Approximation Algorithm

Prof. Palash Dey

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

Week – 11

Lecture 51

Lecture 51 : 3/2-Approximation Algorithm for Multiway Cut

Welcome. So, in the last class we have seen a two factor approximation algorithm for edge multiway cut, it was a combinatorial algorithm. In this class we will see a randomized rounding based $\frac{3}{2}$ factor approximation algorithm for multiway cut. So, let us begin. multi way cut. So, first we will write an integer linear programming formulation for this problem which will be the exact formulation in the sense that ILP-opt will be equal to opt.

So, for that we will have variables for every edge e and every connected component and for every $i \in [k]$ we have a variable $z_{e,i}$ which is 1 if the edge e belongs to the isolating cut for S_i. Now, we will see a little we will observe that each vertex or we can equivalently formulate this problem as each edge that we are removing must belong to an isolating cut. So, we can assume without loss of generality that every solution edge belongs to some isolating cut. or slitting cuts equivalently they are always exist and optimal solution F subset of E such that $G(V, E \setminus F)$ has a connected component or $G(V, E \setminus F)$ can be partitioned this vertex set can be partitioned.

can be partitioned into $C_1, \ldots, C_k$ such that s_i belongs to C_i for all $i \in [k]$ and F is nothing, but $\cup_{i \in [k]} \delta(C_i)$ the boundary edges and this $C_1, \ldots, C_k$ are isolating cuts ok. proof why we can assume without loss of generality it is not very difficult I give it as a homework. If you find difficulty proving this you ask me in the online interaction section session So, with this assumption now we have variables $z_{e,i}$. So, we can assume that each solution edge belongs to some isolating cut and here hence the optimization goal becomes minimizing sum of weights of this edges.

So, if I look at what is the contribution of this edge and edge e to the cut this is

$\sum_{e\in E, i\in[k]} z_{e,i} w_e$. Now, if I do this sum each edge belongs to 2 isolating cuts. So, this contribution is twice this sum is twice the contribution of the edge to the optimization function. So, this is you should multiply with half.

$e\in E$ and this we want to minimize and what we will have constraints to encode that this is satisfied for that what we do we have for every vertex also we have a variable which takes value 1 if the variable belongs to some isolating cut. So, for every vertex u and $i\in[k]$ we have a variable $x_{u,i}$ which takes value 1 if u belongs to the part because we have assumed without loss of generality that in the optimal there is an optimal solution where every vertex belongs to one of those parts $C_1,\ldots,C_k$ you belongs to the part C_i ok. This takes value 1 otherwise $x_{u,i}$ takes value 0 ok. So, now, let us see the constraint because it is a partition each vertex must belong to some part. So, let us write the constraints So, the fact that each vertex must belong to one part can be encoded by the constraint that $\sum_{i=1}^{k} x_{u,i}=1$ for all vertex $u\in V$ ok.

And then we need a constraint that an edge E this edge which is suppose is u v the variable z and suppose this is part C_i variable $z_{e,i}$ will be allowed to take the value 1 if and only if exactly one of the vertices u and v belongs to C_i that means, exactly one of $x_{u,i}$ and $x_{v,i}$ is 1. So, that is can be encoded as $z_{e,i}\geq x_{u,i}-x_{v,i}$ and $z_{e,i}\geq x_{v,i}-x_{u,i}$. You see if exactly one of $x_{u,i}$ and $x_{v,i}$ is 1 and the other is 0, then these two constraints say that $z_{e,i}$ is greater than equal to minus 1 and greater than equal to 1. And hence this says that $z_{e,i}$ is greater than equal to 1, $z_{e,i}$ must be set to 1 if exactly 1 of u and v belongs to C_i that means, exactly 1 of $x_{u,i}$ and $x_{v,i}$ is 1 and the other is 0.

And of course, we know that in C_i s_i belongs. So, we have s_i for all $i\in[k]$ and this constraints we have for all edge $e=\{u,v\}\in E$ ok. And what else we have these variables take value in 0 and 1. So, for all $u\in V$ $i\in[k]$ $x_{u,i}$ belongs to the set $\{0,1\}$ and for all edge $e\in E$ and $i\in[k]$ $z_{e,i}$ belongs takes value either 0 or 1. So, this is the ILP formulation exact formulation.

So, we will relax it for LP relaxation we relax these two, these constraints that $x_{u,i}$ is in between 0 and 1 and $z_{e,i}$ is in between 0 and 1 ok. Again you can get rid of this constraint this because z if you retain $x_{u,i}$ is greater than equal to 0 because $\sum x_{u,i}$ is 1 no $x_{u,i}$ take value more than 1. So, you can safely drop this constraint. Similarly, here you can drop this constraint because we are minimizing an increasing function of $z_{e,i}$ and the constraints for $z_{e,i}$ these are the constraints which depending on values of $x_{u,i}$ and $x_{v,i}$ will say that $z_{e,i}$ is greater than equal to 0, greater than equal to 1, greater than equal to

minus 1. So, in any optimal solution $z_{e,i}$ will never set to a value greater than 1.

So, again we can drop this and make the LP simpler simpler. So, this is the relaxed LP as usual we have LP opt is less than equal to ILP opt which is same as opt ok. Now, we keep another perspective of viewing this problem in terms of matrix. Recall in the last lecture we have discussed how there is an intimate connection between cuts and matrix and that we will see here. For that we define the notion of l_1 distance or l_1 metric.

So, L 1 metric given two points x y in n dimensional Euclidean space, the l_1 metric is a metric such that the distance between x and y the l_1 distance between x and y denoted by x with suffix 1 denoting l_1 metric is sum of the coordinates $i=1,\ldots,n$ absolute value of x_i-y_i. Take it as a homework to check that this is a metric that means, it satisfies the 3 conditions of a metric. So, why metric is needed? You see look at the variable. So, let us define the vector for a vertex $u\in V$ define x_u to be the vector $x_{u,i}$, $i\in[k]$ this is the vector in R to the but even more this is a vector in this also belongs to $\{0,1\}^k$ because each $x_{u,i}$ takes value in between 0 1, but even more we have more structure that each $x_{u,i}$ the sum of the coordinates must be 1. So, that we denote by Δ_k.

or k dimensional simplex. This is exactly the set of k dimensional points such that a_i is greater than equal to 0 for all $i\in[k]$ and their sum is 1 ok. So, what we can see because in the relaxed LP all $x_{u,i}$'s are positive and they sum to 1 here. So, equivalently we have a constraint the constraint can be written as $x_{u,i}$ belongs to Δ_k So, for e also let us denote this as vector edge $e\in E$ or $z_{e,i}$ takes value greater than equal to 0 and $z_{e,i}$ is from here what we can write $z_{e,i}$ is greater than equal to is greater than equal to mod of $x_{u,i}-x_{v,i}$ ok.

And in the optimal solution $z_{e,i}$ will be set to equal to mod of $x_{u,i}-x_{v,i}$. So, the objective function so, not this. So, the relaxed LP can be equivalently written. in the objective function here we can replace $z_{e,i}$ with $z_{e,i}$ is mod of $x_{u,i}-x_{v,i}$. So, let us replace it and let us see what we get half.

$\sum_{e\in E} w_e \sum_{i=1}^{k} |x_{u,i}-x_{v,i}|$. Now, this what is this? This is nothing, but the l_1 norm of x_u and x_v this is l_1 norm of x_u-x_v ok. So, the objective function becomes $\frac{1}{2}\sum_{e\in E} w_e \sum_{i=1}^{k} |x_{u,i}-x_{v,i}|$. So, let us write that here minimize half times $\frac{1}{2}\sum_{e\in E} w_e \sum_{i=1}^{k} |x_{u,i}-x_{v,i}|$ subject to x_u belongs to the k dimensional simplex for all vertex $u\in V$ and s_i must belong to C_i that means, s_i that unit vector should correspond to e_i where the i-th coordinate is 1.

So, x_{s_i} should be equal to e_i, what is e_i? e_i is all 0 except the i-th coordinate is 1. ok. This we have for all $i \in [k]$ and that is it such a crisp linear program. So, next what we will see we will do a randomized rounding based algorithm. So, let us write the pseudo code and in the next lecture we will see the analysis.

So, what we do first as usual we first solve the LP in rounding based algorithm we need an optimal solution of LP. So, let x be an optimal solution of the LP and then we initialize all those C_i's to be empty set. Next we pick a r uniformly randomly from this interval $[0,1]$ uniformly randomly ok. And then we pick up permutation. of 1 to k uniformly randomly and π is a permutation of 1 to k picked uniformly randomly again from the set of all permutations.

And then what we do we in every iteration we look at all the unassigned points initially all the vertices are vertices or points we will use interchangeably all the points are unassigned to the part. And in the first iteration we assign the points which are at most r distance away from s_i or $s_{\pi(i)}$ and assign it to $C_{\pi(i)}$. So, for $i = 1, \ldots, k-1$. So, $C_{\pi(i)}$ is So, I maintain a set x which is the set of all points which are assigned which are which are not assigned sorry which are assigned. So, x is empty set initially all points are no points are assigned.

So, x is empty set $(\pi(i), r)$. This is the ball of radius r centered at $s_{\pi(i)}$ and that means, this is the set of all points which are at most r distance away from $s_{\pi(i)}$, $s_{\pi(i)}$ what is the coordinate of $s_{\pi(i)}$ recall. it is all 0 except and 1 at $\pi(i)$ th position. So, all this elements are assigned to $\pi(i)$ except of course, all those elements or all those points which are already been assigned.

So, X is $X \cup C_{\pi(i)}$. this way I assign I build $k-1$ parts and the remaining things belong to part k. So, $C_{\pi(k)}$. So, here for loop ends $C_{\pi(k)}$ is $V \setminus X$ and then what is the corresponding set corresponding set of edges they are the boundary edges. So, return $\cup_{i=1}^{k} \delta(C_i)$ ok.

So, this is the algorithm. So, in the next class we will show that there is a $\frac{3}{2}$ factor approximation algorithm ok. So, let us stop here.