**Approximation Algorithm**

**Prof. Palash Dey**

**Department of Computer Science and Engineering**

**Indian Institute of Technology, Kharagpur**

**Week – 10**

**Lecture 50**

Lecture 50 : 2- Approximation Algorithm for Multiway Cut

Welcome. So, in the last class we have seen a primal dual based two factor approximation algorithm for generalized Steiner tree problem or Steiner forest. So, from this lecture we will see some approximation algorithms for cut problems which again important class of problems in graph optimization. So, today's topic is cut and matrix. So, let us first define what is a matrix. metric is a pair $(V,d)$, where V is the ground set and this d is the distance function $V \times V \to R_{\geq 0}$ that satisfies three conditions one is $d(u,v)$ is 0 if and only if $u=v$.

Second condition is the distance from u to v is same as the distance from v to u for all $u,v \in V$, this is for all $u,v \in V$. And the third one is called triangle inequality. What is it? It says for all $u,v,w \in V$ $d(u,v)+d(v,w) \geq d(u,w)$. So, this is called a metric and there is a related notion called semi metric which satisfies conditions 2 and 3 and partly satisfies condition 1.

semi metric is a pair $(V,d)$, where d satisfies conditions 2 and 3, but not necessarily 1. instead it satisfies 1 prime which says that $d(u,u)=0$ for all $u \in V$. you see condition 1 is the if and only if condition, hence $u \in V$ for every vertex u every point u, but that is the only case when $d(u,v)$ can be 0 that is relaxed in semi metric. in semi metric we can have 2 different points u and v $u \neq v$ and still $d(u,v)$ equal to 0 that is allowed in semi metric, but not metric, but in semi metric still $d(u,u)$ continues to be 0. Let us see what is the relationship between cuts and matrix.

So, each matrix naturally defines a semi sorry each cut naturally defines a semi matrix. Each cut in an undirected graph naturally defines a semi matrix. Let us see how let G equal to $(V,E)$ be an undirected unweighted graph. for any cut S comma V minus S where S is a subset of V and S is not equal to V and S is also not equal to empty set. that means, this is the set of vertices V and a subset is S.

Now, you define $d: V \times V \rightarrow R_{\geq 0}$ as follows $d(u,u)$ is 1 if exactly one vertex of u and v belongs to S $|S \cap \{u,v\}|=1$ and 0 otherwise. ok and this and this u, v should be an edge. So, check that $(V,d)$ is a semi metric often called the cut semi metric associated with the cut So, now, you see we can pose the min cut, max cut and other cut related problems in terms of semi metric.

For example, the min or max cut problem in an undirected weighted graph G equal to $(V,E)$ and weights $(w_e)_{e \in E}$ is to compute cut semi metric which minimizes or maximizes. this quantity sum over edges E equal to $\{u,v\} \in E\ w_e \times d(u,v)$. So, let us see this use in a concrete problem where we will design an approximation algorithm. So, the problem is called multi way cut problem. It is a generalization of mean cut problem.

So, what is multi way cut? what is input and undirected weighted graph $G=(V,E)$ and weight function $(w_e)_{e \in E}$ and some k vertices $s_1, \ldots, s_k \in V$. They are all distinct vertices the goal compute a set F of edges which minimizes $\sum_{e \in F} w_e$ some of the weights of this edges $w_e$'s we will assume as usual is greater than equal to 0. So, we want to compute a set F of edges with total with minimum total weights and if I remove F from the graph G, every $s_i$ and $s_j$ pair becomes disconnected and there is no path between $s_i$ and $s_j$ for $1 \leq i < j \leq n$ $G(V, E \setminus F)$ ok. So, this is the multi way cut problem for $k=2$ it is the mean cut problem and it is polynomial time solvable for $k=2$.

this is the minimum s to t cut problem which can be solved in polynomial time using any max flow algorithm ok, but from $k=3$ onwards this problem is NP complete. So, what we will show we will see a two factor approximation algorithm a simple two factor approximation algorithm for this problem. So, let us define an notion of isolating cut. An isolating cut for $s_i$ is a cut which disconnects $s_i$ from $s_1, \ldots, s_n$ except $s_i$ is a cut $F_i$ there are two equivalent ways of defining a cut one is a set of edges which corresponds to a partition of the vertices $(S, V \setminus S)$ and the other is that partition of the vertex you define.

So, here we are defining using $F_i$ because this way it will be easy to write the optimization function. An isolating cut for $s_i$ is a cut $F_i$ such that there is no path between $s_i$ and $s_j$ in $G[V, E \setminus F_i]$ for all $j \in [k-i]$ computing isolating cuts can be done in polynomial time by adding a high cost adding a new vertex t connecting it to all $s_1, \ldots, s_k$ except $s_i$ with very high cost edges and then computing a minimum s-t cut. And a min cost or minimum weight isolating cut for $s_i$ can be computed in polynomial time by adding a new vertex t in the graph.

and connecting t or adding an edge between t and $s_j$. for all $j \in [k-i]$ of weight twice summation of all the weights of the edges ok. By adding a new vertex t in the graph adding an edge between t and $s_j$ for all j of weight this and then computing ah minimum weight is decut in the resulting graph. So, what does our algorithm do? We simply compute the an isolating cut a minimum weight isolating cut for every $i \in [k]$. So, we compute a minimum weight isolating cut $F_i$ for all $i \in [k]$ and then output all such edges.

output $\cup_{i \in [k]} F_i$ ok. So, we claim that we this algorithm has an approximation ratio of 2 theorem. Our algorithm has an approximation ratio of at least 2 proof. So, let $F^*$ be an optimal solution. So, if you remove $F^*$ we have connected components and each vertex S i is in the in is in its own connected component.

In particular there is no connected component which has both $s_i$ and $s_j$. So, let us call the connected component where $s_i$ is $C_i$ this is the candidate component. So, what is opt? Opt is $\sum_{e \in F^*} w_e$. Now, you notice that all the boundary edges of $C_i$ they belongs to $F^*$, they are subset of $F^*$ that means, $\delta(C_i)$ is a subset of $F^*$ and each edge appears exactly twice in boundary edges. For example, if the edge is there is an edge between $C_i$ and $C_j$, then this particular edge belongs to both $\delta(C_i)$ and $\delta(C_j)$.

So, then this is greater than equal to twice for all $i \in [k]$ for all edges $e \in \delta(C_i)$ $w_e$, but you see delta $C_i$ is an isolating cut and we have used minimum weight isolating cuts. So, this is greater than equal to twice summation $i \in [k]$ sorry this should not be twice this should be half because each edge appears twice half half $i \in [k]$ and this is weight of $F_i$ ok. And, but summation of weight of f i is at least alg which is summation of weight of f is half alg. So, what we have is alg is less than equal to twice opt, hence it is a 2 factor approximation A slight modification of this algorithm will gives a slightly better approximation guarantee it is $2-\frac{2}{k}$ factor approximation algorithm I leave that as a homework.

In the next class we will see a linear program based randomized rounding technique which gives a $\frac{3}{2}$ factor approximation algorithm for multi way cut. So, let us stop here. Thank you.