Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 04

Lecture 16

Lecture 16 : Edge Coloring

Welcome. So, in the last class we have seen a $\frac{3}{2}$ factor approximation algorithm for metric travelling salesman problem. So, today we will see another application of greedy algorithms and local search heuristic both together for designing approximation algorithm of another problem which is called edge coloring of a graph. So, today's problem is edge coloring of a graph. So, what is the problem? Input is an undirected graph G goal. the goal is to colour the edges of G with the minimum number of colours such that no two edges of the same colour shear and end point ok.

For example, suppose consider this graph say A, B, C, D, E so on. Suppose this is the graph I want to colour the edges in such a way that no two edges of same colour share any end point. For example, this edge between A B I colour it within with one colour C 1, this edge B D I colour it using another colour C 2, this with the AC with may be C 2 again a C D with C 1 again and a E D is with say this E D is with say C 2 and this edge C E may be another colour of C 4. So, this is the valid edge colouring ok.

Number of colours used is 4. And we can say that for this graph this is optimal because there exist a vertex of degree 4 in this graph just to colour the edges incident on that degree 4 vertex I need 4 colours. So, what we have observed is this observation. any graph with maximum degree delta requires at least delta colours to colour all its edges properly. we call a colouring proper a colouring of the edges if no two edge of same end point no two edge of same colour shares any end point.

Now, this naturally leads to a computational question that if I am given a graph G is it always possible or is it possible to colour the edges of that graph with Δ colours where delta is a max degree, but it turns out that this problem is NP complete even when delta equal to 3. So, throughout this lecture capital delta will denote the maximum degree of the graph. Checking if a graph can be properly edge coloured with delta colours is NP complete. even when delta equal to 3. So, it is an NP complete problem even for the graphs where max degree can be 3.

Next we design a approximation algorithm the best possible approximation algorithm that one can hope for and this is the theorem. there is a polynomial time algorithm to properly color the edges of a graph with delta plus 1 colors. because the number of colors used is an integer number this is the best one can hope for. So, this sort of approximation algorithms is said to have additive approximation algorithm that is we have a polynomial time additive one approximation algorithm for properly colour the edges of any graph proof. and the algorithm is a combination of greedy and local search.

So, we maintain a palette of colours set of colours $c_1, c_2, \ldots, c_\Delta, c_{\Delta+1}$ ok. And we using these colours only we will colour all the edges of the graph. So, it is an iterative algorithm initially all the edges of the graph are uncoloured in each iteration. we pick an edge let us call it $\{u, v\}$ and colour it ok.

E_i be the set of edges that are colored. or at the start of ith iteration iteration . So, E_1 initially nothing is colored E_1 is empty set and in each iteration we pick an edge and color it. So, E_1 is a subset of E_2 which is a subset of E_3 and after n iterations all edges are colored. So, E_{n+1} is E the set of edges.

So, what is a natural greedy method? The natural greedy method is suppose I pick an edge $\{u, v\}$ we say colour c a we say a vertex vertex a v lacks a colour c if no edge incident on v has colour c. So, in the beginning every vertex lacks every colour. Now, in the ith iteration I have picked an uncoloured edge uv and want to colour it. So, if there exist a colour c which both u and v lacks then a natural greedy choice is to colour that this edge $\{u, v\}$ with c. If there is a colour c.

So, this set of colours let us denote it by say C. So, if there is a colour c in C that both u and v lack, then we colour the edge $\{u, v\}$ with c. So, in the remaining cases let us show we assume that and we assume that there is no colour which both u and v lacks. So, if both if there is a colour c which both u and v lacks then we then we colour that edge $\{u, v\}$ with that colour c and we are done with this iteration we move to the next iteration.

otherwise we do the following otherwise let a let us call v as v_0 ok. So, let c_0 be a colour that v_0 lacks ok. So, I have u here and this edge I need to colour $\{u, v\}$ which I am calling it v_0 and there is a colour say c_0 which the vertex v lacks why because the maximum degree of the vertex v is Δ and I have a set of $\Delta+1$ colours. So, every vertex lacks at least 1 colour. Next what I do, but this colour c_0 this vertex u does not like.

So, vertex u has an incident as an edge incident on u with colours which is coloured c_1. So, let us call that edge suppose that edges u v_1 which is coloured as c_0. Let u v_1 is already colored c_0. Then again we ask ah. if v_1 if there is a colour which v_1 lacks.

So, suppose c_1 is a colour that v_1 lacks or if there is a colour first the easy case there is a colour. say c' which both u and v_1 lack, then we we colour u v c_0 and u v_1 c_1 and we are done. Otherwise because yeah otherwise if or otherwise we are in the case that all colours that v_1 lacks, u does not lack you have a edge incident on it of that colour. So, otherwise let c_1 be a colour that v_1 lacks let, but u has an age incident on it with colour c_1 let u v_2 is coloured c_2.

is coloured c_1. So, the picture looks like here is u here is then edge that we intend to colour in this iteration $v = v_0$ I have v_1 coloured c_1, v_2 coloured c and this process continues and this process continues till one of the following cases happen. Suppose, here v_i c_i. The first case is so, let us write it v_j maybe or v_i is fine. The first case is there is a colour which colour say c double prime which both u and v_i lax. Then we use shifting colour recolouring technique shifting recoloring technique.

What is the technique? The technique is the generalization of this one. So, we colour this edge. So, you v_0 is coloured c_1 these all these colours are shifted u v_1 is coloured c_2 to U v_i is colored c_{i-1} and sorry this is colored v_{i-1} is colored c_i and u v_i is uncolored So, we shift up shift all these colours to there and now this edge u v_i is uncoloured, but they have a common colour which both of them lacks which is c''. So, then we colour u v_i c''. So, this is the easy case here you see it is a combination of greedy and local search.

So, we are doing greedy, but when we are stuck we are using local move to proceed. The other case is that when I am so, this sequence is called is called fan sequence. So, when I am building this fan sequence there is another way to not able to proceed and this is as follows u v_0 , u v_1 this is coloured u v_2 this is coloured c_2 u v_i this is coloured c_i, but the colour that v_i lacks is already one of the colours between c_1 to c_i. So, other case is the colour that v_i lacks is 1 of $c_1, c_2, \ldots, c_i$ ok. Suppose this v_i lacks the colour c_j and this is v_j this is c_j ok.

So, v_i lacks c_j ok. So, what we do here in this case again we use shifting recoloring from a v_0 to v_j we perform shifting recoloring fromv_0 to v_j . So, then that means, how does the graph look like here is u v_0 this is colored c_1. u v_1 this is coloured c_2 after shifting u v_2 this is coloured c_3 and so on and v_j is now uncoloured and rest is same here is v_i. ok. So, a so, v_j lacks c_j ok and v u also lacks one colour suppose that colour is c.

So, suppose u lacks colour c. Now, consider the subgraph induced by the edges colored C and this is c_i and c_j. Now, because it is a proper edge colour this will be a collection of cycles and path. This subgraph should be a collection of cycles and paths.

Now, all the three vertices u, v_i and v_j lacks one of either c or c_j, the vertices or every vertex in u, v_i, v_j sees exactly 1 of c and c_j. So, hence in this graph this u these vertices u, v_i and v_j can only be end points of the path. Hence the vertices u, v_i, v_j can only be end points of the paths ok. Now, the easy case is this lacks c_j and here also v_j you see after releasing this colour v_j also lacks c_j because u v_j was coloured c_j now that colour is move to u v_{j-1}.

So, v_j also lacks c_j. So, now, only two of them u both v_i and v_j cannot be connected with u. So, only one of them at most be connected. So, suppose case 1 suppose v_j is not connected is not connected with u in the subgraph. Suppose, then suppose u and v_i may be connected. So, then we exchange the colours of the path containing u ok.

So, u and v_i may be connected. So, there could be a path which may goes through this, this path restricted to these colours c and c_j. and in this path suppose in this path only I exchange this colour in any properly edge coloured graph if I exchange 2 colours it remains properly edge coloured you can it is easy to see, but you can also try to prove and convince yourself. So, once I exchange c_j now becomes available at u. Now, u lacks c_j also. Now, once u lacks c_j, v_j already lacks c_j.

So, I can now colour the colour this edge u to v_j as c_j, then we colour this edge u v_j. as c_j ok and we proceed. In the other case when u and v_j is connected by a path in that subgraph we again we can we can we exchange the colours and we again use the shifting technique to make u and v_i uncoloured and then we can colour this edge u v_i with c_j and then again we can proceed. So, you see in all the cases we are able to proceed and these all these steps can be done in polynomial time. Hence, it is a polynomial time algorithm and we never need more than delta plus 1 colours.

Hence, there exist a delta plus 1 edge colouring of the graph which concludes the theorem ok. So, let us stop here. Thank you.