Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 08

Lecture 38

Lecture 38 : Randomized Rounding for Prize Collecting Steiner Tree

couple of lectures we have seen many algorithms for maximum satisfiability problem. In this class we will see again a randomized rounding based algorithm for price collecting Steiner tree problem. we have already seen 3 factor approximation algorithm for this problem using deterministic rounding. We will see how using randomized rounding we can improve this approximation factor using same linear programming relaxation. So, let us see. So, today's problem is price collecting tainer tree.

Let us recall the linear programming relaxation that we had wrote what is the what is the problem let us recall first I am given a graph G, a special vertex r, I want to find a tree T and each edge has a cost $c(e)$. So, and each vertex has a penalty $\pi(v)$. So, if I do not include a vertex v in my tree T then I incur a cost of $\pi(v)$ and all the edges for every edge there is a cost $c(e)$.

So, all the edges that I use the cost is $c(e)$ that I pay. So, what is the and there is a special vertex r that must be there in the tree. So, what was the linear programming relaxation? So, let us write it down. We want to minimize the cost for every edge we had a variable x_e which in ILP formulation x_e would take value 1 if edge is used in the treaty otherwise it should value it should take value 0 plus for each vertex v. we have a variable y_v which will take value 1 if the vertex v is part of the tree otherwise it will take value 0 in the integer linear programming formulation which is the exact formulation of this problem.

So, I pay the cost of $\pi(v)$ or penalty if y_v is 0 is $1-y_v$ ok subject to Now what is the constraint? That if I look at any vertex i and if y_i will be set to 1 will be allowed to set to 1, only if there is a path from r to i. And how did we ensure that there is a path from r to i? That there for every cut which every subset of vertices is which is a subset of $V \setminus \{i\}$ that means, it does not contain i, but it contains r. At least one of the edges must be picked that means, at least one of the x_e value should be 1. So, for all vertex $i \in V$ for all subset $S \subseteq V \setminus \{i\}$ such that $r \in S$. If I look at the boundary edges which I denote by $\delta(S)$.

at least one of the edges must be picked that means, $\sum x_e$. If this sum is 1 then only y_i will be allowed to set 1, in particular if this sum is 0 y_i should be forced to 0. So, this is greater than equal to y_i. ok and y_r should be 1 because r is always there in the tree and for all $i \in V$ $y_i \geq 0$ for all edge $e \in E$. $x_e \geq 0$.

So, this was the LP formulation of price collecting Steiner tree. Now, if you recall what we have done, we have first chosen a threshold $\alpha \in [0,1]$ and in an optimal solution of this LP relaxation. we picked all vertices whose alpha value whose y value is greater than α. So, let (x^*, y^*) be an optimal solution of the LP relaxation ok. And we picked all vertices $u_i \in V$ such that $y_i^* \geq \alpha$ ok.

And then we use the black box algorithm to obtain a minimum cost spanning tree or spanning tree with low cost on u. So, then we compute a Stainer tree on U and there we used a black box lemma which says that the lemma is the expected or the total edge cost of that tree is less than equal to $\frac{2}{\alpha}\sum c_e x_e^*$ So, this is the lemma we used to bound the total edge cost and then subsequently bound the total penalty of missing vertices also. So, this was the idea. Now, what we do is instead of and then we found out what should be the best value of α to get the best approximation factor and we figured out that α should be equal to $\frac{2}{3}$ ok. So, not this $\frac{2}{3}$ with this we run the algorithm and we got a 3 factor approximation algorithm.

Now, in this randomized rounding based technique what we will do? We will choose α uniformly at random from $[0,1]$. this is the first natural thing to do. So, alpha is sampled uniform distribution on 0, But here you see for small values of α if α is very close to 0 then the total edge cost of the tree becomes unbounded. So, what we do is we do not sample alpha uniformly from $[0,1]$, we sample α from uniform distribution, but not from $[0,1]$ from some $[\gamma,1]$. we will again choose what γ value to choose, but again once I fix a γ value the algorithm is exactly same choose an α uniformly randomly from the closed interval $[\gamma,1]$.

Then once α is chosen the same algorithm as deterministic rounding, you pick the vertices whose y^* value is greater than equal to α compute a Steiner tree on U and output that tree that is it compute a Steiner tree T on U output t that is the algorithm. And again this lemma also holds for whatever alpha we have chosen the total edge cost of the tree is less than equal to $\frac{2}{\alpha}\sum c_e x_e^*$. And this is related to this term is related to this is at most know LP-opt this term is small. Now we are, but this cost you see is not deterministic because alpha is random. So, what we need to do is we need to compute here because it is a randomized algorithm we are interested in expected cost.

So, here is the lemma for expected cost expectation of e in edge set of T, c_e this is less than equal to I want to upper bound $\frac{2}{1-\gamma}\ln\frac{1}{\gamma}\sum c_e x_e^*$ which is a deterministic quantity because γ will be a constant which we will decide what γ to choose from the from the analysis of the algorithm. So, for the algorithm γ is constant ok. So, this proof is in straight forward. So, let us proof prove it. So, we had this inequality that e in edge set of T c_e is less than equal to $\frac{2}{\alpha}\sum c_e x_e^*$ is an optimal solution which is constant acts like constant the only randomness is from α.

So, if I compute expectation of it, then this is less than equal to expectation $\frac{2}{\alpha}\sum c_e x_e^*$. Now, this sum acts like a constant 2 also is a constant. So, by linearity of expectation this comes out $2\sum c_e x_e^*$ ok and then we have expectation of $\frac{1}{\alpha}$. Now, α is a continuous random variable distributed uniformly in the interval $[0,1]$. So, this is $2\sum c_e x_e^* \int \frac{1}{\alpha}$.

So, this is α is random variable. So, it is dx and the probability density function is just pdf of α is because it is a uniform distribution $\frac{1}{\gamma}(1-\gamma)$ which is a constant. So, let us write this integral before.

So, $\frac{2}{1-\gamma}\int\frac{1}{x}$ is $\ln x$ ok and the limit is from γ to 1 this is from γ to 1. $\sum c_e x_e^*$. So, this is ln 1 which is $0-\ln\gamma$ which is $\ln\frac{1}{\gamma}$ this is then $\frac{2}{1-\gamma}\ln\frac{1}{\gamma}\sum c_e x_e^*$ which finishes the proof. So, this is the we are able to bound the expected edge cost of our solution. Next what I need to do we need to bound the expected penalty of the vertices which we do not include in the in our solution for that we prove this lemma that expectation of $\sum \pi(i)$ the edge set of T this is less than equal to $\frac{1}{1-\gamma}\sum \pi(i)\left(1-y_i^*\right)$ ok.

So, this what we will prove. So, once we prove this lemma bounds the expected penalty of the vertices that the algorithm misses. So, once we prove this lemma we have bounded both expected cost of edges of the edges that are part of the tree T and the total expected penalty of the vertices that the algorithm misses. So, recall what was u the stay terminal vertices which we need to connect these are all the vertices $i \in V$ such that y_i^* is greater than equal to α. and again observe that any vertex which is not in tree must not be in U.

Any vertex not in the tree T is not in u because it is a standard tree it must contain all the vertices of U may be some other vertices also. And that is the main idea that you know we do not know $V[T]$ with the, but we know U because u is the vertices whose y_i^* value is greater than α. So, what I do first is we bifurcate this is into 2 terms. Now, we do not know $V[T]$, but we know that $V[T]$ is a superset of U. So, in particular $V[G] \setminus V[T] \subseteq V \setminus U$. So, if I just replace this $V[G] \setminus V[T]$ with $V \setminus U$, then we have we have more terms.

So, this is less than equal to expectation $\sum \pi(i)$ ok good. Now, you see for every vertex now let $i \in V$ be any vertex. So, what is the probability that i belongs to U? So, this in can be written down as this is $\sum \pi(i)$ times probability that $i \in U$. So, if i is any vertex what is the probability that $i \in U$? Probability that $i \in U$ you see this is 0 to 1 and there is some suppose here is y_i^*. Now, the probability that $i \in U$ is that there is some γ where if the γ belongs here.

So, this is 2 cases if γ is greater than or equal to y_i^*. then because α is chosen uniformly randomly from $[\gamma, 1]$ and $i \in U$ if and only if $y_i^* \geq \alpha$ then this probability is 0. because γ is the smallest value that α can take and this is the probability that because it is a continuous distribution probability that α is greater than γ is 1 in particular α is less than equal to γ is 0 and y_i^* is less than equal to γ. On other hand is γ is somewhere here. then the probability that $i \in U$ is that α is in this region ok.

So, here in this case if γ is less than y_i^* in the probability that $i \in U$ is only if alpha belongs to this region that α is greater than sorry α is less than y_i^*. So, this probability is $\frac{y_i^* - \gamma}{1-\gamma}$ ok. So, this thing is needed that i incur a cost if probability of i is not in γ. So, in particular probability that $i \notin U$ see if what is the probability that $i \notin U$ again 2 cases one is $\gamma \geq y_i^*$ star if $\gamma \geq y_i^*$ then it is not in there is 1. And if it is less than this then this is $\frac{1-y_i^*}{1-\gamma}$ if $\gamma < y_i^*$ and in this is $\frac{1-y_i^*}{1-\gamma}$.

But in this case also when in this case in this case also you see probability of $i \notin U$ which is 1, this is also if you compare it with $\frac{1-y_i^*}{1-\gamma}$. So, this is less than equal to. because in this case $\gamma > y_i$. So, in both cases what we have shown is this term is less than equal to sorry this term is less than equal to $\frac{1-y_i^*}{1-\gamma}$.

for all $i \in V$. So, in particular what we have is combining everything expectation of $\sum \pi(i)$ this is less than equal to the last term $\sum 1-\gamma$ is constant comes out $\frac{1}{1-\gamma}\pi(v)(1-y_i^*)$. which is exactly what I need to prove ok. So, we have bound the total penalties of the missing vertices and also the total age cost of the vertices and then we will do the standard thing that we will pick a γ which balances both of them and then once we pick the γ we will feed it in this algorithm here and get our randomized algorithm and then we will see what is its approximation guarantee ok. So, let us stop here today in the next class we will finish this