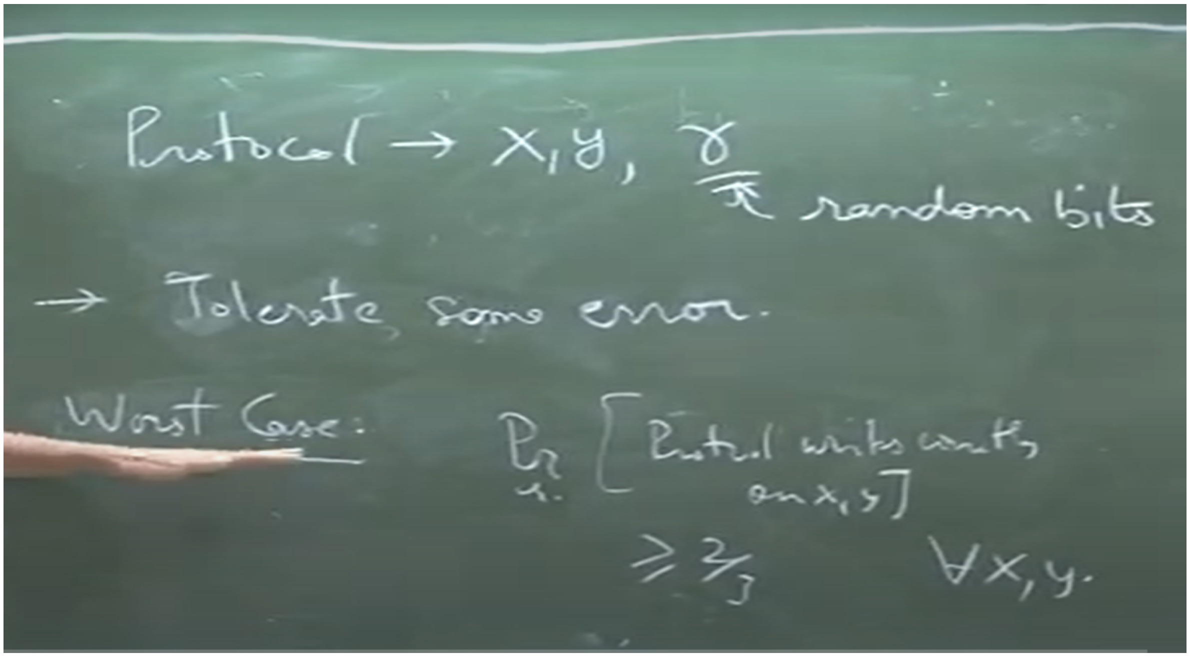
Linear Programming and its Applications to Computer Science

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

Lecture – 35
Randomized Communication Complexity

We all talk about randomized protocols. So, what does it mean? What do you mean by a randomized protocol? But how does the protocol change? Exactly. Now, my protocol will depend on X, Y and some random coin tosses. And then the entire thing this works out and I have a cost with for a particular R the number of communication bits are fixed. And then this model is only advantageous if I allow some error. If I say that I am correct only on most of the R's not on every R.

So, the difference is first I have some random bits my decision in my protocol can depend on random bits. And secondly I have an I can tolerate some error with some probability I can give wrong answer. But what do I mean by probability in this case? What is the source of probability? What is the source of randomness here? Across R. Right. So, the case we are mostly interested in is the worst case it is saying that probability over R protocol works correctly on X, Y is greater than 2 by 3.



Protocol → $x, y, \underset{\uparrow}{r}$ random bits

→ Tolerate some error.

Worst Case: $\Pr_r$ [Protocol works correctly on x, y] $\geq 2/3 \quad \forall x, y$.

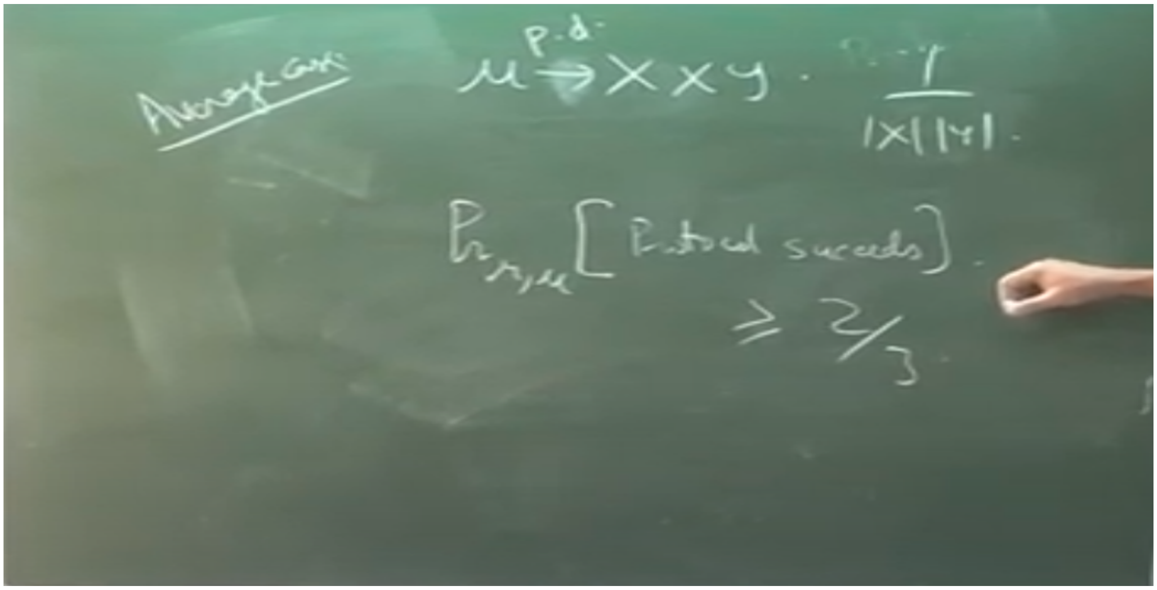
And this holds for all X, Y. This is one measure of saying my protocol works correctly. This is fine? Why 2 by 3? Any number greater than half works. You know why? Yes, any number greater than half works. Do you know why not they anyone else? Just repeat it you can reduce the error as much as you like just by constant number of repetitions.

And in most of these cases we are interested in the asymptotic is the number of bits exchanged n square root n log n things like that. So, constants are for free. So, you can repeat it constant times and change it from 2 by 3 by 4 to 99 by 100. This constant really does not matter until it is more than half. Ok. And more than half is clear right because I can always succeed with probability half.

My answer is 0 and 1 I just toss a coin and answer. Good. This is the worst case communication error or error in a communication protocol right. And a very weak version is average case where we have some input distribution in mind. Not element mu is a probability distribution on my input sets.

I assume that my inputs are coming from some distribution. The most natural probability distribution is uniform right. Pick X with equal probability from here pick Y equal probability from here right. Everything pick every pair you pick with this probability this is the most natural, but not the only one right. So, now in this case what I want is protocol succeeds notice there is no X comma Y here there is no for all X comma Y correct.

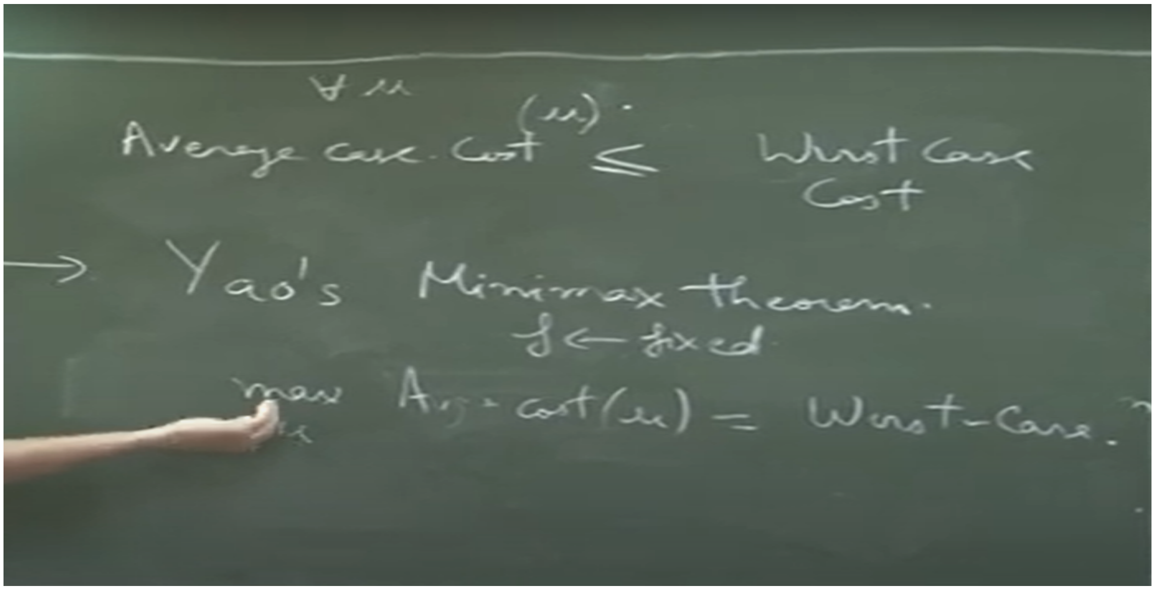
And yeah in complexity theory we mostly care about worst case, but there are people who talk about O we should talk about average case things like that also. Let us not go into the discussion, but at least see that this is harder to satisfy right. For example, if I am correct here if I have a communication protocol with cost C it works here right. Then definitely it works for average case every mu right. That means average case cost is less than equal to worst case cost.



Average case

$\mu \xrightarrow{p.d.} X \times Y$. $\frac{1}{|X||Y|}$.

$\Pr_{x,y,\mu}$[Protocol succeeds] . $\geq 2/3$


∀ μ
Average case Cost (μ) ≤ Worst Case Cost
→ Yao's Minimax theorem.
f ← fixed
max μ Avg-cost(μ) = Worst-Case.

Though this does not make sense right, because what do I mean by average case cost? I should say for all mu average case cost mu is less than worst case cost. Sure agreed do I need to prove this fine right. If I succeed on every input then I succeed on every distribution of input 2 right whatever be the distribution. So, this should ring some bell you know you should be agitated how can this be true right. You see this was a very very strong condition to hold I was saying for any input I should succeed with probability 2 by 3.

Here I am saying let us take this distribution you should you need to be only be correct with high probability on that distribution right. Think of what are the easiest distributions point distribution. So, in some sense this is saying you should only be correct on one pair of input x comma y and then any distribution on that. So, this seems much much easier than that case right. So, obviously one thing is I am maximizing over mu right.

What is the other thing here? This when I am writing this I have an input in F fixed. Given a fixed F these two cost are equal. This mu is not independent of F for a different F a different mu will be difficult. So, what this theorem is saying is you take a function the worst case complexity is equal to the average case complexity for a particular mu. So, again there is a hard mu for every F and that mu depends on F.

So, given an F there will always exist a mu such that the average case on that is equal to worst case. Why is it nice? Because giving lower bounds here is much easier than giving lower bounds here right. So, ultimately you want to give lower bounds here. Remember why were we studying communication complexity? Let us say you want to lower bound set disjointness. You want to say that this much amount of communication needs to be done for set disjointness.

But this tells us find a good mu for set disjointness. If I can prove that the average cost is high then the worst case is high. And it turns out the average cost here when I talk about average cost I can actually reduce to only the deterministic protocols. I will come to that. You have seen that you remember when we proved the Minmax theorem when you fixed x inside the min y we said y need not be random.

It has to be from the vertices only or it there is no distribution there it is the same idea here. So, this is you see that it is already max here this will become a Minmax thing. If I fix a mu this distribution the distribution on min I can skip. And this will. So, in this worst case what I am saying is look at a protocol which communicates c bits c or less than c bits.

This is a valid protocol if this is satisfied for that protocol. Yes for whatever r for whatever x and y it should communicate only c min bits. Let us consider that protocol. So, the cost of the protocol is maximum amount of communication needed for x y r. See what happens is if it is taking lot of time on some r's we said we will just cut it after some time and let it give a random answer we will have this probability less.

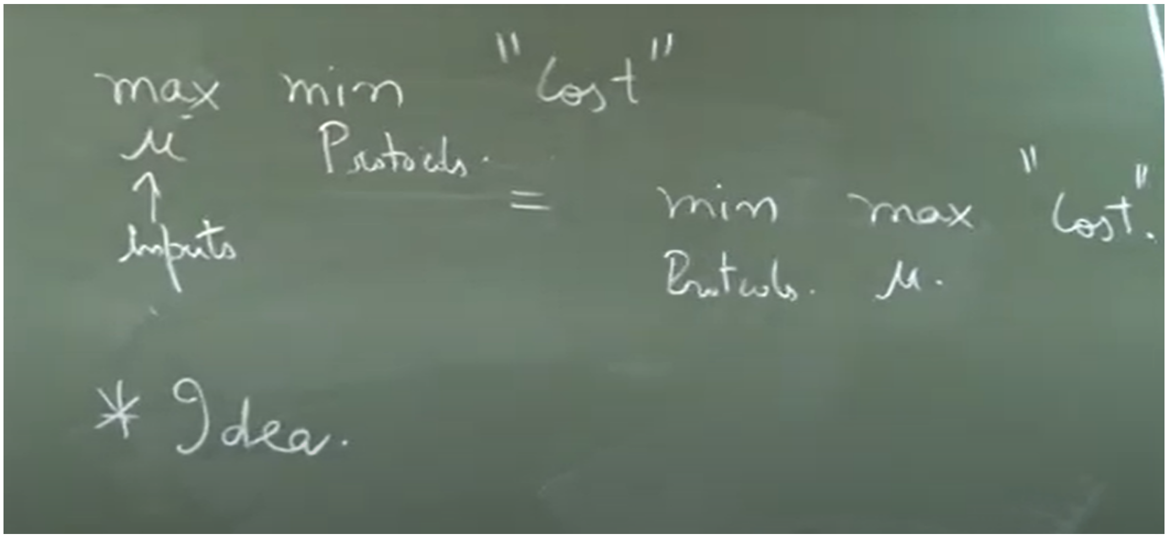
So, it need you need not worry about cases for cases when for some r's it is taking a lot of time. I mean just say I will cut out the protocol. So, the formal way to define it is first I define the cost of the protocol. A protocol is correct for f if this condition is satisfied. What is the cost of protocol worst case over all x y comma r how much communication it is doing.

Now what is the cost of a function? What is the communication cost of a function? The cost of the best protocol. For every function there are lot of protocols which is going to satisfy this. I have defined the cost of a cost of a protocol. So, this is the correctness condition think of this as indeterministic case when you say this is an algorithm for function f if your algorithm computes f correctly all the time. This is the same thing what is the complexity of sorting? You look at all the algorithms which perform correctly on sorting take the best algorithm there.

Now your notion of correctness has changed nothing else. This is the notion of your correctness look at the best protocol which is correct on the function f. So, once again the definition of correct I am changing that is all what I have changed. And like before how much cost? Cost in the worst case whatever be the input whatever is the maximum time I am taking maximum communication I am doing that is the cost of the protocol. Yes for all x y I look at my protocol I check this condition for all x y am I outputting the correct answer with high enough probability if I am then I say that my communication protocol is correct for this function f.

So other way now when I give a lower bound it will basically say that if you take a communication cost let us say I prove a lower bound of c I am saying if you take any communication protocol with cost c there has to input and there has to exist an input x comma y such that this probability is less than 2 by 3. Sounds good do I need to repeat these are all you know coming to the same point through here through here and whatever right. But this is how I want to be correct and then I optimize over all possible algorithms or protocols. Instead of algorithms we have protocols now probably I should write cost is x y and probably you might wonder why things are like this is not the only way to define correctness. This is what we generally call Monte Carlo way of defining correctness and randomize there is another thing which is where we say Las Vegas kind of correctness where the cost is expected cost over correctness.

So, there probably that is where your intuition will match more you have different cost for different randomness and you can say even if you know there might be some hours for which I am taking a lot of time. But in expectation on a particular input x y my cost is small that is another model that is called Las Vegas model of randomize complexity there also there is a maximum theorem by the way. But we are doing it in the Monte Carlo and do not ask me why it is Monte Carlo and why it is Las Vegas I do not know except both are betting places which are relate to probability, but right. So, we want to prove this right and we already see a maximum structure. So, this is going to be protocol this is a distribution over inputs correct and the same cost here this is just the idea.



$$\max_{\underset{\underset{\text{Inputs}}{\uparrow}}{\mu}} \min_{\text{Protocols.}} \text{"Cost"} = \min_{\text{Protocols.}} \max_{\mu.} \text{"Cost".}$$

$*$ Idea.

So, we have to come up with something here. So, that these things make sense this almost looks like average case complexity this will become worst case complexity this also looks like worst case complexity. I am minimizing over protocol for any distribution remember if I fix now protocol here max over mu is max over only the vertices. So, this is saying max over x y some cost. So, this looks very close to what we want to do. So, now what do we do?