Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 08

Lecture 39

Lecture 39 : Randomized Rounding for Prize Collecting Steiner Tree Contd.

So, in the last class we have been doing the price collecting Steiner tree problem and we are doing a randomized rounding based technique and we are almost at the end of the analysis. So, let us briefly recall what we were doing. The problem is price collecting Steiner tree and the LP relaxation is minimize sum of the cost of edges picked $e \in E\ c_e x_e$ plus sum of the penalties of the vertices that we do not pick $\pi(i)(1-y_i)$. So, whenever I pick a vertex i to be part of the tree then there must exist a r to i path. $\sum x_e$ is greater than equal to y_i for all subset $S \subseteq V \setminus \{i\}$ and $r \in S$ and $y_r = 1$ and $x_e \geq 0$ for all edge $e \in E$ and $y_i \geq 0$ for all $i \in V$.

Now, we have seen already one deterministic rounding algorithm we solve it and let (x^*, y^*) be an optimal solution. then we set a threshold $\alpha = \frac{2}{3}$ and we pick all the vertices whose y_i^* value is greater than equal to $\frac{2}{3}$. And those vertices I connect using as few edges as possible as minimum cost as possible and then we have seen that that gives a three factor approximation algorithm. So, we have seen 3 factor approximation algorithm by picking all vertices $i \in V$ with $y_i^* \geq \frac{2}{3}$ and computing a minimum cost Steiner tree on this vertices. So, these we have seen it is a 3 factor approximation algorithm and then we have started looking at randomized rounding based technique we said you pick α uniformly randomly from the range $[\gamma, 1]$ and then we analyzed and we showed that we showed two lemmas lemma 1 is expectation of the cost of edges in the tree $e \in E[T]$. So, the algorithm after picking α uniformly randomly from the interval $[\gamma, 1]$ is the same as the deterministic rounding algorithm. You pick all vertices i with y_i^* value greater than equal to α and compute a minimum cost Steiner tree problem Steiner tree on it. that that is a NP complete problem, but there is an approximation algorithm which we use it as a black box.

So, we have because gamma is chosen in α is chosen uniformly randomly γ is fixed the γ value of the analysis will tell what value of γ to set. Then we have shown that this is less than equal to $\frac{2}{1-\gamma}\ln\frac{1}{\gamma}\sum c_e x_e^*$ lemma. So, lemma 1 bounds the expected cost of edges, lemma 2 bounds the sum of expected sum of penalties of the vertices that the algorithm do not pick. $\sum \pi(i) \le \frac{1}{1-\gamma}\sum \pi(i)\left(1-y_i^*\right)$ ok. So, now, expected ALG. So, ALG is the value of the solution of the out.

So, ALG is the value of the solution. So, expected ALG is expected sum of costs of edges. in T this term plus expected sum of penalties of the vertices in G which are not in treaty. the first term is less than equal to $\frac{2}{1-\gamma}\ln\frac{1}{\gamma}\sum c_e x_e^*$ and the second term is less than equal to $\frac{1}{1-\gamma}\sum \pi(i)\left(1-y_i^*\right)$ ok. So, if I take the min.

So, this is less than equal to minimum of this less than equal to maximum of these 2 terms $\frac{2}{1-\gamma}\ln\frac{1}{\gamma}\sum c_e x_e^* + \frac{1}{1-\gamma}\sum \pi(i)\left(1-y_i^*\right)$. Now this is LP-opt because $\left(x^*, y^*\right)$ is an optimal solution. So, this is LP-opt. So, this is equal to max $\frac{2}{1-\gamma}\ln\frac{1}{\gamma}\sum c_e x_e^*$ and $\frac{1}{1-\gamma}\sum \pi(i)\left(1-y_i^*\right)$. LP-opt and then LP-opt is less than equal to opt. So, this is less than equal to $max\left\{\frac{2}{1-\gamma}\ln\frac{1}{\gamma}, \frac{1}{1-\gamma}\right\} opt$.

So, this approximation factor is $max\left\{\frac{2}{1-\gamma}\ln\frac{1}{\gamma}, \frac{1}{1-\gamma}\right\}$. So, we choose γ so that this term is as small as possible. So, choose γ so that a choose γ in $[0,1]$. So, that $max\left\{\frac{2}{1-\gamma}\ln\frac{1}{\gamma}, \frac{1}{1-\gamma}\right\}$ is as small as possible. So, for that we have to equate these two terms.

So, that is $\frac{2}{1-\gamma}\ln\frac{1}{\gamma} = \frac{1}{1-\gamma}$. So, let us choose γ from the closed interval $[0,1]$. So, in particular γ is not equal to 1. So, $\gamma = e^{-\frac{1}{2}}$ ok.

So, this is the γ we choose. So, the our the approximation ratio of our algorithm is

$\frac{1}{1-e^{\frac{-1}{2}}}$ because at $\gamma = e^{-\frac{1}{2}}$ these 2 terms are same $\frac{2}{1-\gamma}\ln\frac{1}{\gamma} = \frac{1}{1-\gamma}$. So, this is the approximation ratio of our algorithm which is roughly 2.54 and you see this approximation ratio is smaller than 3. So, this is a better algorithm, but it is a randomized algorithm, but it is very easy to de randomize this algorithm.

So, since there are $|V|$ variables y_i^*, there are at most $|V|$ distinct values of y_i^*. So, considering the $|V|$ sets you know for each U_j is all the vertices $i \in V$ such that y_i^* is greater than equal to y_j^* $j \in V$. we obtain all the possible sets of vertices peaked by the algorithm. as the set of terminal vertices of the Steiner tree algorithm.

for any random choice of α ok. So, pictorially know if you plot these values y_i^* values on real line. So, maybe this is y_1^*, maybe this is y_{10}^*, maybe here is y_2^* and so on. And this y i star values suppose this is y_n^* and you see and this suppose this is 0 1. Now it does not matter in this region where α falls all values of α between corresponding between two consecutive y_i^* values will give the same output.

And hence because there are only n such values n such at most n such distinct values there are at most n such sets which we can iterate. Because, expectation of a random variable is always greater than equal to the minimum of the minimum value that the random variable can take. if we run this algorithm for all these possible sets y_j^* and pick the best one the mean cost one, then because minimum is less than equal to expectation this algorithm also achieve the same worst case guarantee. So, our de-randomized algorithm computes ah mean cost Steiner tree for every U_j $j \in V$ Steiner tree let us call T_j and outputs that Steiner tree which achieves minimum objective function value ok. Now, since expectation of a random variable is greater than equal to the minimum value that the random variable takes with positive probability.

the approximation ratio is at most $\frac{1}{1-e^{\frac{-1}{2}}}$ which is roughly 2.54. So, we have de randomized our randomized algorithm in the sense that we have extended the same idea without deteriorating the approximation guarantee of our algorithm. So, one can ask can we have a better approximation guarantee, can we have a better approximation algorithm for this problem. So, we show that if we use this linear programming relaxation we cannot have a better than 2 factor substantially better than 2 factor approximation algorithm.

if we use the same LP relaxation, we cannot have smaller than $2-\frac{2}{n}$ factor approximation algorithm. This is so, because the integral rate gap the integrality gap of this LP relaxation is at least $2-\frac{2}{n}$. So, what is the integrality gap? Let us recall it is the gap is maximum over all instances the ILP opt. which is actually $\frac{opt(I)}{LP}-opt$ ok. So, for that let us consider an instance where we have a cycle C_n cycle of length n, the penalty of each vertex is infinity that means, all vertices must be selected.

The penalty of each vertex is infinity and cost of each edge each edge is 1. Now, what is ILP opt or opt(I) $opt(C_n)$ which is $ILP-opt(C_n)$ you have to pick all edges except 1 edge that is the only solution that is the best solution which is $n-1$ defining $x_e=\frac{1}{2}$ for all edges gives a valid LP solution. take it as a homework that this setting this $x_e=\frac{1}{2}$ of course, $y_i=1$ for all i satisfies all constraints. So, $LP-opt(C_n)\leq\frac{n}{2}$.

Hence, integrality gap is greater than equal to $\frac{n-1}{\frac{n}{2}}$ which is $2-\frac{2}{n}$ ok. So, let us stop here. Thank you.