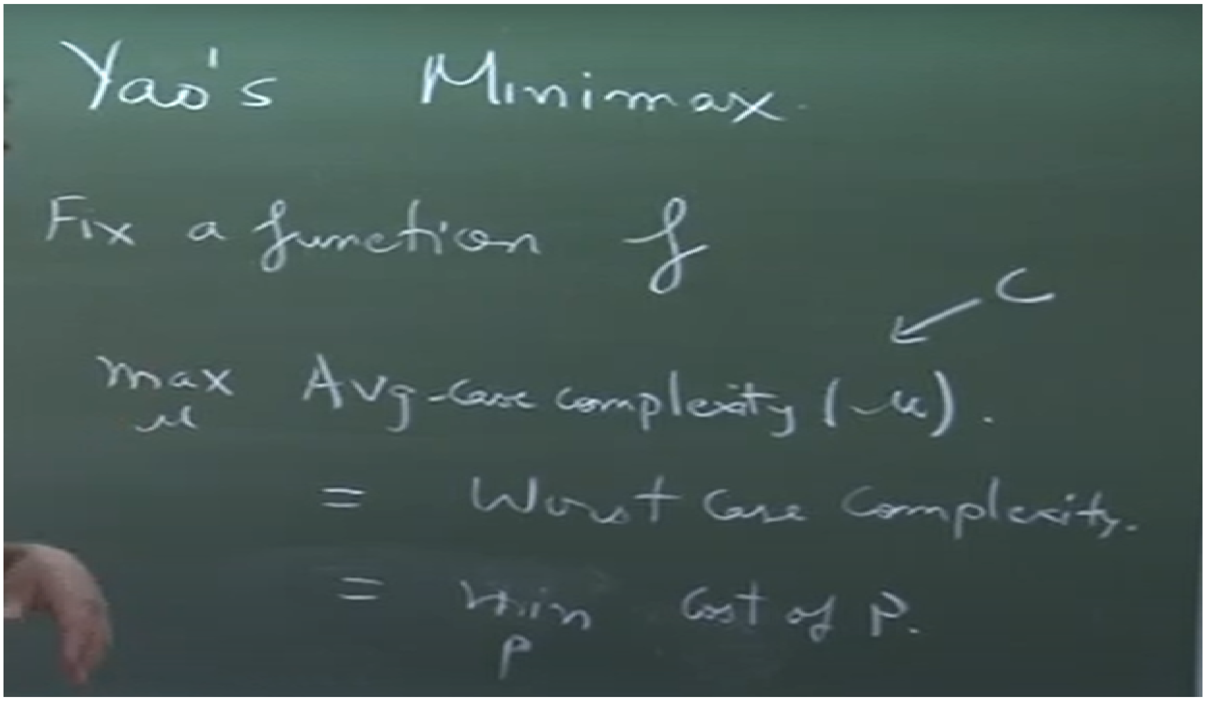
Linear Programming and its Applications to Computer Science

Prof. Rajat Mittal
Department of Computer Science and Engineering
Indian Institute Of Technology, Kanpur

Lecture – 37
Lower bounds using Yao's Minimax

Welcome to another lecture on linear programming. We are looking at Yao's Minimax theorem. We are going to look at the proof once again in a slightly different light. Hopefully this clarifies things. Remember what do we want to do? We wanted to, we had defined the model of communication complexity. We said in the case of randomized communication complexity, if we fix a function f, then the average case complexity over distribution mu, the maximum average case complexity over mu is equal to the worst case complexity.

The reason why we want to view it like this is actually here, the model of communication complexity, the way we define the complexity are not that important. In most of the randomized settings, this kind of a theorem works. I want to give you an intuition of why. If you look at this statement, max over mu average case complexity mu, worst case equal to worst case complexity.



Yao's Minimax.

Fix a function f

$\swarrow C$

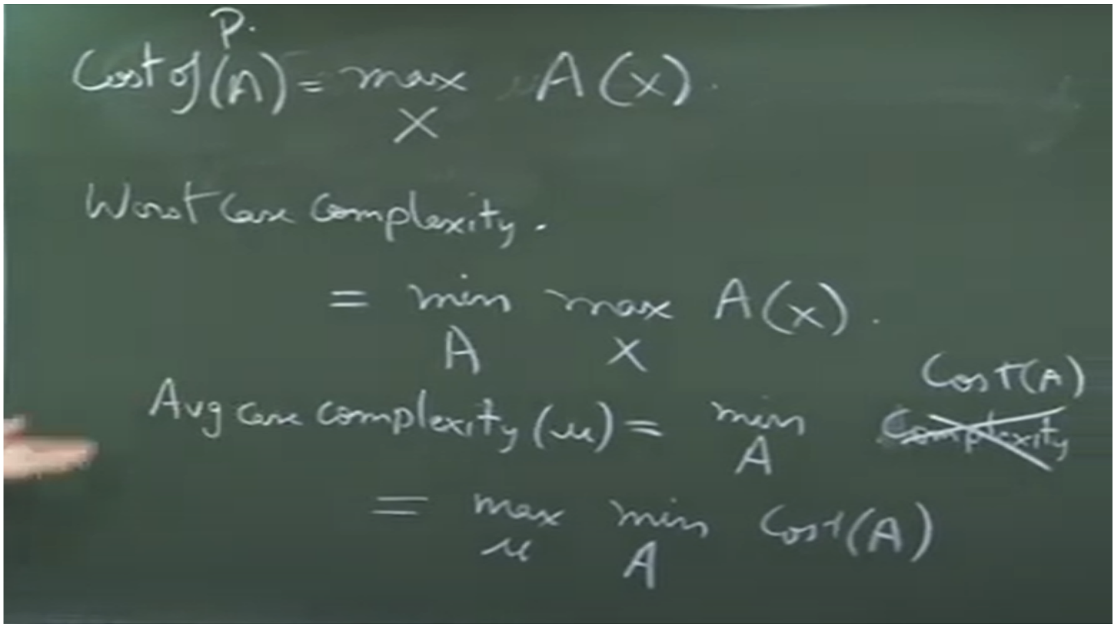
$\max_{\mu}$ Avg-case complexity $(\sim\mu)$.

= Worst Case Complexity.

= $\min_{P}$ Cost of P.

I am not defining what I mean by average case complexity. I am not defining by what I mean by worst case complexity in that particular model. But in general, what is a worst case complexity? For an algorithm, we define the worst case complexity to be maximum over x comma y, cost of the algorithm. So, maximum over input x is, this is the cost of a, cost of any algorithm or it could be a protocol, whatever you want to think of. Then, now my worst case complexity is going to be min over all possible algorithms or protocols because obviously, I want to choose the best algorithm or protocol.

Again, remember which model is this, how exactly I define the cost, I am not talking about it. Similarly, if I talk about this quantity on the LHS, what is the average case complexity over mu? This is minimize over all algorithms, correct or I should say not complexity, but cost of a. Now, cost of a does not depend on x because there is a distribution I have in mind. That is why I cannot write as a of x, it is summation mu x a x. Then, if I want to maximize over mu, this looks like max over mu min over a, cost of a.



$$\text{Cost of } \overset{P.}{(A)} = \max_{X} A(x).$$

Worst Case Complexity.

$$= \min_{A} \max_{X} A(x).$$

$$\text{Avg Case Complexity } (\mu) = \min_{A} \overset{\text{Cost}(A)}{\cancel{\text{Complexity}}}$$

$$= \max_{\mu} \min_{A} \text{Cost}(A)$$

So, you already see min max kind of coming into the plane. This is true for lot of randomized situations query complexity, communication complexity, Las Vegas, Monte Carlo for almost all of them, you can still show this kind of. Again, the intuition is this that all of them look like maximum. They already form a min max structure. The point is in that model, can you construct a nice value of cost, so that these things make sense.

Sounds good? What is the average case complexity of mu? What do I mean by that? I assume that my input distribution is from mu. My x is coming out from mu and then what is the cost? So, that would be, so my x comes with this probability and then, cost of x. This is why this is worst case, max over x. This is average case. Average case means according to a probability distribution.

If this input distribution was uniform, this would be the average cost. So, average cost is generalizing this term, where my weight on my x could be different. That is why this is an average cost with respect to the distribution. Sorry no This cost means this. Once I fix the distribution in mind, the cost with respect to that distribution of an algorithm is this.

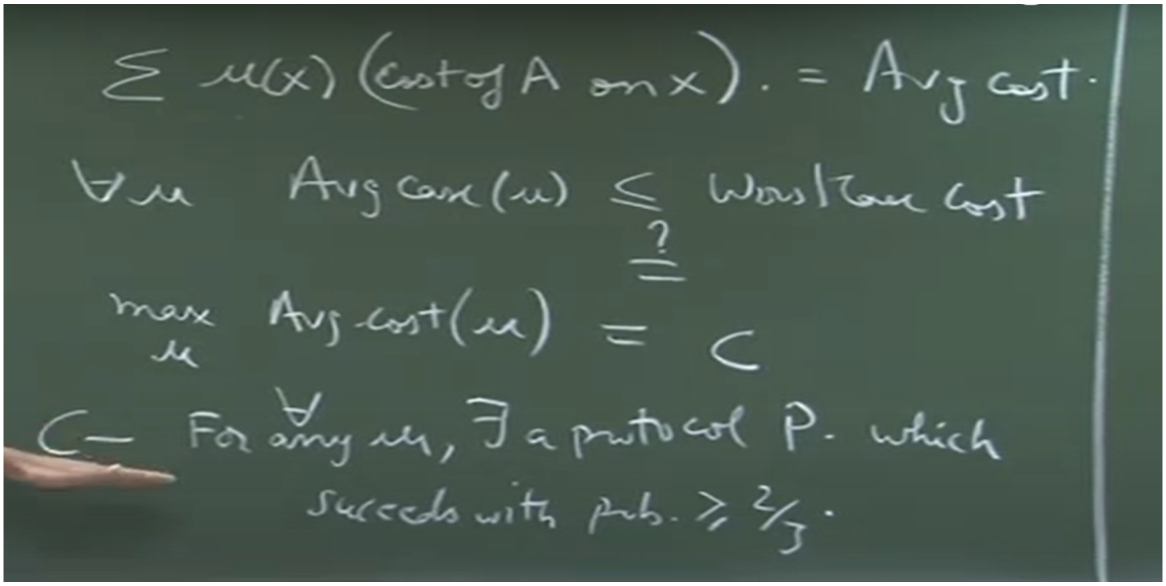
Then, if I want to construct, if I want to figure out the worst case complexity, I am saying you take the maximum over all possible distributions. The average case complexity turns out to be worst case complexity. So, this max mu is not part of cost. I am saying if you maximize the cost, average cost over all possible distributions, it turns out to be worst case. That is the importance of this statement, right

So now, we want to set up this thing for our communication complexity. Again, if I want to prove this thing, one thing is average case cost of mu is less than worst case cost. This is okay right this is the easy part. This is in some sense, the weak duality. Right because, this is saying that if I work, if in C cost, I can answer on every input.

That means, in C cost, I can answer on any distribution of inputs also. That is not surprising. Then, my worst case has to be bigger than this. Now, the important question is how to prove the equality, right. To prove this, then we take the LHS to be C.

Notice, when I call it cost, but it could be any measure of the algorithm. Like yesterday, it was the success probability. So, do not get confused by cost word. It need not be the communication cost. It need not be the communication bits exchange.

It is sub measure of how hard it is for an algorithm to perform. You can say that, fix an error, how much communication am I taking? That is a measure of cost. Other way to do the same thing is, you say you are only allowed C communication. How much error can you make? Right. We are doing it in the opposite way for this kind of setting. This average cost mu, this is what you want to analyze.



$\sum \mu(x)$ (cost of A on x). = Avg cost.

$\forall \mu$ Avg case(μ) $\leq$ WorstCase Cost

$\overset{?}{=}$

$\max_{\mu}$ Avg-cost(μ) $=$ C

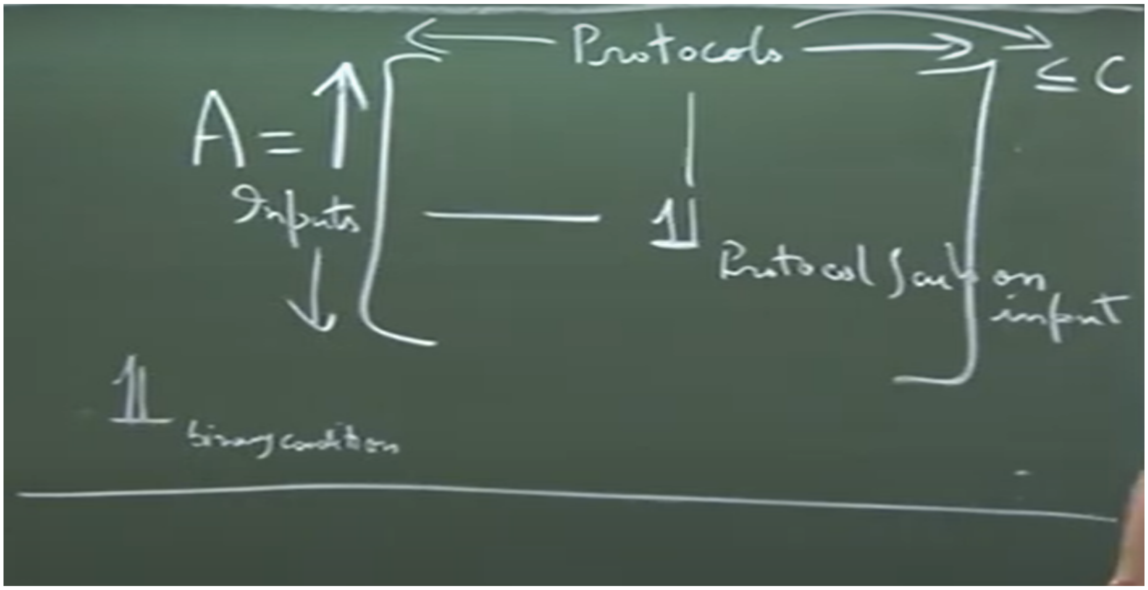
C — For $\overset{\forall}{\text{any}}$ μ, $\exists$ a protocol P. which succeeds with prob. $\geq 2/3$.

I am calling this C. okay I want to show that there is a worst case algorithm, which works with C cost. What does it mean? For any mu, there exists a protocol, a randomized protocol, probably a protocol P, which succeeds with probability greater than. This is in other words for all mu. This is the statement I am making about C. Sounds good? This is how I am defining C.

It is this quantity in other words, in this much communication for any mu, I have a randomized protocol, which is going to work. This is the definition of working. Right Now, after setting this, now I am going to do a max min. but this average cost is going to be completely different. It is not going to talk about communication. Ok I am going to set these things up, define A.

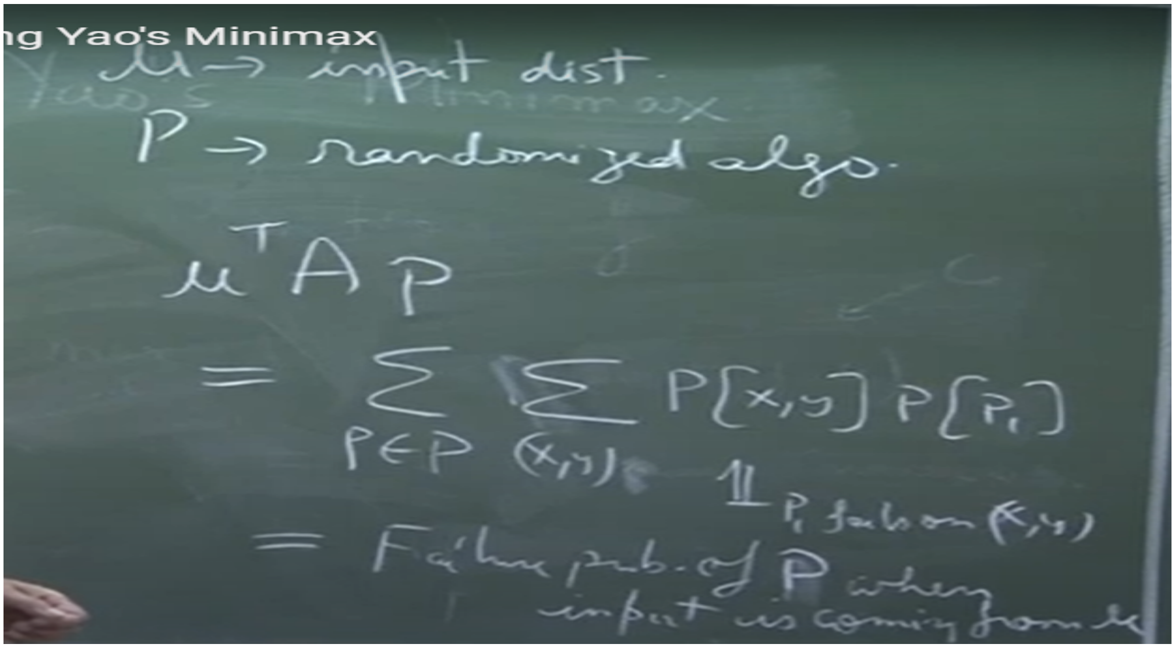
What do I want? I want inputs here, protocols. Again, when I say protocols, they have communication cost less than equal to C. Ok. Now, after defining this, I am going to view of this. I have seen amount of communication. I have fixed that amount of communication.

How much error do I have to make? Right so this is the setting up. And then just to have a change, now let us say define this to be the indicator that protocol fails on the input. right If I look at this entry, this corresponds to an input and a protocol. So, I am defining it to be 1. If it fails, I am defining it to be 0, if it succeeds.



Indicator generally, when someone writes like this, this is 1 and then some binary condition. It says that if this condition is true, then you take the value 1, otherwise 0. This is the notation. How I have defined the matrix A? Now, for any input distribution and a randomized algorithm, remember a randomized algorithm is a distribution over these protocols. What is the meaning of this? This is summation over all protocols in P, whatever is the support of P, summation over all input pairs.

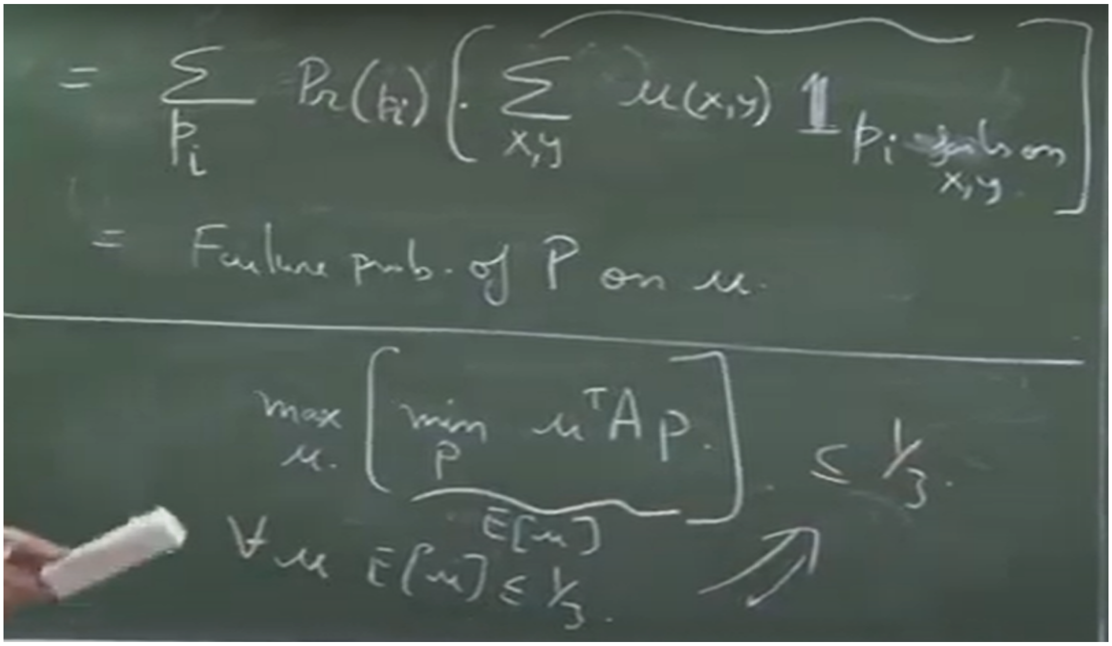
I pick the input pair, I pick the deterministic protocol and then ask P 1 fails on x comma y order. Now, for a randomized protocol, what is the probability of failing? Pick up these things with their original probability and then say what is the failure probability. If you have a protocol which fails with probability 1 by 3, if you have protocol which fails with probability half and you are taking a equal mixture of those, what is your failure probability? Half times 1 by 3 plus half times half. So, this is basically failure probability of capital P, P when input is coming from.


ng Yao's Minimax
$\mu \rightarrow$ input dist.
$P \rightarrow$ randomized algo.
$\mu^T A P$
$= \sum_{P \in \mathcal{P}} \sum_{(x,y)} P[x,y]\, P[P_i]\, \mathbb{1}_{P_i \text{ fails on } (x,y)}$
= Failure prob. of P when input is coming from μ

Notice this P is this mu x y. So, let me just write it once more time. This is summation over protocol. These are the protocols, deterministic protocols which I will be utilizing in my randomize protocol. The failure probability fails x comma y.. So, what is this? Failure probability of, what is this quantity? The failure of probability P i on mu, right?

And if I am taking P i with this much probability, then this is the failure probability over the randomized protocol. Correct? So, this is how you can parse this statement. So, and if still not clear after the class, I can explain it again. This is the failure probability. Now, knowing this, let us write down the average case complexity or average case, what do you want to do? We want to minimize over all possible mu.

Sorry, I want to maximize over mu or forget it, think about this. Minimize over all possible, if I fix mu, I want to minimize the error probability. What did the definition of C tell me? I know for any mu, there always exist a protocol which succeeds with probability 2 by 3. That means, fails with probability at most 1 by 3. That means, if I fix a mu, there exist a protocol such that this error is less than 1 by 3.

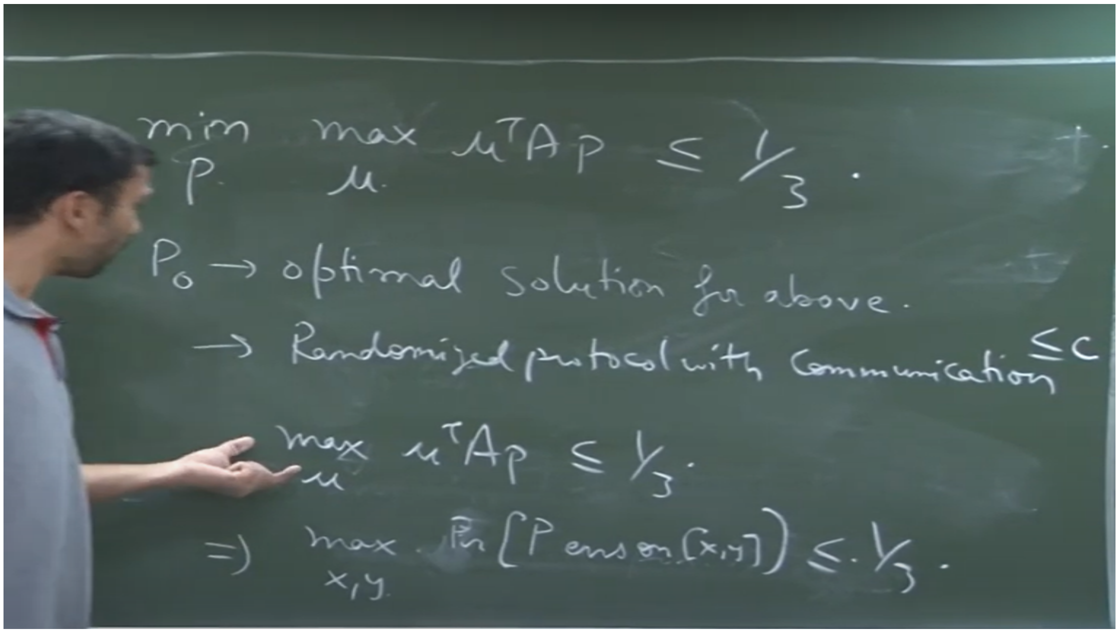

$= \sum_{P_i} \Pr(P_i) \left[\sum_{x,y} \mu(x,y) \mathbb{1}_{P_i \text{ fails on } x,y} \right]$
$=$ Failure prob. of P on μ.
$\max_{\mu} \left[\min_{P} \mu^T A P \right] \le 1/3$.
$E[\mu]$
$\forall \mu \; E[\mu] \le 1/3$.

If there is this error is less than 1 by 3, 3 for any mu. So, for all mu, this quantity let us say error with respect to mu. This implies, this is the way I am lucky. I use the definition of C and now I can erase it. So, now, this is less than 1 by 3.

This is a max min. Now, I am ready to use my Von Neumann's max min or min max theorem. So, I know now that min over P max over mu less than 1 by 3, right. This is just directly coming from duality, strong duality or Von Neumann's max. Now, once I know this, what does this tell me? Suppose, optimal here is attained by P 0.

This is the optimal solution for above. What is P 0? It is a randomized protocol. It is a distribution over deterministic protocols with cost at most C. That means, it is a randomized protocol with communication cost C. So, P 0 with communication. Right? What do I know about this? For this particular P.

Right? A simple consequence of this is max over x y probability P us on x y. After this here. Right? So, max mu transpose A P. I am saying if this is the case, let us look at all the point distributions.



$$\min_{P} \max_{\mu} \mu^T A p \leq 1/3.$$

$P_0 \rightarrow$ optimal solution for above.

$\rightarrow$ Randomized protocol with Communication $\leq C$

$$\max_{\mu} \mu^T A p \leq 1/3.$$

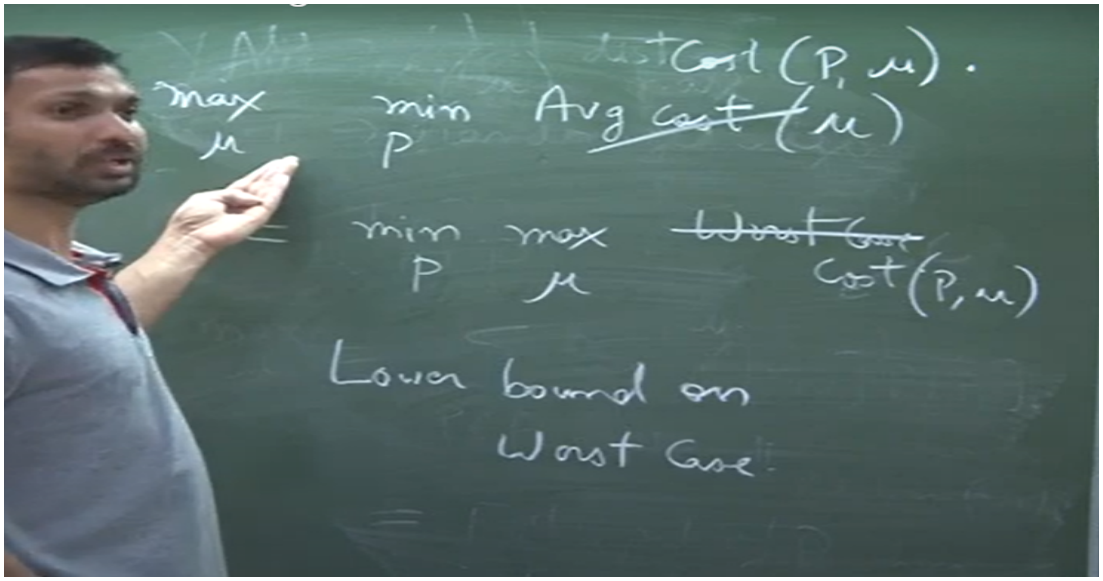
$$\Rightarrow \max_{x,y} \Pr[P \text{ errs on } (x,y)] \leq 1/3.$$

Fix mu to be a particular input x comma y. Then, this is kind of one vector. Sorry, standard basis vector. And then, what I get here is probability that P us on that particular input x comma y. In this, there is only 1 mu x y which is 1.

Everything else is 0. So, then this is the probability that P makes an error or fails on x comma y. So, now, this is saying that I have a protocol whose communication cost is less than C and it works for every input. Communication C works on every input, every distribution of input, one in the same thing. This implies now in the worst case, communication cost is less than C.

This is what I wanted to show. So, I guess the one problem in this is that I start with the cost, where the cost was in terms of communication cost. How much I was doing? And then, we define the cost as oh if the error is 2 C, how much communication do I need? But, when I want to prove this theorem, the way I have set up my matrix A, yes, it turns out to be asking the question in the equivalent, but opposite way. I say that let us fix my communication and say how less an error can I make? And then, everything works out. Make sense? The punch line again is here that the way I have defined the cost is not very important. There are multiple models, there are multiple ways in which you can set up the same experiment and you can get the similar theorem.

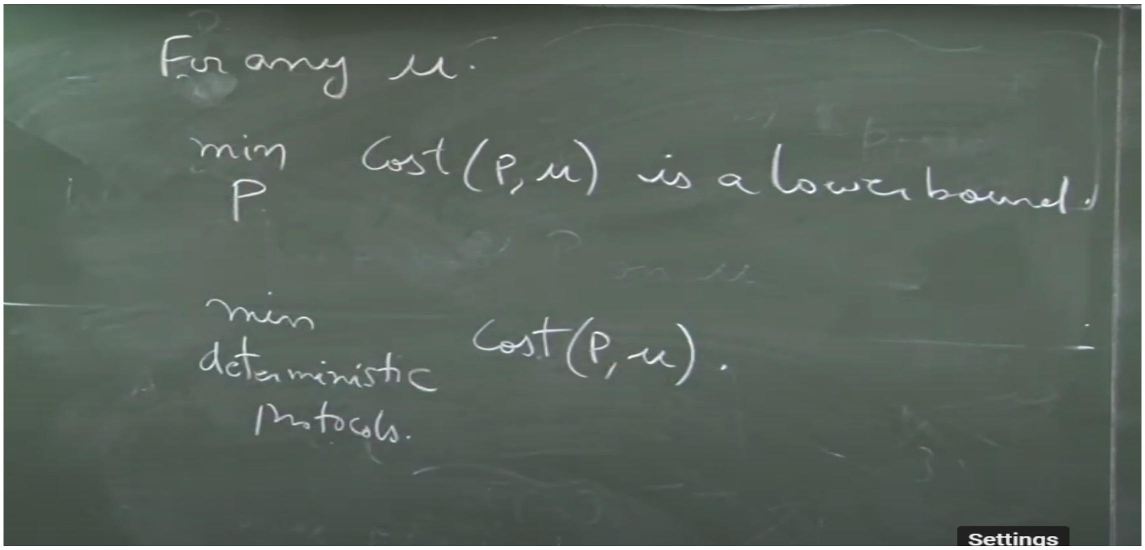
So, Yao's Minimax is not just one theorem. For many randomized models, for many settings, you can always say that the worst case complexity is equal to the maximum average case complexity. And now, once again, why is it so great? It is actually telling you something very fundamental which is great. So, let me just write it in the what was it? Min over, we will exchange if I have to reverse it. Always get it the wrong way.



This is the max average cost. I reverse it. Think of it again. Look at worst possible input. Take the best algorithm. So, the way to remember is if I am taking the cost, I want to minimize over the protocols. If I am taking the success probability here, then I want to maximize over all possible protocols.

So, yesterday we had the opposite order of min max because here it was success probability and not the cost. And the main reason why this theorem is so fantastic is because it allows us to put lower bound on worst case like we have seen. Yes, question? Max of sorry.

I should not say worst case. You are right. Ideally, I should not say. I should say cost of in some sense p comma mu and here I will say cost of p comma mu. It will actually be the same thing which is yes you are right. So, I would say cost of p comma mu and this I have not defined exactly. So, now like in the case of other duality examples, remember what happens is you want to put a lower bound on worst case.



For any μ.

$\min_{P}$ Cost(P, μ) is a lower bound.

$\min_{\text{deterministic protocols}}$ Cost(P, μ).

Duality changes it into a max problem. That means, if I can at any point for any mu min over p, cost p mu is the lower bound on worst case complexity. This is trivial. It is a maximization problem. If I take a feasible solution, I will get a lower bound. More importantly, since I am now only worried about this, do I have to worry about taking a randomized protocol? No, the minimum will be attained at a vertex.

So, then I can say min over deterministic protocols. So, to show that some problem is really hard, I just want to construct a nice mu such that every deterministic protocol not randomized protocol. Remember, I wanted to give a lower bound on randomized protocol which is a much harder thing, much bigger set, but because of this Minmax theorem, I have changed it to saying find a nice distribution. So, that even just the deterministic protocols do not work.