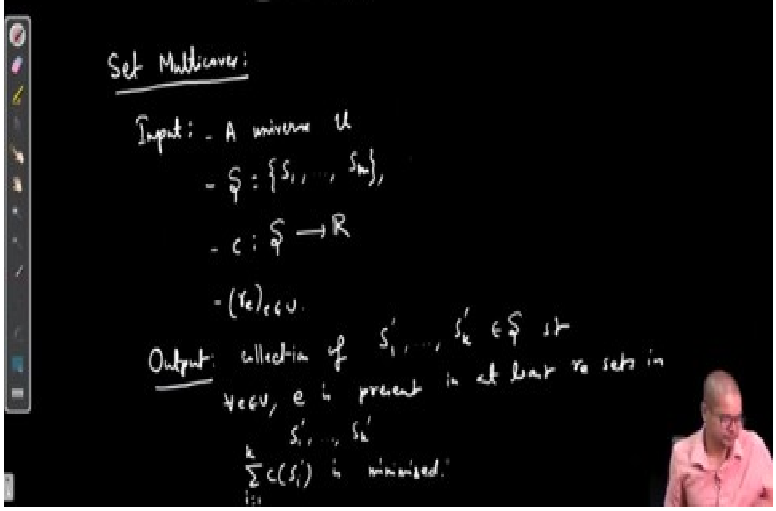
Selected Topics in Algorithm
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 50
Dual Fitting (Continued)

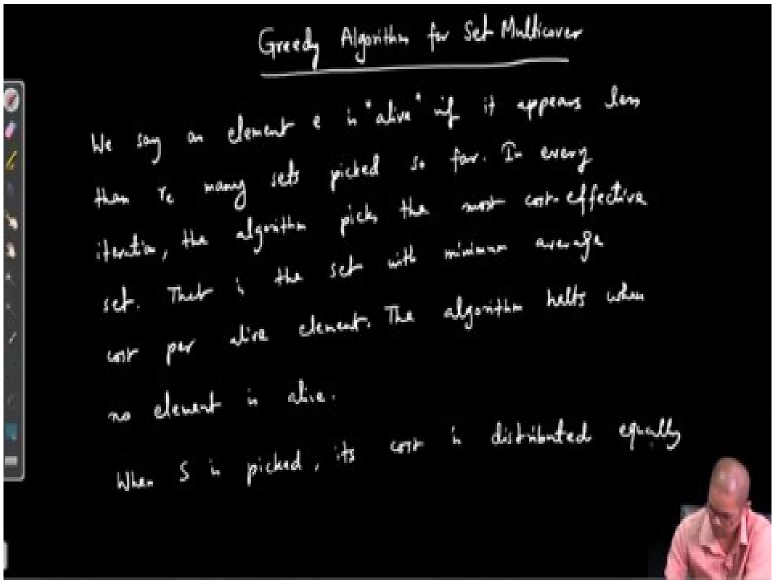
Welcome, so, in the last class we have started looking at dual fitting and using dual fitting we have analysed the greedy set cover algorithm. So, in today's class we will use a greedy algorithm for solving a generalization version of set cover problem and we will analyse it using dual fitting.

(Refer Slide Time: 00:48)



So, dual fitting so our problem is set multi covered. So, what is the input? The input is again a universe u and a collection of sets with cost function c from this collection to real number and for each element in the universe a certain requirement how many times it must be covered so r_e $e \in U$. What is the output? Collection of j 1 to $s_1^{'}, \dots, s_k^{'}$ in curl is such that you know for each element $e \in U$ e is present in at least r_e sets in $s_1^{'}, \dots, s_k^{'}$ that is one and sum of costs is minimized.

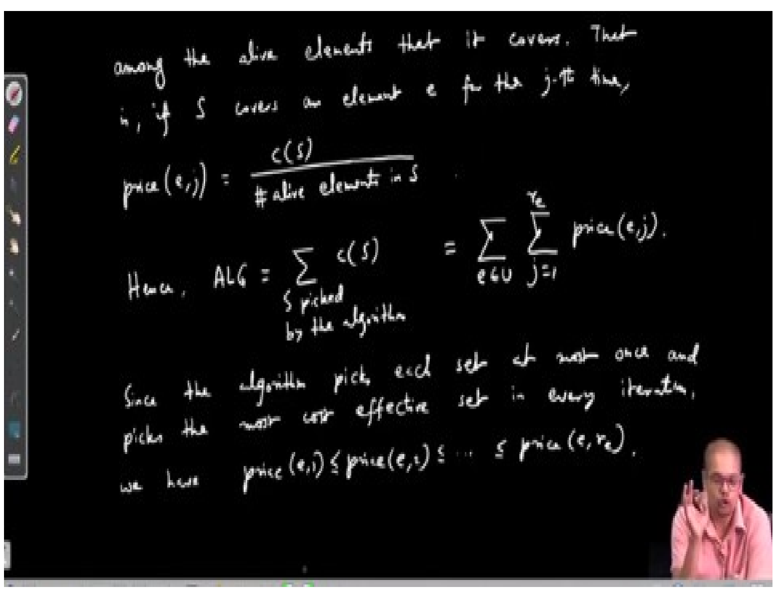
(Refer Slide Time: 03:54)



So, we will extend the greedy algorithm for set cover naturally to this more general set cover. So, greedy algorithm for set multi cover. So, let us say we say an element e is alive if it appears less than r_e many sets picked so far. And in every iteration the algorithm picks the most cost-effective set that is the set with minimum average cost per alive element. That means if for a set S if it contains 10 alive elements then its cost effectiveness is cost of that set by 10.

That is the average price with which it covers an element. The algorithm halts when no element is alive. Again, the cost is distributed across the elements covered.

(Refer Slide Time: 08:09)

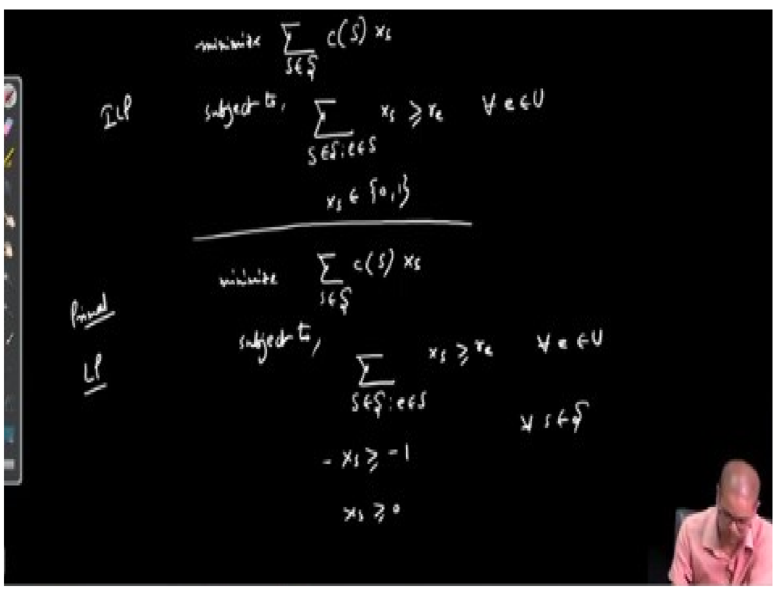


So, when S is picked its cost is distributed equally among the alive elements that it covers. That is if S covers an element e for the jth time then price of e_j price of e_j is the cost of the set picked by number of alive elements in S. Hence ALG the cost of the solution picked by the algorithm is sum of c_s picked by the algorithm and because the cost is distributed among across its alive elements this is sum over $e \in U$, j = 1 to r_e price of e_j.

And also, because the algorithm picks each set at most ones and picks the most cost-effective set in every iteration. So, since the algorithm picks each set at most once and picks the most cost-effective set in every iteration, we have you know price of e_1 is less than equal to price of e_2 less than equal to dot price of e_{r_e}. So, price of an element is a non-decreasing as more of its requirements are fulfilled.

Now we will analyse this algorithm using dual fitting. What is the first step? The first step is to come up with a dual assignment which may be infeasible but which gives a bound on opt.

(Refer Slide Time: 12:31)

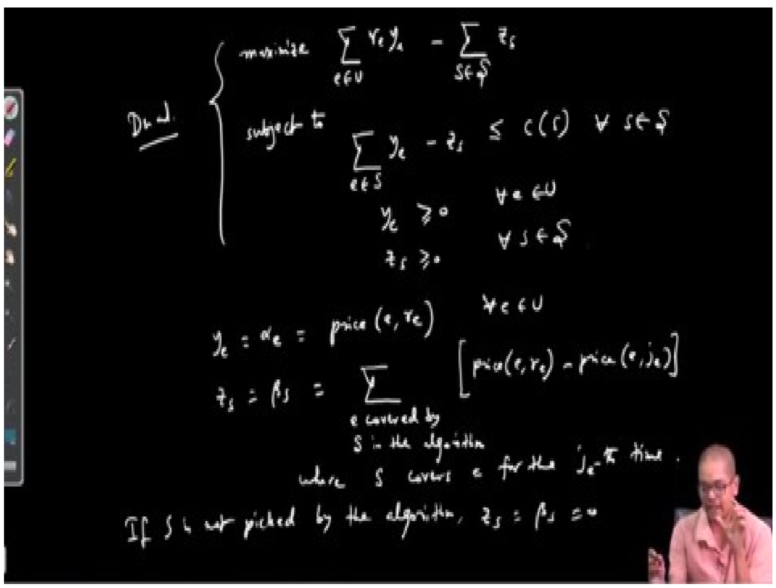


So, for that so what is the dual linear program? So, let us recall or let us write the linear programs. You know minimize summation S in curl is $c(s)x_s$ and then subject to look at all the elements for every element e in all sets which contains e. Among all those sets r_e number of them should be picked this is for all $e \in U$ and x_s in between 0 and 1 so this is the ILP formulation.

Now let us it minimizes $c(s)x_s$ is in curl S subject to this constraint is there that $s\in S$ x_s is greater than equal to r_e for all $e\in U$ and x_s in between 0 and 1 for all $s\in S$. Now here you know because each element needs to be covered more than once we cannot simply drop the constrained x_s is less than equal to 1. The minimum solution means if we simply drop x_s is less than equal to 1 the best solution may pick more than one copy of x_s of an element s of a set S.

So, that is why we need both the constraints and now to write it in the normal form what we need to write is in the standard form that x_s is greater than equal to - x is greater than equal to – 1. Because in the normal form I need to write the constraints in the standard form. I need to write the constraint is greater than equal to form and x_s is greater than equal to 0. So, this is the LP, the primal LP.

(Refer Slide Time: 15:52)



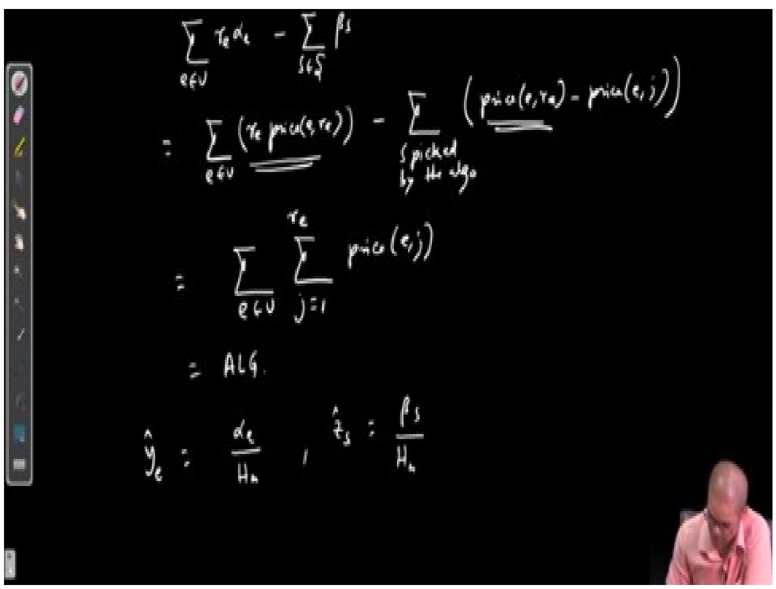
Now let us write its dual maximize now we have more constraints. So, I need to multiply these constraints for every $e\in U$ with a variable called y_e and suppose for each s I have a constraint - x is greater than equal to 1 this is multiplied with z_s. So, the right hand side becomes summation $\sum y_e r_e - \sum z_s$ subject to you know I have the coefficient for every x_s.

And the coefficient for x_s is $\sum y_e r_e - \sum z_s$. This is less than equal to c(s) for all $s\in S$. Of course, we have y_e greater than equal to 0 and z is greater than equal to 0. This is for all $e\in U$ is for all $s \in S$ so this is the dual. So, the first step of dual fitting is to find a dual assignment which sort of matches which gives a lower bound on ALG which gives an upper bound on ALG.

So, ALG is less than equal to something but often it will be equality. So, what should we set? We set y_e to be equal to α_e which is just price of (e, r_e) and z_s also we said this is for all $e \in U$ and $z_s = \beta_s$ which is summation e covered by S in the algorithm. That means what? That means in the algorithm we algorithm picks this z_s to cover certain copy of the algorithm. So, when S is peaked that means then e was alive.

Summation you know price of (e, r_e) - price of (e, j_e) where S covers e for the j_e th time if S is not picked by the algorithm. This if S is picked otherwise if S is not picked by the algorithm, then we set $z_s = \beta_s = 0$. So, first we need to show that the dual objective with this setting of dual variables how does it connect to ALG.

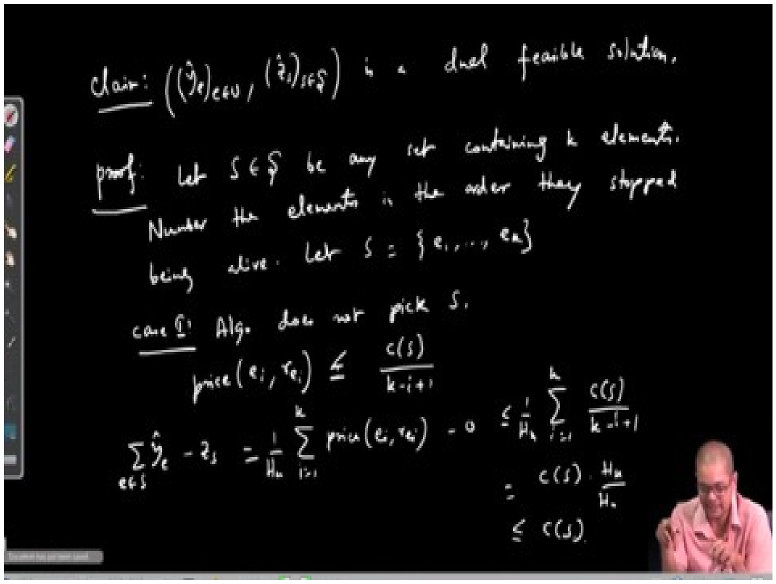
(Refer Slide Time: 20:30)



So, no summation $r_e \alpha_e$, e in U minus the z_s if S is not picked is zero so this sum is over. The set which is picked by the algorithm - β_s, $s \in S$ this is $e \in U$ r_e times price of e, r_e this minus you know S picked by algorithm. This is price of (e, r_e) - price of (e, j_e). Now because this is a valid solution there are already many sets which picks e and hence this term price of (e, r_e) gets cancelled with this and this is what we have is $\sum \sum price(e, j)$ this is nothing but ALG.

So, the first part is to show that ALG is at most the dual objective function but this is not a valid dual solution. So, we define $\widehat{y_e}$ to be $\frac{\alpha_e}{H_n}$ and $\widehat{z_s}=\frac{\beta_s}{H_n}$.

(Refer Slide Time: 23:12)

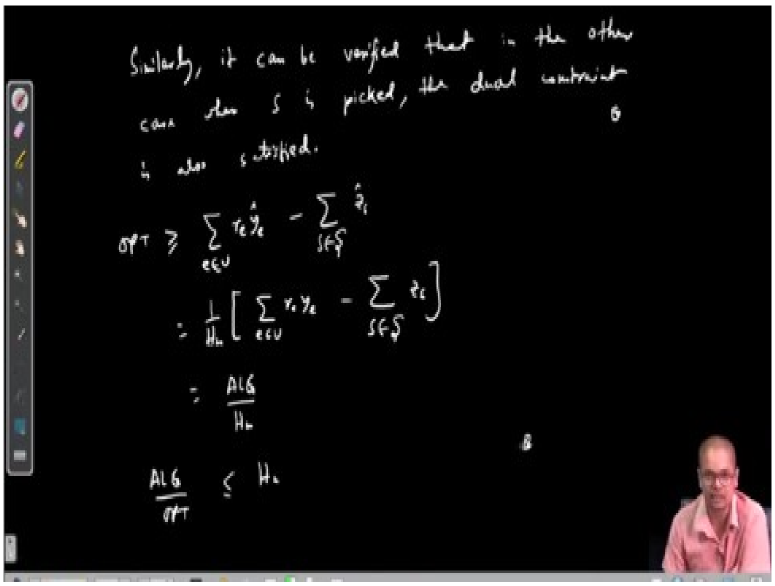


Next, we claim that $\widehat{y_e}$ $e \in U$ $\widehat{z_s}$ $s \in S$ is a dual feasible solution. Proof, so what are the dual constraints? For each set I have a constraint so let $s \in S$ be any set containing K elements number the elements in the order they stopped being alive. So, let $S=\{e_1, \ldots, e_k\}$ that means element e_1 is the first one to have stopped being alive followed by e_2 and so on. Now two cases, first case one algorithm does not pick S.

Then what we are done is Z is zero first of all let us see what are the elements there. A price of (e_i, r_{e_i}) you know when r_{e_i} the r_{e_i} copy of the last copy of e i is picked this set is still available to cover it at the cost of $C(S)$ by k - i + 1 because $e_{i+1}, \ldots, e_k$ those elements are alive. So, at this price this set is available to cover them so and we pick the most cost-effective set. So, price of e_i is less than equal to this. Now let us see.

So, what are the constraints? Summation $y_e r_e$ so summation. So, what is summation $\hat{y}_e$ in S - $z(s)$? I need to show this is less than equal to $C(S)-z(S)$. Now summation this is 1 i = 1 to k because it has k elements price. Now what is $\hat{y}_e$? $\frac{\alpha_e}{H_n}$ and what is α_e? α_e is price of (e, r_e); this is price of $\left(e_i, r_{e_i}\right)$ $\frac{1}{H_n}$ and - 0. But price is this so this is less than equal to $\frac{1}{H_n}\sum_{i=1}^{k}\frac{C(S)}{k-i+1}$. So, this is C S H k by H n which is less than equal to C S. So, the constraints are satisfied when S is not picked.

(Refer Slide Time: 28:02)



Similarly, you can check it that it can be verified that in the other case when S is picked the dual constraint is also satisfied. So, this shows that this is a dual feasible solution. So, what? So, OPT is greater than equal to the dual objective function at this value. So, what is this? This is the $\sum_{e\in U} r_e \hat{y}_e - \sum_{s\in S}\hat{z}(s)$ and what is this? This is $\frac{1}{H_n}\sum_{e\in U} r_e y_e - \sum_{s\in S} z(s)$.

But this is ALG this is $\frac{1}{H_n}$, this is $\frac{ALG}{H_n}$. So, we have $\frac{ALG}{H_n}$ this is less than equal to $\frac{ALG}{OPT}$ is less than equal to H_n. So, hence we have a H_n factor approximation algorithm. So, let us stop here for today.