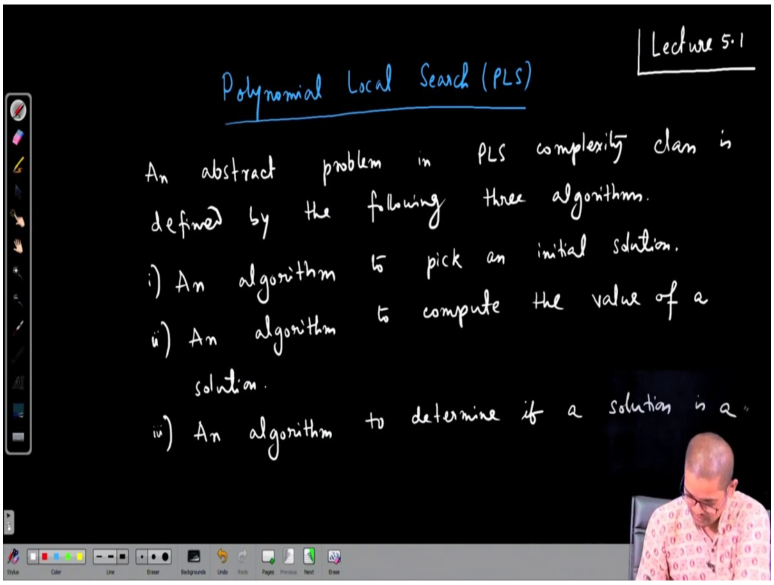
Algorithmic Game Theory
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 21
PSNE for Congestion Games

Welcome, in the last lecture we started the framework or machinery of showing the lower bounds and we started talking about local search problem. So, let us continue that line and this we will continue in for the next couple of lectures.

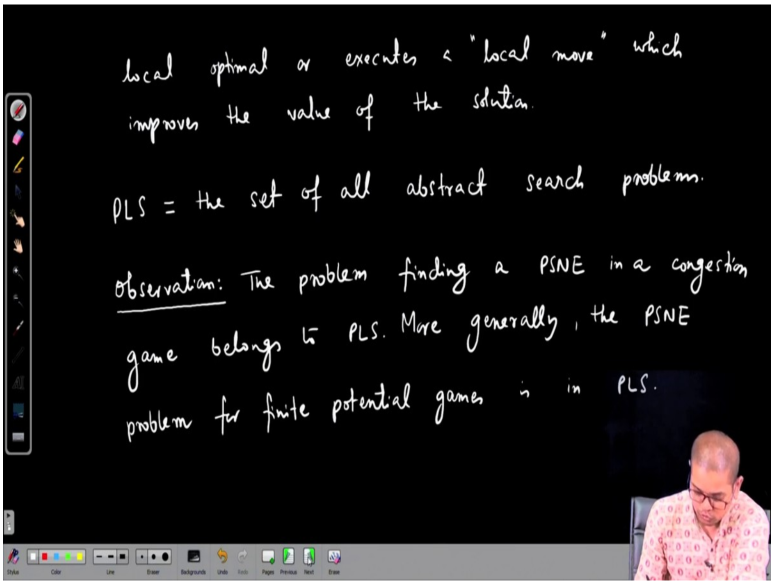
(Refer Slide Time: 00:46)



So, the complexity class that we are defining is called polynomial local search and we observed that these problems are not like decision problems and that is why we need to build up the entire machinery again. So, what is the problem here? So, what is the, an abstract problem in PLS complexity class is defined by the following three algorithms. What is the 1st algorithm?

An algorithm to pick an initial solution, typically this algorithm is very simple you just pick any arbitrary initial solution. Then the 2nd algorithm; an algorithm to compute the value of a solution so, for the local max cut problem it is just computes the weight of the max cut, and an algorithm 3rd algorithm that an algorithm to determine if a solution is a local optimal.

(Refer Slide Time: 04:07)



Is it a local optimal or if it is not, then or it executes a local move which improves the value of the solution.

What is the value of the solution? This is exactly what is computed by the second algorithm. So, for the local max cut problem either you either there is an algorithm which either says that it is a local optimal or it executes a local move; that means, it moves one vertex from this set to another and thereby, which it improves the value of the solution. So, the complexity class PLS is the set of all abstract search problem ok.

Next, what we need so, you see that we observe that our problem of finding an PSNE in a congestion game or network congestion game they all belongs to this PLS because, of the potential function. So, observation: The problem of finding a PSNE in a congestion game belongs to PLS. So, any finite potential game belongs to this for the problem of finding a PSNE in any finite potential game belongs to this complexity class PLS, why? What are the three algorithms?

The first algorithm is you pick any strategy profile the second algorithm we need to we need to define what is called a value of a solution. So, value we define it as the value of the potential or potential function at that strategy profile. And, what is the local move? Local move is that we see that if there is there exist a unilateral beneficial deviation for any player. If there exist such a deviation unilateral deviation we make that deviation and that is the local move and if there is no such deviation then it is a local optimal of the potential function and hence by definition it is a PSNE of the potential game.

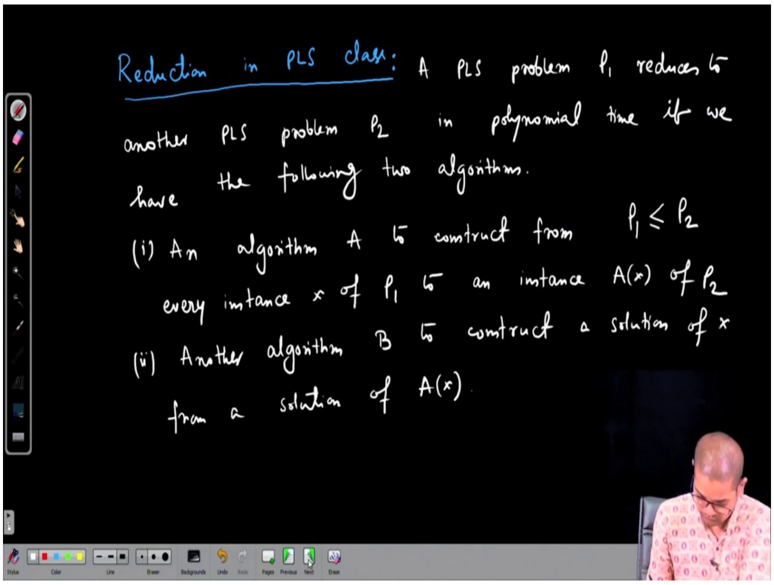
So, more generally, the PSNE problem more generally the PSNE problem for finite potential games is in PLS, but what we will show is that the PSNE for problem for a congestion game is PLS complete or in particular PLS hard and by that we mean that it is one of the hardest problem in PLS.

And if we know how to solve this in polynomial time then we will be able to solve all problems in PLS in polynomial time and there exist many problems in PLS for which people have tried long and we still do not have any polynomial time algorithm.

The an example of such a problem is local maximum cut problem for weighted graph. Recall or see that the local max cut problem for unweighted graph is polynomial time solvable because, by simply carry on this local moves and the maximum cut size could be at most ${}^{m}C_{2}$ and so by or before ${}^{m}C_{2}$ iterations we must find a local max cut.

So, local max cut problem for unweighted graph is clearly polynomial time solvable, but we do not know how to find a local max cut for weighted graph and it is believed to be one of the hardest problem and using this problem we define the PLS the complexity class PLS or PLS hard. So, what is PLS hardness? So, to define PLS hardness we need to define the notion of reduction in the PLS complexity class. So, that is what we define next.

(Refer Slide Time: 10:53)



Reduction in PLS class: So, we say a PLS problem P_1 reduces to another PLS problem P_2 in polynomial time if we have the following two algorithms if we have the following two algorithms.

But what is the motivation, first let us see the motivation when we want when we say that P_1 reduces to P_2 we want to say that if there is an algorithm for solving P_2 in polynomial time we can use that algorithm to solve P_1 ok. So, what other things we I need to use the algorithm for P_2 to design a algorithm for P_1 what other algorithms I need.

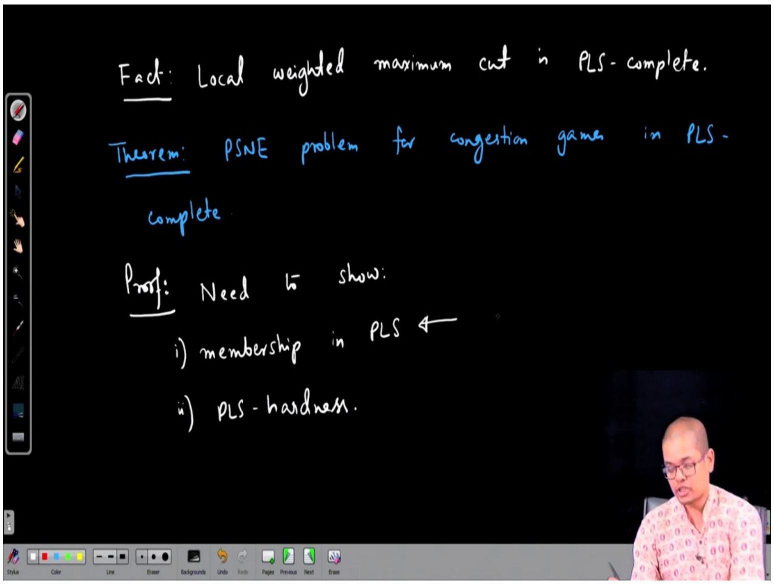
So, I am given an instance of P_1 and I want to use the algorithm for solving P_2, but the algorithm for solving P_2 only works on instances of P_2. So, I need an algorithm to convert an instance of P_1 to an instance of P_2 that is the 1st requirement. An algorithm let us call it A to map or to construct from every instance x of P_1 to an instance $A(x)$ A is the algorithm $A(x)$ of P_2 ok.

And now once I have an algorithm once I have an instance for P_2 I can run the algorithm for $P_2$and get a solution for P_2, but I need a solution for P_1, I need a solution for x which is a which is a problem instance of P_1. So, I need another algorithm to construct a solution of x from a solution of $A(x)$.

So, another algorithm B to construct a solution of P_1 a solution of x from a solution of A x ok. So, what is the solution of x, it is a local optimal optima of this problem P_1 and what is the solution of x it is again a local optima according to the problem P_2 in the problem P_2.

So, one thing is it should be cleared that if we have these two algorithms then if somebody claims that they have a polynomial time algorithm for the problem P_2, then I can use that algorithm to construct another polynomial time algorithm for problem instance x of P_1 or polynomial algorithm for problem P_1 ok.

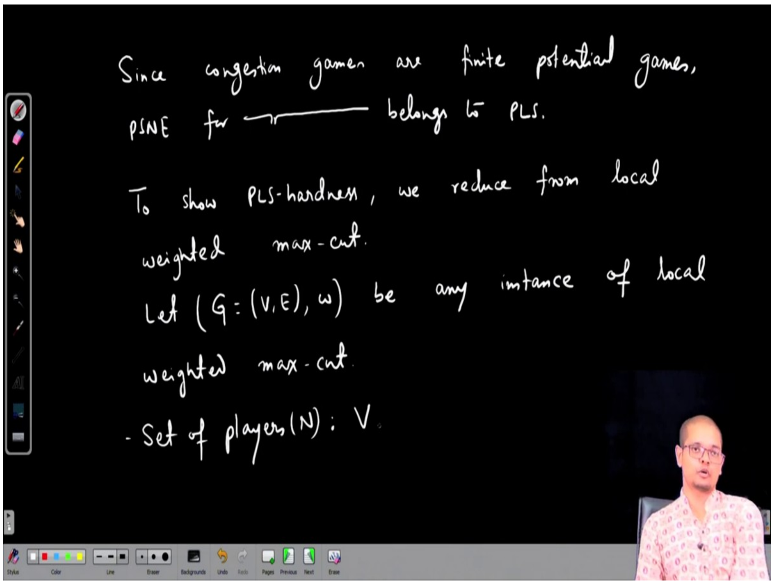
(Refer Slide Time: 16:41)



Now, we so, what we will do is that we will use a fact that so, fact we will use without proof is that local weighted maximum cut is PLS complete, using this we will prove that theorem. PSNE problem for congestion game is PLS complete proof, why? So, for PLS completeness we need to show two things one is the membership in PLS that this problem belongs to the complexity class PLS and we need to show that its PLS hard.

So, let me write need to show membership to PLS membership in PLS and 2nd is PLS hardness. So, why this problem belongs to PLS? This is we all know this is because; congestion games let me write it in next page.

(Refer Slide Time: 19:20)

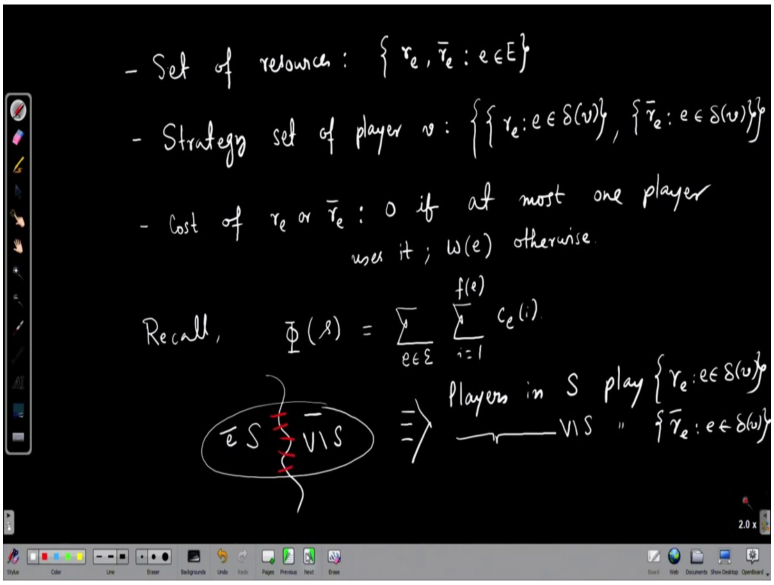


Since, congestion games are finite potential games, assume finite congestion games finite potential games. So, it PSNE for congestion games belongs to PLS ok, because it is like finding a local minima of the potential function. So, membership is done second is PLS hardness and for that we will reduce.

So, let me write, to show PLS hardness we reduce from local weighted max cut ok, we reduce from this. So, let us see the reduction. So, let $G=(V,E)$ and W be any instance of local weighted max cut. What do we mean by reduction? We need to construct an instance of congestion game.

Now, to define a congestion game we need to say what are the players and what is the ground set of elements and what are the strategy sets. So, first define set of players N, this is what, this is a set of vertices V. So, vertices correspond to players.

(Refer Slide Time: 22:25)



Then, the set of resources so, for every edge e I define two resource r_e and $\bar{r}_e$ $e \in E$, this is my set of resources and what is the strategy set of player v say, strategy set of player v it has two strategy sets look at the edges that incident on v.

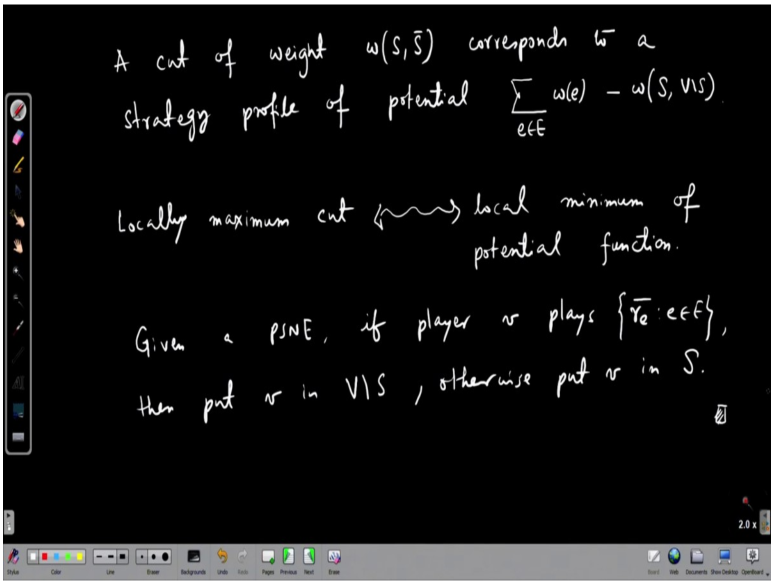
So, let delta v be the set of edges whose one end point is v and you look at those edges and the corresponding r_e that subset is one strategy and another is $\bar{r}_e$, $e \in \delta(v)$. So, player v has only these two strategies. When you define cost of resource cost of r_e or $\bar{r}_e$, this is 0 if at most one player uses it and one otherwise, not one, weight of e otherwise weight of e is defined in the weighted local max cut problem weight of e otherwise ok.

Now, you see to get the to get a feel about the construction. So, what was the potential function the Rosenthal potential, recall $\phi(s)$ what was it, it was you sum over the resources $\sum_{e \in E} \sum_{i=1}^{f(e)} c_e(i)$. Now, let us see the equivalence. So, suppose I have a cut suppose I have a cut and $(S, V \setminus S)$.

So, then the corresponding strategy profile what we I should look at is that players in a players in S play $r_e, e \in \delta(v)$ and players in $V \setminus S$ play $\bar{r}_e, e \in \delta(v)$. So, you see for every for look at any edge e which is entirely belong to S entirely inside S for that edge both it end both its end points the corresponding players play that r_e. So, the load of that a resource r_e is 2.

And hence the cost it contributes is w e, same with any edge $V \setminus S$ the load of r_e is 0, but the load of $\bar{r}_e$ is 2 and hence the total contribution of the cost to the cost is w_e, only the cut edges the corresponding resources does not contribute anything they contribute 0.

(Refer Slide Time: 27:38)



So, what we see is that a cut of weight $w(S,\bar{S})$ corresponds to a strategy profile of potential how much so, for all edges every edge contributes its cost its weight except the cut edges so, w_e e in E minus weight of this cut. In particular if this cut $(S,\bar{S})$ is a local maxima then the strategy profile the corresponding potential the strategy profile will be the local minima of the potential function.

So, locally maximum cut they correspond to local minimum of potential function. Now, local minimum of potential function is exactly what is a PSNE? So, what is the other algorithm? So, given a PSNE, if v if player v plays $\bar{r}_e, e\in E$, then put v in $V\setminus S$, otherwise put v in S.

And the above argument shows that if the strategy profile is a local minimum of potential function. And in particular it is a PSNE then it is it must be a local max cut, the corresponding cut is the corresponding cut $(S, V\setminus S)$ must be a local max cut, which is exactly what we need to prove. This concludes the proof of the statement. So, we will continue in the next class.