vertex y_j missing colour β_j, until the process terminates at some y_h.



There are two reasons for termination:

(R1) β_h is not used at x. In this case we can do a *fan-recolouring*: shift the colours downward so that edge (x, y_j) is coloured with β_j for all $j = 1, \ldots, h$.

(R2) There is a previous edge (x, y_j) with $\beta_{j-1} = \beta_h$. Then we fan recolour upto edge (x, y_j), uncolouring (x, y_j) in the process. Now we consider the α-β_h chains. Notice that each of x, y_h and y_j are missing at least one of α or β_h, and are thus candidate enpoints of these chains. It is not possible for a chain to have 3 endpoints, and so it is not possible for all three vertices to be on the same chain.

If x's chain (starting on the $\beta_{j-1} = \beta_h$-coloured edge (x, y_{j-1})) does not end at y_j then we can swap the colours along the chain and colour (x, y_j) with β_h. If x's chain does not end at y_h then we can continue the fan recolouring until y_h and leave (x, y_h) uncoloured. Switch colours along the chain and colour (x, y_h) with β_h.

□

Remarks: For general graphs, the bound $\Delta + 1$ is not true in general. To see this, consider a triangle with multiple edges in between each pair of vertices. On the bright side, Vizing's proof can be modified to get the bound $\Delta + m$, where m is the maximum multiplicity. In fact, you can get the bound $\max_x\{deg(x) + m_x\}$, where m_x is the multiplicity of the vertex x. For general graphs, the best known bound is $1.1OPT + 0.8$.

Conjecture 1.3 (Seymour [3]) *Define* $\chi^*(G) = \max_{S,|S| odd}\left\{\frac{|E(S)|}{\lfloor\frac{|S|}{2}\rfloor}\right\}$. *It is easy to see that* $\chi^*(G) \leq \chi'(G)$. *The conjecture states that* $\chi'(G) \leq \chi^*(G) + 1$.

2 Randomized Rounding

MAX-SAT In this problem we are given a formula in conjunctive normal form (the conjunction of m clauses, where each clause is the disjunction of its variables). Our goal is to find a boolean assignment that satisfies as many clauses as possible. In the weighted version of this problem we would be given a non-negative weight for each clause, and our goal would be to find a boolean assignment that maximizes the weights of the clauses we satisfy.

Algorithm 2.1 (Johnson's [4]) *Choose the value of each variable independently and uniformly at random.*

Claim 2.2 *The expected number of clauses satisfied by Johnson's algorithm is at least* $\frac{m}{2}$.

Proof: If C is a clause with k variables, then

$$Pr[C \text{ is not satisfied}] = Pr[\text{all of } C\text{'s variables are false}] = \frac{1}{2}^k \leq \frac{1}{2}$$

Thus if we let $X_i = \begin{cases} 1 & \text{if clause } C_i \text{ satisfied} \\ 0 & \text{otherwise} \end{cases}$

$$\begin{aligned} E[\text{number of clauses satisfied}] &= E[\sum_{i=1}^{m} X_i] = \sum_{i=1}^{m} E[X_i] \\ &= \sum_{i=1}^{m} Pr[\text{clause } C_i \text{ satisfied}] \geq \frac{m}{2} \end{aligned}$$

□

Johnson's algorithm seems wasteful because it performs badly when the clauses are small. This motivates us to consider an algorithm that performs well in this case. The idea is to create an LP whose solution will dictate the probabilities to use to set each variables boolean value.

First, we create variables $y_i \in \{0,1\}$, $i = 1,\ldots,n$ so that $y_i = 1 \iff x_j = TRUE$, and variables $z_j \in \{0,1\}$, $j = 1,\ldots,m$, so that $z_j = 1 \iff C_j = TRUE$. The integer program representing MAX-SAT is therefore

$$\begin{array}{lll} \max & \sum_{j=1}^{m} z_j & \\ \text{s.t.} & \sum_{i|x_i \in C_j} y_i + \sum_{i|\overline{x_i} \in C_j} (1-y_i) \geq z_j & \forall j = 1,\ldots,m \\ & y_i \in \{0,1\} & \forall i = 1,\ldots,n \\ & z_j \in \{0,1\} & \forall j = 1,\ldots,m \end{array}$$

We will relax the integrality contraints on y_i and z_j to $0 \leq y_i \leq 1$ and $0 \leq z_j \leq 1$. The solution to this LP will be an upper bound on the solution for the IP. Now think of the variables y_i as probabilities, and set x_i to TRUE independently with probability y_i. With these assignments, we would like to calculate the probability that a clause $C_j = (x_1 \vee x_2 \vee \cdots \vee x_k)$ is satisfied.

$$Pr[\text{clause } C_j \text{ not satisfied}] = Pr[\text{all } k \text{ variables are false}] = \prod_{i=1}^{k}(1-y_i)$$

From the LP constraints we know the solution satisfies $y_1 + y_2 + \cdots + y_k \geq z_j$, and so the probability above is maximized when the y_i's are as small as possible and as similar as possible. In particular, the value is maximized when $y_i = \frac{z_j}{k}$. Thus

$$\begin{aligned} Pr[\text{clause } C_j \text{ not satisfied}] &\leq (1-\frac{z_j}{k})^k \\ Pr[\text{clause } C_j \text{ satisfied}] &\geq 1-(1-\frac{z_j}{k})^k = 1-(1-k\frac{1}{k}z_j + \binom{k}{2}\frac{1}{k^2}z_j^2 - \cdots) \\ &= z_j - \binom{k}{2}\frac{1}{k^2}z_j^2 + \cdots = z_j(1-\binom{k}{2}\frac{1}{k^2}z_j + \cdots) \\ &\overset{(*)}{\geq} z_j(1-(1-\frac{1}{k})^k) \geq z_j(1-e^{-1}) \end{aligned}$$

where inequality $(*)$ follows from $0 \leq z_j \leq 1$. The remainder of the proof appears in lec 6.

References

[1] C.E. Shannon. A theorem on colouring lines of a network. *J. Math. Phys.*, 28:148-151, 1949.

[2] V.G. Vizing. On an estimate of the chromatic class of a p-graph. *Discret. Analiz.*, 3, pp. 25–30, 1964.

[3] P. Seymour. On multi-colourings of graphs and conjectures of Fulkerson and Tutte. *Proc. London Math. Society Ser. (3)*, 38:423-460, 1979.

[4] D.S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.