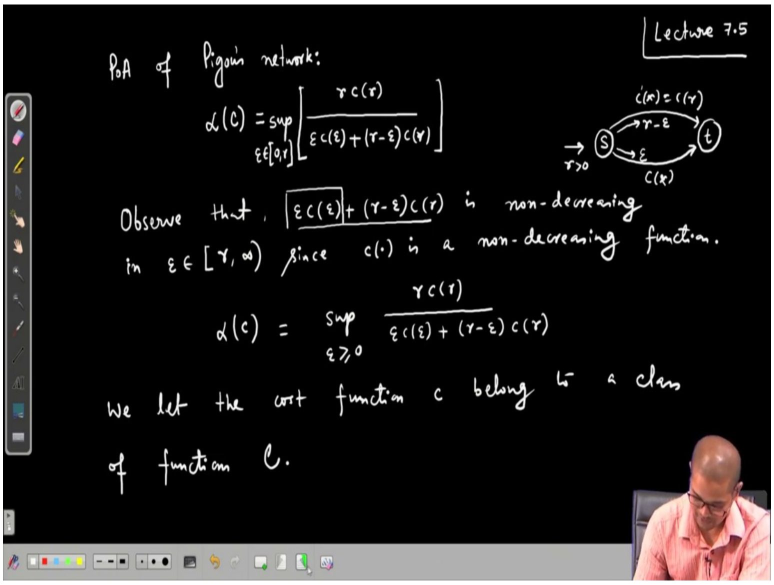
Algorithmic Game Theory
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 35
PoA of Selfish Routing Game

Welcome. In the last lecture, we have studied started studying Price of Anarchy and we were demonstrating Price of Anarchy in the Selfish Network Routing Game and towards that, we have studied the price of anarchy of Braess paradox and Pigou's network and today, we will prove the main result of selfish network routing which shows that the price of anarchy of any network is at most the price of anarchy of Pigou's network. In the last class, we formally defined Pigou's network.

(Refer Slide Time: 01:03)



Now, let us define formally the price of anarchy of Pigou's network. So, price of anarchy we have seen, it depends on the cost function and it increases with the amount of non-linearity in the cost function. So, price of anarchy is alpha, price affinity of the Pigou's network it is a function of the cost function. And it is what? As we have observed that the here is s here is t and in the general Pigou's network, r is the amount of traffic that needs to be sent from s to t.

And the cost function of the bottom edge $c(x)=x$ and the cost function of the top edge $c(x)=c(r)$. So, the top; so, the bottom edge weekly dominates the stop top edge and in the weekly unique weekly dominant strategy equilibrium, all the traffic all the r amount of traffic will follow the bottom edge and the total cost will be $r\,c(r)$.

By on the other hand, so let us consider what happens if I send epsilon amount of traffic on the top edge and $1-\epsilon$ amount of traffic on the bottom edge? So, in $r-\epsilon$ amount of traffic, so the cost from the top edge is $\epsilon\,c(\epsilon)$; $c(\epsilon)$ is the cost of the top edge and epsilon fraction of the traffic is using that edge plus $r-\epsilon$.

See the epsilon amount of traffic is following the bottom edge and the r minus epsilon amount of traffic is following the top edge, $c(r)$. And we want to maximize this ratio over all ϵ; but because its ϵ, these are continuous function and the range of ϵ can be anything from say $[0,r]$. So, we cannot write max because maximum may not be attained, we need to write sup. Sup epsilon in the range $[0,r]$ ok. Now, you observe that the denominator the numerator is fixed.

It does not depend on ϵ; but the denominator depends on ϵ, but it keeps on it is a increasing function of ϵ. So, the denominator $\epsilon\,c(\epsilon)+(r-\epsilon)c(r)$ is non-decreasing in epsilon in even r to ∞. This cost function $c(x)$ is defined for all x greater than equal to 0 and if you set ϵ to be more to if you allow ϵ to be more than r, the first this keeps on increasing.

Because or it should not decrease because the cost function is not decreasing. So, the first part you see increases with epsilon because epsilon increases with ϵ and $c(\epsilon)$ increases with ϵ or at least does not decrease, non-decreasing and the second part strictly increase with ϵ. So, the sum of these two functions increases with ϵ, since c is a non-decreasing function.

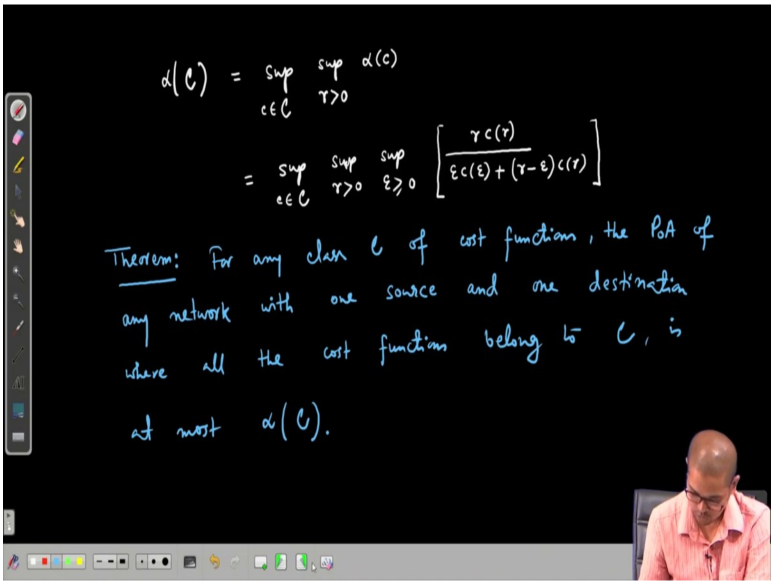
So, we can write alpha c simply as supremum over ϵ greater than equal to 0. Because supremum of this ratio is the infimum of the denominator because the numerator does not depend on ϵ and the numerator keeps on increasing with ϵ. So, we can let ϵ go beyond r and that that does not change the value of price of anarchy. It just simplifies the notation.

Now, we have seen that you know in Pigou's network, the price of anarchy increases as we increase the amount of non-linearity. So, in some sense, we have to restrict the class of functions from where the cost functions can come. In Pigou's network, there is only one cost function. But in an arbitrary large network with many edges and vertices and edges, there can be various cost functions and you know cost functions, it should not be too non-linear.

If it is too non-linear, then price of anarchy bound will also be so high. So, to restrict the amount of non-linear non-linearity that a function cost function can have, what we do is that we use we let the cost function small c vary or not vary belong to a class of cost function; a class of functions C. So, this class could be the set of all linear functions or it could be set of all quadratic functions or it could be set of all cubic functions and so on.

So, it just it ensures that the cost functions are not completely arbitrary and we extend the notion of price of anarchy not with respect to small c with respect to capital C, it says that what is the worst price of anarchy possible, if the cost function small c is allowed to vary in the class of functions capital C.

(Refer Slide Time: 10:16)



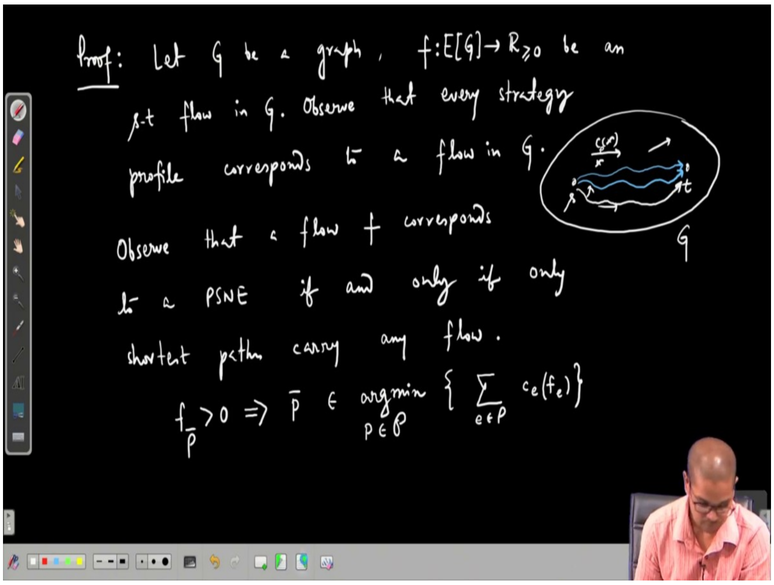
That is we define alpha of capital C is supremum, what the cost function c is in capital C. And we also let r vary because the amount of traffic can vary. So, we also let r vary;

supremum r greater than 0 $\alpha(c)$. So, which is nothing but if you write $\alpha(c)$ from last page, we get $supremum_{c \in C}\, supremum_{r \geq 0}\, supremum_{\epsilon \geq 0} \frac{r\,c(r)}{\epsilon\, c(\epsilon) + (r - \epsilon)\, c(r)}$ ok.

So, now, we are in a state, where we can state the theorem; the main theorem of selfish network. For any class, class c of functions or cost functions, the price of anarchy of any network with one source and one destination, where all the cost functions belong to class c is at most $\alpha(c)$ ok.

So, this is what we mean by saying that the price of anarchy of any selfish network routing game, it does not depend on the network structure so much. It only depends on the amount of non-linearity in the cost function. If we fix the amount of nonlinearity in the cost in the cost function, then the price of anarchy is at most the price of anarchy of the Pigou's network ok. So, the rest of the class will prove this result.

(Refer Slide Time: 14:08)



Proof: Let G be a graph and if, so here is the my arbitrary network G and here is my source node s, here is my destination node t and there is r amount of traffic that is going through various paths, which is like flows. So, if is the flow function. We first observe that any strategy profile corresponds to flows.

The first observation observe that every strategy profile corresponds to a flow in G ok. So, thus, instead of working with strategy profiles, we will be directly working with flows, it will be much more convenient. Next, we observe that how will a PSNE a flow look like. So, next a crucial observation; observe that a flow if corresponds to PSNE if and only if only shortest paths carry non carry any flow; shortest path where.

So, you know you look at this graph G and you consider a flow f. Now, what is flow? It assigns certain traffic to each edge and depending on the traffic, if x is the traffic, the cost of this every edge is $c(x)$. So, with weights being $c(x)$ of every edge; $c_e(x)$, this cost function depends on this edge e also; $c_e(x)$. Now, we have a weighted graph and look for shortest s to t paths.

First observation is that if there is a path from s to t which is not the shortest path, then that path does not carry any flow. In the sense that there must exist at least one edge which has 0 amount of flow. If not suppose there is a path which is from s to t which is not shortest; but carries a non-zero amount of flow; that means, all the edges in that path carries non-zero amount of flow.

Now, because cost functions are continuous and non-decreasing, I can move a little amount of flow from the non shortest path to a shortest path arbitrarily small very small so that the cost of the white path which is not shortest slight drop slightly. And the cost of the shortest path after moving the flow after adding the small amount of flow increases very small.

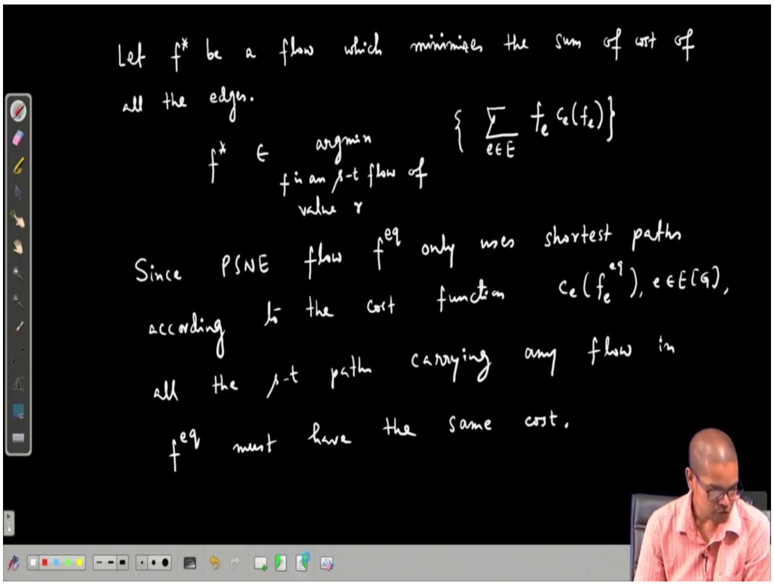
But still the cost of the non shortest path is more than the cost of the path which were shorter. So, in not. So, this demonstrates that you know there exist unilateral deviation, it is possible to change the path of some flows which will benefit the unilateral deviator. So, there exist unilateral beneficial deviation which is exactly what PSNE does not allow.

So, only shortest paths will carry non-zero will carries any flow and any flow means that you know all edges, there must be at least one edge which whose flow value is 0, if a path the path is not the shortest path that is what I mean. So, that means, let me write it. So, if a path; if I take a path say $\bar{p}$ and if its flow is greater than 0; what is the flow along a path? You look at the flow values of all the edges and take the minimum that is the flow flowing through that path.

That suppose the $\bar{p}$ is a path and $f(\bar{p})$ is greater than 0; $f(\bar{p})$ is the flow along the path $\bar{p}$, then it must be the case that $\bar{p}$ is a shortest path. $\bar{p}$ belongs to argmin p in cal P; cal P is the set of all s to t paths; capital P is a path, path is a color set of edges. So, you sum over all the edges in the path $c_e(f_e)$.

This is the cost of this path and this path must be the must be a shortest path with respect to this cost function ok; good ok. Now, you see how should the socially optimal path of optimal flow will look like that means, a flow which minimizes the total cost sum of costs.

(Refer Slide Time: 22:17)

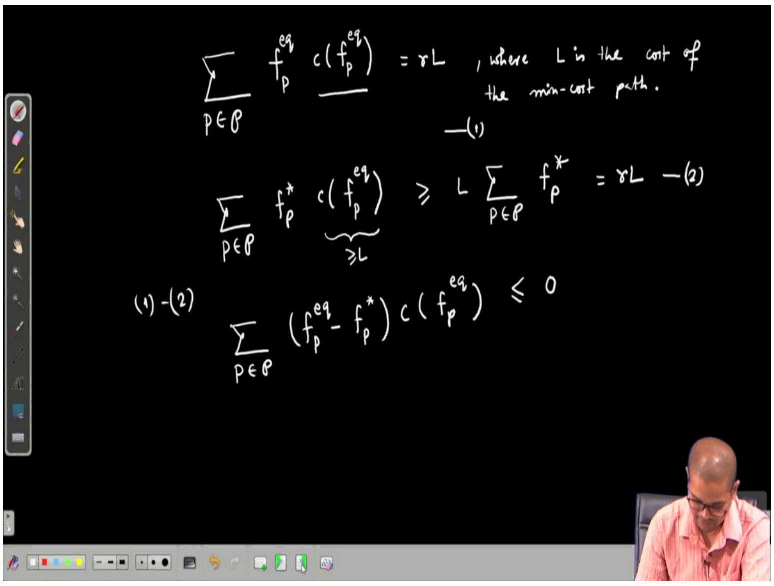


So, let f^* be a flow which minimizes the sum of cost of all the edges; that means, f^* belongs to argmin among all s to t flows f is an s to t flow of value r; what is the cost of this flow? You sum over all the edges $e \in E$ and the cost of this edge is $c_e(f_e)$. And what is the amount of traffic flowing this using this h? f_e; this is the total cost and f star b one such flow which minimizes this total cost.

Next, you observe that you know since a PSNE flow, what is PSNE flow? Look at a PSNE strategy profile and look at the corresponding flow. Since PSNE flow f^{eq} only uses shortest paths according to the cost function, $c_e(f_e^{eq})$ all the paths, all the s to t paths carrying any flow in f^{eq} must have the same cost. Indeed, because all are shortest paths;

only shortest paths can carry flow. If it is not a shortest path, then it then it is not a PSNE flow.

(Refer Slide Time: 26:04)

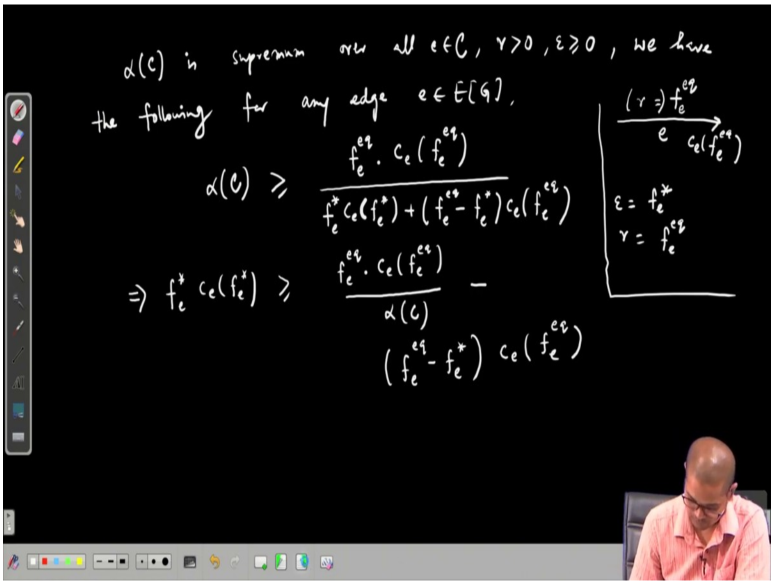


So, now, you see that let us look at this value. We sum over all paths p in cal P f_p^{eq} cost of f_p^{eq}. It is the sum of all costs. Now, you see some paths does not carry flow. The in that case that the f_p^{eq} will be 0. On the other hand, for the other paths which carry flow, the cost of that path will be the same.

So, assume that the cost of the shortest paths be L; then this value this becomes L, L comes out and the total flow must be r. So, these r times L; where, L is the cost of the shortest path or min cost path shortest is same as min cost here ok. Good. So, this is equation 1. Now, what you do is that we replace this f_p^{eq} with f_p^{*} and do the same summation; let us see how what should be its value. f_p^{*} and the cost function remains as it is.

Now, what will be its value? You see that this value is greater than equal to L; the shortest paths for shortest path this value is L, for other paths this value is greater than equal to l. So, this is greater than equal to L times summation p in P f_p^{*} and this sum is r. This is greater than equal to is r L; this is 2. So, we do 1 minus 2, then what we get is summation p in P (f_p^{eq} - f_p^{*})$c(f_p^{eq})$, this is less than equal to 0 ok; good.

(Refer Slide Time: 29:36)

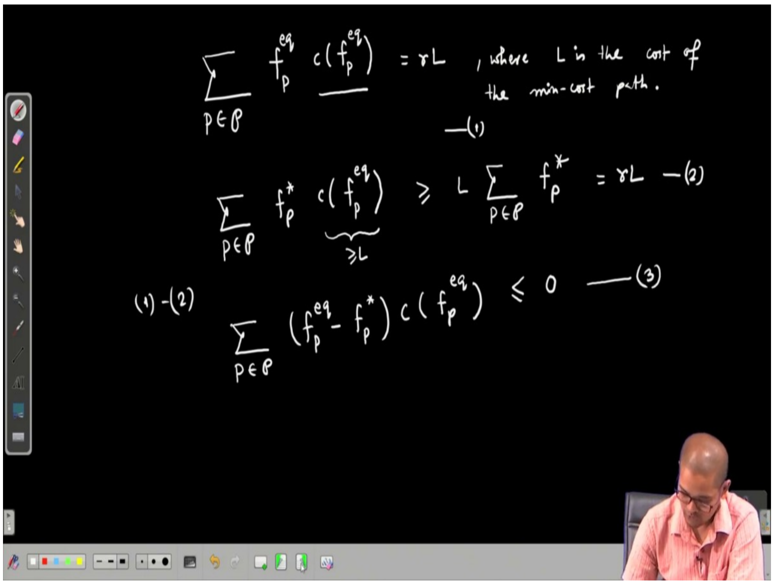


Now, let us see what is alpha c. Alpha c is supremum over all cost function c in C r greater than 0 and epsilon greater than equal to 0 since this. Then, we have the following for every edge; focus on any edge, we have the following for any edge $e \in E[G]$. What we have? You know alpha c is greater than equal to focus on that edge and any edge e, it is the traffic it is carrying is f_e^{eq}, equilibrium traffic and hence the cost is $c_e(f_e^{eq})$.

So, the amount of traffic that this path is carrying is $c_e(f^{eq})$ and in the expression of alpha c, it was $r\,c(r)$. So, replace r with take the value of r to be this. This flow for when you are considering this edge. Now, you will use why or we understand why we generalized the traffic from 1 to r. So, this is $f_e^{eq} c_e(f^{eq})$ by ϵ; the value of epsilon that we put is f^*, f_e^*. So, $\epsilon = f_e^*$ and $r = f_e^{eq}$.

And now, we also appreciate why we allowed epsilon to be more than r because here f_e^* can be more than f_e^{eq}. So, $f_e^* c_e(f_e^*) + (r f_e^{eq} - \epsilon)\ r = f_e^{eq}$. From that, what is $f_e^* c_e(f_e^*)$? This is greater than equal to $f_e^{eq} c_e(f_e^{eq})$ by $\alpha_C - f_e^{eq} - f_e^* c_e(f_e)$ eq ok. So, for every edge, we have this inequality and we have we let us add all those inequalities for all the edges.

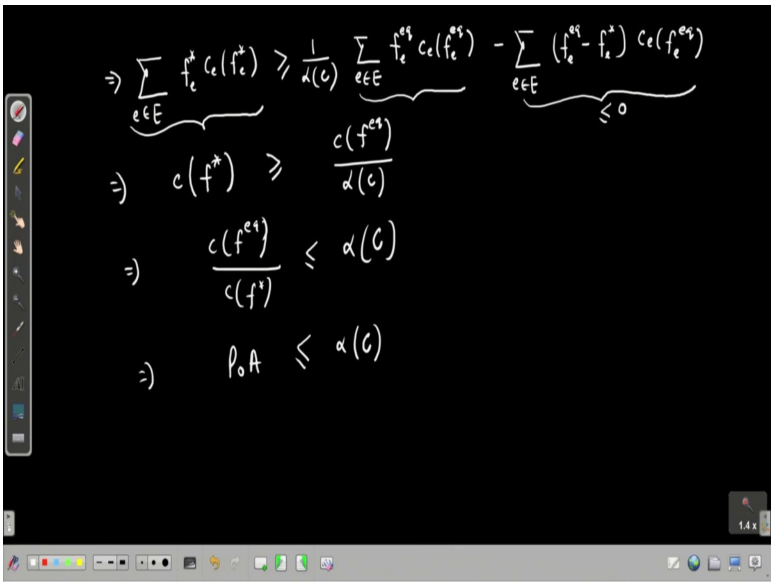
(Refer Slide Time: 34:05)



So, then, we get what? Adding all the inequalities for all the edges, we have $\sum_{e\in E} f_e^* c_e(f_e^*)$ greater than equal to $\frac{1}{\alpha_C}$. Now, what is the thing? This is you see this is the cost of f^* cost of this flow.

This is greater than equal to and this is the cost of the equilibrium flow; $\frac{c(f^{eq})}{\alpha_C}$ and this part without the negative sign, this is less than equal to 0. This is we have seen here, here this equation 3.

(Refer Slide Time: 35:59)



So, without the or with the negative sun sign, this is positive and we can forget that. Because if some term is greater than equal to sum of two terms, each of them is positive, you can get rid of the other one term and the inequality continues to hold. So, now, we have $\frac{c(f^{eq})}{c(f^{*})}$ is greater than less than equal to α_C and the left side is exactly what is called price of anarchy. Price of anarchy is less than equal to α_C.

This is exactly what we need to prove that the price of anarchy of this network is at most at most α_C which is the price of anarchy of Pigou's network. So, is. So, correlative from here is that if the cost functions are linear, then the price of anarchy of any network, it does not matter how complex it is; how many edges or how many vertices it is.

If all the cost functions are linear, then the price of anarchy is at most the price of anarchy of Pigou's network which is four-third which is very very strong result ok. So, we will stop here, ok.