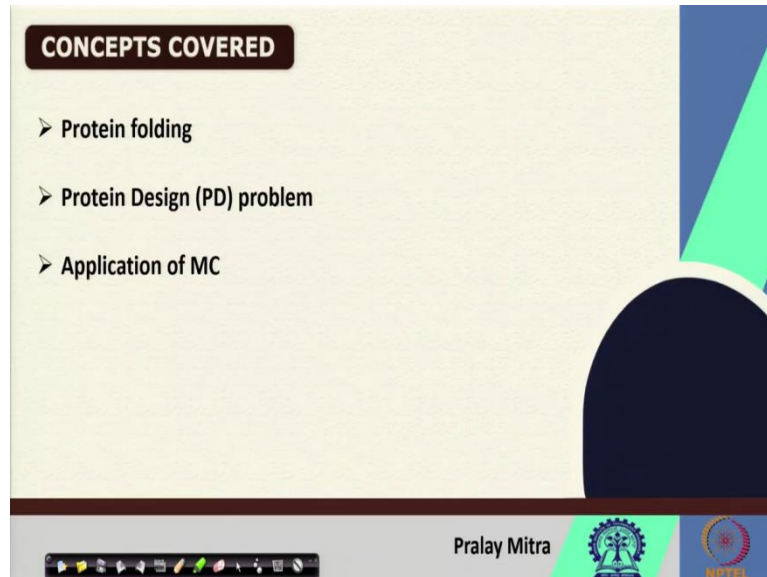


**Algorithms for Protein Modelling and Engineering**  
**Professor Pralay Mitra**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**  
**Lecture 20**  
**Protein Folding (Contd.) and Protein Design**

(Refer Slide Time: 00:22)

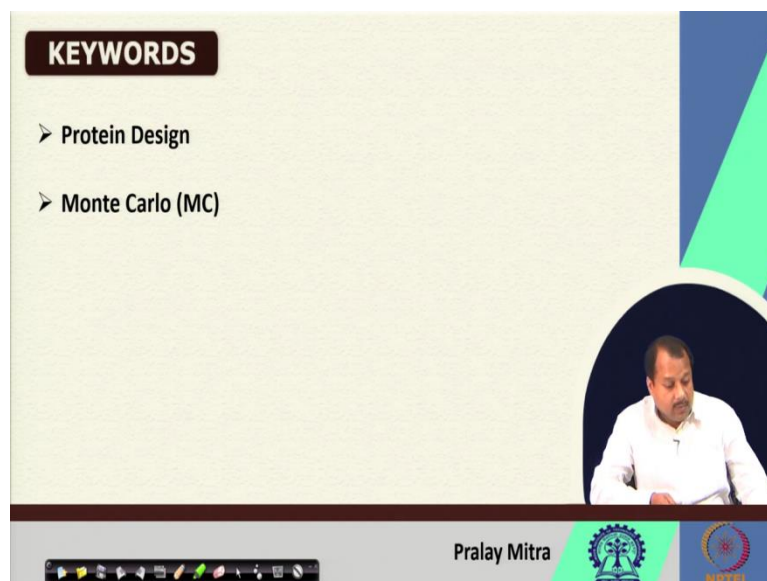


**CONCEPTS COVERED**

- Protein folding
- Protein Design (PD) problem
- Application of MC

Pralay Mitra

The slide features a light beige background with a dark blue and green geometric design on the right side. A dark blue semi-circle is positioned at the bottom right. The footer includes a navigation bar with icons, the name 'Pralay Mitra', and logos for IIT Kharagpur and NPTEL.



**KEYWORDS**

- Protein Design
- Monte Carlo (MC)

Pralay Mitra

The slide features a light beige background with a dark blue and green geometric design on the right side. A video inset in the bottom right corner shows Professor Pralay Mitra speaking. The footer includes a navigation bar with icons, the name 'Pralay Mitra', and logos for IIT Kharagpur and NPTEL.

Welcome back. So, we will be continuing protein folding and then we will introduce a protein design again, the concept that we will be covering is protein folding. We will give the definition of the protein design problem and the application of Monte Carlo in this context. We will be discussing accordingly is the key word here.

(Refer Slide Time: 00:39)

**MC Simulation - Method**

**Algorithm 6:**

**Input:** Number of search steps,  $c$  (current conformation),  $v$  (search neighborhood)

**Output:** Modified conformation [Energy optimized]

**Steps:**

- For each step do  
 $k \leftarrow U(1, n);$   
 $c' \leftarrow M(c, k, v);$   
 $\Delta E \leftarrow E(c') - E(c);$

Handwritten annotations:  
-  $c$ : current  
-  $c'$ : new conformation  
-  $E(c)$ : energy of the current conformation  
-  $E(c')$ : energy of the new conformation

Pralay Mitra

So, the MC simulation method the Monte Carlo simulation method in the context of say protein design and folding. The same backbone or the same architecture of the method will be used all the thing is that. The input will change the score function will change and also the other parameters like temperature with Metropolis criteria those things will be changed.

So, I will mark all those points during the discussion and then you will see that it is the same algorithm which will be used for the protein design as well as the protein folding problem. So, that is a good thing also. So, number of search steps. The current conformation and the search neighborhood is given as an input and you have to output a modified conformation.

Which is not mentioned clearly here is in this modified conformation. So, you have to give one energy optimized one mostly the energy minimum indicates that it is stable. But based upon the definition of the designer who is designing the energy function. It can be the higher the energy it is stable and as such that is no problem.

So, if it is the higher the energy more stable it is and you multiply a negative with that one then it will be lower the energy more stable it will be. Now, algorithm is very simple, it says for each step do  $k \leftarrow U(1, n)$ . So, randomly you generate one number from 1 through  $n$ . So, what is  $n$ ?  $n$  is number of possible conformations.

So, from number of possible conformations, you generate pick one number. So, what I will be doing here is that. So, from 1 through  $n$  randomly I will generate one number after generating that number here there are in conformations in my database. So, the conformations

how I am generating that conformation those information I deposited there and I discussed on the last lecture.

So, it is in the database. And I am assuming that as if there is an index 1 through n through which I can access one conformation. So, there are N such conformations and I am also not ruling out the situation. That say residue level conformation there are in then say segment level M number of conformation are there.

Then p number of say topology level conformations are there and N multiplied with M multiplied with p total number of conformations are there. It is fine. But randomly first I will generate one random number. And I will assume that random number is going to be the index one particular say orientation or that random number will indicate that which conformation to pick from the neighborhood of this new.

So, in my  $M \times k$ ,  $c_k$  and new are the arguments. So,  $c$  indicates the current conformation,  $k$  indicates the random number that I generated right now and new indicates. What is my neighborhood information based upon that one it will output another conformation  $c'$  which is a new conformation.

Now, I have two conformations one is  $c$ . This  $c$  is the current and  $c'$  that is my new. So, when I say  $c$  is my current conformation. So, corresponding to that current conformations there was some energy value. Now, with the  $c'$  the new one I will calculate another energy value. So, that I am assuming  $E$  is my energy value.

So,  $E_{c'}$  will give me the new energy or better to write energy of the new conformations and  $E_c$  indicates the energy of the current conformation. So, the difference between these two new and current is  $\Delta$ .

(Refer Slide Time: 05:59)

Now, if this is delta then you see. This delta is difference of the energy between the new conformation and the current confirmation. Now, you see if my assumption is the lower the energy more stable it is. So, if this is of lower energy compared to  $E_c$ , then what will happen. This delta  $E$  will be negative.

So, if delta  $E$  is less than equals to 0 indicates that  $E_{c'}$  is less than  $E_c$ . And if it is greater than 0. This implies  $E_{c'}$  is greater than  $E_c$ . So, for the second case, when the energy of the new conformation is higher. That is not a better solution compared to the current conformation.

So, I will not consider that one I will consider only delta  $E$  less than 0 equals to 0 that is what is here? If delta  $E$  is less than 0, then  $c$  equals to  $c'$  or the new conformation is going to be the current conformation. Now, of course, during or implementation, you can store the previous energy.

So, that at each step, you need not have to compute your previous energy. So, that is the implementation detail. But algorithmically this indicates that if the energy is less than 0, then I will consider that one otherwise I will not consider that as of now, I did not discuss the last one. So, if delta is less than equals to 0, then  $E_{c'}$  indicates less than  $E_c$ .

So, I will consider that one. So,  $c'$  is going to be  $c$  which means the new conformation is going to be the current conformation. Now, that is one strategy and using that strategy it is true if I ensure that, my energy function is perfect one. So, last time during the calculation of

$\pi x^2 + y^2 \leq r^2$  that was the correct theory and it is the perfect one.

So, I do not have any problem, but if it is the case that my  $E$  or energy function is an approximate one. So, you remember, so we discussed the  $f(x)$  and  $g(x)$  we are approximating. So, given equation was  $f(x)$  and I am approximating to  $g(x)$ , and if you remember that lecture then I mentioned and in the schematic diagram also it was such that on the blue there was a red line and in some position, it was more compared to  $f(x)$  than  $g(x)$ , in some position it is same as the  $f(x)$  and  $g(x)$ .

Now, if the situation is something like that, then I cannot completely rely on my energy function. And if it is the situation, then perhaps it is fine. If  $E_{c'}$  is less than  $E_c$ , then definitely I will consider  $c'$ . But if  $E_{c'}$  greater than  $E_c$ , and say that difference is very small amount or something, then probably it is not a good idea to reject that at all.

Or, probably it is not a good idea to reject that completely. For that, what is the proposal is you generate another random number here in between 0 and 1 then you compute this value. What is this value? It says this  $e$  is exponential minus  $\Delta E$  by  $T$ . It is assuming that the MC simulation method is following boltzmann distribution and if it is following the boltzmann distribution there exists one temperature  $T$ .

This temperature you please remember that it is nothing to this  $T$  is temperature. This  $T$  indicates that the temperature. So, it assumes that Monte Carlo simulation technique is following the boltzmann distribution and the temperature factor in that boltzmann distribution is  $T$  then it computes the exponential of minus  $\Delta E$  by  $T$  and then it is checking whether the random number.

I am generating is greater than this  $e$  to the power minus  $\Delta E$  by  $T$  or not. If it is greater than then with some probability even though the energy function of this  $E_{c'}$  is greater than  $E_c$  else part. Then also I am accepting the new conformation as the current conformation. So, in summary there is one energy function  $e$  but I know that energy function is kind of approximate function like  $g(x)$  on  $f(x)$ . So, the correctness of the  $f(x)$  I cannot.

So, the correctness of the  $e$  as a protein energy function is not verified or validated. So, that is why when say new conformation evaluates an energy which is lower than the current conformation energy then I am accepting. But if it is not then I am not completely rejecting, I

am accepting with some probability. That probability varies from 0 through 1 and if  $e^{-\Delta E/T}$  is lower than that random number  $q \in [0, 1]$ .

One random number  $I$  generated then I am accepting that one. And what is the theory behind the  $e^{-\Delta E/T}$ . So, it says that the MC or Monte Carlo simulation is assuming a following the boltzmann distribution and the temperature in this case is the temperature of the boltzmann distribution. Please note that this temperature is nothing to do with the temperature of the protein of the environment of the protein or the solvent of the protein.

Where the solvent where the protein is this temperature is nothing to do with that one, this temperature is about the distribution of this conformation. If you change these temperature, then accordingly the acceptance rate will change. So, by controlling this  $T$  you can control the acceptance rate on top of what will be accepted by this if condition because it will go to the else part.

(Refer Slide Time: 14:12)

**MC Simulation – Method (contd.)**

Steps:

```
 $\Delta E \leftarrow E(c') - E(c);$   
if( $\Delta E \leq 0$ )  
   $c \leftarrow c'$ ;  
else  
   $q \leftarrow U(0,1);$   
  if( $q > e^{-\Delta E/T}$ )  
     $c \leftarrow c'$ ;
```

Metropolis Criterion

Temperature

Pralay Mitra

IIT Bombay NPTEL

### MC Simulation – Method (contd.)

Steps:

```

 $\Delta E \leftarrow E(c') - E(c);$ 
if( $\Delta E \leq 0$ )
   $c \leftarrow c'$ ;
else
   $q \leftarrow U(0,1);$ 
  if( $q > e^{-\Delta E/T}$ )
     $c \leftarrow c'$ ;

```

Metropolis  
Criterion

Temperature

So, this is called as the Metropolis criterion. So, which one this one. So, up to this if part it was a simple Monte Carlo simulation. If it is less than accept otherwise no need to accept. But after that one what is there. So, it will be through the Metropolis Criterion. Now, combining these two we are getting this Monte Carlo Metropolis criterion.

And the temperature again, this T is the temperature. This T is the temperature. This T again is nothing to do with the physical temperature or the temperature of the protein's environment. This temperature is related with the Monte Carlo simulation technique. It assumes that the solution space which is being explored by this search technique.

Monte Carlo simulation this search technique is following boltzmann distribution and temperature is there. So, by changing this T definitely you can control the acceptance rate. Because you see that if it always gives you very high value, say 0.95 is q greater than this one.

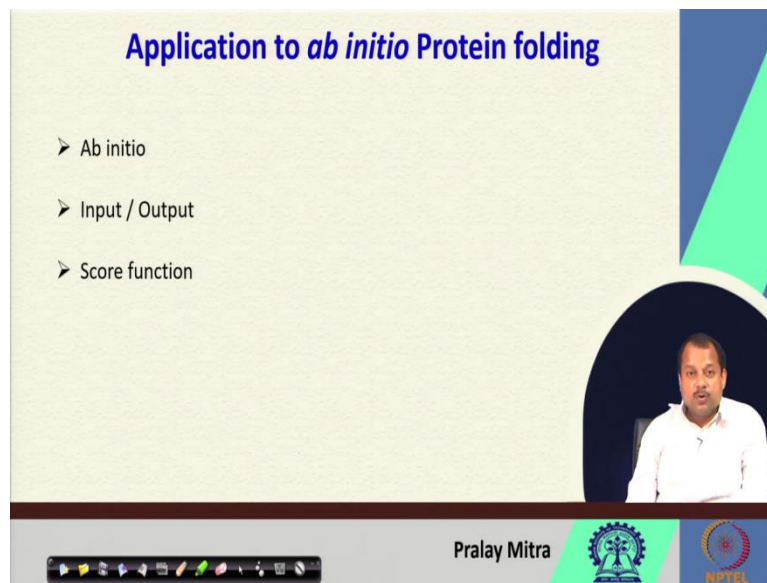
So, only in number of cases when this random number will generate more than 0.95, then only it will accept. On the other hand, if it is a 0.05 then almost everything will be accepted even the negative, even which is higher in energy most of the time that will also be accepted.

So, what will be your acceptance rate or how much say higher energy you will approve or allow will be controlled by this equation. So, this is the Simulation technique. Now, you have to incorporate that conformation here and one energy function that is it, then you can actually have one protein folding problem.

(Refer Time Slide: 16:31)

### Application to *ab initio* Protein folding

- Ab initio
- Input / Output
- Score function



Pralay Mitra

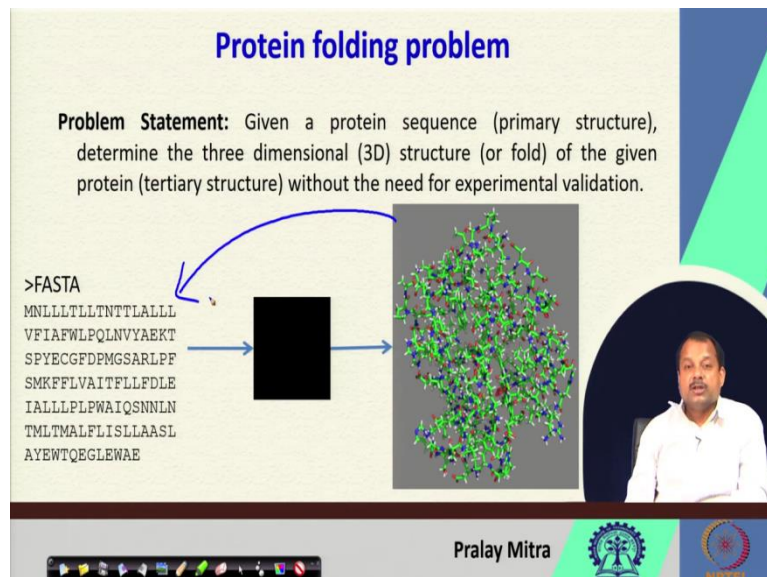
So, the *ab initio* technique that I mentioned is here, when we are starting from the very scratch, when we do not have any knowledge. So, at the residue level, we are identifying the conformation then at the segment level then at the topology level, I am doing that one. So, we are considering those. Then input output that we have discussed now, the score function is remaining. So, that we will discuss later when we will discuss about one protein folding framework.

(Refer Time Slide: 17:03)

### Protein folding problem

**Problem Statement:** Given a protein sequence (primary structure), determine the three dimensional (3D) structure (or fold) of the given protein (tertiary structure) without the need for experimental validation.

```
>FASTA
MNLTLTLLTNTTLALLL
VFIAFWLPQLNVYAERT
SPYECGFDPMGSARLPF
SMKFFLVAITFLLFDLE
IALLLPLPWAIQSNNLN
TMLTMALFLISLLAASL
AYEWTQGLEWAE
```



Pralay Mitra



## Protein folding problem

**Problem Statement:** Given a protein sequence (primary structure), determine the three dimensional (3D) structure (or fold) of the given protein (tertiary structure) without the need for experimental validation.

```
>FASTA
MNL L L T L L N T T L A L L L
V F I A F W L P Q L N V Y A E K T
S P Y E C G F D P M G S A R L P F
S M K F F L V A I T F L L P D L E
I A L L L P L P W A I Q S N N L N
T M L T M A L F L I S L L A A S L
A Y E W T Q E G L E W A E
```

Pralay Mitra

## Protein design problem

**Problem Statement:** Given a protein structure (and hence the protein sequence is also known to you), you need to identify another protein sequence that folds to the given (input) protein structure.

```
TEFARSEGASALASVNP L K T T V E E A L S R G W S V K S G T G T E D
A T K K E V P L G V A A D A N K L G T I A L K P D P A D G T A D I T L T F T M G
G A G P K N K G K I I T L T R T A A D G L W K C T S D Q D E Q F I P K G C S R
```

Amino Acid Sequence

Novel Amino Acid Sequences

Protein 3D Structure  
PDB ID: 1X6Z\_A

Pralay Mitra

So, this was the protein folding problem that we have discussed. Now, there is another problem, which is called as the inverse protein folding, which means that given this as input you need to generate the sequence. So, that particular problem is called as the protein design problem or it is sometimes called as the inverse protein folding problem.

(Refer Time Slide: 18:04)

**Protein design problem**

**Problem Statement:** Given a protein structure (and hence the protein sequence is also known to you), you need to identify another protein sequence that folds to the given (input) protein structure.

TEFARSEGASALASVNLKTTVEEALSRGWSVKSOTGED  
ATKKEVPLGVAADANKLGTIALKPPADGTADITLFTMG  
GAGPKNKGGIITLRTAADGLWKCTSDQDEQFIPKGCGR

Amino Acid Sequence

Novel Amino Acid Sequences

Protein Design/ Inverse Protein Folding

Protein 3D Structure  
PDB ID: 1X6Z\_A

Pralay Mitra

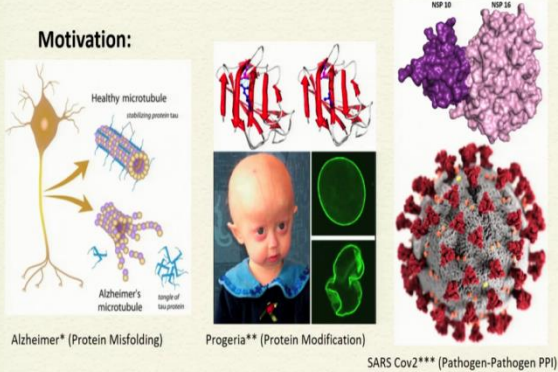
The problem statements says, given a protein structure and hence the protein sequence is also known to you, you need to identify another protein sequence that folds to the given protein structure. So, protein sequence says that here this is my amino acid and you predict this three dimensional structure. Protein design says, this is my three dimensional structure. And since you know the structure, then definitely the sequence is also known to you.

But you have to come up with another sequence which is not the sequence of the structure, but, that sequence must also fold to the structure. So, there is one sequence which will assume this fold structure, you need to come up with another sequence which will also assume to this structure. Then, there will be multiple sequences. So, it assumes the existence of the multiple sequences, which will fold to the same structure. The protein design problem assumes that one. And it has a lot of applications also.

(Refer Time Slide: 18:58)

### Protein design problem

**Motivation:**



Alzheimer\* (Protein Misfolding)    Progeria\*\* (Protein Modification)    SARS Cov2\*\*\* (Pathogen-Pathogen PPI)

\*<https://alzheimersnewstoday.com>    \*\*\*<https://en.wikipedia.org> (2012): 1134-1140.

Pralay Mitra

---

### Protein design problem

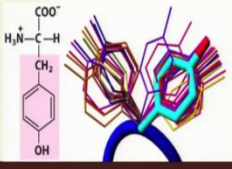
**Challenges:**

- Protein Design is NP-Hard<sup>1</sup> (N.A. Pierce and E Winfree)

20 Amino Acids

YQTRAEVSEALALAEGLKLMVSDALSNGLPSNGTGTKDSTKKSAGIGQPATKGYGTVTVDGDPSTASGGCTITITMLGANHEIKGKTIITLATAESGLWYCGSIDTKFVPSGCSN

Exponentially large numbers of possible sequences:  
 $20^N$  for an  $N$ -residue protein



Many Rotamer states

*Side chain orientations*

<sup>1</sup>Pierce NA, Winfree E. Protein design is NP-hard. Protein engineering. 2003 Oct 1; 15(10):770-87.

Pralay Mitra

So, if I talk about that one. So, a lot of protein related problems, I mean protein related disease, like Alzheimer occurs because of the protein misfolding. Progeria protein modification then SARS Cov2 Pathogen-Pathogen Protein Protein Interactions. So, all are the problems which can be captured if we can understand this protein design problem correctly.

Now, there are few challenges and the challenges are severe in protein design problem. First of all, it is NP-Hard, it is proved that the protein design problem is NP-Hard. There are 20 amino acids. So, theoretically exponentially large number of possibilities  $20^N$ , where  $N$  is the number of amino acid possibilities exist.

So, out of those possibilities which are relevant, apart from that one different rotamer states. Rotamer means different side chain orientations. So, rotamer means different side and orientations also exist. So, accommodating all of them is not a simple problem. So, that is why it is a very challenging problem.

(Refer Time Slide: 20:26)

**Application to *ab initio* Protein design**

$i \leftarrow \text{seq} \quad E(\text{seq}) \quad (E_1)$

$i+1 \leftarrow \text{seq}' \quad E(\text{seq}') \quad (E_2)$

$\Delta E = E_2 - E_1$

if  $(\Delta E \leq 0)$  accept  $\text{seq}'$

also Metropolis

At some positions  $\text{seq}$  is mutated to get  $\text{seq}'$

Pralay Mitra

IIT Bombay NPTEL

So, in case of *ab initio* protein design problem, we try to start from a protein structure and identify what is the protein sequence, then we look only the length of the protein. Say for example, if one protein is given whose length is say 220, then we know what is the sequence fine. Then we start with random 220 amino acids. When it is random, it does not mean that all 220 amino acids will be say alanine or cysteine or say truly.

But randomly at each position we generate out of 20 some amino acid and we will start from there. After starting from there, that is my one instance after having that instance, then what we do is, we evaluate the sequence with respect to that structure that what will be its energy function. We randomly muted in some position and that way we get another sequence we evaluate that is my  $c$  prime.

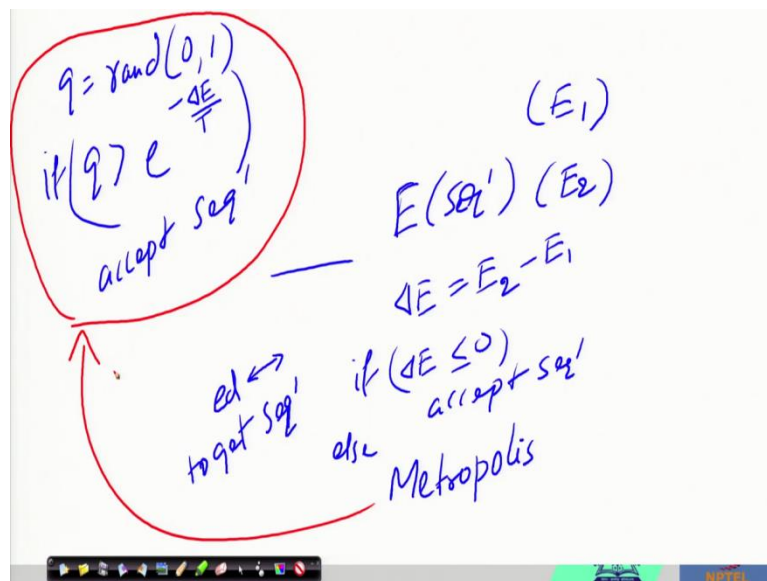
So, that  $c$  prime will also be again evaluated and when that will be evaluated, then I will get another energy function. So, we are starting with some random sequence length is known

only that is 220. At  $i$ th step. I have one sequence  $seq$  and for that I computed the energy also. Now at  $i + 1$  step what I did the  $seq$  has modified and what is the modification at  $seq$  prime the  $seq$  prime is  $seq$  mutated at some position  $seq$  is mutated to get  $seq$  prime.

And when I say mutated, mutated it means that at some location again that location will also be identified randomly at some location or at some random location that particular amino acid will be changed by another amino acid again randomly. So, two random situations, one randomly I will go to one position and then I will change that amino acid by some random amino acid then I will get  $seq$  prime.

So, I will evaluate this  $seq$  prime also. So, two energy function I am getting if I say this is my  $E_1$ , this is my  $E_2$  then I know the rule  $E_2 - E_1$ , if  $\Delta E \leq 0$  then accept  $seq$  prime else. What I need to do Metropolis criteria. What is the Metropolis criteria? So, I am going to a new page what is the Metropolis criteria?

(Refer Time Slide: 24:39)




### Application to *ab initio* Protein design

- Ab initio  $i \leftarrow \text{Seq } E(\text{Seq})$
- Input / Output  $i \leftarrow \text{Seq}'$
- Score function

*At some positions Seq is mutated*

Pralay Mitra



So, first I need to generate a random number say  $q$  rand 0 through 1. If  $q$  greater than  $e^{-\Delta E / T}$  then accept  $\text{Seq}'$ . So, this is my Metropolis criteria. So, this is all about the *ab initio* protein design problem. Now, let us go back to the previous one. So, *ab initio* that I have discussed input output that I have discussed and the score function also I discussed.


(Refer Time Slide: 25:40)

### Parallelism in Monte Carlo Methods


- Monte Carlo methods often amenable to parallelism
- Find an estimate about  $p$  times faster

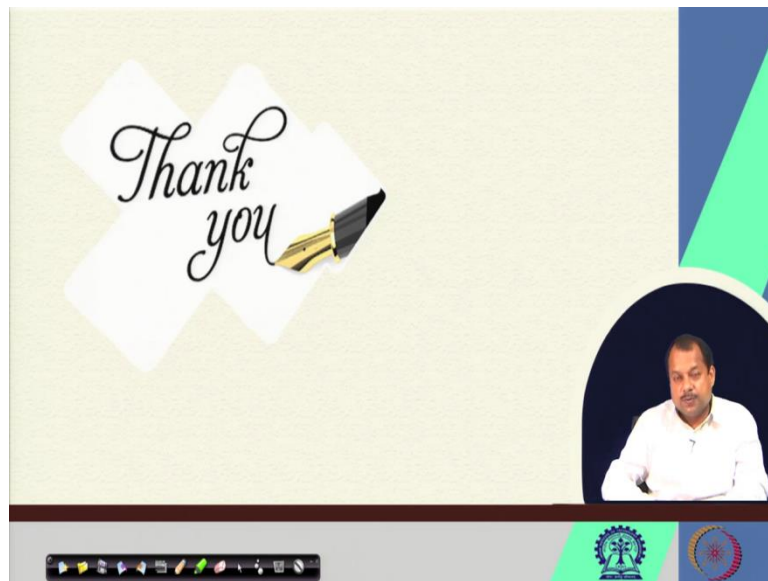
OR

- Reduce error of estimate by  $p^{1/2}$



Pralay Mitra





Now, one final thing regarding this Monte Carlo method that I do not resist to discuss is its inherent parallelism. So, now a days a lot of computers and systems of there, which has multicore. So, more than one processing units are there in one computer even for the laptop and desktop also it is there even in your smartphone.

So, is it possible to exploit that one yes, it is possible because if you look at the algorithm for the Monte Carlo simulation, then you will see that separately you can explore the space and then you can combine their results in order to get the final result.

And parallelism can be exploited one there is no data dependency. Yes, there is some dependency that at  $i$ th position it depends upon  $i + 1$  sequence and its energy. But if you assume that you need to explore say  $n$  number of steps. Now, you are given four core or four processing unit.

So, you can divide  $n$  into four parts  $n$  by 4 each. In each part you run it separately and then you combine that result for final inference, that is possible. And that way you can exploit the parallelism which means that you can gain some speed up, and you can exploit the existing architecture. So, that inherent parallelism is there in Monte Carlo simulation methods.

Monte Carlo method often amenable to parallelism, find an estimate about  $p$  times faster or reduce error of estimate by  $p$  to the power  $1$  by  $2$ . So, this is  $1$  by  $2$ , this is to the power  $1$  by  $2$ . So, these facilities are there.

So, what we discussed in this week to summarize that we started with the Monte Carlo simulation technique. Now, in the Monte Carlo method at the core, there is a random number

generation and we have to make sure that the random number is as much random as possible. We gave the first example on computing the value of pi. And during that time, we mentioned that, how the biasness in the random number can affect our result when we extended the result, or when you extended the concept for computing the integration of some function say  $f(x)$ .

Then, we see that the sampling may vary and the sampling quantum or say the region of the sampling will change based upon that in which region you are sampling, there may be a requirement of very dense sampling or very sparse sampling. And when we look at the folding funnel or the energy landscape of the protein for which we will have more detailed discussion.

We see that folding funnel is also very rugged in nature. So, we have to be careful for that one. And also, during the integration process, we noted that if it is the case that we are not able to compute the actual function  $f(x)$ , then we can approximate that one to some  $g(x)$  and if we approximate to some  $g(x)$  then sometimes we are losing some information that is why in the modified version of the Monte Carlo technique, that is the Metropolis criteria is introduced.

Where if one at say  $i$ th situation, I am getting a lower energy state, then I will definitely accept but if I am not getting lower energy state, then I will not reject completely. But I will accept with some probability value. That probability value will be controlled by one temperature factor that is the Boltzmann temperature assuming that Boltzmann distribution.

It has nothing to do with the physical environment of the protein and its temperature. And we also noted the protein design and protein folding problem we define those two problem and we see summarily that what is the application of the Monte Carlo technique in order to solve those problems. Thank you so much.