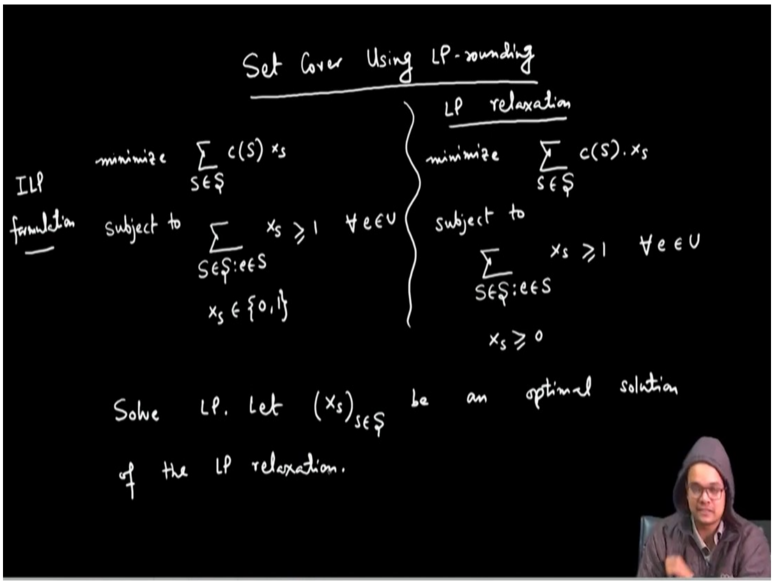
Selected Topics in Algorithms
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module No # 11
Lecture No # 52
Set Cover Using LP Rounding

Welcome so in the last class we have finished dual fitting and we have seen a high level idea of using linear programming rounding. So in today's class we will see linear programming rounding using some examples.

(Refer Slide Time: 00:47)



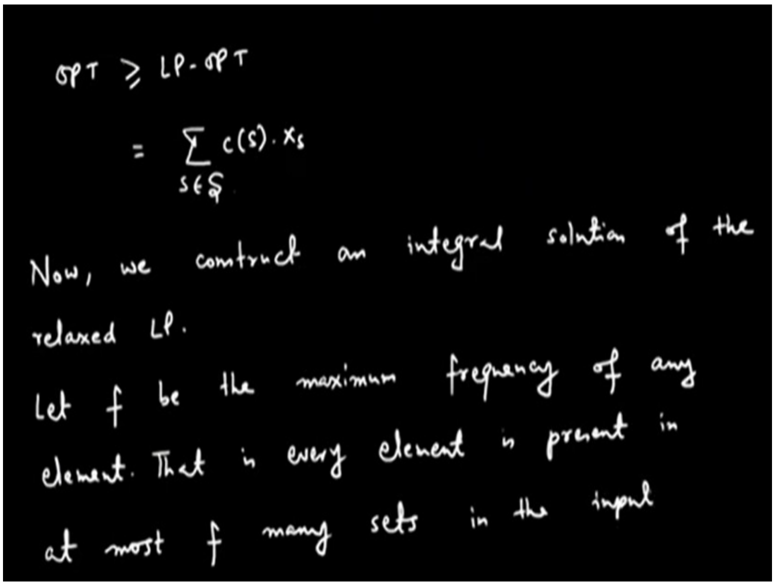
So our first example is set cover using LP rounding so let us recall what was the linear programming formulation the ILP was to minimize $\sum_{s\in S} c(s) x_s$ recall x is a takes value 1 if the algorithm picks it otherwise it is 0. And the requirement the constraint is subject to for each element I go over the $\sum_{s:e\in s} x_s \geq 1$. This is for all element $e \in U$ and we have x is greater x is in between 0 and 1.

So this is the ILP formulation next we relax it LP relaxation minimize summation cost of s times x, of x is in script is subject to summation is in cryptase e in s is greater than equal to 1 this is for all e in u. And I want to write I relax this integrality constraint of x is and allow it to take values greater than equal to 0 less than equal to 1. But because we are minimizing it and the requirement is to pick one set to cover each element. So we can safely delete this constraint recall that, this constraint was or cannot be deleted for set multi cover.

If we because it may be beneficial for cost minimizing cost to pick y same set multiple times so because our problem requires has a constraint that each set can be picked at most once. So their constraint needs to be kept explicitly because they are for set multi cover it may be beneficial for pick a set more than once because of the coverage requirements here it does not make sense to pick any set more than once.

So then what we do is that we solve LP let x_s in stress be an optimal solution to the optimal solution of the LP relaxation step is to use this solution to cover or to construct an integral solution so because this is a solution.

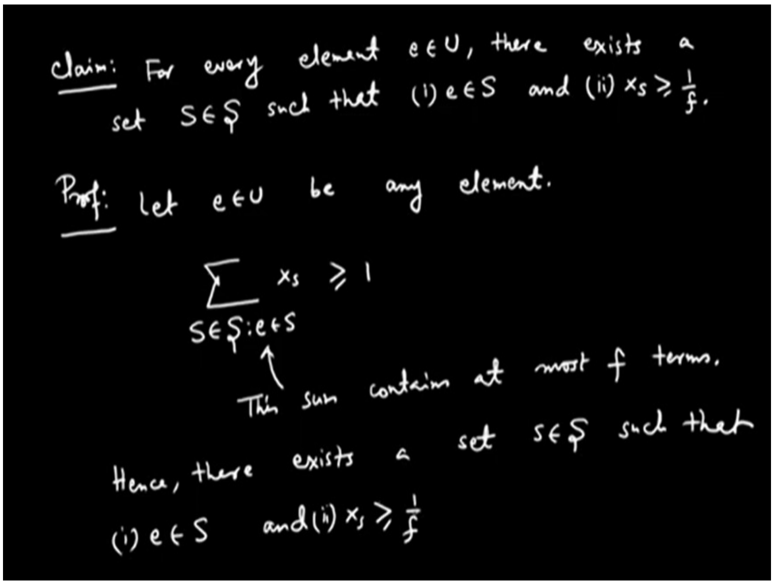
(Refer Slide Time: 06:20)



So we have opt is of course greater than equal to LP opt, and LP opt is summation for this solution summation $c(s)$ times x is in space. Now we construct an integral solution of the relaxed LP and this is the most non-trivial step of this entire process the steps till now is sort of mechanical it is always there. So you start with a solution which is a fractional solution of relaxed LP and how you can use that to construct, an integral solution for the linear program.

So for that there are various methods one is often you need to you need to understand the structure because of the problem is there any special property of that this solution satisfies this fraction solution satisfies and this is what we will see now. So for that let f be the frequency be the maximum frequency of any element what does that mean? That is every element is present in at most if many sets in the input.

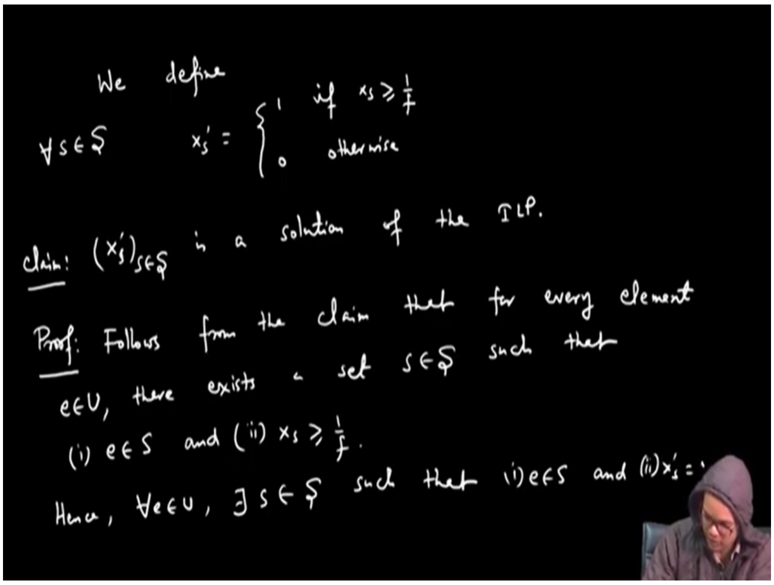
(Refer Slide Time: 09:51)



So with this we claim you look at the constraints for every element $e \in U$ there exists set is in script s such that one the element e belongs to s and 2 x of s is greater than equal to $\frac{1}{f}$ proof. So let $e \in U$ be any element so for u we have a constraint what is that constraint? That $\sum_{s: e \in s} x_s \geq 1$. Now this sum contains at most f terms and they sum up to at least once hence there exists set is in script s.

Such that one this element e belongs to s and x_s is greater than equal to $\frac{1}{f}$ now with this claim at hand we will round it.

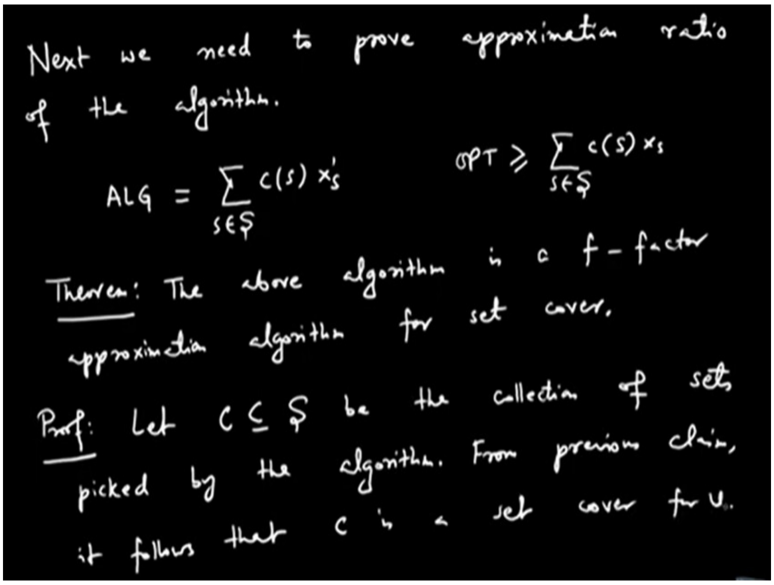
(Refer Slide Time: 13:29)



So what is the rounded, solution? x we define $x_s^{'}$ to be 1 if x is greater than equal to $\frac{1}{f}$ and 0 otherwise this we define for all set is in script s. So observe that $x_s^{'}$ in script s a solution of the ILP y that is because so let us see so again we can to claim so this is a solution to the ILP why? Because this is follows from previous claim this is follows from the fact from the claim that for every element e in u their; exists are set $s \in S$.

Such that x this element e belongs to the set and x_s is greater than equal to $\frac{1}{f}$ hence for every element e in u there exist a set $s \in S$ such that $e \in s$ and $x^{'} = 1$ and this is the exactly the requirement in ILP.

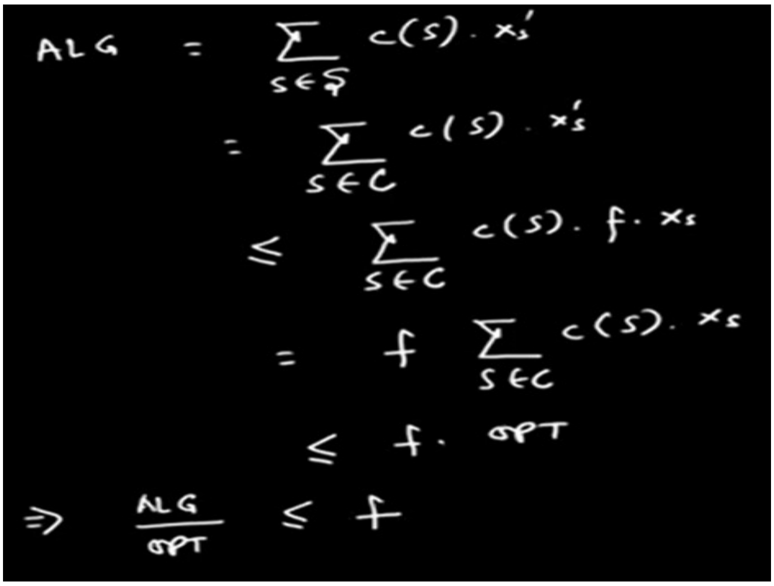
(Refer Slide Time: 16:41)



So now next what we need to do? Next we need to prove, approximation ratio approximation ratio of the algorithm so what is ALG? This is the value of the objective function here at solution x prime s this is summation c of is x prime s is in s. Now observe that so here is the thing theorem we need to compare this with compare this with of is greater than equal to $\sum_{s \in S} c(s) x_s$.

So claim the above algorithm is an f factor, approximation algorithm for set cover proof. So let see subset of s be the collection of sets picked by the algorithm if consider any arbitrary element e so first from our algorithm before from previous claim it follows that c is a set cover for you this you have argued here that is a solution to ILP because it is a solution to ILP. Now the rounding process so now we need to compare, ELG with this so what is ALG?

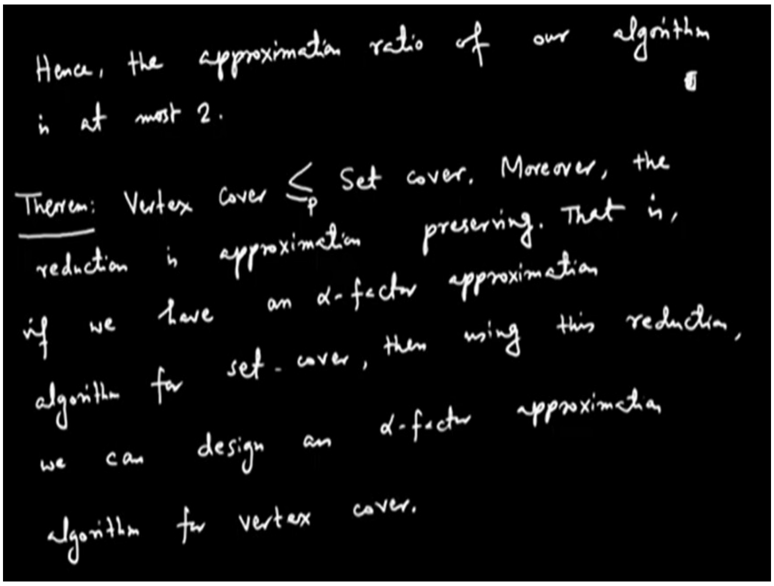
(Refer Slide Time: 21:03)



Let us see ALG is $\sum_{s \in S} c(s) x_s^{'}$ now all the sets except c except this collection c those for the variables $x_s^{'}$ is 0 for those elements. So this is sum over this collection this is c of case times $x_s^{'}$ is because other elements further sets $x_s^{'}$ is 0 but this is for this elements which are the sets picked they are, corresponding x_s value is at least $\frac{1}{f}$. This is less than equal to sum over is in $c(s)$ times f, of, f times x_s this is f times summation is in c cost of s times x_s.

But this is less than equal to f times opt this is from here that of t is greater than equal to this. So hence we have ALG by opt is less than equal to f.

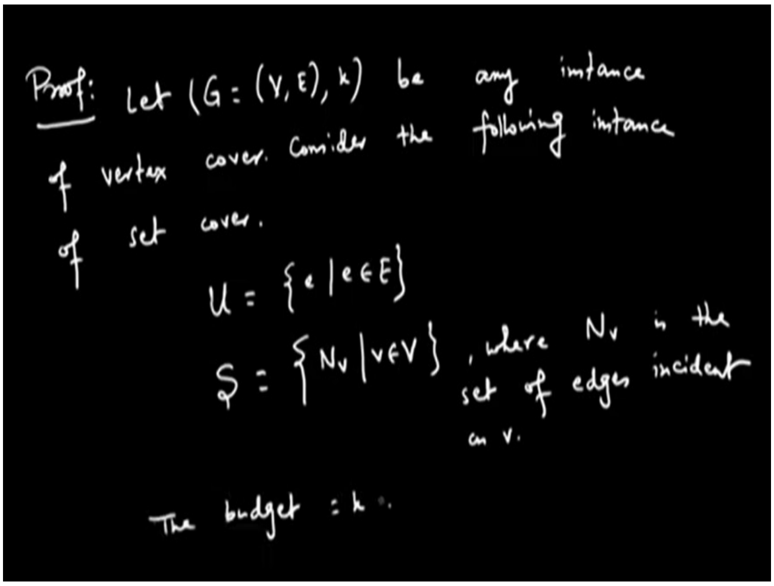
(Refer Slide Time: 23:01)



Hence the approximation ratio of our algorithm is this concludes the, proof. Next we see that how using these we can design a 2 factor approximation algorithm for vertex cover so what is the vertex cover problem? So if there is a nice reduction from vertex cover to set cover so theorem vertex covered reduces in polynomial time to set cover and this will be a something called what is called an approximation preserving.

Moreover the reduction is approximation, preserving what do, I mean by that intuitively speaking if the reduction is approximation is preserving. And if we have an alpha factor approximation algorithm for set cover then using this reduction I can obtain an alpha factor approximation algorithm for vertex cover. So that is if we have and alpha factor approximation algorithm for set cover then using this reduction we can design and alpha factor approximation algorithm for set cover for vertex cover s.

(Refer Slide Time: 27:04)



Proof so let $G=(V,E)$ and k be any instance of vertex cover considered instance of set cover for set cover. For set cover I need to say what is the universe is the set of edges e in e then what are the sets? For each vertex so the collection of sets is for each vertex the set of edges incident on it is the collection of sets. So this is neighbourhood of $v \in V$ where $N(v)$ is the set of edges incident on v and the budget is k. So in the next class we will see the equivalence of these 2 instances and also explain why this is an approximation preserving reduction.