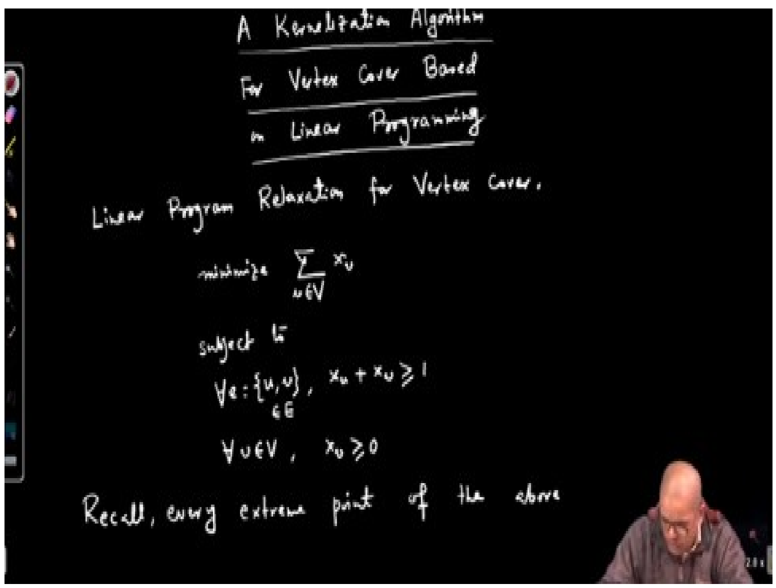
Selected Topics in Algorithm
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 60
Linear Programming Based Kernels

Welcome, in the last class we have defined kernelization algorithms and we have seen easy kernel a quadratic kernel for vertex cover problem. And we have also seen that kernelization algorithm is equivalent to FPT algorithms that means we have an FPT algorithms if and only if there exists a current location algorithm if there exists at least one algorithm for the problem. So, today we will see more non-trivial kernelization algorithm for vertex covered which is based on linear programming.

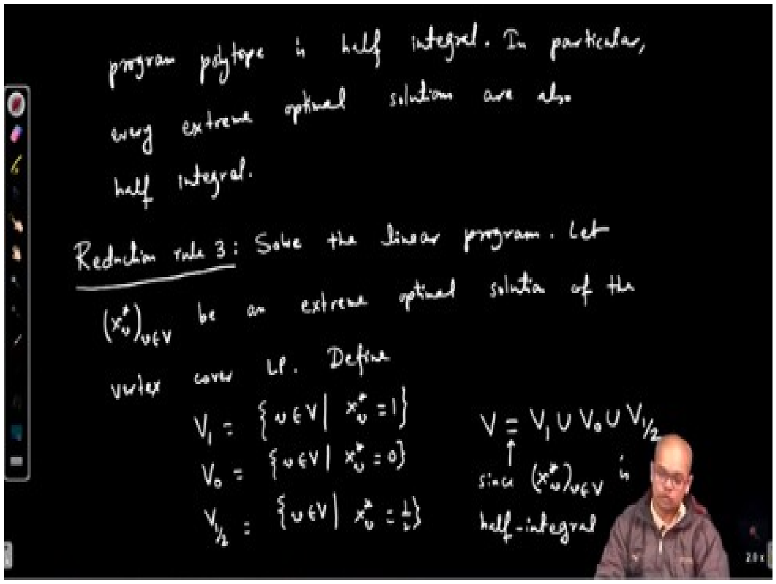
(Refer Slide Time: 00:58)



So, kernelization algorithm for vertex cover based on linear programming. So, let us recall what is the linear program for vertex cover. So, linear program relaxation for vertex covered. So, let us consider the unweighted vertex cover. So, for every vertex we have a variable V and we want to minimize $\sum_{v \in V} x_v$ and what are the constraints subject to for each edge $e=\{u,v\}$ we have this h must be covered that means $x_u + x_v \geq 1$.

And for each edge in E and for each vertex $v \in V$ x_v this is greater than equal to 0. So, now recall that we have proved that you know every extreme point of this linear program is half integral.

(Refer Slide Time: 04:30)

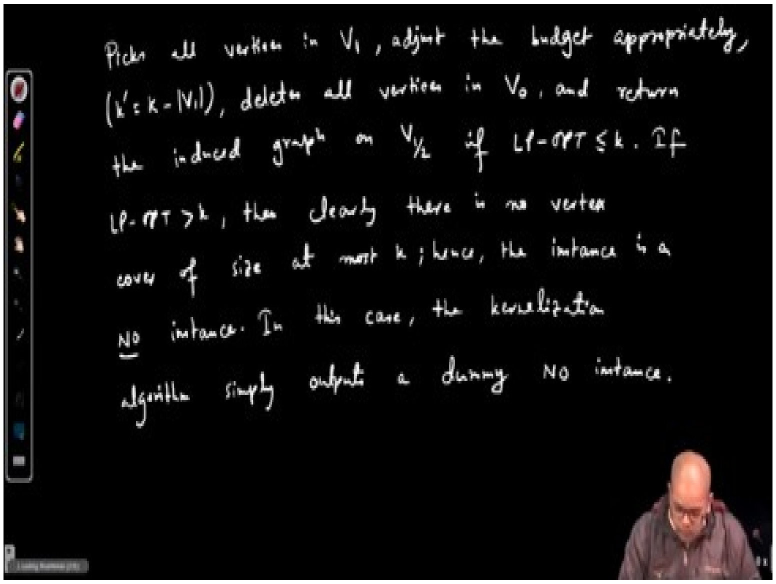


Recall every extreme point of the above linear program poly tope is half integral. In particular every extreme optimal points optimal solutions are also half integral. So, the first step is to compute this extreme compute and optimal solution. So, reduction rule what is the let us write the reduction rule. So, we first apply those reduction rules also this is an additional reduction rule which is so this reduction rule we have 0, 1 to before now we have reduction rule 3.

What does this reduction rule say? That solve the linear program so first we apply reduction 0, 1 and 2 as long as we can and then we apply reduction rule 3. Solve the linear program let $(x_v^*)_{v \in V}$ be an extreme optimal solution of the vertex cover LP then what we do is that, that means what that means x_v is a half integral and we pick all the vertices which are so define these sets V_1 is all those vertices such that x_v^* is 1 V_0 is all the vertices in v such that x_v^* is 0 and $V_{\frac{1}{2}}$ is $v \in V$ such that x_v^* is $\frac{1}{2}$.

So, reduction rule deletes all vertices in V_0 pick all vertices in V_1 and returns the induced graph on $V_{\frac{1}{2}}$.

(Refer Slide Time: 08:43)



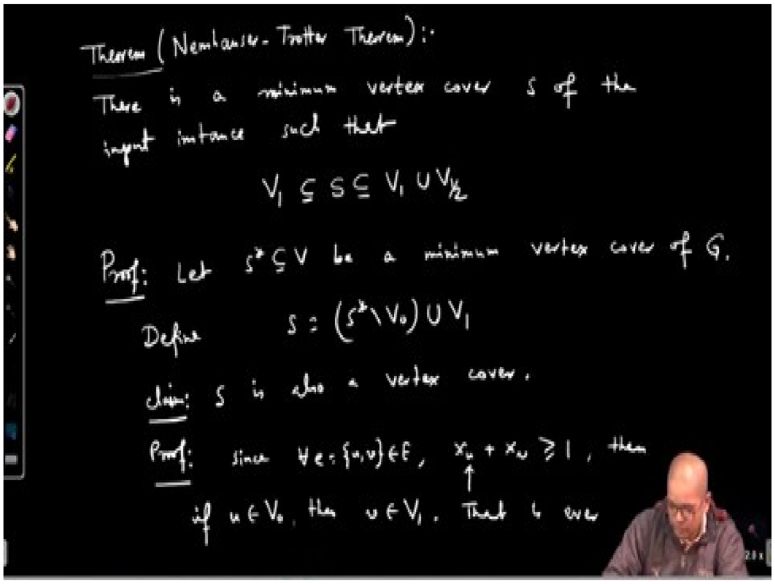
So, picks all vertices in V_1, adjust the budget appropriately that means new k prime is K - size of V_1. Then it deletes all vertices in V_0 and returns the induced graph on $V_{\frac{1}{2}}$ it all does this if LP OPT the optimal value of LP is less than equal to k. If LP OPT is greater than k then you know the size of the minimum vertex cover is at least LP OPT, LP OPT is a lower bound this you have seen in design of approximation algorithms many times.

Because the vertex cover problem is basically an integer linear program and we are doing linear programming relaxation. So, hence the LP OPT is a lower bound on the OPT so if LP OPT is greater than equal to k then clearly there is no vertex covered of size at most k if LP OPT is greater than k then there is no vertices cover of psi at most k, hence the instance is a NO instance. In this case the algorithm, the kernelization algorithm simply outputs dummy NO instance.

But now why if LP OPT is less than equal to k whatever the algorithm is doing why it is an equivalent instance that is not clear. Of course, you know because the solution is half integral all the vertices for all the for every vertex x_v^* will be either 0, 1 and half. So, this part is clear that V is union of V_1, V_0 and $V_{\frac{1}{2}}$ that part is clear. This is since $\left(x_v^*\right)_{v \in V}$ is half integral.

But why you know we are deleting V_0 and picking all vertices in V_1 part of solution why there must exist a minimum vertex cover which is contained in V_1 union V of that is not clear and that needs a separate proof.

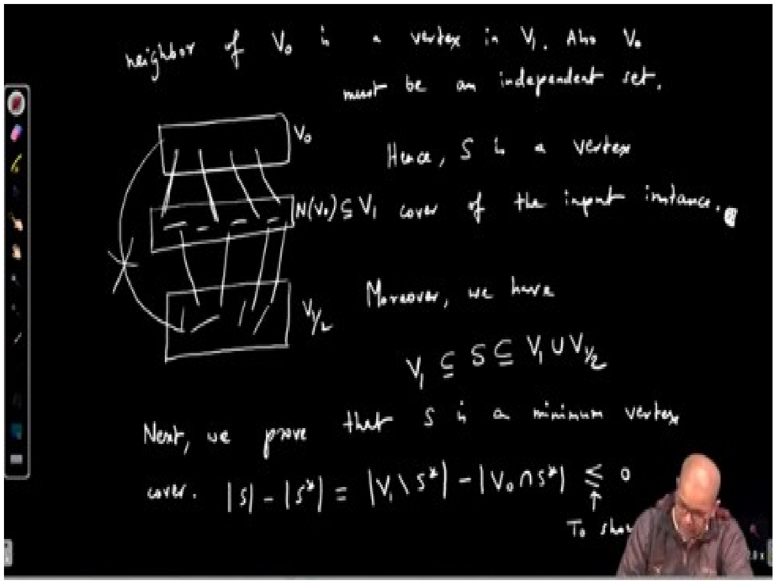
(Refer Slide Time: 14:28)



So, that is our next theorem. It is by Nemhauser and Trotter. It sees that there exists at least one minimum vertex cover which is contains V_1 and is contained in V_1 union $V_{\frac{1}{2}}$. So, there is a minimum vertex cover is of the input instance such that you know S is a subset of V_1 union $V_{\frac{1}{2}}$ and this contained in V_1. So, proof so let s star subset of V be a minimum vertex cover of G. So, what you do is that from e star you remove all vertices in V_0 and pick all vertices in V_1.

So, define, S = S^* - V_0 union V_1, claim S is also a vertex cover, why, this is so because this is true since no every vertex has an edge and at least one edge because there is no isolated vertex and because those reduction rules 0, 1 and 2 has been applied. And since for each edge $e=\{u,v\}\in E$ we have $x_u+x_v\geq 1$. So, if a vertex u belongs to V_0 that means $x_u=0$ then x_v must be equal to 1. So, since this then if u belongs to V_0 then v belongs to V_1.

(Refer Slide Time: 18:46)



That is every neighbour of V_0 is a vertex in V_1 that means here is V_0 there are some edges first see that V_0 must be an independent set also V_0 must be an independent set. There cannot be an edge like this because then both of its endpoints this value is 0 and that means therefore that is this constraint is not satisfied. So, there cannot be an edge like this within V_0, V_0 must be an independent set moreover all the neighbours of V_0.

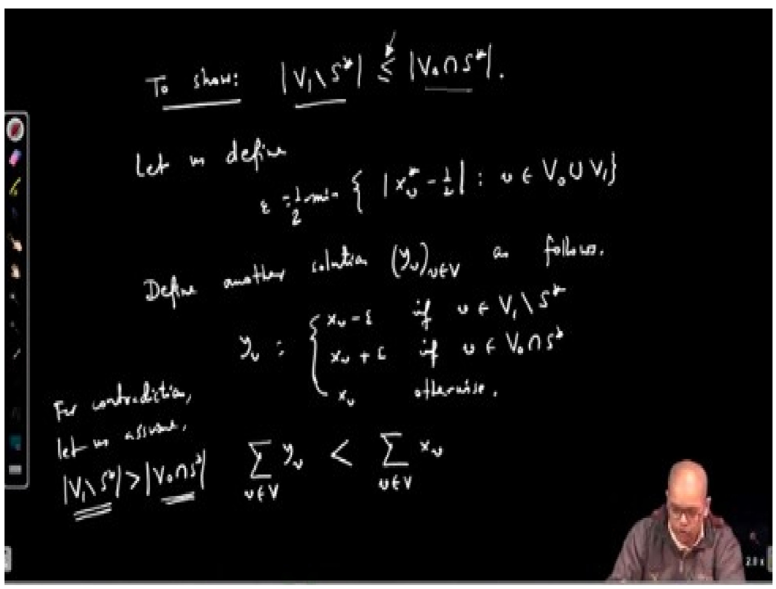
This must be in V_1 is the if this is the neighbour of V_0 this must contained in V_1 it cannot even v in $V_{\frac{1}{2}}$ because if it is in V of then also it is not these edges the constraint for these edges are not satisfied. Now what are the edges, where are the edges. Of course, the edges within V_1 are satisfied because all the edges or we have picked entire V_1 and here is $V_{\frac{1}{2}}$ all the edges within $V_{\frac{1}{2}}$ are satisfied all the edges across V_1 and $V_{\frac{1}{2}}$ are satisfied.

And all the edges between V_0 and V_1 are also satisfied and of course there is no edge between V_0 and $V_{\frac{1}{2}}$ there is no such edge. Hence is a vertex cover, it covers every edge S is a vertex cover of the input instance this proves the same. Moreover, we have S is contained in V_1 union $V_{\frac{1}{2}}$ it contains all vertices in V_1 and it does not contain any vertex of V_0. So, and this is a superset of V_1 it contains all vertices of V_1.

So, this S has all these properties this set the minimum vertex cover we are looking for S all these properties it is a vertex cover. Next, we claim, next we prove that S a minimum vertex covered for that we have a minimum vertex cover namely S^*, S^* is a minimum vertex cover and what is cardinality of S - cardinality of S^* we need to show that this is less than equal to 0. So, what is this so this is no from is what are the vertices which are in S but not in S^*.

This is the vertices in V_1 - S^*. So, these are the vertices which wherein which are in S but in not in S^* minus now there are some vertices which are in S^* but not in S which are these are V_0 intersection S^* because we have, removed all the vertices in V_0. We this is to show this is less than equal to 0 this is to show.

(Refer Slide Time: 24:07)

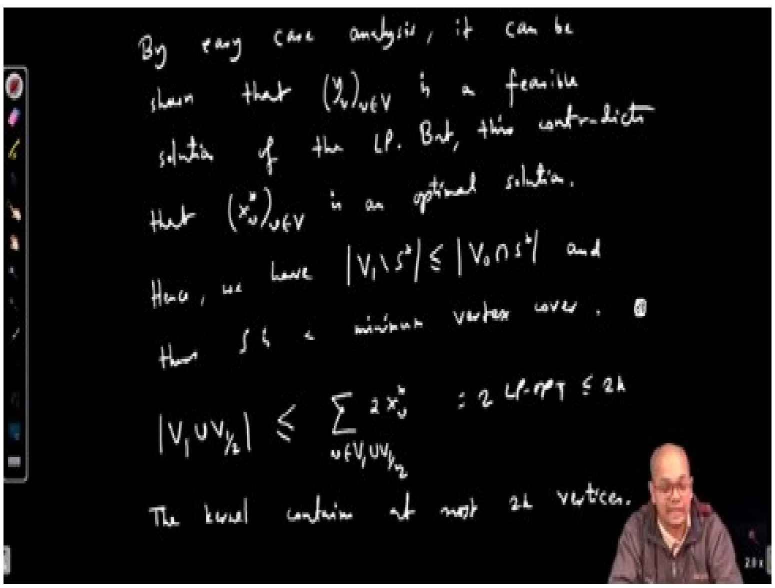


That is to show that cardinality V_1 - S^* this is greater than equal to cardinality V_0 intersection S star. For that let us define epsilon = minimum of mod of x_v^* - half such that v is in V_0 union V_1 make it half of this. For this we now decrease the fractional values from here and increase the fraction values from there. We will construct another solution and will show that if this is not the case if this inequality does not hold then we have a solution of value less than the solution for S term.

So, which is less than LP-OPT. So, define another solution $(y_v)_{v \in V}$ as follows y_v is $x_v - \epsilon$ if v is in V_1 - S star this is $x_v + \epsilon$ if v is in V_0 intersection S star and x_v otherwise. So, now what is $\sum_{v \in V} y_v$ just a minute we need to show that this is less than equal to 0 that means this is less than equal to 0 and proof by contradiction so for contradiction let us assume that mod of V_1 - S star is strictly more than V_0 intersection S star.

Now under this assumption we will show that y_v has less than y_v is less than now all the variables in V - S^* their value has dropped and all the variables in V intersection S^* their value has increased. So, more number of variables values has dropped and less number of values variables has increased. So, sum of y_v is less than sum of x_v then all we need to show that y_v is indeed a feasible solution which can be shown by easy case analysis.

(Refer Slide Time: 28:39)



So, by easy case analysis, it can be shown that $(y_v)_{v \in V}$ is a feasible solution of the LP is $(x_v^*)_{v \in V}$ but this contradicts that $(x_v^*)_{v \in V}$ is an optimal solution. But this contradicts that $(x_v^*)_{v \in V}$ is and optimal solution. Hence, we have this that size of V_1 - S^* this is less than equal to size of V_0 intersection S^* and thus S is a minimum vertex cover. Now what is the size of the kernel? Size of the cardinality V_0 union $V_{\frac{1}{2}}$ is less than equal to not V_1.

So, this is the output of the algorithm I have to do the kernelization algorithm summation you know v in V_1 union $V_{\frac{1}{2}}$ twice $\sum_{v \in V} x_v^*$ but this is twice LP OPT, but LP OPT must be at most k this is less than equal to 2k. So, size of kernel, the kernel contains the output instance which is called kernel contains at most 2k vertices which is an improvement from our quadratic kernel which contains at most k square vertices because it contains at most k square edges. So, we will stop here today.